DESIGNING OPTIMAL NEURO-FUZZY ARCHITECTURES FOR INTELLIGENT QUALITY CONTROL

Stanimir Yordanov Yordanov

Department AIUT, TU Gabrovo, H.Dimitar 4., 5300, Bulgaria, e-mail: sjjordanov@tugab.bg

Keywords: Artificial neural networks, fuzzy logic, neuro-fuzzy systems, intelligent quality control

The integration of the Artificial Neural Network (ANN) and Fuzzy Logic (FL) in one architecture in order, to overcome the individual limitations and to achieve synergetic effects through a combination of these techniques, has a in recent years contributed to a large number of Neuro-Fuzzy (NF) architectures. NF techniques override the classical control methods in many aspects, such as algorithm simplicity, system robustness and the ability to handle imprecision and uncertainties. In this paper are a presented some state-of-art NF models. A further attempt to assess the strengths and weakness of each NF architecture and selection criteria for IC applications is made. Finnalya choice of an optimal NF architecture is made and it's future aplication in a quality control system is presented.

1. INTRODUCTION

Intelligent Control (IC) is an interdisciplinary idea combining computer science, control theory, operations research and artificial intelligent techniques, aiming to achieve optimal control. It is well known that the intelligent systems, which can provide human like expertise such as domain knowledge, uncertain argument, and adaptation to a noisy and time varying environment, are important in solving practical IC problems.

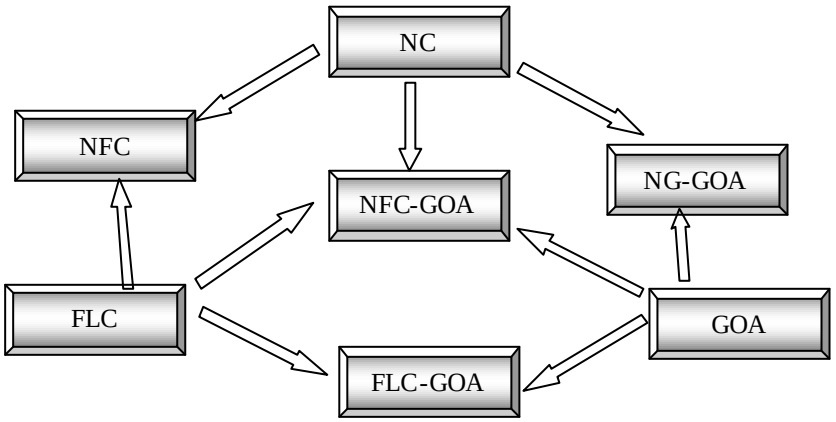

Fig. 1.A general framework for intelligent controllers.

While conventional Neural Controllers (NC) make use of neural network learning techniques to determine the control variables, Fuzzy Logic Controllers (FLC) make use of fuzzy generalized rules or linguistic terms to model the control system. Global Optimization Algorithms (GOA) like Genetic Algorithm (GA),

Simulated Annealing (SA), and Tabu Search (TS) etc. have been widely used to optimize the architectures of NC and FLC. Figure 1 depicts a general framework for various intelligent controllers and interaction between the various techniques. As one important area in the intelligent control, the Neuro-Fuzzy Control (NFC) is a new approach that can learn from the system and then reason about its state.

2. NEURO-FUZZY SYSTEMS

A FLC can utilize the human expertise by storing its essential components in a rule base and database, and perform fuzzy reasoning to infer the overall output value. However there is no systematic way to transform experiences of knowledge of human experts to the knowledge base of a FLC. There is also a need for adapting of some of the learning algorithms to set the outputs within the required error rate.

On the other hand, the learning mechanism of NC does not rely on human expertise. Due to the homogeneous structure of NC, it is hard to extract structured knowledge from either the weights or the configuration of the NC. For many practical problems, a priori knowledge is usually obtained from human experts and it is most appropriate to express the knowledge as a set of fuzzy if-then rules. However, it is not easy to encode this into an NC. Table 1 summarizes the comparison of NC and FLC.

Table 1. Complementary features of NC and FLC

NC	**FLC**
Black box	Easy interpretable
Learning from scratch	Use linguistic knowledge

To a large extent, the drawbacks pertaining to these two approaches seem complementary. Therefore it seems natural to consider building an integrated system combining the concepts of FL and ANN. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special ANN like architecture. However the conventional ANN learning algorithms (gradient descent) cannot be applied directly to such a system as the functions used in the inference process are usually non differentiable. This problem can be overcomed by using differentiable functions in the system output or by not using the standard neural learning algorithms. Several NF models can be found in the literature. The integrated architecture shares data structures and knowledge representations. Some of the major woks in this area are Adaptive Neuro Fuzzy Inference System (ANFIS) [1], Neuro Fuzzy Controller (NEFCON) [3], Fuzzy Net (FUN) [4], Self Constructing Neural Fuzzy Inference Network (SONFIN) [2], Fuzzy Inference Environment Software with Tuning (FINEST) [5], Evolving Fuzzy Neural Network (EfuNN) [6], Dynamic Evolving Fuzzy Neural Network (dmFuNN) [7], and many others [8-12].

3. REALISATION OF NEURO-FUZZY SYSTEMS FOR QUALITY CONTROL

The purpose of the present development - *creation of a system for classification of details and processes on the basises of determined parameters*.

The task for *classification* is more extensive and complex.

Let's assume that classification concerns determined classes of electrotechnological objects for which every new change in the system condition will be ranked to a determined class of cases, on the basis of supervision of determined set of attributes or characteristics. Most of the electrotehnological systems well-known. Hence one cam construct distinguish (nominal) models, intended for identification and control studying of the processes in able the system.

There exist a great number of sets of methods for classification. The three fundamental methods for identification: *statistical*, *machine learning* and *neural network*.

The purposes of every method are:

- To be comparable with the human potentialites for the analysis of the processes, but to surpass them on consistency and clearness of the result.
- Analys of wide spectra of problems and to work a greet set of data
- Successful appendion in practical systems.

In the present development an attempt is made for a combination of the three methods, aiming to obtain an integrated program system for control and management of the processes in electrotehnological installations with discrete or discrete - continuous processes is and to estimate the quality of productions. The structure of the system is shown on fig. 2.

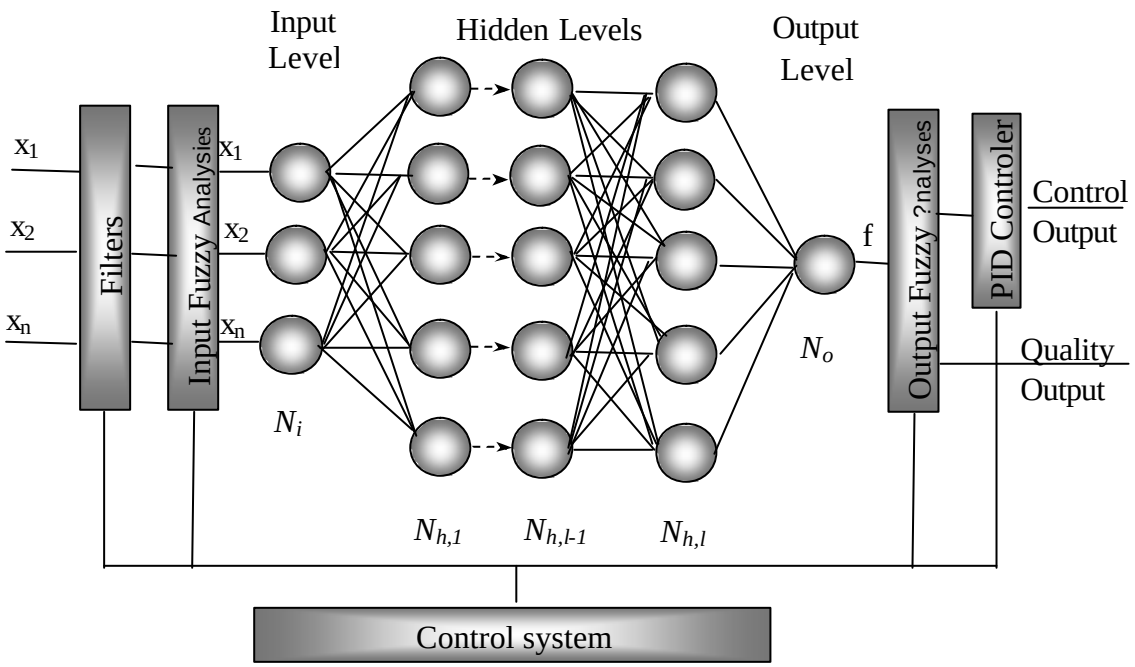

Fig.2 Structure Neuro-Fuzzy system

It comprises of three layers. The first is intended for original data processing. It is realized on the base of the Fuzzy analysis of the data. The processed data are passed on neural model, which identifies the condition of the process and generates a generalized criterion quality. The output of the neural networks, passed to an

analyzing method witch, is realized on the methods on the indistinct control and generates a control signal. Neural identification and Fuzzy control are optional and can function independently or together.

The network consists of an input, an output and three hidden layers The neurons of each layer have different activation functions representing the different stages in the calculation of fuzzy interface. The activation function can be individually chosen for problems.

The input variables are stored in the input neurons. The neurons in the first hidden layer contain the membership functions and this performs a fuzzification of the input values.

In the second hidden layer, the conjunctions (fuzzy-AND) are calculated. The membership functions of the output variables are stored in the third hidden layer. Their activation function is a fuzzy-OR. Finally the out are neurons contain the output variables and passes de-fuzzy activation function.

The network is initialized with a fuzzy rule base and the corresponding membership functions and after using a stochastic procedure (learning technique) that randomly changes the parameters of the membership functions and the connections within the network structure. The learning process is driven by a cost function, which is evaluated after random modification. If the modification resulted in an improved performance the modification is kept, otherwise this is canceled.

The system it is object orietated. Fig. 3 shows the inheritance of be objects in neuronal and the fuzzy parts. There are three base classes TNeuron, TLayer, TNeuralNet. All others are derivatives from them.

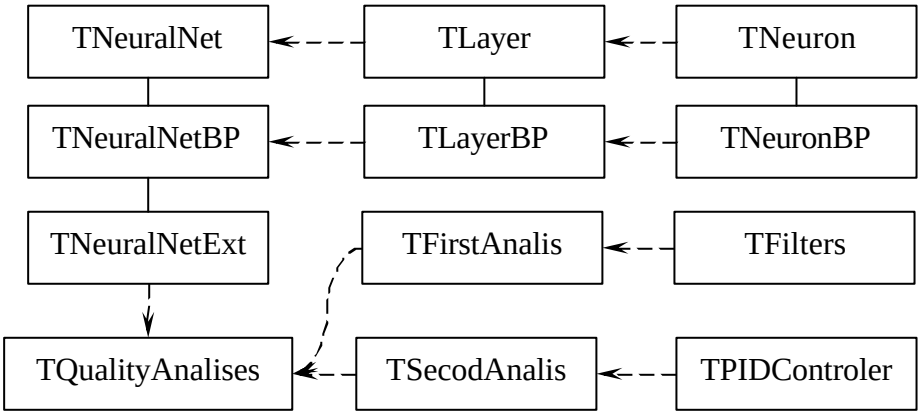

Fig.3 Inheritance of the classes

The analysis of the data is grounded on the functioning of an object of type **TQualityAnalises**. It contains objects of types **TFirstAnalis**, **TSecodAnalis**, **TNeuralNetExt** and calls in a certain sequence methods from them.

The assignment of the classes as foloving:

TQualityAnalises – Basic class performing the analysis of the data.

TNeuron - Base class for the neuron. Contains the basic functional possibilities. The class contains the Weights and the method for the calculation of the output, which is calculated by means of the method ComputeOut.

TNeuronBP- is an inheritor of Tneuron, it serves for a program realization of multilayer neural networks. The ComputeOut method include nonlinear activation function which is band with the property OnActivationF. Two properties, Delta-containing the local mistake and PrevUpdate-containing the values of the weight factors from the previous step of learning are added.

The base class TLayer and its inheritor TLayerBP are appointed to unite the single neurons in a layer, for simplification of the work with the net.

The **TNeuralNet** component is a base component for all kinds of neural networks. TNeuralNet provides the necessary functionality of the derivative components. This component supports methods for work with the layers of the network (AddLayer, DeleteLayer) and methods for manipulation with the initial data (AddPattern, DeletePattern, ResetPatterns). Method Init serves for construction of a neural network. The majority of methods are public which helps to include them easily in other programs.

InitWeights method is added it is intended for the Weight coefficients.

The class **TNeuralNetBP** realizes a multilayered neural network trained on the algorithm of return distribution of the mistake. The following methods are added Compute-calculates the output of the neural network, it is used after a training the network; TeachOffLine-trains the neural network.

TNeuralNetExtented is an inheritor of TNeuralNetBP. The following methods are added SaveNetwork for recording and (LoadNetwork) for reading the trained neural network from file. LoadDataFrom-loads the data for training from a text file; The NormalizeData method serves for normalization of the input and target data; Train – for training a neural network; ComputeUnPrepData- calculates the result of the implementation of the neural network.

TFilters – Realizes a set of software filters. It is filtering the input data.

TPIDControler - This object realizes the classical PID law for regulation in a digital mode. In the system there can exist some objects from this type, in dependence on the number of the adjustable quantities.

TFirstAnalis – performs initial processing of the data. It is realized on the bass of fuzzy-logic.

TSecodAnalis – performs the output analysis and the taking of decision for the management of the system. It is realized on the bass of fuzzy-logic

4. CONCLUSION

The quality management of production is a fundamental priority in the industry. In the present article a method for achieving these aims is described.

The use of artificial neural networks in the control and management module makes the system exceptionally flexible and applicable with a wide range of electro-technological systems

5. REFERENCES

[1] Jang R, Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, PhD Thesis, University of California, Berkeley, July 1992.

[2] Juang Chia Feng, Lin Chin Teng, An Online Self Constructing Neural Fuzzy Inference Network and

its Applications, IEEE Transactions on Fuzzy Systems, Vol 6, No.1, pp. 12-32, 1998.

[3] Nauck D, Kruse R, A Neuro Fuzzy Controller Learning by Fuzzy Error Propagation, In Proceedings of Conference of the North American Fuzzy Information Processing Society NAFIPS '92, Mexico, pp. 388-397, 1992.

[4] Sulzberger SM, Tschicholg-Gurman NN, Vestli SJ, FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks, In Proceedings of IEEE Conference on Neural Networks, San Francisco, pp 312-316, March 1993.

[5] Tano S, Oyama T, Arnould T, Deep combination of Fuzzy Inference and Neural Network in Fuzzy Inference, Fuzzy Sets and Systems, 82(2) pp. 151-160, 1996.

[6] Kasabov N, Evolving Connectionist and Fuzzy Connectionist Systems -Theory and Applications for Adaptive On-line Intelligent Systems, In: Neuro-Fuzzy Techniques for Intelligent Information Processing, Kasabov N and Kozma R, (Eds.), Physica Verlag, 1999.

[7] Kasabov N and Qun Song, Dynamic Evolving Fuzzy Neural Networks with 'm-out-of-n' Activation Nodes for On-line Adaptive Systems, Technical Report TR99/04, Department of information science, University of Otago, 1999.

[8] Zhang QY, Kandel A, Compensatory Neurofuzzy Systems with Fast Learning Algorithms, IEEE Transactions on Neural Networks, Volume 9, No. 1,pp.83-105, 1998.

[9] Mizumoto M, Shi Yan, A New Approach of Neuro-fuzzy Learning Algorithm, Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms, Ruan D (Ed.), Kluwer Academic Publishers, pp. 109-129, 1997.

[10] Takagi H, Hayashi I, NN-Driven Fuzzy Reasoning, International Journal of Approximate Reasoning, Vol.5, pp. 191-212, 1991.

[11] Halgamuge SK, Glesner M, The Fuzzy Neural Approach for Pattern Classification with the Generation of Rules Based on Supervised Learning, In Proceedings of Neuro-Nimes 92, 167-173, 1992.

[12] Zimmermann HG, Neuneier R, Dichtl H, Siekmann S, Modeling the German Stock Index DAX with Neuro-Fuzzy, In Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing, 1996.