Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. IV (2009), No. 2, pp. 104-117

A Novel Fuzzy ARTMAP Architecture with Adaptive Feature Weights based on Onicescu's Informational Energy

Răzvan Andonie, Lucian Mircea Sasu, Angel Caţaron

Răzvan Andonie
Computer Science Department
Central Washington University, Ellensburg, USA
and
Department of Electronics and Computers
Transylvania University of Braşov, Romania
E-mail: andonie@cwu.edu

Angel Caţaron
Department of Electronics and Computers
Transylvania University of Braşov, Romania
E-mail: cataron@vega.unitbv.ro

Lucian Mircea Sasu
Applied Informatics Department
Transylvania University of Braşov, Romania
E-mail: lmsasu@unitbv.ro

Abstract: Fuzzy ARTMAP with Relevance factor (FAMR) is a Fuzzy ARTMAP (FAM) neural architecture with the following property: Each training pair has a relevance factor assigned to it, proportional to the importance of that pair during the learning phase. Using a relevance factor adds more flexibility to the training phase, allowing ranking of sample pairs according to the confidence we have in the information source or in the pattern itself.
We introduce a novel FAMR architecture: FAMR with Feature Weighting (FAMRFW). In the first stage, the training data features are weighted. In our experiments, we use a feature weighting method based on Onicescu's informational energy (IE). In the second stage, the obtained weights are used to improve FAMRFW training. The effect of this approach is that category dimensions in the direction of relevant features are decreased, whereas category dimensions in the direction of non-relevant feature are increased. Experimental results, performed on several benchmarks, show that feature weighting can improve the classification performance of the general FAMR algorithm.
Keywords: Fuzzy ARTMAP, feature weighting, LVQ, Onicescu's informational energy.

1 Introduction

The FAM architecture is based upon the adaptive resonance theory (ART) developed by Carpenter and Grossberg [7]. FAM neural networks can analyze and classify noisy information with fuzzy logic, and can avoid the plasticity-stability dilemma of other neural architectures. The FAM paradigm is prolific and there are many variations of Carpenter's *et al.* [7] initial model: ART-EMAP [9], dARTMAP [8], Boosted ARTMAP [27], Fuzzy ARTVar [12], Gaussian ARTMAP [28], PROBART [21], PFAM [20],

Copyright © 2006-2009 by CCC Publications

Ordered FAM [11], and μARTMAP [14]. The FAM model has been incorporated in the MIT Lincoln Lab system for data mining of geospatial images because of its computational capabilities for incremental learning, fast stable learning, and visualization [25].

One way to improve the FAM algorithm is to generalize the distance measure between vectors [10]. Based on this principle, we introduced in previous work [2] a novel FAM architecture with distance measure generalization: FAM with Feature Weighting (FAMFW). Feature weighting is a feature importance ranking algorithm where weights, not only ranks, are obtained. In our approach, training data feature weights were first generated. Next, these weights were used by the FAMFW network, generalizing the distance measure. Potentially, any feature weighting method can be used, and this makes the FAMFW very general.

Feature weighting can be achieved, for example, by LVQ type methods. Several such techniques have been recently introduced. These methods combine the LVQ classification with feature weighting. In one of these approaches, RLVQ (Relevance LVQ), feature weights were determined to generalize the LVQ distance function [16]. A modification of the RLVQ model, GRLVQ (Generalized RLVQ), has been proposed in [18]. The SRNG (Supervised Relevance Neural Gas) algorithm [17] combines the NG (Neural Gas) algorithm [22] and the GRLVQ. NG [22] is a neural model applied to the task of vector quantization by using a neighborhood cooperation scheme and a soft-max adaptation rule, similar to the Kohonen feature map.

In [1], we introduced the Energy Supervised Relevance Neural Gas (ESRNG) feature weighting algorithm. The ESRNG is based on the SRNG model. It maximizes Onicescu's IE as a criteria for computing the weights of input features. The ESRNG is the feature weighting algorithm we used in [2], in combination with our FAMFW algorithm .

FAMR is a FAM incremental learning system introduced in our previous work [4]. During the learning phase, each sample pair is assigned a relevance factor proportional to the importance of that pair. The FAMR has been successfully applied to classification, probability estimation, and function approximation. In FAMR, the relevance factor of a training pair may be user-defined, or computed, and is proportional to the importance of the respective pair in the learning process.

In the present paper, we focus on the FAMR neural network, the ESRNG feature weighting algorithm, and the distance measure generalization principle. We contribute the following:

1. We introduce a novel FAMR architecture with distance measure generalization: FAMR with Feature Weighting (FAMRFW), adapting the FAMFW model for the FAMR case.

2. Compared to [2], we include new experiments on standard benchmarks.

We first introduce the basic FAM and FAMR notations (Section 2), and the ESRNG feature weighting algorithm (Section 3). In Section 4, we describe the new FAMRFW algorithm, which uses a weighted distance measure. Section 5 contains experimental results performed with the FAMRFW method. Section 6 contains the final remarks.

2 A brief description of the FAMR

We will summarize the FAM standard architecture and the FAMR learning mechanism, which differentiates it from the standard FAM.

2.1 The FAM architecture

A detailed FAM description can be found in Carpenter's *et al.* seminal paper [7], but more simplified presentations are given in [26] and [19].

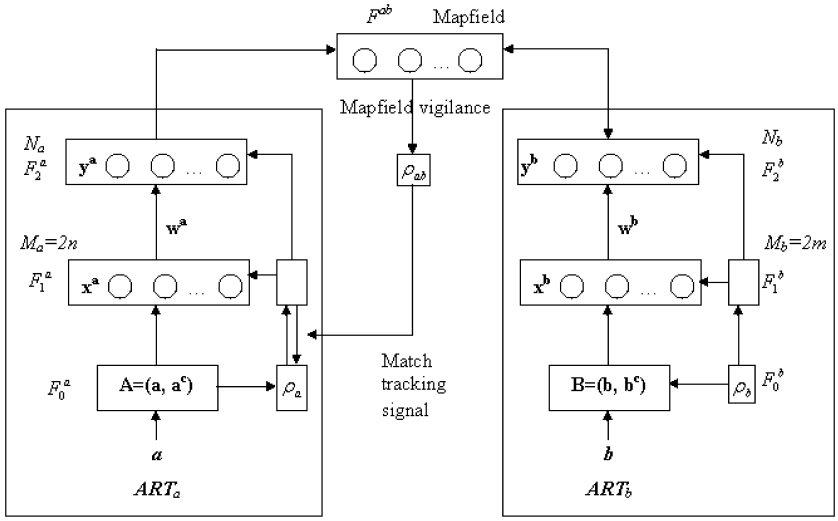


Figure 1: Fuzzy ARTMAP architecture [7].

The FAM architecture consists of a pair of fuzzy ART modules, ART_a and ART_b, connected by an inter–ART module called Mapfield (see Fig. 1). ART_a and ART_b are used for coding the input and output patterns, respectively, and Mapfield allows mapping between inputs and outputs. The ART_a module contains the input layer F_1^a and the competitive layer F_2^a. A preprocessing layer F_0^a is also added before F_1^a. Analogous layers appear in ART_b.

The initial input vectors have the form: $\mathbf{a} = (a_1, \ldots, a_n) \in [0,1]^n$. A data preprocessing technique called *complement coding* is performed by the F_0^a layer in order to avoid node proliferation. Each input vector $\mathbf{a}$ produces the normalized vector $\mathbf{A} = (\mathbf{a}, \mathbf{1} - \mathbf{a})$ whose L_1 norm is constant: $|\mathbf{A}| = n$.

Let M_a be the number of nodes in F_1^a and N_a be the number of nodes in F_2^a. Due to the preprocessing step, $M_a = 2n$. The weight vector between F_1^a and F_2^a is $\mathbf{w}^a$. Each F_2^a node represents a class of inputs grouped together, denoted as a *category*. Each F_2^a category has its own set of adaptive weights stored in the form of a vector $\mathbf{w}_j^a, j = 1, \ldots N_a$, whose geometrical interpretation is a hyper-rectangle inside the unit box. Similar notations are used for the ART_b module. For a classification problem, the class index is the same as the category number in F_2^b, thus ART_b can be substituted with a vector.

The Mapfield module allows FAM to perform associations between ART_a and ART_b categories. The number of nodes in Mapfield is equal to the number of nodes in F_2^b. Each node j from F_2^a is linked to each node from F_2^b via a weight vector $\mathbf{w}_j^{ab}$.

The learning algorithm is sketched below. For each training pattern, the vigilance parameter factor ρ_a is set equal to its baseline value, and all nodes are not inhibited. For each (preprocessed) input $\mathbf{A}$, a fuzzy choice function is used to get the response for each F_2^a category:

$$T_j(\mathbf{A}) = \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{\alpha_a + |\mathbf{w}_j^a|}, \quad j = 1, \ldots, N_a \tag{1}$$

Let J be the node with the highest value computed as in (1). If the resonance condition from eq. (2) is not fulfilled:

$$\rho(\mathbf{A}, \mathbf{w}_J^a) = \frac{|\mathbf{A} \wedge \mathbf{w}_J^a|}{|\mathbf{A}|} \geq \rho_a, \tag{2}$$

then the Jth node is inhibited such that it will not participate to further competitions for this pattern and a new search for a resonant category is performed. This might lead to creation of a new category in ART_a.

A similar process occurs in ART_b and let K be the winning node from ART_b. The F_2^b output vector is set to:

$$y_k^b = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{otherwise} \end{cases} \qquad k = 1, \ldots, N_b \tag{3}$$

An output vector $\mathbf{x}^{ab}$ is formed in Mapfield: $\mathbf{x}^{ab} = \mathbf{y}^b \wedge \mathbf{w}_j^{ab}$. A Mapfield vigilance test controls the match between the predicted vector $\mathbf{x}^{ab}$ and the target vector $\mathbf{y}^b$:

$$\frac{|\mathbf{x}^{ab}|}{|\mathbf{y}^b|} \geq \rho_{ab} \tag{4}$$

where $\rho_{ab} \in [0, 1]$ is a Mapfield vigilance parameter. If the test from (4) is not passed, then a sequence of steps called match tracking is initiated (the vigilance parameter ρ_a is increased and a new resonant category will be sought for ART_a); otherwise learning occurs in ART_a, ART_b, and Mapfield:

$$\mathbf{w}_J^{a(new)} = \beta_a \left(\mathbf{A} \wedge \mathbf{w}_J^{a(old)} \right) + (1 - \beta_a) \mathbf{w}_J^{a(old)} \tag{5}$$

(and the analogous in ART_b) and $\mathbf{w}_{Jk}^{ab} = \delta_{kK}$, where δ_{ij} is Kronecker's delta. With respect to β_a, there are two learning modes: *i)* fast learning for $\beta_a = 1$ for the entire training process, and *ii)* fast-commit and slow-recode learning corresponds to setting $\beta_a = 1$ when creating a new node and $\beta_a < 1$ for subsequent learning.

2.2 The FAMR learning mechanism

The main difference between the FAMR and the original FAM is the updating scheme of the w_{jk}^{ab} weights. The FAMR uses the following iterative updating [4]:

$$w_{jk}^{ab(new)} = \begin{cases} w_{jk}^{ab(old)} & \text{if } j \neq J \\ w_{JK}^{ab(old)} + \frac{q_t}{Q_J^{new}} \left(1 - w_{JK}^{ab(old)} \right) & \\ w_{Jk}^{ab(old)} \left(1 - \frac{q_t}{Q_J^{new}} \right) & \text{if } k \neq K \end{cases} \tag{6}$$

where q_t is the relevance assigned to the tth input pattern ($t = 1, 2, \ldots$), and $Q_J^{new} = Q_J^{old} + q_t$. The *relevance* q_t is a real positive finite number directly proportional to the importance of the experiment considered at step t. This w_{jk}^{ab} approximation is a correct biased estimator of the posterior probability $P(k|j)$, the probability of selecting the k-th ART_b category after having selected the j-th ART_a category [4].

Let $\mathbf{Q}$ be the vector $[Q_1 \ldots Q_{N_a}]$; initially, each Q_j ($1 \leq j \leq N_a$) has the same initial value q_0. N_a and N_b are the number of categories in ART_a and ART_b, respectively. These are initialized at 0. For incremental learning of one training pair, the FAMR Mapfield learning scheme is described by Algorithm 1. The vigilance test is:

$$N_b \, w_{JK}^{ab} \geq \rho_{ab} \tag{7}$$

For a clearer presentation, not to create a confusion between vector relevancies and feature weights, we will assume in all our following experiments that relevancies are set to a constant positive value. Since we actually do not use relevances, is this FAMR equivalent to the standard FAM model, as introduced in [7]? The answer is no, because, unlike the standard FAM: *i)* the FAMR accepts one-to-many relationships; and *ii)* the FAMR is a conditional probability estimator, with an estimated convergence rate computed in [4].

Algorithm 1 The t-th iteration in the FAMR Mapfield algorithm [4].

Step 1. Accept the t-th vector pair $(\mathbf{a}, \mathbf{b})$ with relevance factor q_t.
Step 2. Find a resonant category in ART_b or create a new one.
if $|\mathbf{b} \wedge \mathbf{w}_k^b|/|\mathbf{b}| < \rho_b$, for $k = 1, \ldots, N_b$ **then**
  $N_b = N_b + 1$ {add a new category to ART_b}
  $K = N_b$
  if $N_b > 1$ **then**
    $w_{jK}^{ab} = \frac{q_0}{N_b Q_j}$, for $j = 1, \ldots, N_a$ {append new component to $\mathbf{w}_j^{ab}$}
    $w_{jk}^{ab} = w_{jk}^{ab} - \frac{w_{jK}^{ab}}{N_b - 1}$, for $k = 1, \ldots, K-1$; $j = 1, \ldots N_a$ {normalize}
  end if
else
  Let K be the index of the ART_b category passing the resonance condition and with maximum activation function.
end if
Step 3. Find a resonant category in ART_b or create a new one.
if $|\mathbf{a} \wedge \mathbf{w}_j^a|/|\mathbf{a}| < \rho_a$, for $j = 1, \ldots, N_a$ **then**
  $N_a = N_a + 1$ {add a new category to ART_a}
  $J = N_a$
  $Q_J = q_0$ {append new component to $\mathbf{Q}$}
  $w_{Jk}^{ab} = 1/N_b$, for $k = 1, \ldots, N_b$ {append new row to $\mathbf{w}^{ab}$}
else
  Let J be the index of the ART_a category passing the resonance condition and with maximum activation function.
end if
Step 4. J, K are winners or newly added nodes. Check if match tracking applies.
if vigilance test (7) is passed **then**
  {learn in Mapfield}
  $Q_J = Q_J + q_t$
  $w_{JK}^{ab} = w_{JK}^{ab} + \frac{q_t}{Q_J}(1 - w_{JK}^{ab})$
  $w_{Jk}^{ab} = w_{Jk}^{ab}\left(1 - \frac{q_t}{Q_J}\right)$, for $k = 1, \ldots, N_b$, $k \neq K$
else
  perform match tracking and restart from step 3
end if

3 The ESRNG feature weighting algorithm

We use the ESRNG feature weighting algorithm to compute the generalized distance measure in the FAMRFW. Details of the ESRNG algorithm can be found in [1]. Is is based on Onicescu's IE, and approximates the unilateral dependency of random variables by Parzen windows approximation. Before outlining the principal steps of the ESRNG method, we review the basic properties of the IE.

3.1 Onicescu's informational energy

For a discrete random variable X with probabilities p_k, the IE was introduced in 1966 by Octav Onicescu [24] as $E(X) = \sum_{k=1}^{n} p_k^2$. For a continuous random variable Y, the IE was defined by Silviu Guiaşu [15]:

$$E(Y) = \int_{-\infty}^{+\infty} p^2(\mathbf{y}) d\mathbf{y},$$

where $p(\mathbf{y})$ is the probability density function.

For a continuous random variable Y and a discrete random variable C, the conditional IE is defined as:

$$E(Y|C) = \int_{\mathbf{y}} \sum_{m=1}^{M} p(c_m) p^2(\mathbf{y}|c_m) d\mathbf{y}.$$

In order to study the interaction between two random variables X and Y, the following measure of unilateral dependency was introduced by Andonie *et al.* [3]:

$$o(Y, X) = E(Y|X) - E(Y)$$

with the following properties:

1. o is not symmetrical with respect to its arguments;
2. $o(Y, X) \geq 0$ and the equality holds iff Y and X are independent;
3. $o(Y, X) \leq 1 - E(Y)$ and the equality holds iff Y is completely dependent on X.

This measure quantifies the unilateral dependence characterizing Y with respect to X and corresponds to the amount of information detained by X about Y.

3.2 The feature weighting procedure

ESRNG is an online algorithm which adapts a set of LVQ reference vectors by minimizing the quantization error. At each iteration, it also adapts the input vector feature weights. The core of the method is based on the maximization of the $o(Y, C)$ measure.

To connect input vector $\mathbf{x}_i$ with its class j, represented by vector $\mathbf{w}_j$, we use a simple transform. We consider a continuous random variable Y with its samples $\mathbf{y}_i = \lambda \mathbf{I}(\mathbf{x}_i - \mathbf{w}_j), i = 1, \ldots, N$, where:

- λ is the vector of weights;
- $\mathbf{x}_i, i = 1, \ldots, N$, are the training vectors, each of them from one of the classes $c_1, c_2, \ldots, c_M$;
- $\mathbf{w}_j, j = 1, \ldots, P$, are the LVQ determined class prototypes.

Assuming that the M class labels are samples of a discrete random variable denoted by C, we can use gradient ascend to iteratively update the feature weights by maximizing $o(Y, C)$:

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha \sum_{i=1}^{N} \frac{\partial o(Y, C)}{\partial \mathbf{y}_i} \mathbf{I} (\mathbf{x}_i - \mathbf{w}_j) .$$

From the definition of $o(Y, X)$, we obtain:

$$o(Y, C) = E(Y|C) - E(Y) = \sum_{p=1}^{M} \frac{1}{p(c_p)} \int_{\mathbf{y}} p^2(\mathbf{y}, c_p) d\mathbf{y} - \int_{\mathbf{y}} p^2(\mathbf{y}) d\mathbf{y}. \tag{8}$$

This expression involves a considerable computational effort. Therefore, we approximate the probability densities from the integrals using the Parzen windows estimation method. The multidimensional Gaussian kernel is [13]:

$$G(\mathbf{y}, \sigma^2 \mathbf{I}) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} \cdot e^{-\frac{\mathbf{y}^t \mathbf{y}}{2\sigma^2}} \tag{9}$$

where d is the dimension of the definition space of the kernel, $\mathbf{I}$ is the identity matrix, and $\sigma^2\mathbf{I}$ is the covariance matrix.

We approximate the probability density $p(\mathbf{y})$ replacing each data sample $\mathbf{y}_i$ with a Gaussian kernel, and averaging the obtained values:

$$p(\mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} G(\mathbf{y}-\mathbf{y}_i, \sigma^2\mathbf{I}).$$

We denote by M_p the number of training samples from class c_p. We have:

$$\int_{\mathbf{y}} p^2(\mathbf{y}, c_p)d\mathbf{y} = \frac{1}{N^2}\sum_{k=1}^{M_p}\sum_{l=1}^{M_p} G(\mathbf{y}_{pk}-\mathbf{y}_{pl}, 2\sigma^2\mathbf{I})$$

and

$$\int_{\mathbf{y}} p^2(\mathbf{y})d\mathbf{y} = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N} G(\mathbf{y}_k-\mathbf{y}_l, 2\sigma^2\mathbf{I}),$$

where $\mathbf{y}_{pk}$, $\mathbf{y}_{pl}$ are two training samples from class c_p, whereas $\mathbf{y}_k$, $\mathbf{y}_l$ represent two training samples from any class.

Equation (8) can be rewritten, and we obtain the final ESRNG update formula of the feature weights:

$$\lambda^{(t+1)} = \lambda^{(t)} - \alpha\frac{1}{4\sigma^2}G(\mathbf{y}_1-\mathbf{y}_2, 2\sigma^2\mathbf{I})\cdot(\mathbf{y}_2-\mathbf{y}_1)\mathbf{I}\cdot \cdot(\mathbf{x}_1-\mathbf{w}_{j(1)}-\mathbf{x}_2+\mathbf{w}_{j(2)}),$$

where $\mathbf{w}_{j(1)}$ and $\mathbf{w}_{j(2)}$ are the closest prototypes to $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.

The ESRNG algorithm has the following general steps:

1. Update the reference vectors using the SRNG scheme.
2. Update the feature weights.
3. Repeat Steps 1 and 2, for all training set samples.

This algorithm uses a generalized Euclidean distance. The updating formula for the reference vectors can be found in [1]; we will not explicitly use this formula in the present paper.

The ESRNG algorithm generates numeric values assigned to each input feature, quantifying their importance in the classification task: the most relevant feature receives the highest numeric value. We use these factors as feature weights in the FAMRFW algorithm.

4 FAMRFW – a novel neural model

The FAMRFW is a FAMR architecture with a generalized distance measure. For an ART_a category $\mathbf{w}_j$, we define its size $s(\mathbf{w}_j)$:

$$s(\mathbf{w}_j) = n - |\mathbf{w}_j| \tag{10}$$

and the distance to a normalized input $\mathbf{A}$:

$$dis(\mathbf{A}, \mathbf{w}_j) = |\mathbf{w}_j| - |\mathbf{A}\wedge\mathbf{w}_j| = \sum_{i=1}^{n} d_{ji}, \tag{11}$$

where $(d_{j1}, \ldots, d_{jn}) = \mathbf{w}_j - \mathbf{A} \wedge \mathbf{w}_j$. In [10] it is shown that:

$$T_j(\mathbf{A}) = \frac{n - s(\mathbf{w}_j) - dis(\mathbf{A}, \mathbf{w}_j)}{n - s(\mathbf{w}_j) + \alpha_a} \tag{12}$$

$$\rho(\mathbf{A}, \mathbf{w}_J^a) = \frac{n - s(\mathbf{w}_j) - dis(\mathbf{A}, \mathbf{w}_j)}{n} \tag{13}$$

A generalization of $dis(\mathbf{A}, \mathbf{w}_j)$ is the weighted distance:

$$dis(\mathbf{A}, \mathbf{w}_j; \lambda) = \sum_{i=1}^{n} \lambda_i d_{ji}, \tag{14}$$

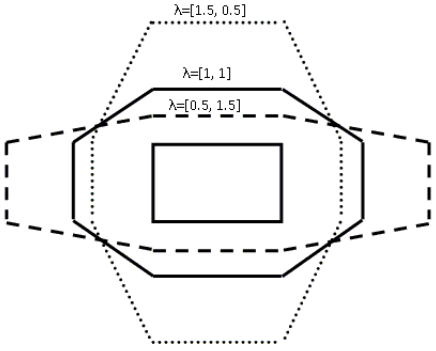
where $\lambda = (\lambda_1, \ldots, \lambda_n)$, and $\lambda_i \in [0, n]$ is the weight associated to the ith feature. We impose the constraint $|\lambda| = n$. For $\lambda_1 = \cdots = \lambda_n = 1$, we obtain in particular the FAMR.

Charalampidis *et al.* [10] used the following weighted distance:

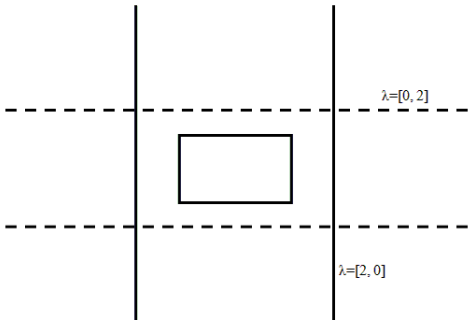
$$dis(\mathbf{x}, \mathbf{w}_j | \lambda, ref) = \sum_{i=1}^{n} \frac{(1-\lambda) l_j^{ref} + \lambda}{(1-\lambda) l_{ji} + \lambda} d_{ji}, \tag{15}$$

where l_j^{ref} is a function of category j's lengths of the hyper-rectangle, and λ is a scalar in $[0, 1]$. In our case, the function $dis(\mathbf{A}, \mathbf{w}_j; \lambda)$ does not depend on sides of the category created during learning, but on the computed feature weights. This makes our approach very different than the one in [10].

The effect of using distance $dis(\mathbf{A}, \mathbf{w}_j; \lambda)$ for a bidimensional category is depicted in Fig. 2(a). The hexagonal shapes represent the points situated at constant distance from the category. These shapes are flattened in the direction of the feature with a larger weight and elongated in the direction of the feature with a smaller weight. This is in accordance with the following intuition: The category dimension in the direction of a relevant feature should be smaller than the category dimension in the direction of a non-relevant feature. Hence, we may expect that more categories will cover the relevant directions than the non-relevant ones.



(a) Bounds for constant weighted distance $dis(\mathbf{A}, \mathbf{w}_j; \lambda)$ for various values of λ. The rectangle in the middle represents a category.



(b) Bounds for constant distance $dis(\mathbf{A}, \mathbf{w}_j; \lambda)$ for the null feature weight. The rectangle in the middle represents the category.

Figure 2: Geometric interpretation of constant distance when using $dis(\mathbf{A}, \mathbf{w}_j; \lambda)$ for bidimensional patterns.

For a null weight feature (Fig. 2(b)), the bounds are reduced to parallel lines on both sides of the rectangle representing the category. In this extreme case, the discriminative distance is the one along the remaining feature dimension. This is another major difference between our approach and the one in [10], where, while using function $dis(\mathbf{x}, \mathbf{w}_j | \lambda, ref)$, the contours of a constant weighted distance are inside

some limiting hexagons. In our method, the contour is insensitive to the actual value of the null weighted feature.

5 Experimental results

We test the FAMRFW for several standard classification tasks, all from the UCI Machine Learning Repository [5]. The experiments are performed on the FAMR and the FAMRFW architectures. The two FAMRFW stages are: *i)* the λ feature weights are obtained by the ESRNG algorithm; *ii)* these weights are used both for training and testing the FAMR.

A nice feature of the FAM architectures and the ESRNG algorithm is the on-line (incremental) learning capability, *i.e.*, the training set is processed only once. This type of learning is especially useful when dealing with very large datasets, since it can reduce significantly the computational overhead. For FAMR training and for both FAMRFW stages we use on-line learning.

5.1 Methodology

For each experiment, we use three-way data splits (*i.e.*, the available dataset is divided into training, validation, and test sets) and random subsampling. Random subsampling is a faster, simplified version of k-fold cross validation:

1. The dataset is randomized.

2. The first 60% of the dataset is used for training and the next 20% for validation (*i.e.*, for tuning the model parameters). The following parameters are optimized using a simple "grid-search" for $\rho_a, \rho_{ab} \in \{0, 0.1, \ldots, 0.9\}$ and $\beta_a \in \{0, 0.1, \ldots, 1\}$. The goal is to allow both fast learning and fast-commit slow-recode. The optimal parameter values are the ones producing the highest PCC and the lowest number of ART_a categories.

3. The network with optimal parameters is trained with the joint training + validation data.

4. The last 20% of the dataset is used for testing. As a result, the percent of correct classification (PCC) and the number of generated ART_a categories are computed.

5. Repeat this procedure six times.

The ρ_a value, optimized during training/validation, controls the number of generated ART_a categories. After training/validation, this number does not change. For $\rho_a > 0$, some test vectors may be rejected (*i.e.*, not classified).

In all our experiments, after the ART_a categories were generated, we set $\rho_a = 0$ for testing. This has the following positive effects:

- All test vectors are necessarily classified.

- We obtain experimentally better classification results, both for the FAMR and the FAMRFW, compared to the ones with optimized ρ_a values. This is shown in Table 1, for all considered classification tasks. The feature weights values in the FAMRFW are the ones mentioned in the following sections.

Table 1: Average PCC test set results using the optimized ρ_a (computed in the validation phase) vs. using $\rho_a = 0$.

	FAMR		FAMRFW	
	optimized ρ_a	$\rho_a = 0$	optimized ρ_a	$\rho_a = 0$
Breast cancer	86.54%	91.22%	91.22%	91.22%
Balance scale	75.86%	76.53%	75.92%	78.13%
Wine recognition	83.33%	84.72%	83.79%	89.35%
Ionosphere	85.44%	88.96%	85.91%	89.43%

5.2 Breast cancer classification

This dataset (formally called Wisconsin Diagnostic Breast Cancer) includes 569 instances. The instances are described by 30 real attributes. The given features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

The FAMRFW generated weights are: $[0.784, 0.816, 0.795, 2.847, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.785, 0.808, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784, 0.829, 0.828, 5.047, 0.784, 0.784, 0.784, 0.784, 0.784, 0.784]$. In Table 2, we observe that the average PCC for the FAMR and the FAMRFW is the same, but the FAMRFW has much less ART_a categories than the FAMR.

Table 2: Classification performance for the Breast Cancer Problem.

Test no.	FAMR		FAMRFW	
	No. of ART_a categories	PCC	No. of ART_a categories	PCC
1	61	93.85%	24	87.71%
2	7	90.35%	7	93.85%
3	10	95.61%	8	91.22%
4	39	85.08%	6	88.59%
5	6	92.98%	6	94.73%
6	6	89.47%	5	91.22%
Average	21.5	91.22%	9.33	91.22%

5.3 Balance scale classification

This dataset was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced. The set contains 625 patterns, with a uneven distribution of the three classes; each input pattern has 4 features.

The ESRNG generated feature weights are $\lambda = [1.002, 1.113, 0.827, 1.058]$. The FAMRFW has better classification accuracy and less ART_a categories than the FAMR (Table 3).

5.4 Wine recognition

The Wine recognition data are the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the 3 types of wines. The dataset contains 178 instances.

Table 3: Classification performance for the Balance Scale Problem.

Test no.	FAMR		FAMRFW	
	No. of ART_a categories	PCC	No. of ART_a categories	PCC
1	95	74.4%	53	75.2%
2	70	80.0%	39	80.0%
3	22	78.4%	54	81.6%
4	75	75.2%	44	85.6%
5	125	71.2%	69	72.0%
6	62	80.0%	107	74.4%
Average	74.83	76.53%	61	78.13%

The ESRNG algorithm produced the weights $\lambda = [0.900, 0.757, 0.659, 1.668, 2.349, 0.702, 1.028, 0.668, 0.774, 0.874, 0.666, 0.701, 1.253]$. The FAMRFW classification results are better, with less generated ART_a categories (Table 4).

Table 4: Classification performance for the Wine Recognition Problem.

Test no.	FAMR		FAMRFW	
	No. of ART_a categories	PCC	No. of ART_a categories	PCC
1	10	88.88%	6	86.11%
2	15	97.22%	10	97.22%
3	32	69.44%	11	86.11%
4	17	83.33%	11	86.11%
5	55	80.55%	39	94.44%
6	12	88.88%	8	86.11%
Average	23.5	84.71%	14.16	89.35%

5.5 Ionosphere

This binary classification problem starts from collected radar datasets. The data come from 16 high-frequency antennas, targeting the free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those passing through the ionosphere. There are 351 instances and each input pattern has 34 features.

The ESRNG generated λ vector is: [0.551,0.520,1.179,1.168,1.301,1.180,0.940,1.272,1.024,0.903, 0.843,0.976,0.870,0.844,0.807,0.877,0.893,1.012,0.994,1.012,0.964,1.061,1.029,1.227,0.978,1.020,0.943, 1.027,1.087,1.032,0.978,1.117,0.999,1.374]. On average, FAMRFW produced much less ART_a categories than the FAMR. This time, the FAMR produced a slightly better PCC (Table 5).

6 Conclusions

According to our experiments, using the feature relevances and the generalized distance measure may improve the classification accuracy of the FAMR algorithm. In addition, the FAMRFW uses less ART_a categories, which is an important factor. The number of categories controls the generalization

Table 5: Classification performance for the Ionosphere Problem.

Test no.	FAMR		FAMRFW	
	No. of ART_a categories	PCC	No. of ART_a categories	PCC
1	28	81.69%	8	90.14%
2	20	81.69%	8	85.91%
3	17	91.54%	7	83.09%
4	9	94.36%	8	88.73%
5	5	90.14%	5	94.36%
6	9	94.36%	5	94.36%
Average	14.66	88.96%	6.83	89.43%

capability and the computational complexity of a FAM architecture. This generalization is a trade-off between overfitting and underfitting the training data. It is good to minimize the number of categories if this does not decrease too much the classification accuracy.

The ESRNG feature weighting algorithm can be replaced by other weighting methods. We have not tested the function approximation capability of the FAMRFW neural network because the ESRNG weighting algorithm is presently restricted to classification tasks. LVQ methods can be extended to function approximation [23] and we plan to adapt the ESRNG algorithm in this sense. This would enable us to test the FAMRFW + ESRNG procedure on standard feature approximation and prediction benchmarks.

Our approach is at the intersection of two major computational paradigms:

1. Carpenter and Grossberg's adaptive resonance theory, an advanced distributed model where parallelism is intrinsic to the problem, not just a mean to speed up [6].

2. Onicescu's informational energy and the unilateral dependency measure. To the best of our knowledge, we are the only ones using Onicescu's energy in neural processing systems.

Bibliography

[1] R. Andonie and A. Caţaron. Feature ranking using supervised neural gas and informational energy. In *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN2005)*, Canada, Montreal, July 31 - August 4, 2005.

[2] R. Andonie, A. Caţaron, and L. Sasu. Fuzzy ARTMAP with feature weighting. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2008)*, Innsbruck, Austria, Febr. 11-13, 2008, 91–96.

[3] R. Andonie and F. Petrescu. Interacting systems and informational energy. *Foundation of Control Engineering*, 11, 1986, 53–59.

[4] R. Andonie and L. Sasu. Fuzzy ARTMAP with input relevances. *IEEE Transactions on Neural Networks*, 17, 2006, 929–941.

[5] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences http://www.ics.uci.edu/~mlearn/MLRepository.html

[6] I. Dziţac and B. E. Bărbat. Artificial intelligence + distributed systems = agents. *International Journal Computers, Communications, and Control*, 4, 2009, 17–26.

[7] 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 Transactions on Neural Networks*, 3, 1992, 698–713.

[8] G. A. Carpenter, B. L. Milenova, and B. W. Noeske. Distributed ARTMAP: A neural network for fast distributed supervised learning. *Neural Networks*, 11, 1998, 793–813.

[9] G. A. Carpenter and W. Ross. ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. *IEEE Transactions on Neural Networks*, 6, 1995, 805–818.

[10] D. Charalampidis, G. Anagnostopoulos, M. Georgiopoulos, and T. Kasparis. Fuzzy ART and Fuzzy ARTMAP with adaptively weighted distances. In *Proceedings of the SPIE, Applications and Science of Computational Intelligence*, Aerosense, 2002.

[11] I. Dagher, M. Georgiopoulos, G. L. Heileman, and G. Bebis. An ordering algorithm for pattern presentation in Fuzzy ARTMAP that tends to improve generalization performance. *IEEE Transactions on Neural Networks*, 10, 1999, 768–778.

[12] I. Dagher, M. Georgiopoulos, G. L. Heileman, and G. Bebis. Fuzzy ARTVar: An improved fuzzy ARTMAP algorithm. In *Proceedings IEEE World Congress Computational Intelligence WCCI'98*, Anchorage, 1998, 1688–1693.

[13] J. C. Principe *et al*. Information-theoretic learning. In S. Haykin, editor, *In Unsupervised Adaptive Filtering*. Wiley, New York, 2000.

[14] E. Gomez-Sanchez, Y. A. Dimitriadis, J. M. Cano-Izquierdo, and J. Lopez-Coronado. μARTMAP: Use of mutual information for category reduction in fuzzy ARTMAP. *IEEE Transactions on Neural Networks*, 13, 2002, 58–69.

[15] S. Guiaşu. Information theory with applications. McGraw Hill, New York, 1977.

[16] B. Hammer, D. Schunk, T. Bojer, and T. K. von Toschanowitz. Relevance determination in learning vector quantization. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001)*, Bruges, Belgium, 2001, 271–276.

[17] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21, 2005, 21–44.

[18] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15, 2002, 1059–1068.

[19] C. P. Lim and R. Harrison. ART-Based Autonomous Learning Systems: Part I - Architectures and Algorithms. In L. C. Jain, B. Lazzerini, and U. Halici, editors, *Innovations in ART Neural Networks*. Springer, 2000.

[20] C. P. Lim and R. F. Harrison. An incremental adaptive network for on-line supervised learning and probability estimation. *Neural Networks*, 10, 1997, 925–939.

[21] S. Marriott and R. F. Harrison. A modified fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural Networks*, 8, 1995, 619–641.

[22] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4, 1993, 558–569.

[23] S. Min-Kyu, J. Murata, and K. Hirasawa. Function approximation using LVQ and fuzzy sets. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, AZ, 2001, 1442–1447.

[24] O. Onicescu. Theorie de l'information. Energie informationnelle. *C. R. Acad. Sci. Paris, Ser. A–B*, 263, 1966, 841—842.

[25] O. Parsons and G. A. Carpenter. ARTMAP neural networks for information fusion and data mining: map production and target recognition methodologies. *Neural Networks*, 16, 2003, 1075–1089.

[26] M. Taghi, V. Baghmisheh, and P. Nikola. A Fast Simplified Fuzzy ARTMAP Network. *Neural Processing Letters*, 17, 2003, 273–316.

[27] S. J. Verzi, G. L. Heileman, M. Georgiopoulos, and M. J. Healy. Boosted ARTMAP. In *Proceedings IEEE World Congress Computational Intelligence WCCI'98*, 1998, 396–400.

[28] J. Williamson. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multi-dimensional maps. *Neural Networks*, 9, 1996, 881–897.