

An Adaptive Resource Allocating Neuro-Fuzzy Inference System with Sensitivity Analysis Resource Control

Minas Pertselakis, Natali Raouzaïou and Andreas Stafylopatis

National Technical University of Athens
School of Electrical and Computer Engineering
9, Iroon Polytechniou Str., Zografou, 157 80, Athens, Greece

Abstract Adaptability in non-stationary contexts is a very important property and a constant desire for modern intelligent systems and is usually associated with dynamic system behaviors. In this framework, we present a novel methodology of dynamic resource control and optimization for neurofuzzy inference systems. Our approach involves a neurofuzzy model with structural learning capabilities that adds rule nodes when necessary during the training phase. Sensitivity analysis is then applied to the trained network so as to evaluate the network rules and control their usage in a dynamic manner based on a confidence threshold. Therefore, on one hand, we result in a well-balanced structure with an improved adaptive behavior and, on the other hand, we propose a way to control and restrict the “curse of dimensionality”. The experimental results on a number of classification problems prove clearly the strengths and benefits of this approach.

1 Introduction

The guiding principle of soft computing is to exploit the tolerance for imprecision by devising methods of computation that lead to an acceptable solution at low cost [1]. The presented paper, following the same philosophy, proposes a methodology that aims to reduce complexity as well as computational cost by applying a novel resource control-via-evaluation technique. This technique leads to a dynamic and robust network structure that abides by the demands of real time operation in non-stationary environments and by modern requirements concerning energy saving.

Structure identification in neurofuzzy modeling is a relatively underestimated field, in contrast to system parameters adjustment [2]. In most system design approaches, the structure, which usually implies the number of input and rule nodes, is presumed and only parameter identification is performed to obtain the coeffi-

icients (e.g. weights) of the functional system. Even though the literature offers some heuristic but practical and systematic methodologies, the problem of structure determination in fuzzy modeling is yet to be solved [3]. There are many issues in practice that remain to be addressed; the “curse of dimensionality”, which leads to high computational cost, and finding the optimum number of rules being the most significant.

A common approach which addresses these problems is node pruning. Node pruning aims to simplify a network by keeping only those nodes that contribute the most to the general solution of the problem at hand. Various such methods have been proposed in literature based on genetic algorithms, reinforcement learning or other soft computing techniques [4-5]. However, more often than not, most of these concepts eliminate the less desired nodes from the network (either input or rule nodes). This action usually leads to reduced computational cost but, on the other hand, it sacrifices bits of information that can affect performance [6].

Our methodology attempts to tackle the problem from a different perspective. On one hand, we propose an updated version of a resource allocating neurofuzzy system [7], which expands its rule base dynamically during training to accommodate more rules to capture the given problem efficiently. On the other hand, we apply a novel sensitivity analysis technique which aims to control and restrict rule usage in real-time. Therefore, we combine efficiently two techniques with opposing philosophies to exploit their advantages and eliminate their individual drawbacks through a hybrid product. The outcome is a well-balanced and possibly optimal rule structure. Experiments on various benchmark classification tasks show that this dynamic and robust approach offers almost the same accuracy as a complete rule base, but with significantly lower amount of computations and less execution time.

The paper is divided in the following sections: Section 2 summarizes the architecture and functionality of the resource allocating subsethood-product neurofuzzy model we employ, whereas in Section 3 we present our proposed methodology of rule base usage control, which relies on the statistical tool of sensitivity analysis. Experimental results can be found in Section 4, while useful deductions and plans for future research conclude the paper in Section 5.

2 Adaptive Resource Allocating Neuro-Fuzzy Inference System

Resource Allocating Network (RAN) architectures [8], were found to be suitable for online modeling of non-stationary processes. The original resource allocating system involved a sequential learning method where the network initially contained no hidden nodes. On incoming training examples, based on certain criteria, the RAN either grew to accommodate more nodes or the existing network parameters were adjusted using a least-mean-square gradient descent learning method.

In our case, a modified RAN methodology is combined with a modern fuzzy neural inference system, named ASuPFuNIS, which has been proven to be a universal approximator [9]. The produced hybrid, ARANFIS, maintains the original functional advantages of its predecessor, while, at the same time, exhibits data-driven knowledge extraction and online adaptation with a resource allocating architecture. During the sequential learning procedure, more hidden nodes can be added in case the few initial ones cannot represent the numerical data.

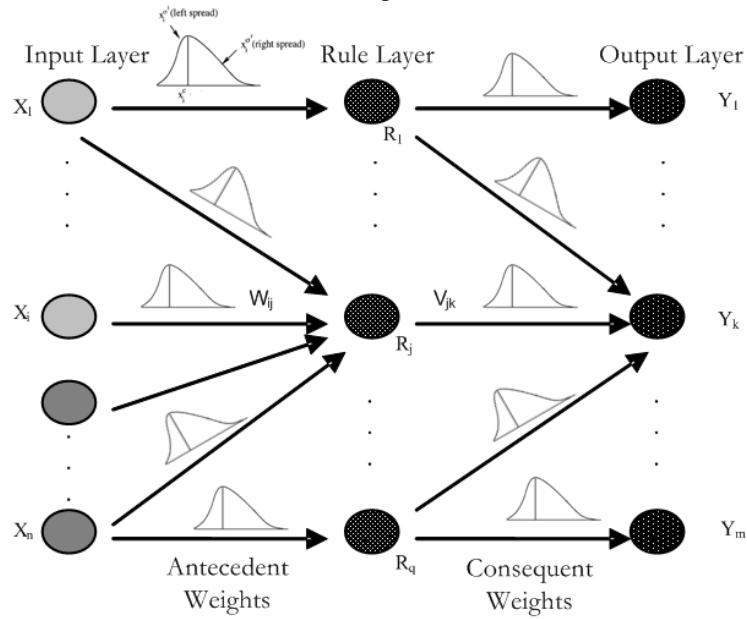


Fig. 1. ARANFIS architecture

ARANFIS architecture, with n input nodes, q hidden nodes and m output nodes, is shown in Fig. 1. Fuzzy antecedent weights are modeled by the center, left spread and right spread of an asymmetric Gaussian fuzzy set, and are denoted by $w_{ij} = (w_{ij}^c, w_{ij}^l, w_{ij}^r)$. In a similar fashion, consequent fuzzy weights from rule node j to output node k are denoted by $v_{jk} = (v_{jk}^c, v_{jk}^l, v_{jk}^r)$. The net activation (or rule strength) z_j of rule node j is a product of all mutual subsethoods that fan-in to node j . The mutual subsethood $E(A,B)$ measures the degree to which a fuzzy set A equals a fuzzy set B .

For more details concerning the evaluation and calculation of operational and learning parameters of the system, the reader is referred to [9] since they remain unchanged. The main novelty lies on the way ARANFIS allocates new rules during the learning procedure. Learning is incorporated into ARANFIS using the gradient descent method. Training data are supplied to ARANFIS in the form of pairs

($\underline{x}(t)$, $\underline{d}(t)$) of input and target vectors. If a new input $\underline{x}(t)$ does not significantly activate any rule and the prediction error is significantly large (eq. 1-2), a new hidden node is created, increasing the total number of rules and setting the crisp values of $\underline{x}(t)$ as the new weight centers of the antecedent part. The weight centers of the consequent connections are set as the difference between the desired and the defuzzified output. The respective weight spreads are initialized with random values in the interval (0,1], which will be fine-tuned as training evolves. The mathematical expressions for the two conditions are:

$$\frac{1}{m} \sum_{k=1}^m (d_k(t) - y_k(t))^2 > \varepsilon, \text{ (prediction error)} \quad (1)$$

$$\max_j \{z_j\} < \delta, \text{ (rule activation)} \quad (2)$$

where m is the number of output nodes and y_k for iteration t is given by:

$$y_k(t) = \frac{\sum_{j=1}^{q(t)} z_j \left(v_{jk}^c + \frac{v_{jk}^{\sigma^r} - v_{jk}^{\sigma^l}}{\sqrt{\pi}} \right) (v_{jk}^{\sigma^l} + v_{jk}^{\sigma^r})}{\sum_{j=1}^{q(t)} z_j (v_{jk}^{\sigma^l} + v_{jk}^{\sigma^r})} \quad (3)$$

Epsilon and delta define the two thresholds upon which the rule insertion conditions are based. Their respective values are usually determined according to the complexity of the given task.

3 Combining Sensitivity Analysis and Resource Allocating Networks

Sensitivity analysis is a statistical tool that provides us with a way to estimate the influence of a slight change in value of one variable X towards another variable Y . In the field of neural networks it has been widely used as a method to extract the cause and effect relationship between the inputs and outputs of a trained network [10-12]. The input nodes that produce low sensitivity values can be regarded as insignificant and can be pruned from the system.

Obtaining the Jacobian matrix by the calculation of the partial derivatives of the output y_k with respect to the input x_i , that is $\partial y_k / \partial x_i$, constitutes the analytical version of sensitivity analysis [10]. Nevertheless, this method assumes that all input variables are numeric and continuous. When the input variables are discrete or symbolic, which is not rare in systems involving fuzzy attributes, the partial derivative cannot be deemed of practical significance.

Our methodology suggests the use of the activation strength z_j of rule j as our indicator of sensitivity towards the output variable y_k of the output node k . We opt to do this for two main reasons: First, the evaluation of the system rules is as important as the input features evaluation and secondly, the activation strength of a

rule is a quantitative and differentiable variable for most modern neurofuzzy systems and thus, we can apply sensitivity analysis on a wide variety of algorithms, regardless of the data set and the nature of the input features. In this paper, we apply this concept on a trained ARANFIS network model with its fully grown rule base.

We define Rule Sensitivity RS_{kj} over a trained rule-output pair p as:

$$RS_{kj}^{(p)} = \frac{\partial y_k}{\partial z_j} \tag{4}$$

Using equation (3), Rule Sensitivity for ARANFIS becomes:

$$RS_{kj}^{(p)} = \frac{\left(v_{jk}^{\sigma^l} + v_{jk}^{\sigma^r} \right) \left(\left(v_{jk}^c + \frac{v_{jk}^{\sigma^r} - v_{jk}^{\sigma^l}}{\sqrt{\pi}} \right) - y_k \right)}{\sum_{j=1}^{q(t)} z_j \left(v_{jk}^{\sigma^l} + v_{jk}^{\sigma^r} \right)} \tag{5}$$

Since each training pair p produces a different sensitivity matrix, we evaluate RS over the entire training set and obtain the average matrix:

$$RS_{kj,avg} = \sqrt{\frac{\sum_{p=1}^N [RS_{kj}^{(p)}]^2}{N}} \tag{6}$$

where N is the total number of patterns of the given data set. To allow accurate comparison among the variables, it is necessary to scale activation values and outputs to the same range using:

$$RS_{kj,avg} = \frac{(\max_{p=1,\dots,N} \{z_j^{(p)}\} - \min_{p=1,\dots,N} \{z_j^{(p)}\})}{(\max_{p=1,\dots,N} \{y_k^{(p)}\} - \min_{p=1,\dots,N} \{y_k^{(p)}\})} \tag{7}$$

The RS_{avg} matrix of the trained network shows, ultimately, the relation of each rule towards each output node. However, we need the relation of each rule towards the general outcome and so we define the significance Φ_j of rule z_j over all outputs as:

$$\Phi_j = \max_{k=1,\dots,K} \{RS_{kj,avg}\} \tag{8}$$

Our dynamic rule evaluation uses the significance Φ_j as a measure to classify the rules into two subsets based on a rather heuristic approach: If the significance Φ_j of a rule is larger or equal to 1 ($\Phi_j \geq 1$), this rule belongs to the main set of rules. Otherwise, if the significance Φ_j of a rule is less than 1 ($\Phi_j < 1$), this rule belongs to the secondary subset of rules. During real-time operation the system first employs the main rule set which carries the most information. If the output activation is below a confidence threshold, we add the rules of the second set, thus expanding our pool of knowledge as shown in figure 2.

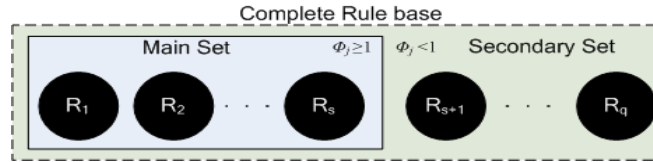


Fig. 2. The rule base split into two subsets.

The selected Confidence measure Cf is an extension of the method suggested in [13], where the reliability criterion for a classifier is the difference between the two greatest output values. In our approach, the confidence measure is slightly modified to accommodate more information:

$$Cf = (Y_{win1} - Y_{win2}) * Y_{win1} < CfT,$$

where Y_{win1} is the numerical value of the output node having the greatest value and Y_{win2} is the numerical output of the node having the second greatest value. If the confidence measure is below a certain threshold CfT , then the system considers the current rule-set insufficient and requests more rules to improve its results.

4 Experimental Results

To solidify our proposed method, a number of experiments were performed using five different datasets with various numbers of attributes as summarized in Table 1.

Table 1. Summarized dataset properties

Dataset	#of Patterns	#of Attributes	#of Classes
Pima Indians	768	8	2
Breast Cancer	699	9	2
Image Segmentation	2310	19	7
Ionosphere	351	34	2
Sonar	208	60	2

We split each dataset in a training set and a testing set on an approximate 60%-40% ratio. We then train our network for 100 epochs, using 0.001 as learning rate. The initialization of weights is random and the initial number of hidden nodes is set equal to 2. Comparisons were made based on the average of ten experiments performed for each dataset. It is worth noting that we compared not only network performance, but execution time values as well, and the percentage of patterns in which only the main subset of rules was employed, in contrast to the number of times the whole rule base was utilized. The careful observation of these factors shows the usefulness of our concept with respect to the computational cost of the network.

Table 2. Comparing properties and test performance for ARANFIS with and without Sensitivity Analysis Resource Control (SARC)

Dataset	Pima Indians	Breast Cancer	Image Segmentation	Ionosphere	Sonar
Epsilon, delta, CFT	0.5,0.1,0.7	0.5,0.1,0.8	0.4,0.1,0.6	0.4,0.1,0.8	0.4,0.1,0.7
Initial Rule number	2	2	2	2	2
Final Rule number (avg.)	8.5	14.5	12.8	48.3	36.4
Number of Main Rules ($\Phi \geq 1$)	2.6	4.1	7.7	14.3	21.4
ARANFIS Performance (%)	80.11	98.23	88.33	91.23	72.43
ARANFIS+SARC Performance (%)	77.67	95.92	85.42	87.80	67.78
ARANFIS Exec. Time (msec)	3.2	14.2	43.5	545.2	942.0
ARANFIS+SARC Exec. Time (msec)	1.7	7.6	28.4	251.9	738.8
Main rule set usage (%)	58.6	53.7	44.8	61.9	31.2

According to the results shown in table 2, system performance with sensitivity analysis resource control, is barely lower in average than the one without the rule usage restrictions. This may not seem beneficial, but if we take into account the flexibility, the lower overall cost and execution times of our proposed system, we can conclude that this network can be much more efficient and useful in practice.

5 Conclusions

In this paper we prove the efficiency that can be produced when one combines two different techniques of opposing philosophies. A novel neurofuzzy system which relies on a well-balanced and possibly optimal network structure is proposed. Our system is able to control its resource (rule) usage dynamically, through a confidence measure, which leads to lower computational cost and a flexible structure. The experimental study showed an improvement in resource usage of more than 30% with a minimal loss in network performance. It should also be noted that the number of main rules, selected by our method, could be an efficient indicator concerning the optimum number of hidden nodes required for a network and for a given task.

Nevertheless, our approach in its current form can be applied only on classification tasks due to its confidence measure. Thus, the need for an alternative criterion to address regression problems is an important issue under consideration. An interesting research topic of structure identification would also be the parallel evaluation of input and rule nodes with an analogous dynamic control mechanism, but this attempt certainly hides significant risks.

References

1. Zadeh L A (1994) Fuzzy logic, neural networks, and soft computing. *Communications of the ACM* 37:77-84
2. Jang JS R, Sun C-T, Mizutani E (1997) *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*. Prentice-Hall Inc., NJ
3. Jang JS R (1994) Structure determination in fuzzy modeling: a fuzzy CART approach. *Proceedings IEEE 3rd International Conf on Fuzzy Systems* 1:480-485
4. Pal T, Pal NR (2003) SOGARG: A Self-Organized Genetic Algorithm-based Rule Generation Scheme for Fuzzy Controllers. *IEEE Transactions on Evolutionary Computation*, 7(4):397-415
5. Mitra S, Hayashi Y (2000) Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework. *IEEE Transactions on Neural Networks* 11:748-768
6. Nurnberger A, Klose A, Kruse R (2000) Effects of Antecedent Pruning in Fuzzy Classification Systems. *Proceedings 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies* 1:154-157
7. Pertselakis M, Tsapatsoulis N, Kollias S, Stafylopatis A (2003) An adaptive resource allocating neural fuzzy inference system. *Proceedings IEEE 12th Intelligent Systems Application to Power Systems (electronic proceedings)*
8. Platt J (1991) A resource-allocating network for function interpolation. *Neural Computing* 3(2):213-225
9. Velayutham CS, Kumar S (2005) Asymmetric Subsethood-Product Fuzzy Neural Inference System (ASuPFuNIS). *IEEE Transactions on Neural Networks* 16(1):160-174
10. Engelbrecht P, Cloete I, Zurada JM (1995) Determining the significance of input parameters using sensitivity analysis. In: Miral J, Sandoval F (eds) *From Natural to Artificial Neural Computing*, LNCS, 930:382-388
11. Zurada JM, Malinowski A, Cloete I (1994) Sensitivity Analysis for Minimization of Input Data Dimension for Feedforward Neural Network. *Proceedings IEEE Symposium on Circuits and Systems*, 447-450
12. Howes P, Crook N (1999) Using Input Parameter Influences to Support the Decisions of Feedforward Neural Networks. *Neurocomputing* 24(1-3):191-206
13. Cordella LP, Foggia P, Sansone C, Tortorella F, Vento M (1999) Reliability Parameters to Improve Combination Strategies in Multi-Expert Systems. *Pattern Analysis and Application* 2:205-214