

A Neuro-Fuzzy System that Uses Distributed Learning for Compact Rule Set Generation

E. Parrado Hernández*, E. Gómez Sánchez*,
Y.A. Dimitriadis*, J. López Coronado**

* Department of Signal Theory, Communications and Telematics Engineering,
School of Telecommunications Engineering, University of Valladolid,
Camino del Cementerio S/N, 47011 Valladolid, Spain
E-mail: emipar@nava.tel.uva.es , edugom@tel.uva.es, yannis@tel.uva.es

** Department of Systems Engineering, Electronics and Industrial Electricity,
School of Industrial Engineering, University of Cartagena,
Paseo Alfonso XIII 48, 30203 Cartagena, Murcia, Spain
E-mail: coronado@plc.um.es

ABSTRACT

ARTMAP based architectures have several desirable properties that make them very suitable for pattern classification problems. However, they suffer from category proliferation. Distributed coding has been proposed as a solution for memory compression. dARTMAP neural network has been introduced as a modification of Fuzzy ARTMAP that, due to distributed learning, achieves code compression while fast stable learning is retained. A critical analysis of dARTMAP architecture and performance in pattern recognition problems is presented here, concluding that distributed learning excels the original Fuzzy ARTMAP only under certain geometrical configurations of the output classes, or in the presence of noise in the training set. A new architecture called dFasArt is presented here, introducing distributed learning into FasArt neuro-fuzzy system, which is more suitable for identification tasks, showing that the advantages of distributed code can be extended to other neural architectures. Experimental results show dFasArt performs similarly to dARTMAP in classification tasks, while being less sensitive to pattern presentation order.

1. INTRODUCTION

Since Adaptive Resonance Theory (ART) was first introduced by Grossberg [12], several neural architectures have been proposed within the ART family featuring fast on-line, stable learning and also allowing incremental learning (on-line adaptation) [5]. These architectures are specially designed for pattern recognition and multidimensional mapping tasks [5]. Furthermore, all ART networks share a set of basic properties that makes them suitable for non-stationary environments and real time applications. These properties include **fast learning** for exceptional cases, which reduces processing time; **dynamic neuron commitment**, which is carried out without network disruption; and **few training epochs** to reach acceptable levels of predictive accuracy.

Within ART family of neural networks, ARTMAP is an architecture that performs supervised multidimensional mapping in response to input/output vector pairs presented in arbitrary order [5]. ARTMAP consists of two self-organizing ART modules linked by a layer of nodes called inter-ART map field. The two ART modules cluster input and output spaces into categories, while the map field forms predictive associations between these categories.

Fuzzy Logic was formally introduced in Zadeh seminal paper [17], and has since then been used in many classification and mapping tasks [16] because it provides a knowledge representation close to linguistic description, thus allowing to express knowledge in the form of understandable IF-THEN rules. However, when expert knowledge cannot be easily transformed into rules, it becomes necessary to develop mechanisms in order to generate these rules from available data. In this sense, neural networks provide a means to automatic construction of the fuzzy rule set due to their self-organizing properties. Several neuro-fuzzy systems have been proposed in the literature, and successfully applied to different engineering problems [11] [14] [15].

Fuzzy ARTMAP [6] is a modification of ARTMAP architecture that introduces some basic principles of fuzzy logic. However, the main motivation of this change was to allow ARTMAP architecture to work with analog input patterns. In this sense, although Fuzzy ARTMAP could be used for the automatic construction of neuro-fuzzy systems by transforming the relations between input and output patterns stored in the inter-ART map field into IF-THEN rules, it is difficult to interpret these rules in fuzzy terms.

To overcome this ambiguity, FasArt neuro-fuzzy system was proposed [2], introducing fuzzy logic in a formal way into Fuzzy ARTMAP architecture, by establishing a duality between categories and fuzzy sets. This is achieved by setting the activation function of a given neuron dual to the membership function of an associated fuzzy set. This feature, in addition to the fact that it performs defuzzification in the output, FasArt has been successfully tested on identification and control tasks [1].

However, all ART based systems present a category proliferation problem, especially in noisy environments, due to fast learning and Winner-Take-All (WTA) coding. Category proliferation becomes a major problem when the aim of the neural network is to construct a set of rules for a fuzzy system, since it involves massive rule generation. This will result into an increase in the processing time of the fuzzy system without noticeable improvement of its predictive accuracy, since most extra rules are redundant. Moreover, such a complex rule set may turn out to be impractical in applications where a human operator should supervise the actions of the fuzzy system. Postprocessing methods for rule merging and reduction of redundancy have been proposed in the literature, although they lose the on-line feature [10]. In comparison to ART networks, other architectures

such as backpropagation trained perceptrons [13] employ fewer neurons, resulting in more compact code. Nevertheless, training these architectures is very slow and a great number of training epochs is necessary. Furthermore, they are not compliant to the stability-plasticity dilemma [12], and thus presentation of new training patterns may cause catastrophic forgetting of previous knowledge. Moreover, information present in the perceptron weights can hardly be transformed into fuzzy rules. These features make backpropagation networks unwieldy for on-line generation of fuzzy systems.

Recently, distributed coding has been proposed as a means to avoid massive commitment of neurons in ART architectures [3]. Prior to this work, several ARTMAP based neural networks such as ART-EMAP [9], ARTMAP-IC [7] and FasArt were proposed that take advantage of distributed code in test mode, but retain WTA code during learning. In [8], distributed ARTMAP (dARTMAP) is introduced as a Fuzzy ARTMAP based architecture that includes distributed code both for learning and test stages in pattern recognition problems. The new architecture inherits fast stable learning from Fuzzy ARTMAP and achieves code compression without significant reduction of predictive accuracy.

This paper makes a critical analysis of dARTMAP in order to extract qualitative conclusions about the advantages and limitations of distributed learning in ARTMAP based neural networks. A distributed version of FasArt is proposed to study the usability of the innovations introduced in dARTMAP algorithm into other ARTMAP based architectures.

The rest of this paper is organized as follows. Section 2 starts with a review of the innovations introduced in dARTMAP architecture. Afterwards, dARTMAP and Fuzzy ARTMAP performances are compared in several pattern recognition toy problems in order to determine the kind of problems in which distributed code outperforms WTA. These benchmarks include diverse geometric outlines and presence of noise in the training set. Section 3 is devoted to study the adaptability of distributed code features to other ARTMAP based architectures, in particular to FasArt. The section presents a brief review of FasArt architecture and introduces dFasArt, a distributed version of FasArt suited for pattern recognition problems. FasArt and dFasArt performances are compared in order to evaluate if distributed code advantages are extendable to other ARTMAP based neural networks. Finally, conclusions and future work are summarized in section 4.

Due to space constraints, this paper is not self contained, and therefore the reader is supposed to be familiar with Fuzzy ARTMAP architecture.

2. CRITICAL REVIEW OF dARTMAP

The dARTMAP architecture [8] has been derived from Fuzzy ARTMAP to perform distributed learning and testing, instead of the original WTA processes implemented in Fuzzy ARTMAP. In Fuzzy ARTMAP, when a pattern is presented to a Fuzzy ART module, competitive activation takes place among its nodes. At the end of the competition only one node will remain active, and then resonance is said to occur. This winner node will code the pattern solely. In a distributed ART, after the competition several nodes remain activated, and then they will code the pattern in proportion to their relative activation.

However, distributed learning causes catastrophic forgetting if it is implemented in the original Fuzzy ARTMAP architecture. To introduce distributed learning while retaining stable fast learning, some innovations have been introduced in Fuzzy

ARTMAP architecture to result in dARTMAP. These innovations include dynamic weights, a new content-addressable memory rule, instance counting, credit assignment and distributed learning laws. The following subsection is devoted to briefly explain these new features.

2.1 Analysis of the innovations introduced in dARTMAP

First of all, it must be clarified that dARTMAP is not a fully distributed system, but a hybrid distributed-WTA. When a pattern is presented, dARTMAP attempts to code it in distributed mode. If this code produces a correct prediction of the pattern, distributed learning proceeds. However, if the prediction is incorrect, the network switches to WTA mode and its operation is exactly that of Fuzzy ARTMAP. This switching to WTA is due to the lack of an efficient distributed inhibition method, i. e. a mechanism that penalizes the nodes responsible of the incorrect prediction without disrupting the stability of the learning process.

Dynamic weights: Fuzzy ARTMAP Long Term Memory (LTM) weights are replaced by dynamic weights. Dynamic weights are a function of both LTM, represented by an activation threshold and Short Term Memory (STM), represented by the node current activation.

Geometrically, each node in Fuzzy ARTMAP has an associated hyperbox whose vertices are given by the weights. In dARTMAP, hyperboxes have a LTM part given by the previously learnt patterns, and a STM part given by the current pattern and the activation of other nodes present in the competition. The LTM part of the hyperbox equals Fuzzy ARTMAP hyperbox, and is only modified when resonance occurs, while the STM part is an enlargement of the LTM hyperbox that is calculated for each particular input pattern.

New content-addressable memory rule: Fuzzy ARTMAP Content-Addressable Memory (CAM) rule features WTA coding, and therefore a new CAM rule is necessary in order to provide distributed coding. In dARTMAP, node activation by patterns is performed in two stages. First, every node is activated like in Fuzzy ARTMAP, using only the LTM part of the hyperbox. Then, the CAM rule carries out a competition among the nodes. As a result of this competition, each node steady state activation will be a fraction of its WTA activation. Therefore, after the competition several nodes remain activated, and they will code the pattern in a distributed way.

Instance counting: Every node keeps count of the patterns that has coded. Steady state node activation is weighted by this instance counting. This procedure is necessary because of the method used in dARTMAP to make distributed prediction for the output class. To find out the prediction, dARTMAP adds up the activation of all the nodes associated to each output class and predicts the one that has reached a higher sum value. Instance counting is introduced due to the fact that, if each node means one addend for the prediction sum, an output class represented by few nodes with each node coding many instances would be easily overcome by another output class represented by many nodes with each node coding few instances. Therefore, instance counting helps balance this effect.

Credit assignment: After a correct output class prediction, and if dARTMAP is in distributed mode, credit assignment takes place before the actual learning of the weights. Due to distributed activation, some minority nodes may remain activated though they lead to wrong output classes. Credit assignment inhibits these nodes ensuring that only those that have contributed to the correct output class will modify their weights to learn the new pattern.

Distributed learning laws: Distributed learning is made according to distributed instar and outstar learning laws [3] [4]. These laws allow changes in the LTM part of dynamic weights only when the steady state size of the coding hyperbox does not already include the new pattern. Unlike Fuzzy ARTMAP, dARTMAP weights are only modified when it is absolutely necessary for the correct coding of the pattern. Distributed learning laws endow dARTMAP with stability and avoid catastrophic forgetting.

2.2 Test of dARTMAP performance

Fuzzy ARTMAP and dARTMAP have been programmed on a sequential machine in order to compare their relative performances. These architectures have been tested in six pattern recognition benchmarks and results are discussed here focusing on the number of nodes recruited by each architecture and their accuracy (correct classification rate), and the standard deviation of both measurements. To extract conclusions about the influence of geometry, order of pattern presentation and presence of noise, several parameters were considered to design the benchmarks:

- Number of output classes.
- Probability of occurrence of each output class.
- Geometric *shape* of the output classes.
- Convexity of the geometric region associated to each output class. A region is *convex* if every two points within can be linked with a straight segment that does not run through another different class region.

The proposed tasks are described bellow, and schematic designs are shown in figure 1.

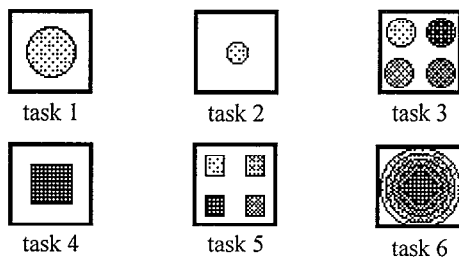


Figure 1 Schematic designs of the benchmarks. Different colors mean different output classes.

Task 1 consists in separating the points in the unit square that lie within a circle placed in the center of the square. The area of each class is 1/2. This task known as *circle in the square* is used in [8]. **Task 2** is the same as task 1 but with the circle being of area 1/3.

Task 3 consists in classifying the points in a unit square that lie within four different circles. Each of them is centered in one of the four quadrants of the square, and has an area of 1/5. **Tasks 4 and 5** are similar to tasks 1 and 3, but with squares instead of circles inside the unit square.

Task 6 consists in classifying the points in a unit square that lie within any of five concentric rings placed in the center of the square. All rings have the same area.

Results of the performance of Fuzzy ARTMAP and dARTMAP on these benchmarks are shown in table 1. These results average

100 simulations with 2000 patterns in the training set and 5000 patterns in the test set. To study the influence of the order of pattern presentation, the different training sets were formed by randomly reordering the same 2000 patterns. All training and test patterns are points generated randomly in the unit square.

Table 1 Performance results of Fuzzy ARTMAP and dARTMAP in the six benchmarks.

	Fuzzy ARTMAP		dARTMAP	
	Nodes	Acc(%)	Nodes	Acc(%)
T-1	21.38±3.71	92.43± 2.09	13.50±7.00	88.39±5.11
T-2	14.30±2.68	93.28± 3.75	9.20± 5.56	80.27±9.38
T-3	49.46±4.63	88.55± 0.98	35.20±9.01	87.18±1.98
T-4	6.98± 3.11	99.64± 0.30	9.22± 6.31	96.27±3.58
T-5	32.64±7.24	96.13± 1.34	31.20±10.52	95.57±1.28
T-6	116.90±8.25	83.97± 1.32	119.30±13.4	80.30±10.32

Influence of geometry: Results in table 1 can be useful to conclude in which cases the innovations introduced in section 2.1 will allow dARTMAP outperform Fuzzy ARTMAP. This improvement is dependent on several conditions discussed below.

a) *Probability of each class:* Tasks 1 and 2 share the same geometry but in task 2 the class *inside the circle* is less probable than the class *outside*, since it is only populated by 1/3 of the patterns in the unit square. While dARTMAP achieves code compression in both tasks, in task 2 correct classification rate falls 13% with respect to Fuzzy ARTMAP. The nodes associated to the more probable classes will code more patterns in the training set than those associated to less probable classes. Thus, due to instance counting, distributed prediction is biased towards more probable classes. During supervised training, this bias is controlled by switching to WTA after mismatches. However, during test, this bias increases prediction error rate in less probable classes, and thus dARTMAP accuracy decreases.

b) *Geometric shape of the output class:* Benchmarks 4 and 5 are similar to 1 and 3 but with different geometric shapes. It can be seen that dARTMAP does not achieve code compression in the tasks with rectangular shapes. Fuzzy ARTMAP two-dimensional nodes are geometrically equivalent to rectangles that learn a new pattern by enlarging their sides to include it. Thus, Fuzzy ARTMAP coding hyperboxes fit perfectly to rectangular output classes, and therefore dARTMAP coding hyperboxes can not reach a better code compression, while distributed code may decrease accuracy.

However, in non-rectangular shape cases like those proposed in tasks 1 and 3, Fuzzy ARTMAP must cover all the area inside the circles with rectangles associated to the *inside* classes. Along the borders, these rectangles will also cover patterns *outside* the circles, which will have to be covered in turn with smaller rectangles leading to the correct output. Distributed coding boxes do not need to cover all their coded patterns due to distributed learning laws. Therefore, those extra nodes associated to the smaller rectangles near the borders are not recruited. It can be concluded that distributed coding is not an advantage when output classes have rectangular shapes, but otherwise it reduces the number of nodes placed in the borders between classes, thus reducing the total number of nodes.

c) *Convexity of the regions:* Tasks 3 and 6 consist of many output classes with non-rectangular shapes. In task 3 these classes are convex, but not in task 6. Results show that dARTMAP code reduction is achieved only if output classes are convex regions. Due to distributed CAM rule, several nodes remain activated after the competition. In the case of convex regions, like circles in task 3, the most activated nodes

correspond to the same output class *inside the circle*, and therefore distributed prediction will be correct. However, when a pattern lies inside a ring in task 6, the most activated node will correctly predict *inside the ring*, but the following nodes in order of activation will predict the adjacent rings, since they are nearer to the pattern than other nodes of its ring at the opposite side of the center. This reason makes the network behave in a WTA-like mode and cancels distributed code advantage.

d) *Number of output classes*: Benchmarks results show that the number of output classes has no influence on distributed code node number reduction, since in both tasks 1 and 3 code compression is achieved in spite of their different number of classes. Furthermore, in both tasks 4 and 6, which also have a different number of classes, there is not a code reduction.

It can be concluded of these experiments that, in pattern recognition problems, distributed code reduces the number of nodes recruited by a Fuzzy ARTMAP network when classes are equally probable, they have associated geometric regions that are convex and with non-rectangular shapes.

Influence of pattern presentation order: Standard deviation values for the number of nodes and achieved accuracy through all six tasks are also shown in table 1. Higher values for dARTMAP show that this architecture is more sensitive to variations in the order of pattern presentation.

Performance in noisy environments: Tasks 3 and 5 were also carried out introducing **Gaussian noise** with 0 mean and 0.05 standard deviation in the training set. As seen above, task 3 poses a problem more suitable for distributed code, while task 5 is more suitable for the original Fuzzy ARTMAP.

Table 2 Performance results in noisy environments

	Fuzzy ARTMAP		dARTMAP	
	Nodes	Accuracy	Nodes	Accuracy
T-3	130.96	76.77%	88.19	81.34%
T-5	112	69.36%	63.31	70.49%

Results in table 2 show that dARTMAP outperforms Fuzzy ARTMAP in both code compression and accuracy, even in task 5. In this task noise has blurred the shape of the rectangles, canceling Fuzzy ARTMAP geometric advantage. Under noisy conditions, dARTMAP is more reliable because its prediction is based on the average of several nodes, while Fuzzy ARTMAP only takes into account one node.

3. ADAPTABILITY OF DISTRIBUTED CODE TO FasArt

FasArt [2] was proposed as a modification of ARTMAP that introduced formally fuzzy logic, and therefore FasArt is especially suitable for the implementation of fuzzy controllers. Thus, code compression becomes a crucial feature in FasArt design. In section 2.2 it has been shown that distributed coding reduces the number of nodes committed by Fuzzy ARTMAP in certain pattern recognition problems. This section is devoted to study the adaptation of the new features found in dARTMAP to FasArt, and to check if distributed code advantages will reproduce in a FasArt based architecture.

3.1 Brief review of FasArt

FasArt combines the basic features of Fuzzy ARTMAP with fuzzy logic theory introduced in a formal way [2]. FasArt establishes a duality between categories and fuzzy sets, by associating the activation function of each neuron to a fuzzy membership function. Due to this duality, the universal approximation principle obtained for fuzzy systems [16] can also be applied to FasArt, as it has been successfully tested in several

identification problems [1]. Furthermore, the fuzzy feature present in FasArt allows representing the knowledge learnt from training patterns in the form of understandable fuzzy IF-THEN rules. However, its ARTMAP nature and WTA coding cause category proliferation, resulting in redundant fuzzy rules generation.

As previously mentioned, FasArt replaces Fuzzy ARTMAP activation rule with triangular fuzzy membership functions. This design choice allows calculating a confidence value for FasArt predictions. While in Fuzzy ARTMAP all patterns coded by one node give the same activation value, in FasArt patterns coded by one node produce different activation depending on the distance to its membership function center.

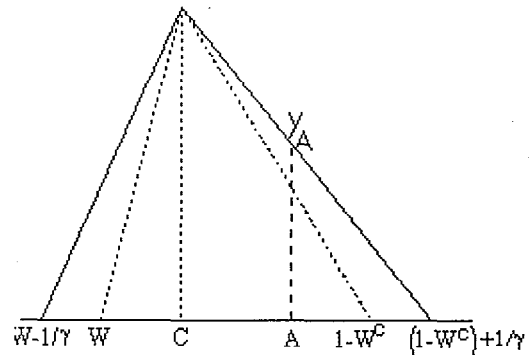


Figure 2 FasArt fuzzy set and membership function for a one-dimensional case, where y_a is the activation for the pattern A .

For the construction of such triangular membership functions, two new features are introduced in FasArt architecture: a weights vector C , that represents the center of the fuzzy membership function; and a user defined parameter γ , that determines what region of the input space is allowed to be learn by a node in the current pattern presentation. The maximum size of the fuzzy support is also restricted by a vigilance parameter, ρ , that has the same function as in Fuzzy ARTMAP algorithm. Weights in C are modified when resonance occurs according to a learning law similar to that applied in Fuzzy ARTMAP to weights W . Therefore, after training C is the prototype of the patterns coded by that node. In figure 2 a FasArt fuzzy membership function is displayed for one dimensional case, where C and γ are the newly introduced features, while weights W and W^C play the same role as in Fuzzy ARTMAP.

3.2 Introduction of distributed learning in FasArt

Although FasArt features distributed test, it performs WTA coding in training stage. To extend the advantages of distributed learning to FasArt, a new architecture called dFasArt is proposed here. Although the final goal of dFasArt will be function identification, as a first step this paper is devoted to check out if the advantages that dARTMAP presents over Fuzzy ARTMAP in pattern recognition problems stand for dFasArt over FasArt. In this sense, dFasArt incorporates the new features found in dARTMAP that have been properly adapted for FasArt architecture as explained below.

In analogy to dARTMAP **coding hyperboxes**, dFasArt fuzzy supports consist of a LTM part that is equivalent to FasArt support, and a STM part that depends on the current pattern presented and the competitive activation of other nodes.

In addition, a new **CAM rule** has been developed for dFasArt in order to combine distributed activation with triangular fuzzy

membership functions. In figure 3 the construction of dFasArt fuzzy support is shown, where weights W and C determine the LTM part of the support. The STM part is added at the side of the fuzzy support nearest to the input pattern. When a pattern is presented, all the nodes lengthen their supports towards it until their sides reach their maximum length, which is taken as 1 since the input space is the unit square. At this stage, membership function is evaluated on this support. If the pattern is included in the LTM part of the support (like pattern B in figure 3), the current activation of the node becomes its steady state activation. Otherwise (like pattern A), the activation is normalized with the

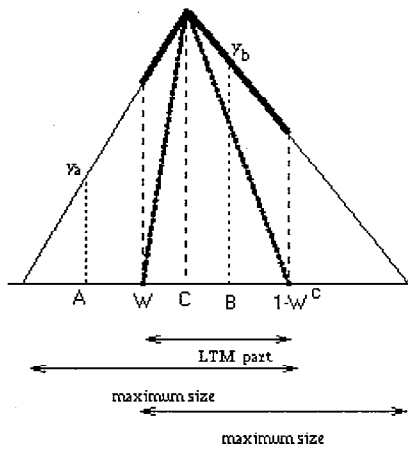


Figure 3 Construction of dFasArt fuzzy support and membership function.

activation of all other nodes. The thick line in figure 3 represents those activation values that will not be normalized because they correspond to patterns inside the LTM support.

Instance counting before output class prediction, **switching to WTA** after incorrect prediction and **credit assignment** before the resonance state are implemented on dFasArt according to the same design criteria stated for dARTMAP.

Distributed learning laws developed for dARTMAP have failed to fit dFasArt algorithm because special treatment for weight vector C is required. Instead, FasArt learning laws are implemented using as learning rate for each node its steady state activation. This implies that patterns close to the support will be learned faster than those further apart.

3.3 Test of dFasArt performance

As stated before, FasArt is suited to function identification problems. Triangular membership function performs better than Fuzzy ARTMAP activation function when inputs are samples of a continuous function rather than patterns to be classified in discrete output classes. Thus, the aim of this paper is not to propose dFasArt as a substitute for dARTMAP, but to prove that FasArt can take advantage of distributed coding in pattern recognition problems in order to eventually extend these results to function identification problems.

As done previously with dARTMAP, the influence of geometry, order of pattern presentation and presence of noise are studied.

Table 3 Performance of FasArt vs. dFasArt

	FasArt		dFasArt	
	Nodes	Accuracy	Nodes	Accuracy
T-1	62.04±9.77	93.34± 3.07	14.22±2.84	91.33±1.46
T-3	92.16±35.39	88.94± 2.24	43.12± 6.72	85.10±1.10

Influence of geometry: In section 2 it was revealed that distributed code only achieves code compression in classification problems with *equally probable classes* that are associated to *convex geometric regions* and with *non-rectangular shapes*. Tasks 1 and 3 of figure 1 are examples of such problems, and therefore FasArt and dFasArt performances have been tested on these two benchmarks to check if distributed code was also advantageous to FasArt based architectures. Results in table 3 show that dFasArt achieves code compression in those cases where dARTMAP did. Comparison of tables 1 and 3 show that dFasArt performance is comparable to those of Fuzzy ARTMAP and dARTMAP. Thus distributed coding turns out helpful to suit FasArt to pattern recognition tasks.

Influence of presentation order: dFasArt is less sensitive to presentation order than FasArt, since table 3 standard deviation values are higher for the latter. Comparison of tables 1 and 3 show that dFasArt is even less sensitive than dARTMAP to presentation order.

Performance in noisy environments: Tasks 3 and 5 were also carried out with FasArt and dFasArt including Gaussian noise as described before. Table 2 showed that distributed learning achieves higher code compression in the presence of noisy using dARTMAP. Table 2 shows that this property can also be extended to FasArt based architectures. Comparison of tables 2 and 4 show that dFasArt performance is again similar to those of Fuzzy ARTMAP and dARTMAP.

Table 4 Performance of FasArt and dFasArt results in noisy environments

	FasArt		dFasArt	
	Nodes	Accuracy	Nodes	Accuracy
T-3	380.8	80.6%	100.35	82.10%
T-5	455	77.8%	117.5	81.81%

According to tables 2 and 4, it can be concluded that distributed code results in more reliable systems in noisy environments, due to the fact that distributed prediction is made on the average of several nodes.

4. CONCLUSIONS

Distributed learning has been recently proposed as a means to solve category proliferation problem in Fuzzy ARTMAP neural network and its derived architectures. In this sense, dARTMAP was developed as a particular architecture that introduced several new features in the original Fuzzy ARTMAP in order to implement distributed learning for pattern recognition problems.

This paper has analyzed those new features and their influence in the neural network performance. These innovations include new **dynamic coding hyperboxes** determined by dynamic weights for distributed code, and a new **CAM rule** to allow distributed activation of the network nodes. In addition, **switching to WTA** after mismatched predictions and **distributed learning laws** are introduced to enable fast stable learning. Finally, **instance counting** balances the effect of classes represented by many nodes in distributed prediction, and **credit assignment** avoids learning patterns by nodes that have not contributed to a correct distributed prediction.

Fuzzy ARTMAP and dARTMAP performances were compared in several pattern recognition benchmarks in order to extract qualitative conclusions about the advantages and limitations of distributed code over WTA in Fuzzy ARTMAP based architectures. The benchmarks were selected varying the number of output classes, the probability of occurrence of them, the

convexity of the geometric regions associated and the shapes of these regions.

The geometry of the problem has revealed an item of major influence on the performance of distributed learning vs. WTA. dARTMAP accuracy is significantly decreased when output classes are not **equally probable** due to instance counting mechanism. In addition, Fuzzy ARTMAP and dARTMAP performances are equivalent when coding not **convex** regions since, in such geometry, distributed CAM rule is forced to switch to WTA. Moreover, if output classes have **rectangular shapes**, dynamic coding hyperboxes can not outperform the original rectangular hyperboxes and distributed prediction reduces accuracy. However, the number of output classes has not been found to affect dARTMAP code compression. In summary, in Fuzzy ARTMAP based architectures distributed learning achieves code compression without affecting accuracy only in problems where output classes are equally probable, and associated to convex regions with non-rectangular shapes.

From the geometric constrains mentioned above, it can be deduced that dARTMAP is useless in one dimension classification problems, since decision borders between output classes are single points and thus Fuzzy ARTMAP coding hyperboxes will fit in better.

Performance in noisy environments has also been tested, showing that dARTMAP code compression is increased when training patterns include noise, even in geometries more suitable for Fuzzy ARTMAP. Moreover, dARTMAP turns out to be more reliable in noisy environments since its output class prediction is based on the average of several nodes, thus minimizing the influence of noise. The influence of order of pattern presentation has also been tested experimentally, showing that dARTMAP is more dependent on this order.

In this paper dFasArt has been introduced as a modification of FasArt neuro-fuzzy system that incorporates distributed learning for pattern recognition problems. The new features found in dARTMAP algorithm have been adapted to FasArt architecture. Therefore, a new **distributed membership function** has been defined, and FasArt **learning laws** have been modified to support distributed learning. dFasArt also incorporates **instance counting, credit assignment and switching to WTA** after prediction mismatch.

FasArt and dFasArt performances have been tested in those geometries suitable to distributed learning. Advantages for distributed learning vs. WTA have also been found in the above mentioned conditions. Experimental results show that dFasArt performance in such geometries is even comparable to those of dARTMAP and Fuzzy ARTMAP. dFasArt has also revealed to be less sensitive to variations in the order of pattern presentation. Furthermore, an increase in dFasArt performance in noisy environments has also been observed.

After successfully applying distributed learning to FasArt based architectures in order to achieve code compression without accuracy reduction, ongoing research focus on the adaptation of the new dFasArt architecture to function identification problems. New improved distributed learning laws seem likely to be developed for dFasArt in order to perform efficient distributed coding of non-convex regions. Furthermore, geometric constrains for distributed learning found in pattern recognition problems will help determine the type of functions where distributed learning will outperform WTA.

Acknowledgements

Authors are especially grateful to Dr. J.M. Cano Izquierdo for his valuable suggestions and support. We also would like to thank M. Araúzo Bravo, R. Sacristán Martínez and M. Martín-Merino Acera for their helpful comments. E. Gómez Sánchez was partially supported by ESPRIT 22416 "MONNET" with Cetedel.

5. REFERENCES

- [1] J.M. Cano, M.J. Araúzo, Y.A. Dimitriadis and J. López, "Non linear systems identification using fuzzy neural systems," *Proceedings of EUFIC'97 5th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany. Vol. 1, Sept. 1997, pp. 384-388.
- [2] J.M. Cano, Y.A. Dimitriadis, M. Araúzo, and J. López Coronado, "FasArt: A new neuro-fuzzy architecture for incremental learning in system identification," in *Proc. Of the IFAC World Congress, IFAC96, San Francisco*, Vol. F, Jun. 1996, pp. 133-138.
- [3] G.A. Carpenter, "A distributed outstar network for spatial pattern learning," *Neural Networks*, Vol. 7, No.1, Jan. 1994, pp. 159-168.
- [4] G.A. Carpenter, "Distributed learning, recognition, and prediction by ART and ARTMAP neural networks," *Neural Networks*, Vol. 10, No.8, Sept. 1997, pp. 1473-1494.
- [5] G.A. Carpenter and S. Grossberg, "Learning, categorization, rule formation, and prediction by fuzzy neural networks," in C.H. Chen, editor, *Fuzzy Logic and Neural Network Handbook*, New York, NY: McGraw-Hill, 1995.
- [6] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. on Neural Networks*, Vol. 3, No.5, Sep. 1992, pp. 698-713.
- [7] G.A. Carpenter and N. Markuzon, "ARTMAP-IC and medical diagnosis: instance counting and inconsistent cases," *Neural Networks*, Vol. 11, 1998.
- [8] G.A. Carpenter, B.L. Milenova, and B.W. Noeske, "Distributed ARTMAP: a neural network for fast distributed supervised learning," *Neural Networks*, Vol. 11, No. 5, Jul. 1998, pp. 793-813.
- [9] G.A. Carpenter and W.D. Ross, "ART-EMAP: a neural network architecture for object recognition by evidence accumulation," *IEEE Trans. on Neural Networks*, Vol. 6, 1992, pp. 805-818.
- [10] G.A. Carpenter and A.H. Tan "Rule extraction, Fuzzy ARTMAP and medical databases", *Proc. of the World Congress on Neural Networks*, Vol. 1, 1993, pp. 501-506
- [11] C. Lin and C. Lin, "An ART-based fuzzy adaptive learning control network," *IEEE Trans. On Fuzzy Systems*, Vol. 5, No.4, Nov 1997, pp. 477-496.
- [12] S. Grossberg, "Adaptive pattern classification and universal recoding. II: Feedback expectation, olfaction and illusions," *Biological Cybernetics*, Vol. 23, 1976, pp. 187-202.
- [13] J. McClelland and D. Rumelhart, "Explorations in parallel distributed processing", *MIT Press*, Cambridge, MA, 1986.
- [14] P. Simpson, "Fuzzy Min-Max neural networks, part 1: classification," *IEEE Trans. on Neural Networks*, Vol. 3, No.5, Sep. 1992, pp. 776-786.
- [15] P. Simpson, "Fuzzy Min-Max neural networks, part 2: clustering," *IEEE Trans. on Fuzzy Systems*, Vol. 1, No.1, Feb. 1993, pp. 32-45.
- [16] L. Wang, *Adaptive fuzzy systems and control*, New York, NY: Prentice Hall, 1994.
- [17] L. Zadeh, "Fuzzy sets," *Information and Control*, Vol. 8, 1965, pp. 338-353.