A Hybrid Fuzzy Wavelet Neural Network Combined with Shuffled Frog Leaping Algorithm for Identification of Dynamic Plant

Reza Sharifian Dastjerdi [1], Ramtin Sadeghi [2], Farshad Kabiri [3], Payam Ghaebi Panah [4]

[1,2,3,4] Department of Electrical Engineering, Lenjan Branch, Islamic Azad University, Isfahan, Iran
sharifian@iauln.ac.ir

Abstract: This paper present a Fuzzy Wavelet Neural Network (FWNN) design based on Shuffled Frog Leaping (SFL) Algorithm to improve the function approximation accuracy and general capability of the FWNN. In presented FWNN, the fuzzy rules that contain wavelets are constructed. Each fuzzy rule corresponds to a sub-wavelet neural network (sub-WNN) consisting of wavelets with a specified dilation value. Orthogonal least square (OLS) algorithm is used to determine the number of fuzzy rules and to purify the wavelets for each rule and SFL algorithm is suggested for learning of FWNN parameters. The structure is tested for the identification of the dynamic plant. Simulation results demonstrate effectiveness and ability of proposed approach.
[Reza Sharifian Dastjerdi, Ramtin Sadeghi, Farshad Kabiri, Payam Ghaebi Panah. **A Hybrid Fuzzy Wavelet Neural Network Combined with Shuffled Frog Leaping Algorithm for Identification of Dynamic Plant.** *Life Sci J* 2012;9(4):4416-4420]. (ISSN: 1097-8135). http://www.lifesciencesite.com. 666

Keywords: Shuffled Frog Leaping Algorithm, fuzzy wavelet neural network, identification.

1. Introduction

In recent years, there has been a growing interest in algorithms inspired from the observation of natural phenomenon, such as fuzzy logic, neural network, and heuristic techniques in many scientific and engineering research areas by the researches around the world [1].

Fuzzy technology is an effective tool for dealing with complex, nonlinear processes characterizing with ill-defined and uncertainty factors. Fuzzy rules are based on expert knowledge. The constructing of knowledge base for some complicated processes is difficult. Thus, there are some methods for constructing of fuzzy rules [2]. On the other hand, some characteristics of neural networks such as learning ability, generalization, and nonlinear mapping are used to deal with signal processing, control system, decision making, and so on. However, the main problem of neural networks is that they require a large number of neurons to deal with the complex problems. Moreover, they also result in slow convergence and convergence to a local minimum. In order to overcome these disadvantages, wavelet technology is integrated into neural networks [3, 4].

Recently, based on the combination of feed-forward neural networks and wavelet decompositions, wavelet neural network (WNN) has received a lot of attention and has become a popular tool for function learning [3]. The main characteristic of WNN is that some kinds of wavelet function are used as the nonlinear transformation in the hidden layer of neural network, so time–frequency property of wavelet is incorporated into the learning ability of neural networks.

However, the main problem of WNN with fixed wavelet bases is the selection of wavelet frames because the dilation and translation parameters of wavelet basis are fixed and only the weights are adjustable. The appropriate wavelet transform will result in the accuracy of approximation. Therefore, there are several different methods proposed to solve the problems [5-7].

The complexity and uncertainty of the system can be also reduced and handled by the concepts of fuzzy logic. The local details of non stationary signals can be analyzed by wavelet transforms. The approximation accuracy of the plant can be improved by the self-learning capabilities of neural networks. Therefore, there are many papers that discuss the synthesis of a fuzzy wavelet neural inference system for signal processing, control problems, identification and pattern recognition [3, 8-12].

In this paper, a FWNN combined with SFL algorithm have been proposed to identification of dynamic plant from input-output observations, inspired by the theory of multi resolution analysis (MRA) of wavelet transforms and fuzzy concepts. FWNN consist of a set of fuzzy rules that each rule corresponding to a sub-WNN consists of single scaling wavelets. The difficulties of selecting wavelets are reduced and orthogonal least-square (OLS) algorithm is used to determine the number of fuzzy rules and to purify the wavelets for each rule. Furthermore, in order to improve the function approximation accuracy and general capability of the FWNN system, a self-tuning process that uses the SFL approach is proposed to adjust the parameters of translation, weights, and membership functions.

By minimizing a quadratic measure of the error derived from the output of the system, the design problem can be characterized by the proposed SFL formulation. The solution is directly obtained without any need for complicated computations. Moreover, the efficient method is expected to have good performance without requiring any derivatives or other auxiliary knowledge.

The paper is organized as follows: to make a proper background, the basic concept of the SFL algorithm is briefly explained in Section II. In Section III, the concepts of FWNN and optimization problem are explained. The results of the proposed FWNN-SFL in a simulation example are given in Section IV and conclusion is drawn in Section V.

2. overview of SFL

The SFL algorithm is a memetic meta-heuristic method that mimics the memetic evolution of group of frogs when seeking for the location that has the maximum amount of available food. The SFL is derived from a virtual population of frogs in which individual frogs represent a set of possible solution. Each frog is distributed to a different subset of the whole population referred to as memeplex. The different memeplexes are considered as different culture of frogs that are located at different places in the solution space (i.e. global search). Each culture of frogs performs simultaneously an independent deep local search using a particle swarm optimization like method. Within each memeplex, the individual frogs hold ideas, that can be influenced by the ideas of other frogs within their memeplex, and evolve through a process of change of information among frogs from different memeplexes.

To ensure global exploration, after a defined number of memeplex evolution steps (i.e. local search iterations), information is passed between memeplexes in a shuffling process. Shuffling improves frog ideas quality after being infected by the frogs from different memeplexes, ensure that the cultural evolution towards any particular interest is free from bias. In addition, to improved information, random virtual frogs are generated and substituted in the population if the local search cannot find better solutions, After this, local search and shuffling processes (global relocation) continue until defined convergence criteria are satisfied. The flowchart of the SFL is illustrated in Fig. 1.

The SFL begins with an initial population of "N" frogs P={ X_1 , X_2 ,…, X_N }created randomly within the feasible space Ω. For S-dimensional problems (S variables), the position of the "ith" frog is represented as X_i =(x_{i1} , x_{i2} ,…, x_{iD}). A fitness function is defined to evaluate the frog's position.

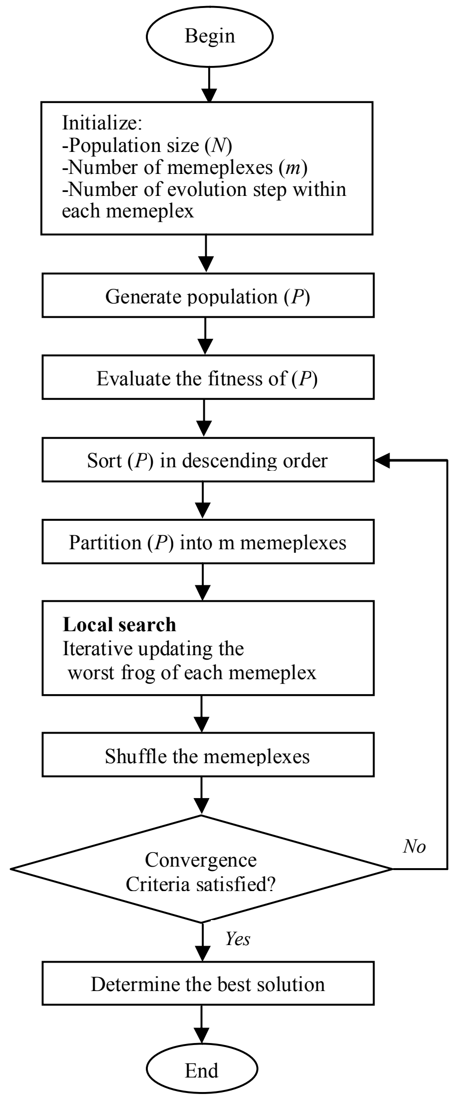

Figure 1. General principle of the SFLA.

Afterward the performance of each frog is computed based on its position. The frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each of which consisting of n frogs (i.e. $N = n \times m$). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the *m*th memeplex, and frog m + 1 back to the first memeplex, and so on.

Within each memeplex, the position of frog ith (D_i) is adjusted according to the different between the frog with the worst fitness (X_w) and the frog with the best fitness (X_b) as shown in (1), where rand () is a random number in the rang [0, 1]. During memeplex evolution, the worst frog X_w leaps toward the best frog X_b . According to the original frog leaping rule, the position of the worst frog is updated as follow:

$$\text{Position change } (D_i) = rand() \times (X_b - X_w) \tag{1}$$

$$X_w(new) = X_w + D, (\|D\| < D_{max}) \tag{2}$$

where D_{max} is the maximum allowed change of frog's position in one jump. If this repositioning process produces a frog with better fitness, it replaces the worst frog, otherwise, the calculation in (1) and (2) are repeated with respect to the global best frog (X_g), (i.e. X_g replaces X_b). If no improvement becomes possible in this case, then a new frog is randomly generated to replace the worst frog. The evolution process is continued for a specific number of iterations ([13, 14]).

3. FWNN and problem formulation

3.1. Fuzzy wavelet neural networks overview

A typical fuzzy wavelet neural network for approximating function y can be described by a set of fuzzy rules such as follow [3]:

R_i : If x_1 is A_1^i and x_2 is A_2^i and … and x_q is A_q^i, then

$$\hat{y}_i = \sum_{k=1}^{T_i} w_{M_i t^k} \Psi_{M_i, t^k}^{(k)}(\underline{x}),\ M_i \in z, t^k \in R^q, w_{M_i}^{t^k} \in R^q, x \in R^q \tag{3}$$

where R_i is the ith rule, c is the number of fuzzy rules. x_j and $\hat{y}_i$ are jth input variable of $\underline{x}$ and output of the local model for rule R_i, respectively. Also M_i is dilation parameter and T_i is the total number of wavelets for the ith rule. $t^k = [t_1^k, t_2^k, ..., t_q^k]$, wher t_j^k denotes the translation value of corresponding wavelet k. Finally, A_i^j is the fuzzy set characterized by the following Gaussian-type membership function.

$$A_j^i(x_j) = e^{-\left(\left(\frac{(x_j - p_{j1}^i)}{p_{j2}^i}\right)^2\right)^{p_{j3}^i/2}} \tag{4}$$

p_{j1}^i, $p_{j2}^i \in R$ and $p_{j3}^i = 2$, where p_{j1}^i represents the center of membership function, p_{j2}^i and p_{j3}^i determine the width and the shape of membership function, respectively. Wavelets $\psi_{M_i, t^k}^{(k)}(\underline{x})$ are expressed by the tensor product of 1-D wavelet functions:

$$\psi_{M_i,t}^{(k)}(\underline{x}) = 2^{\frac{M_i}{2}} \psi^{(k)}(2^{M_i}\underline{x} - \underline{t}^k) = \prod_{j=1}^{q} 2^{\frac{M_i}{2}} \psi^{(k)}(2^{M_i} x_j - t_j^k) \tag{5}$$

By applying fuzzy inference mechanism and let $\hat{y}_i$ be the output of each sub-WNN, the output of FWN for function $y(\underline{x})$ is as follow:

$$\hat{y}_{FWN}(\underline{x}) = \sum_{i=1}^{c} \hat{\mu}_i(\underline{x}) \hat{y}_i \tag{6}$$

where $\hat{\mu}_i(\underline{x}) = \mu_i(x) / \sum_{i=1}^{c} \mu_i(x)$, $\hat{\mu}_i(x) = \prod_{j=1}^{q} A_j^i(x_j)$ and for current input $\underline{x}$ and each function, satisfies $0 \le \hat{\mu}_i \le 1$ and $\sum_{i=1}^{c} \hat{\mu}_i = 1$. Also $\hat{\mu}_i(x)$ determines the contribution degree of the output of the wavelet based model with resolution level M_i. In this paper the applied structure of FWNN is the same as the structure used in [12]. Also orthogonal least-square (OLS) algorithm is used to select important wavelets and to determine the number of fuzzy rules. Details of this OLS algorithm can be found in [15].

3.2. Tuning parameters of FWNN

Assume that there are H input–output pairs, $(x(l), y(l))$, l =1, . . . , H. Our task is to design the fuzzy basis function expansion such that the error between $\hat{y}_{FWN_k}(x(l))$ and $y(l)$ is minimized. Therefore SFL is applied for tuning parameters of FWNN by optimizing the following objective or cost function.

$$E = \sum_{l=1}^{H} \left| \hat{y}_{FWN_k}(x(l)) - y(l) \right|^2 \tag{7}$$

and, the Nth frog is represented as

$$F_N = [p_{j1}^{iN} p_{j2}^{iN} \underline{t}^{kN} w_{M_i}^{N}] \tag{8}$$

which are all free design parameters that to be updated by SFL algorithm in our FWNN model. Summarized the whole proposed approach is illustrated in Fig. 2.

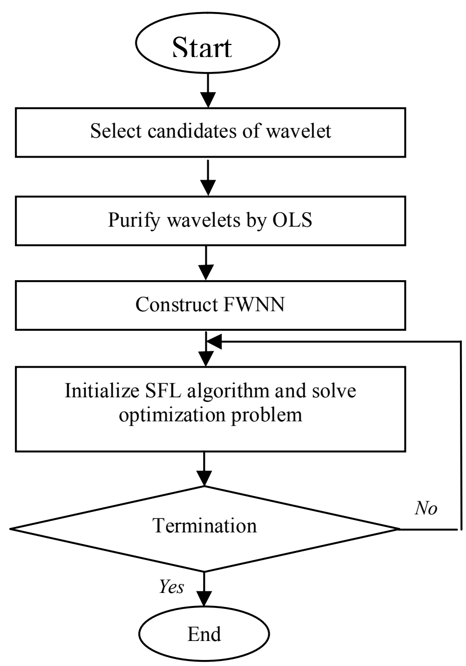

Figure 2. The algorithm of constructing FWNN

4. Design example

The identification problem involves finding the relation between the input and the output of the system. In order to evaluate the effectiveness and efficiency of the proposed approach, we repeat a simulation example from [15], where a FWNN-SFL employed to identify a nonlinear component in a control system. The plant under consideration is governed by the following difference equation:

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)) \qquad (9)$$

Where $y(k)$ and $u(k)$ are the output and input, respectively, at time step k . The unknown function $f(.)$ has the form

$$f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \qquad (10)$$

In order to identify the plant, a series-parallel model governed by the difference equation as follow:

$$\hat{y}(k+1) = 0.3\hat{y}(k) + 0.6\hat{y}(k-1) + F(u(k)) \qquad (11)$$

Was used, where $F(.)$ is the function implemented by FWNN-SFL and its parameters are updated at each time step. In the learning process, the input to the plant and the model is considered as a sinusoid $u(k) = \sin(2\pi k / 250)$ in interval $k = 1$ until $k = 250$ and $u(k)$ is changed to $0.5\sin(2k\pi / 250) + 0.5\sin(2k\pi / 25)$ at $k = 250$ until $k = 500$.

Three fuzzy rules with 4 selected wavelets are represented by OLS algorithm for constructing FWNN. The first step to implement the SFL is generating the initial population where N is considered to be 300. Each population is a solution to the problem which determines the parameters of FWNN, i.e. F_N vector. In this paper, the number of iteration is set to be 1000.

The output and control signal are shown in Fig. 3. As shown in Fig. 3, the output of the model follows the output of the plant almost immediately, even after the adaptation was stopped at $k = 250$ and $u(k)$ was change to $0.5\sin(2k\pi / 250) + 0.5\sin(2k\pi / 25)$.

To compare the obtained result by SFL, a simple Genetic Algorithm (GA) is applied. The number of chromosomes in the population is set to be 200. Also, the number of iterations is considered to be 1000, which is the stopping criteria used in SFL. The approximation of piecewise function obtained by suggested method is presented in Fig. 4.

For the designed parameters, the average best-so-far of each run are recorded and averaged over 10 independent runs. To have a better clarity, the convergence characteristics in finding the best values of update parameters are given in Fig.5. Where shows SFLA performs better than GA at early iterations.

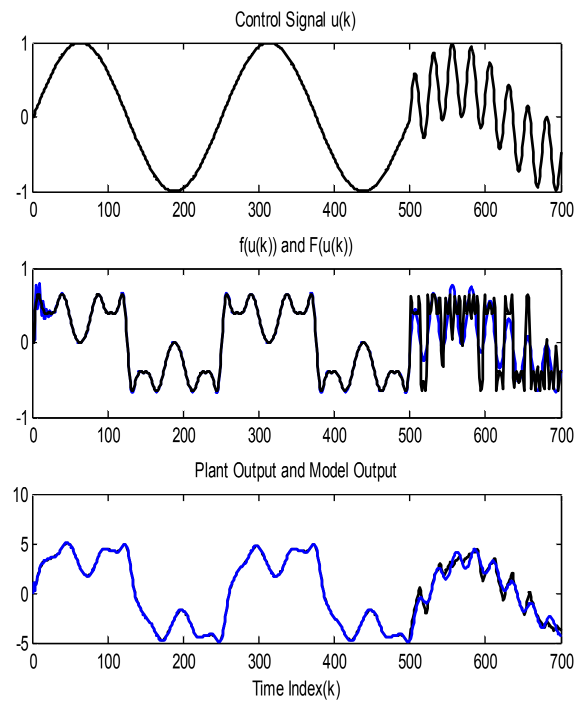

Figure 3. Results of identification, where solid line denotes the output of the plant, dashed line denotes the FWNN-SFL output.

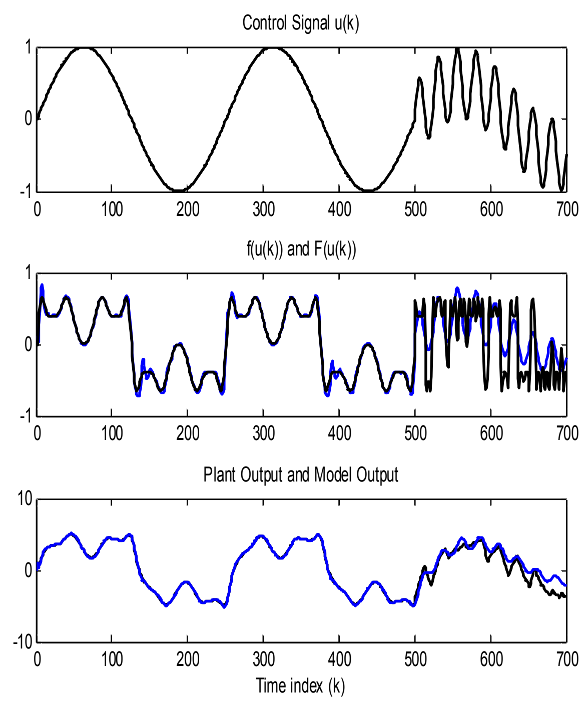

Figure 4. Results of identification, where solid line denotes the output of the plant, dashed line denotes the FWNN-GA output.

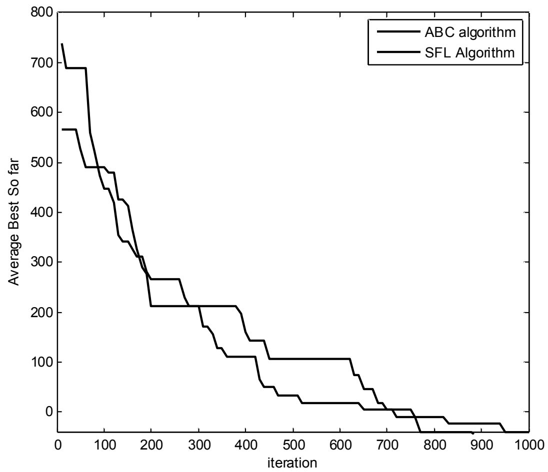

Figure 5. Convergence characteristics of SFLA and GA.

5. Conclusion

In this paper, a new memetic meta-heuristic method called Shuffled Frog Leaping Algorithm (SFLA), combined with FWNN for function learning is proposed. Proposed approach integrates the advantages of fuzzy concepts with the WNNs and evolutionary algorithms. In the presented FWNN, OLS algorithm is used for determining the number of fuzzy rules and to purify the wavelets for each rule and a real version of SFL algorithm is used for tuning parameters of FWNN. The simulation results are presented show the efficiency and effectiveness of the proposed approach. Also, the GA is adopted from literature and applied for comparison. The obtained results demonstrate that SFL has faster convergence and better performance than GA.

Corresponding Author:
Reza Sharifian Dastjerdi
Department of Electrical Engineering,
Lenjan Branch, Islamic Azad University,
Isfahan, Iran.
E-mail: sharifian@iauln.ac.ir

References

[1] M. Thuillard, Wavelets in Soft computing, World Scientific, Singapore, 2001.
[2] R. R. Yager, L. A. Zadeh, Eds., Fuzzy Sets, Neural Networks and Soft Computing. New York: Van Nostrand Reinhold, 1994.
[3] D.W.C. Ho, P.A. Zhang, J. Xu, Fuzzy wavelet networks for function learning, IEEE Transactions on Fuzzy Systems 9 (1), 2001, pp. 200–211.
[4] S. Tang Tzeng, Design of fuzzy wavelet neural networks using the GA approach for function approximation and system identification, Fuzzy Sets and Systems 161, 2010, pp. 2585–2596.
[5] Q. Zhang, Using wavelet networks in nonparametric estimation, IEEE Transactions on Neural Networks 8, 1997, pp. 227–236.
[6] J. Chen, D.D. Bruns, WaveARX neural network development for system identification using s systematic design synthesis, Industrial & Engineering Chemistry Research 34, 1995, pp. 4420–4435.
[7] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Transactions on Neural Networks 3, 1992, pp. 889–898.
[8] R.H. Abiyev, O. Kaynak, Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study, IEEE Transactions on Industrial Electronics 55 (8), 2008, pp. 3133–3140.
[9] R.H. Abiyev, O.Kaynak, Identification and control of dynamic plants using fuzzywavelet neural networks, 2008 IEEE International Symposium on Intelligent Control, San Antonio, Texas, USA, September 3–5, 2008, pp. 1295-1301.
[10] Y. Lin, F.Y. Wang, Predicting chaotic time-series using adaptive wavelet-fuzzy inference system, in: Proceedings of 2005 IEEE Intelligent
[11] C.K. Lin, S.D. Wang, Fuzzy modeling using wavelet transform, Electronic Letters 50 (6), 1996, pp.1317–1334.
[12] E. Elbeltagi, A Modified shuffled-frog leaping algorithm for optimizing bridge-desk repairs, international conference on bridge management systems.2006.
[13] M.M. Eusuff, K.E. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," Engineering Optimization, Vol. 38, no. 2, 2006, pp.129-154.
[14] S. R. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing.Englewood Cliffs, NJ: Prentice-Hall, 1997.
[15] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 ,1991.

12/7/2012