A Modified Fuzzy ARTMAP Architecture for Incremental Learning Function Approximation

Răzvan Andonie
Department of Electronics and Computers
Transylvania University of Braşov
Romania
email: andonie@deltanet.ro

Lucian Sasu
Department of Computer Science
Transylvania University of Braşov
Romania
email: lmsasu@unitbv.ro

Valeriu Beiu
School of EE and CS
Washington State University, Pullman
USA
email: vbeiu@eecs.wsu.edu

ABSTRACT
We will focus here on approximating functions that map from the vector-valued real domain to the vector-valued real range. A Fuzzy ARTMAP (FAM) architecture, called Fuzzy Artmap with Relevance factor (FAMR, defined in [1]) is considered here as an alternative to function approximation. FAMR uses a relevance factor assigned to each sample pair, proportional to the importance of the respective pair during the learning phase, and is a generalization of PROBART (a FAM architecture defined in [2]). Like other FAM–based systems, FAMR can be incrementally trained.

KEY WORDS
Fuzzy ARTMAP, function approximation, incremental learning

1 Introduction

A classical application of neural networks is the prediction of functions that are known only at a certain number of points. We will focus here on approximating functions that map from the vector-valued real domain to the vector-valued real range. Problems of this type occur almost everywhere, e.g. in the prediction of economic or social data, in weather forecasting, in signal processing, in data mining, and so on. There are many curve-fitting methods, such as splines, that will do a wonderful job. Neural networks can be also added to the arsenal of robust data-fitters. Compared to splines, neural networks may look less competitive. However, in some applications neural network models are proved to be a very good alternative. One of these applications is incremental function approximation.

In the context of supervised training, *incremental learning* means learning each input-output sample pair, without keeping it for subsequent processing. An incremental learning function approximator should be able to adapt to new information (i.e., the function value for one point) without corrupting or forgetting previously learned information– the so-called *stability-plasticity* dilemma addressed by Carpenter and Grossberg [3]. Incremental learning function approximators are very attractive in data mining applications with large data sets.

The FAM family of neural networks, having the roots in Carpenter's *et al.* seminal paper [4] is known to be one of the few models with incremental learning capability. The FAM maps subsets of R^m to R^n, accepting both binary and analogue inputs in the form of pattern pairs. Carpenter *et al.* [4] have tested the function approximation capability of FAM on a continuous sinusoidal function[4]. The approximation is incrementally improved as each new data point is added. The asymptotic accuracy of the approximation can be controlled.

The FAM paradigm is prolific and there are many variations the initial model: ART-EMAP[5], dARTMAP[6], Boosted ARTMAP[7], Fuzzy ARTVar[8], Gaussian ARTMAP[9], PROBART [2], PFAM[10, 11], μARTMAP[12]. Potentially, all can be used for function approximation. We will refer here to the function approximation results obtained using the original FAM, and PROBART.

FAMR is a FAM architecture defined [1] as an incremental learning system for general classification and estimation of the probability that an input belongs to a given class. Each training pair has a *relevance factor* assigned to it. This factor is proportional to the importance of the respective pair in the learning process. 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. The training sequence may include sample pairs from sources with different levels of noise.

We aim to analyze here the FAMR capability for function approximation tasks and compare our results to the performances using the original FAM and PROBART.

Section 2 of this paper will review the necessary FAM and PROBART details. In Section 3, we present the FAMR model. Section 4 presents the experimental results, followed by the closing remarks in Section 5.

2 FAM and PROBART

To allow comparison between FAMR, PROBART, and FAM, the following description of FAM and PROBART architectures is included here.

FAM includes a pair of ART modules (ART_{a} and ART_{b}) that create stable recognition categories in response to arbitrary sequences of input patterns. These modules are linked by an inter-ART module called Mapfield whose purpose is to determine whether the correct mapping has been established from inputs to outputs or not. The ART_{a} and ART_{b} vigilance parameters ρ_{a}, respectively ρ_{b}, control the matching mechanism inside the modules.

During learning, FAM updates its Mapfield weights to estimate the probability that an input belongs to a given output class: the strength of the weight projecting from the selected ART_{a} category to the correct ART_{b} category is increased, while the strength of the weights to other ART_{b} categories are decreased. A Mapfield vigilance parameter ρ_{ab} calibrates the degree of predictive mismatch, necessary to trigger the search for a different ART_{a} category. If the weight projecting from the active ART_{a} category through the Mapfield to the active ART_{b} category is smaller than ρ_{ab} (vigilance test), then the system responds to the unexpected outcome through the so-called *match tracking*, that triggers an ART_{a} search for a new input category.

Once an ART_{a} category J is chosen, whose prediction of the correct ART_{b} category is strong enough, match tracking is disengaged, and the network is said to be in a resonance state. In this case, Mapfield learns by updating the weights of associations between ART_{a} and ART_{b} categories.

PROBART is a modification of FAM motivated by empirical findings on the operational characteristics of FAM under certain conditions [2]. In this case, Mapfield weight w_{jk}^{ab} is the frequency of associations between the jth ART_{a} category and the kth ART_{b} category. The relative frequency $w_{jk}^{\mathrm{ab}}/|\mathbf{w}_j^{\mathrm{ab}}|$ is the empirical estimate of the posterior probability $P(k|j)$ that ART_{a} category j is associated to ART_{b} category k. There is no match tracking phase. The predicted value for an input pattern activating the Jth ART_{a} category is:

$$\mu_{Jm} = \frac{1}{|\mathbf{w}_J^{\mathrm{ab}}|} \sum_{n=1}^{N_{\mathrm{b}}} \epsilon_{nm} w_{Jn}^{\mathrm{ab}}, \quad m = 1, \ldots, 2M_{\mathrm{b}} \tag{1}$$

where μ_{Jm} is the expected value of the mth component of the predicted output pattern associated with the current input pattern, $|\mathbf{w}_J^{\mathrm{ab}}|$ is the total number of associations of Jth ART_{a} category and each category from ART_{b}, ϵ_{nm} is the mth component of the nth ART_{b} category and the Jth ART_{a} category, w_{Jn}^{ab} is the frequency of associations between Jth ART_{a} category and nth ART_{b} category and M_{b} is the dimension of the output vector.

Equation (1) can also be written as:

$$\mu_{Jm} = \sum_{n=1}^{N_{\mathrm{b}}} \epsilon_{nm} p_{Jn}, \tag{2}$$

where p_{Jn} is the relative frequency of association between the Jth ART_{a} category and the nth ART_{b} category given by $p_{Jn} = w_{Jn}^{\mathrm{ab}}/|\mathbf{w}_J^{\mathrm{ab}}|$.

3 FAMR

Fuzzy ARTMAP with Relevance factor (FAMR) was defined [1] as a general classification and posterior probability estimation tool. We will show now that FAMR can be used for function approximation as well. We need to review first the FAMR algorithm, using a slight modification that allows us to accept zero values for probabilities and make the model more general.

Let us consider a sequence of independent experiments according to the finite probability distribution $P(a_1), \ldots, P(a_n)$, where $P(a_i) \geq 0$ is the probability of outcome a_i, $\sum_{i=1}^{n} P(a_i) = 1$. These *objective probabilities* are not known and will be estimated at each step based on the previous observations. A criterion for a qualitative differentiation of the experiments is represented by the relevance associated to each experiment. The *relevance* q_t is a real positive finite number directly proportional to the importance of the experiment considered at step t ($t = 1, 2, \ldots$). This number may be either of objective or subjective nature.

The following estimation procedure makes use of both the results and the relevances of the present, and previous experiments.

The *subjective probability* of outcome a_i ($i = 1, \ldots, n$) at step t ($t = 1, 2, \ldots$) is given by:

$$w_t(a_i) = \frac{\left(q_0 w_0(a_i) + \sum_{s=1}^{t} q_s \delta_s(a_i)\right)}{Q_t} \tag{3}$$

where: if at step t we get outcome a_j, $\delta_t(a_j) = 1$ and $\delta_t(a_i) = 0$ for $j \neq i$; $w_0(a_i) \geq 0$ is the initial subjective probability, $\sum_{i=1}^{n} w_0(a_i) = 1$; $q_0 \geq 0$ is the initial relevance, and $Q_t = \sum_{s=0}^{t} q_s$.

At each step t ($t = 0, 1, \ldots$) we have a probability vector with $w_t(a_i) \geq 0$ ($i = 1, \ldots, n$), $\sum_{i=1}^{n} w_t(a_i) = 1$.

Relation (3) can be rewritten in a recursive form:

$$w_t(a_i) = w_{t-1}(a_i) + A_t \left(\delta_t(a_i) - w_{t-1}(a_i)\right) \tag{4}$$

where $A_t = q_t/Q_t$ ($t = 1, 2, \ldots$).

Let $w_t^{(n)}(a_i)$ be the subjective probabilities at step t ($t = 1, 2, \ldots$), for n possible outcomes. What is happening if at some step we get a new outcome, a_{n+1}? Assuming we have $w_0^{(n)}(a_i) = 1/n$ ($i = 1, \ldots, n$), then the new subjective probabilities $w_t^{(n+1)}(a_i)$ for $n + 1$ possible outcomes may be obtained by the following relations:

$$w_t^{(n+1)}(a_{n+1}) = q_0/(n+1)Q_t$$
$$w_t^{(n+1)}(a_i) = w_t^{(n)}(a_i) - w_t^{(n+1)}(a_{n+1})/n, i = 1, \ldots, n \quad (5)$$

Relations (5) will be used in the dynamic allocation of ART_b categories (*Step 2* in Algorithm 1.)

FAMR essentially presents a modified updating scheme of the Mapfield weights. Mapfield weight w_{jk}^{ab} can be considered an estimate of the posterior probability $P(k|j)$. This enables us to use formula (4) to update the weights w_{jk}^{ab} [1]:

$$w_{Jk}^{\mathrm{ab(new)}} = \begin{cases} w_{jk}^{\mathrm{ab(old)}} & \text{if } j \neq J \\ w_{JK}^{\mathrm{ab(old)}} + A_t(1 - w_{JK}^{\mathrm{ab(old)}}) & \\ w_{Jk}^{\mathrm{ab(old)}}(1 - A_t) & \text{if } k \neq K \end{cases} \quad (6)$$

Is w_{jk}^{ab} a good estimate of $P(I_{\mathbf{b}}|I_{\mathbf{a}})$, where $I_{\mathbf{a}}$ and $I_{\mathbf{b}}$ are intervals based around input pattern $\mathbf{a}$ and output pattern $\mathbf{b}$, respectively? As depicted by Marriott and Harrison [2] the feedback via match tracking alters this estimation. One way to avoid this problem is to eliminate match tracking. This approach is used in PROBART and ensures that a given input to ART_{a} will always select the same category.

For some very relaxed conditions imposed to the values of the relevance factor and if match tracking is not used then, for each ART_{a} category j ($j = 1, \ldots, N_{\mathrm{a}}$) and each ART_{b} category k ($k = 1, \ldots, N_{\mathrm{b}}$), w_{jk}^{ab} converges with probability one to $P(k|j)$ [1].

Match tracking can be avoided by setting $\rho_{\mathrm{ab}} = 0$. Eliminating match tracking is not always convenient, because match tracking controls category proliferation in ART_{a}. Meanwhile, it is difficult to say something about this probability approximation in the presence of match tracking, since in this case w_{jk}^{ab} is not necessarily a good estimate of the posterior probability with respect to the already processed data. In our experiments, match tracking has not significantly altered probability estimation.

Let $\mathbf{Q}$ be the vector $[Q_1 \ldots Q_{N_{\mathrm{a}}}]$. N_{a} and N_{b} are the number of categories in ART_{a}, respectively ART_{a}, initialized with 0. The procedure for incremental learning of one training pair is described as Algorithm 1.

The FAMR vigilance test is:

$$N_b \, w_{JK}^{\mathrm{ab}} \geq \rho_{\mathrm{ab}} \quad (7)$$

The relevance factor can be used as a noise filter. For instance, if we have data sources with different levels of noise, assigning to the sources relevance factors inverse proportional to the noise level, may improve the training process [1].

For $\rho_{ab} = 0$ (no match tracking), $q_0 = 0$, $q_t = q$, $0 < q < \infty$ ($t = 1, 2, \ldots$), probability estimate w_{jk}^{ab} is the relative frequency estimating the posterior probability $P(k|j)$. This can be observed from the nonrecursive formula (3). Therefore, PROBART can be considered a particular case of FAMR.

Step 1. Accept vector pair $(\mathbf{a}, \mathbf{b})$ with relevance factor q.
Step 2. If necessary, create category K in ART_{b}:
  $N_{\mathrm{b}} = N_{\mathrm{b}} + 1$
  $K = N_{\mathrm{b}}$
  if $N_{\mathrm{b}} > 1$ **then**
    $\{$append new component to $\mathbf{w}_j^{\mathrm{ab}}\}$
    $w_{jK}^{\mathrm{ab}} = \frac{q_0}{N_{\mathrm{b}} Q_j}$ for $j = 1, \ldots, N_{\mathrm{a}}$
    $\{$normalize$\}$
    $w_{jk}^{\mathrm{ab}} = w_{jk}^{\mathrm{ab}} - \frac{w_{jK}^{\mathrm{ab}}}{N_{\mathrm{b}} - 1}$ for $k = 1, \ldots, K-1$, $j = 1, \ldots N_{\mathrm{a}}$
  endif
Step 3. If necessary, create category J in ART_{a}:
  $N_{\mathrm{a}} = N_{\mathrm{a}} + 1$
  $J = N_{\mathrm{a}}$
  $\{$append new component to $\mathbf{Q}\}$
  $Q_J = q_0$
  $\{$append new line to $\mathbf{w}^{\mathrm{ab}}\}$
  $w_{Jk}^{\mathrm{ab}} = 1/N_{\mathrm{b}}$ for $k = 1, \ldots, N_{\mathrm{b}}$
Step 4. J, K are winners or new added nodes
  if vigilance test (7) is passed **then**
    $\{$learn in Mapfield$\}$
    $Q_J = Q_J + q$
    $w_{JK}^{\mathrm{ab}} = w_{JK}^{\mathrm{ab}} + \frac{q}{Q_J}(1 - w_{JK}^{\mathrm{ab}})$
    $w_{Jk}^{\mathrm{ab}} = w_{Jk}^{\mathrm{ab}} \left(1 - \frac{q}{Q_J}\right)$ for $k = 1, \ldots, N_{\mathrm{b}}$, $k \neq K$
  else
    perform match tracking and restart from step 3
  endif

Algorithm 1: **One iteration in the FAMR Mapfield algorithm.**

The initial values (probabilities and relevance) can influence the stability of the system (i.e., how fast it learns), especially for the first iterations. For relatively small training sets, this influence is significant. Since the initial probabilities in Algorithm 1 are equal, we control the influence of the initial values by setting appropriate values to q_0.

4 Experiments and Results

The experiments were performed on the function used as a benchmark in [2]. This function was used to test the mapping accuracy, to compare FAMR's ability of recomposing the underlying function whose outputs were corrupted by Gaussian noise, and to prove the effectiveness of the relevance factor.

The performance indicators used were the maximum absolute error ($MAXAE$), and the root mean square error:

$$RMSE = \sqrt{\frac{1}{N} \sum_{p=1}^{N} \|\mathbf{d}_p - \mathbf{y}_p\|^2} \quad (8)$$

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

For a better illustration of the performance, the error profile (the difference between the real and the actual output values) is plotted below the network's outputs, thus giving more detailed information in a visual form.

There are two prediction strategies one may consider. The first one is the 'weighted' prediction, where the weighted average of the output values is used, as in relation (2). The second strategy is a coarse-grained prediction: the predicted output value is the centroid of the Kth ART_{b} category having the largest value in the w_J^{ab} vector. In our tests, as one may expect, the coarse-grained strategy was always inferior to the weighted strategy. Therefore, we will refer only to results using weighted prediction.

We have used only incremental learning, though the network is able to improve its performance using off-line processing, when the training set is reprocessed. The initial relevance factor q_0 was set to 0. For our function approximation tasks, we have obtained better results than when using a positive initial relevance.

In the prediction phase, we took $\rho_{\mathrm{a}} = 0$; thus, any input pattern is assigned to an ART_{a} category and subsequently to an output vector. We have used $\rho_{\mathrm{ab}} = 0$, i.e. no match tracking.

4.1 Noise–free mapping task

The first experiment investigates the FAMR's performance compared to PROBART and FAM. We try to use a variable relevance factor, proportional to the variation of the function computed on training data. The variation of a function that exhibits slope changes can be measured by its oscillation. We have computed here the variation as the number of monotony changes within an interval of function values from training data.

For this task, we have considered function: $f : [0,1] \to [0.230, 0.771]$

$$f(x) = \frac{10+\sum_{t=1}^{7}\sin(10t\cdot x)}{20} \tag{9}$$

(Fig. 1.)

For $q_t(t = 1, 2, \ldots)$, we applied two strategies. First, we took $q_t = 1$, obtaining the particular case of PROBART, but in a different implementation. Second, q_t was computed as follows: we have divided the interval $[0, 1]$ in 10 equal nonoverlapping subintervals. For each subinterval, we assigned a q_t between 1 and 10, linearly dependent of the variation within the subinterval. For each training pair, the relevance factor was the one computed for the corresponding subinterval.

In order to compare the FAMR's performance with FAM and PROBART, we tested FAMR using 10 different noise–free training sets, and 10 data sets generated independently of the training patterns. The results (ART categories number, $MAXAE$ and $RMSE$ values) were averaged for the 10 data sets.

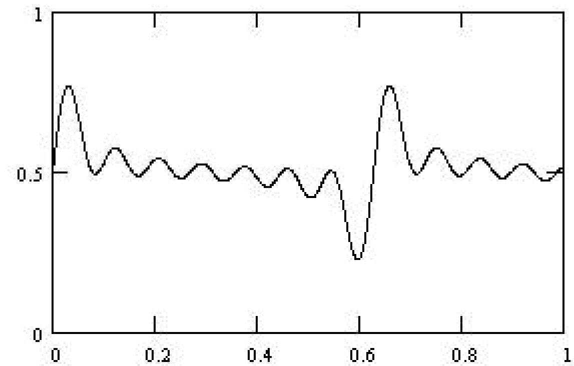

Figure 1. The noise-free training function.

Table 1. FAM, PROBART and FAMR performance with noise-free data. The network has been both trained and tested using the same noise-free data file. The vigilance parameters for FAMR were $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.992$. The graphical representations of the approximation and error profiles are given in Fig. 2.

	ART_{a} cat. no.	ART_{b} cat. no.	$RMSE$	$MAXAE$
FAM	312	53	0.0074	0.0100
PROBART	110	53	0.0169	0.0755
FAMR $q_t = 1$	105	53.6	0.0121	0.0786
FAMR $q_t \in [1, 10]$	105	53.6	0.0121	0.0786

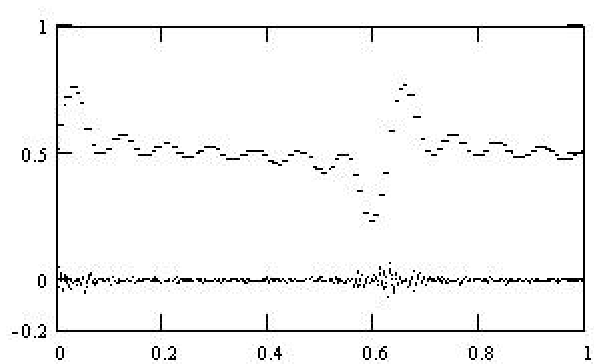

Figure 2. FAMR performance with noise–free data, for $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.99s$ and $q_t \in [1, 10]$. The lower part shows the error profile. The corresponding numerical results are given in Table 1.

In [2], the FAM and PROBART systems were compared using the following parameters: $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.99$ (for both FAM and PROBART), $\rho_{\mathrm{ab}} = 0.9$ (for FAM), and $\rho_{\mathrm{ab}} = 0$ (for PROBART.) Using these values for FAMR resulted in a considerably smaller number of ART categories. Therefore, we used slightly larger values for the two vigilance parameters: $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.992$.

In order to diminish the lack of match tracking phase in PROBART, the authors also considered a higher value for ρ_{a} and ρ_{b}: $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.999$; the same value was also considered in our FAMR tests.

The numerical results for smaller values of ρ_{a} and ρ_{b} are given in Table 1, and the estimation and error profile in Fig. 2. The results for FAM and PROBART are from [2].

From the error profile, we can conclude that for only one training epoch, the network could not accommodate sufficiently in the rapidly changing regions of f. The $RMSE$ and $MAXAE$ values were the same for the two strategies, and the $RMSE$ value was smaller for FAMR than for PROBART, for a smaller number of ART_{a} and ART_{b} categories.

The numerical results obtained for $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.999$ are reported in Table 2, and graphical representation is given in Fig. 3.

Table 2. FAMR performance with noise-free data (1000 training pairs), using increased vigilance ($\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.999$). The corresponding graphical representation is given in Fig. 3.

	ART_{a} cat. no.	ART_{b} cat. no	$RMSE$	$MAXAE$
PROBART	499	243	0.0016	0.0084
FAMR $q_t = 1$	482	219.6	0.0033	0.0335
FAMR $q_t \in [1, 10]$	482	219.6	0.0033	0.0335

We observe that using a variable relevance proportional to the variation does not influence the performance of the system. The small performance differences between PROBART and FAMR with $q_t = 1$ could be explained by the different datasets used.

4.2 The Effectiveness of the Relevance Factor

The second experiment considered the impact of the relevance factor when gaining patterns from two data sources having different levels of noise. We considered the following benchmark: 1000 training patterns were generated by two data sources, both of them adding Gaussian noise to the outputs values of the function (9); each training pattern had probability 0.5 of being generated by one of the data

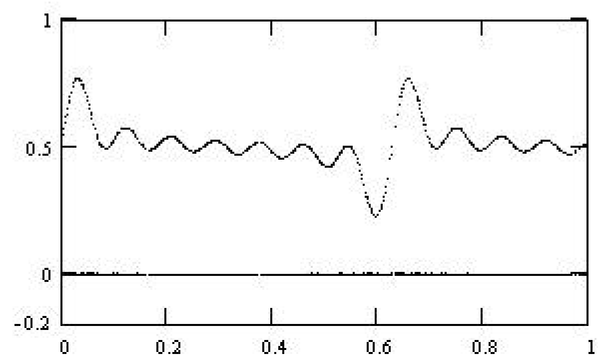

Figure 3. FAMR performance with noise-free data illustrating the effect of increased vigilance ($\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.999$ on the error profile. The relevance factor q_t belongs to $[1, 10]$. The corresponding numerical results are given in Table 2.

sources. The output pattern y_p corresponding to an input x_p was computed as follows:

$$y_p = \begin{cases} f(x_p) + 0.02 \cdot N(0,1) & \text{for the } 1^{st} \text{ data source} \\ f(x_p) + 0.1 \cdot N(0,1) & \text{for the } 2^{nd} \text{ data source} \end{cases} \tag{10}$$

where $N(0, 1)$ denotes the normal distribution of mean 0 and standard deviation 1. The two data sources had the associated relevance factors $q_t = 10$ and $q_t = 1$, respectively, for each training pattern they generated. The $RMSE$ and $MAXAE$ values obtained were 0.0206 and 0.1549, respectively (the reported results were obtained for 10 different datasets; each test set contained 1000 patterns, independently generated of the training set.)

These results were compared to the ones obtained when considering $q_t = 1$, regardless of the data source. In this latter case, the number of ART categories was the same as for the former experiment, while the numerical performance indicators were weaker: $RMSE = 0.0298$ and $MAXAE = 0.1558$. While the $MAXAE$ difference is not so high, the more sensitive indicator $RMSE$ made the difference between the two cases. The numerical results are given in Table 3. In both experiments we used $\rho_{\mathrm{a}} = \rho_{\mathrm{b}} = 0.992$.

Table 3. The effectiveness of the relevance factor. The training and test sets contained 1000 patterns, independently generated.

q_t values	ART_{a} cat. no.	ART_{b} cat. no.	$RMSE$	$MAXAE$
1,10	105	50.3	0.0206	0.1549
1	105	50.3	0.2980	0.1558

5 Conclusions

The FAMR algorithm developed here expands the range of FAMR applications by allowing incremental prediction of real functions. When the initial relevance is zero and all other relevances are constant, FAMR is equivalent to PROBART.

Using a relevance factor proportional to the variation of the function on training data does not lead to improvements of the learning phase. A better idea could be to use for each subinterval a variable number of training pairs, proportional to the variation of the function on that subinterval. However, this is only possible when we have enough training data and can use only a subset of this data.

When considering multiple data sources, correlating the relevance factor to the noise level in each data source resulted in higher performances. The relevance factor played in this case the role of a noise filter, establishing a more accurate representation of the underlying function.

As stated in the Introduction, FAM is capable of mapping subsets of R^m to R^n. FAMR is also capable of such multidimensional mappings.

References

[1] R. Andonie and L. Sasu. A Fuzzy ARTMAP Probability Estimator with Relevance Factor. In *Proceedings of the 11th European Symposium on Artificial Neural Networks (ESANN 2003)*, April 23-25, Bruges, Belgium, 2003.

[2] S. Marriott and R. F. Harrison. A Modified Fuzzy ARTMAP Architecture for the Approximation of Noisy Mappings. *Neural Networks*, 8(4):619–641, 1995.

[3] G. A. Carpenter and S. Grossberg. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *IEEE Computer*, 21(3):77–88, 1988.

[4] 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(5):698–713, 1992.

[5] 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(4):805–818, 1995.

[6] G.A. Carpenter, B.L. Milenova, and B.W. Noeske. Distributed ARTMAP: A neural network for fast distributed supervised learning. *Neural Networks*, 11(4):793–813, 1998.

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

[8] 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*, pages 1688–1693, Anchorage, 1998.

[9] J. Williamson. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 9(5):881–897, 1996.

[10] C. P. Lim and R. F. Harrison. An Incremental Adaptive Network for On-line Supervized Learning and Probability Estimation. *Neural Networks*, 10(5):925–939, 1997.

[11] C. P. Lim and R. F. 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.

[12] 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(1):58–69, 2002.