Non-Fuzzy Rule-Networks Based on Hard Clustering Algorithm

Keon-Jun Park, Jun-Myung Lee, Jung-Won Choi and Yong-Kab Kim*,

Department of Information and Communication Engineering,
Wonkwang University, 344-2, Shinyong-dong, Iksan-si, Chonbuk, 570-749 South Korea bird75@wonkwang.ac.kr, ykim@wonkwang.ac.kr

Abstract. A new design of non-fuzzy-networks based on hard c-means (HCM) are introduced in this paper. To generate the ruels and design the networks, we use HCM clustering algorithm. The premise part of the rules of the proposed networks is expressed by the hard partition of input space generated by HCM clustering algorithm. The partitioned local spaces indicate the rules of the proposed networks. The consequence part of the rule is represented by polynomial functions. And back-propagation algorithm is used to learn the coefficients of the polynomial functions. The proposed networks are evaluated with the example for nonlinear process.

Keywords: Fuzzy Neural Networks (FNNs), Non-Fuzzy Rule, hard partition, Hard c-means clustering algorithm, Nonlinear process.

1 Introduction

Fuzzy-neural hybridization comes from a hybrid intelligent model that combines of fuzzy inference systems and neural networks [1, 2, 3]. Fuzzy neural networks (FNNs) refer to synergize these two hybrid techniques. There have been many approaches to synthesize and apply for these fields. Typically, FNNs are represented by fuzzy “if-then” rules while the back propagation (BP) is used to optimize the parameters. The designers find it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data. The generation of the fuzzy rules and the adjustment of its membership functions were done by trial and error and/or operator’s experience.

In this paper, we propose a new design of non-fuzzy rules-networks by means of hard partition of input space using hard c-means (HCM) clustering algorithm [4]. The premise part of the rules is realized with the aid of the hard partition of input space generated by HCM clustering algorithm. The consequence part of the rule is represented by polynomial functions. And the coefficients of the polynomial functions are learned by BP algorithm. The proposed networks are evaluated through the numeric experimentation for nonlinear process.

* Corresponding author : ykim@wonkwang.ac.kr

2 Non-Fuzzy Rule-Networks

2.1 The structure of the NFR-based Networks

The structure of the NFR-based Networks involves HCM clustering algorithm in the premise part and neural networks present in the consequence part of the rules.

The proposed NFR-based Networks are implied by the hard scatter partition of input spaces. In this sense, each rule can be viewed as a certain rule of the following format.

$$R_j : If\ x_1\ and \cdots and\ x_d\ is\ H_j\ Then\ y_j = f(x_1, \cdots, x_d). \quad (1)$$

As far as inference schemes are concerned, we distinguish these cases:

Type 1 (Simplified Inference): $f = w_{j0}$

Type 2 (Linear Inference): $f = w_{j0} + \sum_{k=1}^{d} w_{jk} x_k$

Type 3 (Quadratic Inference):

$$f = w_{j0} + \sum_{k=1}^{d} w_{jk} x_k + \sum_{k=1}^{d} w_{j(d+k)} x_k^2 + \sum_{k=1}^{d} \sum_{l=k+1}^{d} w_{jz} x_k x_l$$

Type 4 (Modified Quadratic Inference):

$$f = w_{j0} + \sum_{k=1}^{d} w_{jk} x_k + \sum_{k=1}^{d} \sum_{i=k+1}^{d} w_{jz} x_k x_i .$$

To be more specific, R^j is the j-th fuzzy rule, while H_j denotes j-th membership grades using HCM clustering algorithm. w's are consequent parameters of the rule. The functionality of each layer is described as follows.

[Layer 1] The nodes in this layer transfer the inputs.

[Layer 2] The nodes here are used to calculate the membership degrees using HCM clustering algorithm.

[Layer 3] The nodes in this layer realize a certain inference process.

$$h_j = \alpha_j y_j . \quad (2)$$

[Layer 4] The nodes in this layer compute the outputs.

$$\hat{y} = \sum_{j=1}^{n} h_j . \quad (3)$$

2.2 The learning algorithm

The parametric learning of the network is realized by adjusting connections of the neurons and as such it could be realized by running a standard back-propagation (BP) algorithm. The performance index is based on the Euclidean distance. As far as learning is concerned, the connections are adjusted in a standard fashion.

Quite commonly to accelerate convergence, a momentum coefficient β is being added to the learning expression.

3 Experimental Studies

In this section, we discuss numerical example to apply to the nonlinear process. This time series data (296 input-output pairs) coming from the gas furnace nonlinear process has been intensively studied [5]. The delayed terms of methane gas flow rate $u(t)$ and carbon dioxide density $y(t)$ are used as six input variables organized in a vector format as $[u(t\text{-}3), u(t\text{-}2), u(t\text{-}1), y(t\text{-}3), y(t\text{-}2), y(t\text{-}1)]$. $y(t)$ is the output variable. The first part of the data set (consisting of 148 pairs) was used for training purposes. The remaining part of the series serves as a testing data set. We consider the MSE as a performance index. We construct the networks for two dimensional system with $u(t\text{-}3)$, $y(t\text{-}1)$ and also construct the networks for six-dimensional system with the entire inputs.

Table 1 summarizes the performance index for training and testing data by setting the number of clusters and inference type for two and six dimensional systems. Here, PI and E_PI stand for the performance index for the training data set and the testing data set, respectively.

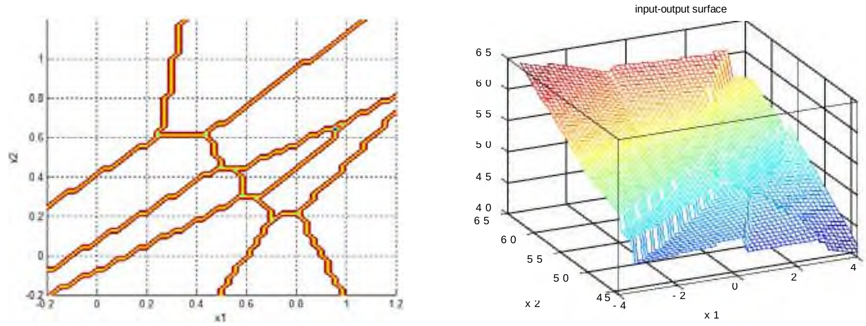
Table 1. Performance index for the proposed networks

No. of Rules	Type	2D		6D	
		PI	E_PI	PI	E_PI
2	Simplified	3.589	3.804	3.779	4.641
	Linear	0.025	0.344	0.026	0.385
	Quadratic	0.078	0.426	0.110	0.616
	M Quadratic	0.030	0.365	0.052	0.446
3	Simplified	1.921	2.205	2.131	2.733
	Linear	0.022	0.354	0.023	0.238
	Quadratic	0.035	0.372	0.132	0.506
	M Quadratic	0.026	0.399	0.044	0.304
5	Simplified	1.035	1.703	1.936	3.263
	Linear	0.023	0.364	0.034	0.576
	Quadratic	0.025	0.343	0.041	0.610
	M Quadratic	0.021	0.366	0.051	0.406
10	Simplified	0.675	1.415	1.298	2.580
	Linear	0.072	0.310	0.067	0.630
	Quadratic	0.117	0.291	0.106	0.757
	M Quadratic	0.065	0.298	0.075	0.614

From the Table 1, we selected the best network with ten rules (clusters) with modified quadratic inference that exhibits PI = 0.117 and E_PI = 0.291 for two-dimensional system.

From the same table, we also selected the best network with three rules (clusters) with linear inference that exhibits PI = 0.023 and E_PI = 0.238 for six-dimensional system. From the results, we constructed the non-fuzzy rule-network that had a small number of rules in high dimensional problem because the generation of the rules depends on the clusters.

Fig. 1 shows 10 hard-partitioned input spaces using HCM clustering algorithm and input-output space in the selected model in two-dimensional system.



(a) Hard partitioned input spaces (b) Input-output space

Fig. 1. Partitioned input spaces and input-output space for the selected network in two-dimensional system.

4 Conclusion

This paper has introduced the non-fuzzy rule-networks based on hard-partition of input space to generate the rules of the system for nonlinear process. The input spaces of the proposed networks were divided as the hard partition using HCM clustering algorithm. By this method, we constructed the networks that are compact and simple in high dimension. From the results, we were able to design the preferred networks. Through the use of a performance index, we were able to achieve a balance between the approximation and generalization abilities of the resulting networks.

References

1. Yamakawa,T.: A Neo Fuzzy Neuron and Its Application to System Identication and Predition of the System Behavior. Proceeding of the 2nd International Conference on Fuzzy logic & Neural Networks, (1992) 447--483
2. Buckley, J. J. and Hayashi, Y.: Fuzzy neural networks: A survey. Fuzzy Sets Syst. 66, 1--13, (1994)
3. Jang, J.S.R., Mizutani, E. and Sun, C.T.: Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence, Prentice Hall, NJ, 1997.
4. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, (1981)
5. Box, G. E. P., Jenkins, G. M.: Time Series Analysis: Forecasting and Control, 2nd ed., Holden-Day, San Francisco, CA (1976)