

Self-Organizing Neuro-Fuzzy Inference System*

Héctor Allende-Cid¹, Alejandro Veloz¹, Rodrigo Salas²,
Stereon Chabert², and Héctor Allende¹

¹ Universidad Técnica Federico Santa María; Dept. de Informática; Valparaíso-Chile
vector@inf.utfsm.cl, avelozb@inf.utfsm.cl, hallende@inf.utfsm.cl

² Universidad de Valparaíso; Departamento de Ingeniería Biomédica; Valparaíso-Chile
rodrigo.salas@uv.cl, stereon.chabert@uv.cl

Abstract. The architectural design of neuro-fuzzy models is one of the major concern in many important applications. In this work we propose an extension to Rogers's ANFIS model by providing it with a self-organizing mechanism. The main purpose of this mechanism is to adapt the architecture during the training process by identifying the optimal number of premises and consequents needed to satisfy a user's performance criterion. Using both synthetic and real data, our proposal yields remarkable results compared to the classical ANFIS.

Keywords: ANFIS, Flexible Architecture, Nonlinear modeling.

1 Introduction

Nowadays most Machine Learning models rely on a very rigid structure, where the configuration of the artificial neurons and their connections must be defined in advance, as most Artificial Neural Networks. The user is the one that should define a set of parameters to construct a specific model, as for example the number of neurons, a crucial and difficult decision in many real life problems. In this case the user has to use his empirical intuitions based on experience to determine the amount of nodes and layers. However, an unexperienced user might have difficulties trying to establish the best possible architecture.

For these reasons, we propose to improve the ANFIS model by providing it with flexibility to its architecture. There have been some studies in this subject like in [6] and [7]. Our new model is able to change its topology in order to automatically adjust its architecture to model the available complex and highly dimensional data. The proposed model can be applied to many engineering and scientific applications.

The structure of this paper is organized as follows: in the next section, we introduce the ANFIS model. In section 3 we present our proposed model. In section 4, we give some experimental results to show the capabilities of the proposal. Finally, in the last section, we discuss the concluding remarks and we delineate some future works.

* This work was supported by the Fondecyt 1070220 and 11060036 research grants and DGIP-UTFSM grant.

2 ANFIS: Adaptive-Network-Based Fuzzy Inference System

The *Adaptive Network-based Fuzzy Inference System* (ANFIS) was proposed by Jyh-Shing Roger in 1993 [4]. The model is a fuzzy inference system implemented in the framework of adaptive neural networks that can construct an input-output mapping based on both human intelligence and data samples. In this section we present the architecture and learning procedure of the ANFIS.

The ANFIS is considered in the literature as a neuro-fuzzy model. This model is suitable for modeling ill-defined and uncertain systems. Furthermore, as fuzzy inference system employs fuzzy “if-then” rules, it can model the qualitative facets of how the human learns or reasons without employing precise quantitative analyses. The fuzzy modeling or fuzzy identification used in ANFIS was first explored systematically by Takagi and Sugeno [5]. It consists in expressions of the form *IF A THEN B*, where *A* is a fuzzy premise characterized by appropriate membership functions and *B* is a non-fuzzy or crisp consequent part. ANFIS is a feed-forward adaptive network that employs Takagi and Sugeno’s fuzzy if-then rules, where the premise and consequent parameters are inferred from the training data set.

Let the input vector be $(x_1, \dots, x_d)' \in \mathcal{X}$, $\mathcal{X} \subseteq \mathbb{R}^d$, where v' is the transpose of the vector v . Define the extended input vector as $\mathbf{x} = (x_1, \dots, x_d, 1)'$ and let the output be $y \in \mathcal{Y}$, $\mathcal{Y} \subseteq \mathbb{R}$. The rule base consists on K fuzzy *if-then* rules of Takagi and Sugeno’s type [5], i.e. for each k rule we have:

$$\text{Rule } k: \text{ If } x_1 \text{ is } A_1^{(k)} \text{ and } \dots \text{ and } x_d \text{ is } A_d^{(k)}, \text{ then} \\ f_k(\mathbf{x}, \Theta_k) = \theta_1^{(k)} x_1 + \dots + \theta_d^{(k)} x_d + \theta_{d+1}^{(k)} = \Theta_k' \mathbf{x}.$$

where $\Theta_k = (\theta_1^{(k)}, \dots, \theta_d^{(k)}, \theta_{d+1}^{(k)})'$ is the vector of the consequent parameters of size $d+1$ of the k -th rule. These rules are modeled with an ANFIS model whose reasoning is summarized as follows. The architecture of ANFIS consists of five layers of neurons. The first layer is used to fuzzify the input variables; the T-norm operator is employed in the second layer to obtain the strengths of the rules; and the third layer normalizes these rules strengths. The fourth layer computes the weighted hyperplane, where the consequents of the rules are determined. Finally, the output of the network is calculated as the summation of all the incoming signals pertaining to the previous layer.

The nodes of **layer 1** (*Fuzzification Layer*) compute the degree to which a given input x_i satisfies the linguistic quantifier $A_i^{(k)}$. The output of the node is given by the membership function $\mu_{A_i^{(k)}}(x_i)$. In this work we use the gaussian-type membership function:

$$\mu_{A_i^{(k)}}(x_i; \eta_i^{(k)}) = \exp \left[- \left(\frac{x_i - \nu_i^{(k)}}{\sigma_i^{(k)}} \right)^2 \right]$$

where $\eta_i^{(k)} = \{\nu_i^{(k)}, \sigma_i^{(k)}\}$ are the *premise parameters* that should be estimated for the linguistic label $A_i^{(k)}$, the $\nu_i^{(k)}$ parameter stands for the location while $\sigma_i^{(k)}$ is the width of the membership function of the linguistic operator $A_i^{(k)}$.

The nodes of **layer 2** (*Generalized "AND Layer"*) consist in $T - norm$ operators that perform the generalized AND. Each node of this layer represents the firing strength of some specific rule. In this work we use the product $T - norm$, i.e.,

$$w_k = w_k(\mathbf{x}; \eta^{(k)}) = \mu_{A_1^{(k)}}(x_1) \times \dots \times \mu_{A_d^{(k)}}(x_d), \quad k = 1..K \tag{1}$$

The **layer 3** (*Normalization Layer*) computes the *normalizing firing strengths* of the weights of the previous layer: $\bar{w}_k = \bar{w}_k(\mathbf{x}; \eta^{(k)}) = \frac{w_k}{\sum_{j=1}^K w_j}$, $k = 1..K$.

The nodes of **layer 4** (*Consequent Layer*) compute the weighted hyperplane that approximates the nonlinear mapping, i.e.,

$$\bar{f}_k(\mathbf{x}; \eta^{(k)}, \Theta_k) = \bar{w}_k(\mathbf{x}; \eta^{(k)}) f_k(\mathbf{x}; \Theta_k) = \bar{w}_k(\theta_1^{(k)} x_1 + \dots + \theta_d^{(k)} x_d + \theta_{d+1}^{(k)}) = \bar{w}_k \Theta_k' \mathbf{x}$$

where \bar{w}_k is the output of the k -th node of layer 3.

Finally, **layer 5** (*Network Output*) consists in a single node that computes the overall output as the summation of all the incoming signals:

$$g(\mathbf{x}; \eta, \Theta) = \sum_{k=1}^K \bar{w}_k(\mathbf{x}; \eta^{(k)}) f_k(\mathbf{x}; \Theta_k)$$

where $\eta = (\eta_1^{(1)}, \dots, \eta_d^{(K)})'$ and $\Theta = (\Theta_1, \dots, \Theta_K)'$ correspond to the premise and consequent set of parameters respectively.

To estimate the parameters, the classical ANFIS employs an hybrid learning procedure that uses backpropagation learning algorithm to determine premise parameters η and the Least Mean Square (LMS) estimation procedure to determine the consequent parameters Θ . During the estimation of the consequent parameters Θ_k , $k = 1..K$, the premise parameters remains fixed. The updating rule is given by

$$\Theta_k(t + 1) = (A' \cdot A)^{-1} \cdot A' \cdot B \tag{2}$$

where $A = W_k \cdot X$ and $B = (Y - g(X; \eta, \Theta)) + (A \cdot \Theta_k(t))'$. W_k is a diagonal matrix where the k -th element of the diagonal is \bar{w}_k , and X is a matrix where each row corresponds to a data sample \mathbf{x} , Y is a vector containing the output training sample and $g(X; \eta, \Theta)$ is the output of the ANFIS for all the training data. The premise parameters $\eta_i^{(k)} = \{\nu_i^{(k)}, \sigma_i^{(k)}\}$ are estimated iteratively by the following updating rule:

$$\begin{aligned} \nu_i^{(k)}(t + 1) &= \nu_i^{(k)}(t) + 4\alpha(t) \frac{1}{(\sigma_i^{(k)})^2} (x_i - \nu_i^{(k)}) \bar{w}_k (f_k - g)(y - g) \\ \sigma_i^{(k)}(t + 1) &= \sigma_i^{(k)}(t) + 4\alpha(t) \frac{1}{(\sigma_i^{(k)})^3} (x_i - \nu_i^{(k)})^2 \bar{w}_k (f_k - g)(y - g) \end{aligned} \tag{3}$$

where the consequents parameters remains fixed and $\alpha(t)$ is the learning rate function. Please refer to reference [4] for further details.

3 SONFIS: Self-Organizing Neuro-Fuzzy Inference System

In this work we propose an extension to Rogers’s ANFIS model. The basic structure of the proposal and its functionality is similar to the ANFIS model explained in the previous section. However, during the learning procedure, our proposed model self organizes its architecture in order to automatically identify the number of premises and consequents needed to model the available complex and highly dimensional data.

The self-organization learning procedure consists in the following steps. In the first stage we construct a base model with a predefined number of nodes and we estimate the parameters using iteratively equations (2) and (3).

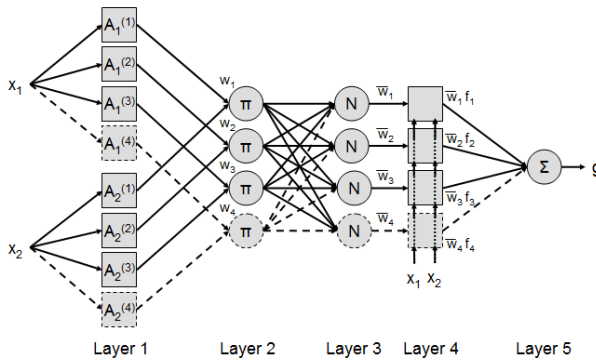


Fig. 1. The SONFIS architecture is similar to the ANFIS, however the model can add or eliminate a fuzzy if-then rules as it is shown in the segmented path

During the second stage we proceed to apply three types of operators: *Grow Net*, *Split Membership Functions* and *Vanish Membership Functions*. These operators are applied iteratively until the net self organizes and stabilizes satisfying some user’s performance criterion. Before applying any operator, the current base model is frozen meaning that none of the parameters can be any longer updated.

The *Grow Net* operator consists in adding a new rule: introducing new membership functions to the premises for each input dimension and a new hyperplane to the consequents. For each input \mathbf{x} of the training data we compute the firing strength w_k for all the K rules with equation (1). If the maximum of these strengths is less or equal than a user defined threshold δ to the power of d , where d is the dimension of the input space, i.e., $\max_{k=1..K} w_k \leq \delta^d$, then we say that the sample (\mathbf{x}, y) is not well-modeled by the current model. We add it to a “bad samples” set, together with the information of the best matching rule that currently best model the sample, i.e.,

$$w_\kappa = \arg \max_{k=1..K} w_k(\mathbf{x}, \eta^{(k)}) \tag{4}$$

After having revised all the training data, we group the data into the set \mathcal{V}_κ according to their best matching rule w_κ . For each group that has more than N_{grow} samples, where N_{grow} is user-defined, we construct and add a new membership function for each dimension $\mu_{A_i^{(K+1)}}(x_i), i = 1..d$, with the premise parameters initialized with the mean and standard deviation of the samples belonging to this group, i.e.,

$$\nu_i^{(K+1)} = \frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} x_i^{(\kappa)} \quad \sigma_i^{(K+1)} = \sqrt{\frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} (x_i^{(\kappa)} - \nu_i^{(K+1)})^2}$$

The consequent parameters of the new rule are initialized randomly.

The *Split Membership Functions* operator consists in splitting a rule that has a bad performance into two new rules. To evaluate the rule performance, the training set is partitioned in K sets where the sample (\mathbf{x}, y) is assigned to the set \mathcal{V}_k if its best matching rule (4) is w_k . For each set we compute the mean square error,

$$E_k = \frac{1}{N_k} \sum_{(\mathbf{x}, y) \in \mathcal{V}_k} (y - g(\mathbf{x}; \eta, \Theta))^2$$

where N_k is the number of samples belonging to the set \mathcal{V}_k . If the performance of the rule k , E_k , is higher than a user defined threshold ϵ and N_k is higher than the minimum required samples N_{split} , where N_{split} is user-defined, then the rule is divided into two new rules. If the premise parameters of the k -th rule are ν and σ , then the premise parameters of the new rules are:

$$\nu_i^{(K+1)} = \nu - \frac{\sigma}{2}; \quad \sigma_i^{(K+1)} = \frac{\sigma}{2}; \quad \nu_i^{(K+2)} = \nu + \frac{\sigma}{2}; \quad \sigma_i^{(K+2)} = \frac{\sigma}{2}$$

The consequents parameters are set randomly. After the inclusion of the new rules, the k -th rule is eliminated.

The *Vanish Membership Functions* operator consists in eliminating rules that model less than N_{vanish} sample data, where N_{vanish} is user-defined. To accomplish this, we introduce an age_k variable that starts from zero and is increased by one if the rule models no data, i.e. if the set \mathcal{V}_k is empty, and the rule is unfrozen. If the age variable of the k -th rule reach the threshold λ , then the k -th rule is eliminated, where λ is user-defined. If the set \mathcal{V}_k is no longer empty, then the age_k variable is set back to 0.

After all three operators are applied to the model, all the unfrozen parameters (rules) are updated according to equations (2) and (3). After this step, the whole net architecture is frozen. The operators and the training steps are applied iteratively until the architecture self-stabilizes, and no longer changes. Finally, all the parameters (frozen and unfrozen) of the network are updated in the last iteration.

4 Experimentation

In this section we show the performance of our proposed *Self-Organizing Neuro-Fuzzy Inference System (SONFIS)* model compared to the static

Adaptive-Network-Based Fuzzy Inference System (**ANFIS**), in both synthetic and real datasets, the latter were obtained from a benchmark site. For the synthetic experiment we have created a synthetic data set based on the following function:

$$f(x, y) = \frac{\sin(x^2) * \exp(-x) + \sin(y^2) * \exp(-y) + 0.2338}{0.8567}$$

defined in the range $[0, 3] \times [0, 3]$. For further explanation of this experiment see [1]. In the experiments with real datasets, we used the Bodyfat and Boston Housing datasets, both obtained from the StatLib repository [2], while the Building dataset was obtained from the Proben1 collection [3]

The data of both synthetic and real datasets were separated in training (60%) and test sets (40%). All the results reported were obtained for each model as the mean value of the metrics computed for 20 runs with the same set of data. We compare the performance of the classic ANFIS(1) model with 1 membership function corresponding to the starting point of our proposal, the Classic ANFIS(K) model with a number of rules K equal to the mean value of the rules created by our proposed algorithm, and the SONFIS model. The ANFIS(1) was trained with 5 epochs, the SONFIS was trained with 5 epochs for each iteration until it self-organize, and the second ANFIS(K) model was trained with a number of epochs equal to five times the mean value of the resulting iterations of our algorithm. The parameters chosen for the SONFIS model are summarized in table 1.

Table 1. Operators parameters of the SONFIS model

Dataset	ϵ	N_{split}	δ	N_{grow}	λ	N_{vanish}
Synthetic	0.01	12	0.5	4	2	12
Bodyfat	0.01	10	0.01	10	2	10
Boston	0.1	30	0.5	14	2	30
Building	0.13	30	0.82	10	2	30

Figure 2 shows the approximations obtained with both the ANFIS with 7 membership functions and the SONFIS models. Qualitatively the SONFIS obtain a better fit than the ANFIS and, furthermore, the model was able to identify the number of rules needed for the adjustment. Table 2 quantitatively shows the summary results for each data set (Dataset) of the mean number of rules (Rules), the training mean square error (MSE-train), the best performance during the training (Min-Train), the test mean square error (MSE-Test), the best performance during the test (Min-Test) and the mean number of epochs (Epochs).

In the Synthetic experiment, our proposed SONFIS algorithm outperforms the ANFIS models in every measurement, where the mean square error of the training and test sets of SONFIS model are better than the classical approaches in one order of magnitude, and they are statistically significant.

In the real experiment, the SONFIS model obtained better results than the classic ANFIS in most of the data sets in both the training and test set. However

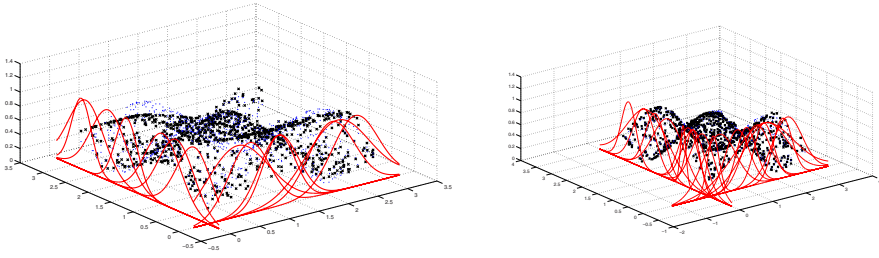


Fig. 2. Synthetic Experiment Approximations results obtained with the (left) ANFIS and (right) SONFIS models

Table 2. Summary results of the performance evaluation of the ANFIS(1), SONFIS and ANFIS(T)

Dataset	Algorithm	Rules	MSE-Train	Min-Train	MSE-Test	Min-Test	Epochs
Synthetic	ANFIS(1)	1	0.0626 ± 0.058	0.0369	0.0638 ± 0.060	0.0358	5
	ANFIS(7)	7	0.0233 ± 0.006	0.0134	0.0237 ± 0.006	0.0136	20
	SONFIS	6.63	0.0053 ± 0.003	0.0015	0.0056 ± 0.003	0.0015	20
Bodyfat	ANFIS(1)	1	5.5823 ± 11.1048	0.2453	4.8492 ± 7.15	0.3369	5
	ANFIS(3)	3	3.6576 ± 3.54	0.2156	0.9457 ± 0.9188	0.3115	15
	SONFIS	2.95	0.2412 ± 0.013	0.1934	1.0001 ± 0.185	0.2986	15
Boston	ANFIS(1)	1	13137 ± 6516	387.45	12107 ± 6128	398.35	5
	ANFIS(9)	9	9.78 ± 2 · 10⁻⁵	9.78	325.61 ± 89.32	130.20	50
	SONFIS	9.5	15.99 ± 15.63	2.10	29.65 ± 35.47	5.06	50
Building	ANFIS(1)	1	0.5443 ± 0.239	0.1636	0.5354 ± 0.238	0.1560	5
	ANFIS(18)	18	0.1660 ± 0.210	0.0067	0.1503 ± 0.158	0.0063	50
	SONFIS	18.15	0.0322 ± 0.066	0.006	0.0307 ± 0.061	0.0061	50

in the Bodyfat test set the ANFIS(3) obtained a low mean square error but with a very high variance compared to the SONFIS. In the Boston Housing dataset the ANFIS(9) was overtrained outperforming the SONFIS in the training set, but it obtained a very bad performance in the test set. And finally, in the Building dataset, we can observe that our algorithm outperforms the ANFIS models.

5 Conclusion

In this paper we have introduced a flexible architecture algorithm for *ANFIS* called SONFIS. Our proposed model self organizes its architecture in order to automatically identify the number of premises and consequents needed to model the available complex and highly dimensional data. The adaptation is performed introducing Takagi and Sugeno's fuzzy if-then rules that can improve the fitting performance.

Our algorithm shows better results in both synthetic and real data sets. We investigated three benchmark data known as Bodyfat, Boston and Building. The comparative study with the classic ANFIS (with 1 rule and with a specific

number of rules) shows that our Algorithm outperforms the alternative models with statistical significance, obtaining good results in both synthetic and real data. Further studies are needed in order to analyze convergence and the stability of the proposed model under changes of the operator's parameters. Studies in order to compare our proposed model with other algorithms are also needed.

A fixed number of fuzzy rules is not necessary anymore, contrary to the classical ANFIS algorithm. This is a useful improvement in applications where an unexperienced user needs to use Neural Networks without knowing the optimal net architecture.

References

1. Moraga, C., Salas, R.: A new aspect for the optimization of fuzzy if-then rules. In: 35th International Symposium on Multiple-Valued Logic, pp. 160–165. IEEE-CS Press, Los Alamitos (2005)
2. Department of Statistics at Carnegie Mellon University, Stat Lib - datasets archive, <http://lib.stat.cmu.edu/>
3. Prechelt, L.: PROBEN1 - A set of benchmarks and benchmarking rules for neural network training algorithms, Tech. Report 21/94, Germany, Anonymous FTP: /pub/papers/techreports/1994/1994-21.ps.Z on ftp.ira.uka.de (1994)
4. Jang, J.-S.R.: ANFIS: Adaptive-network-based fuzzy inference system. *IEEE, Transaction on Systems, Man and Cybernetics* 23(3), 665–685 (1993)
5. Takagi, T., Sugeno, M.: Derivation of fuzzy control rules from human operator's control actions. In: Proc. IFAC Symp. Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55–60 (1983)
6. Jeen-Shing, W., Lee, C.S.G.: Self-adaptive neuro-fuzzy inference systems for classification applications. *IEEE Transactions on Fuzzy Systems* 10, 790–802 (2002)
7. Castellano, G., Fanelli, A.M.: A Self-Organizing Neural Fuzzy Inference Network. In: IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000), vol. 5, pp. 14–19 (2000)