

Nonlinear Time Series Prediction Based on Lyapunov Theory-Based Fuzzy Neural Network and Multiobjective Genetic Algorithm

Kah Phooi Seng¹ and Kai Ming Tse²

¹ School of Engineering & Science, Monash University (Malaysia),
Bandar Sunway, 46150 PJ, Malaysia
jasmine.seng@engsci.monash.edu.my

² School of Microelectronics, Griffith University
Kessels Rd, Nathan QLD 4111, Australia
k.tse@griffith.edu.au

Abstract. This paper presents the nonlinear time series prediction using Lyapunov theory-based fuzzy neural network and multi-objective genetic algorithm (MOGA). The architecture employs fuzzy neural network (FNN) structure and the tuning of the parameters of FNN using the combination of the MOGA and the modified Lyapunov theory-based adaptive filtering algorithm (LAF). The proposed scheme has been used for a wide range of applications in the domain of time series prediction. An application example on sunspot prediction is given to show the merits of the proposed scheme. Simulation results not only demonstrate the advantage of the neuro-fuzzy approach but it also highlights the advantages of the fusion of MOGA and the modified LAF.

1 Introduction

Time series prediction is a very important practical application with a diverse range of applications including economic and business planning, inventory and production control, weather forecasting, signal processing and control [1]. As a result, there has been considerable interest in the application of intelligent technologies such as neural networks (NNs) and fuzzy logic [2]-[3]. More recently, these two computationally intelligent techniques have been viewed as complementary leading to developments in fusing the two technologies [4], with a number of several successful neurofuzzy systems reported in the literature [5]. Such work has demonstrated the superior prediction capabilities of a fuzzy neural network as compared to the conventional neural network approach [5]-[6].

In this paper, we employ the fuzzy neural network (FNN) for nonlinear time series prediction. A new combination of the modified Lyapunov theory-base filtering algorithm (LAF) and multi-objective genetic algorithm (MOGA) [10] is used to adjust the network parameters. The proposed scheme provides not only the advantages of fuzzy

logic and NN but it also offer additional advantages offered by the modified LAF and MOGA. The weight parameters of FNN in the consequence part are adaptively adjusted by the modified LAF. The MOGA is used to tune the parameters of the membership functions (MBFs) in the premise part. Most real world problems require the simultaneous optimisation of multiple criteria/objectives. In this case, MOGA can provide the solution to these problems. In the proposed scheme, 2 types of error: instantaneous and a prior errors defined in later section are the multiple criteria to be solved by MOGA. The proposed scheme has been used for a wide range of applications in the domain of time series prediction. The theoretical prediction mechanism of the proposed scheme is further confirmed by the simulation example for real world data such as sunspot forecasting.

The paper is organized as follow: section 2 briefly describes the main features of the proposed FNN and the two criteria. Section 3 presents the fuzzy neural network learning: structure learning and parameter learning. The parameter learning 1 – the modified LAF algorithm is presented in section 4. Section 5 describes the MOGA and the transformation of the multiobjective function into a new function so that single objective optimization methods can be used. The prediction results are presented in section 6. The finally section 7 concludes the paper with a discussion of the significance of the results.

2 Fuzzy Neural Network

Neural-fuzzy systems have been applied to many fields successfully since a decade ago. In this section, the fuzzy logic inference system can be implemented as a five-layer NN (Fig. 1). This type of architecture is the most common among neural fuzzy inference systems. Given the training input data $x_n, n = 1, 2 \dots N$, and the desired output $d_m, m=1,2 \dots M$, the inference rules of simplified fuzzy reasoning can be established by experts or generated based on numerical data as proposed by Wan and Mendel [11]. The rule base contains the following form:

$$\begin{aligned}
 R^i : & \text{IF } x_1 \text{ is } A_1^i \text{ and } \dots x_N \text{ is } A_N^i, \\
 & \text{THEN } y_1 \text{ is } w_1^i \text{ and } \dots y_M \text{ is } w_M^i
 \end{aligned}
 \tag{2.1}$$

where i is a rule number, the A_N^i 's are MBF's of the antecedent part and w_M^i 's are real numbers of the consequent part.

The operation of the this system can be described layer by layer as follows:

Layer 1: Fuzzification

This layer consists of linguistic variables. The crisp inputs $x_n, n=1,2 \dots N$ are fuzzified by using MBFs of the linguistic variables A_N^i . Usually, triangular, trapezoid, Gaussian or bell-shaped membership functions are used.

Layer 2. Rule Nodes

The second layer contains one node per each fuzzy if-then rule. Each rule node performs connective operation between rule antecedents (if-part). Usually, the minimum or the dot product is used as intersection AND. The union OR is usually done using maximum operation. In our example case the firing strengths μ_i , of the fuzzy rules are computed according to

$$\mu_i = A_1^i(x_1) \cdot A_2^i(x_2) \cdot \dots \cdot A_N^i(x_N) \tag{2.2}$$

Layers 3-5: Normalization, Consequence & Summation

In the third layer, the firing strengths of the fuzzy rules are normalized. Layer 4 is related to consequent fuzzy labels which are singletons in our case. The values of the singletons are multiplied by normalized firing strength. The final layer computes the overall output as the summation of the incoming signals. Therefore the output y_m of the fuzzy reasoning can be represented by the following equation:

$$y_m = \frac{\sum_i^Q \mu_i w_m^i}{\sum_i^Q \mu_i} \tag{2.3}$$

$$Y = [y_1, y_2, \dots, y_M] \tag{2.4}$$

After the fuzzy logic rules and network structure have been established, the learning algorithm can then applied to adjust the parameters of the MBFs in the premise part and the weights in the consequence parts. In this paper, we proposed to use the modified LAF algorithm to adaptively adjust the weights in the consequence parts and MOGA to tune the parameters of MBFs.

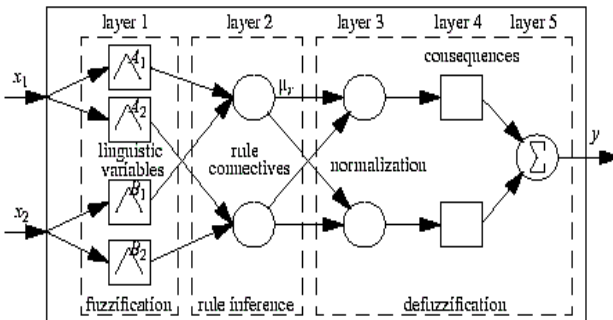


Fig. 1. The Fuzzy Neural Network

Fig. 1 illustrates the overall process of the proposed scheme for the prediction problem. The layer 5 consists of 1 summation node or 1 output, $y_i(t)$ which is defined as

$$y_1(t) = \frac{\sum_i^Q \mu_i w_m^i(t)}{\sum_i^Q \mu_i} \tag{2.5}$$

$y_2(t)$ is not another output node of FNN as shown in Fig. 2 and it is only computed using (2.6)

$$y_2(t) = \frac{\sum_i^Q \mu_i w_m^i(t-1)}{\sum_i^Q \mu_i} \tag{2.6}$$

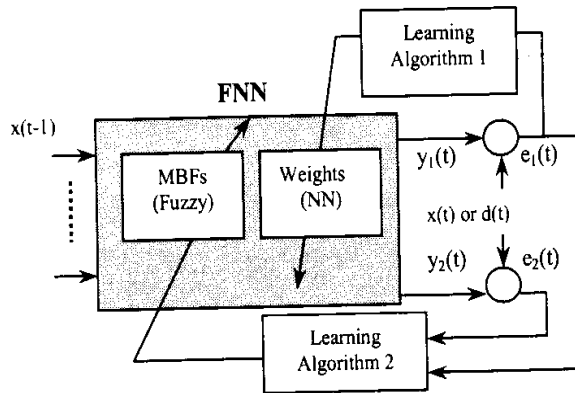


Fig. 2. The block diagram of FNN with the modified LAF + MOGA

3 Fuzzy Neural Network Learning

There are two major learning in fuzzy neural network: the structure learning and the parameter learning. The structure learning is to generate initial fuzzy rules when there is no initial fuzzy rule established by the experts. The initial fuzzy rules can be generated based on numerical data. Wang and Mendal have proposed a simple and straightforward algorithm [11]. The main steps of the algorithm are described as follows:

First, we have to determine the input and output variables and construct the numerical data set. The k^{th} numerical data set can be formed as

$$\{x'_1(t), x'_2(t), \dots, x'_n(t)\} \rightarrow \{y'_1(t), y'_2(t), \dots, y'_n(t)\}$$

where the left hand side and the right hand side specify the input and output numerical data. Second, calculate the membership degree for each variable by a row vector denoted by $\bar{x}_i(t)$ for inputs and $\bar{y}_j(t)$ for outputs as

$$\bar{x}_i(t) = [A_i^1(t), A_i^2(t), \dots, A_i^{N_i}(t)] \tag{3.1}$$

$$\bar{y}_j(t) = [B_j^1(t), B_j^2(t), \dots, B_j^{M_j}(t)] \tag{3.2}$$

$A_i^1(t), A_i^2(t), \dots, A_i^{N_i}(t)$ and $B_j^1(t), B_j^2(t), \dots, B_j^{M_j}(t)$ are the membership degrees for $x'_i(t)$ and $y'_j(t)$. Third, calculate the importance degree to each data pair as

$$\bar{h}(t) = \prod_{i=1}^n \max \{ \bar{x}_i(t) \} \tag{3.3}$$

Fourth, construct the fuzzy rules for all data pairs. For instance, a fuzzy rule has this form:

$$\begin{aligned} R^{(t)} : & \text{ IF } x_1 \text{ is } A_1^2 \text{ and } x_2 \text{ is } A_2^2 \\ & \text{ THEN } y_1 \text{ is } B_1^1 \text{ and } y_2 \text{ is } B_2^2 \end{aligned} \tag{3.4}$$

Fifth, delete the conflict fuzzy rules, while two rules have the same fuzzy set in IF part but a different set in THEN part. The proper rule is selected according to the highest importance degree.

4 Parameter Learning Algorithm 1 – The Modified LAF

The LAF algorithm in [7] has been modified to adaptively adjusts the weights of FNN in the consequence part. Those weights in the consequence part are updated as follow:

$$w_m^i(t) = w_m^i(t-1) + g_m^i(t)\alpha_m(t) \tag{4.1}$$

where $g_m^i(t)$ is the adaptation gain and $\alpha_m(t)$ is defined as

$$\alpha_m(t) = d_m(t) - \frac{\sum_i \mu_i w_m^i(t-1)}{\sum_i \mu_i} \tag{4.2}$$

The adaptation is given by

$$g_m^i(t) = \frac{\mu_i(t)}{\|U(t)\|^2} \left(1 - k \frac{|e_m(t-1)|}{|\alpha_m(t)|} \right) \tag{4.3}$$

where $0 < k \leq 1$. $U(t) = [\mu_1, \mu_1, \dots, \mu_Q]$.

It is noticeable that the values of $U(t)$ and α_m in (4.3) may be zero and rise singularities problem. Therefore the adaptation gain may be modified as (4.4)

$$g_m^i(t) = \frac{\mu_i(t)}{\|U(t)\|^2 + \lambda_1} \left(1 - k \frac{|e_m(t-1)|}{|\alpha_m(t)| + \lambda_2} \right) \tag{4.4}$$

where $0 < k \leq 1$, and λ_1, λ_2 are small positive numbers.

5 Parameter Learning Algorithm 2 - MOGA

The MOGA is used to tune the parameters of the membership functions (MBFs) in the premise part. Without the need of linearly combining multiple attributes into a composite scalar objective function, evolutionary algorithms incorporate the concept of Pareto's optimality or modified selection schemes to evolve a family of solutions along the tradeoff surface. The manner of Pareto optimality is one of the useful approaches to solve multiobjective optimization problems. In this paper we employ the weighted sum-based optimization method.

5.1 Weighted Sum Based Optimization

In a weighted sum-based optimization, multiobjective $F=(f_1, \dots, f_2)$ is transformed into

$$F_w = \sum_{i=1}^k w_i f_i$$

so that single objective optimization methods can be used. Preferences

are used for specifying weights, With reference to Fig. 2, two fitness F_1 , and F_2 can be evaluated from $e_1(t)$ and $e_2(t) \forall t$. Thus, these two objective functions is transformed in an overall fitness function $F = w_1 F_1 + w_2 F_2$, where $w_1 + w_2 = 1$. For example, we may choose $w_1 = 0.3$ and $w_2 = 0.7$.

5.2 Computational Algorithms

The procedure of the MOGA algorithm is described as follows:

1. Initialization: Training data is clustered to generate 9 centroids based on which the Gaussian MBFs (mean and variance) are evaluated. 80 potential candidates $P(t)$ are created by varying $\pm 20\%$ of the MBF s.
2. Evaluate the overall fitness F . Select candidates proportional to their fitness relative to the others in $P(t)$ using the Stochastic Universal Sampling technique.
3. Applying genetic operators, whole arithmetic: crossover, mutation, and adaptation with the best candidate by adding a perturbation to the relative best-fit candidate, to reproduce new candidates.
4. Combine all new candidates with the $P(t)$ to form the new population for the next generation.
5. Repeat step 2, 3, 4 until termination condition is satisfied.

6 Simulation Example

In order to demonstrate the performance of the proposed method, we have applied the method to prediction of the sunspot data. Sunspot data is used as a benchmark for many years by researchers. Data file of the Sunspot times series is download from [9]. It consists the sunspot data from the year 1700 to 1999 (300 Samples).

Fig. 3 shows the plot of the sunspot time series. Fig. 4 shows the mean squared error of $e_2(t)$ giving $MSE=0.0159$ at the 30th generation. Fig. 3 show no distinct difference between the $y_2(t)$ and $x(t)$. Fig. 5 and 6 reveal some weight parameters of the FNN. A computer program developed in MATLAB software is used to implement the proposed scheme.

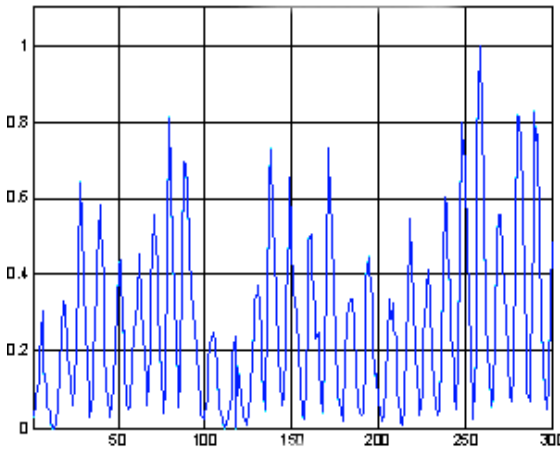


Fig 3. Normalized sunspot time series & the predictor output

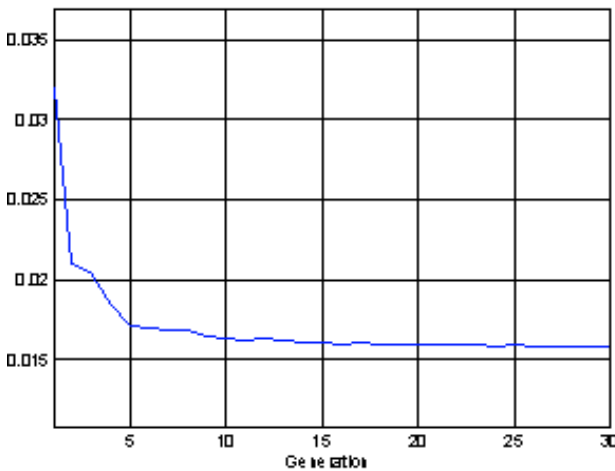


Fig. 4. MSE of $e_2(t)$

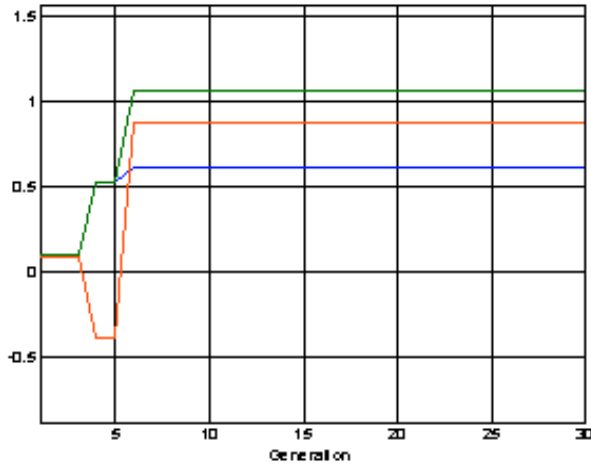


Fig. 5. Weight parameters of FNN

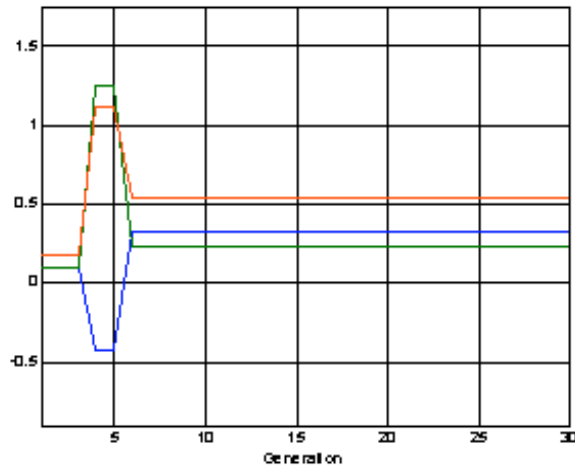


Fig. 6. Weight parameters of FNN

7 Conclusion

This paper has presented a new approach in designing a FNN with MOGA and the modified LAF techniques. The previous section clearly demonstrate the performance of the proposed FNN for the prediction of nonlinear time series. The modified LAF has provided the fast error convergence to the training of FNN. On the other hand, MOGA has added advantage of global optimization to the FNN training based on two criteria/objectives. The results have emphasized the benefits of the fusion of fuzzy and NN technologies as well as the advantages of the fusion of the modified LAF and MOGA. This increases in transparency of the neurofuzzy approach overcomes the

drawback of FNN with gradient techniques and/or GA in the conventional NNs or FNNs. In general the prediction capability (accuracy) of this system is proportional to its granularity (the number of fuzzy sets) in the premise part and the numbers of weights in the consequence part. Future works need to be conducted in this area. Many issues need to be addressed regarding simulations, practical implementations, and the further analysis on the theoretical parts of the proposed scheme.

Reference

- [1] Box, GEP; Jenkins, GM: "Time series analysis: forecasting and control", Oakland CA: Holden-Day, 1976.
- [2] Wang, LX,- Mendel. YM: "Generating fuzzy rules by learning from examples", IEEE Trans. Systems Man and Cybernetics, Vol 2". No. 6, pp. 1414-1427, 1992.
- [3] Moody, J; Darken, C: "Fast learning in networks of locally tuned processing units", Neural Computation, pp. 281-294, Vol. 1, 1989~
- [4] Brown, M- Harris, C: "Neurofuzzy adaptive modelling and control", Prentice-Hall, 1994.
- [5] Jang, Roger JS: "Predicting chaotic time series with fuzzy IF-THEN rules" Proc. IEEE 2nd Int. Conf. on Fuzzy Systems, pp. 1079-1084 Vol. 2, 1993.
- [6] Jang, Roger J S; Sun, C-T: "Neuro-fuzzy modelling and Control" Proc. of the IEEE, pp. 378-406, Vol 83, No 3 March 1995.
- [7] Seng Kah Phooi, Zhihong Man, H.R. Wu, "Lyapunov Theory-based Radial Basis Function Networks for Adaptive Filtering", IEEE Transaction on Circuit and System 1, vol 49, no. 8, pp.1215-1221, 2002.
- [8] Slotine, J-J. E. and Li, W. Applied. nonlinear control, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [9] <http://www.astro.oma.be/SIDC/index.html>
- [10] Deb, K., "Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. Proceedings of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN-99),1999.
- [11] L. X. Wang and J. Mendel, "Generating fuzzy rules by learning from examples", IEEE transactions on systems, Man, Cybernetics, vol. 22, pp. 332-342, 1994.