Combining GAs And RBF Neural Networks for Fuzzy Rule Extraction from Numerical Data

Manolis Wallace[1,2], and Nicolas Tsapatsoulis[2,3]

[1] University of Indianapolis, Athens Campus, 9 Ipitou Str., Syntagma
105 57 Athens, Greece
wallace@uindy.gr
[2] School of Electrical and Computer Engineering, National Technical University of Athens,
9 Iroon Polytechniou Str., Zographou, 157 73, Athens, Greece
{wallace, ntsap}@image.ntua.gr
[3] Dept. of Computer Science, University of Cyprus, 75 Kallipoleos Str.
P.O. Box 20537, CY-1678, Nicosia, CYPRUS
nicolast@ucy.ac.cy

Abstract. The idea of using RBF neural networks for fuzzy rule extraction from numerical data is not new. The structure of this kind of architectures, which supports clustering of data samples, is favorable for considering clusters as if-then rules. However, in order for real if-then rules to be derived, proper antecedent parts for each cluster need to be constructed by selecting the appropriate subspace of input space that best matches each cluster's properties. In this paper we address the problem of antecedent part construction by (a) initializing the hidden layer of an RBF-Resource Allocating Network using an unsupervised clustering technique whose metric is based on input dimensions that best relate the data samples in a cluster, and (b) by pruning input connections to hidden nodes in a per node basis, using an innovative Genetic Algorithm optimization scheme.

1 Introduction

Extracting if-then rules from numerical data using an RBF neural network can be achieved in the following framework: (a) The RBF-hidden nodes combine inputs in an AND form creating the antecedent part of the rule; that is rule antecedents are considered the input to hidden connections, (b) output nodes combine the outputs of the hidden nodes in an OR form; that is the rule consequents are the hidden to output connections, (c) knowledge in the form of if-then rules can be derived from clustering numeric data, (d) fuzziness is achieved both in the hidden and the output nodes forcing the activation and final output to be in the interval [0 1] instead of having crisp values.

Although the above framework seems reasonable there are two important problems: (i) All inputs are used in the antecedent part of the rule; this leads to inefficiency in creating real linguistic rules especially in cases where the input dimension is relatively high and (ii) the classic clustering approach used in RBF neural networks does not account for specifying different weights for the various input dimensions. While for the second problem one could consider the use of a different metric for creating a more "rule-like" clustering, the first problem is not that easy to solve. In

this paper we address both problems by: (1) using Genetic Algorithms for selecting the appropriate inputs for each hidden node separately; this is radically different from the classic combination of RBF and GAs which focus in feature selection for the whole network [1][2][3] and (2) applying an unsupervised clustering technique, that is based on a data dependent metric, for initializing the parameters of the hidden nodes in the RBF network. In the proposed method clusters are created by data samples that are based on these dimensions that relate them best. This is clearly a more "rule-like" approach that the classic unsupervised clustering methods. The modifications proposed above are applied on an modified Resource Allocation Network consisting of RBF hidden nodes to account for learning required for rule extraction from numerical data.

2 Preliminaries

One approach for extracting rules from numerical data is to apply a supervised training procedure on a domain ***D*** from which the learner has access to a representative example set ***E*** of pairs $\{ \underline{x}, \underline{d}(\underline{x}) \}$ (numerical data), $\underline{x} \in \Re^n$, $\underline{d} \in \Re^m$. By the end of learning, performed on set ***E***, a set of parameters ***G*** (typically represented as matrices) that model the function $g : \Re^n \to \Re^p$, is available so that $\left\| G(\underline{x}) - g(\underline{x}) \right\| < \varepsilon$, ε>0.

In the proposed method the set of parameters consists of four matrices corresponding to the mean vectors (matrix **M)** and spreads (matrix **Σ**) of the hidden RBF nodes, to the association of the hidden nodes to output classes (matrix **W**), and to the association of input dimensions to hidden nodes (matrix **A**), i.e., which subspace of input space need to be considered for each hidden node for maximum performance in clustering. The values of matrices **M, Σ** and **W** are estimated during the formal training of the RBF network (see Section 2.2) while the matrix **A** is computed by applying GAs optimization to the (already) trained RBF network (see Section 3).

2.1 Unsupervised clustering of high dimensional data

In this work, we extend the classic agglomerative clustering algorithm in order to incorporate soft feature selection in the inter cluster distance estimation process, thus providing an output that is more effective (better results) and more efficient (faster convergence) for initializing the network. Let c_1 and c_2 be two clusters of data samples. Let also r_i, $i \in \{1..F\}$ be a distance metric defined in space $\mathbb{R}^{S_i} \subseteq \mathbb{R}^S$, F the count of distinct metrics that may be defined among a pair of clusters, S the count of features for the data samples and S_i the count of features considered by the i-th sample-to-sample distance metric. A distance metric between the two clusters, when considering the i-th sample-to-sample distance metric, is given by $f_i(c_1, c_2) = \sqrt[\kappa]{\dfrac{\sum_{a \in c_1, b \in c_2} \left(r_i(\underline{a_i}, \underline{b_i}) \right)^\kappa}{|c_1| \cdot |c_2|}}$ where $\underline{a_i}$, $\underline{b_i}$ are the positions of data samples a and b

in feature space $\mathbb{R}^{S_i}$, $|c_1|$, $|c_2|$ are the cardinalities of clusters c_1 and c_2 respectively and $\kappa \in \mathbb{R}$ is a constant. The "context" is a selection of features that should be considered when calculating an overall distance value; we define it as a vector $\underline{ctx} \in \mathbb{R}_+^F$ with $\sum_{i=1}^{F} ctx_i = 1$. Given a context, the overall distance between clusters c_1 and c_2 is calculated as $f^*(c_1,c_2) = \sum_{i=1}^{F} \left(ctx_i(c_1,c_2)\right)^{\lambda} \cdot f_i(c_1,c_2)$. In the non-trivial cases the optimal context is provided by $ctx_i(c_1,c_2) = ctx_F(c_1,c_2) \cdot \left(\frac{f_F(c_1,c_2)}{f_i(c_1,c_2)}\right)^{\frac{1}{\lambda-1}}$, $\forall i \in \{1..F-1\}$ and $ctx_F(c_1,c_2) = \frac{1}{\sum_{i=1}^{F} \left(\frac{f_F(c_1,c_2)}{f_i(c_1,c_2)}\right)^{\frac{1}{\lambda-1}}}$. For the sake of space, the proof for this is omitted and the reader is directed to [4] for more details on the proposed extension to the agglomeration process.

2.2 RBF network initialization and training

Learning is incorporated into the network using the gradient descent method, while a squared error criterion is used for network training. The squared error $e(t)$ at iteration t is computed in the standard way: $e(t) = \frac{1}{2}\sum_{k=1}^{p} \left(d_k(t) - y_k(t)\right)^2$ where $d_k(t)$ is the desired output and $y_k(t)$ is the output of neuron k given by $y_k(t) = \frac{1-e^{2z_k}}{1+e^{2z_k}}$, $z_k = \left(\underline{w_k}\right)^T \cdot \underline{\phi(t)}$ where $\underline{w_k} = \left[w_{k1}, w_{k2} ... w_{kq(t)}\right]^T$ are the weights connecting the RBF hidden neurons with the output neurons (note that these parameters are constrained to have binary values so as to better accommodate the extraction of if-then rules) and $\underline{\phi(t)}$ is the output of the hidden layer. For the sake of space, the reader is directed to [6] for more details on the training of the RBF network.

3. Derivation of the antecedent part of if-then rules using Genetic Algorithms

The last step for the creation of if-then rules is the derivation of the antecedent part through the estimation of matrix **A.** The initialization of the hidden layer based on the results of the clustering method, described in Section 3, supports the construction of clusters with similarities in subspaces of the input space. However, when the training of the RBF network concludes all inputs are connected to all hidden neurons, thus all

values of **A** matrix are set to one. In order to construct a proper antecedent part several input connections to hidden neurons need to removed. Moreover, these connections need, in general, to be different for each hidden node so as allow us to consider each hidden neuron as an if-then rule. In order to accommodate the above requirement a genetic algorithm optimization procedure is followed as described below.

Let ϕ_i be the activation of *i*-th hidden neuron and **I** be the set of data samples of training set **E** (which in addition to **I** contains the corresponding target vectors). Let also S_i be the subset of **I** ($S_i \subset I$) such that every data sample belonging to it ($\forall \underline{x} \in S_i$) activates the most the *i*-th hidden neuron. The aim of the training is to find a string that optimizes the activation ϕ_i over set S_i. For this purpose a genetic algorithm (GA) optimization scheme is used. We utilize a "per rule" feature selection methodology for the construction of the antecedent part of the rules. The coding that has been selected models the presence or absence of the corresponding input dimension in the antecedent part of a rule as 1 or 0 respectively. The fitness function *F* that is used is given by $F(S_i) = \frac{1}{card(S_i)} \sum_{\underline{x} \in S_i} \phi_i(\underline{x})$ where *card(Si)* is the cardinality of set S_i and $\phi_i(\underline{x})$ is the activation of the *i*-th hidden neuron when fed by the input vector $\underline{x}$. The objective is to find the binary string that maximizes the fitness function $F(S_i)$. The realization of the genetic operators reproduction, mutation and crossover is as follows: ***Reproduction***. The fitness function $F(S_i)$ is used in the classical "roulette" wheel reproduction operator that gives higher probability of reproduction to the strings with better fitness according to the following procedure: i) an order number, q, is assigned to the population strings. That is q ranges from 1 to Np, where Np is the size of population, ii) the sum of fitness values (Fsum) of all strings in the population is calculated, iii) the interval [0 Fsum] is divided into Np sub-intervals each of one being, iv) a random real number R0 lying in the interval [0 Fsum] is selected, v) the string having the same order number as the subinterval of R0 is selected and vi) steps (4) and (5) are repeated Np times in order to produce the intermediate population to which the other genetic operators will be applied. ***Crossover***. Given two strings of length *k* (parents) an integer number $r \in \aleph_k$ is randomly selected. The two strings retain their gene values up to gene *r* and interchange the values of the remaining genes creating two new strings (offspring). ***Mutation***. This operator is applied to each gene of a string and it alters its content, with a small probability.

4. Experimental results

The *iris* data were used to validate the proposed method as far as the rule extraction efficiency and the classification performance is concerned. When trained with the iris data the proposed combination of RBF-RAN and GAs creates three rules and achieves an overall classification performance of 96.7%. The estimated matrices are

given below:

$$\mathbf{M}=\begin{bmatrix}0.66 & 0.59 & 0.50\\ 0.30 & 0.28 & 0.34\\ 0.56 & 0.43 & 0.15\\ 0.20 & 0.13 & 0.03\end{bmatrix},\ \mathbf{\Sigma}=\begin{bmatrix}0.13 & 0.10 & 0.07\\ 0.06 & 0.06 & 0.08\\ 0.11 & 0.10 & 0.04\\ 0.05 & 0.04 & 0.02\end{bmatrix}\ \mathbf{A}=\begin{bmatrix}0 & 0 & 1 & 1\\ 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0\end{bmatrix},\ \mathbf{W}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

We should note that the pruning of input to hidden nodes connections due to the GA optimization has no degradation effect on the classification performance which remains 96.7%. Comparisons with other methods (with the help of [5]), as far is the classification performance (resubstitution accuracy), are given in Table I. We observe that the proposed method (RBF-GAs) outperforms all listed soft computing techniques, with the exception of FuGeNeSys, creating as few as three rules. More results from the application of the proposed methodology are available and equally promising, but are omitted for the sake of space.

Table 1. Comparison of RBF-GAs with other techniques for Iris Data Classification

Method	Rules	Resubstitution Accuracy (%)
FuGeNeSys	5	100
NEFCLASS	7	96.7
ReFuNN	9	95.3
EFuNN	17	95.3
FuNe-I	7	96.0
RBF-Gas	3	96.7

5. Conclusions

In this paper we propose an innovative hybrid architecture, which combines resource allocating properties, novel clustering techniques for multi-dimensional problems and evolutionary weight connection purging in order to incorporate (a) fast classifying capabilities, (b) expert knowledge modeling, (c) knowledge extraction from numerical data. The proposed approach embeds rule-based knowledge directly into its architecture while its resource allocating structure enables new rules to be created. The latter is very important for two reasons: (a) there are several domains in which no estimation about the number of rules that are required to solve a particular problem is available, (b) rules can be created to model a changing of a context.

References

1. J. Yang, V. Honavar, "Feature Subset Selection Using A Genetic Algorithm," ACM computing 1991
2. D. Addison, S. Wermter, G. Arevian, "A Comparison of Feature Extraction and Selection Techniques," *Proceedings of the International Conference on Artificial Neural Networks (ICANN'03),* Istanbul, Turkey, Supplementary Proceedings pp. 212-215, June 2003.

3. Growing Compact RBF Networks Using a Genetic Algorithm," VII Brazilian Symposium on Neural Networks (SBRN'02) November 2002
4. M. Wallace, S. Kollias, "Robust, Generalized, Quick and Efficient Agglomerative Clustering" Proceedings of 6th International Conference on Enterprise Information Systems (ICEIS), Porto, Portugal, April 2004.
5. L. I. Kuncheva and J. C. Bezdek, "Nearest prototype classification: Clustering, genetic algorithms, or random search?" *IEEE Trans. Syst.*
6. Wallace M., Tsapatsoulis N., Kollias S. "`Intelligent Initialization of Resource Allocating RBF Networks" Neural Networks, 2005

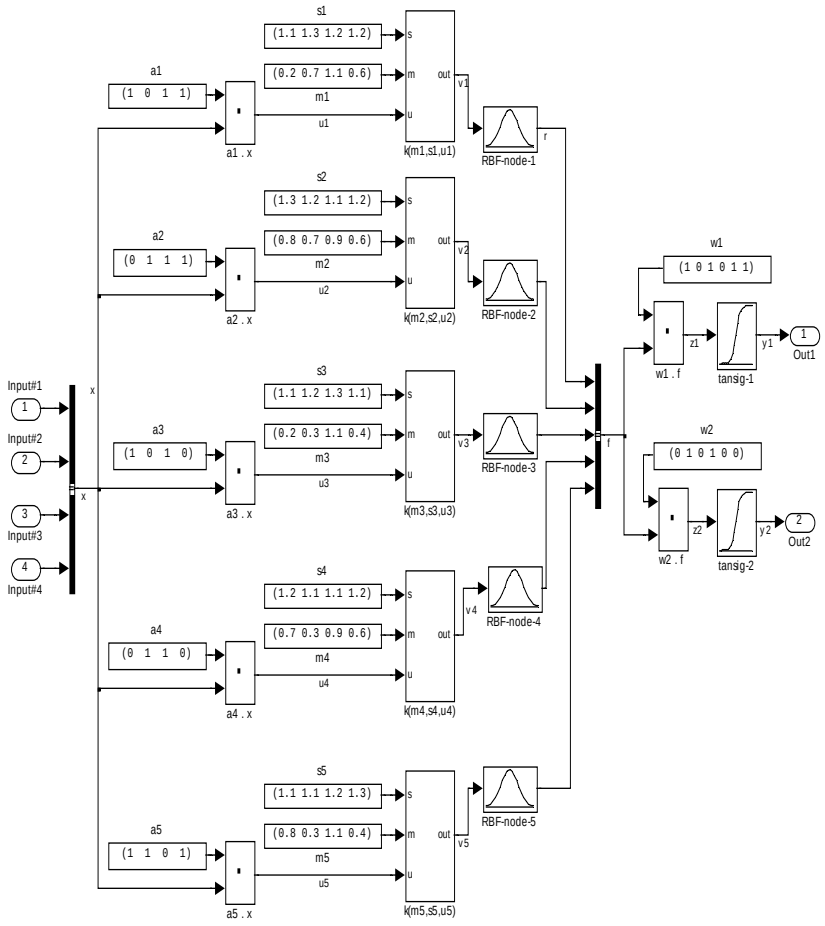

Fig. 1. The proposed RBF architecture for rule extraction