

Information Granulation-Based Multi-layer Hybrid Fuzzy Neural Networks: Analysis and Design

Byoung-Jun Park¹, Sung-Kwun Oh¹, Witold Pedrycz², and Tae-Chon Ahn¹

¹School of Electrical, Electronic and Information Engineering,
Wonkwang University, Korea
{lcap, ohsk, tcahn}@wonkwang.ac.kr

²Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB T6G 2G6, Canada
and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
pedrycz@ee.ualberta.ca

Abstract. In this study, a new architecture and comprehensive design methodology of genetically optimized Hybrid Fuzzy Neural Networks (gHFNN) are introduced and a series of numeric experiments are carried out. The gHFNN 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 gHFNN. The consequence part of the gHFNN is designed using PNN.

1 Introductory Remarks

Efficient modeling techniques should allow for a selection of pertinent variables and a formation of highly representative datasets. The models should be able to take advantage of the existing domain knowledge and augment it by available numeric data to form a coherent data-knowledge modeling entity. The omnipresent modeling tendency is the one that exploits techniques of Computational Intelligence (CI) by embracing fuzzy modeling [1-6], neurocomputing [7], and genetic optimization [8].

In this study, we develop a hybrid modeling architecture, called genetically optimized Hybrid Fuzzy Neural Networks (gHFNN). In a nutshell, gHFNN 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. 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 [9] is constructed on a basis of Group Method of Data Handling (GMDH) [10]

method 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 Conventional Hybrid Fuzzy Neural Networks (HFNN)

The architectures of conventional HFNN [11,12] result as a synergy between two other general constructs such as FNN and PNN. Based on the different PNN topologies, the HFNN distinguish between two kinds of architectures, namely basic and modified architectures. Moreover, for the each architecture we identify two cases. In the connection point, if input variables to PNN used on the consequence part of HFNN are less than three (or four), the generic type of HFNN does not generate a highly versatile structure. Accordingly we identify also two types as the generic and advanced. The topologies of the HFNN depend on those of the PNN used for the consequence part of HFNN. The design of the PNN proceeds further and involves a generation of some additional layers. Each layer consists of nodes (PNs) for which the number of input variables could the same as in the previous layers or may differ across the network. The structure of the PNN is selected on the basis of the number of input variables and the order of the polynomial occurring in each layer.

3 The Architecture and Development of Genetically Optimized HFNN (gHFNN)

The gHFNN 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 [13] and PNN [9].

3.1 Fuzzy Neural Networks and Genetic Optimization

We use FNN based on two types of fuzzy inferences, that is, simplified (Scheme I) and linear fuzzy inference-based FNN (Scheme II) as shown in Fig. 1. The notation used in Fig.1 requires some clarification. The “circles” denote units of the FNN while “N” identifies a normalization procedure applied to the membership grades of the input variable x_i . The output of the “ Σ ” neuron is described by a nonlinear function $f_i(x_i)$. Finally, the output of the FNN is governed by the following expression.

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

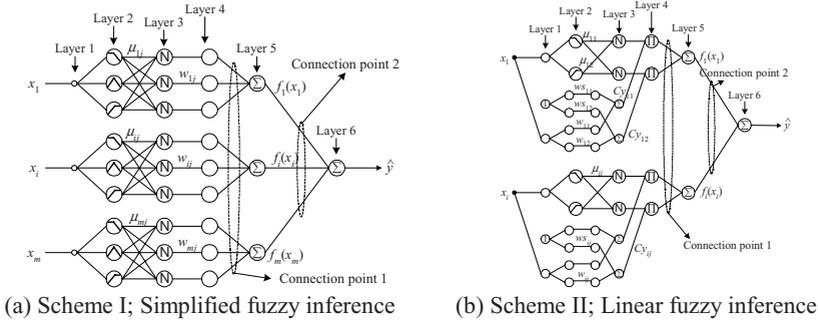


Fig. 1. Topologies of fuzzy set-based FNN

We can regard each $f_{i(x_i)}$ given by (1) as the following mappings (rules).

$$\text{Scheme I} - R^j : \text{If } x_i \text{ is } A_{ij} \text{ then } Cy_{ij}=w_{ij} \tag{2}$$

$$\text{Scheme II} - R^j : \text{If } x_i \text{ is } A_{ij} \text{ then } Cy_{ij}=ws_{ij}+w_{ij}x_i R_j \tag{3}$$

R^j is the j -th fuzzy rule while A_{ij} denotes a fuzzy variable of the premise of the fuzzy rule and represents a membership function μ_{ij} . 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). They express a connection existing between the neurons as visualized in Fig. 1. Mapping from x_i to $f_i(x_i)$ in (2) is determined by the fuzzy inferences and a standard defuzzification.

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

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP) algorithm [14]. For the simplified fuzzy inference-based FNN, the update formula of a connection in Scheme I is as follow.

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

Where, y_p is the p -th target output data, \hat{y}_p stands for the p -th actual output of the model for this specific data point, η is a positive learning rate and α is a momentum coefficient constrained to the unit interval. The inference result and the learning algorithm in linear fuzzy inference-based FNN use the mechanisms in the same manner as discussed above.

Genetic algorithms (GAs) are optimization techniques based on the principles of natural evolution. In essence, they are search algorithms that use operations found in natural genetics to guide a comprehensive search over the parameter space [8]. 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.

3.2 Genetically Optimized PNN (gPNN)

When we construct PNs of each layer in the conventional PNN [9], 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 (to be called “gPNN”). The overall genetically-driven optimization process of PNN is shown in Fig. 2.

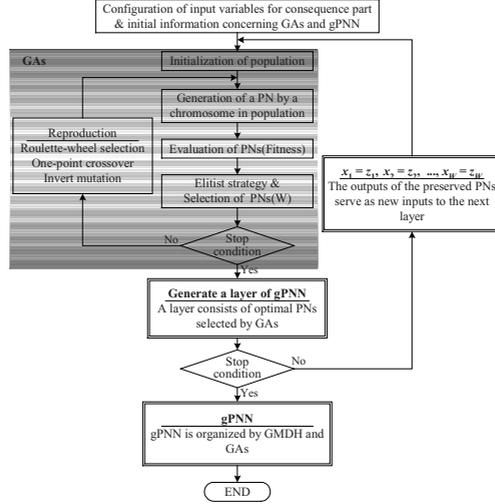


Fig. 2. Overall genetically-driven optimization process of PNN

4 The Algorithms and Design Procedure of gHFNN

The premise of gHFNN: FNN (Refer to Fig. 1)

[Layer 1] Input layer.

[Layer 2] Computing activation degrees of linguistic labels.

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

[Layer 4] Multiplying a normalized activation degree of the rule by connection. If we choose Connection point 1 for combining FNN with gPNN as shown in Fig. 1, a_{ij} is given as the input variable of the gPNN.

$$a_{ij} = \bar{\mu}_{ij} \times Cy_{ij} = \mu_{ij} \times Cy_{ij} \quad (6)$$

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

[Layer 5] Fuzzy inference for the fuzzy rules. If we choose Connection point 2, f_i is the input variable of gPNN.

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

The consequence of gHFNN: gPNN (Refer to Fig. 2)

[Step 1] Configuration of input variables.

If we choose the first option (Connection point 1), $x_1=a_{11}, x_2=a_{12}, \dots, x_n=a_{ij} (n=i \times j)$.

For the second option (Connection point 2), we have $x_1=f_1, x_2=f_2, \dots, x_n=f_m (n=m)$.

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

[Step 3] Initialization of population.

[Step 4] Decision of PNs structure using genetic design. We divide the chromosome to be used for genetic optimization into three sub-chromosomes as shown in Fig. 4(a).

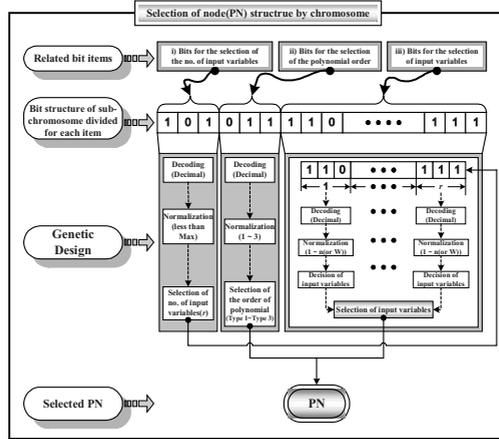


Fig. 3. The PN design using genetic optimization

Table 1. Different forms of regression polynomial forming a PN

Number of inputs Order of the polynomial	2	3	4
1 (Type 1)	Bilinear	Trilinear	Tetralinear
2 (Type 2)	Biquadratic-1	Triquadratic-1	Tetraquadratic-1
2 (Type 3)	Biquadratic-2	Triquadratic-2	Tetraquadratic-2

[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).

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

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

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

5 Experimental Studies

The performance of the gHFNN is illustrated with the aid of a time series of gas furnace [14]. 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 $y(t)$.

Table 2. Performance index of gHFNN for the gas furnace

Premise part				CP	Consequence part				PI	EPI			
Fuzzy Inference	No. of rules (MFs)	PI	EPI		Layer	No. of inputs	Input No.	T					
Simplified	6 (2+2+2)	0.0248	0.126	01	1	4	5	2	1	3	3	0.0220	0.135
					2	4	15	28	26	3	2	0.0209	0.135
					3	4	5	1	25	16	3	0.0205	0.131
					4	4	1	2	28	26	2	0.0190	0.128
					5	4	18	6	28	1	2	0.0175	0.125
				02	1	3	2	3	2	.	3	0.0221	0.135
					2	4	11	2	15	13	2	0.0194	0.126
					3	4	28	2	11	17	2	0.0190	0.116
					4	4	2	6	23	27	1	0.0188	0.114
					5	4	1	12	7	26	3	0.0182	0.112
Linear	6 (2+2+2)	0.0256	0.143	01	1	4	6	3	1	5	3	0.0218	0.136
					2	4	6	24	16	30	3	0.0197	0.124
					3	3	4	16	26	.	1	0.0196	0.121
					4	4	22	24	1	13	1	0.0193	0.119
					5	3	11	18	21	.	3	0.0191	0.117
				02	1	3	1	2	3	.	3	0.0232	0.130
					2	4	12	15	13	6	2	0.0196	0.120
					3	2	19	30	.	.	2	0.0194	0.115
					4	4	2	21	11	5	1	0.0188	0.113
					5	4	13	3	26	25	1	0.0184	0.110

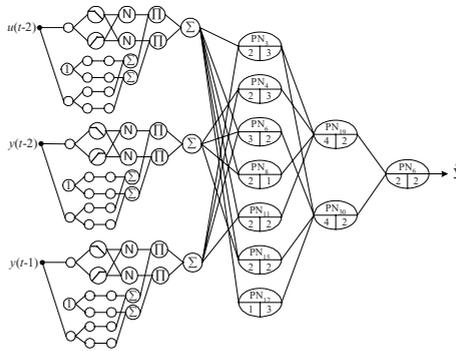


Fig. 4. Optimal topology of genetically optimized HFNN for the gas furnace (in case of linear fuzzy inference)

Table 2 summarizes the results of the optimized architectures according to connection points based on each fuzzy inference method. In Table 2, the values of the performance index of output of the gHFNN depend on each connection point based on the individual fuzzy inference methods. The optimal topology of gHFNN is shown in Fig. 4. Fig. 5 illustrates the optimization process by visualizing the performance index in successive cycles. It also shows the optimized network architecture when taking into consideration gHFNN based on linear fuzzy inference and connection point (CP) 2, refer to Table 2. Table 3 contrasts the performance of the genetically developed network with other fuzzy and fuzzy-neural networks studied in the literatures.

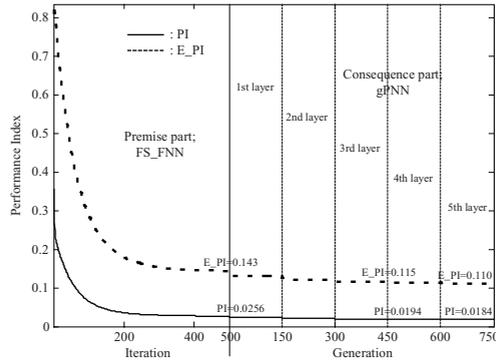


Fig. 5. Optimization procedure of gHFNN by BP learning and GAs

Table 3. Comparison of performance with other modeling methods

Model		PI	EPI	No. of rules	
Box and Jenkin’s model [14]		0.710			
Pedrycz’s model [1]		0.320			
Xu and Zailu’s model [2]		0.328			
Sugeno and Yasukawa’s model [3]		0.190			
Kim, et al.’s model [15]		0.034	0.244	2	
Lin and Cunningham’s mode [16]		0.071	0.261	4	
Fuzzy	Complex [4]	Simplified	0.024	0.328	4(2×2)
		Linear	0.023	0.306	4(2×2)
	Hybrid [6] (GAs+Complex)	Simplified	0.024	0.329	4(2×2)
		Linear	0.017	0.289	4(2×2)
	HCM+GAs [5]	Simplified	0.022	0.333	6(3×2)
		Linear	0.020	0.264	6(3×2)
FNN [13]		Simplified	0.043	0.264	6(3+3)
		Linear	0.037	0.273	6(3+3)
SOFNN		Generic [11]	0.017	0.250	4 rules/5 th layer
		Advanced [12]	0.019	0.264	6 rules/5 th layer
Proposed model (gHFNN)		Simplified	0.018	0.254	4 rules/5 th layer
			0.018	0.112	6 rules/5 th layer
		Linear	0.015	0.258	4 rules/5 th layer
			0.018	0.110	6 rules/5 th layer

6 Concluding Remarks

The comprehensive design methodology comes with the parametrically as well as structurally optimized network architecture. 1) As the premise structure of the gHFNN, 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 vertices 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 gHFNN 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 gHFNN 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.

Acknowledgement. This work was supported by Korea Research Foundation Grant (KRF-2003-002-D00297)

References

1. W. Pedrycz: An identification algorithm in fuzzy relational system. *Fuzzy Sets and Systems*, 13 (1984) 153-167
2. C. W. Xu, Y. Zailu: Fuzzy model identification self-learning for dynamic system. *IEEE Trans. on Syst. Man, Cybern.*, SMC-17(4) (1987) 683-689
3. M. Sugeno, T. Yasukawa: A Fuzzy-Logic-Based Approach to Qualitative Modeling. *IEEE Trans. Fuzzy Systems*, 1(1) (1993) 7-31
4. S.-K. Oh, W. Pedrycz: Fuzzy Identification by Means of Auto-Tuning Algorithm and Its Application to Nonlinear Systems. *Fuzzy Sets and Systems*, 115(2) (2000) 205-230,.
5. B.-J. Park, W. Pedrycz, S.-K. Oh: Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation. *IEE Proceedings-Control theory and application*, 148(5) (2001) 406-418
6. S.-K. Oh, W. Pedrycz, B.-J. Park: Hybrid Identification of Fuzzy Rule-Based Models. *International Journal of Intelligent Systems*, 17(1) (2002) 77-103
7. K. S. Narendra, K. Parthasarathy: Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. *IEEE Transactions on Neural Networks*, 2 (1991) 252-262
8. Z. Michalewicz: *Genetic Algorithms + Data Structures = Evolution Programs*. (1996) Springer-Verlag, Berlin Heidelberg
9. S.-K. Oh, W. Pedrycz, B.-J. Park: Polynomial Neural Networks Architecture: Analysis and Design. *Computers and Electrical Engineering*, 29(6) (2003) 653-725
10. A. G. Ivahnenko: The group method of data handling: a rival of method of stochastic approximation. *Soviet Automatic Control*, 13(3) (1968) 43-55
11. B.-J. Park, D.-Y. Lee, S.-K. Oh: Rule-Based Fuzzy Polynomial Neural Networks in Modeling Software Process Data. *International Journal of Control, Automation and Systems*, 1(3) (2003) 321-331
12. H.-S. Park, S.-K. Oh: Rule-based Fuzzy-Neural Networks Using the Identification Algorithm of GA hybrid Scheme. *International Journal of Control, Automation and Systems*, 1(1) (2003) 101-110
13. S.-K. Oh, W. Pedrycz, H.-S. Park: Hybrid Identification in Fuzzy-Neural Networks. *Fuzzy Sets and Systems*, 138(2) (2003) 399-426
14. D. E. P. Box, G. M. Jenkins: *Time Series Analysis, Forecasting, and Control*, 2nd edition (1976) Holden-Day, San Francisco
15. E. Kim, H. Lee, M. Park, M. Park: A Simply Identified Sugeno-type Fuzzy Model via Double Clustering. *Information Sciences*, 110 (1998) 25-39
16. Y. Lin, G. A. Cunningham III: A new Approach to Fuzzy-neural Modeling. *IEEE Transaction on Fuzzy Systems*, 3(2) 190-197