Proceedings of KFIS Spring Conference 2004 Volume 14, Number 1

다층 퍼지뉴럴 네트워크의 설계

Design of Multi-layer Fuzzy Neural Networks

박병준, 박건준, 오성권
원광대학교 전기전자 및 정보공학부

Byoung-Jun Park, Keon-Jun Park, Sung-Kwun Oh
Department of Electrical Electronic and Information Engineering, Won-Kwang University, 344-2, Sinyong-Dong, IKSAN, JEONBUK, KOREA, 570-749
E-mail : ohsk@wonkwang.ac.kr

요 약

In this study, a new architecture and comprehensive design methodology of genetically optimized Multi-layer Fuzzy Neural Networks (gMFNN) are introduced and a series of numeric experiments are carried out. The gMFNN architecture results from a synergistic usage of the hybrid system generated by combining Fuzzy Neural Networks (FNN) with Polynomial Neural Networks (PNN). FNN contributes to the formation of the premise part of the overall network structure of the gMFNN. The consequence part of the gMFNN is designed using PNN.

1. Introduction

Efficient modeling techniques should allow for a selection of pertinent variables and a formation of highly representative datasets. The omnipresent modeling tendency is the one that exploits techniques of Computational Intelligence (CI) by embracing fuzzy modeling [1,2], neurocomputing [3], and genetic optimization [4].

In this study, we develop a new architecture and comprehensive design methodology, called genetically optimized Multi-layer Fuzzy Neural Networks (gMFNN). In a nutshell, gMFNN is composed of two main substructures driven to genetic optimization, namely a fuzzy set-based fuzzy neural network (FNN) and a polynomial neural network (PNN). From a standpoint of rule-based architectures, one can regard the FNN as an implementation of the antecedent part of the rules while the consequent is realized with the aid of a PNN. The role of the FNN is to interact with input data, granulate the corresponding input spaces(viz. converting the numeric data into representations at the level of fuzzy sets). In the first case (Scheme I) we concentrate on the use of simplified fuzzy inference. In the second case (Scheme II), we take advantage of linear fuzzy inference. The role of the PNN is to carry out nonlinear transformation at the level of the fuzzy sets formed at the level of FNN. The PNN that exhibits a flexible and versatile structure [5] is constructed on a basis of Group Method of Data Handling (GMDH [5]) and genetic algorithms (GAs). The design procedure applied in the construction of each layer of the PNN deals with its structural optimization involving the selection of optimal nodes (polynomial neurons; PNs) with specific local characteristics (such as the number of input variables, the order of the polynomial, and a collection of the specific subset of input variables) and addresses specific aspects of parametric optimization. To assess the performance of the proposed model, we exploit a well known time series data. Furthermore, the network is directly contrasted with several existing intelligent models.

2. The architecture and development of genetically optimized Multi-layer Fuzzy Neural Networks(gMFNN)

2.1 The premise part; Fuzzy Neural Networks(FNN)

The gMFNN emerges from the genetically optimized multi-layer perceptron architecture based on fuzzy set-based FNN, GAs and GMDH. These networks result as a synergy between two other general constructs such as FNN [9] and PNN [5].

We use FNN based on two types of fuzzy inferences, that is, Scheme I and Scheme II. The output of the FNN is governed by the following expression.

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m) = \sum_{i=1}^{m} f_i(x_i) \qquad (1)$$

We can regard each $f_i(x_i)$ given by (1) as the following rules.

· Scheme I

R^j : If x_i is A_{ij} then $Cy_{ij}=w_{ij}$ (2)

· Scheme II

R^j : If x_i is A_{ij}

then $Cy_{ij}=ws_{ij} + w_{ij}\, x_i$ (3)

w_{ij} is a constant in (2), and ws_{ij} is a constant and w_{ij} is an input variable consequence of the fuzzy rule in (3). Mapping from x_i to $f_i(x_i)$ in (2) is determined by the fuzzy inferences and a standard defuzzification.

$$f_i(x_i) = \sum_{j=1}^{z} \mu_{ij}(x_i) \cdot w_{ij} \Big/ \sum_{j=1}^{z} \mu_{ij}(x_i) \qquad (4)$$

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP [9]). The update formula of a connection in Scheme I is as follow.

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) + \alpha (w_{ij}(t) - w_{ij}(t-1)) \qquad (5)$$

The inference result and the learning algorithm in linear fuzzy inference-based FNN use the mechanisms in the same manner as discussed above. GAs are optimization techniques based on the principles of natural evolution [4]. In order to enhance the learning of the FNN and augment its performance of a FNN, we use GAs to adjust learning rate, momentum coefficient and the parameters of the membership functions of the antecedents of the rules.

2.2 The consequence part; Polynomial Neural Networks(PNN)

When we construct PNs of each layer in the conventional PNN [5], such parameters as the number of input variables (nodes), the order of polynomial, and input variables available within a PN are fixed (selected) in advance by the designer. This could have frequently contributed to the difficulties in the design of the optimal network. To overcome this apparent drawback, we introduce a new genetic design approach; especially as a consequence we will be referring to these networks as genetically optimized PNN. The overall genetically-driven optimization process of PNN is shown in Fig. 1.

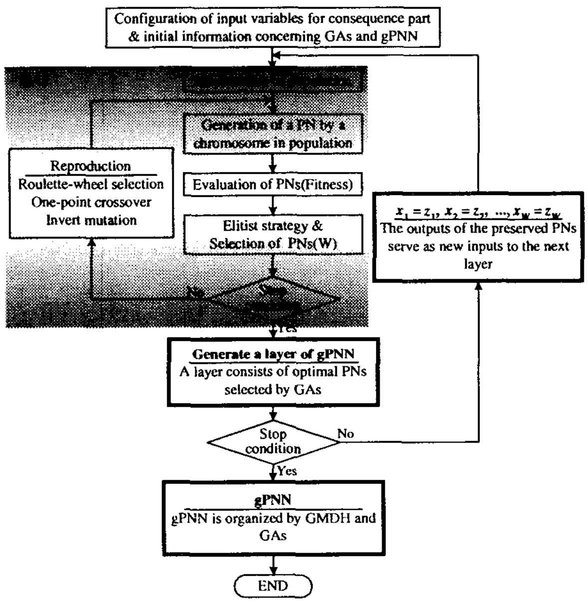

Fig. 1 Overall genetically-driven optimization process of PNN

3. The algorithms and design procedure of genetically optimized Multi-layer Fuzzy Neural Networks

<u>*The premise of gMFNN: FNN*</u>

[Layer 1] Input layer.

[Layer 2] Computing activation degrees of linguistic labels.

[Layer 3] Normalization of a degree activation of the rule.

[Layer 4] Multiplying a normalized activation degree of the rule by connection. If we choose this point(first option) for combining FNN with gPNN, a_{ij} is given as the input variable of the gPNN.

$$a_{ij} = \bar{\mu}_{ij} \times Cy_{ij} = \mu_{ij} \times Cy_{ij} \tag{6}$$

$$\begin{cases} \text{Simplified}: Cy_{ij} = w_{ij} \\ \text{Linear} \quad : Cy_{ij} = ws_{ij} + w_{ij} \cdot x_i \end{cases} \tag{7}$$

[Layer 5] Fuzzy inference for the fuzzy rules. If we choose this point(second option), f_i is the input variable of gPNN.

[Layer 6; Output layer of FNN] Computing output of a FNN.

· *The consequence of gMFNN: gPNN*

[Step 1] Configuration of input variables. If we choose the first option, $x_1=a_{11}$, $x_2=a_{12},\cdots$, $x_n=a_{ij}$ ($n=i\times j$). For the second option, we have $x_1=f_1$, $x_2=f_2,\cdots$, $x_n=f_m$ ($n=m$).

[Step 2] Decision of initial information for constructing the gPNN.

[Step 3] Initialization of population.

[Step 4] Decision of PNs structure using genetic design.

[Step 5] Evaluation of PNs.

[Step 6] Elitist strategy and selection of PNs with the best predictive capability.

[Step 7] Reproduction.

[Step 8] Repeating Step 4-7.

[Step 9] Construction of their corresponding layer.

[Step 10] Check the termination criterion (performance index).

$$E(PI \ or \ EPI) = \frac{1}{n}\sum_{p=1}^{n}(y_p - \hat{y}_p)^2 \tag{8}$$

[Step 11] Determining new input variables for the next layer.

The gPNN algorithm is carried out by repeating Steps 4-11.

4. Experimental studies

The performance of the gMFNN is illustrated with the aid of a time series of gas furnace [10]. The delayed terms of methane gas flow rate, u(t) and carbon dioxide density, y(t) are used as system input variables. We utilizes 3 system input variables such as u(t-2), y(t-2), and y(t-1). The output variable is y(t). Fig. 2 illustrates the optimization process by visualizing the performance index in successive cycles. It also shows the optimized network architecture when taking into consideration gMFNN based on linear fuzzy inference and second option. Table 1 contrasts the performance of the genetically developed network with other fuzzy and fuzzy-neural networks studied in the literatures.

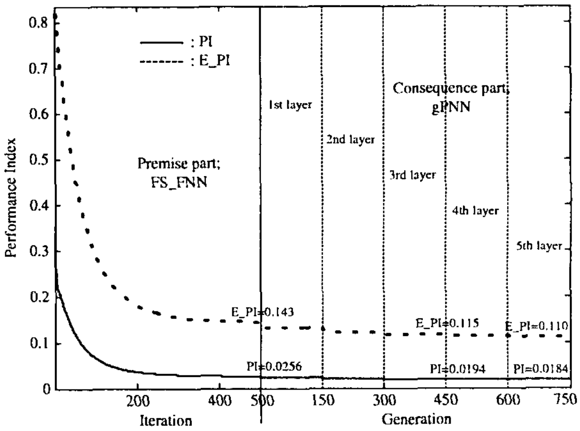

Fig. 2 Optimization procedure of gMFNN by BP learning and GAs

Model			PI	EPI	No. of rules
Kim, et al.'s model[11]			0.034	0.244	2
Lin and Cunningham's model[12]			0.071	0.261	4
Fuzzy	Hybrid[2] (GAs+Complex)	Simplified	0.024	0.329	4(2×2)
		Linear	0.017	0.289	4(2×2)
	HCM+GAs[1]	Simplified	0.022	0.333	6(3×2)
		Linear	0.020	0.264	6(3×2)
FNN [9]		Simplified	0.043	0.264	6(3+3)
		Linear	0.037	0.273	6(3+3)
SOFPNN		Generic [7]	0.017	0.250	4 rules /5th layer
		Advanced [8]	0.019	0.264	6 rules /5th layer
Proposed model (gMFNN)		Scheme I	0.018	0.112	6 rules /5th layer
		Scheme II	0.018	0.110	6 rules /5th layer

Table 4 Comparison of performance with other modeling methods

5. Conclusion

The comprehensive design methodology comes with the parametrically as well as structurally optimized network architecture. 1) As the premise structure of the gMFNN, the optimization of the rule-based FNN hinges on genetic algorithms and back-propagation (BP) learning algorithm: The GAs leads to the auto-tuning of vertexes of membership function, while the BP algorithm helps obtain optimal parameters of the consequent polynomial of fuzzy rules through learning. And 2) the gPNN that is the consequent structure of the gMFNN is based on the technologies of the extended GMDH and GAs: The extended GMDH is comprised of both a structural phase such as a self-organizing and evolutionary algorithm, and a parametric phase of least square estimation-based learning, moreover the gPNN architecture is driven to genetic optimization, in what follows it leads to the selection of the optimal nodes. In the sequel, a variety of architectures of the proposed gMFNN driven to genetic optimization have been discussed. The experiments helped compare the network with other intelligent models - in all cases the previous models came with higher values of the performance index.

감사의 글

이 논문은 2003년도 학술진흥재단의 지원에 의하여 연구되었슴(KRF-2003-002-D00297).

6. Reference

[1] Park B. J., Pedrycz W., and Oh S. K., "Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation", *IEE Proceedings-Control theory and application*, Vol. 148, No. 5, pp. 406-418, 2001.

[2] Oh S. K., Pedrycz W., and Park B. J., "Hybrid Identification of Fuzzy Rule-Based Models", *International Journal of Intelligent Systems*, Vol. 17, No. 1, pp. 77-103, 2002.

[3] Narendra K. S., and Parthasarathy K., "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 2, pp. 252-262, 1991.

[4] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag., 1996

[5] Oh S. K., Pedrycz W., and Park B. J., "Polynomial Neural Networks Architecture: Analysis and Design", *Computers and Electrical Engineering*, Vol. 29, No. 6, pp. 653-725, 2003.

[6] Ivahnenko A. G., "The group method of data handling: a rival of method of stochastic approximation", *Soviet Automatic Control*, Vol. 13, No. 3, pp. 43-55, 1968.

[7] Park B. J., Oh S. K., and Jang S. W., "The Design of Adaptive Fuzzy Polynomial Neural Networks Architectures Based on Fuzzy Neural Networks and Self-Organizing Networks", *Journal of Control, Automation and Systems Engineering*, Vol. 8, No. 2, pp. 126-135, 2002(In Korean).

[8] Park B. J., and Oh S. K., "The Analysis and Design of Advanced Neurofuzzy Polynomial Networks", *Journal of the Institute of Electronics Engineers of Korea*, Vol. 39-CI, No. 3, pp. 18-31, 2002(In Korean).

[9] Oh S. K., Pedrycz W., and Park H. S., "Hybrid Identification in Fuzzy-Neural Networks", *Fuzzy Sets and Systems*, Vol. 138, No. 2, pp. 399-426, 2003.

[10] Box D. E. P., and Jenkins G. M., *Time Series Analysis, Forecasting, and Control*, Holden-Day, SanFransisco, 1976.

[11] Kim E., Lee H., Park M., and Park M., "A Simply Identified Sugeno-type Fuzzy Model via Double Clustering", *Information Sciences*, Vol. 110, pp. 25-39, 1998

[12] Lin Y., and Cunningham G. A., "A new Approach to Fuzzy-neural Modeling", *IEEE Transaction on Fuzzy Systems*, Vol. 3, No. 2, pp. 190-197, 1997.

[13] 오성권, "프로그래밍에 의한 컴퓨터 지능(퍼지, 신경회로망 및 진화알고리즘을 중심으로)", 내하출판사, 2002.