Genetically Optimized Hybrid Fuzzy Neural Networks Based on Simplified Fuzzy Inference Rules and Polynomial Neurons

Sung-Kwun Oh[1], Byoung-Jun Park[2], Witold Pedrycz[3], and Tae-Chon Ahn[2]

[1] Department of Electrical Engineering, The University of Suwon, San 2-2 Wau-ri,
Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, South Korea
ohsk@suwon.ac.kr
[2] Department of Electrical Electronic and Information Engineering, Wonkwang University,
344-2, Shinyong-Dong, Iksan, Chon-Buk, 570-749, South Korea
[3] Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB T6G 2G6, Canada
and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Abstract. We introduce an advanced architecture of genetically optimized Hybrid Fuzzy Neural Networks (gHFNN) and develop a comprehensive design methodology supporting their construction. The gHFNN architecture results from a synergistic usage of the hybrid system generated by combining Fuzzy Neural Networks (FNN) with Polynomial Neural Networks (PNN). As to the consequence part of the gHFNN, the development of the PNN dwells on two general optimization mechanisms: the structural optimization is realized via GAs whereas in case of the parametric optimization we proceed with a standard least square method-based learning.

1 Introductory Remarks

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], [2], [3], [4], [5], [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). The role of the FNN is to interact with input data, granulate the corresponding input spaces. 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 se-

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 798 – 803, 2005.

lection 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.

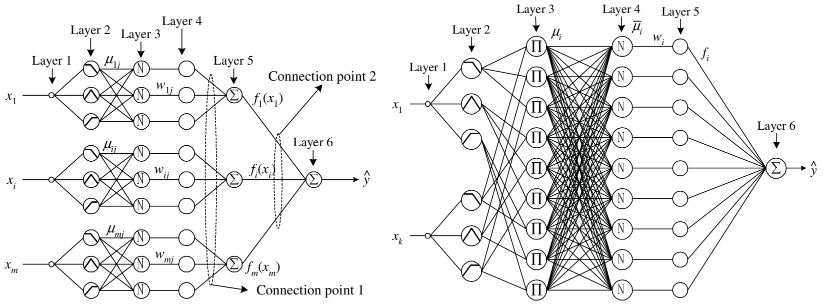
2 Conventional Hybrid Fuzzy Neural Networks (HFNN)

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 Genetically Optimized HFNN (gHFNN)

3.1 Fuzzy Neural Networks Based on Genetic Optimization

We consider two kinds of FNNs (viz. FS_FNN and FR_FNN) based on simplified fuzzy inference. The fuzzy partitions formed for each case lead us to the topologies visualized in Fig. 1.



(a) FS_FNN; individual input variables (b) FR_FNN; ensemble of input variables

Fig. 1. Topologies of FNN

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a BP algorithm [14]. 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)

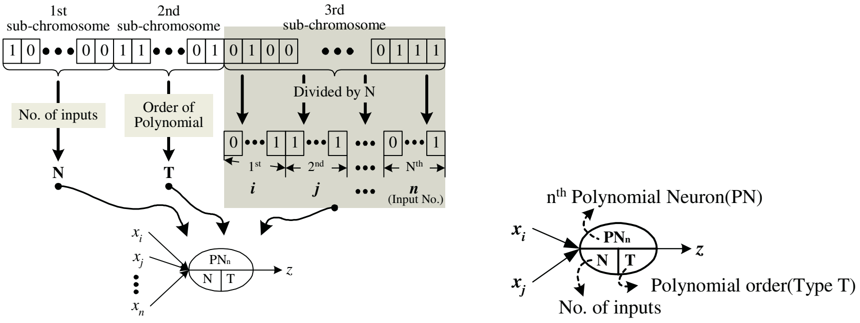
When we construct PNs of each layer in the conventional PNN [9], such parameters as the number of input variables (nodes), the order of polynomial, and input variables available within a PN are fixed (selected) in advance by the designer. This could have frequently contributed to the difficulties in 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”).

4 The Algorithms and Design Procedure of gHFNN

The premise of gHFNN: FS_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.
[Layer 5] Fuzzy inference for the fuzzy rules.
[Layer 6; Output layer of FNN] Computing output of a FNN.

The design procedure for each layer in FR_FNN is carried out in a same manner as the one presented for FS_FNN.

The consequence of gHFNN: gPNN (Refer to Fig. 2)
[Step 1] Configuration of input variables.
[Step 2] Decision of initial information for constructing the gPNN.



(a) Design of PNs using chromosome (b) Formation of each PN

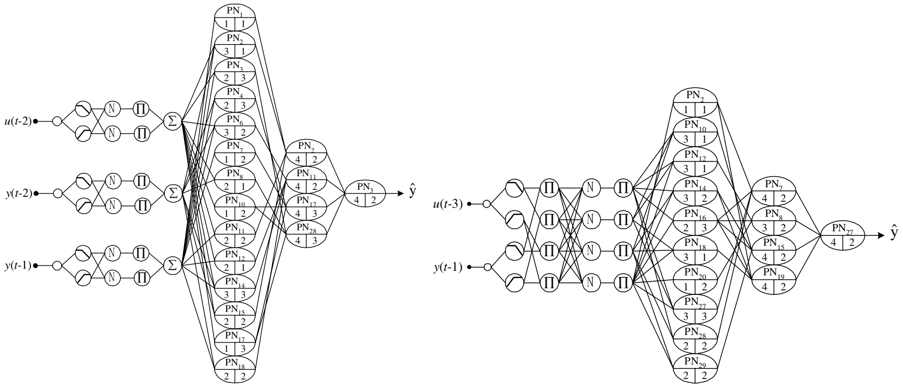
Fig. 2. The PN design using genetic optimization

[Step 3] Initialization of population.
[Step 4] Decision of PNs structure using genetic design. as shown in Fig. 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).
[Step 11] Determining new input variables for the next layer.

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

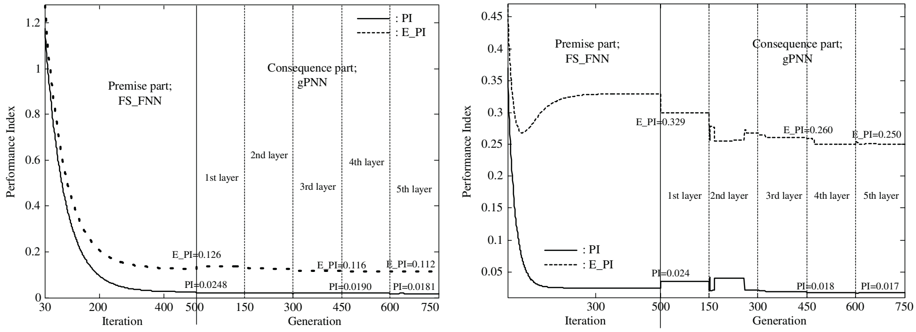
5 Experimental Studies

The performance of the gHFNN is illustrated with the aid of a time series of gas furnace [14].



(a) In case of using FS_FNN with Type II (b) In case of using FR_FNN with Type I

Fig. 3. Optimal topology of genetically optimized HFNN for the gas furnace



(a) In case of using FS_FNN with Type II (b) In case of using FR_FNN with Type I

Fig. 4. Optimization procedure of gHFNN by BP learning and GAs

We use two types of system input variables of FNN structure, Type I and Type II to design an optimal model from gas furnace data. Type I utilize two system input variables such as *u*(*t*-3) and *y*(*t*-1) and Type II utilizes 3 system input variables such as *u*(*t*-2), *y*(*t*-2), and *y*(*t*-1). The output variable is *y*(*t*).

The optimal topology of gHFNN is shown in Fig. 3. Fig. 4 illustrates the optimization process by visualizing the performance index in successive cycles. Table 1 contrasts the performance of the genetically developed network with other fuzzy and fuzzy-neural networks studied in the literatures.

Table 1. 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]			0190		
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)
SOFPNN		Generic [11]	0.023	0.277	4 rules/5th layer (NA)
			0.020	0.119	6 rules/5th layer (22 nodes)
		Advanced [12]	0.019	0.264	4 rules/5th layer (NA)
			0.017	0.113	6 rules/5th layer (26 nodes)
Proposed model (gHFNN)		FS_FNN	0.020	0.265	4 rules/3rd layer (12nodes)
			0.019	0.116	6 rules/3rd layer (19 nodes)
		FR_FNN	0.018	0.260	4 rules/3rd layer (15nodes)
			0.018	0.114	7 rules/3rd layer (9 nodes)

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 GAs and BP: 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 gHFNN is based on the technologies of the PNN and GAs: The PNN 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 PNN is driven to genetic optimization, in what follows it leads to the selection of the optimal nodes.

Acknowledgement. This work has been supported by KESRI(R-2004-B-133-01), which is funded by MOCIE(Ministry of commerce, industry and energy).

References

1. Pedrycz, W.: An identification algorithm in fuzzy relational system. Fuzzy Sets and Systems. **13** (1984) 153-167
2. Xu, C.W., Zailu, Y.: Fuzzy model identification self-learning for dynamic system. IEEE Trans. on Syst. Man, Cybern. SMC-**17**(4) (1987) 683-689
3. Sugeno, M., Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modeling. IEEE Trans. Fuzzy Systems. **1**(1) (1993) 7-31
4. Oh, S.K., Pedrycz, W.: Fuzzy Identification by Means of Auto-Tuning Algorithm and Its Application to Nonlinear Systems. Fuzzy Sets and Systems. **115**(2) (2000) 205-230
5. Park, B.J., Pedrycz, W., Oh, S.K.: Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation. IEE Proceedings-Control theory and application. **148**(5) (2001) 406-418
6. Oh, S.K., Pedrycz, W., Park, B.J.: Hybrid Identification of Fuzzy Rule-Based Models. International Journal of Intelligent Systems. **17**(1) (2002) 77-103
7. Narendra, K.S., Parthasarathy, K.: Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. IEEE Transactions on Neural Networks. **2** (1991) 252-262
8. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin Heidelberg. (1996)
9. Oh, S.K., Pedrycz, W., Park, B.J.: Polynomial Neural Networks Architecture: Analysis and Design. Computers and Electrical Engineering. **29**(6) (2003) 653-725
10. Ivahnenko, A. G.: The group method of data handling: a rival of method of stochastic approximation. Soviet Automatic Control. **13**(3) (1968) 43-55
11. Park, B.J., Oh, S.K., 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. **8**(2) (2002) 126-135 (In Korean)
12. Park, B.J., Oh, S.K.: The Analysis and Design of Advanced Neurofuzzy Polynomial Networks. Journal of the Institute of Electronics Engineers of Korea. **39**-CI(3) (2002) 18-31 (In Korean)
13. Oh, S.K., Pedrycz, W., Park, H.S.: Hybrid Identification in Fuzzy-Neural Networks. Fuzzy Sets and Systems. **138**(2) (2003) 399-426
14. Box, D. E. P., Jenkins, G. M.: Time Series Analysis, Forecasting, and Control, 2nd edition Holden-Day, SanFransisco. (1976)
15. Kim, E., Lee, H., Park, M., Park, M.: A Simply Identified Sugeno-type Fuzzy Model via Double Clustering. Information Sciences. **110** (1998) 25-39
16. Lin, Y., Cunningham III, G. A.: A new Approach to Fuzzy-neural Modeling. IEEE Transaction on Fuzzy Systems. **3**(2) 190-197
17. Park, H.S., Park, B.J., Kim, H.K., Oh, S,K,: Self-Organizing Polynomial Neural Networks Based on Genetically Optimized Multi-Layer Perceptron Architecture. International journal of Control, Automations, and Systems. **2**(4) (2004) 423-434