Proceedings of the International Symposium on Evolving Intelligent Systems, Plamen Angelov, Dimitar Filev, and Nikola Kasabov (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Clustering to train an evolving radial basis function European Conference on Artificial Intelligence

Jose de Jesus Rubio[1] and **Jaime Pacheco**[1] and **Raul Rivera**[1]

Abstract. In this paper, we propose the backpropagation algorithm to train online an evolving radial basis function. Structure and parameter learning are updated at the same time in our algorithm, we do not make difference in structure learning and parameter learning. It generate groups with an online clustering. The center is updated in order to get that the center is near to the incoming data in each iteration, in this way, It does not need to generate a new rule in each iteration, i.e., it does not generate many rules and It does not need to prune the rules. We give a time varying learning rate for backpropagation training in the parameters.

1 Introduction

In the last few years, the application of fuzzy neural networks to nonlinear system identification is very active area [10],[11]. Fuzzy modeling involves structure and parameters identification. The second one is usually addressed by some gradient descent variant, e.g., the least square algorithm and backpropagation. The online identification is addressed in this paper.

In on line identification, model structure and parameter identification are updated immediately after each input-output pair has been presented, that is, after each iteration. On line identification also includes; i) model structure and ii) parameter identification. There are a few online methods in the literature. In [7] the input space is partitioned according to a aligned clustering-based algorithm. After the number of rules is decided, the parameters are tuned by recursive least square algorithm, it is called SONFIN. In [8] it is the recurrent case of the above case, it is called RSONFIN. In [13] the input space is automatically partitioned into fuzzy subsets by adaptive resonance theory mechanism. Fuzzy rules that tend to give high output error are split in two, by a specific fuzzy rule splitting procedure. In [9] he proposes that the radius to make clustering updates.

In this paper, we propose the backpropagation algorithm to train online an evolving radial basis function. Structure and parameter learning are updated at the same time in our algorithm, we do not make difference in structure learning and parameter learning. It generate groups with an online clustering. The center is updated in order to get that the center is near to the incoming data in each iteration, in this way, It does not need to generate a new rule in each iteration, i.e., it does not generate many rules and It does not need to prune the rules. We give a time varying learning rate for backpropagation training in the parameters.

[1] Seccion de Estidios de Posgrado e Investigacion - ESIME Azcapotzalco - Instituto Politecnico Nacional, Mexico, email: jrubioa@ipn.mx

2 Fuzzy neural networks for nonlinear identification

Consider following unknown discrete-time nonlinear system:

$$y(k-1) = f\left[X(k-1)\right] \tag{1}$$

Where $X(k-1) = [x_1(k-1)\dots x_N(k-1)] = [y(k-2),\dots,y(k-n-1),u\,(k-2)\,,\ \dots,u\,(k-m-1)] \in \Re^N$ $(N = n+m)$ is the input vector, $|u(k-1)|^2 \leq \overline{u}$, $y(k-1)$ is the output of the plant, f is general nonlinear smooth function $f \in C^\infty$. In the case of [6], we consider the radial basis function neural network as:

$$\begin{aligned} \widehat{y}(k-1) &= a(k-1)/b(k-1) \\ a(k-1) &= \sum_{j=1}^{M} v_j(k-1) z_j(k-1), \\ b(k-1) &= \sum_{j=1}^{M} z_j(k-1) \\ z_j(k-1) &= \prod_{i=1}^{N} \exp\left[-(\tfrac{x_i(k-1)-c_{ij}(k-1)}{\sigma_{ij}(k-1)})^2\right] \end{aligned} \tag{2}$$

Where $x_i(k-1)$ are inputs of system (1), $(i = 1\dots N)$, $c_{ij}(k-1)$ and $\sigma_{ij}(k-1)$ are the centers and the widths of the gaussian function, respectively, $(j = 1\dots M)$, $v_j(k-1)$ is the output of the gaussian function.

3 Structure identification

We use the online clustering to train the structure of the algorithm. Choososing an appropriate number of hidden neurons is important in designing the online clustering for neuro fuzzy systems, because too many hidden neurons result in a complex evolving system that may be unnecessary for the problem and it can cause overfitting [6], whereas too few hidden neurons produce a less powerful neural system that may be insufficient to achieve the objective. We view the number of hidden neurons as a design parameter and determine it based on the input-output pairs and on the number of elements of each hidden neuron.

The basic idea is to group the input-output pairs into clusters and use one hidden neuron for one cluster as in [9], [14]; that is, the number of hidden neurons equals the number of clusters.

One of the simplest clustering algorithms is the nearest neighborhood clustering algorithm. In this algorithm, first we put the first data as the center of the first cluster. Then, if the distances of a data to the cluster centers are less than a pre-specified value (the radius r), put this data into the cluster whose center is the closest to this data; otherwise, set this data as a new cluster center. The details are given as follows.

Let $x_i(k-1)$ are newly incoming pattern, then we get:

$$p(k-1) = \max_{1 \le j \le M} z_j(k-1) \tag{3}$$

If $p(k-1) < r$, then a new rule is generated (each rule correspond to each center) and $M = M+1$ where r is a selected radius, $r \in (0,1)$. Once a new rule is generated, the next step is to assign initial centers and widths of the corresponding membership functions.

$$\begin{array}{c} c_{i,M+1}(k) = x_i(k) \quad v_{M+1}(k) = y(k) \\ \sigma_{i,M+1}(k) = rand \in (0,1) \end{array} \tag{4}$$

If $p(k-1) \ge r$, then a rule is not generated and in the case that $z_j(k-1) = p(k-1)$ we have the winner rule j^*, the centers of this rule are updated as:

$$\begin{array}{c} c_{ij^*}(k) = c_{ij^*}(k-1) \\ + \frac{1}{1+x_i^2(k-1)+c_{ij^*}^2(k-1)} \left(x_i(k-1) - c_{ij^*}(k-1)\right) \end{array} \tag{5}$$

4 Parameter identification

We define the identification error as:

$$e(k-1) = \widehat{y}(k-1) - y(k-1) \tag{6}$$

If we define:

$$\begin{array}{c} D_{1ij}(k-1) \\ = \frac{2\left[v_j(k-1) - \widehat{y}(k-1)\right] z_j(k-1) \left[x_i(k-1) - c_{ij}(k-1)\right]}{b(k-1)\sigma_{ij}^2(k-1)} \\ D_{2ij}(k-1) \\ = \frac{2\left[v_j(k-1) - \widehat{y}(k-1)\right] z_j(k-1) \left[x_i(k-1) - c_{ij}(k-1)\right]^2}{b(k-1)\sigma_{ij}^3(k-1)} \\ D_{3j}(k-1) = \frac{z_j(k-1)}{b(k-1)} \end{array} \tag{7}$$

We use the following learning law to updated the weights of neural identifier:

$$\begin{array}{l} c_{ij}(k) = c_{ij}(k-1) \\ -\eta(k-1) D_{1ij}(k-1) e\,(k-1) \\ \sigma_{ij}(k) = \sigma_{ij}(k-1) \\ -\eta(k-1) D_{2ij}(k-1) e\,(k-1) \\ v_j(k) = v_j(k-1) \\ -\eta(k-1) D_{3j}(k-1) e\,(k-1) \end{array} \tag{8}$$

Where $j = 1 \ldots M$, $i = 1 \ldots N$ and $D_{1ij}(k-1)$, $D_{2ij}(k-1)$, and $D_{3j}(k-1)$ are given in (7). The dead-zone is applied to $\eta(k-1)$ as:

$$\eta(k-1) = \begin{cases} \frac{\eta}{1+q(k-1)} & \text{if } e^2\,(k-1) \ge \frac{\overline{\zeta}^2}{1-\eta} \\ 0 & \text{if } e^2\,(k-1) < \frac{\overline{\zeta}^2}{1-\eta} \end{cases} \tag{9}$$

Where $q(k-1) = D_{1ij}^2(k-1) + D_{2ij}^2(k-1) + D_{3j}^2(k-1)$, $0 < \eta \le 1$.

5 The proposed algorithm

The proposed algorithm is as follows:

0) Select the following parameters: $0 < r < 1 \in \Re$, $0 < \eta < 1 \in \Re$, $\overline{\zeta}^2 \in \Re$.

1) For the first data $k = 1$, $M = 1$, $v_1(1) = y(1)$, $(i = 1 \ldots N)$, $c_{i1}(1) = x_i(1)$, $\sigma_{i1}(1) = rand \in (0,1)$.

2) For the other data $k \ge 2$, evaluate $z_j(k-1)$ and $b(k-1)$ with (2), evaluate $\widehat{y}(k-1)$ with (2), evaluate $p(k-1)$ with (3), update $c_{ij}(k)$, $\sigma_{ij}(k)$ and $v_j(k)$ with (7), (8) and (9). where $D_{1ij}(k-1)$, $D_{2ij}(k-1)$, and $D_{3j}(k-1)$ are given in (7).

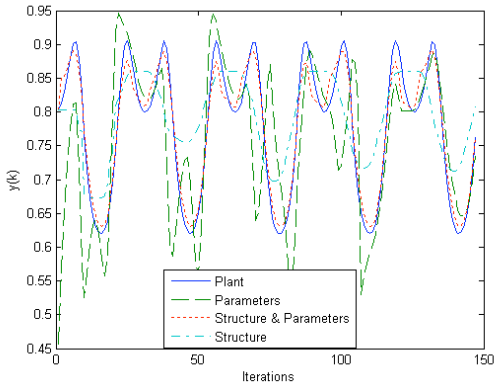

Figure 1. Identification results for the example 1

3) if $p(k-1) < r$, then a new rule is generated ($M = M+1$) where $r \in (0,1)$, assign initial values to $c_{i,M+1}(k)$, $\sigma_{i,M+1}(k)$, and $v_{M+1}(k)$ with (4), go to 2.

4) If $p(k-1) \ge r$, then a rule is not generated and in the case that $z_j(k-1) = p(k-1)$ we have the winner rule j^*, this rule is updated with (5), go to 2.

6 Simulation

In this section, the suggested online evolving proposed algorithm is applied to nonlinear system identification. In both examples the least mean square error is used for the comparison results. The least mean square error is defined as [9]:

$$MSE = \frac{1}{N} \sum_{k=1}^{N} e^2(k-1) \tag{10}$$

Example 1 Consider the nonlinear system given in [14]:

$$y(k) = 0.52 + \frac{0.1x_1 + 0.28x_2}{x_1^2 + x_2^2} - 0.06x_1x_2 \tag{11}$$

with $x_1(k) = \sin^2(10/k)$ and $x_2(k) = \cos^2(10/k)$. We compare our algorithm called Structure & parameters using the parameters $\eta = 0.9$, $r = 0.6$ with the case where only the structure is trained called Structure using the parameter $r = 0.6$ and with the case where the only the parameters are trained called Parameters using the parameter $\eta = 0.3$ and 2 neurons in the hidden layer. The comparison results of the identification are given in Figure 1 and in Table 1.

Table 1: Results for Example 1

Methods	Rules	MSE
Structure	4	0.0023
Parameters	2	0.0045
Structure & parameters	2	1.6409×10^{-4}

The least mean square error for the Structure algorithm is 0.0023, for Paremeters algorithm is 0.0045 and for Structure & parameters

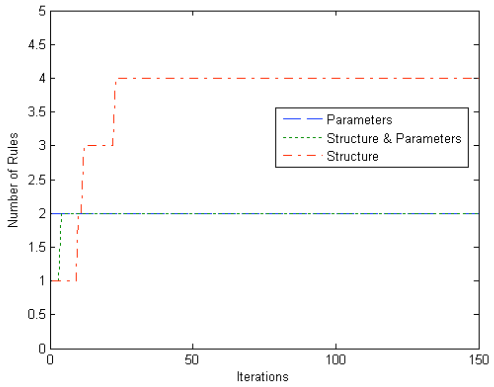

Figure 2. Number of rules for example 1

algorithm is 1.6409×10^{-4}. The Structure algorithm generate 4 rules and the Structure & parameters algorithm generate 2 rules. In Figure 2, the comparison results for the number of rules is shown.

Form the simulation results shown in example 1 in it can be seen that the proposed algoritghm has a good behavior for the identification of nonlinear systems.

7 Conclusion

In this paper we presented a quick and efficient approach for system modeling using the the backpropagation algorithm to train an evolving radial basis function network. The proposed neural fuzzy network uses the online clustering to train the structure, the backpropagation to train the parameters. The structure identification and parameter learning are done online.

REFERENCES

[1] M.F. Azem, M.Hanmandlu and N. Ahmad, Structure identification of generalized adaptive neuro-fuzzy inference systems, *IEEE Trans. on Fuzzy Syst.*, Vol.11, No.6, 666-681, 2003.

[2] M. Brown, C.J. Harris, *Adaptive Modelling and Control*, Macmillan Pub.Co., Prentice Hall, New York, 1994.

[3] S.L. Chiu, Fuzzy Model Identification based on cluster estimation, *Journal of Intelligent and Fuzzy Systems*, Vol.2, No.3, 1994.

[4] J.R. Hilera, V.J. Martines, *Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones*, Adison Wesley Iberoamericana, U.S.A , 1995.

[5] J. S. R. Jang, AFNFIS: Adaptive-Network-Based Fuzzy inference system, *IEEE Trans. on Systems, Man and Cybernetics*, Vol.23, 665-685, 1993.

[6] J. S. R. Jang and C. T. Sun, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, NJ 07458, 1997.

[7] C.F. Juang and C.T. Lin, An On-line Self Constructing Nural Fuzzy Inference Network and Its Applications, *IEEE Trans. on Fuzzy Syst.*, Vol.6, No.1, 12-32, 1998.

[8] C.F. Juang and C.T. Lin, A Recurrent Self-Organizing Fuzzy Inference Network, *IEEE Trans. on Neural Networks*, Vol.10, No.4, 828-845, 1999.

[9] N. Kasabov, Evolving Fuzzy Nural Networks for Supervised/Unsupervised Online Knowledge-Based Learning, *IEEE Trans. on Systems, Man and Cybernetics*, Vol.31, No.6, 902-918, 2001.

[10] C.T.Lin, *Neural Fuzzy Control Systems with Structure and parameter learning*, New York: World Scientific, 1994.

[11] S. Mitra, Y. Hayashi, Neuro-Fuzzy rule Generation: Survey in Soft Computing Framework, *IEEE Trans. on Neural Networks*, Vol.11, No.3, 748-769, 2000.

[12] I. Rivals and L. Personnaz, Neural Network Construction and Selection in non linear modelling, *IEEE Trans. on Neural Networks*, Vol.14, No.4, 804-820, 2003.

[13] S.G.Tzafestas and K.C. Zikidis, On-Line Neuro-Fuzzy ART-Based Structure and parameter learning TSK Model, *IEEE Trans. on Systems, Man and Cybernetics*, Vol.31, No.5, 797-803, 2001.

[14] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ 07458, 1997.