Draft full paper submitted to First International Workshop on Hybrid Methods for Adaptive Systems

Merging and Constrained Learning for Interpretability in Neuro-Fuzzy Systems

R. P. Paiva, A. Dourado
CISUC – Centro de Informática e Sistemas da Universidade de Coimbra
Departamento de Engenharia Informática, PÓLO II da Universidade de Coimbra,
Pinhal de Marrocos, P 3030, Coimbra, Portugal
Tel. +351-239-790000, Fax: +351-239-701266, e-mail: {ruipedro, dourado}@dei.uc.pt

Abstract.
A methodology for development of linguistically interpretable fuzzy models from data is developed. The implementation of the model is conducted through the training of a neuro-fuzzy network. Structure of the model is firstly obtained by subtractive clustering, allowing the extraction of a set of relevant rules from input-output data. The model parameters are then tuned via the training of a neural network through backpropagation. Interpretability goals are pursued through membership function merging and some constrains on the tuning of parameters. The assignment of linguistic labels to each of the membership functions is then possible. The model obtained for the system under analysis can be described, in this way, by a set of linguistic rules, easily interpretable.

Keywords: neuro-fuzzy learning, fuzzy systems, clustering, interpretability, transparency.

1. Introduction

Extracting knowledge from data is a very interesting and important task in information science and technology. Sometimes it is necessary that the resulting models be interpretable in order to understand the system under study [1][2][3]. Fuzzy modeling founds here its maximum potential. but it has associated the difficulty to quantify the fuzzy linguistic terms. Neuro-fuzzy networks appear as a tool to surpass the limitation.

In this paper a methodology is developed carried out in two main phases: in the first one a set of fuzzy rules is obtained; in the second one the parameters of the membership functions of the fuzzy system are tuned. A balance between accuracy and interpretability is pursued.

Linguistic models are used instead of Takagi-Sugeno models. Additionally, parameter learning is constrained and similar membership functions are merged, in order to ease the assignment of linguistic labels to the final fuzzy sets.

In Section 2 the main issues of fuzzy structure and parameter learning are presented. Subtractive clustering, used for structure and parameter learning is presented in Section 3 In Section 4, the strategies for implementation of interpretable models are developed. The methodologies are applied to the Mackey-Glass chaotic time series, in Section 5 and finally, some conclusions are drawn in Section 6.

2. Structure and parameter learning

Let it be assumed, without loss of generality, a single-input single-output (SISO) model, with one input, *u*, and one output, *y*, from where *N* data samples are collected (1):

$$Z^N = \{[u(1), y(1)], [u(2), y(2)], ..., [u(N), y(N)]\} \tag{1}$$

Using the data collected, the goal is to derive a fuzzy model, represented by a set of rules of type R_i (2):

$$R_i: If\ \ y(t-1)\ is\ A_{1i}\ \ and\ \ u(t-d)\ is\ B_{1i}\ \ then\ \ \hat{y}(t)\ is\ C_{1i} \tag{2}$$

where *d* represents the system time delay and A_{ji}, B_{ji} and C_{ji} denote linguistic terms associated to each input and output. Those terms are defined by their respective membership functions $m_{A_{ji}}, m_{B_{ji}}, m_{C_{ji}}$. The previous structure is called a *FARX* structure (Fuzzy Auto Regressive with eXogenous inputs) as a generalization of the well-known ARX structure. Thus, the selection of a set of rules of type (2), as well as the definition of the fuzzy sets A_{ji}, B_{ji} and C_{ji}, constitute some project issues specific to fuzzy systems.

Chiu's subtractive clustering is initially applied to obtain a set of *g* fuzzy rules [4] composing the model structure. A set of points is defined as possible group centers, each of them being interpreted as an energy source. In subtractive clustering, the center candidates are the data samples themselves.

After determining a model structure, the model parameters, i.e., the centers and standard deviations of the Gaussian membership functions, should be tuned. In the present work, such task is performed by training a fuzzy neural network based on [5] (Figure 1). This network is composed by five layers: an input layer, a fuzzification layer, a rule layer, an union layer and an output or defuzzification layer, sequentially.

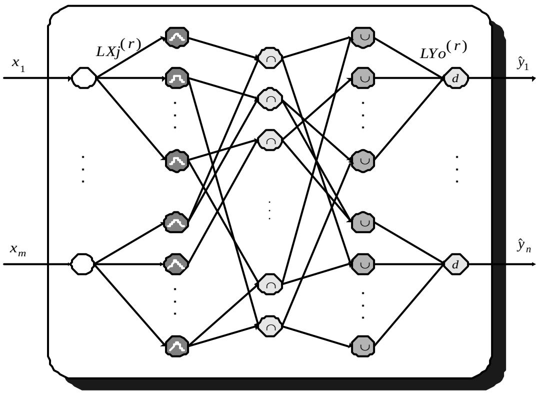

Figure 1. Neuro-fuzzy network.

The notation used is as follows:

- $a_i^{(p2)}$: activation of the neuron *i* in layer 2, regarding the training pattern *p* (*i* denotes an input term: "*input*");
- $a_r^{(p3)}$: activation of the neuron *r* in layer 3, regarding the pattern *p* (*r* denotes "*rule*");
- $a_s^{(p4)}$: activation of the neuron *s* in layer 4, regarding the pattern *p* (*s* denotes "*S-norm*");
- $a_o^{(p5)} = y_o^{(p)}$: activation of the neuron *o* in layer 5, i.e., output, regarding the pattern *p* (*o* denotes "*output*");
- $y_o^{(p)}$: desired activation for neuron *o* in layer 5, i.e., for the network output, regarding pattern *p*.

The *input layer* simply receives data from the external environment and passes them to the next layer.

In the second layer, the *fuzzification layer*, each of the cells corresponds to a membership function associated to each of the inputs. Since this work assumes the goal of obtaining interpretable models, two-sided Gaussian functions are proposed to used (Figure 2). The activation of each of the neurons in this layer is given by (3).

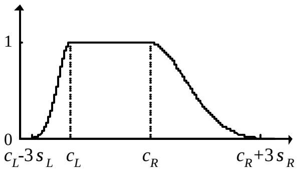

Figure 2. Two-sided Gaussian function.

$$a_i^{(p2)} = \begin{cases} e^{-\frac{\left(x_j^{(p)} - c_{ijL}\right)^2}{2s_{ijL}^2}} & , x_j^{(p)} < c_{ijL} \\ 1 & , c_{ijL} \le x_j^{(p)} \le c_{ijR} \\ e^{-\frac{\left(x_j^{(p)} - c_{ijR}\right)^2}{2s_{ijR}^2}} & , x_j^{(p)} > c_{ijR} \end{cases} \tag{3}$$

Here, c_{ijL} and s_{ijL} represent, respectively, the center and standard deviation of the left component of the i^{th} membership function related to the j^{th} input. For the right component, the index *R* is used. Such parameters constitute the eights of the layer one to layer two links ($LXj^{(r)}$ in Figure). In the same expression, $x_j^{(p)}$ denotes the p^{th} pattern associated do input *j*.

As for the neurons in the *rule layer*, their function consists of performing the antecedent conjunction of each rule, by means of some T-norm. By experimental testing, it was concluded that truncation operators (e.g., minimum) lead to better results than algebraic operators (e.g., product), when interpretability is desired. So, operator minimum is selected for fuzzy conjunction (4):

$$a_r^{(p3)} = T - \underset{i=1}{\overset{na_r}{norm}}\left(a_i^{(p2)}\right) = \min_{i=1}^{na_r}\left(a_i^{(p2)}\right) \tag{4}$$

where na_r stands for the number of inputs in the antecedent of rule *r*.

The fourth layer, called the *union layer*, is responsible for integrating the rules with the same consequents, via some S-norm. Once again, truncation operators are preferred, namely operator maximum (5).

$$a_s^{(p4)} = S - \underset{r=1}{\overset{nr_s}{norm}}\left(a_r^{(p3)}\right) = \max_{r=1}^{nr_s}\left(a_r^{(p3)}\right) \tag{5}$$

There, nr_s stands for the number of rules which have the neuron *s* as consequent.

As for the *output layer*, or *defuzzification layer* (*d*, in Figure), the layer four to layer five links ($LYo^{(r)}$ in the same figure) define the parameters of the membership functions associated to the output linguistic terms. Thus, based on these membership functions and on the activation of each rule, its neurons should implement a defuzzification method suited for two-sided Gaussian functions, as the one presented in [6] (6):

$$\hat{y}_o^{(p)} = a_o^{(p5)} = \frac{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}\left(c_{osL}s_{osL} + c_{osR}s_{osR}\right) a_s^{(p4)}}{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}\left(s_{osL} + s_{osR}\right) a_s^{(p4)}} \tag{6}$$

where c_{osL} and s_{osL} represent the center and standard deviation of the left component of the s^{th} membership function related to output *o*.In the previous expression, $|T(Y_o)|$ stands for the number of membership functions associated to each linguistic output variable Y_o. The main idea of the defuzzification method proposed is to weight the activation of each rule, not only by the centers, right and left, but also by their standard deviations.

Based on the function performed by each neuron, the network is trained in batch mode, via the well-known backpropagation algorithm.

3. Interpretability and transparency

Fuzzy systems should be linguistically interpretable.

However, this issue is often ignored, being given prevalent relevance to the approximation capabilities. As Nauck and Kruse refer [7], in case interpretability is not a major concern, it is important to consider other more adequate methods.

.The unrestricted parameter learning may lead to a highly complex set of membership functions, for which it will be difficult to assign linguistic labels. It is therefore important to impose adequate restrictions for parameter learning, so that interpretability is attained. Two-sided Gaussian functions are appealing due to their increased flexibility, which permits to control function overlapping and improves function distinguishability.

Three main criteria for model interpretability are defined. The first one, and most important, is related to function distinguishability. The others come from human cognitive issues: the number of rules and the number of membership functions associated to each variable should not be excessive. In the present case, these issues are monitored in order to obtain a satisfactory trade-off between model accuracy and interpretability.

3.1. Merging of membership functions

Structure learning by means of clustering techniques leads usually to initial membership functions with a high similarity degree. That makes the model lack transparency and originates an excessive number of parameters to adjust. It seems useful to merge functions with a high enough similarity degree.

In order to perform function merging, directed to rule base simplification, it was concluded in [8] that S_1 (1) is a very adequate similarity measure. There, the similarity between two fuzzy sets A and B is given by the result of the division of the area of their intersection by the area of their union:

$$S_1(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} \tag{1}$$

where the fuzzy intersection and union are performed, respectively, by the operators minimum and maximum.

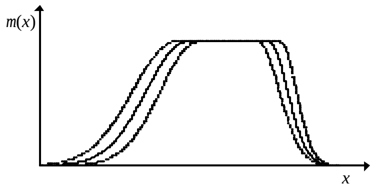

Figure 3. Membership function merging.

After detecting the most similar pair of membership functions, if their degree of similarity is above some threshold, those functions are merged. The new function results by averaging the parameters of the original functions, i.e., centers and standard deviations, as is depicted in Figure 3. There, the original functions are represented in dashed lines and the resulting function in solid line. The procedure of membership function merging, one pair in each iteration, continues until no more pairs satisfy the merging threshold.

As a result of function merging, the rule base is updated. In fact, the rules regarding the fuzzy sets merged will then contain the new function obtained.. Therefore, the rule base may be simplified in case redundant rules are obtained. Besides that, situations of inconsistency may result, if rules with the same antecedents have different consequents. This may be a consequence of deficient structure learning or may indicate that the merging threshold should be adjusted.

3.2. Restricted Parameter Learning

After rule base simplification through function merging, it is essential to guarantee the interpretability is maintained during parameter optimization. The optimization procedure is monitored so that function distinguishability is attained, as follows:

(i) *limit overlapping* It is assumed that the overlapping degree between two functions is excessive in case the supreme of the support of the function to the right, i.e., its right "zero", goes beyond the right zero of the function to the right. The same reasoning applies to the left component of the functions. Formally, it turns out (8):

$$\begin{aligned} c_{kR} + 3s_{kR} &\le c_{iR} + 3s_{iR} \\ c_{kL} - 3s_{kL} &\ge c_{jL} - 3s_{jL} \end{aligned} \tag{8}$$

where k refers to some membership function and i and j are, respectively, its right and left neighbor functions. In case function overlapping does not satisfy the constraints presented in (8), the standard deviations of function k are altered in order to keep those conditions. Therefore, the right and left components are changed as in (9) and (10), respectively:

$$s_{kR} = \frac{c_{iR} + 3s_{iR} - c_{kR}}{3} \tag{9}$$

$$s_{kL} = \frac{c_{jL} - 3s_{jL} - c_{kL}}{-3} \tag{10}$$

(ii) *guaranteed function distance*

Besides overlapping monitoring, it was concluded that function distance should also be checked. This procedure aims to avoid inclusions, i.e., functions total or almost totally "inside" other functions. Furthermore, the fact that the functions are not too close also improves model interpretability. Thus, the constraint (11) for the minimal distance between functions was defined:

$$\begin{aligned} c_{iL} - c_{kR} &\le a\left(X_{\max} - X_{\min}\right) \\ c_{kL} - c_{jR} &\le a\left(X_{\max} - X_{\min}\right) \end{aligned} \tag{11}$$

where $a \in [0;1]$ denotes the percentage of the domain

[$X_{\min}$; $X_{\max}$] used for calculating the minimal distance allowed. In case this condition does not apply, the function centers are changed as follows (12):

$$c_{kR}^{new} = \frac{c_{kR} + c_{iL}}{2} - \frac{a(U_{\max} - U_{\min})}{2}$$

$$c_{iL}^{new} = \frac{c_{kR} + c_{iL}}{2} + \frac{a(U_{\max} - U_{\min})}{2} \tag{12}$$

In this situation, the new centers will be based on the average of the right and left original centers of the two functions compared, from which their values are altered in order to guarantee the distance required.

Despite the restrictions imposed, it may turn out that the final model is not sufficiently interpretable, as a result of the trade-off between interpretability and accuracy. Therefore, it is useful to perform function merging every *x* training epochs.

4. Simulation Results

One of the most commonly used case studies in system identification consists of the prediction of the Mackey-Glass chaotic time series [9], described by equation (13).

$$\dot{x}(t) = \frac{0.2x(t-t)}{1+x^{10}(t-t)} - 0.1x(t) \tag{13}$$

The time series does not show a clear periodic behavior and it is also very sensible to initial conditions.

The problem consists of predicting future values of the series.

The application of the technique described previously is carried out based on identification data from the "*IEEE Neural Network Council, Standards Committee, Working Group on Data Modelling Benchmarks*", which are also used in the analysis of several other methodologies. So, in order to obtain a numeric solution the fourth order Runge-Kutta method was applied. For integration, it was assumed $x(t)$=0, t<0, and a time interval of 0.1. The initial condition $x(0)$=1.2 and the parameter τ=17 were also defined. In this case, [$x(t$-18), $x(t$-12), $x(t$-6), $x(t)$] are used to predict $x(t$+6). Based on the parameterization described, data was obtained in the interval $t \in$ [0; 2000], after what 1000 input-output pairs were selected from $t \in$ [118; 1117]. The data collected are depicted in Figure .

Using the samples obtained, the chaotic time series was modeled, according to the procedures described in the previous sections. Thus, the parameter r_a was assigned the value 0.5, resulting 9 fuzzy rules. Next, the network, with four inputs and one output, was trained, defining 0.65 for the merging threshold and x = 200.

So, after 800 epochs the RMS (Root Mean Square) error was 0.0228 for training data and 0.0239 for test data. As for the number of membership functions for the variables $x(t$-18), $x(t$-12), $x(t$-6), $x(t)$ and $x(t$+6), it resulted, respectively, 5, 4, 5, 4 and 5, leading to 92 adjustable parameters.

In Figure the results obtained for test data are depicted. It can be seen that they are satisfactory.

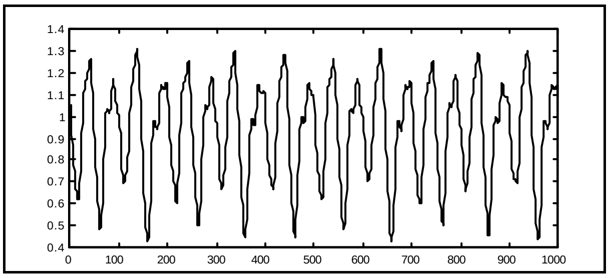

Figure 4 Chaotic time series: identification data.

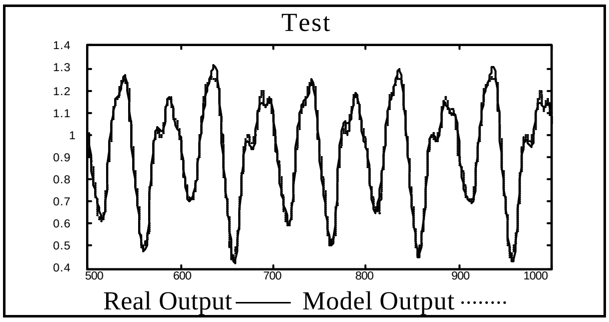

Figure 5. Chaotic series: output prediction.

As for membership functions, the results obtained are presented in Figure . As can be seen, it is not too difficult to assign linguistic terms to each of the membership functions. In the same figure, the labels *VS*, *S*, *M*, *B* and *VB* denote, respectively, the linguistic terms "very small", "small", "medium", "big" and "very big". Thus, the fundamental dynamics of the chaotic time series are interpreted according to Table .

5. Conclusions

In this paper a neuro-fuzzy methodology for the implementation of real interpretable fuzzy models is described. By the application of subtractive clustering, an initial structure for the fuzzy model was obtained, which is used for the initialization of a fuzzy neural network. However, adjusting membership function parameters without any constraints leads usually to complex overlapping between functions, which limits interpretability. Therefore, a learning scheme to allow the development of interpretable fuzzy models is proposed. The methodology presented is based on similar membership function merging and on constrains regarding parameter tuning, in order to improve function distinguishability in terms of distance and overlapping. The approach described is applied to the prediction of the Mackey-Glass chaotic time series, resulting a satisfactory trade-off between model accuracy and interpretability. However, it is important to point out that the results are not always acceptable. In fact, as complexity grows, the constraints imposed may lead to inaccurate models, which, consequently, are of no use. Clearly, it can be said that interpretability bounds accuracy and vice-versa.

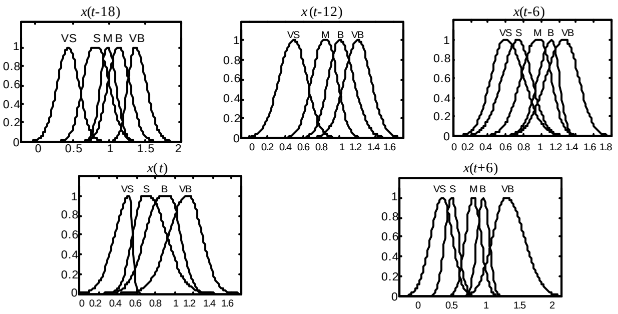

Figure 6. Membership functions obtained.

Rule	*x*(*t*-18)	*x*(*t*-12)	*x*(*t*-6)	*x*(*t*)		*x*(*t*+6)
1	M	VB	B	VB	Þ	B
2	B	VB	M	S		S
3	S	M	M	VB		VB
4	M	M	VS	VB		B
5	S	B	S	VS		M
6	S	VB	VB	B		M
7	S	VS	S	B		B
8	VS	VS	M	B		B
9	VB	VB	VB	B		VS

Table 1. Linguistic description of the series.

Comparing to NEXPROX [7] (Table), the results obtained are clearly better.

Method	Nr. Rules	Nr. Param.	RMSE
Paiva and Dourado	9	92	0.0239
NEFPROX (A)	129	105	0.0332
NEFPROX (G)	26	38	0.0671

Table 2. Chaotic series: comparison with other techniques.

References

[1] Espinosa J. and Vandewalle J. (2000). Constructing fuzzy models with linguistic integrity from numerical data-AFRELI Algorithm, IEEE Transactions on Fuzzy Systems, 8(5), 591-600.

[2] Fuessel, D. and R. Isermann (2000), Hierarchical motor diagnosis utilizing structural knowledge and a self-learning neuro-fuzzy scheme, IEEE Trans on Industrial Electronics, vol. 47 (5), p. 1070-1077, Oct. 2000.

[3] Jin Y.C. (2000). Fuzzy modelling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. On Fuzzy Systems, 8(2), 212-221.

[4] Chiu S. L. (1994). “Fuzzy model identification based on cluster estimation”, *Journal of Intelligent and Fuzzy Systems*, Vol. 2, No. 3, pp. 267-278.

[5] Lin C.- T. (1995). “A neural fuzzy control scheme with structure and parameter learning”, *Fuzzy Sets and Systems*, Vol. 70, pp. 183-212.

[6] Paiva R. P. (1999). *Identificação Neuro-Difusa: Aspectos de Interpretabilidade* (Neuro-Fuzzy Identification: Interpretability Issues), MSc Thesis, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Portugal (in Portuguese).

[7] Nauck D. and Kruse R. (1999). “Neuro-fuzzy systems for function approximation”, *Fuzzy Sets and Systems*, Vol. 101, pp. 261-271.

[8] Setnes M. (1995). *Fuzzy Rule-Base Simplification Using Similarity Measures*, MSc Thesis, Department of Electrical Engineering, Delft University

[9] Mackey M. C. and Glass L. (1977). “Oscillation and chaos in physiological control systems”, *Science*, vol. 197, pp. 287-289.

ACKNOWLEDGEMENTS

This work was supported partially by the Portuguese Ministry of Science and Technology (FCT-MCT) POSI Program supported by European Union and by , under the program PRAXIS XXI Alcine Project.