Monograph:

Tokhi, M.O., Hossain, M.A., Baxter, M.J. et al. (1 more author) (1994) Heterogeneous and Homogeneous Parallel Architectures for Real-Time Adaptive Active Vibration Control. Research Report. ACSE Research Report 532 . Department of Automatic Control and Systems Engineering

A Modified Fuzzy ARTMAP Architecture for the Approximation of Noisy Mappings

S. Marriott 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 522

June 1994

A Modified Fuzzy ARTMAP Architecture for the Approximation of Noisy Mappings

S. Marriott 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

A neural architecture, fuzzy ARTMAP (Carpenter et al, 1992), is considered here as an alternative to standard feedforward networks for noisy mapping tasks. It is one of a series of architectures based upon adaptive resonance theory or ART (Carpenter et al, 1991a; 1991b; 1992). Like other ART based systems, fuzzy ARTMAP has advantages over feedforward networks and is especially suited to classification-type problems . Here it is used to approximate a noisy mapping. Results show that properties which confer useful advantages for classification problems do not necessarily confer similar advantages for noisy mapping problems. One particular feature, match-tracking, is found to cause over-learning of the data. A modified variant is proposed, without match-tracking, which stores probability information in the map field. This information is subsequently used to compute output estimates. The proposed fuzzy ARTMAP variant is found to outperform fuzzy ARTMAP in a mapping task.

1. Introduction

Mapping approximation is an important area of research which has widespread application in many fields. While the use of standard curve-fitting techniques continues, investigations into the automation of approximation methods contribute to such fields as those of autonomous decision support and adaptive control. Any adaptive system operating within an information environment has to be capable of interpreting aspects of that environment in order to respond appropriately.

Artificial neural networks offer a possible approach to this interpretation problem in that they provide a means of approximating noisy mappings. There is a body of work relating to the ability of feedforward networks to learn arbitrary mappings (Cybenko, 1989; Funahashi, 1989; Hornik *et al*, 1989; Girosi and Poggio, 1990; Park and Sandberg, 1991; Cardaliaguet and Euvrard, 1992; Ito, 1992; Hornik, 1993). Both Cybenko (1989) and Funahashi (1989) provide proofs that a layered feedforward network consisting of one hidden layer is capable of approximating any continuous function under some mild conditions. These fundamental results have been built upon and extended by various authors (e.g. Ito, 1992; Hornik, 1993).

Although indicating the capabilities of feedforward neural networks, these theoretical results give rise to practical problems such as the determination of the number of nodes in the hidden layer(s) (Fujita, 1992) and increasing the network information capacity during operation. Both of these problems stem from the nature of the feedforward architecture which distributes information pertaining to the mapping throughout the network. The global architecture, coupled with localised error-correcting learning mechanisms, does not allow new information to be incorporated into the network following training. If further data is added to the original training set then re-training with the augmented data set is required. Ascertaining the optimum network configuration from the outset is an empirical process and, once established, the network size is fixed (Fujita, 1992).

Another cause of inaccuracy in feedforward networks is the problem of local minima of the error function (Baba, 1989, Lippmann, 1987). As the network state vector follows a learning trajectory through error-weight space it can become trapped in states which are stable but are not the global minimum for the cost function. These states, or local minima, constitute undesirable solutions of the mapping approximation problem. Without *a priori* information it is impossible to distinguish between these local minima and the desired global minimum.

The commonly used learning algorithm for feedforward networks is the *gradient descent* or *error back propagation* algorithm (Rumelhart *et al*, 1986) which attempts to minimise the mean square error energy function by adjusting the network weight vector according to the method of steepest descent. The main problems of this method are the inability to predict convergence in advance and, assuming convergence, whether the resultant approximation is sufficiently accurate (Cardaliaguet and Euvrard, 1992; Van Ooyen and Nienhuis, 1992).

This paper considers an alternative architecture, fuzzy ARTMAP (Carpenter *et al*, 1992), which has inherent properties that offer a possible solution to some of the problems encountered by conventional feedforward networks. Modifications to the original architecture are also proposed in the form of a variant which is identified as PROBART to distinguish it. The performance of both the original architecture and its proposed variant are assessed in a complex mapping task.

Fuzzy ARTMAP is one of a class of neural network architectures developed by Carpenter, Grossberg and co-workers based upon adaptive resonance theory (ART) (Grossberg, 1980; Carpenter and Grossberg, 1987a, 1987b, 1989; Carpenter *et al*, 1991a, 1991b, 1992). It is capable of mapping subsets of $\Re^m$ to subsets of $\Re^n$, accepting both binary and analogue inputs in the form of pattern pairs. It is also possible to code inputs according to their degree of fuzzy set membership.

As an extension of ARTMAP (Carpenter *et al* 1991a), fuzzy ARTMAP (Carpenter *et al*, 1992) makes use the operations of fuzzy set theory (Zadeh, 1965; Kosko, 1992), instead of those of classical set theory to govern the dynamics of ARTMAP. ART-based systems offer certain advantages over other neural network architectures, such as multilayer feedforward networks. These include the dynamic allocation of nodes without network disruption, fewer training cycles required to reach acceptable levels of predictive accuracy and guaranteed convergence (Carpenter *et al*, 1991,1992). The latter property results from the use of monotonically decreasing weights which also ensures stable learning.

The fuzzy ARTMAP system is especially suited to classification problems (Fu, 1994) and is capable of learning autonomously in a non-stationary environment. On-line learning is possible with a distinction being made between rare but significant events and more common associations. The representation of pattern associations by individual nodes facilitates rule extraction in the form of if-then relations (Carpenter and Tan, 1993). Another property of fuzzy ARTMAP is its ability to resolve sub-classes by dynamically increasing the stringency of class membership conditions when mis-classification occurs.

An important area of concern encountered in conjunction with autonomous learning systems is that of the stability-plasticity dilemma (Carpenter and Grossberg, 1987a). This term is given to the fundamental conflict between plasticity, which enables a system to learn new associations, and stability, which buffers the system against continuous recoding by establishing stable states. ART-based systems attempt to resolve this dilemma by maintaining a balance between familiar and novel input patterns. Responsiveness to novel inputs can become problematic when dealing with noisy function/mapping approximations because of inherent uncertainty as to what constitutes novelty, i.e. could a conflicting association be a rare event or simply transient noise? There is no easy solution to this problem and compromises abound in the field of neural networks which range between the two extremes of conservatism and responsiveness. In this context neural networks can be seen as filters which can do little to reduce the effects of noisy disturbances if responsiveness to novel inputs is high or fail to register significant rapidly changing events if responsiveness is lowered in order to avoid the network being swamped by noise. This is an on-going problem likely to be faced by developers and users of neural network architectures.

The modifications to the fuzzy ARTMAP system arose from investigations into the performance of fuzzy ARTMAP with noisy input data. PROBART is based around the concept of building up probabilistic information regarding inter-layer node associations. The probabilistic information can be used either to generate a predicted output in the form of an expected value (e.g. weighted average) or to generate the most likely value based upon the frequency of inter-layer node associations. By combining output information in this way PROBART offers a solution to the noise/novelty problem encountered by fuzzy ARTMAP when used to approximate mappings, while it still retains many of the attractive properties of fuzzy ARTMAP such as stable category generation and rule representation.

Both fuzzy ARTMAP and PROBART consist of two self-organising fuzzy ART modules (Carpenter *et al,* 1991b) linked by a layer of nodes called the map field. The essential differences between the two architectures result from different inter-module linkage dynamics mediated by the map field.

Section 2 of this paper describes the fuzzy ART architecture, its dynamics, and considers some of the motivations behind its features. The combination of two fuzzy ART modules to form fuzzy ARTMAP is discussed in Section 3 as background and the modifications are introduced. Finally, a comparison of fuzzy ARTMAP and PROBART performance is described in Section 5 followed by a discussion of points arising from the investigation.

2. Fuzzy ART

To allow comparison between PROBART and fuzzy ARTMAP and to make the paper self-contained, the following description of the fuzzy ARTMAP architecture is included here.

Each fuzzy ART module consists of three fields, or layers, of nodes: an input field, a matching field and a choice field. A schematic outline of a fuzzy ART module is shown in Figure 1. The input field, F_0 stores the current input vector and transmits it to the matching field, F_1 which also receives top-down input from the choice field F_2; this latter field representing the active category assignment of the input data.

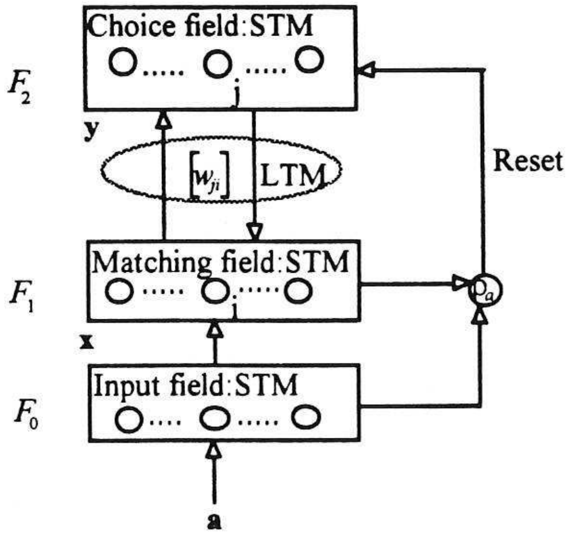


Figure 1. The fuzzy ART module. This illustrates the Relationship between long term memory (LTM) and short term memory (STM).

TheF_0 activity vector is denoted by $\mathbf{I} = (I_1,\dots,I_M)$, $I_i \in [0,1] \in \Re, \forall i = 1,\dots,M$. The F_1 and F_2 activity vectors are denoted by $\mathbf{x} = (x_1,\dots x_M)$ and, $\mathbf{y} = (y,\dots,y_N)$ respectively.

Each F_2 node represents a class or category of inputs grouped together around an exemplar or prototype generated during the self-organising activity of the fuzzy ART module. Furthermore, each F_2 category node, j has its own set of adaptive weights stored in the form of a vector $\mathbf{w}_j = (w_{j1}, w_{j2},\dots,w_{jM})$, $\forall j = 1,\dots,N$.

These weights represent the long term memory (LTM) traces which evolve during network operation. The initial weight vector values are given by:$w_{ji}(0) = 1$, $\forall j = 1,\dots,N$ $\forall i = 1,\dots,M$.

Unlike ARTMAP sub-systems (ART1 modules), the fuzzy ARTMAP components (fuzzy ART modules) differ in that the weight matrix $[w_{ji}]$ includes both top-down and bottom-up weight information.

With no categories being allocated to F_2 nodes at this stage, the nodes are said to be *uncommitted* (Carpenter *et al,* 1992). Once a category node is chosen to represent a category it then becomes *committed.* The parameters which govern Fuzzy ART dynamics are:

i) α, a choice parameter, where $\alpha \cong 0$,

ii) β, a learning rate parameter, where $\beta \in [0,1]$ and

iii) ρ, a vigilance parameter, where $\rho \in [0,1]$.

These parameters will be introduced and described in the relevant contexts below.

2.1. Choice field activity.

The choice field (F_2) nodes operate with winner-takes-all dynamics modelled by the F_2 output function (choice function)

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \quad \forall \mathbf{I} \in [0,1]^M, \tag{1}$$

where $\mathbf{I}$ is the given input vector, $\mathbf{w}_j$ is the $j^{th} F_2$ node weight vector,

$(\mathbf{p} \wedge \mathbf{q})_i \equiv \min_i(p_i, q_i)$, is the fuzzy AND operator, and the L^1 norm $|.|$ is defined by

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

The overall F_2 winner, node J, is selected by $T_J = \max_j\{T_j : j = 1,\ldots,N\}$ to represent a category choice for a given input vector $\mathbf{I}$.

$T_j(\mathbf{I})$ reflects the degree of match between the current input, $\mathbf{I}$ and the LTM of the j^{th} node, $\mathbf{w}_j$.. The ratio, $0 \leq \frac{|\mathbf{p} \wedge \mathbf{q}|}{|\mathbf{q}|} \leq 1$, gives a measure of the fuzzy subsethood of $\mathbf{q}$ with respect to $\mathbf{p}$. The limit, $\frac{|\mathbf{p} \wedge \mathbf{q}|}{|\mathbf{q}|} = 1$ indicates that $\mathbf{q}$ is a fuzzy subset of $\mathbf{p}$.

Specifically, if $\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{w}_j|} = 1$, which occurs when $|\mathbf{I} \wedge \mathbf{w}_j| = |\mathbf{w}_j|$,then $\mathbf{w}_j$ is a fuzzy subset of $\mathbf{I}$. The greatest degree of match between input and weight vectors, for competing nodes, ensures selection as $\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{w}_j|} > \frac{|\mathbf{I} \wedge \mathbf{w}_k|}{|\mathbf{w}_k|}$ gives $\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} > \frac{|\mathbf{I} \wedge \mathbf{w}_k|}{\alpha + |\mathbf{w}_k|}$ and, thus, $T_j(\mathbf{I}) > T_k(\mathbf{I})$ as desired.

The learning rate parameter,α breaks the deadlock between competing nodes when $\mathbf{w}_j$ and $\mathbf{w}_k$ are both fuzzy subsets of $\mathbf{I}$, by selecting the node j such that $|\mathbf{w}_j| > |\mathbf{w}_k|$.

This is because $T(\mathbf{I})$ is monotonically increasing so that, $|\mathbf{I} \wedge \mathbf{w}_j| = |\mathbf{w}_j|$ giving

$T_j(\mathbf{I}) = \frac{|\mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}$. Thus for $|\mathbf{w}_j| > |\mathbf{w}_k|$, $T_j(\mathbf{I}) > T_k(\mathbf{I})$.

In the case that $T_j = T_k$ for some $j,k \leq N$, such that $T_j, T_k > T_l \;\; \forall l \neq j,k$

the node with the lowest index is chosen.

2.2. Matching field activity

The F_1 layer activity is governed both by bottom-up F_0 layer and top-down F_2 layer activity according to

$$\mathbf{x} = \begin{cases} \mathbf{I} & \text{if } F_2 \text{ is inactive} \\ \mathbf{I} \wedge \mathbf{w}_J & \text{if the } J^{th} \; F_2 \text{ node is active.} \end{cases}$$

If node J is active, $\mathbf{w}_J$ represents an expected pattern or template fed down from F_2; this template is combined with the input vector present across F_0 to produce a resultant vector. The ratio of the magnitude of the resultant vector to the magnitude of the input vector gives the degree of match. This ratio, or match function, is denoted by $\frac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|}$ and must fulfil the criterion

$$\frac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|} \geq \rho, \qquad (2).$$

to ensure that the input vector belongs to the chosen category. This state is known as *resonance* and allows learning to occur in the relevant section of the LTM weight matrix. The parameter ρ is the *vigilance* parameter.

The situation where $\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} < \rho$, known as *mismatch,* causes the system to reset and inhibits the winning node ($T_J = 0$) which is, thus, unable to re-enter the competition from which a new winner is selected. The cycle continues with multiple re-presentations of the input vector until the input is either assigned to an existing category or becomes the exemplar for a newly created category.

This approach, with individual nodes representing categories, allows for dynamic adjustment of network size without disrupting previously acquired information as happens with, for example, feedforward networks. Extra nodes are simply assigned as and when required to represent new categories or pattern clusters. Both the fuzzy ARTMAP and the PROBART implementations discussed in this paper use dynamic node allocation. However a fixed number of nodes can be allocated at the outset if desired.

2.3. Learning

Following a successful search, LTM changes are made according to

$$\mathbf{w}_J^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_J^{(old)}) + (1-\beta)\mathbf{w}_J^{(old)} \qquad (3)$$

for the winning F_2 node, J. These changes correspond to the notion of learning.

The learning rate parameter, β, with $0 \leq \beta \leq 1$ ensures that the new weight vector $\mathbf{w}_J$ is a convex combination of the resultant vector across F_1 and the F_2 layer expectation template. For $\beta = 1$, known as *Fast-Commit-Fast-Recode (FCFR)*, F_1 resultant vectors directly replace the present category exemplars.

An option, *Fast-Commit-Slow-Recode(FCSR)*, allows for initial fast learning prior to the convex combination learning rule of equation (3) by setting $\beta = 1$ for uncommitted nodes only. Thus, $\mathbf{w}_J^{(new)} = \mathbf{I}$ initially.

2.4. Complement coding

According to Carpenter *et al,* (1991a, 1991b, 1992) normalisation of the input vectors is required to prevent category proliferation. In Carpenter *et al,* (1991) it is proved geometrically that, without complement coding, the monotonically decreasing weight components would eventually result in many categories clustering near to the origin with others being created to replace them. For example, on the real line, when all categories to the left of an input value are inhibited, the first category to the right will be selected as any categories further to the right will result in a smaller activation value for the function $T(I)$. Furthermore, the condition of equation (2) is always fulfilled as $I < w_J$ gives $\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} = \frac{|\mathbf{I}|}{|\mathbf{I}|} = 1 \geq \rho$

An alternative proof, illustrating category proliferation in the real line will be found in Appendix B.

Normalisation is represented by $|\mathbf{I}| \equiv \gamma, \quad \forall \mathbf{I} \in [0,1]^M$ for some $\gamma > 0$. To achieve this for arbitrary $\mathbf{I} \in [0,1]^M$ take $\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) \in [0,1]^{2M}$ where $\mathbf{a} \in [0,1]^M$ is the original input and $\mathbf{a}^c = \mathbf{1} - \mathbf{a}$ where $\mathbf{1} = (1,1,\ldots,1)$, and, $|\mathbf{1}| = M$.

Thus, the new F_0 layer input vector, $\mathbf{I}$ is complement coded and of dimension 2M with $|\mathbf{I}| = M, \quad \forall \mathbf{I} \in [0,1]^{2M}$.

3. Fuzzy ARTMAP

For heteroassociative tasks, two connected fuzzy ART modules are required with each module receiving either the input (stimulus) or output (response) component of each pattern pair to be associated. Thus, the input and output spaces are organised into distinct categorised sets during processing. The heteroassociative network discussed here is fuzzy ARTMAP which uses a layer of nodes, called the map field, to link the two fuzzy ART modules. This configuration is illustrated in Figure 2. The main function of the map field is to associate compressed representations of the original pattern pair components.

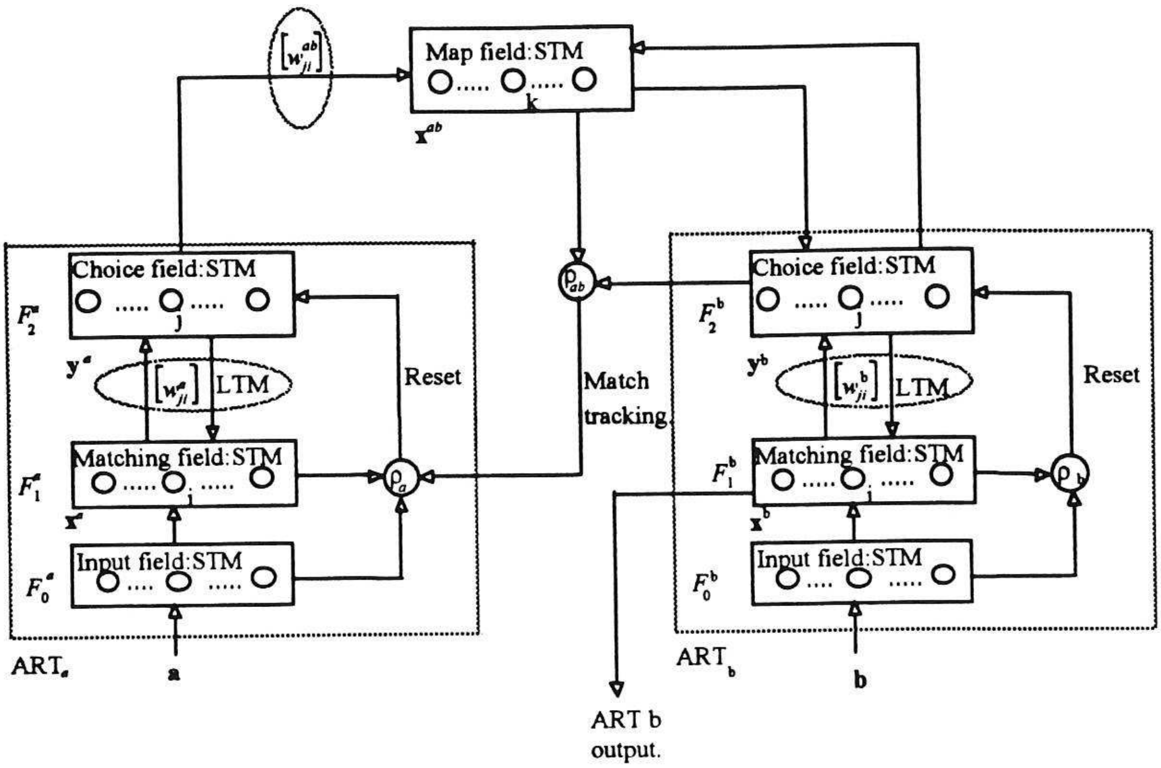


Figure 2. The fuzzy ARTMAP system.

The two fuzzy ART modules, referred to as ARTa and ARTb, accept inputs in complement coded form denoted by $\mathbf{I}_a = (\mathbf{a}, \mathbf{a}^c)$ and $\mathbf{I}_b = (\mathbf{b}, \mathbf{b}^c)$ respectively.

Following the convention of Carpenter *et al* (1992), the ARTa F_1 and F_2 layers are denoted by F_1^a and F_2^a respectively, with output vectors $\mathbf{x}^a = (x_1^a, \ldots, x_{2M_a}^a)$ and $\mathbf{y}^a = (y_1^a, \ldots, y_{N_a}^a)$ respectively. Let, $\mathbf{w}_j^a = (w_{j1}^a, w_{j2}^a, \ldots, w_{j,2M_a}^a)$ denote the j^{th} ARTa weight vector.

Similarly, the F_1^b and F_2^b output vectors are denoted by $\mathbf{x}^b = (x_1^b, \ldots, x_{2M_b}^b)$ and $\mathbf{y}^b = (y_1^b, \ldots, y_{N_b}^b)$ respectively, and $\mathbf{w}_k^b = (w_{k1}^b, w_{k2}^b, \ldots, w_{k,2M_b}^b)$ denotes the k^{th} ARTb weight vector.

The map field is denoted by F^{ab} with output vector $\mathbf{x}^{ab} = (x_1^{ab}, \ldots, x_{N_b}^{ab})$ and weight vector $\mathbf{w}_j^{ab} = (w_{j1}^{ab}, w_{j2}^{ab}, \ldots, w_{j,N_b}^{ab})$ for the j^{th} F_2^a node to F^{ab}.

Activity vectors are reset to zero between data presentations.

3.1. Map field activation

Map field activation is governed by both F_2^a and F_2^b activity in the following way:

$$\mathbf{x}^{ab} = \begin{cases} \mathbf{y}^b \wedge \mathbf{w}_J^{ab} & \text{if the } J^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is active,} \\ \mathbf{w}_J^{ab} & \text{if the } J^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is inactive,} \\ \mathbf{y}^b & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active,} \\ 0 & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \quad (4).$$

The four cases will be considered in order below.

i) F_2^a active and F_2^b active:

This corresponds to a pattern pair $(\mathbf{I}_a, \mathbf{I}_b)$ being present. $\mathbf{I}_a$ elicits an ARTa category selection with, say, the J^{th} F_2^a node winning the competition. This index, J will correspond to a weight vector, $\mathbf{w}_J^{ab}$ in the map field which links the F_2^a node with a predicted F_2^b layer activation. This predicted F_2^b node represents the ARTb category associated with the presently active ARTa category.

Simultaneously, the ARTb input, $\mathbf{I}_b$ has excited a category represented by the F_2^b output $\mathbf{y}^b = (\ldots,0,1,0,\ldots)$ with a 1 in the k^{th} position indicating node k is active.

The fuzzy AND operation, $\mathbf{y}^b \wedge \mathbf{w}_J^{ab}$ ensures that the map field activity is non-zero only if the predicted and actual ARTb categories coincide (the kth category being predicted by ARTa) or if node J is uncommitted; all components of $\mathbf{w}_J^{ab}$ being equal to unity in the latter case.

ii) F_2^a active and F_2^b inactive.

This corresponds to prediction with $\mathbf{w}_J^{ab}$ representing the ARTb category associated with the currently active ARTa category. Heteroassociative mapping is achieved by working backwards within the ARTb module; the fuzzy ARTb weight vector associated with the predicted F_2^b node represents the expectation template fed down from F_2^b to F_1^b; this corresponds to the current exemplar for that ARTb category and, thus, the predicted output.

iii) F_2^a inactive and F_2^b active.

In this case only an ARTb input is present; thus, the map field activation represents the active ARTb category via the one-to-one relationship between the map field and ARTb.

iv) The final case represents the network in a quiescent state with no inputs impinging upon it.

3.2. Match tracking

The concept of vigilance is extended in fuzzy ARTMAP by allowing the ARTa vigilance parameter, ρ_a to vary whilst the ARTb vigilance parameter is fixed for a given training cycle. When an input is first presented, ρ_a is set to its baseline value, $\overline{\rho}_a$. The map field vigilance parameter, ρ_{ab} governs matching between ARTa and ARTb categories through the condition $\frac{|\mathbf{x}|}{|\mathbf{y}|} \geq \rho_{ab}$. If this is not fulfilled, i.e. the ARTa category results in an incorrect prediction, a mismatch occurs which sets off match tracking activity. This consists in increasing ρ_a such that $\rho_a > \frac{|\mathbf{I}_a \wedge \mathbf{w}^a_J|}{|\mathbf{I}_a|}$ to prevent reselection of the J^{th} F^a_1 node. Then the ARTa search cycle is carried out once more to select a new ARTa category which correctly predicts the current ARTb category. One of three conditions must occur to end the match tracking cycle: a matching ARTa category is selected from those already learned by ARTa, a new category is created (during training) or the condition $\rho_a > 1$ occurs which leads to shutdown of F^a_2 until a new ARTa input becomes active.

3.3. Pattern pair association

Pattern pairs are associated via their compressed representations or category nodes. LTM information regarding inter-module F_2 node linkages is stored in the map field weight matrix which assigns a vector to each ARTa node reflecting the associated ARTb node.

Initially, $w^{ab}_{jk}(0) = 1, \quad \forall j = 1,\dots,N_a, \quad \forall k = 1,\dots,N_b$.

When resonance occurs, in which the J^{th} ARTa category becomes active, w^{ab}_J is set equal to $\mathbf{x}^{ab}$.

A clearer idea of heteroassociative learning and prediction under FCSR is gained by considering the operation of fuzzy ARTMAP when presented with a previously unseen pattern pair which does not belong to any of the current categories. The pattern pair $(\mathbf{I}_a, \mathbf{I}_b)$ causes new categories J and K to be created in ARTa and ARTb respectively. The map field activation is given by $\mathbf{x}^{ab} = \mathbf{y}^b \wedge \mathbf{w}^{ab(old)}_J = \mathbf{y}^b$ (K^{th}vector entry = 1 only)

as the J^{th} ARTa node is uncommitted (all entries =1). Map field learning requires $\mathbf{w}^{ab(new)}_J = \mathbf{x}^{ab}$ which gives $\mathbf{w}^{ab}_J = \mathbf{y}^b$

If $\mathbf{I}_a$ is presented alone, the J^{th} ARTa node is selected which predicts the K^{th} ARTb category through the J^{th} map field weight vector.

4. PROBART

PROBART is the result of modifications to the fuzzy ARTMAP system motivated by empirical findings to determine the operational characteristics of fuzzy ARTMAP under certain conditions; a comparative analysis of fuzzy ARTMAP and PROBART operation is presented below. First, the fuzzy ARTMAP modifications incorporated into PROBART are described together with a description of its operation.

As with fuzzy ARTMAP, PROBART uses a pair of fuzzy ART modules linked by a map field; this is where the similarity ends owing to different map field dynamics. The inputs are again accepted in complement coded form. The notation introduced above in the sections describing fuzzy ARTMAP is retained in the description of PROBART Exceptions are described where appropriate.

4.1. Map field activation

In PROBART equation (4) is replaced by

$$\mathbf{x}^{ab} = \begin{cases} \mathbf{y}^b + \mathbf{w}_J^{ab} & \text{if the } J^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is active,} \\ \mathbf{w}_J^{ab} & \text{if the } J^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is inactive,} \\ \mathbf{y}^b & \text{if } F_2^a \text{ is inactive and } F_b^2 \text{ is active,} \\ 0 & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \qquad (5)$$

in which the fuzzy AND operation ($\wedge$) is replaced by vector addition (+). As will become apparent, this allows the nodal association frequency counts maintained in LTM to be incremented.

Before interpreting equation (5) it is important to realise that the map field weight matrix now contains information about the frequency with which pairs of ARTa and ARTb categories are associated e.g. $w_{jk}^{ab} = f, \quad f \in \mathrm{N}$, where N is the set of natural numbers. This indicates that the j^{th} ARTa node has been associated with the k^{th} ARTb node f times during the training phase.

Initial map field weight values are given by

$w_{jk}^{ab}(0) = 0 \quad \forall j = 1,\ldots,N_a, \quad \forall k = 1,\ldots,N_b.$

The four cases of equation (5) are analogous to those given in equation (4) .

i) F_2^a active and F_2^b active:

As with fuzzy ARTMAP, the pattern pair $(\mathbf{I}_a, \mathbf{I}_b)$ results in selection of the J^{th} ARTa category and the K^{th} ARTb category. The vector $\mathbf{y}^b$ is, again, a unit vector with the K^{th} entry equal to one. The vector $\mathbf{x}^{ab}$ now represents the updated frequency distribution of node associations between the J^{th} ARTa node and nodes in the ARTb F_2^b layer; the map field weight matrix entry w_{JK}^{ab} being incremented by one, reflecting the new association.

ii) F_2^a active and F_2^b inactive.

Analogous to fuzzy ARTMAP, this corresponds to prediction but care has to be taken to determine in which sense the prediction is made. The implementation of PROBART discussed in this paper uses a weighted average given by

$$\mu_{Jm} = \frac{1}{\left|\mathbf{w}_J^{ab}\right|}\sum_{n=1}^{N_b} \varepsilon_{nm} w_{Jn}^{ab}, \quad m = 1,\ldots,2M_b \tag{6}$$

where μ_{Jm} is the expected value (mean) of the m^{th} component of the predicted output pattern associated with the J^{th} ARTa node, $\left|\mathbf{w}_J^{ab}\right|$ is the total number of associations of ARTb nodes with the J^{th} ARTa node, ε_{nm} is the m^{th} component of the n^{th} ARTb category exemplar and w_{Jn}^{ab} is the frequency of association between the n^{th} ARTb node and the J^{th} ARTa node. Other possible prediction measures can be used. These include: choosing the exemplar with the highest frequency, giving relative association frequency information, and using alternative higher order moments. The predicted ARTb output vector is denoted by $\mu_J = (\mu_{J1},\ldots,\mu_{J,M_b})$.

Note that only the first M_b components which are not complement coded are meaningful and correspond to the original pattern pair data, with $\hat{\mathbf{b}} = \mu_J$ being an estimate of the true output $\mathbf{b}$ associated with input pattern $\mathbf{a}$.

Equation (6) can also be written as $\mu_{Jm} = \sum_{n=1}^{Nb} \varepsilon_{nm} p_{Jn}$

where p_{Jn} is the empirically estimated probability of association between the J^{th} ARTa node and the n^{th} ARTb node given by $p_{Jn} = \frac{w_{Jn}^{ab}}{\left|\mathbf{w}_J^{ab}\right|}$

Conditions iii) and iv) are identical to those in fuzzy ARTMAP.

4.2. Learning

As with fuzzy ARTMAP $\mathbf{w}_J^{ab(new)} = \mathbf{x}^{ab}$ but note that there is now no match tracking. The ARTa vigilance parameter, ρ_a, is held constant to maintain fixed category sizes. This is to prevent corruption of frequency information as will become apparent from the fuzzy ARTMAP and PROBART comparisons discussed below. Thus, during training, supervised associations are not judged to be correct or incorrect but recorded as they occur. More frequent associations are more heavily weighted in prediction mode. Note that the map field vigilance parameter,ρ_{ab} is not required for PROBART.

Although not investigated below, as with fuzzy ARTMAP, it is possible for PROBART to be operated in an on-line mode and in a non-stationary information environment. In the latter case, node association frequencies would change concomitantly with changes in underlying trends.

5. The mapping task

A continuous non-linear signal was used for comparison of fuzzy ARTMAP and PROBART performance:

$f:[0,1] \subset \Re \rightarrow [0,1] \subset \Re$ with,

$f(x) = (\sin(10x) + \sin(20x) + \sin(30x) + \sin(40x) + \sin(50x) + \sin(60x) + \sin(70x) + 10)/20$, and x in radians. See Figure 3.

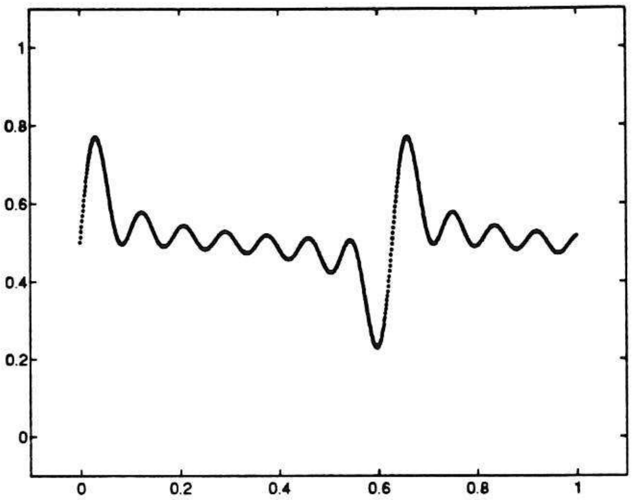


Figure 3. Noise-free test signal.

The range of the test function $f:\Re \rightarrow \Re$ is [0.2295, 0.7705] for the input domain [0,1]. Gaussian noise, derived from a zero mean source with unit variance, is added to the signal with a scale factor of 0.02. Thus, the corrupted output signal for pattern pair p is given by $y_p = y(p) = f(x_p) + 0.02\varepsilon_p$, where $\varepsilon_p \sim N(0,1)$ is the Gaussian noise, for the pth pattern pair and x_p is the x-coordinate of this pair. The x-coordinates were randomly chosen from a uniformly distributed source.

The training and testing files were generated with different sets of x-coordinates unless otherwise stated The testing sets being noise-free coordinates, or pattern pairs, (x_p, y_p) chosen at random from the test curve, $f(x)$.

For all experiments the choice parameter, $\alpha = 0.001$ and the learning mode chosen was FCFR unless otherwise stated.

5.1. Performance measures

Performance is judged by both the root mean square error (RMSE) and maximum absolute error (MAXAE) measures. The RMSE value is computed by

$$RMSE = \sqrt{\frac{1}{N}\sum_{p=1}^{N}\left\|\mathbf{d}_p - \mathbf{y}_p\right\|^2},$$

where $\mathbf{d}_p$ is the desired output for pattern p, $\mathbf{y}_p$ is the actual output and N is the number of patterns used for training or testing.

In the following tables, TR denotes the noisy training set, TE(NF) denotes the noise-free test set using the same x-coordinates as the noisy training set, and TE denotes the noise-free test set selected using a different x-coordinate sample. The purpose of TE is to test the generalisation of the mapping.

Mean results are based upon a sample size of 5 RMSE or MAXAE values from separate runs which are averaged to give an indication of performance. Maximum and minimum values are included to indicate the range of variation between runs.

As a further illustration of network performance, the error profile is plotted below the actual network output signal. RMSE and MAXAE error measures alone are very coarse indicators of network performance, especially when applied over the whole curve. Error profiles provide more detailed information in a visual form

For the simulations described below, a more comprehensive set of results will be found in Appendix A.

Simulation 1

Fuzzy ARTMAP was trained on both noise-free and noisy data. Its parameters were set as follows: $\alpha = 0.001$, $\rho_a = 0.99$, $\rho_b = 0.99$ and $\rho_m = 0.9$. Both the training and test sets consisted of 1,000 data pairs.

For the training signal without noise:

Table 1. Typical results:

Categories.			
ARTa	ARTb	RMSE	MAXAE
312	53	0.0074	0.01

For the typical results, only a single training epoch was required for fuzzy ARTMAP to acquire an internal representation of the test mapping signal with the RMSE ranging between approximately 1% of the input signal at its maximum point to approximately 3% at its minimum point. This is shown in Figure 4.

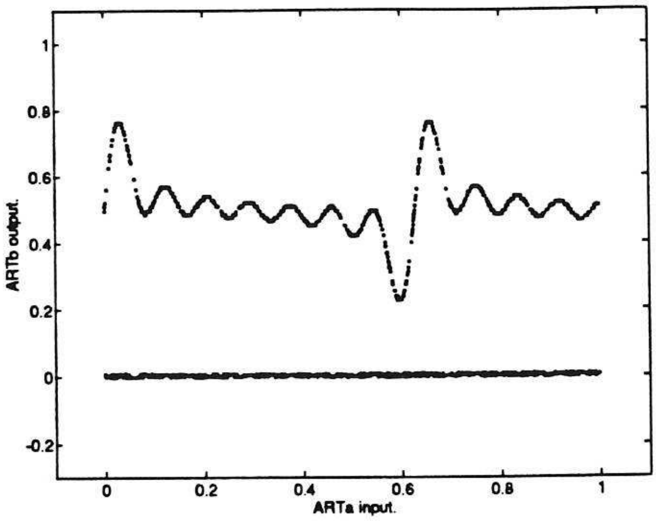


Figure 4. Fuzzy ARTMAP performance with noise-free data.

Note the uniformity of the error profile which attains an absolute maximum range of only 4.4% to 1.3% of the test signal at its maximum and minimum points respectively.

The effect of match-tracking on the fuzzy ARTa module is immediately apparent from the distribution of category numbers between the two modules in Table 1. Taking the ratio of the total input signal range (1.0) to the total output signal range (0.541) predicts a category ratio of approximately 2:1 for the ARTa and ARTb modules respectively. This ratio assumes that both modules have the same vigilance parameters and, hence, the same input resolution or category sizes. At the beginning of each training pattern pair presentation the condition $\rho_a = \rho_b$ is fulfilled. For the typical results of Table 1, match-tracking has increased the ratio to about 6:1 by reducing category sizes through increased vigilance in order to resolve sub-categories. Data compression of approximately 3.3 data points per category node is achieved.

For the training signal with noise:

Table 2. Typical results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
806	61	0.0137	0.0302	0.0302	0.0878	0.0678	0.0679

When fuzzy ARTMAP is trained with the same typical input signal as above but distorted by noise, two training epochs are required to obtain the lowest training RMSE value. Both training epochs consist of presenting the pattern pairs and adjusting the network weights after each individual presentation on the basis of erroneous predictions. A single training epoch requires that the whole training file be processed in this way. Following training, the training file is used purely as a test file (with the learning mode disabled) to assess the current learning progress. The disabling action prevents further learning from taking during testing. The typical results are illustrated in figure 5.

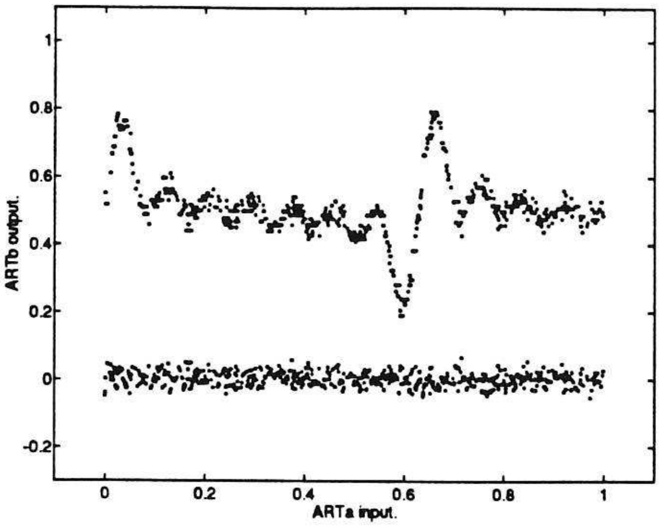


Figure 5. Fuzzy ARTMAP performance with noisy data.

The error profile, coupled with the number of ARTa categories, indicates that each disturbance is being faithfully recorded on an almost individual basis. Its characteristics do not vary across the input domain. Thus, it appears that the source of error has not been effectively filtered or altered.

FCFR results ($\beta = 1$) are quoted as both the RMSE and MAXAE measures did not vary greatly for various values of β in the range 0.1 to 1. Variation of β, using FCSR, did not appear to effect noise suppression through equation (3) with the maximum measured difference between training RMSE values for this data set being approximately 4% of the lowest value. This apparent insensitivity to β was consistently observed and was the result of the high vigilance values confining categories within narrow ranges. This situation is depicted graphically in Figure 6.

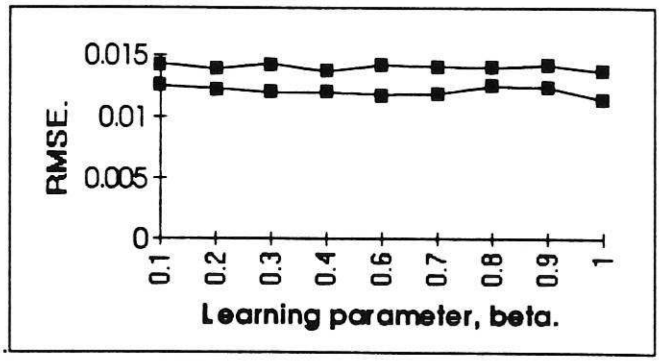
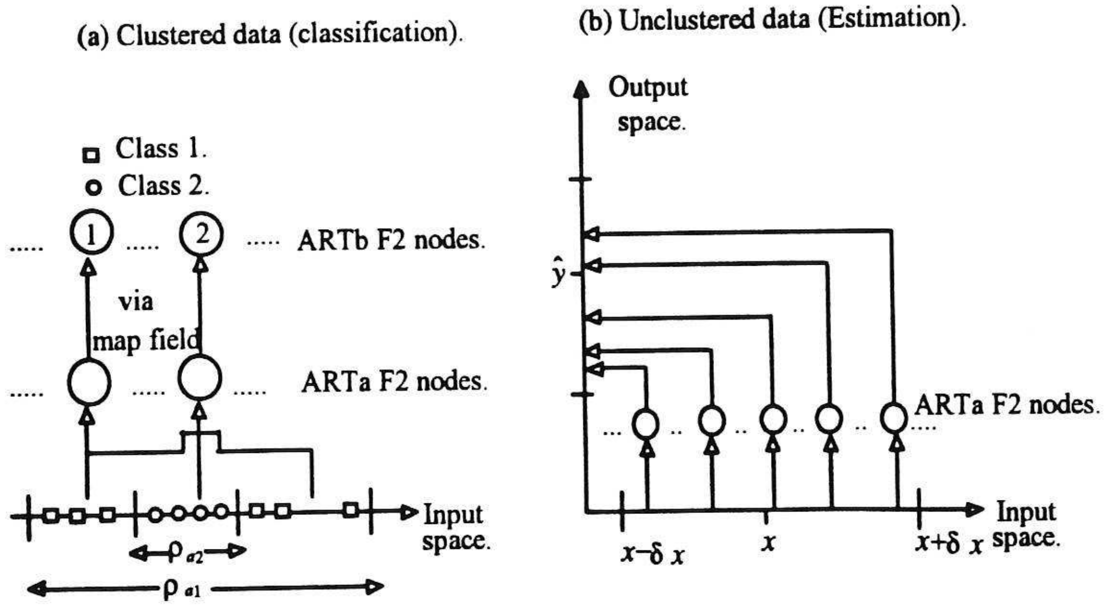


Figure 6. Plot of RMSE against β for two different runs with $\rho_a = \rho_b = 0.99$ illustrating the lack of effectiveness of β in reducing noise.

Note the significant increase, when comparing Tables 1 and 2, in the number of ARTa categories required to represent the noisy mapping while the number of ARTb categories did not increase unduly The latter increase reflects an extended ARTb input range as a consequence of noise. The large number of ARTa categories did not reduce when β was varied using FCSR. The mean ratio of approximately 1.25 data points per ARTa category (Appendix A. Table A1.4) indicates that fuzzy ARTMAP appears to be learning the noisy signal in contrast with the underlying mapping. This observation is further confirmed by the RMSE results for the training data set, with the mean noise-free testing RMSE value (TE(NF)) being greater than twice that of the mean noisy training RMSE (TR) (Appendix. Table A1.4) after training fuzzy ARTMAP on noisy data.

However, this example must not be taken to indicate poor performance by the network in general. The data here is highly disorganised, having no clusters, while fuzzy ARTMAP performs best with clustered data. Match tracking allows sub-clusters to be resolved in classification problems by varying the ARTa vigilance parameter during learning, but this enhanced performance mechanism becomes a disadvantage in highly disorganised data sets such as those used here. To understand the operation of match tracking under these circumstances, refer to Figure 7(a) where, for clustered data, the category delimited by ρ_{a1} maps to an ARTa node and via the map field to, say, ARTb category 1 (class 1). If data is found within the ARTa node category which does not map to category 1, match tracking increases ARTa vigilance to $\rho_{a2} > \rho_{a1}$ which leads to the activation or formation of a sub-category capable of being associated with ARTb category 2. This mechanism is suited to classification problems. Thus, sub-categories are formed which allow learning of infrequent but perhaps significant features which may be ignored or averaged out by other architectures including PROBART.


(a) Clustered data (classification).

(b) Unclustered data (Estimation).

Figure 7. Comparison of classification and estimation modes.

With unclustered data deviations in ARTb values are treated as novel features and new ARTa sub-categories are created individually to encompass many of the data points (see Figure 7(b)). Thus, a small subset of the input space may be mapped to a larger range of output space determined by the noise which is treated as a multitude of predicted output classes. Ideally, the range of output space would be transformed to provide an estimated output which the given input range $x \pm \delta x$ would map to, but this does not happen. In other words, fuzzy ARTMAP does not map an input belonging to the δ – neighbourhood of x to an estimate $\hat{y}_1$. It creates a sub-category for such inputs and individually maps them to the noisy outputs with which they are associated during training.

Simulation 2

PROBART was trained on the same sets of noisy and noise-free data used in simulation 1. The parameters: $\alpha = 0.001, \rho_a = 0.99, \rho_b = 0.99$ are set identically to those in the previous experiment wherever possible. The map field vigilance does not exist in PROBART as match tracking has been removed.

For the training signal without noise:

Table 3. Typical results:

Categories.			
ARTa	ARTb	RMSE	MAXAE
110	53	0.0169	0.0755

Figure 8 illustrates the performance of PROBART with noise-free data after a single epoch (typical results).

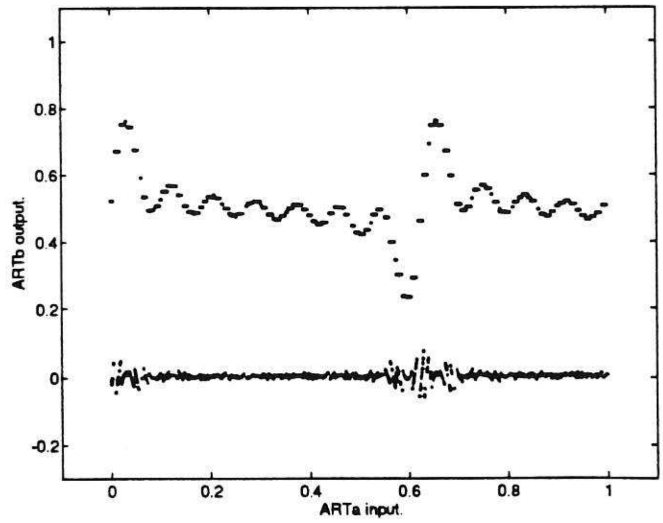


Figure 8. PROBART performance with noise-free data.

Note the different error profile when Figure 8 is compared with Figure 4. The former is not uniform, exhibits structural properties and is considerably larger in magnitude at some points, notably where large increases in signal slope occur. As will become apparent, this is a consequence of the trade-off between plasticity and stability. When match tracking is removed, sensitivity to rapidly fluctuating noise signals is greatly reduced as ARTa sub-categories are not created to represent the noisy associations. However, this fixed quantization of the input domain leads to inaccuracies in signal representation. The relative importance of these inaccuracies, compared to overall noise reduction with noisy signals and increased generalisation, depends upon the application.

For the training signal with noise:

Table 4. Typical results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
112	61	0.0322	0.0189	0.0202	0.1057	0.0769	0.0905

Figure 9 shows the typical results of this simulation after a single epoch.

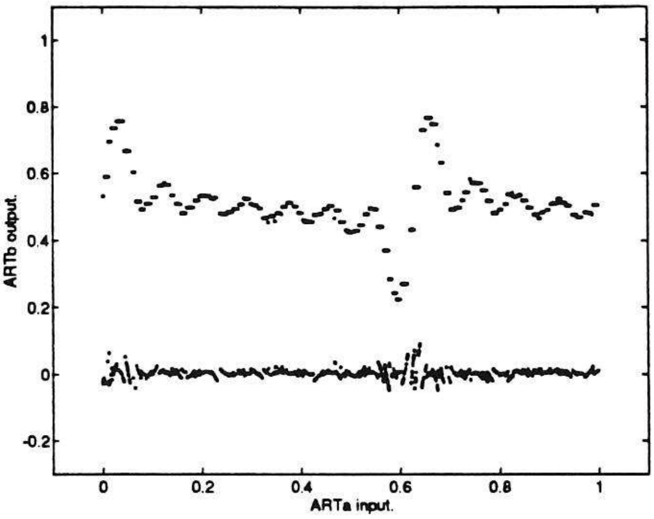


Figure 9. PROBART performance with noisy data.

Fast learn results are, again, quoted with only a 5% maximum variation from the lowest training RMSE value for $0.1 \leq \beta \leq 1$.

The predicted category ratio of 2:1 for the number of ARTa nodes compared to the number of ARTb nodes is reflected in both Table 3 and Table 4. Again, the increase in ARTb nodes is a consequence of output range extension by the additive Gaussian noise.

The mean ratio of 9.0 data points per ARTa category indicates that PROBART uses a coarser partitioning of the input space than that generated by fuzzy ARTMAP to represent the function/mapping domain. This reduction in categories results from the use of a fixed ARTa vigilance which, unlike fuzzy ARTMAP, does not allow subdivision of existing categories. In mapping problems this data compression is desirable to prevent the network from degenerating into a 'look-up' table and, thus, being incapable of generalisation.

Observe in Table A2.4. of Appendix A that the mean noise-free test RMSE value (TE(NF))is lower than the mean noisy training RMSE value (TR) (both sets of data used as test data following training with noisy data). As expected, this indicates that the opposite effect to that observed in fuzzy ARTMAP simulations is taking place. PROBART tends to learn the underlying signal which is, of course, the objective of this work.

The larger mean RMSE of PROBART (Table A2.1. in Appendix A) for the noise-free training/testing data set compared to that exhibited by fuzzy ARTMAP(Table A1.1. in Appendix A) results from the fixed vigilance which limits the input domain partitioning. The reduction in resolution in rapidly changing signal regions (increasing gradient) is apparent from Figures 8 and 9 both in the actual output signals and the error profiles. Thus, prediction errors are increased in those subsets of the input domain where small ARTa inter-category distances give rise to larger ARTb inter-category distances in the function range. These errors, unrelated to noise, account for a sizeable proportion of the RMSE value in PROBART simulations trained with a noisy data set.

Comparison of Tables A2.4 and A1.4 in Appendix A reveals that PROBART reduces the mean RMSE value for the noise-free test set to 67% of the value for fuzzy ARTMAP. This gain in performance is considerably enhanced when comparing the number of ARTa categories generated by both systems. PROBART has achieved generalisation, using approximately one seventh of the number of ARTa category nodes required by fuzzy ARTMAP.

To investigate the gradient/error relationship further, an experiment was performed using a straight line as the training function, where the gradient was varied in the range 1-10 for a fixed vigilance of 0.99 at fixed intercepts The results of a single experiment consisting of 5 runs of the same noise-free training file using different gradients is shown in Figure 10. The test file used was identical to the training file to eliminate the introduction of errors related to the use of different x-coordinate values.

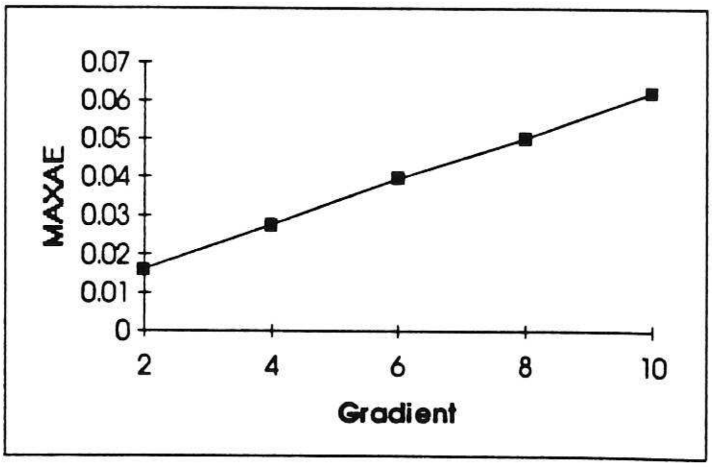


Figure 10. Plot of maximum absolute error vs. gradient.

Note the linear relationship between the maximum absolute error and the gradient confirming that, as expected, rapidly changing signal regions decrease predictive accuracy. This linearity was consistently observed. Thus, signal quantization, resulting from the use of fixed vigilance parameters, introduces inaccuracies which can only be removed by increasing system vigilance to provide finer coverage of the input (stimulus) space and output (response) space. Reduction of the quantization interval size is used to compensate for the removal of match-tracking. The effect of increasing both the ARTa and ARTb vigilance parameters to increase signal resolution was investigated in the following simulations.

Simulation 3

PROBART was trained using the same noise-free data and value of α but with increased vigilance parameters: $\rho_a = 0.999, \rho_b = 0.999$.

Again, the test file was identical to the noise-free training file and consisted of 1,000 coordinate pairs.

Table 5. Typical results:

Categories.			
ARTa	ARTb	RMSE	MAXAE
499	243	0.0016	0.0084

The results are illustrated in Figure 11.

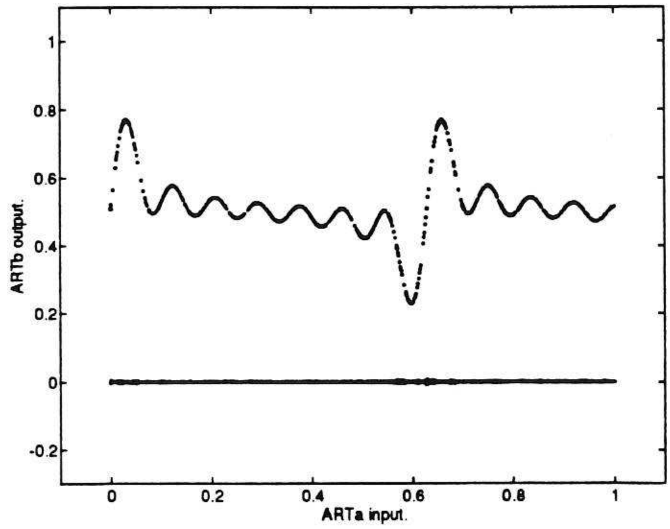


Figure 11. PROBART performance with noise-free data and increased vigilance

Note the improvement in the error profile over that of Figure 8. Disturbances in the profile in areas of rapid signal change have been greatly reduced.

Compared with the mean noise-free results of simulation 2 (Appendix A, Table A2.1), both the ARTa and the ARTb modules have shown an approximately five-fold increase in the mean number of category nodes (Appendix A, Table A3.1). These increases are reflected in the reduction of both mean error measures to about 10% of the previous values. Thus, the signal has been represented more accurately but at the expense of an increase in overall network size. Again, varying β made very little difference, producing less than 10% maximum variation in the range of RMSE values for the typical results quoted.

However, the benefits of simply increasing input/output space resolution are not realised when noisy training data is used as the following simulation illustrates.

Simulation 4

PROBART was trained with the noisy data set used previously in simulations 1 and 2 with parameters set as for simulation 3.

Table 6. Typical results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
504	277	0.0208	0.0196	0.0192	0.0527	0.0544	0.0545

Results of a typical run are illustrated in Figure 12.

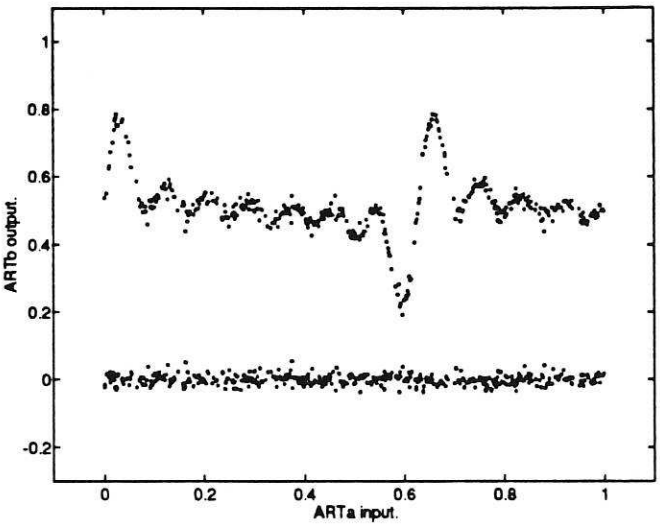


Figure 12. PROBART performance with noisy data and increased vigilance.

The error profile bears some similarity to that of Figure 5 and reflects the increased vigilance leading to reduced category sizes and poorer generalisation.

Comparing the mean results (Appendix A, Table A4.1) with the second mean set of simulation 2 (Appendix A, Table A2.4), it is apparent that a five-fold increase in the number of ARTa nodes has resulted in a 40% decrease in training RMSE (TR) and negligible change in both testing RMSE values. The mean MAXAE has been reduced in all three cases with a 50% reduction in mean training error (TR). Thus, although the testing RMSE values, TE(NF) and TE, are comparable, comparison of Figures 9 and 12 gives a clearer indication of what is happening.

This altered performance is explained by considering the ratio of approximately 2 data points per ARTa node which gives small samples for averaging to give an estimated output value. Thus, estimates are based on smaller sample sizes and are correspondingly less accurate.

Simulation 5

Increasing the number of training data points to 10,000 and using similar parameters gives the following results:

Table 7. Typical results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
1145	618	0.0265	0.0096	0.0114	0.0785	0.0255	0.0472

Note that the mean RMSE value for the test set (TE) (Appendix A, Table 4.4) after training on a noisy data file of 10,000 points has been reduced to about 56% of its previous value for 1,000 data points (Appendix A, Table 4.1). There is also an additional two-fold increase in ARTa category nodes. This latter increase is explained by the increased number of uniformly distributed x-coordinates causing the packing density of ARTa nodes to rise, restricted only by the vigilance parameter.

The following graph, Figure 13, illustrates the variation in test RMSE for ARTa and ARTb Vigilance in the range 0.99-0.999 for the typical data set used throughout. The general trend appears to indicate a reduction in RMSE for increased vigilance as expected.

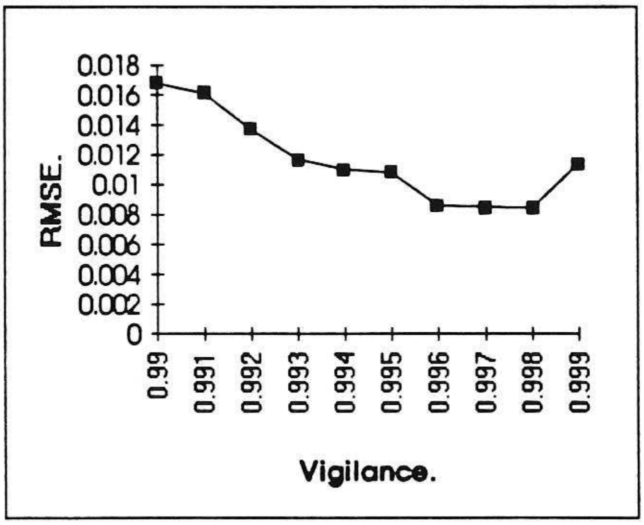


Figure 13. Plot of test (TE) RMSE vs. vigilance for the typical data mentioned above.

The upturn for a vigilance value of 0.999 further confirms the hypothesis that high vigilance values lead to smaller sample sizes and, thus, less accurate estimates of output values. There is a fundamental conflict between providing an adequate partitioning of the ARTa input space and adequate sample sizes for calculating the expected output value

Figure 14 illustrates the effect of increasing the size of the noisy training file for the typical data.

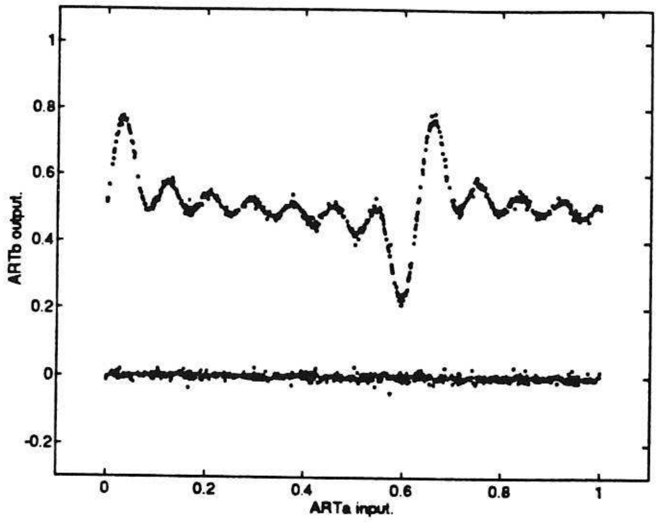


Figure 14. PROBART performance when trained on 10,000 point noisy data file.

For $\rho_a = \rho_b = 0.998$, the following results were obtained:

Table 8. Typical results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
608	341	0.0276	0.0079	0.0084	0.0779	0.0219	0.0269

Which gives a further reduction of the test set RMSE (TE)over and above the typical value obtained in simulation 5 to 44% of that obtained with a 1,000 point training file (Table 6).

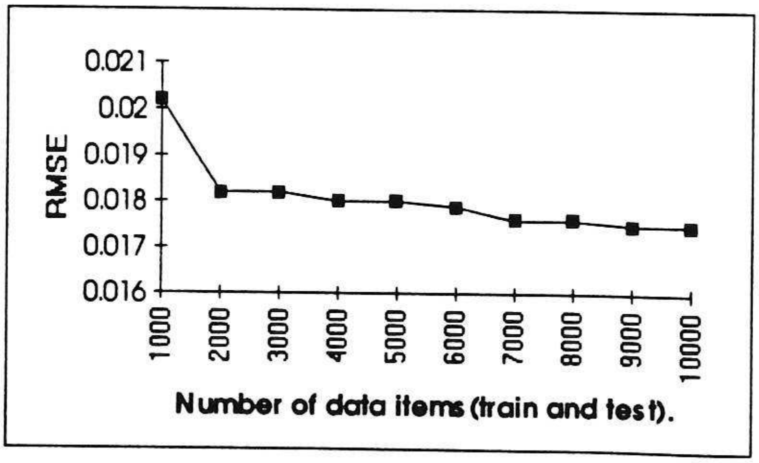


Figure 15 Plot of RMSE value against increasing data set size for a fixed vigilance.

Figure 15 illustrates the stability of RMSE values for increasing training data size. The slight improvement for the larger amounts of data is explained by the increased cover density of the input and output spaces by exemplars and their category zones. Changes in RMSE values are directly affected by changes in the vigilance parameters.

Increasing the amount of data only serves to pack the existing categories and create new categories limited by the vigilance values.

6. General discussion

Both fuzzy ARTMAP and PROBART perform effectively with noise-free data, requiring only one pass through the training file (one epoch) for optimum learning in the RMSE sense (lowest error energy). In contrast with fuzzy ARTMAP, PROBART carries out a single epoch for all training and testing as match-tracking has been removed. This prevents distortion of the computed probabilities (frequency count/ total pattern pairs). For example, for a fixed vigilance, an output, y_1 has the conditional probability given the interval I_{x_1} of $p(y_1|I_{x_1})$ for an interval I_{x_1} based around an exemplar x_1. Were the interval partitioned into two sub-intervals $I_{x_{11}}$ and $I_{x_{12}}$, by increasing vigilance (formation of a sub-category), there is no method of allocating the current frequency count based upon interval I_{x_1} to intervals $I_{x_{11}}$ and $I_{x_{12}}$ individually. Thus, $p(y_1|I_{x_{11}})$ and $p(y_1|I_{x_{12}})$ cannot be derived from $p(y_1|I_{x_1})$. Also, feedback via match tracking alters the frequency of inter-ART node associations by assessing current inputs on the basis of previous data and not by recording raw frequencies. This situation cannot reflect a true empirical frequency distribution upon which the estimated outputs or pattern association probabilities are based.

Fuzzy ARTMAP extremely good at classification problems but match tracking tends to cause the allocation of many nodes for noisy mappings with the noisy disturbances seen as novel features. The dynamics expressed in equation (3) do not act as an effective filter at high vigilance levels (≥ 0.9) using FCSR. This is a consequence of LTM exemplar weights being very near to the noisy input values which fall into their categories. The convex combination of equation (7) gives LTM weight values close to the original exemplar values.

It is difficult to classify neural networks as good or bad on the basis of raw results alone. Overall performance also depends upon the problem to which the network or algorithm is applied. Another factor is the degree of specialisation of the network. Enhanced performance is often obtained at the expense of decreasing generality, i.e. the architecture moves away from being general purpose and becomes oriented towards a particular problem or problem schema. This specialisation frequently requires the incorporation of *a priori* information or structure into the neural network and its dynamics and, thus, restricts its range of applicability.

To a certain extent, PROBART is a trade-off between performance and generality in that better performance could no doubt be obtained using a more specialised network architecture but it does not require *a priori* information about the mapping to be learned.

Given that PROBART deviates significantly from fuzzy ARTMAP, it begs the question why use fuzzy ARTMAP at all? The answer lies in the known attractive properties of ART, in particular, their stability. Other clustering algorithms based, say, on Euclidean distance are known to have stability problems under some circumstances. Moore (1989) cites the *Cluster Euclidean* algorithm which chooses the node coding for the nearest exemplar to the input vector in the Euclidean distance sense. Incorporating equation (7) to give the *Cluster Unidirectional* algorithm (Moore, 1989) removes the endless cycling of weight vectors but suffers from the category proliferation problem countered by the use of complement coding in fuzzy ART.

7. Multidimensional mappings

As stated in the introduction, fuzzy ARTMAP is capable of mapping subsets of $\Re^m$ to $\Re^n$. PROBART is also capable of such mappings. A visual illustration of this capability is included here in the form of a continuous non-linear mapping from $\Re^2$ to $\Re$ which is shown in Figure 16.

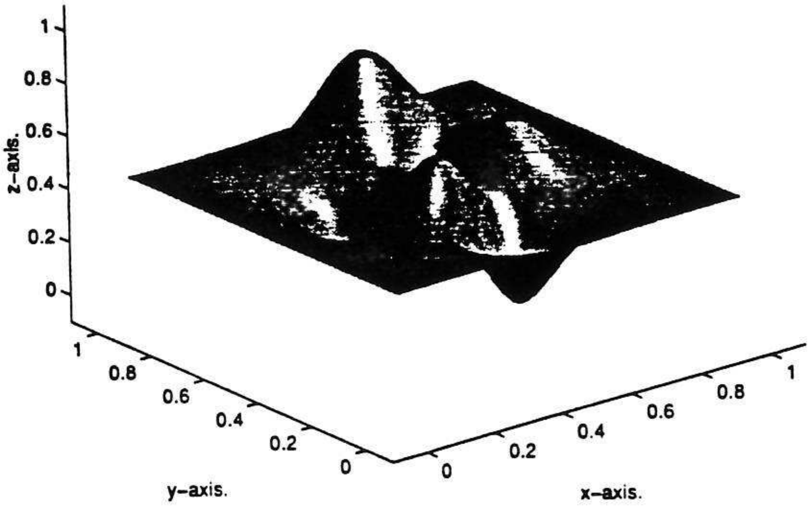


Figure 16. Non-linear test signal $[0,1]^2 \in \Re^2 \rightarrow [0,1] \in \Re$.

Again, Gaussian noise, derived from a zero mean source with unit variance, is added to the signal with a scale factor of 0.02. Conditions and performance measures are similar to those used in the previous single variable mapping but are generalised for the present multivariable mapping.

Simulation 6

Fuzzy ARTMAP was trained on noisy data. Its parameters were set as follows: $\alpha = 0.001, \rho_a = 0.99, \rho_b = 0.99$ and $\rho_m = 0.9$. Both the training and test sets consisted of 1,000 data pairs.

For the training signal with noise:

Table 9. Typical results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
955	63	0.0075	-	0.0235	0.01	-	0.077

The network output and error profile are shown in Figures 17(a) and 17(b) respectively.

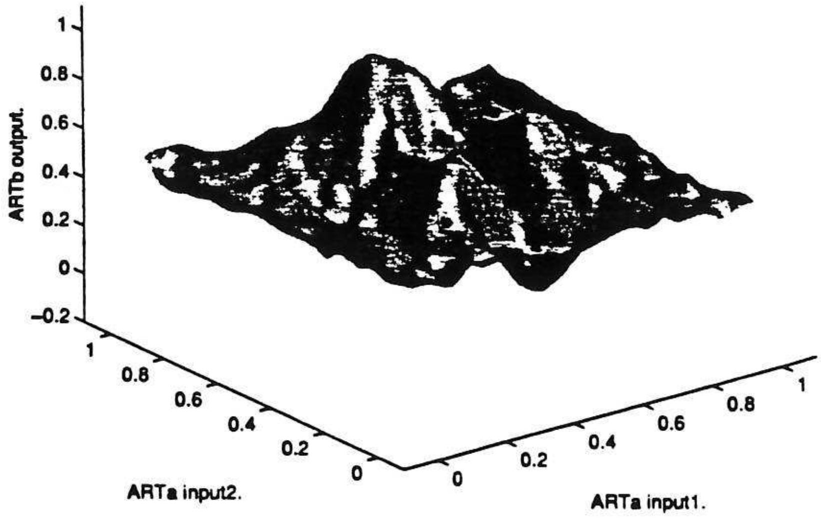


Figure 17(a). Fuzzy ARTMAP output for the noisy non-linear signal with training and testing files consisting of 1,000 data points.

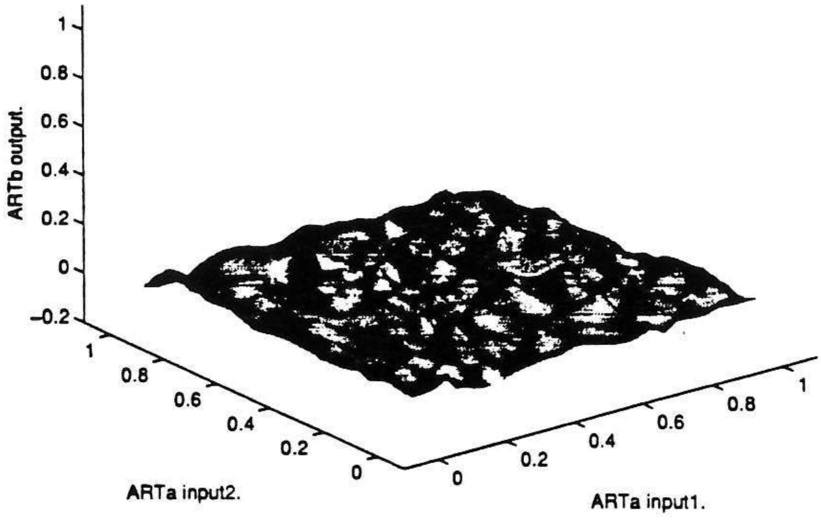


Figure 17(b). Error profile for the above simulation.

Fuzzy ARTMAP requires almost one node per data item. Thus, it acts as a look-up table by storing and retrieving individual pattern pairs. The error profile reproduces the original errors almost faithfully as nearly all individual errors are recorded. It is also apparent from Table 9, as with the single variable examples, fuzzy ARTMAP has learnt the noisy signal.

Changing the learning parameter, β made very little difference. Using values of 0.5 and 0.9 gave testing RMSE values of 0.0236 and 0.0235 respectively. The numbers of ARTa categories were 955 and 950 respectively. The high vigilance parameters for

ARTa and ARTb prevented the occurrence of large changes in RMSE values during training.

Increasing the number of training data points to 5,000 and using similar parameters (FCFR) gives the following results:

Simulation 7

Table 10. Typical Results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
4528	101	0.0076	-	0.0307	0.01	-	0.0743

These results are illustrated in Figures 18(a) and 18(b).

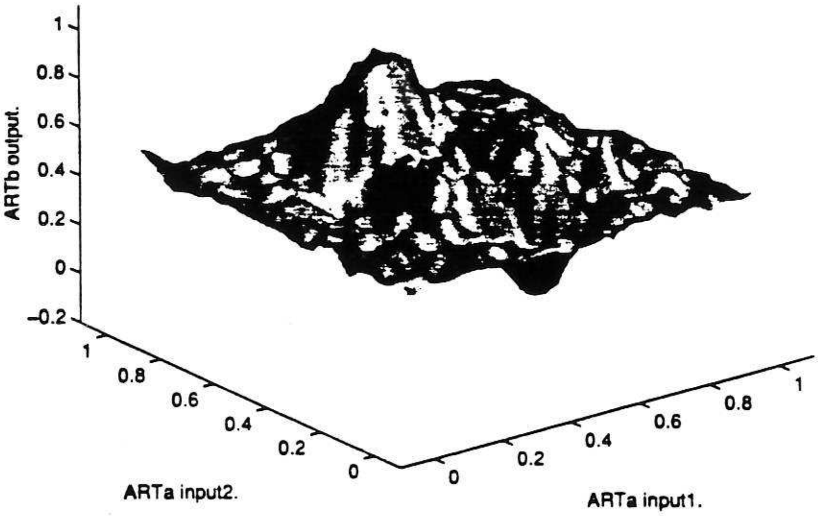


Figure 18(a). Fuzzy ARTMAP output for noisy non-linear signal when trained on 5,000 data points. Test set remains at 1,000 data points.

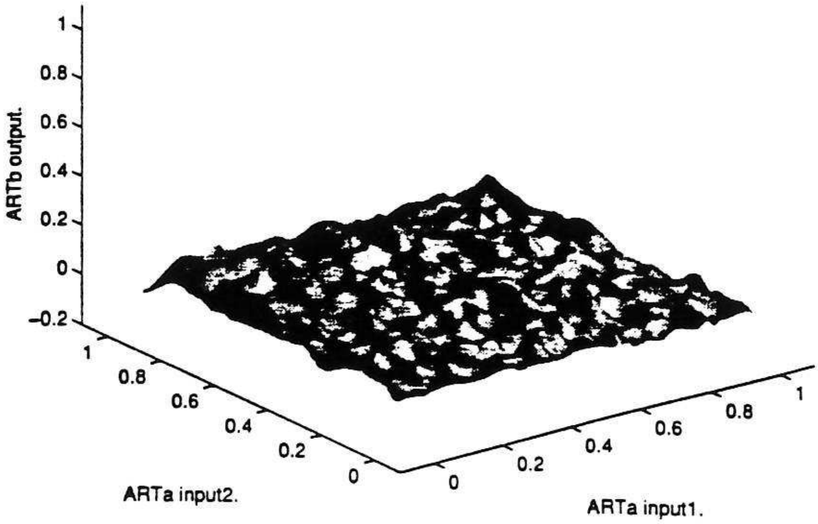


Figure 18(b). Error profile for the 5,000 data point run.

Note the number of ARTa categories which indicate that, as expected, little generalisation has occurred.

Simulation 8

PROBART was trained on the same sets of noisy and noise-free data used in simulation 6. The parameters are set identically to those in that simulation except for the map field vigilance which is not required.

For the training signal with noise:

Table 11. Typical results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
739	63	0.0163	-	0.0196	0.0497	-	0.0775

These results are illustrated in Figures 19(a) and 19(b).

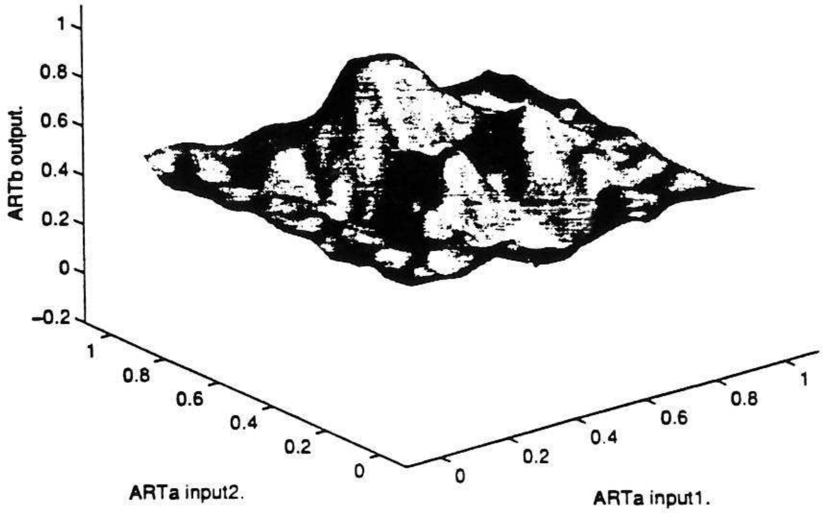


Figure 19(a). PROBART output for noisy non-linear signal. Training and testing files both consisted of 1,000 data points.

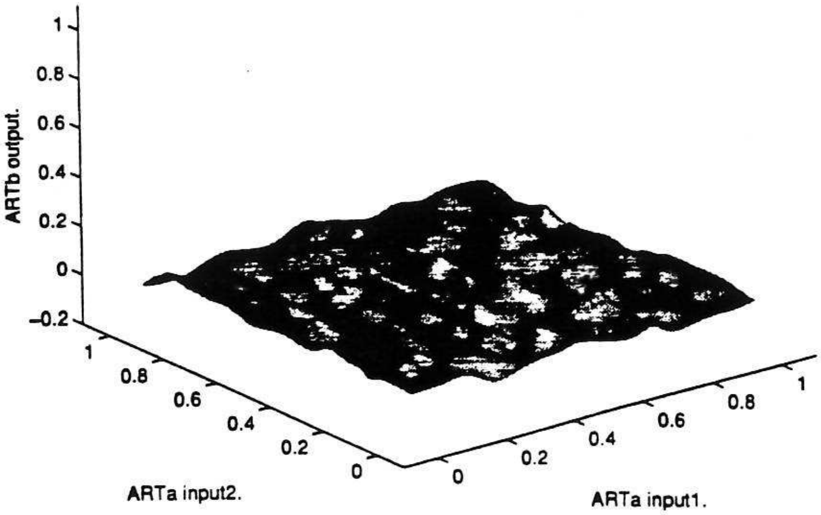


Figure 19(b). Error profile for PROBART run illustrated in Figure 19(a).

Note that, compared to simulation 6, approximately 23% fewer ARTa nodes are required to represent the mapping for a comparable value of testing RMSE.

The following simulation illustrates further reductions in the number ofARTa nodes for PROBART relative to fuzzy ARTMAP.

Simulation 9

Table 12. Typical Results.

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
2283	101	0.0232	-	0.0216	0.065	-	0.067

These results are illustrated in Figures 20(a) and 20(b).

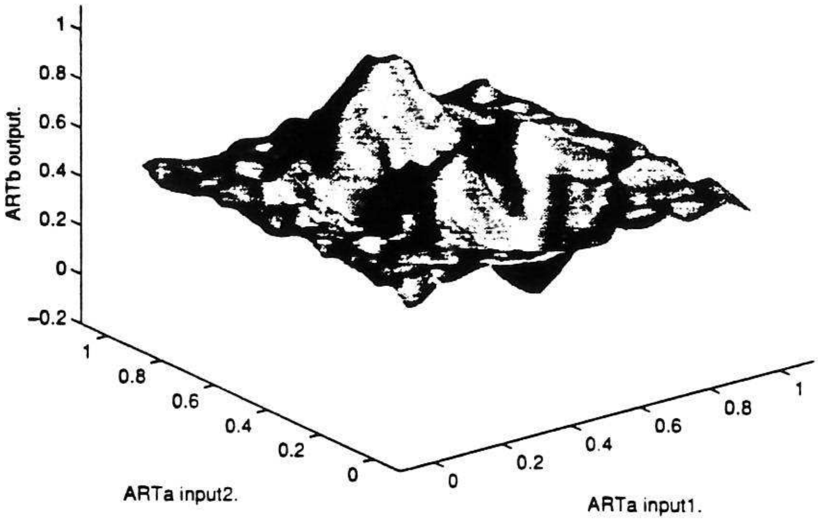


Figure 20(a). PROBART output for noisy non-linear signal when trained on 5,000 data points. Test set remains at 1,000 data points.

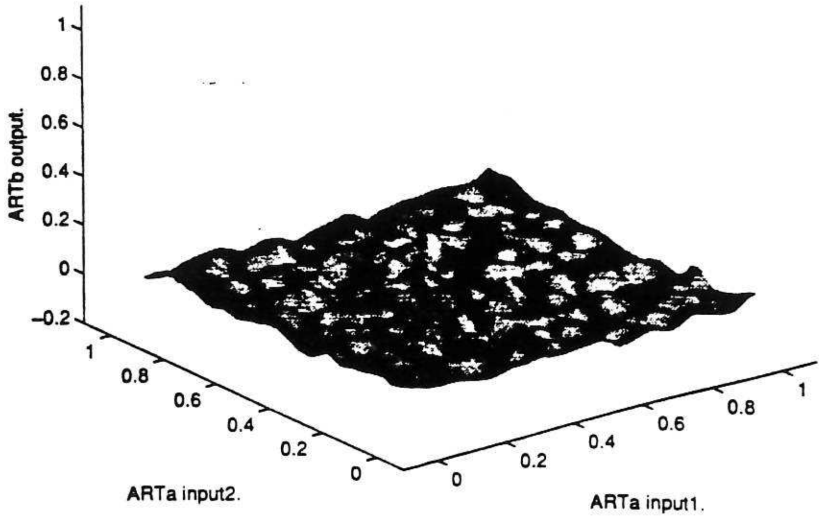


Figure 20(b). Error profile for the 5,000 data point run.

Comparing Table 12 with Table 10 shows a reduction of approximately 50% in the number of ARTa nodes required to represent the mapping. This reduction is not at the expense of testing RMSE (TE) which has been reduced by 30%. This indicates the improved performance offered by PROBART when dealing with larger data sets.

8. Conclusions

It goes without saying that some neural networks do better at certain tasks than others. Often, a specialised network will outperform its more general counterpart but suffers from the disadvantage of requiring *a priori* information pertaining to the learning task. Thus, autonomy is reduced as operator knowledge is built into the network to guide learning. ART-based systems are self-organising and so reduce the need for intervention. They exhibit attractive properties such as the ability to operate in non-stationary environments and to learn continuously new associations following training, without disrupting previous learning. However, the independence of nodes, as in fuzzy ARTMAP, leads to over learning and reduced generalisation as noisy associations are treated as novel associations in noisy mapping problems The mechanism of match-tracking which allows sub-categories to be resolved in classification problems causes categories to proliferate when noisy mapping approximations are carried out. PROBART goes some way to rectifying this by using probability information, combined from various nodes, to estimate output values.

The benefits of using PROBART when dealing with noisy mappings include a reduction in RMSE values, an improved error profile, a sizeable reduction in the number of ARTa category nodes and increased generalisation.

REFERENCES

BABA, N. (1989). A New Approach for Finding the Global Minimum of Error Function of Neural Networks. *Neural Networks,* **2,** pp. 367-373.

CARDALIAGUET, P. and EUVRARD G. (1992). Approximation of a Function and its Derivative with a Neural Network. *Neural Networks,* **5,** pp. 207-220.

CARPENTER, G.A., and GROSSBERG, S, (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., and GROSSBERG, S, (1987b). ART 2: Self-organisation of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics,* **26,** pp. 4919-4930.

CARPENTER, G.A., and GROSSBERG, S, (1989). ART 3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures. *Neural Networks,* **3,** pp. 129-152.

CARPENTER, G.A., GROSSBERG, S., MARKUZON, N., REYNOLDS, J.H., and ROSEN, D.B.,(1992). Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on Neural Networks,* **3,** pp. 698-712.

CARPENTER, G.A., GROSSBERG, S. and REYNOLDS, J.H, (1991a). 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., and ROSEN, D.B.,(1991b). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks,* **4,** pp. 759-771.

CARPENTER, G.A., and Tan, A-H. (1993). Rule Extraction, Fuzzy ARTMAP, and Medical Databases. In *Proceedings, World Congress on Neural Networks, Portland, OR,* Vol I, PP. 501-506. Lawrence Erlbaum Associates. Hillsdale, NJ.

CYBENKO, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematical Control, Signals, and systems,* **2,** pp. 303-314.

FU, L. (1994). Neural Networks in Computer Intelligence. McGraw-Hill inc. Ch 3. pp. 101-105

FUJITA, O. (1992). Optimization of the Hidden Unit Function in Feedforward Neural Networks. *Neural Networks,* **5,** pp. 755-764.

FUNAHASHI, K-I (1989). On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks,* **2,** pp. 183-192.

GIROSI, F. and POGGIO, T. (1990). Networks and the Best Approximation Property. *Biological Cybernetics* **63,** pp. 169-176.

GROSSBERG, S., (1980). How Does a Brain Build a Cognitive Code? *Psychological Review,* **1,** pp. 1-51.

HORNIK, K. (1993). Some New Results on Neural Network Approximation. *Neural Networks,* **6,** pp. 1069-1072.

HORNIK, K., STINCHCOMBE, M. and WHITE, H. (1989). Multilayer Feedforward Networks are Universal Approximators., *Neural Networks,* **2**, pp. 359-366.

ITO, Y. (1992). Approximation of Continuous Functions on $\Re^d$ by Linear Combinations of Shifted Rotations of a Sigmoid Function With and Without Scaling. *Neural Networks,* **5**, pp. 105-115.

KOSKO, B., (1992). Neural Networks and Fuzzy Systems: A dynamical Systems Approach to Machine Intelligence. Prentice-Hall International, Inc., ch7 pp 263 - 298.

LIPPMANN, R. P., (1987). An introduction to Computing with Neural Nets. *IEEE ASSP Magazine,* April 1987, pp 4-22.

MOORE, B., (1989). ART 1 and Pattern Clustering. In TOURETZKY, D. *et al* (Eds.), *Proceedings of the 1988 Connectionist Models Summer School,* pp 174-185. San Mateo, CA, Morgan Kaufmann Publishers.

PARK, J. and SANBERG, I. (1991). Universal Approximation using Radial Basis Function Networks. *Neural Computation,* **3**.

RUMELHART, D. E., HINTON, G.E., and WILLIAMS, R.J. (1986). Learning Internal Representation by Error Propagation. In RUMELHART, D.E. and McLELLAND (EDs.), *Parallel Distributed Processing,* **I**, pp. 318-362 Cambridge, MA,MIT Press.

VAN OOYEN, A. and NIENHUIS, B. (1992). Improving the Convergence of the Back-Propagation Algorithm. *Neural Networks,* **5**, pp. 465-471.

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

Appendix A

Simulation 1

Fuzzy ARTMAP trained with a noise-free training signal:

Table A1.1. Mean results:

ARTa	ARTb	RMSE	MAXAE
298	52	0.0074	0.01

Table A1.2. Worst case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0076	3.31% - 0.99%	0.01	4.36% - 1.3%

Table A1.3. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0073	3.18% - 0.95%	0.01	4.36% - 1.3%

Training signal distorted by noise:

Table A1.4. Mean results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
798	62	0.0131	0.291	0.293	0.0871	0.0717	0.0679

Table A1.5. Worst case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0304	13.25% - 3.95%	0.0698	30.41% - 9.06%

Table A1.6. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0278	12.11% - 3.61%	0.0648	28.24% - 8.41%

Simulation 2

PROBART trained with a noise-free training signal

Table A2.1. Mean results:

ARTa	ARTb	RMSE	MAXAE
113	53	0.0175	0.0783

Table A2.2. Worst case results.

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0185	8.06% - 2.4%	0.085	37.04% - 11.03%

Table A2.3. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0169	7.36% - 2.19%	0.0729	31.76% - 9.46%

For the training signal distorted by noise:

Table A2.4. Mean results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
111	62	0.0316	0.195	0.0206	0.1005	0.0815	0.0839

Table A.2.5. Worst case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0228	9.93% - 2.96%	0.0974	42.44% - 12.64%

Table A2.6. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0196	8.54% - 2.54%	0.0729	31.76% - 9.46%

Simulation 3

PROBART trained with noise-free data and increased vigilance.

Table A3.1. Mean results:

ARTa	ARTb	RMSE	MAXAE
509	243	0.0015	0.0073

Table A3.2 Worst case results.

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0016	0.7% - 0.21%	0.0084	3.66% - 1.09%

Table A3.3. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0015	0.65% - 0.19%	0.0061	2.65% - 0.79%

Simulation 4

PROBART trained with noisy data and increased vigilance.

Table A4.1. Mean results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
513	279	0.0193	0.0199	0.0197	0.0541	0.057	0.0566

Table A4.2. Worst case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0206	8.98% -2.67%	0.0648	28.24% - 8.41%

Table A4.3. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0189	8.23% - 2.45%	0.0498	21.7% - 6.46%

Table A4.4. Mean results:

Categories		RMSE			MAXAE		
ARTa	ARTb	TR	TE(NF)	TE	TR	TE(NF)	TE
1131	620	0.0265	0.0089	0.011	0.0814	0.0225	0.0426

Table A4.5. Worst case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0117	5.1% -1.52%	0.0472	20.57% - 6.13%

Table A4.6. Best case results:

RMSE (TE)	Error range.	MAXAE (TE)	Error range.
0.0103	4.49% - 1.52%	0.0388	16.91% - 5.04%

Appendix B

This appendix illustrates the proliferation of categories by fuzzy ARTMAP on the real line when complement coding is not applied. This derivation differs from that given in Carpenter *et al* (1991b) by applying real analysis to adjacent categories to establish choice regions and category movement rather than the geometric interpretation. Carpenter *et al (1992)* gives a geometric interpretation of the effect of complement coding in reducing the proliferation of categories.

Let w_{s-1} and w_s denote the exemplars for nodes s-1 and s respectively where $w_{s-1}, w_s \in [0,1] \subset \Re$. Without loss of generality, assume

$$0 \leq w_{s-1} < w_s \leq 1 \qquad \text{(A1)}$$

and that for all inputs, I considered here

$$w_{s-1} \leq I \leq w_s \qquad \text{(A2)}$$

for some $s-1, s \in N$. See Figure A1.

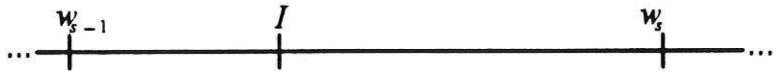


Figure A1. Two adjacent categories in the real line.

Any input, I, can be parameterised in the range

$$I(\lambda) = w_{s-1} + \lambda(w_s - w_{s-1}) \qquad \text{(A3)}$$

where $0 \leq \lambda \leq 1$. Henceforth, $I(\lambda)$ will be denoted by I.

In this case, the choice function of equation (1) gives

$$T_{s-1}(I) = \frac{w_{s-1}}{\alpha + w_{s-1}} \qquad \text{(A4)}$$

and,

$$T_s(I) = \frac{w_{s-1} + \lambda(w_s - w_{s-1})}{\alpha + w_s} \qquad \text{(A5).}$$

Consider the effect of the parameter λ. Three cases naturally arise, viz:-

i)$\lambda = 0$,

ii)$\lambda = 1$,

iii)$0 < \lambda < 1$.

For $\lambda = 0$, from equation (A3), $I = w_{s-1}$, and from equation (A5) $T_s(I) = \dfrac{w_{s-1}}{\alpha + w_s}$.

Also, $T_{s-1}(I) = \dfrac{w_{s-1}}{\alpha + w_{s-1}}$ by equation (A4).

Now, from equation (A1), $w_s > w_{s-1}$ which implies that $T_{s-1}(I) > T_s(I)$, and node s-1 wins as expected.

For $\lambda = 1$, $I = w_s$, $T_{s-1}(I) = \dfrac{w_{s-1}}{\alpha + w_{s-1}}$, and $T_s(I) = \dfrac{w_s}{\alpha + w_s}$.

So, by the monotonic property of $T(I)$, $w_s(I) > w_{s-1}(I)$ gives $T_s(I) > T_{s-1}(I)$

and node s wins as expected.

For $0 < \lambda < 1$ a question naturally arises as to where the decision boundary for adjacent exemplars is.

Equating $T_{s-1}(I)$ and $T_s(I)$ gives $\dfrac{w_{s-1}}{\alpha + w_{s-1}} = \dfrac{w_{s-1} + \lambda(w_s - w_{s-1})}{\alpha + w_s}$ and solving for λ gives

$$\lambda_b = \frac{w_{s-1}}{\alpha + w_{s-1}} \qquad \text{(A6)}$$

where λ_b is the boundary value of λ.

Thus, λ_b is slightly less than one and depends upon α. This means that all inputs in the range given by equation (A2) map to node s-1 unless they are within a small distance of node s. This is proved in the following theorem:

Theorem:

$\forall I$ such that $w_{s-1} \le I < w_{s-1} + \lambda_b(w_s - w_{s-1})$, $w_{s-1} > 0$,

where λ_b is given by equation (A6), I

maps to the s-1 th category.

Proof:

Let $\lambda = \gamma\lambda_b$, $\quad 0 < \gamma < 1$,

i.e. $0 < \lambda < \lambda_b$, as required, so that,

$$T_{s-1}(I) = \frac{w_{s-1}}{\alpha + w_{s-1}},$$

and,

$$T_s(I) = \frac{w_{s-1} + \gamma\lambda_b(w_s - w_{s-1})}{\alpha + w_s}.$$

Now,

$w_s > w_{s-1}$.

Multiplication of both sides by $(1-\gamma)$ and further application of the algebra of inequalities leads to,

$$w_{s-1}(\alpha + w_s) > w_{s-1}(\alpha + \gamma w_s) + (1-\gamma)w_{s-1}^2$$

and,

$$\frac{w_{s-1}}{\alpha + w_{s-1}} > \frac{w_{s-1}(\alpha + w_{s-1}) + \gamma w_{s-1}(w_s - w_{s-1})}{(\alpha + w_{s-1})(\alpha + w_s)}$$

$$= \frac{w_{s-1} + \gamma\lambda_b(w_s - w_{s-1})}{(\alpha + \omega_s)}$$

giving, $T_{s-1}(I) > T_s(I)$

for $0 < \gamma < 1$.

The condition $T_{s-1}(I) > T_s(I)$

requires

$$\frac{w_{s-1}}{\alpha + w_{s-1}} > \frac{w_{s-1} + \lambda(w_s - w_{s-1})}{\alpha + w_s}$$

giving

$$w_{s-1} > \lambda(\alpha + w_{s-1})$$

which leads to

$$\lambda < \frac{w_{s-1}}{\alpha + w_{s-1}}.$$

Also, $\lambda > 0$ and $\alpha > 0$ finally giving

$$0 < \lambda < \frac{w_{s-1}}{\alpha + w_{s-1}} < 1.$$

Therefore,

$T_{s-1}(I(\lambda)) > T_s(I(\lambda))$, for λ in the above range. ♦

Thus, all inputs between exemplars w_{s-1} and w_s map to category s-1 except for those in a small exclusion zone $(w_{s-1} + \lambda_b(w_s - w_{s-1}), w_s)$ determined by α.

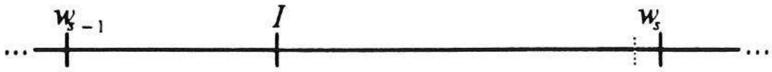


Figure A2. Two adjacent categories in the real line illustrating exclusion zone near to category s.

Note that the above only determines the winning node through $T(I)$ and not category membership which depends upon the match criterion .

Match Criterion

Equation (2) states the match criterion

$$\frac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|} \geq \rho,$$

which gives $w_{s-1} \geq \rho I$ for node s-1.

Thus, $I \leq \frac{w_{s-1}}{\rho}$ is required for a match to occur.

Category Proliferation

Consider what happens when

$$\ldots < w_{s-2} < w_{s-1} < I < w_s < w_{s+1} < \ldots$$

By previous results, $T_{s-1}(I) > T_s(I)$, but, $I > \frac{w_{s-1}}{\rho}$ ensures that node s-1 is inhibited.

Again, $T_{s-2}(I) > T_s(I)$, by previous results, but $I > \frac{w_{s-2}}{\rho}$, causes inhibition of node s-2.

Thus, all nodes, $l \leq s-1$ are inhibited.

Now,

$$T_k(I) = \frac{I}{\alpha + w_k}, \quad \forall k \geq s$$

giving

$$T_s(I) > T_{s+1}(I) > T_{s+2}(I) > \ldots \text{ as } w_s < w_{s+1} < w_{s+2} < \ldots.$$

So, by the above, all nodes, l with exemplars $w_l < w_s$, $l < s$, are inhibited so node s is selected giving $T_s(I) = \frac{I}{\alpha + w_s} > \frac{I}{\alpha + 1}$ for an uncommitted node.

This means that the next available node is selected which has its exemplar w'_s replaced by I as the match criterion gives $\frac{|\mathbf{I} \wedge \mathbf{w}_s|}{|\mathbf{I}|} = \frac{I}{I} = 1 > \rho$ for $\rho < 1$, regardless of the distance between I and w'_s.

Thus, as I<w_s.,, exemplars drift towards the origin as their magnitudes are reduced. This causes the creation of more categories in areas of input space made devoid of exemplars by this drifting effect.

Although stable by the monotonic decreasing of weights, the network suffers from proliferation of category nodes unless complement coding is used.