PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 1039–1057

Neural Networks

www.elsevier.com/locate/neunet

Study of distributed learning as a solution to category proliferation in Fuzzy ARTMAP based neural systems

Emilio Parrado-Hernández[a,*], Eduardo Gómez-Sánchez[b], Yannis A. Dimitriadis[b]

[a]*Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911 Leganés, Madrid, Spain*

[b]*Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Valladolid, Camino del Cementerio S/N, 47011 Valladolid, Spain*

Received 20 July 2000; accepted 22 November 2002

Abstract

An evaluation of distributed learning as a means to attenuate the category proliferation problem in Fuzzy ARTMAP based neural systems is carried out, from both qualitative and quantitative points of view. The study involves two original winner-take-all (WTA) architectures, Fuzzy ARTMAP and FasArt, and their distributed versions, dARTMAP and dFasArt.

A qualitative analysis of the distributed learning properties of dARTMAP is made, focusing on the new elements introduced to endow Fuzzy ARTMAP with distributed learning. In addition, a quantitative study on a selected set of classification problems points out that problems have to present certain features in their output classes in order to noticeably reduce the number of recruited categories and achieve an acceptable classification accuracy.

As part of this analysis, distributed learning was successfully adapted to a member of the Fuzzy ARTMAP family, FasArt, and similar procedures can be used to extend distributed learning capabilities to other Fuzzy ARTMAP based systems.

Keywords: Distributed learning; Fuzzy ARTMAP; dARTMAP; FasArt; Category proliferation; Neuro-fuzzy systems; Classification

1. Introduction

Adaptive resonance theory (ART) is derived from Grossberg and Carpenter's work in human learning processes (Grossberg, 1982b, 1988). Since then, several artificial neural network architectures based on ART postulates have been proposed. This family includes ART-1 (Carpenter & Grossberg, 1987, 1988), for unsupervised learning of binary input patterns; ARTMAP (Carpenter, Grossberg, & Reynolds, 1991a) for supervised learning of binary input patterns; and Fuzzy ART (Carpenter, Grossberg, & Rosen, 1991b) and Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992), which introduce some elements from fuzzy sets theory in order to deal with both analog and binary patterns for unsupervised and supervised learning, respectively.

All ART based neural networks share a set of properties that make them very suitable for applications requiring on-line performance. These properties include a solution of the *stability–plasticity* dilemma (Grossberg, 1982a), which allows incremental learning in time-varying environments; fast stable learning, multiple generalization scales and fast convergence with a relatively small number of training patterns.

As stated before, Fuzzy ARTMAP introduces some Fuzzy Logic terms enabling the knowledge acquired by the network to be expressed easily into fuzzy IF–THEN rules (Carpenter et al., 1992), which makes Fuzzy ARTMAP a powerful tool for constructing neuro-fuzzy systems. In the literature, many examples of these Fuzzy ARTMAP based neuro-fuzzy systems can be found, among them, FALCON (Lin & Lin, 1997), Fuzzy Min–Max (Simpson, 1992, 1993) and FasArt (Cano-Izquierdo, Dimitriadis, Araúzo-Bravo, & Araúzo-Bravo 1996; Cano-Izquierdo, Dimitriadis, Gómez-Sánchez, & López-Coronado, 2001) have been reported to perform successfully on pattern recognition and function approximation tasks.

However, Fuzzy ARTMAP based neural systems suffer from a category proliferation problem, affecting their

* Corresponding author. Tel.: +34-91-624-8759; fax: + 34-91-624-8749.

E-mail address: emipar@tsc.uc3m.es (E. Parrado-Hernández).

0893-6080/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0893-6080(03)00009-1

Nomenclature

$\vec{I} = (\vec{a}, \vec{a}^c)$	input vector, in complementary code
M	input vector dimension
N	number of used units in F_2 layer
$\vec{W}_j = \{w_{ji}\}$	weights in F_2 layer for unit j in Fuzzy ARTMAP, FasArt and dFasArt
$\vec{C}_j = \{c_{ji}\}$	new center weights in FasArt and dFasArt of unit j
τ_{ji}	top–down threshold between units j in F_2 layer and i in F_0 in dARTMAP
τ_{ij}	bottom–up threshold between units j in F_2 layer and i in F_0 in dARTMAP
T_j	activation of unit j for Fuzzy ARTMAP and dARTMAP, membership function for FasArt and dFasArt
μ_{ji}	activation contributed by variable i to unit j
α	choice parameter in Fuzzy ART
β	learning rate
β^c	learning rate for $\vec{C}_j$ in FasArt and dFasArt
ρ	vigilance parameter
γ	fuzzification rate in FasArt
λ	maximum size of each side of the fuzzy support in dFasArt
$\vec{y} = \{y_j\}$	F_2 layer output
$\vec{Y} = \{Y_j\}$	F_3 layer output
$\vec{c}_j$	instance counting F_2 units in dARTMAP and dFasArt
$\vec{\sigma} = \{\sigma_i\}$	prototype for the evaluation of matching criterion in dARTMAP
$\vec{G}_j = \{G_{ji}\}$	geometric center of the fuzzy support in dFasArt
$\vec{b}$	desired output
$\lvert\vec{x}\rvert = \sum_{i=1}^{M} x_i$	L^1 norm
$a \wedge b = \min\{a, b\}$	fuzzy AND operator
$[\phi]^+ = \max\{\phi, 0\}$	rectification operator

Superscripts and subscripts a or b refer to ARTa and ARTb, respectively

generalization capability (Carpenter, Milenova, & Noeske, 1998b). Moreover, when Fuzzy ARTMAP architecture is incorporated into a Fuzzy system, category proliferation may result in the generation of redundant rules, which leads to a large rule set[1] and an increase in processing time. Furthermore, if the rule set is to be manipulated by a human operator, the system complexity becomes a major problem. In addition, the presence of noise in the training set augments category proliferation since noise complicates relations among input and output data, and may introduce contradictory training patterns (Cano-Izquierdo, 1997; Marriott & Harrison, 1995).

Approaches to solving the category proliferation problem in ART neural systems can be divided into two main groups: those which seek an off-line solution and those which try to preserve the original on-line characteristic of ART systems. The former solutions are basically represented by postprocessing methods, like rule pruning (Carpenter, 1994), that take place after training has been completed. The latter methods imply the introduction of changes in the original Fuzzy ARTMAP architecture in order to avoid massive commitment of neurons during the training phase. These on-line methods must guarantee that the derived architecture retains Fuzzy ARTMAP stable learning. Within on-line solutions, distributed learning has been proposed recently by Carpenter (1997) and Carpenter et al. (1998) in dARTMAP (distributed ARTMAP) architecture.

According to ART principles, knowledge representation or coding by a neural network can be distributed among all the neurons. However, all practical implementations of ART architectures use competitive WTA (winner-take-all) learning, i.e. only the neuron that wins a competition for the coding of each input pattern actually modifies its weights in order to learn the pattern. In spite of this, several ARTMAP based neural architectures, such as ART-EMAP (Carpenter & Ross, 1995), ARTMAP-IC (Carpenter & Markuzon, 1998) and FasArt take advantage of other distributed features (different from learning), like defuzzification at test. Apart from the ART family, many other artificial neural networks (ANNs) include distributed learning, like backpropagation trained multi-layer perceptrons (MLPs) (Rosenblatt, 1958, 1962; Rumelhart, Hinton, & Williams, 1986), or decision based neural networks (DBNN's) (Kung & Taur, 1995), typically yielding less complex architectures, but without ARTs desirable properties of fast, stable, and incremental learning.

Therefore, dARTMAP intends to merge distributed MLP code compression and Fuzzy ARTMAP fast on-line learning (Carpenter et al., 1998). In Carpenter (1997) and Carpenter et al. (1998), dARTMAP is presented as an extension of Fuzzy ARTMAP, capable of fast, stable on-line distributed learning. Moreover, dARTMAP distributed learning laws do not cause catastrophic forgetting. In those papers, dARTMAP is tested against the original Fuzzy ARTMAP in the circle-in-the-square benchmark, showing that distributed learning reduces the number of categories from 16.7 to 11.7 with accuracy decreasing from 92 to 90.6% (Carpenter et al., 1998).

The work described in this paper aims to study systematically the introduction of distributed learning in Fuzzy ARTMAP based neural systems as a means to reduce category proliferation in these architectures.

[1] In Fuzzy ARTMAP based neuro-fuzzy systems each category implies one rule.

The methodology starts with a theoretic study of the innovations that dARTMAP introduces in the original Fuzzy ARTMAP. Afterward, the FasArt neuro-fuzzy system is adapted to use distributed learning so that the impact of distributed learning can be analyzed in two different Fuzzy ARTMAP based architectures (dARTMAP and the new distributed version of FasArt, called dFasArt) and more general conclusions can be extracted. A set of benchmark problems has been selected to test the dependence of distributed learning advantages over WTA on the characteristics of the problem. These benchmarks include both pattern recognition and function approximation tasks.

The outline of the paper is as follows. Section 2 includes a brief review of WTA Fuzzy ARTMAP and FasArt neural architectures, as they form the basis of the distributed systems studied in the next sections. Section 3 is devoted to a qualitative analysis of the dARTMAP architecture and introduces dFasArt. The new elements of both architectures are explained and the expectations of performance according to this analysis are given. Section 4 starts with a description of the benchmark set selection; afterward, performances of the four neural systems on those benchmarks are discussed. Finally, Section 5 reviews the most important conclusions extracted from the present work and identifies areas for future research.

2. Review of fuzzy ARTMAP and FasArt

A Fuzzy ARTMAP neural network consists of two Fuzzy ART modules linked by a layer of neurons called the *interART map* (see Fig. 1). Each Fuzzy ART module performs unsupervised clustering either in the input or the output space, and the map stores relationships among the clusters created by both Fuzzy ART modules.

2.1. Fuzzy ART

In Fig. 1, an outline of the Fuzzy ART neural architecture is shown (one Fuzzy ART module inside each dotted rectangle) as part of Fuzzy ARTMAP. Basically, Fuzzy ART consists of three neural layers: *preprocessing* F_0, *matching* F_1 and *competitive* F_2.

Every input vector component, a_i, must be normalized between 0 and 1. Layer F_0 is formed by $2M$ neurons, with M being the dimension of the input vectors, and provides the complement code of the input vectors according to the following expression:

$$I_i = \begin{cases} a_i & 1 \le i \le M \\ 1 - a_{i-M} & M+1 \le i \le 2M \end{cases} \quad (1)$$

Layer F_1 is also formed by $2M$ neurons and its function is to verify the match between input patterns and prototypes learned by the network. Finally, layer F_2 is a competitive layer. It works as a *content addressable memory* (Carpenter et al., 1998) where each neuron stores a prototype of a class of input vectors. F_2 is formed by a total number of N neurons which are recruited dynamically as they are needed to encode new classes of incoming vectors. Each layer is connected to the next through a set of *adaptive weighted paths*. These weights, W_{ij}, form the *long term memory* (LTM) element of the neural network and evolve during the training phase. Every weight is initialized to 1 at the beginning of the training and monotonically decreases as the training proceeds and patterns are learned. This monotonical decrease of weights guarantees the eventual stability of the network.

Unsupervised learning in Fuzzy ART is performed in the following way. Each input pattern, $\vec{a}$, is put into its *complement code*, $\vec{I}$, according to expression (1), and then it is transmitted through F_1 to layer F_2. Each neuron j in F_2 receives an *activation*, $T_j(\vec{I})$, that is a function of the input pattern and the LTM weights:

$$T_j = \frac{|\vec{I} \wedge \vec{W}_j|}{\alpha + |\vec{W}_j|} \qquad j = 1, \ldots, N \quad (2)$$

where $\vec{W}_j = [W_{j1}, W_{j2}, \ldots, W_{j2M}]$ are the weights associated with neuron j; $|\cdot|$ is the L^1 norm, $|\vec{x}| = \sum_{i=1}^{M} x_i$; $x \wedge y = \min\{x, y\}$ is the fuzzy AND operator, for vectors, $\vec{x} \wedge \vec{y} = \vec{v}$ with $v_i = x_i \wedge y_i$ (Zadeh, 1965) and $\alpha \in [0, \infty]$ is a choice parameter (typically $\alpha \approx 0^+$).

At this point, the neurons in F_2 hold a WTA *competition* to select which neuron, J, is going to learn the pattern:

$$T_J = \max_j \{T_j\} \qquad j = 1, \ldots, N \quad (3)$$

After the competition, only the output of the winning neuron remains set to 1 and descends through the top–down weighted paths so that the prototype of neuron J is presented in layer F_1. In F_1 the *matching* between the input pattern, $\vec{I}$, and the winner prototype, $\vec{W}_J$, is evaluated according to a criterion determined by a user defined parameter $\rho \in [0, 1]$. The criterion is applied as follows:

- If $(|\vec{I} \wedge \vec{W}_J|/|\vec{I}|) \ge \rho$, then the input is considered to belong to match prototype in J and pattern is learn by neuron J.
- If $(|\vec{I} \wedge \vec{W}_J|/|\vec{I}|) < \rho$, then the system is reset and neuron J is inhibited so that it no longer enters the competition for the current pattern. In addition to this, a *match tracking* mechanism raises the value of parameter ρ so that the next winner must be closer to the pattern. After this new competition, another winner is selected. Eventually, a new neuron in F_2 will be committed if none of the current neurons is found to match the pattern sufficiently. This can happen not necessarily after all existing neurons have been evaluated (Georgiopoulos, Fernlund, Bebis, & Heileman, 1996).

When a winner successfully passes the matching criterion, *learning* occurs. LTM weights are updated

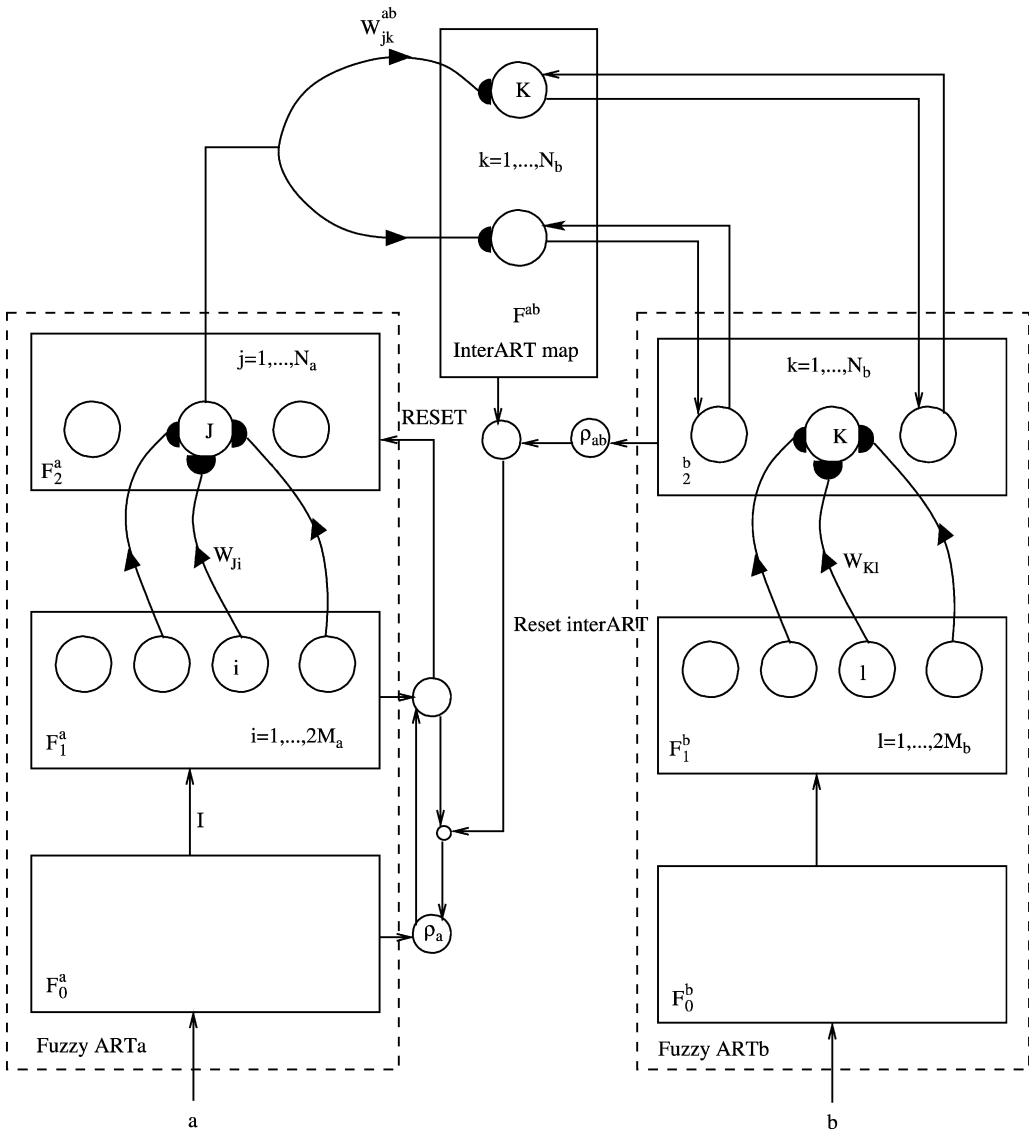

Fig. 1. Fuzzy ARTMAP structure. This neural network consists in two unsupervised Fuzzy ART modules (inside the dotted rectangles) that perform a clustering in both input and output spaces, and a neural layer called interART map that links the input categories to the output ones enabling supervised learning.

according to the following learning law:

$$\vec{W}_J^{\text{new}} = \beta(\vec{W}_J^{\text{old}} \wedge \vec{I}) + (1-\beta)\vec{W}_J^{\text{old}} \tag{4}$$

where $\beta \in [0,1]$ is the learning rate: $\beta \to 0^+$ implies slow learning, while $\beta = 1$ implies fast learning and each pattern is incorporated to the knowledge stored by the network in just one iteration. From this point on, fast learning is assumed throughout the rest of the paper.

2.2. Fuzzy ART geometry

With complement coding of patterns and the L^1 norm, each F_2 neuron can be represented geometrically as a *hyperbox* in $\Re^M$ covering all the patterns that it has already learned. The size of the hyperbox R_j associated with neuron j, is determined by weights $\vec{W}_j$ as showed in Fig. 2.

Competition in layer F_2 has also a geometric interpretation. Activation function, T_j, is a measure of the *distance* between the pattern $\vec{a}$ and R_j (Fig. 2). Therefore, the neuron with the box lying nearest to the pattern will receive the highest activation. Parameter α in Eq. (2) is used to break ties when several boxes include the pattern; in such case, the smaller the box is, the higher the activation received.

Finally, the learning process can be viewed as the expansion of the winner neuron box toward the pattern. If fast learning is applied, the box grows until it actually covers the pattern, while under slow learning the box just expands toward the pattern but without covering it.

2.3. Fuzzy ARTMAP

The Fuzzy ART architecture described in Section 2.2 is capable of unsupervised learning of either binary or analog

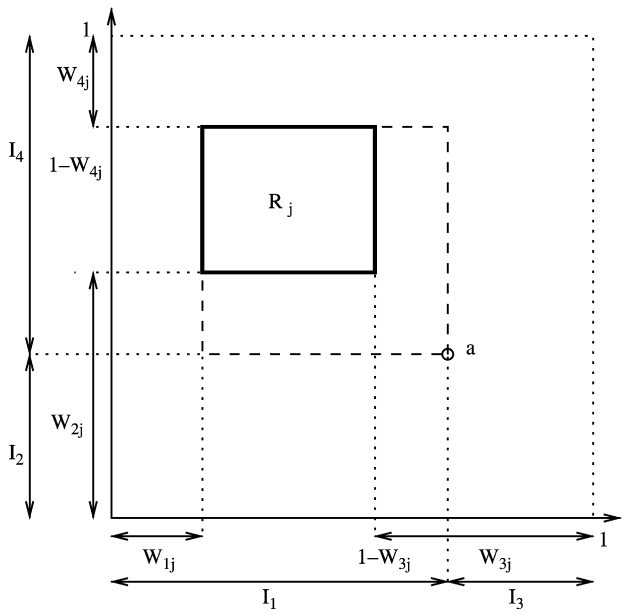

Fig. 2. Geometric interpretation of Fuzzy ART. Box R_j is associated with neuron j, while $\vec{I} = [I_1, I_2, I_3, I_4]$ is the complement code of input pattern $\vec{a}$. Size of box R_j is determined by weighs associated with neuron j, $\vec{W}_j = [W_{1j}, W_{2j}, W_{3j}, W_{4j}]$. In a generic M dimensions case, R_j size on dimension i is determined by W_{ij} and $W_{i+M,j}$.

input vectors. For *supervised learning*, two Fuzzy ART modules are connected through the interART map (Fig. 1). Fuzzy ARTa receives a vector, $\vec{a}$, from the input space, and Fuzzy ARTb receives the desired output, $\vec{b}$, when $\vec{a}$ is presented.

When a neuron in F_2^a (the F_2 layer in Fuzzy ART a) is recruited, it is linked through the map with the neuron in F_2^b that stores the prototype that matches the corresponding $\vec{b}$.

The operation of the map is as follows. After $\vec{a}$ and $\vec{b}$ are presented to the corresponding Fuzzy ART modules, a neuron J from F_2^a will win the competition for coding $\vec{a}$ and another one, K from F_2^b, will win the competition for $\vec{b}$. Then two cases may arise:

- If neuron J and neuron K are linked through the map, then there is an interART matching and learning proceeds independently in both Fuzzy ARTs.
- If neurons J and K are not linked, then *interART reset* is triggered: neuron J is inhibited and Fuzzy ARTa is reset so that a new category search takes place. This reset and search process continues until a neuron J' that produces an interART matching is found. The latter will always occur when a new F_2^a category is created.

One of the most valuable features of Fuzzy ARTMAP is that relations among neurons stored in the map can be expressed in terms of *IF–THEN* rules. The link between neurons J and K can be stated in this manner: IF $\vec{a}$ belongs to category J THEN the desired output belongs to category K in the output space.

2.4. FasArt

As stated Section 1, the study of the impact of distributed learning on Fuzzy ARTMAP based architectures is extended to the FasArt neuro-fuzzy system so that more general conclusions can be drawn. FasArt expands the scope of this study in two ways. From the neural network point of view, FasArt shares Fuzzy ARTMAP architecture (Fig. 1) and dynamics, like WTA activation and match tracking, but implements a different neural activation function. On the other hand, due to its fuzzy system nature, FasArt is more suitable for dealing with other engineering problems, such as function approximation (Cano-Izquierdo et al., 2001).

FasArt was proposed to overcome some Fuzzy ARTMAP defects from the point of view of Fuzzy Sets Theory. FasArt can also be interpreted as a neuro-fuzzy system (Cano-Izquierdo et al., 2001). FasArt incorporates fuzzy sets, categories and defuzzification into the Fuzzy ARTMAP architecture in a formal manner, and identifies the degree of confidence with which IF–THEN rules are triggered.

The duality between neural and fuzzy natures in FasArt is established in the following way. Each neuron in the FasArt F_2 layer is associated with a fuzzy set whose support is equivalent to the corresponding Fuzzy ARTMAP hyperbox. FasArt replaces the L^1 norm neural activation by *triangular fuzzy membership functions* (Fig. 3). To construct the membership function of Fig. 3, two new elements need to be introduced in the FasArt architecture: weight vectors $\vec{C}$ and a user defined parameter γ. Vector $\vec{C}_j$ stores the center of the fuzzy set associated with neuron j and follows the same dynamics as $\vec{W}_j$, while γ determines what region of the input space is allowed to be learned by a neuron in the current pattern presentation, since, as shown in Fig. 3, it determines the patterns with a non-zero membership function value. This way, activation function, T_j, is computed according to a triangular fuzzy membership function, given by:

$$T_j = \prod_{i=1}^{M} \mu_{ji}(I_i) \qquad j = 1, \ldots, N \tag{5}$$

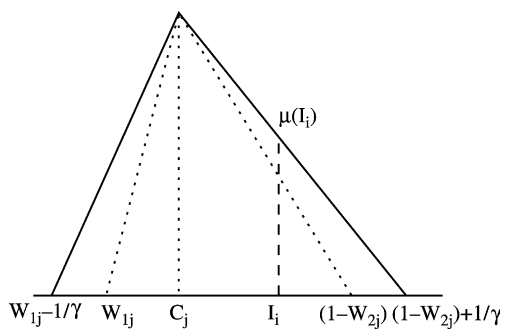

Fig. 3. FasArt triangular fuzzy membership function and fuzzy support in an 1D example. According to expression (6), μ is the activation reached by the I_i component of the current pattern. Fuzzy support is determined by neuron j weights $\vec{W}_j$ and $\vec{C}_j$, and parameter γ.

$$\mu_{ji}(I_i) = \begin{cases} \max\left\{0, \dfrac{\gamma(I_i - W_{ij}) + 1}{\gamma(C_{ij} - W_{ij}) + 1}\right\} & \text{if } I_i \leq C_{ij} \\ \max\left\{0, \dfrac{\gamma(1 - I_i - W_{i+M,j}) + 1}{\gamma(1 - C_{ij} - W_{i+M,j}) + 1}\right\} & \text{if } I_i > C_{ij} \end{cases} \tag{6}$$

Therefore, the FasArt membership function value increases as patterns lie nearer to the center of the support, while Fuzzy ARTMAP activation function gives the same value for all the patterns lying inside the box associated with the neuron.

In addition, FasArt is endowed with a learning law for the center of the winner fuzzy support, $\vec{C}_J$. This law is analogous to the law given for LTM weights, $\vec{W}_J$:

$$\vec{C}_J^{\text{new}} = \vec{C}_J^{\text{old}} + \beta_c(\vec{I} - \vec{C}_J^{\text{old}}) \tag{7}$$

where $\beta_c \in [0, 1]$ is the learning rate.

Furthermore, FasArt incorporates defuzzification of the output in function approximation tasks. This processing is allowed due to the correspondence between fuzzy sets and categories established in FasArt. The expression for the defuzzified output $\hat{b}(\vec{I})$ is built over the fuzzy sets generated in the output space by the corresponding unsupervised module:

$$\hat{b}(\vec{I}) = \frac{\sum_{j=1}^{N^a} T_j C_l^b}{\sum_{j=1}^{N^a} T_j} \tag{8}$$

where l is the neuron in F_2^b predicted by j in F_2^a, and N^a is the number of neurons in F_2^a. Defuzzification stands as a precedent for distributed features incorporated into a Fuzzy ARTMAP based neural system, since the output is calculated as a combination of all the rule antecedents (F_2^a neurons) activated by the input pattern.

With respect to the geometric interpretation, FasArt fuzzy supports are no longer hyperboxes. In Fig. 4, a representation of a FasArt fuzzy set with parameters $\vec{W} = [0, 0, 1, 1]$, $\vec{C} = [0.325, 0.52]$ and $1/\gamma \approx 0$ is displayed. The contour plot indicates that FasArt will fit better in oval shaped output classes, while Fuzzy ARTMAP boxes fit better in squared ones. This fact will enable us to study the impact of distributed learning and its relationship to the geometric shape of the borders between output classes.

3. Qualitative analysis of dARTMAP and dFasArt

3.1. Methodology

The work presented in the remainder of this paper is divided in two main parts: a qualitative and a quantitative analysis of the introduction of distributed learning in Fuzzy ARTMAP based neural systems. The qualitative analysis examines the relationships between distributed learning and category reduction in those systems. The analysis starts with a theoretic justification of the reasons why distributed learning may reduce the recruitment of superfluous categories. Then, all the innovations introduced into the original Fuzzy ARTMAP WTA architecture are revised from the point of view of their individual contribution to enable fast and stable distributed learning. Afterward, these innovations are adapted to fuzzy set peculiarities in order to endow a FasArt neuro-fuzzy system with distributed learning. The resulting neuro-fuzzy architecture, called dFasArt (*distributed* FasArt), is capable of fast and stable

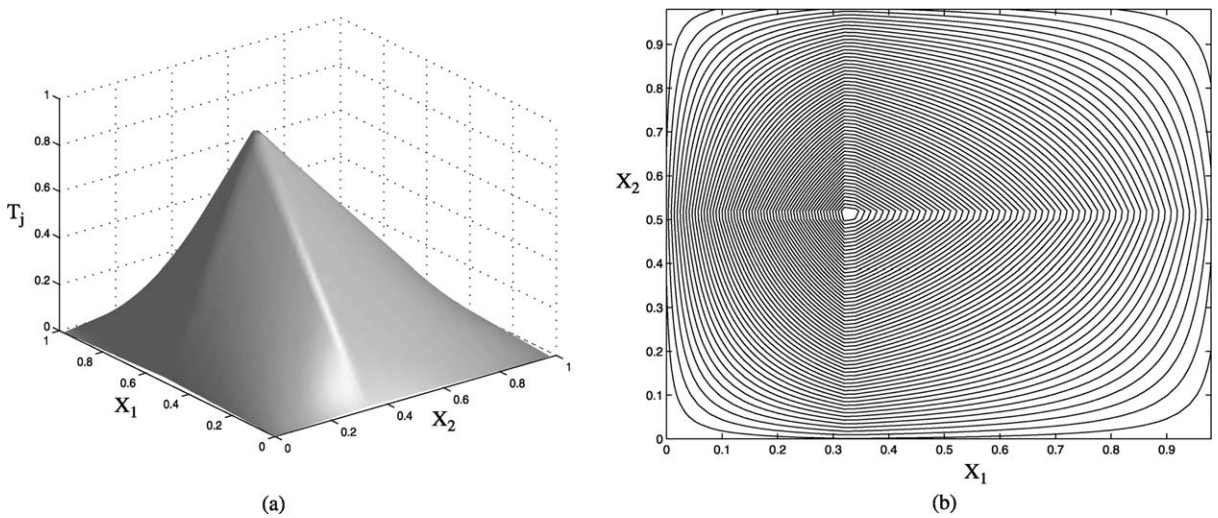

Fig. 4. (a) Graphic representation and (b) contour plot of a 2D FasArt activation function. The fuzzy set associated with this particular neuron is determined under the following conditions: $\vec{W} = [0, 0, 1, 1]$, $\vec{C} = [0.325, 0.52]$ and $(1/\gamma) \approx 0$. Fuzzy support on axis X_1 is given by weights W_1 and W_3 and center C_1, while the one on axis X_2 is given by W_2, W_4 and C_2.

distributed learning and posses meaningful interpretations as both neural and fuzzy systems. In addition to this, the development of dFasArt enables a study of the feasibility and capability of the introduction of distributed learning in other neural systems based on Fuzzy ARTMAP. Moreover, conclusions about the influence of distributed learning on new elements such as those deriving from fuzzy sets theory can be drawn. Finally, the development of dFasArt may be considered an extraction of the general advantageous features of distributed learning and a systematic method for adapting those features to any member of the Fuzzy ARTMAP family.

After the qualitative approximation, a quantitative analysis of distributed learning capabilities is carried out in the experimental work (see Section 4). This second part of the work involves comparing, in several benchmark problems, the distributed versions, dARTMAP and dFasArt, with the original WTA systems, Fuzzy ARTMAP and FasArt. The analysis focuses on the relative reduction in the number of categories and the test accuracy achieved by the distributed systems.

3.2. Analysis of the new elements introduced in dARTMAP

Distributed learning is aimed to correct a major source of category proliferation that is a consequence of Fuzzy ARTMAP WTA learning dynamics. This undesirable defect arises because the learning of every pattern that was not previously covered by a hyperbox implies either the recruitment of a new box or the expansion of a committed one. As every box R_j expands, the activation, T_j, engendered by previously learned input patterns decreases (according to Eq. (2)). This fact may force the recruitment of some new neurons in order to re-learn some of the patterns that j had correctly learned before, because now these patterns induce the highest activation, T_j', in neurons connected to a wrong output class through the interART map.

This phenomenon becomes particularly noticeable near the classification borders. As training proceeds, the whole input space gets completely covered with boxes leading to different output classes, which means that non-square classification borders must be approximated by rectangles. This fact causes a lot of overlapping among boxes leading to different output classes and results in an accentuation of category proliferation.

Distributed coding can help alleviate this problem in the following way. With distributed coding, each learned region of the input space is coded by a group of neurons instead of just one neuron whose box covers all the learned patterns. According to this, an uncovered pattern lying in a region learned in distributed mode produces the same effect as a pattern lying inside a WTA box and thus neither a box expansion nor a neuron recruitment are needed to learn it. Therefore, in distributed learning boxes are not forced to cover all the input space and less learning is involved, thus there is less opportunity for category proliferation to develop.

The aspects of distributed dynamics that solve the category proliferation problem can be summarized as follows. When a pattern is presented to the network, F_2 neurons are activated in a distributed manner, i.e. the value of the activation received by each neuron has to be a function of its LTM component (as in Fuzzy ARTMAP) and of the activation reached by the other neurons. Then, F_2 output is calculated by means of a new competitive rule, that allows several winner neurons to stay active after the competition and learn the pattern in a distributed mode. Supervised learning demands a mechanism to find the output class to which the pattern belongs from the neurons that won the competition. Finally, the actual distributed learning stage proceeds and pattern features are incorporated into the distributed code.

Some innovations are introduced in dARTMAP architecture to enable distributed learning and they are explained below. In Fig. 5, an outline of the dARTMAP structure and neural layers showing these new elements is depicted.

3.2.1. Dynamic weights to enable distributed activation and help avoid catastrophic forgetting

Activation of F_2 neurons when an input pattern is presented to the network is evaluated as a function of the stored weights and the current input pattern. Since dARTMAP requires a new distributed activation, Fuzzy ARTMAP multiplicative weights, W_{ij} are replaced by dynamic weights $[y_j - \tau_{ij}]^+$, where $[a]^+ = \max\{a, 0\}$.

These dynamic weights consist of an LTM threshold, τ_{ij} and of the current value of the neuron output, y_j. For the bottom–up path, τ_{ij} is related to the Fuzzy ARTMAP W_{ij} through the following expression:

$$W_{ij} = 1 - \tau_{ij} \qquad i = 1, \ldots, 2M; \; j = 1, \ldots, N \tag{9}$$

so that the LTM parts of both systems are equivalent. Therefore, thresholds are initialized to 0 and monotonically increase during training. An expression analogous to Eq. (9) stands for the top-down paths.

The current output, y_j, is a *short term memory* (STM) component of the dynamic weight and represents the dependence of the neuron activation on the other neurons.

Furthermore, dynamic weights contribute to avoid catastrophic forgetting during the learning stage, since LTM thresholds impose a limit on the total change suffered by the network after the learning of an individual pattern. As will be explained in Section 3.2.5, in distributed learning laws the learning rate for each connection is multiplied by the value of its dynamic weight, so that only those connections with non-zero weights ($y_j > \tau_{ij}$) modify their LTM thresholds.

In addition to this, dynamic weights involve a new geometric interpretation of dARTMAP. As shown in Fig. 6,

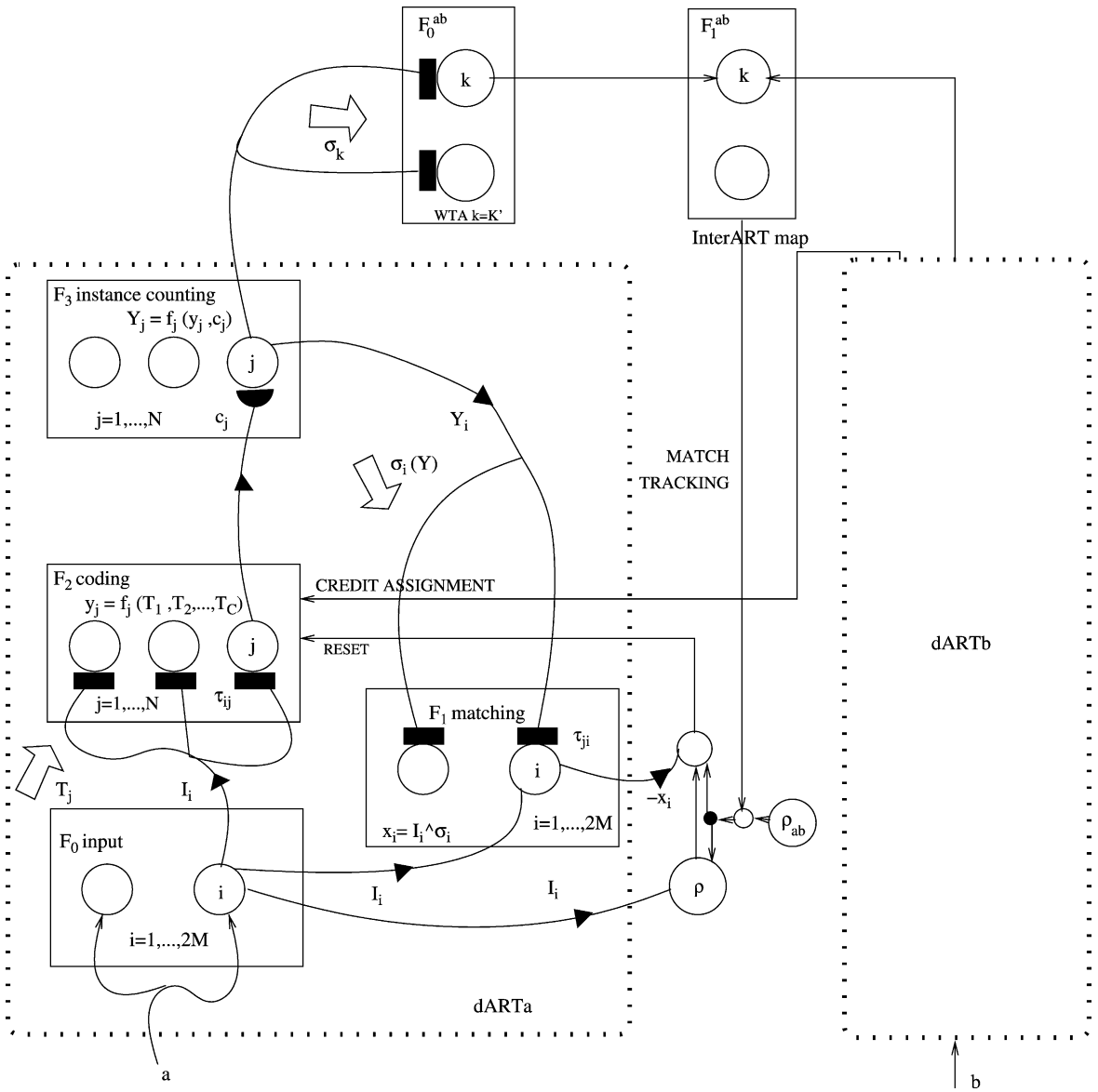

Fig. 5. dARTMAP structure. The dARTb module on the right is the symmetric of the dARTa module on the left. The differences with Fuzzy ARTMAP structure are the introduction of F_3 *instance counting* neural layer, and the information flow loop ($F_0 \rightarrow F_2 \rightarrow F_3 \rightarrow F_1$ instead of Fuzzy ARTMAP $F_0 \rightarrow F_1 \rightarrow F_2$). The structure and function of F_0, F_1 and F_2 neural layers remain the same. Black rectangles indicate the additive thresholds of the dynamic weights, while black semicircles indicate multiplicative weights.

hyperboxes consist of an LTM part plus an STM part. The LTM part is equivalent to the corresponding Fuzzy ARTMAP coding hyperbox, while the STM part can be interpreted as a transitory lengthening toward the current pattern location that depends on the activity of y_j, which in turn depends on other neurons activity. Because of this STM lengthening, there is no need of covering the whole input space with boxes anymore, and thus neuron recruitment is reduced. As long as the F_2 dynamics cover the space, there is no need for learning to expand the LTM boxes.

The matching criterion, ρ, is evaluated using a hyperbox resulting from the combination of the top-down dynamic weights of all the neurons that remain active after the competition.

3.2.2. Distributed activation and competitive rule that allows multiple winners

The value of the activation that each neuron reaches after a pattern presentation is calculated in two stages. At the first one, a temporary activation value, T_j ($y_j = 1, \tau_{1j}, \tau_{2j}, \ldots, \tau_{2M,j}$) is computed using only the LTM thresholds and the input pattern according to:

$$T_j(y_j = 1) = \sum_{i=1}^{2M} I_i \wedge [1 - \tau_{ij}]^+ + (1 - \alpha) \sum_{i=1}^{2M} \tau_{ij} \qquad (10)$$

with $\alpha \in [0, 1]$, $j = 1, \ldots, N$. At this moment, $y_j = 1$ is assumed for all F_2 neurons.

Then, neurons interact in a competitive fashion to find the STM component of their dynamic weights, y_j.

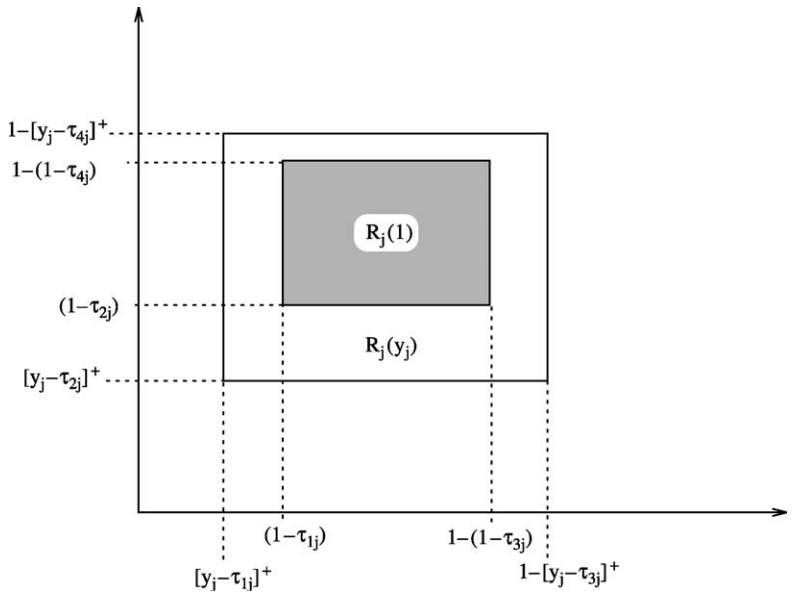

Fig. 6. Geometric interpretation of dARTMAP in 2D. dARTMAP hyperboxes have an LTM part, in gray, and an STM part, in white. LTM part, $R_j(1)$, is equivalent to the corresponding Fuzzy ARTMAP box since dARTMAP thresholds τ_{ij} are related with Fuzzy ARTMAP weights W_{ij} through expression (9). STM part ($R_j(y_j) - R_j(1)$) depends on the location of the current pattern and on the activation reached by the other neurons.

The competition determines the steady state output reached by each neuron for the current pattern using the expression:

$$y_j(T_1, T_2, \ldots, T_N) = \frac{1}{1 + \sum_{\lambda \neq j} \left[\frac{(2-\alpha)M - T_j}{(2-\alpha)M - T_\lambda} \right]^p} \tag{11}$$

with $p \geq 1$

This competitive rule inherits Fuzzy ARTMAP activation function philosophy of giving higher values to neurons with boxes lying closer to the pattern, and to neurons with smaller boxes in case of equal proximity.

If there is any box whose size is of just one pattern (point box) that includes the current input pattern, then expression (11) is no longer valid, and output is calculated as follows:

$$y_j = \begin{cases} \dfrac{1}{\Lambda} & \text{if } j \text{ includes the pattern} \\ 0 & \text{if } j \text{ does not include the pattern} \end{cases} \tag{12}$$

where Λ is the number of point boxes that include the pattern.

After the competition the dynamic weights values, $[y_j - \tau_{ij}]^+$, are completely determined and available for the rest of the input pattern processing. All neurons with $y_j > 0$ are said to have won the competition and contribute as non-zero components in the distributed code of the pattern $\vec{y} = [y_1, y_2, \ldots, y_N]$; but only those neurons with $y_j > \tau_{ij}$ will modify their LTM thresholds to learn the pattern.

3.2.3. Output class prediction implemented as a voting balanced by instance counting

In Fuzzy ARTMAP, the output given by the network is simply the category of Fuzzy ARTb linked with the winner in Fuzzy ARTa through the map. In dARTMAP, a *voting strategy* is implemented, that takes into account the output of every neuron in dARTa that remains active after the competition. The outputs of every neuron linked to the same class are summed, and the network classifies the pattern into the class that has the maximum sum.

Every neuron associated with a particular output class, k, contributes to the sum for class k, thus the prediction is biased toward classes encoded by many F_2^a neurons versus classes encoded by a few neurons. An F_3 neural layer is introduced in the dARTMAP architecture to balance this situation (see Fig. 5). The F_3 layer output is a vector $\vec{Y} = [Y_1, \ldots, Y_N]$ resulting from the normalization of the F_2 output, $\vec{y}$, by the number of patterns previously coded by each category:

$$Y_j = \frac{c_j y_j}{\sum_{\lambda=1}^{N} y_\lambda c_\lambda} \qquad j = 1, \ldots, N \tag{13}$$

where c_j is the number of patterns previously coded by neuron j. This signal $\vec{Y}$ is used to evaluate the matching criterion in layer F_1.

3.2.4. Distributed match tracking that ensures a sufficient resemblance between the distributed code and the pattern

According to the ART dynamics, an evaluation of the match between the input pattern and the distributed code must take place before the learning stage in order to guarantee that the code $\vec{Y}$ is an acceptable representation of that pattern. A distributed match criterion is evaluated between the input pattern and a prototype determined as a combination of the active neurons' top-down dynamic

weights:

$$\sigma_i = \sum_{j=1}^{C} [Y_j - \tau_{ji}]^+ \qquad i = 1, \ldots, 2M \tag{14}$$

Then, if $(|\vec{I} \wedge \vec{\sigma}|/|\vec{I}|) \geq \rho$ the match criterion is satisfied and learning proceeds. Otherwise, reset is triggered and the network operation is switched to WTA mode. After the pattern is correctly classified in the WTA mode (following Fuzzy ARTMAP dynamics), the network is switched back to the distributed mode and the next pattern is presented.

Moreover, if the output class predicted by the voting is not the correct one, the interART reset is triggered and the network is also switched to WTA.

3.2.5. Distributed learning laws to avoid catastrophic forgetting, and credit assignment to preserve network stability during learning

Finally, the LTM thresholds are modified according to the distributed learning laws. In these equations, the learning rate for each threshold is multiplied by its dynamic weight. This dynamic learning rate updates every connection by an amount that is determined by its neuron output, y_j (calculated in a distributed way) and its learning capability, τ_{ij}, instead of the uniform Fuzzy ARTMAP learning rate. Moreover, dynamic weights impose a bound on the total change introduced into the neural network because connections with zero dynamic weights (zero learning rate) will not be updated. This fact, in addition to learning laws that only allow monotonic changes in the weights, prevents catastrophic forgetting.

The bottom–up thresholds are updated as follows:

$$\tau_{ij}^{\text{new}} = \tau_{ij}^{\text{old}} + \beta[y_j - \tau_{ij} - I_i]^+ \tag{15}$$

while the top–down thresholds are updated by:

$$\tau_{ji}^{\text{new}} = \tau_{ji}^{\text{old}} + \beta[\sigma_i^{\text{old}} - I_i] + \frac{[Y_j - \tau_{ji}^{\text{old}}]^+}{\sigma_i^{\text{old}}} \tag{16}$$

Both equations reduce to Fuzzy ARTMAP learning law (4) in the WTA mode (after a mismatch).

In addition to this, instance counting is updated by:

$$c_j^{\text{new}} = c_j^{\text{old}} + y_j \tag{17}$$

Since $\sum_{j=1}^{C} y_j = 1$ $(0 \leq y_j \leq 1)$, it can be assumed that each active neuron j has learned a fraction y_j of a presented pattern after the stabilization of the network.

Previous to thresholds updating, a *credit assignment* procedure takes place in order to reset all the F_2^a neurons that remain active but do not lead to the correct output class. In this way, the discrimination capability of the system is guaranteed since only those neurons that have contributed to the correct prediction are allowed to learn the pattern. After credit assignment, the outputs of the winners are recalculated so that $\sum_{j=1}^{C} y_j = 1$ is satisfied.

3.3. dFasArt: the result of adapting distributed learning dynamics to FasArt

To complete this qualitative analysis, this section studies how to adapt distributed learning to the FasArt neuro-fuzzy system. This will broaden the comparative between distributed and WTA systems. Readers interested in the comparative between Fuzzy ARTMAP and FasArt systems should refer to Cano-Izquierdo et al. (2001). Some dARTMAP elements like instance counting, match tracking, credit assignment, switching to WTA after mismatch and output class prediction are exported to dFasArt architecture without any modification. However, further adaptation is needed for *fuzzy set construction*, a *competitive rule* and *learning laws*.

3.3.1. Distributed fuzzy sets construction

dFasArt fuzzy sets, like dARTMAP categories, must be formed by an LTM part plus an STM lengthening. FasArt already has a similar structure, since a segment of length $1/\gamma$ is added to each side of the fuzzy support prior to the evaluation of the membership function (see Fig. 3). However, this lengthening is a fixed value and does not depend on either the other neurons activity or the location of the current pattern.

Moreover, FasArt fuzzy support size is determined by two parameters: γ, which determines the support maximum increase after each iteration, and ρ, that limits the global increase (during the whole training) of the support. Under certain conditions, interaction between the two parameters may cause recruitment of superfluous neurons, thus boosting category proliferation. In addition to this, the FasArt triangular membership functions give the highest activation value to the neuron with the largest support that include the pattern, which is opposed to Fuzzy ARTMAP and dARTMAP activation, which gives the highest value to the smallest box.

To make FasArt consistent with other distributed ART systems, a new way of building the fuzzy sets is proposed in dFasArt. The new fuzzy sets inherit triangular membership functions, weight vectors $\vec{W}_j$, and centers $\vec{C}_j$, but the role of parameters γ and ρ is now played by a single parameter λ, *the maximum size of each side of the fuzzy support*. In Fig. 7, the construction of a dFasArt fuzzy set is shown. When a pattern is presented, each side of the support stretches until it reaches a size λ. This lengthening forms the STM part. Successive learning increases the LTM part (weights), causing the STM part to decrease.

3.3.2. Distributed competitive rule

Competition in the dARTMAP F_2^a layer, given by expression (11), can be interpreted as a normalization of activation values, T_j, since the output of the layer is such that verifies $\sum_{j=1}^{N} y_j = 1$. Because of the dFasArt's fuzzy nature, the activations are also the degree of membership to fuzzy sets. In this sense, a normalization of the activations

would mean a loss of information about the degree of truth of the individual fuzzy rule associated with each set. To avoid this problem, the dFasArt competition occurs in the following way. All triangular membership functions are evaluated individually for each fuzzy set using the supports constructed as shown in Fig. 7; this temporary membership value is analogous to that of $T_j(y_j = 1)$ calculated in dARTMAP. Afterward, each neuron output is given by the following competitive rule:

$$y_j = \begin{cases} T_j & I_i \in [W_{ij}, 1 - W_{i+M,j}] \ \forall i = 1, \ldots, M \\ T_j \cdot \dfrac{T_j}{\sum_{l=1}^{N} T_l} & \text{otherwise.} \end{cases} \quad (18)$$

If the pattern is included in the LTM part of the support the membership degree is not normalized, preserving the rule degree of truth; otherwise, the membership value T_j is weighted by the ratio of T_j to all the activities. This normalization can be interpreted as a system resistance to modify some rules, when there are other rules that have already involved the pattern. Geometrically, the output normalization is a shortening of the STM part caused by other neurons activity.

After a distributed competition in dFasArt, a match at F_1 is not evaluated because, since the prototypes are also fuzzy rules, a combination of these rules to verify the matching criterion has no sense.

3.3.3. Distributed learning laws

Finally, it is necessary to adapt distributed learning laws to dFasArt weights and center. The equation for weights is derived straightforward from that given for dARTMAP bottom-up thresholds (15) and the expression for transforming thresholds into LTM weights (9):

$$W_{ij}^{\text{new}} = W_{ij}^{\text{old}} - \beta[y_j - (1 - W_{ij}^{\text{old}}) - I_i]^+ \quad (19)$$

This law does not incur catastrophic forgetting since the same bound given for dARTMAP learning stands for dFasArt.

The center of the support also needs special treatment because if FasArt equation (7) is applied directly in distributed learning, the center may be moved out of the support limits. To avoid this, dFasArt center learning distinguishes two cases:

- If the pattern is included in the LTM part of the support and the neuron is active, the center is modified according to Eq. (7).
- If the pattern lies within the STM part, $\vec{C}_j$ is moved toward the geometric center of the support after the weights W_{ij} have been updated, instead of toward the pattern, according to:

$$\vec{C}_j^{\text{new}} = \frac{\vec{C}_j^{\text{old}} \cdot c_j^{\text{old}} + \vec{G}_j^{\text{new}} \cdot (c_j^{\text{new}} - c_j^{\text{old}})}{c_j^{\text{new}}} \quad \vec{I} \text{ in STM} \quad (20)$$

where $\vec{G}_j$ is the geometric center of the support, and c_j is the instance counting for neuron j. Multiplication by instance counting is introduced in the equation due to the fact that center motion implies decreasing the membership of certain points in the fuzzy set and increasing it in others. The more patterns learned by the neuron, the more representative the geometric center of those patterns is and thus it seems reasonable that the system opposes center motions.

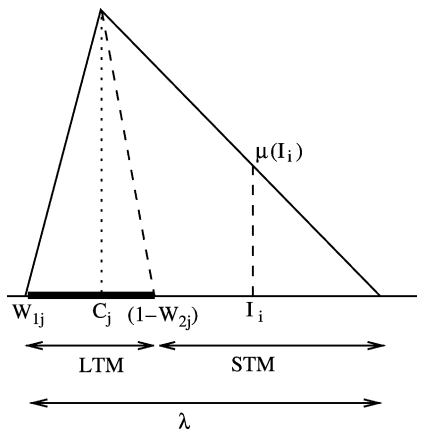

Fig. 7. Construction of dFasArt fuzzy set in 1D with parameter λ. The LTM part of the support (thick line) is equivalent to the corresponding FasArt fuzzy support since it is also determined by LTM weights W_{ij} and C_j. When a pattern $\vec{I}$ is presented, fuzzy support in enlarged toward $\vec{I}$ location until each side reaches a total length of λ. This enlargement forms the STM part of the fuzzy support.

4. Experimental work: quantitative analysis of distributed learning

Experimental work aims to get quantitatively asses distributed learning's impact on category reduction in the studied systems. Category reduction usually involves a small loss of test accuracy. This negative effect is also measured in order to evaluate whether the reduction in the coding set size compensates for the increased test error rate. A pruning postprocessing is also evaluated to compare the performance of distributed and WTA systems under equal category number conditions.

A collection of benchmarks has been selected to compare distributed architectures with the corresponding WTA ones. Seven synthetic classification tasks have been used to reveal distributed learning dependencies on the geometric characteristics of the problem. These benchmarks are based on the circle-in-the-square task (Carpenter et al., 1998) and include simple geometric figures, like squares and circles, in a two dimensional space (the unit square) so as to focus each benchmark on some particular features (see Fig. 8).

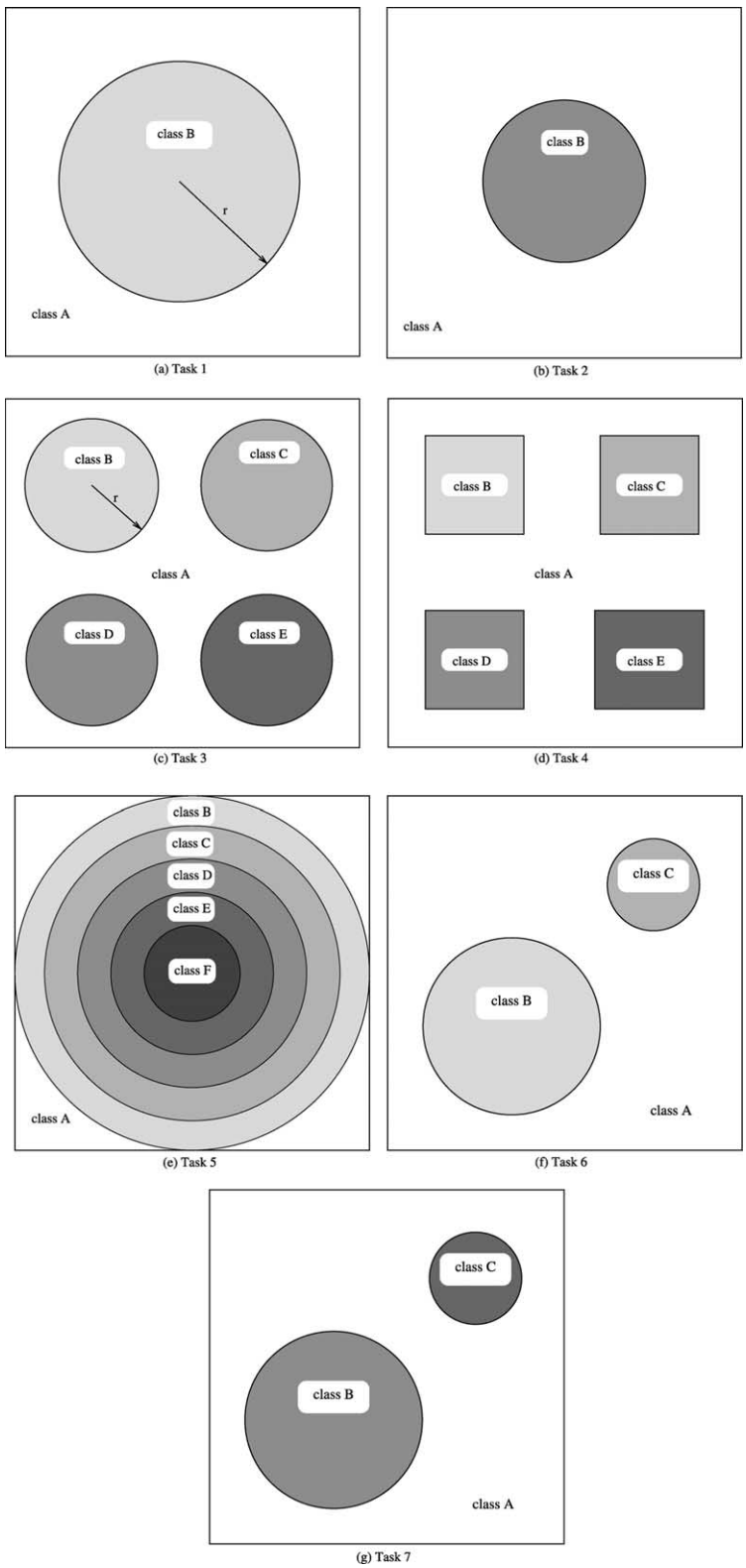

Fig. 8. Classification benchmarks. Different gray scale indicates different output class. All training sets except for task 7 are obtained through uniform sampling of the unit square, therefore the presence of patterns belonging to a certain class is proportional to the area occupied by this class. (a) Task 1: classify the points within a circle in the unit square with both classes being equiprobable (circle-in-the-square benchmark of (Carpenter et al., 1998), (b) task 2: classify the points within a circle in the unit square but class *outside* is far more probable than class *inside*, (c) task 3: classify the points within four circles in the unit square, (d) task 4: classify the points within four squares in the unit square, (e) task 5: classify the points within five concentric rings in the unit square, (f) task 6: classify the points within two circles in the unit square where the number of patterns of each class is proportional to its size, and (g) task 7: same than task 6, but now class *A* has 50% of the patterns, class *B* 30% and class *C* 20%.

Additionally, one $\Re^2 \rightarrow \Re^1$ function approximation task is also included to confirm the results obtained in classification tasks and because FasArt is specially suited for these kinds of problems.

4.1. Benchmark selection

This subsection is focused on the particular problem features that will be studied and the benchmarks that will be used to investigate each feature. A summary of the benchmarks is included in Table 1 and schematics of the classification tasks (T-1 to T-4-n) geometries are depicted in Fig. 8.

4.1.1. Geometries that favor the reduction in the number of categories

As stated in the qualitative analysis, distributed learning is intended to avoid a massive recruitment of categories by reducing the amount of learning involved during the training, i.e. not covering the whole input space with coding boxes. Therefore, we expect the geometrical configuration of the output classes to be of certain influence on the applicability of distributed learning to category reduction. In this sense, there may be problems whose classes are difficult to learn without being completely covered with boxes.

To study this effect, tasks T-3 and T-5 in Fig. 8 are proposed. Task T-3 consists in classifying the points lying in four circles, each one associated with a different class, while in T-5 circles are replaced by annulus. In both tasks all the classes have the same area and similar number of instances. The main difference between both problems is that while circles in T-3 may be coded by few boxes without being completely covered, annulus in T-5 must be completely covered with boxes. Therefore, smaller category reduction rates are expected (if any) in those problems with difficult geometries for distributed learning.

Table 1
Set of benchmarks selected for the experimental work

Name	Dimension	Noise	Number of classes	Probability of each class
T-1	2	No	2	(50/50)
T-2	2	No	2	(70/30)
T-3	2	No	5	(20/20/20/20/20)
T-4	2	No	5	(20/20/20/20/20)
T-5	2	No	6	(16.6/16.6/16.6/ 16.6/16.6/16.6)
T-6	2	No	3	(75/20/5)
T-7	2	No	3	(50/30/20)
T-3-n	2	Yes	5	(20/20/20/20/20)
T-4-n	2	Yes	5	(20/20/20/20/20)
Func -App	2	No	Continuous range	–
Func -App-n	2	Yes	Continuous range	–

For each benchmark, the dimension of the input, the presence or absence of noise in the training set, the number of classes and the probability of each class are given. The training and test sets have 2000 patterns for tasks T-1, T-2, T-3, T-4, T-5, T-6, T-7, T-3-n and T-4-n and 5000 patterns for tasks Func-App and Func-App-n.

4.1.2. Geometries parallel to the axis

Fuzzy ARTMAP approximates the geometric regions associated with each class by hyperrectangles. Consequently, the application of distributed learning to problems with rectangular shapes is expected to achieve category reduction rates smaller than those obtained in problems involving other shapes. On the other hand, FasArt coding surfaces are not rectangles (Fig. 4), and therefore any differences due to distributed coding should not be related to the shape of input pattern distributions.

Tasks selected for this point are T-3 and T-4 (see Fig. 8). In both tasks there are five equiprobable classes, four different *inside* classes and one *outside*; the inside classes are circles in T-3 and rectangles in T-4.

4.1.3. Number of patterns belonging to each class

The output of each neuron in a WTA Fuzzy ARTMAP system depends on both the distance between the box associated with the neuron and the input pattern, as well as on the size of the box. Distributed systems introduce a new element to determine the output of the neuron: the number of patterns learned by the neuron. Therefore, the performance of the system does not only depend on the geometrical shape of the regions forming each class, but also on the quantity of patterns inside those regions and the neighbor regions (associated with different classes). In this sense, neurons coding regions with a high number of patterns will reach a high value of instance counting; therefore, the prediction of the network may turn out to be biased toward the classes associated with those regions.

The combination of the size and the number of patterns of each region in relation to the sizes and number of patterns of the adjoined regions may affect the performance of the system in terms of both category reduction and accuracy. A small and low populated region could be considered as mere noise and incorporated by the system into a more populated region of a different output class.

To measure this effect, two pairs of problems have been selected. The first one involves tasks T-1 and T-2 (see Fig. 8). Both tasks involve classifying the patterns lying inside a circle centered on the unit square. But while in T-1 the circle covers 50% of the area and 50% of the patterns, in T-2 the circle covers only 30% of both area and patterns. The second pair of experiments is formed by T-6 and T-7. The geometrical shapes and sizes of the classes are the same in both problems. In T-6, class *outside* covers 75% of the area and of the patterns; class *big circle* covers 20% of the area and patterns; and class *small circle* 5% of the area and patterns. In T-7, class *outside* covers 50% of the patterns, class *big circle* covers 30% of the patterns, and class *small circle* covers 20% of the patterns. This way, T-6 is a situation

somehow similar to that in T-2. However, in T-7, although there is not a global balance in the number of patterns, output classes are more or less balanced at a local scale of resolution.

4.1.4. Presence of noise

Distributed learning was also proposed as a means for making Fuzzy ARTMAP more robust against the presence of noise in the training set (Carpenter et al., 1998). Distributed label assignment is expected to work better in noisy environments since the decision about the label relies on several neurons rather than just on the winner of the competition, as in WTA architectures.

The impact of distributed learning on increasing the system robustness against noise is studied in this work, repeating tasks T-3 and T-4 after perturbing the training set with Gaussian noise of 0 mean and 0.05 standard deviation.

4.1.5. Function approximation

A $\Re^2 \rightarrow \Re^1$ function given by expression (21) and represented graphically in Fig. 9 was used to train the four studied systems. This function was also used in Marriott and Harrison (1995)

$$f(x,y) = 3[1-(6x-3)]^2 e^{-(6x-3)^2-[(6y-3)+1]^2} - 10\left[\frac{6x-3}{5} - (6x-3)^3 - (6y-3)^5\right] e^{-(6x-3)^2-(6y-3)^2} - \frac{1}{3}e^{-[(6x-3)+1]^2-(6y-3)^2} \quad (21)$$

FasArt and dFasArt performed output defuzzification because of their fuzzy nature, while for Fuzzy ARTMAP and dARTMAP, the center of the coding box in F_2^b associated with the output label was given as the predicted output value.

The influence of noise was studied by introducing additive Gaussian noise with zero mean and 0.04 SD to each pattern in the training set.

4.2. Results

Software simulations of the four studied systems were developed and run on the benchmarks described earlier. For the classification tasks in Fig. 8, the training set was formed by 2000 patterns selected randomly from the unit square. Results are the average of 100 different training sets. The same 5000 test patterns were employed for all the presentations. Network parameters for Fuzzy ARTMAP and dARTMAP were $\rho_a = 0.0$, $\alpha = 0.001$ and $\beta = 1.0$; for FasArt they were $\rho_a = 0.0$, $\gamma = 10$ and $\beta = 1.0$; while for dFasArt they were $\lambda_a = 1.0$, $\rho_a = 0.0$ and $\beta = 1.0$.

In order to compare fairly the drop in accuracy with the reduction of the number of categories achieved by distributed systems, the classification tasks also have been evaluated on the WTA systems, pruned until they had the same number of F_2^a categories than the corresponding distributed ones. This way, we intend to show whether the loss of accuracy is due only to fewer categories or whether distributed learning contributes to not only a more compact codification, but also a more efficient one. The pruning algorithm consists of an iterative scheme that at each step removes the category such that the classification error over the training set is minimized. To reduce from n categories to $n-1$, we classify the training data with each of the n networks of $n-1$ categories, retaining that with smaller classification error on the training set. We repeat this procedure until the total number of categories is that achieved by the distributed architecture. Then, the error indices can be compared fairly.

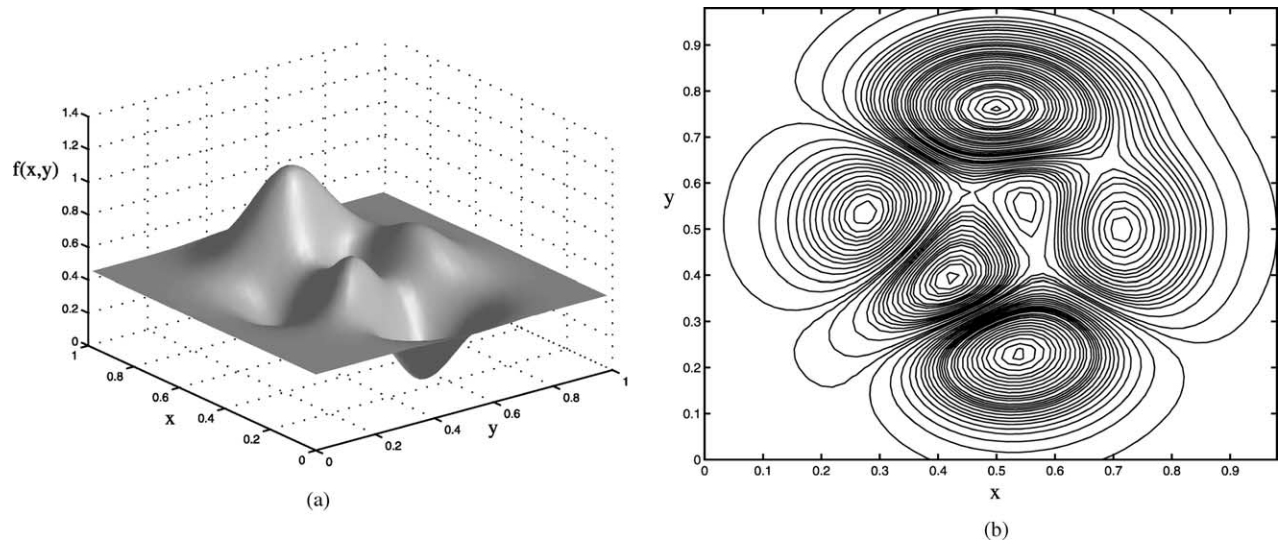

Fig. 9. (a) Graphic representation of function $f(x,y)$, given by expression (21) used for the approximation task. (b) Contour plot of $f(x,y)$.

Table 2
Results for the classification tasks with Fuzzy ARTMAP and dARTMAP

Task	Fuzzy ARTMAP		dARTMAP		Pruned Fuzzy ARTMAP	
	Number of categories	Accuracy (%)	Number of categories	Accuracy (%)	Number of categories	Accuracy (%)
T-1	23.22	93.72	12.34	88.90	12	89.79
T-2	17.64	94.92	6.96	75.79	6.96	84.11
T-3	53.10	88.16	33.38	87.19	33.38	88.18
T-4	31.96	96.21	30.72	95.68	30.26	95.44
T-5	124.82	84.56	125.14	80.85	124.82	84.56
T-6	21.20	94.82	8.64	78.93	8.56	90.22
T-7	20.74	94.61	12.96	95.23	12.54	95.25
T-3-n	145.70	76.52	84.70	81.07	84.70	81.59
T-4-n	175.56	72.76	110.74	80.40	110.18	78.51

In tasks T-x-n ($x = 3, 4$) training patterns are perturbed with noise. For both architectures, the number of categories and the test accuracy are displayed. The last two columns show the performance achieved by a system resulting from pruning Fuzzy ARTMAP until it had the same number of categories than dARTMAP.

Results are shown in Table 2 for the Fuzzy ARTMAP versus dARTMAP comparison and Table 3 for the FasArt versus dFasArt.

For the function approximation task, 5000 training patterns were randomly selected from $[0, 1] \times [0, 1]$ and presented in 100 different orders to compute the averages. Another 5000 patterns were used as the test set. The accuracy of the systems was measured using the mean absolute error (MAE). Network parameters were $\rho_a = 0.0$, $\rho_b = 0.98$, $\alpha = 0.001$ and $\beta = 1.0$ for Fuzzy ARTMAP and dARTMAP; $\rho_a = 0.0$, $\rho_b = 0.98$, $\gamma_a = 1.0$, $\gamma_b = 50$ and $\beta = 1.0$ for FasArt and $\lambda_a = 1.0$, $\lambda_b = 0.02$, $\rho_a = 0.0$ and $\beta = 1.0$ for dFasArt. To make defuzzification meaningful, parameters γ and λ are modified so that in each defuzzification at least three sets have non-zero membership function value, i.e. testing is carried out in a locally distributed fashion. Results on the function approximation task are shown in Tables 4 and 5.

Table 3
Results for the classification tasks with FasArt and dFasArt

Task	FasArt		dFasArt		Pruned FasArt	
	Number of categories	Accuracy (%)	Number of categories	Accuracy (%)	Number of categories	Accuracy (%)
T-1	63.42	96.30	30.50	92.78	30.50	94.83
T-2	56.12	97.00	12.24	96.28	12.24	92.36
T-3	122.30	92.76	67.38	87.95	67.38	91.74
T-4	139.26	91.82	69.32	91.29	69.32	89.56
T-5	278.7	95.95	179.74	73.39	179.74	73.18
T-6	62.18	96.33	22.24	89.76	22.24	94.16
T-7	57.68	96.27	24.32	94.91	24.32	94.67
T-3-n	355.48	77.21	176.38	80.40	176.38	84.69
T-4-n	405.24	73.06	214.16	78.72	214.16	79.84

In tasks T-x-n training patterns are perturbed with noise. For both architectures, the number of categories and the test accuracy are displayed. The last two columns show the performance achieved by a system resulting from pruning FasArt until it had the same number of categories than dFasArt.

Table 4
Results for the approximation task with Fuzzy ARTMAP and dARTMAP

Noise	Fuzzy ARTMAP			dARTMAP			Relative	
	Cat(A)	Cat(B)	MAE	Cat(A)	Cat(B)	MAE	↓ Cat(A) (%)	↑ MAE(%)
No	1350.52	62.68	0.015	1309.48	62.68	0.064	3.04	318.18
Yes	4133.94	64.14	0.033	3863.84	64.14	0.057	6.53	111.63

For both architectures, the number of categories recruited by each unsupervised module (Cat(A) and Cat(B)) and the MAE are shown. The last two columns show the reduction in the number of categories (↓ Cat(A)) and the increase in test MAE (↑ MAE) achieved by dARTMAP over Fuzzy ARTMAP results.

Table 5
Results for the approximation task with FasArt and dFasArt

Noise	FasArt			dFasArt			Relative	
	Cat(A)	Cat(B)	MAE	Cat(A)	Cat(B)	MAE	↓ Cat(A) (%)	↑ MAE(%)
No	1802.0	62.6	0.006	1619.78	61.00	0.0071	10.11	18.33
Yes	4439.24	63.52	0.016	4045.0	62.44	0.017	8.88	6.25

For both architectures, the number of categories recruited by each unsupervised module (Cat(A) and Cat(B)) and the MAE are shown. The last two columns show the reduction in the number of categories (↓ Cat(A)) and the increase in test MAE (↑ MAE) achieved by dFasArt over FasArt results.

4.3. Discussion

4.3.1. Geometries that favor the reduction in the number of categories

The comparison between dARTMAP and Fuzzy ARTMAP performance on tasks T-3 and T-5 shows that the geometrical configuration of the output classes is a key factor for the usability of distributed learning. In T-3 each circle can be coded in a distributed way without covering its whole surface with boxes and a category reduction rate of about 30% is achieved. On the other hand, in T-5 no reduction is achieved since the annulus is a geometric figure that cannot be learned without being completely covered with boxes.

In general terms, output classes with their patterns spread over several different regions of the input space are not going to experience distributed learning reduction in the number of categories.

With respect to FasArt and dFasArt, it has to be noticed that there is a certain reduction in the number of fuzzy sets, but this is due mainly to the improvements made in the construction of those fuzzy sets, not to distributed learning.

4.3.2. Geometric shape of output classes

In T-4, dARTMAP does not achieve a significant category reduction (4%) in relation to that obtained in T-3 (37.14%). As explained before, it is due to the fact that Fuzzy ARTMAP coding rectangles fit perfectly in those class domain geometries and there is no place for further category reduction. Moreover, the distributed architecture only slightly decreases the accuracy achieved by the WTA one.

On the other hand, dFasArt achieves significant category reduction rates (over 40%) in both tasks, since FasArt does not take any particular advantage of the geometric shapes of the output classes.

4.3.3. Number of patterns belonging to each class

The result of the comparison on the first pair of tasks, T-1 and T-2, reveals that the proportion of patterns belonging to each class in the training set is a key factor for the accuracy of distributed learning. While in T-1 a significant reduction in the complexity of the network (46.86%) leads to a 5.14% drop in accuracy, in T-2 the precision falls to 75.79%, which indicates that dARTMAP is predicting the majority class (here, the baseline precision would be 70%). This negative effect is due to the fact that instance counting biases the prediction of the output class toward the more populated classes. As introduced before, the output of a neuron in a distributed system is a function of the distance to the pattern and of the number of patterns previously learned. This experiment points out that under certain conditions the influence of the instance counting may be excessively dominant over the distance.

However, the results with dFasArt in terms of classification error are more accurate. This is a consequence of dFasArt performing a locally distributed test instead of a global one. In dFasArt, due to the fuzzy triangular activation, the trade-off between distance to the pattern and instance counting is somewhat balanced. This balance is enforced by the penalty suffered by the neurons whose LTM support does not include the pattern.

The second pair of experiments, T-6 and T-7, confirms the idea that the distributed learning usability is strongly dependent on a function of the geometric shape (size) and the number of patterns contained inside each one of the regions into which each class is divided. In T-6 dARTMAP achieves a good complexity reduction, but again its accuracy drops to a value close to that we would achieve using a predict-the-most-probable-class strategy. However, in T-7, where the populations of the classes have been chosen so that they are more or less balanced, dARTMAP outperforms Fuzzy ARTMAP in both accuracy and complexity. The results corresponding to dFasArt confirm this strong dependence of distributed learning performance on the relation between the number of patterns and the geometric size of the region containing those patterns.

The explanation for this behavior observed in the distributed systems is that they perform a kind of *low-pass filtering* of the information from the input space that leads to an increase of their generalization capability. In this sense, the size of the region and the number of patterns inside it determine if the distributed system is going to consider this region as valuable information or just noisy patterns inside a bigger region associated with a different class. In other words, the region taken as *dominant* by the distributed system absorbs the less significant region.

According to this reasoning, in T-2, dARTMAP considers most of the circle to be a set of noisy patterns in class *outside*. In T-6, both circles are considered to be

mostly noise; hence the classification error (21%) is slightly below the proportion of patterns inside both circles (25%). On the other hand, the dFasArt locally distributed test relies mostly on those neurons whose boxes are closest to the pattern. This fact makes the filtering carried out by distributed learning more selective. In T-2, dFasArt perfectly classifies the circle in spite of being far less populated than the other class. However, in T-6 dFasArt considers most of the big circle (20%) of the patterns as a significant region, but cannot detect properly the small circle (5% of the patterns), which results in a classification error of about 10%.

4.3.4. Presence of noise

Fuzzy ARTMAP and dARTMAP performances on connected geometries with noisy training data (T-3-n and T-4-n) achieve similar category reduction rates (over 30%) as those without noise. However, dARTMAP results in tasks with noise show a significant improvement in test accuracy of about 6%, which agrees with expectations.

Experiments with FasArt and dFasArt confirm the robustness of distributed systems since distributed learning contributes to filter the noise contained in data.

4.3.5. Function approximation

In Fig. 9b, a contour plot of the approximation of $f(x, y)$ made with a Fuzzy ARTMAP neural network is shown. Notice that the distribution of classes obtained in such a way usually yields very unconnected problem geometries, so distributed learning shall be of reduced usability. Experimental results of the four systems prove this statement. Category reductions do not exceed 10% (Tables 4 and 5), which definitely is not a significant gain over WTA. The MAE of dARTMAP regression widely exceeds that achieved with Fuzzy ARTMAP, while dFasArt MAE has more acceptable figures. The difference in MAE is due mostly to the fact that dFasArt performs defuzzification of the output, added to the fact that dARTMAP uses all the committed neurons for the label assignment, while the dFasArt competitive rule performs a local distributed label assignment.

4.3.6. Pruning or distributed learning to reduce category proliferation

After studying the capabilities of distributed learning as a means to reduce the complexity of Fuzzy ARTMAP based systems, here we analyze its performance in comparison to a postprocessing of the network consisting of pruning those categories that least contribute to classifying the patterns correctly. In the experiments carried out in this work, the pruning stops when the WTA system achieves the same number of categories than the distributed one.

Results in Tables 2 and 3 illustrate that pruned Fuzzy ARTMAP achieves a smaller classification error than dARTMAP in seven out of the nine proposed tasks. These differences are $<1\%$, except for T-2 and T-6, where the configuration of the problem is clearly unfavorable for distributed learning, as explained earlier. On the contrary, T-4 is a problem whose conditions are conducive to Fuzzy ARTMAP; distributed learning performs better than pruning. This is due to the fact that Fuzzy ARTMAP fits perfectly into T-4's geometry, and none of the categories requires pruning.

With respect to dFasArt and FasArt, pruned FasArt outperforms dFasArt in five out of the nine tasks and both performances are equivalent in two tasks. Between these two systems, differences are greater than between Fuzzy ARTMAP and dARTMAP (around 2 and 5%).

Therefore, it seems that in certain problems a pruned system that was trained according to a WTA algorithm may outperform a distributed system in terms of the reduction of the complexity of the network. However, distributed systems preserve the on-line characteristic of Fuzzy ARTMAP learning. Moreover, pruning may be computationally costly, at least optimal pruning algorithms involving the evaluation of n networks to reduce from n to $n-1$ categories, unless heuristic methods are developed to make training and pruning computationally affordable.

5. Conclusions and future work

This paper studies the usability of distributed learning as a means to reduce category proliferation in Fuzzy ARTMAP without loss of its on-line and fast learning features.

An important contribution of this paper is the study of the portability of distributed learning into other members of the Fuzzy ARTMAP family. The qualitative analysis of the new elements introduced in dARTMAP together with the procedure for adapting those elements into FasArt architecture can be used as a basis for endowing other Fuzzy ARTMAP based systems with distributed learning capability. We compared the distributed architectures and the original WTA ones, and the WTA ones after pruning, according to several classification and regression benchmarks, in order to study quantitatively the impact of distributed learning on category reduction and test accuracy. Experimental results show that in terms of classification error, pruning may be a better choice to reduce the number of categories in certain problems. However, the systems resulting from this training and postprocessing scheme do not preserve the on-line feature of Fuzzy ARTMAP, and its computational cost may be extremely high.

The first conclusion extracted from this work is that sparse output classes whose patterns are spread through different regions in the input space are very difficult to learn in a distributed manner while reducing the number of categories achieved with a WTA system. Function approximation problems require a clustering of the output space that result in classes with a high degree of sparsity. Consequently, distributed learning results in low usability in these problems. In addition, if the geometric shapes of the output

classes are parallel to the axis, i.e. classes form hyperrectangles, the applicability of distributed learning is also reduced since Fuzzy ARTMAP coding hyperrectangles fit perfectly into these geometries and there is no possible reduction in the number of categories if a distributed learning is performed.

The main conclusion is that distributed systems perform a low pass filtering of the input space that may improve their generalization capabilities with respect to that of original WTAs. This is because the activation reached by a neuron depends on the number of patterns that it has already learned. In this sense, distributed systems tend to mistake small regions with few patterns as groups of noisy patterns inside a larger region associated with a different output class. A distributed test that gives more importance to the distance between the neuron and the pattern (like that performed in dFasArt) contributes to alleviate this effect.

Furthermore, because of the instance counting mechanism, classes with a low probability of occurrence are classified by dARTMAP with high error rates despite being correctly learned. In dFasArt this effect is attenuated by (i) the triangular membership functions, and (ii) a competitive rule that increases the difference of activation between neurons that have already learned the pattern and those that have not learned it yet.

The above mentioned improvement in the generalization capabilities of distributed systems makes them outperform the WTA ones when training patterns are affected by noise. Experiments with pruned WTA systems confirm this statement, since once the neurons that have learned merely noise are removed from the network, performance of the WTA system increases.

Finally, the porting of distributed learning onto dFasArt has confirmed all the conclusions reported here and has pointed out the convenience of performing a locally distributed test. The locally distributed test performed by FasArt and dFasArt appears to yield more accurate systems than the dARTMAP test, which employs all the committed neurons. Future work should look into exploiting local features in a distributed way. Therefore, the key for full usability of distributed systems over WTA ones can be found in (i) a locally distributed test that outperforms both WTA and a globally distributed test, and (ii) a competitive rule that (a) enhances differences between neuron activations as described before and (b) balances the geometrical features of the problem and the influence of the number of patterns of each output class and their distribution into regions.

Acknowledgements

This research was made during E. Parrado-Hernández's stay at Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática, Universidad de Valladolid.

Authors are especially grateful to Dr J.M. Cano Izquierdo for his valuable suggestions and support. We also thank Dr M.J. Araúzo Bravo, Dr G.I. Sainz Palmero, Dr J.López Coronado and R. Sacristán Martínez for their helpful comments. Finally, we would like to thank Ashley Williams for her support reviewing the English usage.

Mr E. Parrado-Hernández has been partially supported by Spain CICYT grant TIC 1999-0216.

References

Cano-Izquierdo, J (1997). *Neuro-fuzzy models for identification and control*. PhD Thesis, School of Industrial Engineering, University of Valladolid (in Spanish).

Cano-Izquierdo, J., Dimitriadis, Y., Araúzo-Bravo, M., & López-Coronado, J. (1996). FasArt: a new neuro-fuzzy architecture for incremental learning in system identification. *Proceedings of the IFAC World Congress*, *IFAC96*, 133–138.

Cano-Izquierdo, J., Dimitriadis, Y., Gómez-Sánchez, E., & López-Coronado, J. (2001). Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, *14*(4/5), 407–425.

Carpenter, G. (1994). Fuzzy ART. In B. Kosko (Ed.), *Fuzzy Engineering*. Carmel: Prentice-Hall.

Carpenter, G. (1997). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks*, *10*(8), 1473–1494.

Carpenter, G., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, *37*, 54–115.

Carpenter, G., & Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, *21*(3), 77–88.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J., & 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., & Reynolds, J. (1991a). ARTMAP: supervised real-learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*(5), 565–588.

Carpenter, G., Grossberg, S., & Rosen, D. (1991b). Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*(1), 759–771.

Carpenter, G., & Markuzon, N. (1998a). ARTMAP-IC and medical diagnosis: instance counting and inconsistent cases. *Neural Networks*, *11*(2), 323–336.

Carpenter, G., Milenova, B., & Noeske, B. (1998b). Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks*, *11*(5), 793–813.

Carpenter, G., & Ross, W. (1995). ART-EMAP: a neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, *6*(4), 805–818.

Georgiopoulos, M., Fernlund, H., Bebis, G., & Heileman, G. L. (1996). Order of search in fuzzy ART and fuzzy ARTMAP: effect of the choice parameter. *Neural Networks*, *9*(9), 1541–1559.

Grossberg, S. (1982a). *Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models. Studies of mind and brain*, Boston, MA: Reidel, pp. 425–447.

Grossberg, S. (1982b). *A theory of human memory: Self-organization and performance of sensor-motor codes, maps, and plans. Studies of mind and brain*, Boston, MA: Reidel, pp. 498–639.

Grossberg, S. (1988). Nonlinear neural networks: principles, mechanisms, and architectures. *Neural Networks*, *1*(1), 17–61.

Kung, S., & Taur, J. (1995). Decision-based neural networks with signal/image classification applications. *IEEE Transactions on Neural Networks*, *6*(1), 170–181.

Lin, C., & Lin, C. (1997). An ART-Based fuzzy adaptive learning control network. *IEEE Transactions on Fuzzy Systems*, *5*(4), 477–496.

Marriott, S., & Harrison, R. (1995). A modified Fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural Networks*, *8*(4), 619–641.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan Books.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In J. Rumelhart, & D. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructures of cognitions*. Cambridge, MA: MIT Press.

Simpson, P. (1992). Fuzzy min–max neural networks. Part 1: classification. *IEEE Transactions on Neural Networks*, *3*(5), 776–786.

Simpson, P. (1993). Fuzzy min–max neural networks. Part 2: clustering. *IEEE Transactions on Fuzzy Systems*, *1*(1), 32–45.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.