12-2012

Function Approximation with ARTMAP Architectures

Lucian M. Sasu

Răzvan Andonie

INT J COMPUT COMMUN, ISSN 1841-9836
7(5):957-967, December, 2012.

Function Approximation with ARTMAP Architectures

L.M. Sasu, R. Andonie

Lucian M. Sasu
1. Transilvania University of Braşov
Mathematics and Computers Department
Romania, 500091 Braşov, Iuliu Maniu, 50
lmsasu@unitbv.ro
2. Siemens Corporate Technology
Romania, 500096 Braşov, 15 Noiembrie, 46
E-mail: lucian.sasu@siemens.com

Răzvan Andonie
1. Computer Science Department
USA, Central Washington University, Ellensburg
400 East University Way
Ellensburg, WA 98926, USA
2. Transilvania University of Braşov
Electronics and Computers Department
Romania, 500024 Braşov, Politehnicii, 1
E-mail: andonie@cwu.edu

Abstract: We analyze function approximation (regression) capability of Fuzzy ARTMAP (FAM) architectures - well-known incremental learning neural networks. We focus especially on the universal approximation property. In our experiments, we compare the regression performance of FAM networks with other standard neural models. It is the first time that ARTMAP regression is overviewed, both from theoretical and practical points of view.

Keywords: fuzzy ARTMAP, universal approximation, regression.

1 Introduction

The approximation of functions that are known only at a certain number of discrete points is a classical application of neural networks. Almost all approximation schemes can be mapped into some kind of network that can be dubbed as a "neural network" [1]. A neural network has the *universal approximation property* if it can approximate with arbitrary accuracy an arbitrary function of a certain set of functions (usually the set of continuous function) on a compact domain. The drawback is that such an approximation may need an unbounded number of "building blocks" (i.e., fuzzy sets or hidden neurons) to achieve the prescribed accuracy. Therefore it is reasonable to make a trade-off between accuracy and the number of the building blocks, by determining the functional relationship between them.

Historically, of fundamental importance was the discovery [2] that a classical mathematical result of Kolmogorov (1957) was actually a statement that for any continuous mapping $f : [0,1]^n \subset \Re^n \longrightarrow \Re^m$ there must exist a three layered feedforward neural network of continuous type neurons that implements f exactly. This existence result was the first step. Cybenko [3] showed that any continuous function defined on a compact subset of $\Re^n$ can be approximated to any desired degree of accuracy by a feedforward neural network with one hidden layer using sigmoidal nonlinearities. Many other papers have investigated the approximation capability of three layered networks in various ways. In addition to sigmoid functions, more general functions can be used as activation functions of universal approximator feedforward networks [4].

Girosi and Poggio proved that radial basis function (RBF) networks also have universal approximation property [1]. Hartman and Kowalski [5] proved that a one hidden layer neural network with Gaussian hidden nodes is a universal approximator for real-valued maps defined on convex, compact sets of $\Re^n$. Additional related papers are [6] and [7].

The Fuzzy ARTMAP (FAM) family of neural networks is one of the best known incremental learning systems. There are many variations of Carpenter's *et al.* [8] initial FAM model, including Gaussian ARTMAP (GAM) [9], PROBART [10], FAMR [11], GART [12], [13], and AppART [14]. Compared to FAM classification, the function approximation (regression) capability of FAM was less frequently addressed. It is our goal here to discuss FAM regression capability for different FAM architectures.

The FAM maps subsets of $\Re^n$ to $\Re^m$, accepting both binary and analogue inputs in the form of pattern pairs. The initial FAM, PROBART, and the FAMR architectures have been used for incremental regression estimation. Since the initial FAM was proved to be universal approximator [15], it is reasonable to believe that members of the FAM family may also have the universal approximation capability. However, since some of the FAM variations are quite different than the initial FAM, each model should be considered individually.

The Bayesian theory allows for elaboration of general neural network training methods [16].Recently, Vigdor and Lerner have combined the Bayesian theory and the FAM introducing the Bayesian ARTMAP (BA) [17]. Like the GAM and the GART networks, during training, the BA uses Gaussian categories and FAM competitive learning. However, the BA prediction phase is very different than the FAM competitive algorithm, being a Bayesian approach. Vigdor and Lerner have compared the BA performance with respect to classification accuracy, learning curves, number of categories, sensitivity to class overlapping and risk with those of the FAM. Generally, the BA outperformed the FAM in classification tasks. Up to our contribution, the BA regression capability was not discussed or tested.

Our paper is the first overview of both theoretical and practical aspects of FAM regression, considering several major FAM architectures: the initial FAM of Carpenter *et al.*, PROBART, FAMR, BA, GAM, and AppART. We discuss universal approximation capabilities of these FAM models. In our experiments, we compare the regression performance of FAM networks with standard neural networks: Multi Layer Perceptron (MLP), RBF, General Regression Neural Network (GRNN), and FasBack. Section 2 reviews the main notations and paradigms of FAM. In Section 3, we discuss the universal approximation capability of the following FAM architectures: the original FAM, PROBART, FAMR, BA, and AppART. We synthesize our comparative experiments in Section 4. Section 5 contains the final remarks.

2 Fuzzy ARTMAP

A FAM consists of a pair of fuzzy ART modules, ART_a and ART_b, connected by an inter-ART module called Mapfield, F^{ab}. ART_a contains a preprocessing layer F_0^a, an input (or short-term memory) layer F_1^b and a competitive layer F_2^b. The following notations apply: M_a is the number of nodes in F_1^a, N_a is the number of nodes in F_2^a, and $\mathbf{w}^a$ is the weight vector between F_1^a and F_2^a. We say that a node – also called a category – from F_2^a is *uncommitted* if it has not learned yet an input pattern, and *committed* otherwise. Analogous layers and notations are used in ART_b. Each node j from F_2^a is linked to each node from F_2^b via a weight vector $\mathbf{w}_j^{ab}$ from F^{ab}, the jth row of the matrix $\mathbf{w}^{ab}$, $1 \leq j \leq N_a$. All weights are initialized to 1.

All input vectors are complement-coded by the F_0^a layer in order to avoid category proliferation [8], [18], [19]: the input vector $\mathbf{a} = (a_1, \ldots, a_n) \in [0,1]^n$ produces the normalized vector $\mathbf{A} = (a_1, \ldots, a_n, 1-a_1, \ldots, 1-a_n)$. During pattern processing, the operator $\wedge$ used is the fuzzy

AND operator defined as $(\mathbf{p} \wedge \mathbf{q})_i = \min(p_i, q_i)$, where $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{q} = (q_1, \ldots, q_n)$. $|\cdot|$ denotes the L_1 norm.

Before learning a normalized input vector $\mathbf{A}$, the vigilance parameter factor ρ_a is reset to its baseline value $\overline{\rho}_a$ and each input category is considered as not inhibited, competing for the current input pattern. A fuzzy choice function is computed for every ART_a category: $T_j(\mathbf{A}) = \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{\alpha_a + |\mathbf{w}_j^a|}$, for $1 \leq j \leq N_a$. The non-inhibited node of index J having the maximum fuzzy choice function value is further checked whether it passes the resonance condition, i.e. if the input is similar enough to the winner's prototype: $|\mathbf{A} \wedge \mathbf{w}_J^a|/|\mathbf{A}| \geq \rho_a$. If this condition is not fulfilled, then the node having index J is inhibited and another non-inhibited node maximizing the fuzzy choice function is considered as above. If no such node exists, a new node with index J is created to represent the input vector. In parallel, a similar step is performed in the ART_b module; we obtain output vector $\mathbf{y}^b = (\delta_{iK})_{1 \leq i \leq N_b}$, where K is the index of the output winner node ($1 \leq K \leq N_b$) and δ_{ij} is Kronecker's delta. If input node J is newly added, then we associate it with the current output: $\mathbf{w}_{Jk}^{ab} = \delta_{kK}$ and this association becomes permanent. Each time input node J is activated, it predicts as output value the only index k for which $w_{Jk}^{ab} = 1$. If node J is not new, then we check whether its predicted value is K. If the prediction is incorrect, a new activity (called *match tracking*) is triggered in ART_a solely. Otherwise, learning occurs in both ART_a and ART_b:

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

where $\beta_a \in (0, 1]$ is the learning rate parameter. A similar learning step takes place in ART_b.

The match tracking raises the ρ_a threshold for the current input pattern: $\rho_a = \delta + |\mathbf{A} \wedge \mathbf{w}_J^a|/|\mathbf{A}|$. If $\rho_a > 1$ then the current input pattern is rejected; otherwise, the search for an appropriate input category is continued, as described above.

For each F_2^a category we have the following geometrical interpretation. Node w_j^a is a hyperrectangle R_j inside the n-dimensional hypercube, having size $n - |w_j|$ [8]. Learning, as in equation (1), is equivalent to expanding the hyperrectangle towards the current input pattern, unless this pattern is not already in R_j. If $\beta_a = 1$, then R_j expands to $R_j \oplus \mathbf{a}$, the minimal hyperrectangle containing both R_j and input pattern $\mathbf{a}$. A similar geometrical interpretation applies to ART_b.

3 FAM Architectures used in Regression

3.1 The initial FAM for regression

The FAM regression capability was first tested by Carpenter *et al.* for univariate real functions [8]. Input categories were considered to predict not real values, but real intervals. The experiments targeted the study of predicted output intervals' geometry and the number of resulted categories for various values of ρ_b. For the test set, the authors counted the matchings between predicted output categories and actual output values. A matching between $f(a)$ and the predicted output category (a rectangle) R_K^b was established if the size of $R_K^b \oplus f(a)$ did not exceed $(1 - \rho_b)$. As expected, the number of matchings increased with ρ_b.

Verzi *et al.* [15] proved that a slightly modified FAM version can be used to universally approximate any measurable function in $L^p([0, 1])$. More specifically, given $1 \leq p < \infty$, for every $f \in L^p([0, 1])$, $f \geq 0$, a series of FAM computable functions s_n with the following property were determined: functions s_n approximate f in the limit and s_n are dense in $L^p([0, 1])$. One can extend this result to the initial FAM.

3.2 PROBART for function approximation

PROBART is a modification of FAM motivated by empirical findings on the operational characteristics of FAM under certain conditions [10]. The authors replaced the Mapfield update rule FAM by

$$\mathbf{w}_J^{ab} = \begin{cases} \mathbf{y}^b + \mathbf{w}_J^{ab} & \text{if the } J\text{-th } F_2^a \text{ node is active and } F_2^b \text{ is active} \\ \mathbf{w}_J^{ab} & \text{if the } J\text{-th } F_2^a \text{ node is active and } F_2^b \text{ is inactive} \end{cases} \tag{2}$$

Thus, w_{jk}^{ab} indicates the number of associations between the j-th ART_a node and k-th ART_b node. Initially, $w_{jk}^{ab} = 0$, i.e. no association has been made yet.

There is no match tracking phase. The predicted value for an input pattern activating the Jth ART_a category is

$$\mu_{Jl} = \frac{1}{|\mathbf{w}_J^{ab}|} \sum_{k=1}^{N_b} \epsilon_{kl} w_{Jk}^{ab}, \quad 1 \le l \le M_b \tag{3}$$

where μ_{Jl} is the expected value of the l-th component of the predicted output pattern associated with the current input pattern, $|\mathbf{w}_J^{ab}|$ is the total number of associations of the J-th ART_a category and each category from ART_b, and ϵ_{kl} represents the kth ART_b category. Specifically, for PROBART the authors considered ϵ_{kl} as the lth component of the kth ART_b category exemplar. Only the first m components of each output category w_k^b are meaningful for computing the prediction corresponding to the current input pattern.

Equation (3) can be written as $\mu_{Jl} = \sum_{k=1}^{N_b} \epsilon_{kl} p_{Jk}$, where p_{Jk} is the empirically estimated association probability between the Jth ART_a category and the kth ART_b category: $p_{Jk} = w_{Jk}^{ab}/|\mathbf{w}_J^{ab}|$.

3.3 The FAMR Model for Function Approximation

The FAMR (Fuzzy ARTMAP with Relevance factor), a version of the FAM, has a novel learning mechanism. We will review here the FAMR basic notations (details in [11]) and discuss its function approximation capabilities.

The main difference between the FAMR and the initial FAM is the update method of the w_{jk}^{ab} weights. The FAMR uses the following updating formula [11]:

$$w_{jk}^{ab(new)} = \begin{cases} w_{jk}^{ab(old)} & \text{if } j \neq J \\ w_{JK}^{ab(old)} + \frac{q_t}{Q_J^{new}}\left(1 - w_{JK}^{ab(old)}\right) & \\ w_{Jk}^{ab(old)}\left(1 - \frac{q_t}{Q_J^{new}}\right) & \text{if } k \neq K \end{cases} \tag{4}$$

where q_t is the relevance assigned to the t-th input pattern ($t = 1, 2, \dots$) 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. Initially, each Q_j ($1 \le j \le N_a$) has the same initial value q_0.

To maintain the stochastic nature of each $\mathbf{w}_j^{ab}$ row in Mapfield, we modified the Mapfield dynamics: when a new input category is created, a new row filled with $1/N_b$ is added to $\mathbf{w}^{ab}$; when a new ART_b category indexed by K is added, each existing input category is linked to it by $w_{jK}^{ab} = \frac{q_0}{N_b Q_j}$, and the rest of elements w_{jk}^{ab} are decreased by $\frac{w_{jK}^{ab}}{N_b - 1}$, for $1 \le j \le N_a$, $1 \le k \le N_b$, $k \neq K$. The update in eq. (4) preserves the stochastic property of each row. Finally, the vigilance test is changed to: $N_b\, w_{JK}^{ab} \ge \rho_{ab}$.

According to [11], this w_{jk}^{ab} approximation is a correct biased estimator of posterior probability $P(k|j)$, the probability of selecting the k-th ART_b category after having selected the j-th ART_a.

To estimate the corresponding output value for a given input pattern, FAMR uses the same formula as in eq. (3), but in this case ϵ_k contains the coordinates of the kth ART_b category centroid. During the FAMR training process, the l-th component of the centroid can be updated by Kohonen's learning rule: $\epsilon_{kl}^{b(new)} = \epsilon_{kl}^{b(old)} + (b_l - \epsilon_{kl}^{b(old)})/size_J^b$.

This rule incorporates an idea from [20]. The value $size_J^b$ is the number of output vectors of the k-th ART_b category and b_l is the l-th component of $\mathbf{b}$, the output vector of the current training pair $(\mathbf{a}, \mathbf{b})$.

3.4 The Bayesian ARTMAP Function Approximation Algorithm

In BA, in contrast to FAM, w_j^a is not a weight vector (a prototype), but simply a category label. Also, the ART categories are Gaussians, similar to the GAM. Each BA category j is characterized by the n-dimensional vector $\hat{\boldsymbol{\mu}}_j^a$ (mean), the $n \times n$ covariance matrix $\hat{\boldsymbol{\Sigma}}_j^a$, and the count number of training patterns clustered to category j, n_j^a. Analogous notations appear in ART_b, where one provides m-dimensional vectors.

The associations between input and output categories are stored inside the Mapfield module, as PROBART does, and one can approximate the conditional probability $P(w_k^b|w_j^a)$ as $\hat{P}(w_k^b|w_j^a) = w_{jk}^{ab}/\sum_{l=1}^{N_b} w_{jl}^{ab}$.

The following description uses ART_a notations; analogous notations are used for ART_b. All existent ART_a categories compete to represent the current input pattern. The posterior probability of category j given input $\mathbf{a}$ is estimated according to Bayes' theorem:

$$\hat{P}(w_j^a|\mathbf{a}) = \frac{\hat{p}(\mathbf{a}|w_j^a)\hat{P}(w_j^a)}{\sum\limits_{i=1}^{N_a} \hat{p}(\mathbf{a}|w_i^a)\hat{P}(w_i^a)} \tag{5}$$

where $\hat{P}(w_j^a)$ is the estimated prior probability of the j-th ART_a category, $\hat{P}(w_j^a) = n_j^a/\sum_{i=1}^{N_a} n_i^a$.

The conditional probability $p(\mathbf{a}|w_j^a)$ is estimated using all patterns already associated with Gaussian category w_j^a:

$$\hat{p}(\mathbf{a}|w_j^a) = \frac{1}{(2\pi)^{n/2}\left|\hat{\boldsymbol{\Sigma}}_j^a\right|^{1/2}} \cdot \exp\left\{-\frac{1}{2}(\mathbf{a} - \hat{\boldsymbol{\mu}}_j^a)^t(\hat{\boldsymbol{\Sigma}}_j^a)^{-1}(\mathbf{a} - \hat{\boldsymbol{\mu}}_j^a)\right\} \tag{6}$$

During the category choice step in ART_a, the winning category J is the one maximizing the posterior probability $\hat{P}(w_j^a|\mathbf{a})$.

The following vigilance test is performed: $S_J^a \leq S_{MAX}^a$, where $S_J^a = \left|\hat{\boldsymbol{\Sigma}}_J^a\right|$ is the hyper-volume of the winning category, and S_{MAX}^a is an upper bound threshold. During processing a training pattern, S_{MAX}^a may decrease from its initial value $\overline{S_{MAX}^a}$. In contrast, S_{MAX}^b remains unchanged. Every newly recruited category inside an ART_a (ART_b) module is centered in the current pattern and has the initial covariance matrix set to $\lambda(S_{MAX}^b)^{1/m} \cdot \mathbf{I}_m$ (and $\lambda(\overline{S_{MAX}^b})^{1/n} \cdot \mathbf{I}_n$, respectively), where λ is a small positive constant. This is done when none of the categories fulfills the vigilance test. Adding a new input (output) category triggers the addition of a new zero-filled line (column) to the association matrix $\mathbf{w}^{ab}$.

If the connection strength $\hat{P}(w_K^b|w_J^a)$ between winning categories w_J^a and w_K^b is below a fixed threshold P_{min}, then S_{MAX}^a is slightly decreased under the current winner input category's S_J, and the quest for another input category is continued. Otherwise, if the current winner input category was not newly added during processing the current pattern, ART_a learns the current pattern:

$$\hat{\boldsymbol{\mu}}^a_J(new) = \frac{n^a_J}{n^a_J+1}\hat{\boldsymbol{\mu}}^a_J(old) + \frac{1}{n^a_J+1}\mathbf{a}\ , \tag{7}$$

$$\hat{\boldsymbol{\Sigma}}^a_J(new) = \frac{n^a_J}{n^a_J+1}\hat{\boldsymbol{\Sigma}}^a_J(old) + \frac{1}{n^a_J+1}(\mathbf{a}-\hat{\boldsymbol{\mu}}^a_J(new))(\mathbf{a}-\hat{\boldsymbol{\mu}}^a_J(new))^t * I_n \tag{8}$$

$$n^a_J = n^a_J + 1 \tag{9}$$

Unless w^b_K is a newly added category for the current training pattern, an analogous learning process in ART_b takes place. Finally, the Mapfield association counter w^{ab}_{JK} is updated.

After learning, the BA can be used for prediction. We estimate the probabilistic association of an output category w^b_k with input test pattern $\mathbf{a}$:

$$\hat{P}(w^b_k|\mathbf{a}) = \frac{\sum\limits_{j=1}^{N_a}\hat{P}(w^b_k|w^a_j)\hat{p}(\mathbf{a}|w^a_j)\hat{P}(w^a_j)}{\sum\limits_{l=1}^{N_b}\sum\limits_{j=1}^{N_a}\hat{P}(w^b_l|w^a_j)\hat{p}(\mathbf{a}|w^a_j)\hat{P}(w^a_j)} \tag{10}$$

As in [21], we assume the conditional independence of activating categories w^b_k and w^a_j, given input pattern $\mathbf{a}$. For function approximation the following average formula is used:

$$\hat{f}(\mathbf{a}) = \sum_{k=1}^{N_b}\hat{P}(w^b_k|\mathbf{a})\cdot\hat{\boldsymbol{\mu}}^b_k \tag{11}$$

Since, under certain mild conditions on the kernel function, RBF networks are universal approximators [1], [5], [6], [7], and the FAM also has universal approximation capability [15], it looks natural for the BA, which is essentially a FAM architecture with Gaussian categories, to be universal approximator. However, this statement can not be directly deducted from the RBF and FAM results. This is was a good reason for us to proof the following theoretical result [22]:

Theorem 1. *BA is a universal approximator on a compact set $X \subset \Re^n$.*

3.5 AppART: Hybrid Stable Learning for Universal Function Approximation

AppART [14] is an ART-based neural network model that incrementally approximates continuous-valued multidimensional functions through a higher-order Nadaraya–Watson regression.

An input pattern $\mathbf{x}$ is feedforwarded from input layer $F1$ to the $F2$ layer. The $F2$ layer consists of N categories, modeling a local density of the input space using Gaussian receptive fields with mean $\boldsymbol{\mu_j}$ and standard deviation $\boldsymbol{\sigma_j}$. A match criterion is used to detect whether the current leaning pattern activates an existing $F2$ category or a new one should be added. The match function is:

$$G_j = \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{x_i-\mu_{ji}}{\sigma_{ji}}\right)^2\right),\ 1 \le j \le N \tag{12}$$

If all G_j values are below threshold ρ_{F2}, a new node is recruited to represent the current input pattern. Otherwise, the input strength of each $F2$ node is computed as $g_j = I(G_j > \rho_{F2}) \cdot (\eta_j G_j / \prod_{i=1}^n \sigma_{ji})$, where η_j is a measure of the prior activation probability of the jth category, and I is the binary indicator function: $I(P) = 1$ iff P is true. The activation values v_j of the $F2$ nodes are obtained by normalizing g_j. One can use v_j as an approximation of the posterior probability $P(j|\mathbf{x})$ of category j given input pattern $\mathbf{x}$.

The P and O layers together compute the prediction of the network. In the P layer, there are $m+1$ nodes whose corresponding values are computed as: $a_k = \sum_{j=1}^{N} \alpha_{kj} v_j$ $(1 \leq k \leq m)$, $b = \sum_{j=1}^{N} \beta_j v_j$ where α_{kj} and β_j are weights connecting each $F2$ category to the each node in the P layer. Each α_{kj} is the sum of values of output feature k, learned when the jth $F2$ node was active. β_j counts how many patterns the jth $F2$ category has learned. Output layer O has m output nodes, whose predictions are $o_k = I(b > 0) \cdot a_k/b$.

Incorrect predictions are detected by comparing a threshold ρ_O with the degree of closeness between the prediction of the network and the desired output. If an incorrect prediction is produced, a match tracking mechanism (similar to the one in FAM) is triggered. This might produce a new $F2$ node or find a more suitable node for the current input pattern.

The learning process takes place for μ_j, σ_j, η_j, α_j and β_j:

$$\eta_j(t+1) = \eta_j(t) + v_j,\ \mu_{ji}(t+1) = (1 - \eta_j^{-1} v_j)\mu_{ji}(t) + \eta_j^{-1} v_j x_i$$

$$\lambda_{ji}(t+1) = (1 - \eta_j^{-1} v_j)\lambda_{ji}(t) + \eta_j^{-1} v_j x_i^2,\ \sigma_{ji}(t+1) = \sqrt{\lambda_{ji}(t+1) - \mu_{ji}(t+1)^2}$$

$$\alpha_{kj}(t+1) = \alpha_{kj}(t) + \epsilon^{-1} v_j y_k,\ \beta_j(t+1) = \beta_j(t) + \epsilon^{-1} v_j$$

A common value $\gamma_i = \gamma_{common}$ may be used for the standard deviation in case of all input features.

An important theoretical result of AppART is [14]:

Theorem 2. *AppART with $\rho_{F2} = 0$, $\rho_O = 0$ and $\gamma_i = \gamma_{common}$, $1 \leq i \leq n$ behaves as GRNN.*

Since the GRNN can be viewed as a normalized RBF expansion, one can transitively apply to AppART two important properties of RBF networks: the universal approximation and the best approximation properties [1].

4 Experimental Results

For the first test, we consider function [10] $f : [0,1] \to [0,1]$ defined by $f(x) = (10 + \sum_{t=1}^{7} \sin(10tx))/20$.

We use independent, randomly generated datasets for training, validation and testing, consisting of 800, 200, and 1000 patterns, respectively. Each training pattern is a $(x, f(x))$ input-output pair. The testing set was not used in the training phase, but only to assess generalization performance.

The BA parameters $\overline{S^a_{MAX}}$, S^b_{MAX}, and P_{min} are optimized on the validation set by trial and error, for $\overline{S^a_{MAX}}, S^b_{MAX} \in \{10^{-3}, 5{\cdot}10^{-4}, 10^{-4}, 5{\cdot}10^{-5}, 10^{-5}, 5{\cdot}10^{-6}\}$ and $P_{min} \in \{0, 0.1, \ldots, 0.9\}$.

The BA with optimized parameters (i.e., generating the lowest RMSE on the validation set) was trained on the training+validation dataset. The generalization performance of the trained BA was assessed on the testing set in two ways (see Table 4):

1. The "BA(1)", corresponds to a BA network with unbounded number of categories.

2. For "BA(2)", we considered only BA models with similar number of input categories as for PROBART.

	ART_{a} categories no.	ART_{b} categories no.	$RMSE$
FAM	312	53	0.0074
PROBART	110	53	0.0169
BA(1)	185.6	57.8	0.0076
BA(2)	111.0	35.8	0.0106

Table 1: FAM, PROBART, and BA performance for regression on data generated by function f.

The RMSE for BA(1) and BA(2) were each averaged for five different runs, using each time randomly generated training, validation, and test sets. The results for PROBART and FAM are from [10]. FAM in our experiments is Carpenter's initial FAM version.

The BA(1) results are very similar to the FAM results, but for a considerably smaller number of input categories. On average, BA(2) produced one more input category than PROBART, while improving the RMSE by 40.23%. It is quite difficult to directly compare the resulted BA(2) and FAM, since BA(2) has 64.42% less input categories than the FAM.

Considering both the RMSE score and the number of input categories, we may conclude that, for this experiment, the BA performs better than the FAM and PROBART.

In the second test, we use the fifth-order chirp function [14]: $g(x) = 0.5+0.5\sin(40\pi x^5)$. Marti *et al.* have experimentally compared the function approximation performance of the following neural models [14]: AppART, Multi Layer Perceptron (MLP), RBF, General Regression Neural Network (GRNN), FAM, GAM, PROBART, and FasBack [23]. The reported score was the mean squared error (MSE). The authors run the training algorithms for several epochs. The data set consisted of 10000 points $x \in [0, 1]$, of which 70% were used for training and the rest for testing. The cited paper does not fully describe the parameter values used for each of the networks.

In our experiment, we partition a dataset of 10000 patterns into a 4000 patterns training set, a 3000 validation set, and a 3000 patterns testing set. We perform a trial and error search for $\overline{S^a_{MAX}}, S^b_{MAX} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, $P_{min} \in \{0, 0.1, \ldots, 0.9\}$. The values producing the best MSE on the validation set are used to train the BA on the train+validation dataset, and the testing set MSE was reported. The above procedure are repeated five times, for randomly generated datasets. We only use single epoch training.

Table 2 contains the results for MLP, RBF, GRNN, FAM, GAM, PROBART, FasBack, AppART and BA. For the first eight neural networks the results are from [14]. BA produces a very good MSE score for this regression task, most likely due to the optimized parameter values obtained by trial and error.

Comparing the MSE BA score, obtained by single epoch training, and those reported in [14], where multi-epoch training was used, we can state that the BA clearly performs better.

5 Conclusions

Theoretical universal approximation results were obtained for several FAM architectures:

- Explicit results were obtained for a slight variation of the initial FAM and for the BA.
- Implicit results, derived by association with other networks: FAMR, PROBART, and AppART.

Model	MSE	Training epochs
MLP	0.4362	30000+
RBF	0.2701	10000
GRNN	0.1540	150
FAM	0.1802	140
GAM	0.1521	45
PROBART	0.1435	50
FasBack	0.0915	10000
AppART	0.0803	30
BA	0.0086	1

Table 2: BA vs. other neural networks generalization performance for data generated by function g.

The result showing FAM networks to be universal approximators is an important fact in establishing the utility of FAM architectures. A learning algorithm which is known to be a universal approximator can he applied to a large class of interesting problems with the confidence that a solution is at least theoretically available. Experimentally, FAM architectures performed well compared to other neural function approximators.

The FAM model, as well as other universal approximators, suffer from the curse of dimensionality, as defined by Bellman [24]: an exponentially large number of ART categories may be required to reach a final solution. Therefore, the universal approximation capability of a network is an generally an existential result, not a constructive procedure to obtain a guaranteed compact network approximation of an arbitrary function. An important problem we have not addressed here is that of determining the network parameters so that a prescribed degree of approximation is achieved (see [25]).

The FAM and its offsprings are incremental learning models. Therefore, they may be used for fast approximation of massive streaming input data. This may be a serious plus when compared to other neural predictors.

How could a neural posterior probability estimator, like the BA, be used in risk assessment and decision theory? One possibility would be to combine the inferred posterior probabilities with a loss function, as suggested for a more general framework in [26]. This way, we could obtain an incremental learning risk assessment tool capable of processing fast large amounts of data.

Bibliography

[1] Girosi, F.; Poggio, T. (1989); Networks and the Best Approximation Property, *Biological Cybernetics*, 63: 169-176.

[2] Hecht-Nielsen, R. (1987); Kolmogorov's mapping neural network existence theorem, *Proceedings of IEEE First Annual International Conference on Neural Networks*, 3: III-11–III-14.

[3] Cybenko, G. (1992); Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems*, 5(4): 455-455.

[4] Chen, T.; Chen, H. (1995); Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Transactions on Neural Networks*, 6(4): 911-917.

[5] Hartman, E; Keeler, J.D.; Kowalski, J.M. (1990); Layered neural networks with Gaussian hidden units as universal approximations, *Neural Computations*, 2(2): 210-215.

[6] Park, J.; Sandberg, I.W. (1991); *Neural Computations*, 3(2): 246-257.

[7] Park, J.; Sandberg, I.W. (1993); *Neural Computations*, Approximation and radial-basis-function networks, 5(2): 305-316.

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

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

[10] Marriott, S.; Harrison, R.F. (1995); *Neural Networks*, A modified fuzzy ARTMAP architecture for the approximation of noisy mappings, 8(4): 619-641.

[11] Andonie, R.; Sasu, L. (2006); *IEEE Transactions on Neural Networks*, Fuzzy ARTMAP with Input Relevances, 17: 929-941.

[12] Yap, K.S.; Lim, C.P. Abidi, I.Z. (2008); *IEEE Transactions on Neural Networks*, A Hybrid ART-GRNN Online Learning Neural Network With a ε-Insensitive Loss Function, 19: 1641–1646.

[13] Yap, K.S.; Lim, C.P. Junita, M.S. (2010); *Journal of Intelligen & Fuzzy Systems*, An enhanced generalized adaptive resonance theory neural network and its application to medical pattern classification, 21: 65-78.

[14] Marti, L.; Policriti, A.; Garcia, L. (2002); *Hybrid Information Systems, First International Workshop on Hybrid Intelligent Systems, Adelaide, Australia, December 11-12, 2001, Proceedings*, AppART: An ART Hybrid Stable Learning Neural Network for Universal Function Approximation, 93-119.

[15] Verzi, S.J.; Heileman, G.L.; Georgiopoulos, M.; Anagnostopoulos, G.C. (2003); *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2003)*, Universal Approximation with Fuzzy ART and Fuzzy ARTMAP, (3): 1987-1992.

[16] MacKay, D.J.C. (1996); *Computation in Neural Systems*, Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks, 6: 469 - 505.

[17] Vigdor, B.; Lerner, B. (2007); *IEEE Transactions on Neural Networks*, The Bayesian ARTMAP, 18: 1628-1644.

[18] Moore, B. (1988); *Proceedings of the 1988 Connectionist Model Summer School*, ART1 and Pattern Clustering, 174-185.

[19] Carpenter, G.A.; Grossberg, S.; Reynolds, J.H. (1991); *Neural Networks*, Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system, (4): 759-771.

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

[21] Lerner, B.; Guterman, H. (2008); *Computational Intelligence Paradigms - Studies in Computational Intelligence, Springer*, Advanced Developments and Applications of the Fuzzy ARTMAP Neural Network in Pattern Classification, 137: 77-107.

[22] Sasu, L; Andonie, R. (2012); The Bayesian ARTMAP for Regression, *under review.*

[23] Izquierdo, J.M.C.; Dimitriadis, Y.A.; Coronado, J.L. (1997); *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, FasBack: matching-error based learning for automatic generation of fuzzy logic systems, 3: 1561 -1566.

[24] Bellman, R.E. (1961), *Rand Corporation Research studies*, Adaptive control processes: a guided tour.

[25] Andonie, R. (1997); *Dealing with Complexity: A Neural Network Approach*, The Psychological Limits of Neural Computation, 252-263.

[26] Duda, R.O.; Hart, P.E.; David G.S (2000); Pattern Classification, 2nd edition.