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

Department of Computer Science

PO Box 600
Wellington
New Zealand

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

On the Match Tracking Anomaly of the ARTMAP Neural Network

Guszti Bartfai

Technical Report CS-TR-95/1
January 1995

Abstract

This article analyses the *match tracking anomaly* (*MTA*) of the ARTMAP neural network. The anomaly arises when an input pattern exactly matches its category prototype that the network has previously learned, and the network generates a prediction (through a previously learned associative link) that contradicts the output category that was selected upon presentation of the corresponding target output. Carpenter at al. claimed that such anomalous situation will never arise if the (binary) input vectors have the same number of 1's (Carpenter et al., 1991a).

This paper shows that such situations *can* in fact occur. The *timing* according to which inputs are presented to the network in each learning trial is crucial: if the target output is presented to the network *before* the corresponding input pattern, certain pattern sequences will lead the network to the *MTA*. Two kinds of *MTA* are distinguished: one that is independent of the choice parameter (β) of the ART_b module, and another that is not. Results of experiments that were carried out on a machine learning database demonstrate the existence of the match tracking anomaly as well as support the analytical results presented here.

Keywords: Match tracking, ARTMAP, Adaptive resonance theory, Supervised learning, Self-organization, Hierarchical clustering, Machine learning, Zoo database.

Publishing Information

This paper has been accepted for publication in *Neural Networks*.

Author Information

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

1 Introduction

The ARTMAP neural network architecture (Carpenter et al., 1991a) is a supervised learning system capable of self-organising stable recognition categories in response to arbitrary sequences of input patterns. It is built up of a pair of Adaptive Resonance Theory (Carpenter and Grossberg, 1987a) modules (ART_a and ART_b) that are connected through an inter-ART associative memory module. The network has an internal mechanism that conjointly maximises predictive generalisation and minimises predictive error through a self-regulating process called *match tracking*. Whenever the network makes a wrong prediction through a previously learned associative link, the *vigilance* parameter ρ_a of the ART_a module will be raised by the minimal amount needed to correct the predictive error at the ART_b module. The ART_a module will then start searching for another category for the current input until it finds a correct prediction, or creates a new ART_a category and its associative link to the corresponding ART_b category. The ARTMAP architecture has been applied to several machine learning benchmark problems (Carpenter et al., 1991a; Carpenter et al., 1992), and compared favourably with traditional AI learning methods.

There are, however, anomalous situations that can occur during training, in which an ART_a input matches its category prototype perfectly and the network makes a wrong prediction (Carpenter et al., 1991a). Raising ART_a vigilance above that (perfect) matching level will prevent the network from finding any other ART_a category, and the wrong prediction will not be corrected. Carpenter and al. claimed that this situation (we shall call it *Match Tracking Anomaly*, or *MTA* henceforth) will never arise if the (binary) input patterns have the same number of 1's (Carpenter et al., 1991a, pages 584-585).

In this article, we show that the *MTA can* in fact arise even if the inputs have the same number of 1's and the training set is non-contradictory. More specifically, we discuss the importance of the timing according to which the input and target patterns are presented to the network in each learning trial. We first show (in section 4.1) that the *MTA* will *never* arise if each input (with the same number of 1's) is presented to the network such that it can make a prediction (through an existing associative link) and "prime" itself before the corresponding (non-contradictory) target is presented. This timing proves to be crucial in that if the ART_b (target) input is presented *before* the ART_a input, the *MTA can* arise (see section 4.2). We show this by specifying a set of initial conditions for the network and a sequence of input/target pairs that will lead the network to the *MTA*. Furthermore, we show that there are two distinct classes of conditions that cause the *MTA*: one that depends solely on the sequence in which inputs are presented from a training set (see the proof in section 4.2.1), and another, in which *MTA* can only occur if the *choice parameter* (β) of the ART_b module is above a certain threshold, which depends on the size of the target inputs (see the proof in section 4.2.2). The combined effect of these two conditions can be observed clearly from the results of the experiments we carried out on a benchmark machine learning database (in section 5).

We have also extended the classes of problems to which the ARTMAP can be applied by allowing the ART_b module, too, to develop "true" clusterings. Carpenter et al. introduced the ARTMAP network to be used for pattern classification where the ART_b (target) inputs are expected *category codes* to which the corresponding inputs should belong. This implies that there is no overlap between any two target patterns. Therefore the ART_b module will develop the same categories on a given sequence of training patterns at *any* non-zero vigilance level. This makes timing of input presentations irrelevant (see section 4.3). This issue was, therefore, of no special concern in (Carpenter et al., 1991a), and timing was only discussed in

Appendix A2 where all possible combinations of ART_a and ART_b input timings were listed with and without predictions. If, however, the ARTMAP network is applied to a more general "mapping" problem where target outputs may overlap, the level of ART_b vigilance will affect the development of categories. One such problem is to learn a *two-level hierarchy* of input classes from a training set where each input pattern is identical to its target pair (i.e. auto-associative map). We discuss this in more detail in section 5.

Section 2 introduces the ARTMAP network at a level that is necessary for understanding the main results of this paper. Section 3 defines the Match Tracking Anomaly (*MTA*) — which is the centre of this article. Section 4 discusses the circumstances under which the *MTA* condition cannot and can occur. Relevant theorems and proofs are given in sections 4.1 and 4.2, respectively. A theorem regarding the important case of ART_b vigilance equal to 1 is shown in section 4.3. The experiments are discussed and the results are presented in section 5, followed by a discussion in section 6. Finally, conclusions are drawn in section 7.

2 The ARTMAP network

Adaptive Resonance Theory (ART) architectures are neural networks that develop stable recognition codes in real time by self-organisation, in response to arbitrary sequences of input patterns (Carpenter and Grossberg, 1987a). They were designed to solve the stability-plasticity dilemma that every intelligent machine learning system has to face: how to keep learning from new events without forgetting previously learned information. ART networks were originally proposed to accept binary input patterns, and were later extended for both continuous and binary inputs (Carpenter and Grossberg, 1987b; Carpenter et al., 1991b)[1].

An ART module has three layers: the *input* layer ($F0$), the *comparison* layer ($F1$), and the *recognition* layer ($F2$), with m, m and n neurons, respectively (see module ART_a or ART_b in figure 1). The neurons, or nodes, in the F2 layer represent input categories. The F1 and F2 layers interact with each other through weighted bottom-up and top-down connections, which are modified when the network learns. There are additional gain control signals in the network (not shown in figure 1) that regulate its operation, but those will not be detailed here.

At each presentation of a non-zero binary input pattern $\mathbf{x}$ ($x_i \in \{0,1\}, i = 1,2,...,m$), the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node j in the F2 layer, the bottom-up activation $T_j = \sum_{i=1}^{m} x_i Z_{ij}$ is calculated, where Z_{ij} is the strength of the bottom-up connection between F1 node i and F2 node j. Since both the input and the bottom-up weight vectors are binary (with Z_{ij} being the normalised version of z_{ij}), T_j can also be expressed as

$$T_j = |\mathbf{x} \cap \mathbf{Z}_j| = \frac{|\mathbf{x} \cap \mathbf{z}_j|}{\beta + |\mathbf{z}_j|} \tag{1}$$

where $|\cdot|$ is the norm operator ($|\mathbf{x}| \equiv \sum_{i=1}^{m} x_i$), $\mathbf{z}_j$ is the (binary) top-down template (or prototype) of category j, and $\beta > 0$ is the *choice parameter* (Carpenter et al., 1991a). Then the F2 node J that has the highest bottom-up activation is selected, i.e. $T_J = \max\{T_j | j = 1,2,...n\}$. The prototype vector of the winning node J ($\mathbf{z}_J$; $z_{Ji} \in \{0,1\}, i = 1,2,...,m$) is then sent down to the F1 layer through the top-down connections, where it is compared to the current input pattern: the strength of the match is given by

[1]From now on, we shall use ART to refer to the binary ART (called ART1) unless otherwise stated. Also, for easy reference, this paper attempts to use the same notation as (Carpenter et al., 1991a), since their claim is the focus of this article.

$$\frac{|\mathbf{x} \cap \mathbf{z}_J|}{|\mathbf{x}|},$$

which is compared with a system parameter ρ called *vigilance* ($0 < \rho \leq 1$). If the input matches sufficiently, i.e. the match strength $\geq \rho$, then it is assigned to F2 node J and both the bottom-up and top-down connections are adjusted for this node:

$$\mathbf{Z}_J(new) = \frac{\mathbf{x} \cap \mathbf{z}_J(old)}{\beta + |\mathbf{x} \cap \mathbf{z}_J(old)|}$$

and

$$\mathbf{z}_J(new) = \mathbf{x} \cap \mathbf{z}_J(old) \quad ^2. \tag{2}$$

If the stored prototype $\mathbf{z}_J$ does not match the input sufficiently (match strength $< \rho$), the winning F2 node J is reset for the period of presentation of the current input. Then another F2 node (or category) will be selected, whose prototype will be matched against the input. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input closely enough, or allocates a new F2 node. Then learning takes place as described above. After an initial period of self-stabilisation, the network will directly (i.e. without search) access the prototype of one of the categories it has found in a given training set. The higher the vigilance level, the larger number of smaller, or more specific, categories will be created. (If $\rho = 1$, the network will learn every unique input perfectly with a different category.)

An important feature of the category selection method of the ART network is that if categories $J1$ and $J2$ are proper subsets of an input pattern $\mathbf{x}$, i.e. $|\mathbf{x} \cap \mathbf{z}_{J1}| = |\mathbf{z}_{J1}|$ and $|\mathbf{x} \cap \mathbf{z}_{J2}| = |\mathbf{z}_{J2}|$, the category whose prototype is the *larger* subset will be selected first (for the vigilance test). This can be seen from (1), and that the $f(y) = \frac{y}{\beta+y}$ function monotonically increases for $\beta > 0$. This feature will play an important role in proving one of the *MTA* conditions (see section 4.2.1).

The architecture of the ARTMAP network can be seen in figure 1. It consists of two ART modules that are linked together through an "inter-ART" associative memory, called *map field* (F^{ab}). Module ART$_a$ (with a *baseline* vigilance $\overline{\rho_a}$) learns to categorise input patterns presented at layer F_0^a, while module ART$_b$ (with vigilance ρ_b) develops categories of target patterns presented at layer F_0^b. Modules F_2^a and F^{ab} are fully connected via associative links whose strengths are adjusted through learning. There are one-to-one, two-way, and non-modifiable connections between nodes in the F^{ab} and F_2^b layers, i.e. each F_2^b node is connected to its corresponding F^{ab} node, and vice versa. A new association between an ART$_a$ category J and an ART$_b$ category K is learned by setting the corresponding $F_2^a \rightarrow F^{ab}$ link to one and all other links from the same ART$_a$ node to zero. When an input pattern is presented to the network, the F^{ab} layer will receive inputs from both the ART$_a$ module (through the previously learned $J \rightarrow K$ associative link) and the ART$_b$ module (from the active F_2^b category node). If the two F^{ab} inputs match, i.e. the network's prediction is confirmed by the selected target category, the network will learn by modifying the prototypes of the chosen ART$_a$ and ART$_b$ categories according to the ART learning equations shown above. If there is a mismatch at the F^{ab} layer, a map field reset signal will be generated, and a process called *match tracking* will start, whereby the baseline vigilance level of the

[2] This is the so called fast learning mode of the network, which is typically used in binary ART.

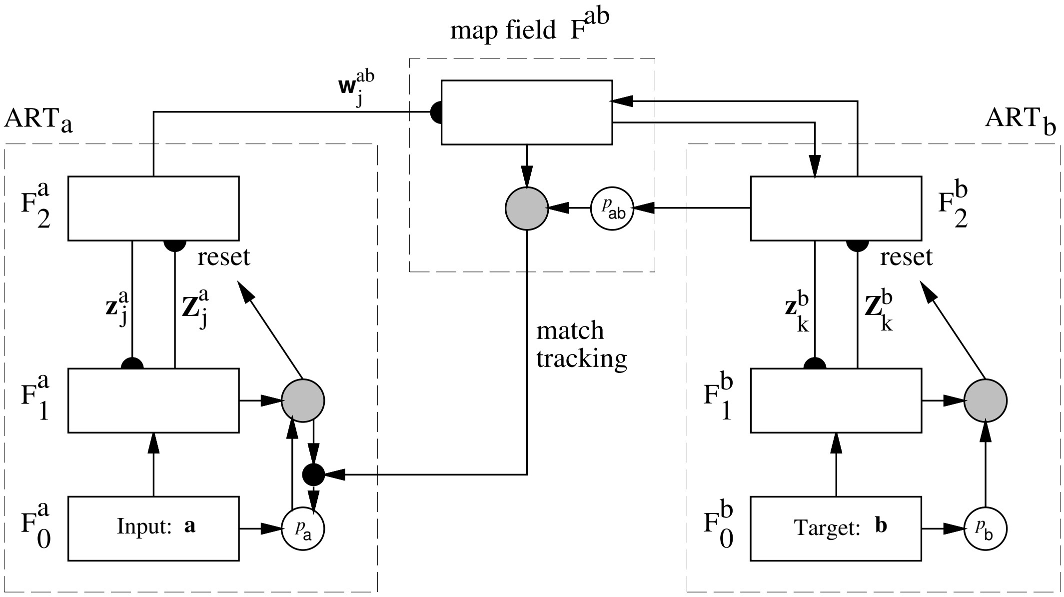

Figure 1: Architecture of the ARTMAP network. (This figure is a simplified version of the one that appeared in (Carpenter et al., 1992).)

ART$_a$ module will be raised by the minimal amount needed to cause mismatch with the current ART$_a$ input at the F_1^a layer. This will subsequently trigger a search for another ART$_a$ category, whose prediction will be matched against the current ART$_b$ category at the F^{ab} layer again. This process continues until the the network either finds an ART$_a$ category that predicts the category of the current target correctly, or creates a new F_2^a node and a corresponding link in the map field, which will learn the current input/target pair correctly. The ART$_a$ vigilance is then allowed to return to its resting level ($\overline{\rho_a}$).

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

3 The Match Tracking Anomaly (MTA)

The purpose of this article is to analyse the behaviour of the Match Tracking Anomaly (*MTA*), which was introduced in (Carpenter et al., 1991a, pages 584-585) as follows:

> "In particular, if $\mathbf{a} \subseteq \mathbf{z}_J^a$ [$\mathbf{a}$ is an ART$_a$ input here], match tracking makes $\rho_a > 1$, so $\mathbf{a}$ cannot activate another category in order to learn the new prediction. The following anomalous case can thus arise. Suppose that $\mathbf{a} = \mathbf{z}_J^a$ but the ART$_b$ input $\mathbf{b}$ mismatches the ART$_b$ expectation $\mathbf{z}_K^b$ previously associated with J. Then match tracking will prevent the recoding that would have associated $\mathbf{a}$ with $\mathbf{b}$. That is, the ARTMAP system with fast learning and choice will not learn the prediction of an exemplar that *exactly* matches a learned prototype when the new prediction contradicts the previous predictions of the exemplars that created the prototype. This situation does not arise when all ART$_a$ inputs $\mathbf{a}$ have the same number of 1's [...]"

(The same number of 1's in the input can be guaranteed by representing input patterns in *complement coding* (Carpenter et al., 1991b), according to which both input $\mathbf{a}$ and its complement $\mathbf{a}^c$ (where every bit in $\mathbf{a}$ is inverted) are shown to the network. Thus the norm of the new input vector $|(\mathbf{a}, \mathbf{a}^c)|$ will be constant.)

It seems obviously true that the anomalous situation does not arise since the norm of an ART$_a$ category prototype $\mathbf{z}_J^a$ can only be reduced through learning due to (2). Once this reduction occurs, no input $\mathbf{a}$ can be a proper subset of category prototype $\mathbf{z}_J^a$ (i.e. $|\mathbf{a} \cap \mathbf{z}_J^a| = |\mathbf{a}|$) if the inputs have the same norm (i.e. number of 1's). The problem can occur only when an input exactly matches a learned prototype that has not been recoded, which then produces a wrong prediction. In the classification problems in (Carpenter et al., 1991a), where target patterns were non-overlapping category codes, it would mean that the network had previously learned a different target category for the same input. So the match tracking anomaly could only arise if there were contradictory target categories for the same input in the training set.

In the following, we show that in more general learning tasks, in which target outputs are arbitrary binary vectors (not just simple category codes), the *MTA can* in fact arise even if the inputs to the network are complement coded, and the training set is non-contradictory, i.e. no two identical inputs have different target values.

4 Conditions for the Match Tracking Anomaly

In this section, we present conditions under which the match tracking anomaly can and cannot occur. It will be shown that the timing of presentation of the ART$_a$ inputs and ART$_b$ target outputs as well as the ART$_b$ vigilance level are of particular importance.

4.1 Timing conditions under which MTA will *never* arise

The following theorem states the condition under which the match tracking anomaly will *not* occur.

Theorem 1 *The match tracking anomaly will* never *arise in the ARTMAP network when complement coded input/target pairs from a non-contradicting training set are presented to the network such that in each learning trial the ART$_a$ input is presented first, and the network is allowed to make a prediction through the F^{ab} map field before the ART$_b$ (target) input is presented.*

If the ARTMAP network is able to make a prediction before the target is presented, the ART$_b$ module will be "primed" by the F_2^b layer, which will send down its category template to the F_1^b layer via the top-down pathways. We assume that the network dynamics are such that this top-down template will be tested first at the F_1^b layer as the target vector is registered into the F_0^b layer. If this top-down template (i.e. the network's prediction) and the target are sufficiently close to each other, no reset wave will be triggered, and the current F_2^b selection will be confirmed, and subsequently learned.

Proof: Let us assume that the network has previously learned to associate ART$_a$ category J with ART$_b$ category K through presentation of the $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ input/target pair. Let us assume furthermore that input pattern $\mathbf{a}^{(i)}$ is coded perfectly by ART$_a$ category J, i.e. $\mathbf{a}^{(i)} = \mathbf{z}_J^a$, which is the necessary condition for the *MTA* to arise if the inputs are complement coded (since $\mathbf{a}^{(i)} \subseteq \mathbf{z}_J^a$ can only be true if $\mathbf{a}^{(i)} = \mathbf{z}_J^a$). This can always be guaranteed by setting ρ_a to 1 so only perfect matches will be accepted at the vigilance test.

We now show that the *MTA* will *not* arise when the input/target pair $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ is presented again at some later stage during learning. Since there is no other input/target pair involving $\mathbf{a}^{(i)}$ for which the *MTA* condition *could* arise (assuming a non-contradictory training set), we shall conclude that the *MTA* will *never* arise under the circumstances stated in this theorem.

When the input/target pair $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ is presented again, the *MTA* condition can arise only if the network makes a wrong prediction first. Since the input set is non-contradictory, this can only be possible if the ART_b module selects a category other than K. For that to happen, however, first the ART_b vigilance test at the F_1^b layer has to fail with the network's prediction $\mathbf{z}_K^b$, i.e.

$$\frac{|\mathbf{b}^{(i)} \cap \mathbf{z}_K^b|}{|\mathbf{b}^{(i)}|} < \rho_b \tag{3}$$

must hold.

In complement coding, the norm of the target patterns is constant (here M_b), therefore

$$|\mathbf{z}_k^b| \geq \rho_b M_b \tag{4}$$

for all F_2^b category prototypes, and it follows from (3) and (4) that

$$|\mathbf{b}^{(i)} \cap \mathbf{z}_K^b| < |\mathbf{z}_K^b| \tag{5}$$

should also be true. However, $\mathbf{z}_K^b$ is already a subset of $\mathbf{b}^{(i)}$ since F_2^b node K has already coded input $\mathbf{b}^{(i)}$ before (hence the prediction), and therefore

$$|\mathbf{b}^{(i)} \cap \mathbf{z}_K^b| = |\mathbf{z}_K^b| \tag{6}$$

which directly contradicts (5). Therefore it is impossible that the (first) vigilance test will fail with the current prediction $\mathbf{z}_K^b$ when target $\mathbf{b}^{(i)}$ is re-presented at any later stage. Consequently, the *MTA* condition will never arise. □

4.2 Timing conditions under which MTA *can* arise

The following sections analyse conditions under which *MTA can* occur. In particular, we show that the *MTA* can occur when a target pattern is presented to the network *before* the input (i.e. the network is not able to make a prediction), even if all inputs to the network are complement coded. We also distinguish between two *MTA* conditions: one that is independent of the ART_b choice parameter (β_b), and another that is not.

Since in real-time situations, for which the ART systems were designed, there is no guarantee that ART_a inputs will always arrive first, this problem is a valid one that needs to be investigated.

Theorem 2 *The match tracking anomaly* can *arise in the ARTMAP network when complement coded input/target pairs from a non-contradicting training set are presented to the network such that some ART_b (target) inputs are presented* before *their corresponding ART_a input pair. There are at least two distinct conditions under which the MTA can occur: one that is independent of the ART_b choice parameter (β_b), and another that is not.*

We denote the "β_b-independent" *MTA* condition by *MTA-S* (for "*MTA* subset", as explained below), and the "β_b-dependent" one by *MTA-β*. We shall still use *MTA*, however, to refer to the match tracking anomaly in general, and shall be more specific only when necessary.

We shall prove this theorem by specifying initial conditions and sequences of training patterns that will lead to the *MTA* in each of the two cases mentioned in the theorem. We note, however, that there might exist other conditions as well that are different from these two, although the results of experiments we carried out do not suggest this.

Although the two conditions (*MTA-S* and *MTA-β*) are different in that one does not depend on β_b and the other does, they have much in common. In both cases,

- the network has previously seen the $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ input/target pair, which causes the *MTA*,
- the network has already an associative link between ART$_a$ category $J1$ and ART$_b$ category $K1$,
- category $J1$ perfectly encodes input $\mathbf{a}^{(i1)}$ (i.e. $\mathbf{z}^a_{J1} = \mathbf{a}^{(i1)}$),
- category $K1$ has already been recoded, i.e. $|\mathbf{z}^b_{K1}| < M_b$ (where M_b is the constant size of target patterns), and also includes target $\mathbf{b}^{(i1)}$ (i.e. $\mathbf{z}^b_{K1} \subset \mathbf{b}^{(i1)}$).
- there exists another ART$_b$ category ($K2$) "nearby", which is more specific than $K1$ (ie $|\mathbf{z}^b_{K2}| > |\mathbf{z}^b_{K1}|$),
- ART$_a$ vigilance parameter ρ_a is assumed to be high enough so no other $\mathbf{a}^{(j)} \neq \mathbf{a}^{(i1)}$ inputs from the training set will be coded by category $J1$ (this limit depends on the particular training set, but the condition can always be guaranteed by $\rho_a = 1$),
- ART$_b$ vigilance parameter ρ_b is assumed to be low enough that target $\mathbf{b}^{(i1)}$ will be found sufficiently close to both ART$_b$ category $K1$ and $K2$ (i.e. $|\mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K1}| > \rho_b M_b$ and $|\mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K2}| > \rho_b M_b$).
  Note that this also implies that ρ_b must be lower than 1.

In both cases, the *MTA* will arise when the $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ input/target pair is presented again, and ART$_b$ category $K2$ "captures" target input $\mathbf{b}^{(i1)}$, which contradicts the network's prediction of ART$_b$ category $K1$ (through its previously learned $J1 \rightarrow K1$ association). Since input $\mathbf{a}^{(i1)}$ matches ART$_a$ category $J1$ perfectly, the network will not be able to find a better match through match tracking, therefore the *MTA* will emerge.

The difference between the two conditions is that in the *MTA-S* case both the $K1$ and $K2$ ART$_b$ category prototypes ($\mathbf{z}^b_{K1}$ and $\mathbf{z}^b_{K2}$) are proper subsets of target $\mathbf{b}^{(i1)}$ (hence the name *MTA-S*), i.e. $|\mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K1}| = |\mathbf{z}^b_{K1}|$ and $|\mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K2}| = |\mathbf{z}^b_{K2}|$, therefore "capturing" of $\mathbf{b}^{(i1)}$ by $K2$ will not depend on β_b (see the paragraph on the ART network's category selection in case of proper subsets in section 2). In the *MTA-β* case, ART$_b$ category $K2$ prototype $\mathbf{z}^b_{K2}$ is *not* a proper subset of target $\mathbf{b}^{(i1)}$ (while $K1$ prototype $\mathbf{z}^b_{K1}$ still is), so the selection of ART$_b$ category $K1$ or $K2$ will depend on the strength of the bottom-up match between $\mathbf{z}^b_{K2}$ and target $\mathbf{b}^{(i1)}$, which in turn depends on the value of β_b due to (1).

Figures 2 and 3 illustrate graphically the *MTA-S* and *MTA-β* conditions, respectively.

We prove theorem 2 by showing the existence of *MTA* in either of the two conditions mentioned in the theorem (defined as *MTA-S* and *MTA-β* above). We specify initial conditions and training sequences in either case that will lead to the *MTA*. We must point out, however, that these training sequences are artificial (and simplified) examples that are only used to prove the *existence* of *MTA*. In real applications, the necessary initial conditions (as a result of prior learning) as well as the "critical training sequence" can be observed through a variety of actual training sequences. Moreover, the two distinct *MTA* conditions (*MTA-S* and *MTA-β*) can even have a combined effect if β_b is above a threshold, which will

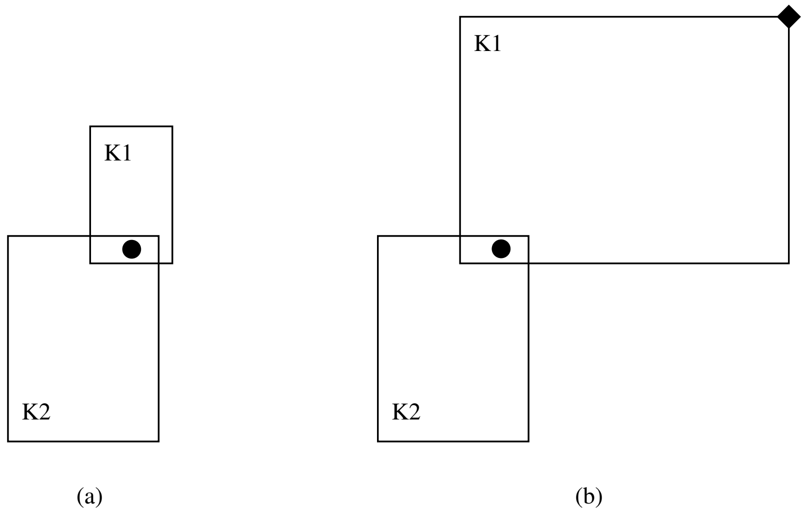

Figure 2: Graphical illustration of the *MTA-S* condition. Target patterns are represented as points on the two-dimensional plane. Categories are represented as rectangles, the area of each rectangle being proportional to the size of the category they represent. Figure 2a shows that category $K1$ is initially smaller (i.e. more specific) than $K2$, and target $\mathbf{b}^{(i1)}$ (circle) is inside (i.e. proper subset of) both $K1$ and $K2$. If at some later stage, another target input is presented (diamond, in figure 2b) that causes $K1$ to generalise to include that input; $K1$ could grow larger than $K2$ as a result. When target $\mathbf{b}^{(i1)}$ is re-presented, this time category $K2$ will capture it since it has now become more specific than $K1$. This will cause the anomaly to emerge as the network still predicts $K1$ through its previously learned $J1 \rightarrow K1$ link. See detailed discussion in section 4.2.1.

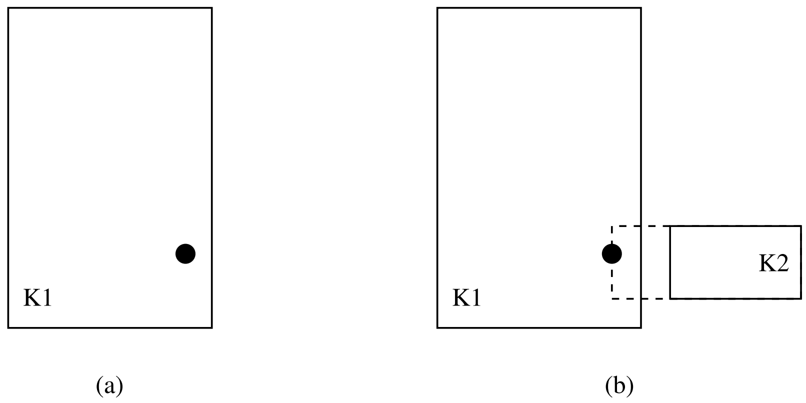

Figure 3: Graphical illustration of the *MTA-*β condition. Target patterns are represented as points on the two-dimensional plane. Categories are represented as rectangles, the area of each rectangle being proportional to the size of the category they represent. Figure 3a shows category $K1$ and target input $\mathbf{b}^{(i1)}$ (circle) inside it. As a result of further training, some other target(s) have created category $K2$ "near" $K1$ (see figure 3b). $K2$ is more specific than $K1$, and initially does not include target $\mathbf{b}^{(i1)}$, although it could (with low enough ρ_b vigilance). When target $\mathbf{b}^{(i1)}$ is re-presented, it will be captured by category $K2$ if it achieves a higher bottom-up activation than $K1$. This depends on the degree of overlap between $\mathbf{b}^{(i1)}$ and $\mathbf{z}^b_{K2}$, *and* the choice parameter β_b due to (1). See detailed discussion in section 4.2.2.

produce an even richer variety of paths leading to the *MTA*. These were clearly observed in the experiments we carried out, the results of which are presented in section 5.

4.2.1 Proof of the *MTA-S* condition

In this section, we show how the ARTMAP network can be led to the *MTA-S* condition. First we shall assume that the network is in a particular state after prior learning, and show that presenting the input/target pair sequence $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$, $(\mathbf{a}^{(i2)}, \mathbf{b}^{(i2)})$, $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ then will cause the *MTA* to arise. Then we show that the particular (assumed) state of the network before this sequence *can* result from prior learning.

Let us assume that the network has previously learned to associate ART$_a$ category $J1$ with ART$_b$ category $K1$. Also, ART$_a$ category $J1$ has perfectly coded input $\mathbf{a}^{(i1)}$, and there is no other input in that category, i.e.

$$\mathbf{z}^a_{J1} = \mathbf{a}^{(i1)}. \tag{7}$$

There is also another input $(\mathbf{a}^{(i2)})$, for which

$$|\mathbf{a}^{(i2)} \cap \mathbf{z}^a_{J1}| < \rho_a M_a \tag{8}$$

where M_a is the norm of the input patterns.

Furthermore, target inputs $\mathbf{b}^{(i1)}$ and $\mathbf{b}^{(i2)}$, and another ART$_b$ category, $K2$, satisfy the following conditions:

$$\mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K1} = \mathbf{z}^b_{K1}, \quad \mathbf{b}^{(i1)} \cap \mathbf{z}^b_{K2} = \mathbf{z}^b_{K2}\ , \tag{9}$$

i.e. both category prototypes are proper subsets of target $\mathbf{b}^{(i1)}$, and

$$|\mathbf{b}^{(i2)} \cap \mathbf{z}^b_{K1}| > |\mathbf{b}^{(i2)} \cap \mathbf{z}^b_{K2}|\ , \tag{10}$$

i.e. target $\mathbf{b}^{(i2)}$ matches $K1$ category prototype $\mathbf{z}^b_{K1}$ better than $\mathbf{z}^b_{K2}$, and

$$|\mathbf{z}^b_{K1}| > |\mathbf{z}^b_{K2}| > |\mathbf{b}^{(i2)} \cap \mathbf{z}^b_{K1}| \geq \rho_b M_b\ , \tag{11}$$

i.e. category $K1$ is more specific than $K2$, and matches target $\mathbf{b}^{(i2)}$ well enough, but not as well as category prototype $\mathbf{z}^b_{K2}$.

If we now present the input/target pair $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ first, ART$_b$ category $K1$ will be selected due to (9) and (11), which will be confirmed by the network through the selection of category $J1$ upon presenting input $\mathbf{a}^{(i1)}$ due to (7), and the existing $J1 \rightarrow K1$ association[3]. Due to (7) and (9), category prototypes $\mathbf{z}^a_{J1}$ and $\mathbf{z}^b_{K1}$ will not change during learning.

Next, input/target pair $(\mathbf{a}^{(i2)}, \mathbf{b}^{(i2)})$ is presented. Here, target input $\mathbf{b}^{(i2)}$ will select ART$_b$ category $K1$ again due to (10) and (11). The network will then either find an existing ART$_a$ category associated with ART$_b$ category $K1$, or will create a new ART$_a$ category to learn this association. In either case, ART$_a$ category $J1$ will not be affected due to (8). ART$_b$ category $K1$, however, will be modified this time, i.e.

$$\mathbf{z}^b_{K1}(new) = \mathbf{b}^{(i2)} \cap \mathbf{z}^b_{K1}(old).$$

[3] Alternatively, ART$_a$ category $J1$ can be a newly created category, which will learn to associate ART$_b$ category $K1$ during this trial, but it does not affect the proof.

Step	Input $\mathbf{a}^{(i)}$	ART$_a$ cat. #	$F_2^a \to F_2^b$ Mapping	ART$_b$ categories		Target $\mathbf{b}^{(i)}$
...	...	...	...	$\mathbf{z}_{K1}^b$	$\mathbf{z}_{K2}^b$	...
k	$\mathbf{a}^{(i1)}$	$J1$	$J \to K1$	$\triangleright$ $\mathbf{z}_{K1}^b$ $\triangleleft$	$\mathbf{z}_{K2}^b$	$\mathbf{b}^{(i1)}$
k+1	$\mathbf{a}^{(i2)}$	$\neq J1$	$\neq J1 \to K1$	? $\mathbf{b}^{(i2)} \cap \mathbf{z}_{K1}^b$ $\triangleleft$	? $\mathbf{z}_{K2}^b$	$\mathbf{b}^{(i2)}$
k+2	$\mathbf{a}^{(i1)}$	$J1$	$J \to 1?2$	$\triangleright$ $\mathbf{b}^{(i2)} \cap \mathbf{z}_{K1}^b$	$\mathbf{z}_{K2}^b$ $\triangleleft$	$\mathbf{b}^{(i1)}$

Table 1: A sequence of input/target patterns that causes the *MTA* under conditions discussed in the text. The ART$_a$ categories are not shown here, only the index of the winning node in each step. Two of the developed ART$_b$ categories, which are referred to in the text, are shown. The symbol "$\triangleright$" appears beside the ART$_b$ category that the network predicts through the $F_2^a \to F_2^b$ associations in each trial ("?" means it does not matter which category is predicted). The symbol "$\triangleleft$" denotes the category that is selected by the ART$_b$ module before the corresponding input is presented. So in step k, the network's prediction is confirmed (i.e. "$\triangleright$" and "$\triangleleft$" appear at the same category), while there is a mismatch in step k+2, which eventually causes the *MTA*.

Then input/target pair $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ is presented again. Here, target input $\mathbf{b}^{(i1)}$ will read out ART$_b$ category $K2$ this time due to (9) and (11). However, $\mathbf{a}^{(i1)}$ input will again read out ART$_a$ category $J1$ since it still matches $\mathbf{z}_{J1}^a$ perfectly. As a result the $J1 \to K1$ prediction will not be confirmed at the F^{ab} layer, and the match tracking process will be triggered. This will, however, cause *MTA* since $\mathbf{a}^{(i1)}$ matches $\mathbf{z}_{J1}^a$ perfectly (i.e. $|\mathbf{a}^{(i1)} \cap \mathbf{z}_{J1}^a||\mathbf{a}^{(i1)}|^{-1} = 1$), and ρ_a cannot be raised above 1.

Table 1 illustrates the above sequence and its effect on the network.

To complete the proof, we also need to show that such initial assumption for the ART$_b$ category prototypes $\mathbf{z}_{K1}^b$ and $\mathbf{z}_{K2}^b$ and target pairs $\mathbf{b}^{(i1)}$ and $\mathbf{b}^{(i2)}$ is "realistic", i.e. they can result from previous training of the network.

It is, however, not trivial to see. We cannot do it simply by first creating the proper $J1 \to K1$ association and $\mathbf{z}_{J1}^a$, $\mathbf{z}_{K1}^b$ prototypes, and then another ART$_b$ category ($K2$ with a prototype $\mathbf{z}_{K2}^b$), which is also a proper subset of another input (here $\mathbf{b}^{(i1)}$, and see (9) as the assumption) such that $|\mathbf{z}_{K1}^b| > |\mathbf{z}_{K2}^b|$, as was assumed in (11). The network would not have selected category $K2$ when $K1$ is clearly a better choice. This situation can only come about if the required $J1 \to K1$ association and $\mathbf{z}_{J1}^a$, $\mathbf{z}_{K1}^b$ prototypes are created first, and then a particular sequence of input/target patterns is presented, through which a new ART$_b$ prototype ($\mathbf{z}_{K2}^b$) is created first, which is then gradually "eroded" (i.e. its norm decreased) in small enough pieces until condition (11) is reached, while leaving $\mathbf{z}_{K1}^b$ unchanged in each trial. For simplicity, here we assume the smallest possible size for a "piece" by which $|\mathbf{z}_{K2}^b|$ is decreased, which is one.

More precisely, we present a sequence of input/target pairs $(\mathbf{a}^{(j)}, \mathbf{b}^{(j)}, j = 1, 2, ...)$ such that each input and the category prototypes in question will satisfy

$$|\mathbf{b}^{(j)} \cap \mathbf{z}_{K1}^b| = 1 \ , \tag{12}$$

and

$$|\mathbf{z}_{K2}^b(new)| = |\mathbf{b}^{(j)} \cap \mathbf{z}_{K2}^b(old)| = |\mathbf{z}_{K2}^b(old)| - 1 \tag{13}$$

until $|\mathbf{z}^b_{K1}| > |\mathbf{z}^b_{K2}|$ (see condition (11)).
Now we need to show that in each step

$$\frac{1}{\beta + |\mathbf{z}^b_{K1}|} < \frac{|\mathbf{z}^b_{K2}| - 1}{\beta + |\mathbf{z}^b_{K2}|}. \tag{14}$$

In other words, ART$_b$ category $K2$ will be selected (over $K1$) at the F_2^b layer and allowed to change, while $K1$ remains unchanged.

This can be shown by considering the following two cases:
Case 1: $2 < |\mathbf{z}^b_{K2}| \leq |\mathbf{z}^b_{K1}|$.

Here, (14) will hold since the numerator of the right hand side is greater than 1, and the denominator is smaller than that of the left hand side.

Case 2: $2 < |\mathbf{z}^b_{K1}| < |\mathbf{z}^b_{K2}|$.

To see that (14) will hold in this case as well, we need to see first that

$$\frac{1}{\beta + |\mathbf{z}^b_{K1}|} < \frac{1}{\beta + |\mathbf{z}^b_{K1}|} + \frac{|\mathbf{z}^b_{K1}| - 2}{\beta + |\mathbf{z}^b_{K1}|} = \frac{|\mathbf{z}^b_{K1}| - 1}{\beta + |\mathbf{z}^b_{K1}|}. \tag{15}$$

Now (14) holds again, since the

$$f(x) = \frac{x-1}{\beta + x}$$

function increases monotonically when $x > 2$ (here $|\mathbf{z}^b_{K1}| > 2$) with arbitrary $\beta > 0$.

Therefore, the assumptions about the network's initial state (see inequalities (9) to (11)) were realistic. This completes the proof of the *MTA-S* condition in theorem 2. □

A concrete example with a complete sequence leading to the *MTA-S* is given in table 2.

4.2.2 Proof of the *MTA-β* condition

This section shows that there are situations where the *MTA* can occur if β_b is *above* a certain threshold[4]. As in the previous section, we provide initial conditions and a training sequence that will lead to the *MTA-β*.

Let us take three input/target pairs $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$, $(\mathbf{a}^{(i2)}, \mathbf{b}^{(i2)})$ and $(\mathbf{a}^{(i3)}, \mathbf{b}^{(i3)})$. The patterns are complement coded (as was assumed), thus the norm of the input and target vectors ($\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$) is constant (M_a and M_b, respectively).

Firstly, let ρ_a be high enough to cause ART$_a$ to allocate an uncommitted F_2^a node for each new, previously unseen pattern. This can always be guaranteed if $\rho_a = 1$.

The target patterns satisfy the following conditions:

$$|\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i2)}| = L_{12} \quad (L_{12} > 0), \tag{16}$$

[4]For clarity, we shall refer to β_b as β from here on unless the distinction from β_a needs to be made clear.

Step	Input $\mathbf{a}^{(i)}$	ART$_a$ cat #	$F_2^a \to F_2^b$ Mapping		ART$_b$ categories $\mathbf{z}_1^b$		$\mathbf{z}_2^b$		Target $\mathbf{b}^{(i)}$
1	1 1 1 1 1	1	$1 \to 1$		1 1 1 1 1	$\triangleleft$	unused		1 1 1 1 1
2	1 1 1 0 0	2	$2 \to 1$		1 1 1 x x	$\triangleleft$	unused		1 1 1 0 0
3	0 0 0 1 1	3	$3 \to 2$		1 1 1 x x		0 0 0 1 1	$\triangleleft$	0 0 0 1 1
4	1 0 0 1 1	4	$4 \to 2$		1 1 1 x x		x 0 0 1 1	$\triangleleft$	1 0 0 1 1
5	0 1 0 1 1	5	$5 \to 2$		1 1 1 x x		x x 0 1 1	$\triangleleft$	0 1 0 1 1
6	0 0 1 1 1	6	$6 \to 2$		1 1 1 x x		x x x 1 1	$\triangleleft$	0 0 1 1 1
7	1 1 1 1 1	1	$1 \to 1$	$\triangleright$	1 1 1 x x	$\triangleleft$	x x x 1 1		1 1 1 1 1
8	1 0 0 0 0	7	$7 \to 1$		1 x x x x	$\triangleleft$	x x x 1 1		1 0 0 0 0
9	1 1 1 1 1	1	$1 \to 1?2$	$\triangleright$	1 x x x x		x x x 1 1	$\triangleleft$	1 1 1 1 1

Table 2: An example sequence of input/target pairs that lead the ARTMAP network to the *MTA-S*. Here, each input pattern is identical to its corresponding target (i.e. auto-associative learning), and the norm of all patterns is 5. For clarity, the complement patterns are not shown. In category prototypes, a "x" means "don't care", which corresponds to a "00" bit pair in complement coding. For the ART$_a$ module, only the winning nodes are shown. We assume that the ART$_a$ vigilance is high enough (e.g. $\rho_a = 0.9$) so the 11111 pattern will be the only one in its own category. ART$_b$ vigilance is less than 1/5 (e.g. $\rho_b = 0.1$). The use of the symbols "$\triangleright$" and "$\triangleleft$" is the same as in table 1. (The absence of the "$\triangleright$" symbol in rows 1-6 indicates that the network did not have a prediction in these trials, and the corresponding $J \to K$ links were created in those trials.) The $F_2^a \to F_2^b$ mappings that the network created or used in each trial are also shown. Steps 1 to 6 show the "erosion" process of ART$_b$ category $\mathbf{z}_2^b$, while steps 7 and 8 "prepare" the *MTA* condition that occurs in step 9. (Here, concrete category numbers are used, since these are the only patterns presented to the network.)

$$|\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i3)}| = L_{13} \quad (L_{13} > 0), \tag{17}$$

such that

$$L_{13} > L_{12}, \tag{18}$$

and

$$|\mathbf{b}^{(i2)} \cap \mathbf{b}^{(i3)}| = 0 \ . \tag{19}$$

From (16), (17) and (19), it follows that[5]

$$L_{12} + L_{13} \leq M \ \ . \tag{20}$$

Also, let

$$\rho_b < \frac{L_{12}}{M} \ . \tag{21}$$

We now show that under these conditions, presenting the above patterns in the order of $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$, $(\mathbf{a}^{(i2)}, \mathbf{b}^{(i2)})$, $(\mathbf{a}^{(i3)}, \mathbf{b}^{(i3)})$, $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ will cause MTA to emerge if β is above a certain threshold.

In this case, we start with an untrained network, and look at the behaviour of the ART$_b$ module. (We also assume that category nodes become committed in the order of 1, 2, 3, ..., and thus these numbers will be used instead of $J1$, $K1$, etc. used in the previous sections.)

In the first step, when the $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ pair is presented, $\mathbf{b}^{(i1)}$ will be learned by category 1, and the corresponding $1 \rightarrow 1$ link will also be created (since input $\mathbf{a}^{(i1)}$ will be learned by ART$_a$ category 1).

In the second step, target $\mathbf{b}^{(i2)}$ will be captured by category 1, as it will achieve the maximal bottom-up activation (no other F_2^b nodes are active as yet, and the initial bottom-up weights are chosen to be small enough for the uncommitted nodes), and

$$|\mathbf{b}^{(i2)} \cap \mathbf{z}_1^b| = L_{12} > \rho_b M \ ,$$

due to (21). Prototype $\mathbf{z}_1^b$ is then modified to learn $\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i2)}$.

In the third step, target $\mathbf{b}^{(i3)}$ will create a new ART$_b$ category $\mathbf{z}_2^b$ as there will be a total mismatch between $\mathbf{b}^{(i3)}$ and $\mathbf{z}_1^b$ due to (19).

The interesting case is step four, when pattern $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$ is presented again. Here both existing ART$_b$ categories will match the input sufficiently due to (18) and (21), so the selection will be based entirely on the bottom-up activation levels, which can be calculated in the following way.

The bottom-up weights for ART$_b$ categories 1 ($\mathbf{z}_1^b$) and 2 ($\mathbf{z}_2^b$) will be

$$\frac{1}{\beta + L_{12}} \ \text{and} \ \frac{1}{\beta + M} \ ,$$

respectively. The bottom-up activations will therefore be

$$\frac{L_{12}}{\beta + L_{12}} \ \text{and} \ \frac{L_{13}}{\beta + M} \ ,$$

[5] For clarity, we refer to M_b simply as M since M_a is irrelevant to the proof here.

Step	Input	ART$_a$ categories $\mathbf{z}_1^a$	$\mathbf{z}_2^a$	$\mathbf{z}_3^a$	$F_2^a \rightarrow F_2^b$ Mapping	ART$_b$ categories $\mathbf{z}_1^b$	$\mathbf{z}_2^b$	Target
1	$\mathbf{a}^{(i1)}$	$\mathbf{a}^{(i1)}$	unused	unused	$1 \rightarrow 1$	$\mathbf{b}^{(i1)}$	unused	$\mathbf{b}^{(i1)}$
2	$\mathbf{a}^{(i2)}$	$\mathbf{a}^{(i1)}$	$\mathbf{a}^{(i2)}$	unused	$2 \rightarrow 1$	$\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i2)}$	unused	$\mathbf{b}^{(i2)}$
3	$\mathbf{a}^{(i3)}$	$\mathbf{a}^{(i1)}$	$\mathbf{a}^{(i2)}$	$\mathbf{a}^{(i3)}$	$3 \rightarrow 2$	$\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i2)}$	$\mathbf{b}^{(i3)}$	$\mathbf{b}^{(i3)}$
4	$\mathbf{a}^{(i1)}$	$\mathbf{a}^{(i1)}$	$\mathbf{a}^{(i2)}$	$\mathbf{a}^{(i3)}$	$1 \rightarrow 1?2$	$\mathbf{b}^{(i1)} \cap \mathbf{b}^{(i2)}$	$\mathbf{b}^{(i3)}$	$\mathbf{b}^{(i1)}$

Table 3: The sequence of input/target pairs $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$, $(\mathbf{a}^{(i2)}, \mathbf{b}^{(i2)})$, $(\mathbf{a}^{(i3)}, \mathbf{b}^{(i3)})$, $(\mathbf{a}^{(i1)}, \mathbf{b}^{(i1)})$, and the developed ARTMAP categories and $F_2^a \rightarrow F_2^b$ associations. Since ρ_a is high enough, each new input creates a new ART$_a$ category in steps 1, 2 and 3. The winning ART$_b$ categories in each step are the ones that appear on the right-hand side of the arrow in the "$F_2^a \rightarrow F_2^b$ Mapping" column. The re-presentation of the first pair in step 4 will cause the *MTA* if the bottom-up activation of ART$_b$ node 2 is higher than that of node 1. (Here, concrete category numbers are used, since these are the only patterns presented to the network.)

respectively, due to (1). For fixed L_{12}, L_{13} and M, β can always be set such that the $\mathbf{b}^{(i1)}$ target will select either category 1 or 2. This can be seen by examining the

$$\frac{L_{12}}{\beta + L_{12}} < \frac{L_{13}}{\beta + M} \tag{22}$$

inequality. If (22) is satisfied, then ART$_b$ category 2 will be selected, and thus the *MTA* will emerge.

Rearranging (22), we get

$$\beta > \frac{L_{12}(M - L_{13})}{L_{13} - L_{12}}. \tag{23}$$

For fixed L_{12}, L_{13} and M that satisfies (20), the occurrence of the *MTA* condition will therefore be determined by the choice of β. Since the recommended setting of β is to be small (Carpenter et al., 1991a, p. 579), it is worthwhile looking at how the lower limit of β in (23) is affected by the choice of L_{12}, L_{13} and M (which are determined by the problem at hand).

To get the minimum value of the right-hand side of (23), we need to minimise $L_{12}(M - L_{13})$, and maximise $L_{13} - L_{12}$. The minimum values for L_{12} and $M - L_{13}$ are 1 and L_{12} (from (20)), respectively. The resulting choice for L_{13} ($= M - L_{12}$) will maximise $L_{13} - L_{12}$. So the lower limit for β is

$$\beta_{min} = \frac{1}{M - 2} \quad . \tag{24}$$

Therefore, for a given M_b, β_b should always be chosen to be smaller than $\frac{1}{M_b - 2}$ to avoid the *MTA*-β for patterns that satisfy conditions (16) to (21).

Table 3 shows the developed ART$_a$ and ART$_b$ categories as well as $F_2^a \rightarrow F_2^b$ links in response to the input/target pairs discussed here.

This completes the proof of the *MTA*-β condition in theorem 2 as well as theorem 2 as a whole. □

Step	Input	ART$_a$ categories $\mathbf{z}_1^a$	$\mathbf{z}_2^a$	$\mathbf{z}_3^a$	$F_2^a \to F_2^b$ Mapping	ART$_b$ categories $\mathbf{z}_1^b$	$\mathbf{z}_2^b$	Target
1	1 1 1 1	1 1 1 1	unused	unused	$1 \to 1$	1 1 1 1	unused	1 1 1 1
2	1 0 0 0	1 1 1 1	1 0 0 0	unused	$2 \to 1$	1 x x x	unused	1 0 0 0
3	0 1 1 1	1 1 1 1	1 0 0 0	0 1 1 1	$3 \to 2$	1 x x x	0 1 1 1	0 1 1 1
4	1 1 1 1	1 1 1 1	1 0 0 0	0 1 1 1	$1 \to 1?2$	1 x x x	0 1 1 1	1 1 1 1

Table 4: An example input/target pattern sequence that demonstrates the *MTA*-β condition. Here, each input pattern is identical to its corresponding target, and the norm of all patterns is 4. The ARTMAP categories and links that were developed in response to this sequence are also shown. For clarity, the complement patterns are not shown here. In category prototypes, a "x" means "don't care", which corresponds to a "00" pair in complement coding. The ART$_a$ and ART$_b$ vigilance parameters (ρ_a, ρ_b) are set to 0.8 and 0.3, respectively. If $\beta > \frac{1}{2}$, the *MTA* will occur in step 4 when input/target pair (1111, 1111) is re-presented, because ART$_b$ category 2 will be selected. This contradicts the network's prediction of category 1 through the link that was created in step 1. The *MTA* will thus emerge.

A concrete example is given in table 4, with $M = 4$, $L_{13} = 3$, and $L_{12} = 1$. Here, if $\beta > \frac{1}{2}$, the *MTA* condition will arise.

4.3 ART$_b$ vigilance level and *MTA*

This section deals with an important special case of the ARTMAP network where the ART$_b$ vigilance level (ρ_b) is 1. This setting is commonly used in pattern classification applications where outputs represent target classes. The following theorem reassures us that the *MTA* will *not* arise when $\rho_b = 1$.

Theorem 3 *The match tracking anomaly will* never *arise in the ARTMAP network if* $\rho_b = 1$ *regardless of the timing used to present complement coded input/target pairs from a non-contradicting training set.*

Proof: If $\rho_b = 1$, then no ART$_b$ category will be recoded during training, i.e. $|\mathbf{z}_k^b| = M_b$ for all ART$_b$ category prototypes. Therefore, upon presentation of input/target pair $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ that the network has seen before, either a previously learned $J1 \to K$ association will be confirmed, in which case $\mathbf{z}_{J1}^a = \mathbf{a}^{(i)}$, or a new $J2 \to K$ association will be created through match tracking, in which case $|\mathbf{z}_{J2}^a| < M_a$ and therefore raising ρ_a will not cause *MTA*. In neither of these cases will ART$_b$ category K be recoded so the *MTA* will not arise. □

5 Experimental results

In the previous sections, we proved the existence of the match tracking anomaly by specifying initial conditions and constructing training sequences that lead to the *MTA*. These were relatively simple and "clean" examples that let us concentrate on the underlying problem. In reality, however, there are a number of different ways the network can be driven into the *MTA*, which are dependent on the training set and the presentation order of input/target pairs during training. For example, before any step of the above example training sequences, there can be a number of input/target presentations that are irrelevant to the occurrence of a given *MTA*. This section presents experimental results that will demonstrate the existence of the two *MTA* conditions analysed here.

We carried out experiments on a machine learning benchmark database. Unlike in (Carpenter et al., 1991a), where target outputs were codes for correct output categories, here the ARTMAP network was used to develop two-level class hierarchies from the training set (Bartfai, 1994). In particular, each input was identical to its target pair, and both the ART_a and ART_b modules developed clusterings of patterns from the same training set at different levels of vigilance.

For the experiments, we used the "zoo" machine learning benchmark database (Murphy and Aha, 1992). It contains 101 instances of animals described with 18 attributes. Out of these attributes, we used the 15 boolean ones that indicate the presence or absence of certain features like "hair", "aquatic", "domestic" and so on. We also used the "number of legs" attribute, which is a set of 6 integers. The "type" (or class) attribute was ignored.[6] The patterns were presented to the ARTMAP network in *complement coding*. However, complement coding was defined for binary patterns only in (Carpenter et al., 1991a). Since one of the features here (see "legs" above) is not binary, we had to encode that feature as well in a way that was compatible with complement coding. For this, we have defined a coding scheme that we call *generalised complement coding*, which can be applied to arbitrary-sized sets.

According to generalised complement coding, a value v_i of a set $\mathcal{S} = \{v_1, \ldots, v_s\}$ will be mapped to a binary vector ($\mathbf{c}$) of length s such that

$$c_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, \ldots, s.$$

So, for example, if an attribute takes on the value 4 from the set of $\{0, 2, 4, 5, 6, 8\}$, then the corresponding binary vector will be 001000. Note that if $s = 2$, this scheme will be equivalent to the original form of complement coding.

This way, we can keep the size of the (binary) input pattern to the network constant, thereby implementing a form of "binary normalisation" of the input. In the case of the database used in the above way, the input and target patterns were 36-element binary vectors, in which the number of 1's was the same (16).

The simulations were carried out using a public domain neural network simulator program (`PlaNet`, see (Miyata, 1991)) running under Unix and X-windows.

The network parameters were chosen to be similar to those presented in (Carpenter et al., 1991a). In particular, the initial values of the bottom-up weights in both ART modules were chosen such that F_2 nodes become active in the order $j = 1, 2, ...$ and were small enough so the network selected an uncommitted ART_a or ART_b node only if $|\mathbf{a}^{(i)} \cap \mathbf{z}_j^a| = 0$ or $|\mathbf{b}^{(i)} \cap \mathbf{z}_k^b| = 0$, respectively, for all committed nodes. Also, the β_a and β_b parameters were sufficiently small that, among committed nodes, bottom-up activations

$$\frac{|\mathbf{a}^{(i)} \cap \mathbf{z}_j^a|}{\beta_a + |\mathbf{z}_j^a|} \quad \text{and} \quad \frac{|\mathbf{b}^{(i)} \cap \mathbf{z}_k^b|}{\beta_b + |\mathbf{z}_k^b|}$$

were determined by the sizes of $|\mathbf{a}^{(i)} \cap \mathbf{z}_j^a|$ and $|\mathbf{b}^{(i)} \cap \mathbf{z}_k^b|$ relative to $|\mathbf{z}_j^a|$ and $|\mathbf{z}_k^b|$, respectively. This also satisfied the $\beta_b < \beta_{min}$ condition in section 4.2.2. The network was used in *fast learning* mode, i.e. during learning, the connection weights were allowed to reach their asymptotic values while the current input/target pair was presented.

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

β	Total Epochs	*MTA* in epochs Total	%	*MTA* in trials Total	%
0.001	215	30	14.0	78	0.4
0.01	215	30	14.0	78	0.4
0.1	215	30	14.0	78	0.4
1.0	252	49	19.4	119	0.5
10.0	370	228	61.6	1270	3.4
100.0	491	250	50.9	1662	3.4

Table 5: Frequencies of *MTA* occurrence at different levels of the β choice parameter when the ARTMAP network was trained on the "zoo" database (see details in text). Each row shows the results of the same set of 100 training sessions. In each training session, the network was trained completely on the training set, i.e. training stopped when no further weight changes occurred in the network when for a full training cycle (or epoch). The ART$_a$ and ART$_b$ vigilance parameters (ρ_a, ρ_b) were 0.4 and 0.1, respectively. The "Total Epochs" column shows the total number of presentations of the entire training set during the 100 training sessions. The "*MTA* in epochs" columns show the number of epochs in which *MTA* occurred at least once (both in absolute and percentage). The "*MTA* in trials" columns show the total number of learning trials (i.e. input/target presentations) in which *MTA* occurred during the 100 training sessions.

Training proceeded by showing input/target pairs repeatedly, until the network stabilised itself, i.e. each input read out its prototype directly, so no further learning took place.

Table 5 shows the results of measurements we carried out into the frequency of *MTA* when the network was trained on the "zoo" database. It can be seen that with small β values, the *MTA* condition occurred in about 14% of the training epochs (i.e. one full presentation of the entire training set). This accounts for only 0.4% of the total number of input/target presentations. It demonstrates that the *MTA* condition can arise even when β is small[7]. These are the *MTA-S* cases that were discussed in section 4.2.1. However, we can see a sharp rise in the frequency of *MTA* as β increases beyond a certain threshold. This demonstrates that increasing β does affect the *MTA* (see *MTA*-β discussed in section 4.2.2). The fact that this threshold appears in the range of $[0.1, 1]$, which is slightly higher than the limit $(\frac{1}{M-2} = 14^{-1} \approx 0.07)$ that was derived in section 4.2.2, suggests that *MTA*-β does not take immediate effect, probably because here it is superimposed on the *MTA-S*, or that the patterns did not satisfy all of the conditions from (16) to (21), (19) in particular. In the latter case, even the theoretical limit would possibly be lower than what was derived in section 4.2.2.

We also observed that, on average, two out of every three *MTA* occurrences were due to race conditions at the bottom-up activations (i.e. two or more F_2^b nodes achieved the same maximal bottom-up activation for the current input). In the simulations, we broke these ties by declaring the node with the smallest index the winner. As a result, some true *MTA* conditions may have remained unnoticed as the above tie breaking method chose the correct category (whereas another one would not have). We believe, however, that this did not affect the results significantly. (Besides, the aim of these experiments were to demonstrate that

[7]We tried even smaller values for β. The numbers did not change until rounding took effect due to the finite accuracy of representable numbers in computer simulations. Very often, the two effects (*MTA* and rounding) cancelled each other out suggesting falsely that the *MTA* condition will disappear if β becomes very small. This may serve as a warning about the limits of computer simulations.

Vigilance		*MTA* % of	
ρ_b	ρ_a	Epochs	Trials
0.1	0.1	0	0
	0.4	14.0	0.4
	0.7	36.8	1.0
	1.0	80.2	7.7
0.4	0.1	0	0
	0.4	0	0
	0.7	50.0	1.8
	1.0	94.9	9.9
0.7	0.1	0	0
	0.4	0	0
	0.7	0	0
	1.0	54.1	1.5
1.0	0.1	0	0
	0.4	0	0
	0.7	0	0
	1.0	0	0

Table 6: *MTA* occurrences at different ART_a and ART_b vigilance levels ($\rho_a, \rho_b \in \{0.1, 0.4, 0.7, 1\}$). The choice parameter (β) was fixed at 0.001 here. All other settings are the same as in table 5.

these anomalous situations do exist, and not to derive empirical formulae on just how often they occur.)

We also carried out a separate set of experiments where the ART_a and ART_b vigilance levels were varied. The results for various combinations of ρ_a and ρ_b can be seen in table 6.

First, it supports theorem 3 since no *MTA* occurred when ρ_b was 1. Also, an interesting pattern can be observed in the table: the *MTA* will not occur if $\rho_a \leq \rho_b$. This is because in the ARTMAP network, the ART_b module will always "force" its categorisation onto ART_a through the internal feedback mechanism. In these "auto-associative" experiments, the ART modules were presented with identical input patterns. As a result, they developed identical categories for $\rho_a \leq \rho_b$. Since no recoding of an ART_b category could occur without recoding its corresponding ART_a category too, the *MTA* did not arise in these cases. For the remaining combinations (i.e. $\rho_a > \rho_b$), we can see an increasing effect of ρ_a on the *MTA* frequencies. This is simply because as ρ_a increases, more and more inputs will be encoded perfectly by ART_a category nodes, which is a necessary condition for *MTA* (see section 3).

Finally, if the timing of the input and target presentations were according to theorem 1 (i.e. ART_a input presented before ART_b target), no *MTA* occurrences were found in any of the training sessions we carried out. This supports the claim of theorem 1, and further underlies the importance of the timing according to which inputs are presented to the ARTMAP network.

6 Discussion

In the previous sections, we proved the existence of the match tracking anomaly in the ARTMAP network.

The question now arises whether the match tracking anomaly can be eliminated even if the timing of input presentations is "undesired". We note first that in pattern recognition problems, this can be done simply by setting $\rho_b = 1$ (see theorem 3 in section 4.3). In general, however, the ARTMAP network needs some protection mechanism against *MTA*.

In the experiments, for example, where the ARTMAP network was used to develop two-level class hierarchies (Bartfai, 1994), and the target was presented before the input in each trial (i.e. the "undesired" timing), we modified the standard ARTMAP learning method as follows:

> Whenever the *MTA* condition arose, the network found an uncommitted ART_a node and learned the input and its association with the current target (in the original algorithm no learning occurred in these cases). It also "undid" the existing (wrong) association by freeing up the corresponding ART_a node (i.e. making it uncommitted again) and resetting its $F_2^a \rightarrow F^{ab}$ link. (This way, we ensured all ART_a category prototypes remained unique, and every input could access its category directly after self-stabilisation, both of which are important features of ART networks.)
> In *MTA*-free learning trials, the standard ARTMAP learning method was applied.

This modification was an adequate solution to eliminate the *MTA* condition since the training set was not contradictory here. In general, however, the occurrence of *MTA* may also be the result of a contradictory training set, in which case, learning the new association and forgetting the old one "blindly" may not be the best strategy. Therefore, in a real-world, autonomous learning environment, the network should be equipped with additional mechanisms that ensure correct operation. In particular, we recommend that

1. the choice parameter be chosen to be sufficiently small to avoid the "β-dependent" *MTA* condition (experimental results suggest that the limit may be lower than what was derived in section 4.2.2),

2. the network be extended with an "internal self checking" process that is activated whenever the *MTA* occurs.
   This process should first disable further learning temporarily, and then let the ART_a module make an internal prediction (while keeping the target input registered in the F_0^b layer) to get the correct timing, and check for the *MTA* condition again. If *MTA* persists, it will be an indication of a contradicting input/target pair, in which case the network should not learn on the current inputs. If *MTA* disappears, which indicates that the target input was presented first and that was the cause of *MTA*, the network should learn the current input/target pair with the correct ("ART_a first, then ART_b") timing.
   It could be the subject of further research to validate the correctness if this procedure, and to see if it can be implemented as an extension to the current match tracking process.

We also note that although the "anomalous situation" that can arise in the ARTMAP network was reported *in relation to* the match tracking process (Carpenter et al., 1991a), the problem itself lies *not* with the match tracking process *itself.* Whatever method is used to correct a wrong prediction, if the ART_a category *matches the input perfectly*, there is no way the network can find another category that will capture the same input (thereby produce the correct output) on future trials. In fact, we could even replace the match tracking process with some other method that can correct a wrong prediction of the network, for instance the one that was used in (Tan, 1992), and would still face the same problem. So regardless of the mechanism used to correct a wrong prediction, it is important that the network be able to "prime" itself with its "expectation" before the target is presented, and subsequently learned.

7 Conclusion

This paper analysed the Match Training Anomaly of the ARTMAP network. We showed that it can occur even when the inputs from a non-contradictory training set are presented to the network in complement coding. We conclude that the timing according to which the input and target patterns are presented to the network in each learning trial is critical if the ART_b vigilance level is less than 1.

> *To avoid the match tracking anomaly when input patterns are complement coded, the ART_a input has to be presented first, letting the network "prime" the F_1^b layer by its own "expectation" (through previously learned $F_2^a \rightarrow F_2^b$ associations), before the target input is presented to the ART_b module.*

If we cannot guarantee this timing, as in most real-world, autonomous learning tasks, we have to equip the network with additional mechanisms, like the one recommended in the previous section, which ensure its correct operation.

We expect the *MTA* condition will also be present in other ARTMAP architectures, like the Fuzzy ARTMAP network (Carpenter et al., 1992), for example. It would be interesting to see how the theorems presented here should change so they incorporate continuous inputs as well as slow learning mode. This again can be the subject of future research.

Acknowledgement

This work was in part supported by Victoria University of Wellington (Research Grant No.RGNT/594/371).

I am grateful to Peter Andreae (Department of Computer Science, Victoria University of Wellington, New Zealand) for his critical reading of the drafts, valuable suggestions for improving the style and presentation of this article, and even providing further insights into the similarities and differences of the two *MTA* conditions discussed here.

I wish to thank the reviewers for their constructive comments, too.

References

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

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

Carpenter, G. and Grossberg, S. (1987b). ART2: Self-Organizing of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics*, 26(23):4919–4930.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J., and Rosen, D. (1992). Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on Neural Networks*, 3(5):698–713.

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

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

Miyata, Y. (1991). *PlaNet, A Tool for Constructing, Running, and Looking into a PDP Network*.

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

Tan, A.-H. (1992). Adaptive Resonance Associative Map: A hierarchical ART system for fast stable associative learning. In *Proceedings of IJCNN*, volume 1, pages 860–865.