One-Dimensional Linear Local Prototypes for Effective Selection of Neuro-Fuzzy Sugeno Model Initial Structure

Jacek Kabziński

Institute of Automatic Control, Technical University of Lodz, Stefanowskiego 18/22, 90 924 Lodz, Poland, jacek.kabzinski@plodz.pl

Abstract. We consider a Takagi-Sugeno-Kang (TSK) fuzzy rule based system used to model a memory-less nonlinearity from numerical data. We develop a simple and effective technique allowing to remove irrelevant inputs, choose a number of membership functions for each input, propose well estimated starting values of membership functions and consequent parameters. All this will make the fuzzy model more concise and transparent. The final training procedure will be shorter and more effective.

Keywords: fuzzy modeling, neuro-fuzzy systems.

1 Introduction

We consider a Takagi-Sugeno-Kang (TSK) fuzzy rule based system used to model a memory-less nonlinearity from numerical data. It is typical for such application to perform a training of the system to improve modeling accuracy. One of the first (and very successful) approaches to the problem was Adaptive Neuro-Fuzzy Inference System (ANFIS) proposed by Jang [1]. The idea was to realize a fuzzy inference system as a neural network and to apply neural network learning techniques to train the fuzzy model. The ANFIS technique (where least squares optimization of consequent parameters is used and gradient decent error back propagation is applied to tune membership functions parameters) is offered as a standard tool of Matlab and is still very popular and employed to solve various modeling problems - for example [2,3]. Several other learning algorithms were developed to train fuzzy models including genetic, bacterial and PSO algorithms [4,5,6].
Before the training of parameters is started we have to propose initial structure of the fuzzy model. For a real-world problems we may consider a great number of potential inputs, but it is essential to select only the really important ones for system description. Also the number of membership functions for each input should be carefully chosen to compromise between model accuracy and complexity. The necessity of ANFIS input selection was noticed already by Jang [7] and this aspect of fuzzy model complexity reduction was investigated by many researchers [8], but the problem remains open. The most popular approaches to initial system architecture selection are based on several clustering techniques[8].

The purpose of this work is to develop a simple and effective technique allowing to remove irrelevant inputs, choose a number of membership functions for each input, propose well estimated starting values of membership functions and consequent parameters. All this will make the fuzzy model more concise and transparent. The resulting final training procedure will be shorter and more effective. The proposed approach is based on a concept of fuzzy projection of the data on each input space [9,10], combined with piece-wise linear approximation allowing to choose automatically a number of membership functions according to the required accuracy and to estimate starting values of all fuzzy system parameters. We demonstrate the effectiveness of the proposed technique investigating several examples and comparing results with popular clustering methods.

2 Neuro-fuzzy inference system

We consider a neural network realisation of TSK fuzzy model [1]. The network possesses N inputs $x_1,...,x_N$, one output y, m_i membership functions μ_{i,j_i}, $j_i \in \{1,....m_i\}$ associated with the i-th input $i \in \{1,....N\}$, $N_R = \prod_{i=1}^{N} m_i$ rules of the form:

$$IF \quad (x_1 \ IS \ \mu_{1,j_1}) \quad AND...AND \quad (x_N \ IS \ \mu_{N,j_N}) \quad THEN$$
$$y = p_{1,r}x_1 + ... + p_{N,r}x_N + q_r = f_r(x_1,...,x_N) \tag{1}$$

where $j_i \in \{1,....m_i\}$ and r is the rule number. We can choose any smooth membership function, for example generalized bell-shape function:

$$\mu_{i,j}(x) = \frac{1}{1 + \left(\dfrac{x-c}{a}\right)^{2b}} \tag{2}$$

The rule firing strength is calculated as:

$$w_r = \prod_{i=1}^{N} \mu_{i,j_i}(x_i) \tag{3}$$

for the r-th rule and the normalised firing strength is:

$$\overline{w}_r = \frac{w_r}{\displaystyle\sum_{i=1}^{N_R} w_i} \quad . \tag{4}$$

The final output is calculated as:

$$y = \sum_{r=1}^{N_R} \overline{w}_r f_r(x_1, \ldots, x_N) \quad . \tag{5}$$

It is well known that this Sugeno fuzzy model may be realised as a five-layer neural network: the first layer calculates the membership functions values, the second layer - firing strengths, the third - normalised firing strengths, the fourth and the fifth calculates the final output [1]. The standard training procedure proposed for ANFIS combines gradient descent back-propagation method to tune membership functions parameters and least squares method to find optimal consequent parameters.

The system is supposed to model a memory-less nonlinearity given by input-output data with possible extraneous inputs. We assume that that the input-output data are given by

$$\left(x_{1,k}, \cdots, x_{\widetilde{N},k} \right) \to y_k \quad \widetilde{N} \geq N, \quad k \in \{1, \ldots, m\} \quad . \tag{6}$$

3 Initial structure selection procedure

To test the significance of the i-th input, $i \in \{1, \ldots, \widetilde{N}\}$ we consider a single-input fuzzy model (SIFM) described below:

- input - x_i , output - c_i,
- input linguistic categories: x_i IS $x_{i,k}$ $\quad k \in \{1, \ldots, m\}$,
- membership functions:

$$\varphi_{i,k}(x) = \frac{1}{1 + \left(\dfrac{x - x_{i,k}}{a} \right)^{2b}} \quad , \tag{7}$$

- rules:

$$\text{IF} \quad x_i \quad \text{IS} \quad x_{i,k} \quad \text{THEN} \quad c_i = p_{i,k} x_i + q_{i,k}$$

$$p_{i,k} = \frac{y_k}{x_{i,k}}, \quad q_{i,k} = 0 \quad \text{if} \ x_{i,k} \neq 0$$

$$p_{i,k} = 0, \quad q_{i,k} = y_k \quad \text{if} \ x_{i,k} = 0 \tag{8}$$

$$k \in \{1, \ldots, m\}.$$

The action curve given by the output c_i of this system for the input data - $x_{i,k}$ generalises information coded by $x_{i,k} \to y_k$. The degree of this generalisation depends on parameter a . As we want to cover the whole range of the *i-th* input by a few (say 3 or 5) membership functions, it is reasonable to take such a that the set in which a membership function is 'active', say $\{x \colon \varphi_{i,k}(x) > 0.3\}$ covers 10-30% of the

interval $\left[\min_k(x_{i,k}), \max_k(x_{i,k})\right]$. It may be proved that the shape of the action curve of SIFM is robust to single outliers in the measured data and to the measurement noise. If the *i-th* input is inessential the curve generated by corresponding SIFM will be flat, if it is meaningful the curve will cover significant part of $\left[\min_k(y_k), \max_k(y_k)\right]$.

Building SIFM for every input, plotting and testing it's action curve $A_i = \{(x_{i,k}, c_i(x_{i,k})) \mid k \in \{1, \cdots, m\}\}$ we are able to classify the importance of the input and to select N significant inputs with corresponding curves. We must remember that sometimes data symmetry may lead to false diagnosis that the input is irrelevant one. In this case other techniques for input selection must be used [8], or the modelling domain should be narrowed, or nonlinear data transformation may be applied.

Selection of membership functions for each input is based on piece-wise linear approximation of action curves derived above. Uniform or mean-square approach are both applicable. The input data have to be sorted in ascending order and repeated values must be removed – it is important especially for grid-type data. It is not necessary to obtain continuous piece-wise linear approximation. Mean square approximation means that starting from the left we are to approximate (by a linear function) maximal number of subsequent points $(x_{i,k}, c_i(x_{i,k}))$, under the condition that the mean-square error is limited by a given parameter δ. Uniform approximation consists in placing the longest possible line segments ('sticks') inside a δ-wide 'tube' surrounding the curve $A_i = \{(x_{i,k}, c_i(x_{i,k})) \mid k \in \{1, \cdots, m\}\}$ on the plane. Both approaches are easy to implement numerically and produces equivalent results. Of course design parameter δ influences the number of intervals and so the number of membership functions. As the result of piece-wise linear approximation for the *i*-th significant input we obtain m_i intervals

$$I_{i,j} = \left(x_{\min i,j}, x_{\max i,j}\right) \quad j \in \{1,2,\cdots,m_i\} \ . \tag{9}$$

and a linear polynomial

$$P_{i,j}(x_i) = p_{1\,i,j}x_i + p_{0\,i,j}, \quad x_i \in \left(x_{\min i,j}, x_{\max i,j}\right) \quad j \in \{1,2,\cdots,m_i\} \ . \tag{10}$$

for each interval.

The next step is to build a neuro-fuzzy (ANFIS) model to imitate curve $A_i = \{(x_{i,k}, c_i(x_{i,k})) \mid k \in \{1, \cdots, m\}\}$ with small number of membership functions. The proposed membership functions for the *i*-th significant input will be m_i generalized bell-shape functions $\mu_{i,j}(x)$ with parameters

$$a_{i,j} = \frac{1}{2}\left(x_{\max i,j} - x_{\min i,j}\right), \quad c_{i,j} = \frac{1}{2}\left(x_{\max i,j} + x_{\min i,j}\right), \quad j \in \{1,2,\cdots,m_i\} \ . \tag{11}$$

So the function $\mu_{i,j}(x)$ is spanned over $I_{i,j}$ and centred at the middle point of $I_{i,j}$. The choice of the third parameter $b_{i,j}$ is arbitrary – it is reasonable to start with

$b_{i,j} = 1.5$ for all membership functions. The rules for the proposed SISO neuro-fuzzy model will be:

$$IF \quad x_i \quad IS \quad \mu_{i,j} \quad THEN \quad c_i = p_{1i,j}x_i + p_{0i,j} \quad , \qquad j \in \{1,2,...,m_i\}, \tag{12}$$

where starting values of parameters are taken from piecewise linear approximation results and are given by (10) and (11). Each rule describes a linear, local fuzzy prototype of fuzzy-filtered one-dimensional data.

Single input single output neuro-fuzzy models proposed for each significant input are trained using the data $(x_{i,k}, c_i(x_{i,k}))$, $k \in \{1,\cdots,m\}$. As the starting values of parameters (10, 11) are carefully chosen and as we train one-dimensional model, the training is fast and the results are accurate. Finally, after the training we get N one-dimensional neuro-fuzzy models, the i-th one is equipped with

- m_i membership functions $\mu_{i,j}(\cdot)$ with optimised parameters $(\hat{a}_{i,j}, \hat{b}_{i,j}, \hat{c}_{i,j})$,

- m_i rules $IF \quad x_i \quad IS \quad \mu_{i,j} \quad THEN \quad c_i = \hat{p}_{1i,j}x_i + \hat{p}_{0i,j}$ with optimised parameters $(\hat{p}_{1i,j}, \hat{p}_{0i,j})$.

The multi-input model, which will be a starting point for the final training is equipped with $N_R = \prod\limits_{i=1}^{N} m_i$ rules:

$$IF \ (x_1 \ IS \ \mu_{1,j_1}) \ AND...AND \ (x_N \ IS \ \mu_{N,j_N}) \ THEN \ y = \hat{p}_{11,j_1}x_1 + ... + \hat{p}_{1N,j_N}x_N \cdot$$

$$q_r = \frac{1}{N}\left(\hat{p}_{01,j_1} + ... + \hat{p}_{0N,j_N}\right), \qquad j_i \in \{1,2,\cdots,m_i\} \tag{13}$$

and membership functions parameters are $(\hat{a}_{i,j}, \hat{b}_{i,j}, \hat{c}_{i,j})$. The above neuro-fuzzy inference system is trained by a selected procedure to obtain the final model.

4 Examples

The results of the proposed approach were compared with three standard methods of generation initial structure of fuzzy inference system from numerical data applied in Matlab [11]:

- *genfis1* - generates a FIS structure from a training data set, using a grid partition on the data (no clustering), number and type of membership functions is given by the user;
- *genfis2* - generates Fuzzy Inference System structure from data using subtractive clustering, cluster radii is given by the user;
- *genfis3* - generates Fuzzy Inference System structure from data using FCM clustering, number of clusters is given by the user.

User-defined parameters were chosen to obtain the same number of membership functions as generated automatically by the proposed method.

We are to model two 2-dimmensional functions:

a) $y = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$ (Riastragin function),

b) $y = (x_1^2 + \sin(6x_2))^2$

over $[0,1] \times [0,1]$. We generate 400 points of grid data in $[0,1] \times [0,1]$ and calculate corresponding values of y. We make a false assumption that the third input influences the function – we consider the triples (x_1, x_2, x_3) where value of x_3 is generated on random from $[0,1]$. As we see in both cases the curve A_3 is flat - we can eliminate input x_3 as an inessential one. Piece-wise linear mean square approximation was applied and for $\delta = 0.001$ numbers of membership functions was: for problem a) $[m_1, m_2] = [6,6]$, for problem b) $[m_1, m_2] = [2,3]$. Because of parameters generated by the linear, local prototype approach, the initial position of one-dimensional FIS is close to the desired one and the training is fast and effective. Finally the initial FIS was generated according to (13) and compared with those generated by *genfis1-3*.
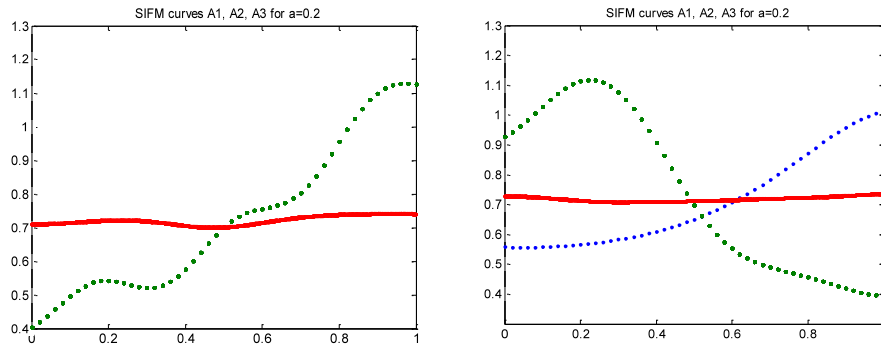
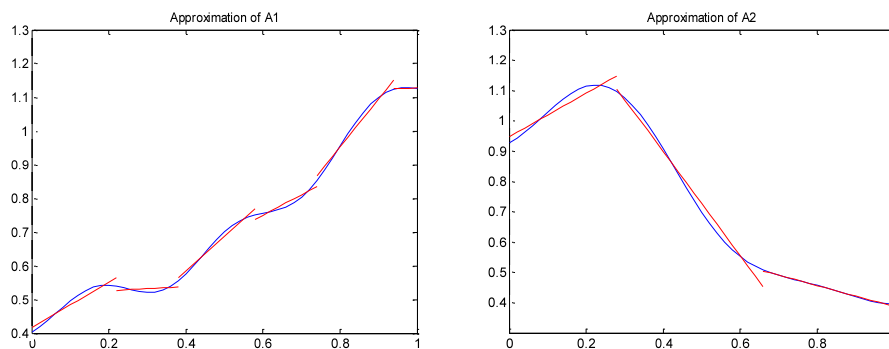**Fig.1.** Single-input fuzzy model curves *left*: function a), right: function b)

Fig.2. Approximation of single-input fuzzy model curves *left*: function a), *right*: function b).

For both functions the starting error as well as the whole learning performance was superior in case of the proposed method. The design procedure is illustrated in fig. 1-5.
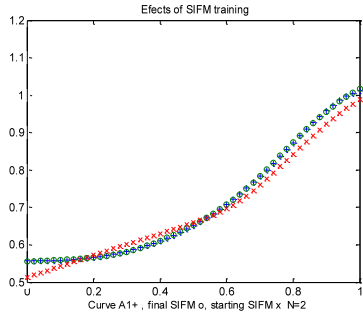
Fig.3. Training of a single-input fuzzy model to the action curve, function b).
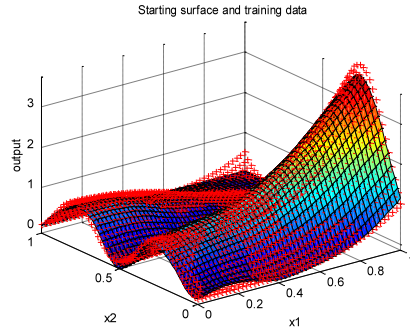

Fig.4. Action surface of an initial FIS (before the final training) and numerical data +, function b).
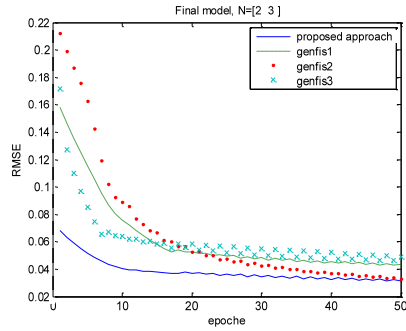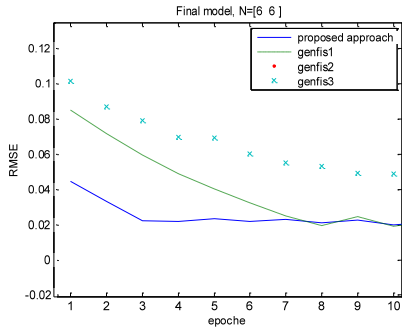

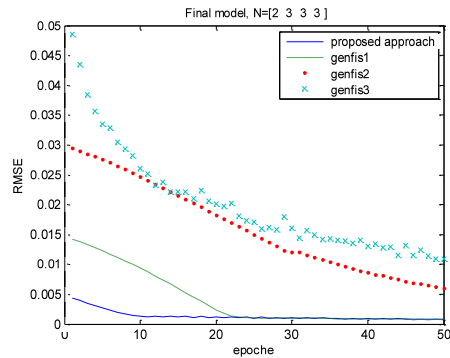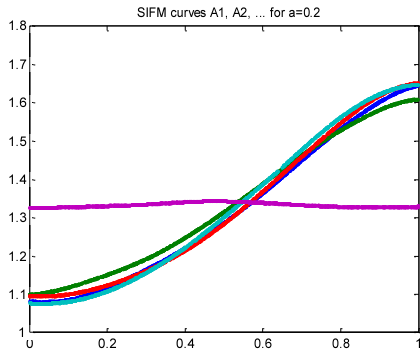Fig.5. Final model training: left function a), right function b).

Fig.6. Modeling a multidimensional ball: *left* - single-input fuzzy curves, *right* - final model training

The next task was modeling a section of a multidimensional ball with one 'dumb' input. 1000 points were generated on random in the unit multi-dimensional cube. The results are presented in fig. 6. Again the proposed method is better than the standard clustering procedures.

5 Conclusions

We propose simple and effective procedure for creating neuro-fuzzy TSK models of complex systems from the numerical input-output data. Significant inputs, number of membership functions for each input and initial values of all system parameters are set automatically based on accuracy requirements. Because the initial structure and parameters are set properly, we need a few training iterations for the neural network representation of our model to converge. The generated FIS may be used as a starting point to any training algorithms – also those incorporating reduction of fuzzy rules what is highly recommended, although was not discussed here. The proposed technique was compared with clustering methods implemented in Matlab. Results obtained from the proposed method provided several times smaller starting training error, leading to simpler models, smaller errors and reduction of necessary training time. Several important problems were not discussed here, for example elimination of relevant but dependent inputs.

References

1. Jang JR.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23, 665 – 684 (1993)
2. Buyukbingola E., Sismanb A., Akyildizc M., Alparslanb N.F and Adejared A.: Adaptive neuro-fuzzy inference system (ANFIS): A new approach to predictive modeling in QSAR applications: A study of neuro-fuzzy modeling of PCP-based NMDA receptor antagonists, Bioorganic & Medicinal Chemistry, 15, 4265-4282, (2007)
3. Kovac N., Bauk S.: The ANFIS-based route preference estimation in sea navigation, Journal of Maritime Research, III, 69-86 (2006)
4. Alcalá R., Alcalá-Fdez J., Casillas J., Cordón O., F. Herrera F., Local Identification of Prototypes for Genetic Learning of Accurate TSK Fuzzy Rule-Based Systems, International Journal Of Intelligent Systems, 22, 909–941 (2007)
5. Cheng-Jian Lin, Shang-Jin Hong, The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition, Neurocomputing 71, 297–310 (2007)
6. Gal L., Botzheim J., Koczy L.T., Modified bacterial memetic algorithm used for fuzzy rule base extraction, Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology, 425-431 (2008)
7. Jang JR.: Input selection for ANFIS learning, IEEE Int. Conf. Fuzzy Systems, 2, 1493-1499 (1996)
8. Chengulin Hu, Feng Wan, Input selection in learning systems: a brief review of some important issues and recent developments, Proceedings of the 18th international conference on Fuzzy Systems, 530-535 (2009)
9. Lin Y., Cunningham III G.A., Coggeshal S.V., Using fuzzy partitions to create fuzzy systems from input-output data and set the initial weights in a fuzzy neural network, IEEE Trans. on Fuzzy Systems , 5, 614-621 (1997)
10. Kabziński J. Woźniak P., Kuźmiński K., Effective selection of neuro-fuzzy Sugeno model architecture, 97-100 Proc. of 17-th IASTED International Conf. Modelling, Identification & Control (1998)
11. The MathWorks, Inc. http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/ genfis1.html, genfis2.html, genfis3.html,