IJE TRANSACTIONS B: ApplicationVol. 27, No. 8, (August 2014) 1185-1194

International Journal of Engineering

Journal Homepage: www.ije.ir

A Differential Evolution and Spatial Distribution based Local Search for Training Fuzzy Wavelet Neural Network

H. A. Bazoobandi, M. Eftekhari*

Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

PAPER INFO

Paper history:
Received 27October 2013
Received in revised form 15March2014
Accepted in 17 April 2014

Keywords:
Fuzzy Wavelet Neural Network (FWNN)
Memetic Algorithm
Differential Evolution
Spatial Distribution Local Search
Adaptive Selection Strategy

ABSTRACT

Many parameter-tuning algorithms have been proposed for training Fuzzy Wavelet Neural Networks (FWNNs). Absence of appropriate structure, convergence to local optima and low speed in learning algorithms are deficiencies of FWNNs in previous studies. In this paper, a Memetic Algorithm (MA) is introduced to train FWNN for addressing aforementioned learning lacks. In proposed MA, Differential Evolution (DE) is utilized as the global search. The main contributions of this paper aresummarized in three sections. (I) Proposing a new, fast and effective local search based on spatial distribution that is named Spatial Distribution Local Search (SDLS). SDLS can adjust the step size of movements toward better neighbor solutions adaptively. (II) Introducing a selection method to select appropriate individuals from current population for local refinement in MA. This property decreases the computational cost of MA and leads to tuning the local search frequency in an adaptive way. (III) Improving the selection operator in standard DE by an adaptive strategy. In this strategy, worse offspring has a chance to be replaced with its parent to prevent trapping in local optima and controlling the selection pressure. The proposed MA is compared with several training algorithms of FWNNs over some benchmark problems. Experimental results obtained, confirm the effectiveness of the proposed MA for improving the convergence rate and modeling accuracy in comparison to the other training methods.

doi: 10.5829/idosi.ije.2014.27.08b.04

1. INTRODUCTION

In much recent researches, soft computing methodologies such as fuzzy logic, Neural Networks (NNs), and wavelet neural networks have been used. Especially fuzzy wavelet neural network has been used as a viable tool for solving technological processes. Fuzzy logic can handle uncertainty in terms of structure and parameters of dynamic systems. Neural networks have well training ability to learn from examples and wavelet functions have fine properties such as localization and multi-resolution. FWNN integrates fuzzy logic, neural network, and wavelet functions to achieve a single powerful model which has all of the aforementioned properties. A wide range of problems can be solved using FWNNs with more efficiency than traditional NNs such as function learning [1], prediction problems [2], and system identification problems [3-5].

There are two challenges about employment of FWNNs. The first one is how to initialize FWNN parameters, and the second which is more essential is how to train it. A number of methods have been introduced in the literature to initialize parameters of wavelet-based neural networks which we review them in the following briefly. Simplest methods for the aim of initialization of FWNN parameters are random values [6-8] or pre-defined constant values [2]. Clustering methods such as K-means, and Subtractive have been proposed [9, 10] to initialize the network parameters. In some researches, parameters will be selected from a pre-defined library such that the error is minimized [11]. However, precise initialization of parameters becomes critical when local search approaches are used for the

*Corresponding Author's Email: m.eftekhari@mail.uk.ac.ir (M. Eftekhari)

Please cite this article as:H. A. Bazoobandi, M. Eftekhari, A Differential Evolution and Spatial Distribution based Local Search for Training Fuzzy Wavelet Neural Network. International Journal of Engineering (IJE) Transactions B: Applications Vol. 27, No. 8, (2014): 1185-1194

aim of FWNN training (like the aforementioned references). Global searches generally can escape from local optima even without good initialization. In this paper, we concentrate on training phase and initialization is performed randomly in the intervals of input spaces because DE is utilized as a global search.

Training an FWNN is important due to good generalization capability of the final model and fast speed of convergence in training phase. Much of the research has employed derivative based methods for training FWNN. Linear derivative based methods such as Recursive Orthogonal Least Square (ROLS) has been used to tune just linear part of FWNN parameters [3]. Researches in which just a linear method for learning has been used, usually determine nonlinear part of parameters with pre-defined constant values. To tune overall parameters of FWNN, nonlinear derivative based methods also were widely used such as Back Propagation (BP) and Stochastic Gradient (SG) [7, 12, 13]. Some hybrid derivative-based approaches also were proposed in the literature for tuning FWNN parameters. Since Least Square (LS) algorithm usually is much faster than nonlinear methods, LS was used to tune linear part of parameters (weights) and Extended Kalman Filter (EKF) was utilized to adjust nonlinear ones [1, 14, 15]. Bodyanskiy and Vynokurova combined a modified version of BP learning method based on chain rule of differentiation and gradient descent optimization of local criterion [10]. Abiyev et al. introduced type-2 FWNN which employs fuzzy c-means clustering to design the antecedent part, and the gradient algorithm to design the consequent part of fuzzy rules [16]. In the other hand, derivative free optimization approaches also were employed for training FWNN. Sampling theory was utilized for training wavelet networks by Zhang [17]. Evolutionary algorithms (EAs) also have shown suitable results for training FWNN. For example, a real coded Genetic Algorithm (GA) was used [8]. Simultaneously Wei et al. have utilized PSO in their work [4]. Among EAs, DE performs greedy and has good exploitation ability. This feature makes DE suitable for training NNs. In a number of recent researches [18, 19], DE outperformed the other training methods among EAs and gradient-based ones.In the present paper, an MA isproposed for training FWNN. Proposed MA combines DE as a global search to find suitable regions in the search area and a new local search based on spatial distribution, which tries to find better solutions in the neighborhood of current solution. Selection operator in DE is replaced by a method which worse offspring has a chance to be replaced by its parent to survive in the next generation. Furthermore, a new strategy is introduced to adjust the local search frequency and find appropriate individuals for local refinement in an adaptive way. The rest of paper is organized as follows: in Section 2 FWNN preliminaries are defined. Section 3 describes our proposed MA. Simulation results are analyzed in Section 4. Finally, concluding remarks appear in Section 5.

2. FWNN

The structure of FWNN can be illustrated by a number of fuzzy TSK-type IF-THEN rules. The antecedent parts of rules in FWNN are similar to standard TSK-type fuzzy rules, but in the consequent part of them, wavelet functions have been embedded. Wavelet functions can improve the computational power of neuro-fuzzy system. Each rule can extract properties of its own intervals; therefore can be seen as a local model. The form of i^{th} rule is shown in the followingform:

$$\begin{aligned} &R^i: \text{ If } x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i, \ldots, \text{ and } x_q \text{ is } A_q^i \\ &\text{Then } y_i = w_i \sum_{j=1}^{q} \psi_{ij}(x_j), \end{aligned} \tag{1}$$

where x_j $(1<j<q)$ indicates input signals, y_i $(1<i<c)$ is the output signal of the rule R^i. Output signal for each rule is a linear combination of q wavelets. A_j^i shows membership function for j^{th} input of i^{th} rule, and w_i is a weight coefficient between input signals and i^{th} output. ψ_{ij} shows a family of wavelets obtained from a single mother wavelet function $\psi(x)$ by translation and dilation, see Equations (2) and (3).

$$\psi_{ij}(x_j) = \psi\left(\frac{x_j - t_{ij}}{d_{ij}}\right), \; d_{ij} \neq 0, \tag{2}$$

where t and d describe the translation and dilation parameters, respectively. Translation moves the wavelet either to left or to right on time axis. If we decrease the dilation parameter then wavelet will be compressed and rapidly-changed details of the original signal are extractable. On the contrary, high values for dilation parameter make the wavelet stretch and therefore slowly-changed details of signal could be studied. In this study, we use the Mexican Hat function as mother wavelet, which is defined as follows:

$$\psi(x) = \frac{1}{\sqrt{|d|}}(1-2x^2)\exp\left(\frac{-x^2}{2}\right). \tag{3}$$

Gaussian membership function is used in the antecedent part of fuzzy rules. The membership degree in the Gaussian function is calculated as follows:

$$A_j^i(x_j) = \exp\left[-\left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right], \tag{4}$$

where c and σ show the center and half-width of the Gaussian membership function, respectively.

The estimated output for each sample can be calculated by the following equation:

$$u = \frac{\sum_{i=1}^{c} \mu_i y_i}{\sum_{i=1}^{c} \mu_i}, \tag{5}$$

where μ_i indicates the firing strength of i^{th} fuzzy rule, which is defined as follows:

$$\mu_i(x) = \prod_{j=1}^{q} A_j^i(x_j). \tag{6}$$

3. PROPOSED MEMETIC ALGORITHM

MAs are population-based meta-heuristics composed of an EA and a set of local searches which are activated within the generation cycle [20]. MAs include two phases: global search and local search. Global search is performed by an EA in a memetic search to find better regions of solution space and increase the exploration ability. Local search phase in MA tries to enhance the exploitation ability.Thus, if a good tradeoff between exploration and exploitation (global and local searches) can be created, then nice optimal solutions will become available as well as enhancement in convergence speed.
In this section, a new MA is proposed for training FWNN which combines DE as a global search, and a local search based on spatial distribution. The main goal of proposed memetic algorithm is to achieve a fine balance between global search (i.e. finding new promising regions of the solution space) and local search (i.e. concentrating the search on promising regions) to find better solutions.

Overfittig is one of the important issues in modelling that occurs when prediction error of the final model in training data becomes very lower than unseen data. There are two basic aspects which can be considered about overfitting in NNs: model complexity and training algorithm [18]. If the final model is to be complex then the probability of overfitting occurrence will increase. The structure of final FWNN models is generally simpler than the other similar modeling tools such as NNs. This feature decreases the probability of overfitting in FWNNs. From another point of view, training algorithms are effective to avoid overfitting. Simulation results in previous researches have confirmed the ability of DE in training NNs and FWNNs [18, 19]. Therefore, it can be concluded that the combination of FWNNs and DE can avoid overfitting potentially.

3. 1. Global Search Phase

DE is a fast population-based EA that searches more greedy and less stochastic compared with other EAs. Subsequent generations in DE are denoted by $g = 0, 1, ..., g_{max}$. DE operators (i.e. mutation, crossover, and selection) are applied to individuals from current population repeatedly until the termination criterion is met.Without loss of generality, a minimization problem F(X) with D dimensions can be shown as follows:

$$\text{Min } F(X), \quad X = [x^1, x^2, ..., x^D], \tag{7}$$

where F(X) is the objective function, here F(x) is the error of our FWNN model. X is a decision vector. In our memetic approach, decision vector describes the FWNN parameters and is coded for i^{th} individual as follows:

$$X^i = [\vec{c_i} \;\; \vec{\sigma_i} \;\; \vec{t_i} \;\; \vec{d_i} \;\; \vec{w_i}], \tag{8}$$

where each of the above five parameters will be a vector. If the length of problem input space is equal to *q* and we use *c* rules in FWNN model, then the size of X^i is equal to $D = (4q+1)c$.

3. 1. 1. Mutation

Mutation type is the main difference between DE and GA. In DE, mutated vector V can be generated as follows:

$$V = X_{r1} + F(X_{r2} - X_{r3}) \tag{9}$$

where F is called scaling factor which provides the amplification to the difference between two individuals and is usually taken in [0, 1] [19]. X_{r1}, X_{r2}, and X_{r3} are selected randomly from current population in the simplest way. However, in this study they are selected by a roulette wheel procedure.

3. 1. 2. Crossover

In standard version of DE, crossover is defined as follows:

$$U = \begin{cases} V & \text{rand} < CR \\ X & \text{Otherwise} \end{cases} \tag{10}$$

where U is a new offspring producing by mating of V and X. CR is called crossover rate which is limited to [0, 1].

3. 1. 3. Selection

Selection is a procedure, which makes competition in evolution cycle. The candidate offspring U and its parent X are competing with each other to survive in the next generation. In standard DE, the best individual from {X, U} will be selected to continue survival in the next generation. This simple method may cause trapping in local optima. To address such situation we propose a selection operator which is as follows:

$$X_{new} = \begin{cases} 1)\ U \text{ if fitness}(U) < \text{fitness}(X) \\ 2)\ U \text{ else if, rand} > \dfrac{\exp(-\text{fitness}(X)) + (\frac{g}{g_{max}})}{2} \\ 3)\ X \text{ Otherwise} \end{cases} \tag{11}$$

where $\text{fitness}(.)$ is a function that evaluates the degree of goodness of an individual. g and g_{max} are current and the maximum number of iterations in DE, respectively. This simple procedure makes DE more powerful in escaping from local optima and controlling the selection pressure. Probability of replacement ofthe worse offspring with its parent directly depends on individual fitness and current iteration. The first part of Equation (11) shows the situation in which the offspring (U) is better than the parent (X). The second part indicates the situation that the offspring (U) is worse than its parent.

3. 2. Local Search Phase

Local search would be performed meanwhile global search would be carried out to improve the quality of solutions and convergence speed of MA.

3. 2. 1. Which Individual Appropriate to Undergo Local Refinement?

In the literature [21], the impact of local search frequency (frequency of local search performed during global search) was studied. It was concluded in that work [21] that a good parameterization of local search frequency has a strong impact on the performance of MA. However, parameterization of an MA may be extremely hard [21]. Generally, local search in MAs is performed with a fixed frequency. In this section, the impact of local search frequency is considered and an adaptive solution is proposed to adjust it.

There are common ways for selecting individuals to undergo local refinement. Local search can call for all individuals or for number of individuals, usually better ones or even the best [22]. Zhang et al. employed a probability measure to judge about accomplishment of local search on current population [23]. Bao et al. proposed a selection method, which selects better individuals for exploitation and then ignores their neighbor individuals with a pre-defined constant radius [24]. Determination of such radius is difficult, problem dependent, and needs trial and error. Many other methods have been proposed in the literature; however, these procedures seem too blind. To this end, a new strategy is proposed in this paper to find appropriate individuals for local refinement and tuning the local search frequency. The introduced strategy makes a tradeoff between the exploration ability in the first generations and exploitation in final ones.

In the proposed strategy, after each generation of DE, a radius parameter is calculated adaptively based on the variance of current population fitness and the current generation as follows:

$$\mathrm{radius}(g) = \exp(\mathrm{var}(\mathrm{fitness}(g))).(g_{max} - g), \tag{12}$$

where $\mathrm{var}(\mathrm{fitness}(g))$ is the variance of current population fitness values. In the next stage, a list which is named "exploitation list" is constructed at the beginning of local search phase which includes all individuals in the population. At first, the fittest individual will be selected from exploitation list which undergoes local refinement. Individuals similar to selected individual are removed from exploitation list based on adaptive radius that is caculated by Equation (12). This procedure is continually performed until the exploitation list becomes empty. It is clear from Equation (12) that radius takes higher values in the first generations and lower values in final ones. Therefore the freuency of local search becomes lower in the first generations and exploration ability increases. In contrast, exploitation ability will be enhanced in final generations by increasing the local search frequency.

3. 2. 2. Spatial Distribution Local Search (SDLS)

In this section, a local search based on spatial distribution is proposed . In SDLS, the step size of movement is given by normally distributed random number with zero mean and variance s^2, where s is calculated as follows:

$$\sigma_t = \sigma_{min} + \left(\frac{t_{max} - t}{t_{max}}\right)^{nmi} .(\sigma_{max} - \sigma_{min}), \tag{13}$$

where s_{min} and s_{max} are lower and upper bounds for standard deviation, respectively. nmi represents nonlinear modulation index, which controls the importance of the generation parameter. In Equation (13), t shows the current iteration and t_{max} denotes the maximum number of local search iterations. Steps of SDLS are given in Figure 1.In Figure 1, sign is a vector which shows the direction of movements for each parameter.

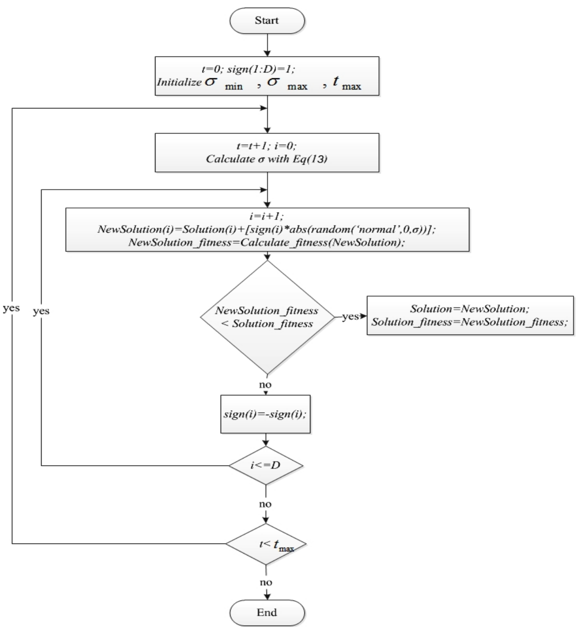

Figure 1. The Flowchart of SDLS

4. RESULTS AND DISCUSSIONS

In this section, some known problems taken from the literature are employed to verify the effectiveness of the proposed DE-SDLS. Evaluation of models is given in terms of the final model performance (error and complexity). Furthermore, the computational cost of our proposed MA is compared with the FWNN-GA [8]. For the aim of an exhaustive comparison, 8 different versions of DE are simulated and utilized in comparisons. Descriptions about these methods are available from other works [25]. Parameters of DE-SDLS are chosen as follows: NP=50, F=0.02, CR=0.5 and t_{max} =50.

4. 1. Approximation of a Piecewise Function

The aim is to approximate the Equation (14) piecewise single variable function.

$$f(x)=\begin{cases}-2.186x-12.864, & -10\le x\le -2\\ 4.246x, & -2\le x\le 0\\ 10e^{-0.05x-0.5}\sin[(0.03x+0.7)x], & 0\le x\le 10.\end{cases} \tag{14}$$

200 samples are generated, which are distributed uniformly over [-10, 10]. Comparison between the original and the output of trained FWNN by DE-SDLS are depicted in Figure 2. For the purpose of comparison with other reported works, the following RMSE is used as the performance criterion:

$$RMSE=\sqrt{\frac{\sum_{l=1}^{L}(y_l-y_l^d)^2}{\sum_{l=1}^{L}\left(y_l^d-\bar{y}\right)^2}}, \tag{15}$$

$$\bar{y}=\frac{1}{L}\sum_{l=1}^{L}y_l^d, \tag{16}$$

where y_l^d is the desired output, y_l shows the estimated output of FWNN model, and $\bar{y}$ is the mean of y_l^d and can be calculated by Equation (16). L is the number of training samples;here is equal to 200. Figure 3 shows the improvement of RMSE values during the training process. The comparison of the DE-SDLS algorithm and the other reported works are given in Table 1. It can be seen from Table 1 that the proposed method dominates previous works in both obtained error and model complexity. Although the SSF-wavenet finds better error value it must be noted that the SSF-wavenet uses more complex model than the others.

4. 2. Identification of Nonlinear Dynamic Plant

In this example, the performance of our proposed MA is evaluated on a second order nonlinear dynamic plant. The process of identification problem in this example can be described with the following relation:

$$y(k)=f(y(k-1),y(k-2),y(k-3),u(k),u(k-1)), \tag{17}$$

where y(k) and u(k) describe the output and input signals of the plant in time k, respectively.

TABLE 1. Comparison of results of our MA (SDLS-DE) and the other works for example 1.

Method	Parameters		RMSE
	# of rules	Epoch	
DE	4	5000	0.4640
TDE	4	5000	0.179
DEahcSPX	4	5000	0.442
DEPSR	4	5000	0.2937
jDE	4	5000	0.2894
OBDE	4	2000	0.5059
DEGL	4	5000	0.2068
SADE	4	5000	0.4561
DE-SDLS(Proposed)	**4**	**200**	**0.0105**
FWN [1]	6	-	0.021
SSF-wavent [3]	13	-	0.0071
MWNN-MH [6]	-	-	0.053
FWNN-GA [8]	4	5000	0.0303
FWN [12]	10	-	0.022
Type-2 FWNN [16]	4	5000	0.010
ANFIS	9	400	0.019

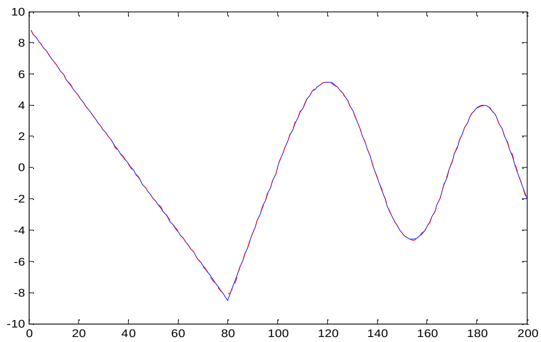

Figure 2. Comparison between the original function (solid line) and the estimated output of FWNN (dashed line) for example 1.

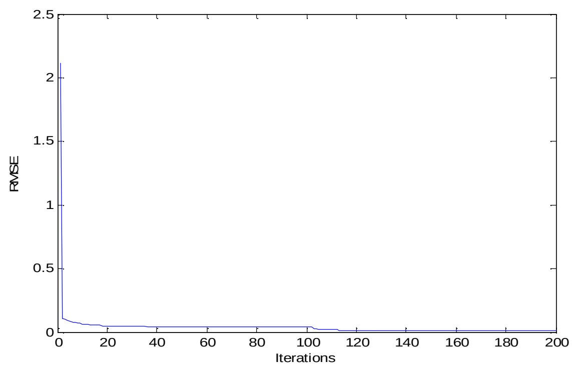

Figure 3.RMSE values obtained during learning for example 1.

Function f(.) is defined by the following equation:

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^{\,2} + x_2^{\,2}}. \quad (18)$$

Input signals u(k) can be calculated as follows:

$$u(k) = \begin{cases} \sin\left(\frac{\pi k}{25}\right), & k < 250 \\ 1, & 250 \le k < 500 \\ -1, & 500 \le k < 750 \\ 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) + \\ 0.6\sin\left(\frac{\pi k}{10}\right), & 750 \le k < 1000. \end{cases} \quad (19)$$

Two fuzzy rules and 1000 samples are employed to identify the input function behavior. The following RMSE is used as the performance criterion:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{L}\left(y(i) - y_d\right)^2}{L}} \quad (20)$$

Figure 4 shows the difference between desired and estimated output. The speed of error reduction can be seen from Figure 5. Figures 4 and 5 show the ability of our proposed DE-SDLS MA in identification of nonlinear dynamic plant.

Comparisons with other methods are given in Table 2. The DE-SDLS method shows superior performance while it employs very simpler model.

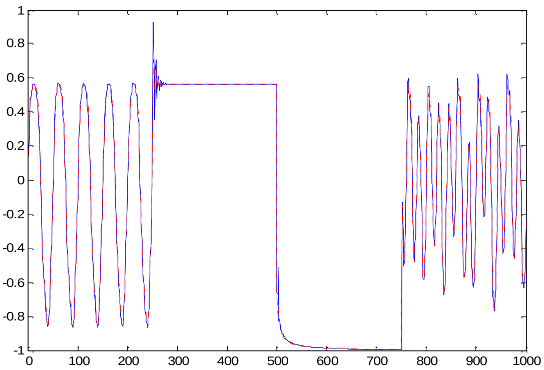

Figure 4. Comparison between the original function (solid line) and the estimated output of FWNN (dashed line) for example 2.

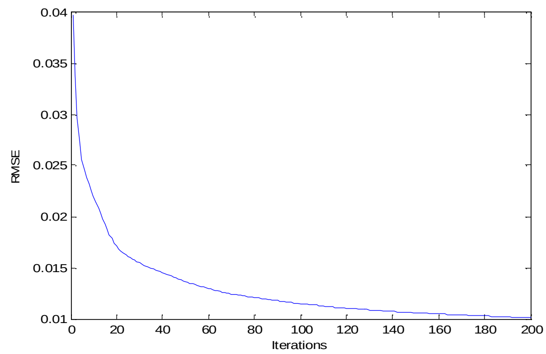

Figure 5. RMSE values obtained during learning for example 2.

TABLE 2. Comparison of results of our MA (SDLS-DE) and the other works for example 2.

	Parameters		
Method	**# of rules**	**Epoch**	**RMSE**
DE	2	5000	0.0548
TDE	2	5000	0.0522
DEahcSPX	2	5000	0.0404
DEPSR	2	5000	0.0472
jDE	2	5000	0.0704
OBDE	2	2000	0.0761
DEGL	2	5000	0.0491
SADE	2	5000	0.0369
DE-SDLS(Proposed)	**2**	**200**	**0.0101**
FWNN-S [5]	32	5000	0.0208
FWNN-R [5]	32	5000	0.0152
FWNN-M [5]	32	5000	0.0192
FWNN [7]	3	200	0.0292
FWNN [7]	5	200	0.0282
FWNN-GA [8]	2	5000	0.029
HAWNFS [10]	-	30	0.0183
RFNN [13]	16	100	0.0114
Feedforward neural network*	-	-	0.0203
ANFIS*	36	400	0.03

*Results are taken from [10].

4. 3. Identification of Nonlinear Dynamic Plant

In this example, another nonlinear system identification problem is considered to evaluate the performance of our proposed training method. This problem is defined in the following form:

$$y(k) = 0.72y(k-1) + 0.025y(k-2)u(k-2) + 0.01u^2(k-3) + 0.02u(k-4) \quad (21)$$

where u can be obtained in a similar way as the previous example by Equation (19). The RMSE appeared in Equation (20) is employed as performance measure. Comparison of results for DE-SDLS with the other works is given in Table 3.

It is clear from Table 3 that the proposed MA achieves much better results in terms of both error and model complexity and outperforms all of the previous works strictly. There exist another important matter which can be figured out from Table 3 that the EAs are powerful for training FWNN.

It can be seen that the EAs in Table 3 find better RMSE values than local search ones such as FWNN [5], FWN [7], and type-2 FWNN [16].The estimated output by FWNN is compared with the desired output of the model in Figure 6. RMSE values improvement during the training is depicted in Figure 7.

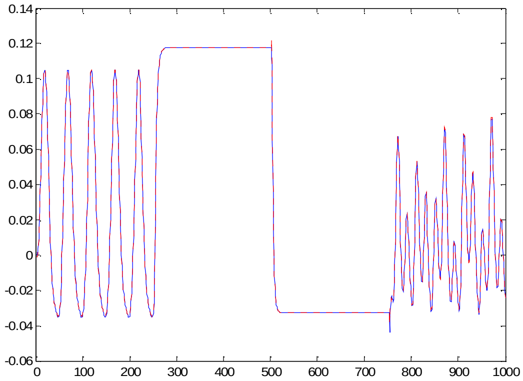

Figure 6. Comparison between the original function (solid line) and the estimated output of FWNN (dashed line) for example 3.

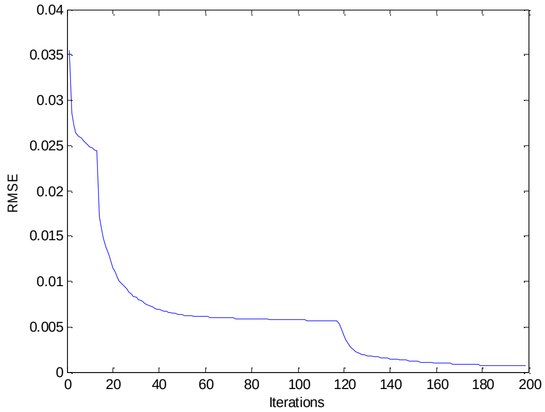

Figure 7.RMSE values obtained during learning for example 3.

TABLE 3. Comparison of results of Our MA (SDLS-DE) and the works for example 3.

Method	Parameters: # of rules	Parameters: Epoch	RMSE
DE	3	5000	0.0053
TDE	3	5000	0.0035
DEahcSPX	3	5000	0.0036
DEPSR	3	5000	0.0064
jDE	3	5000	0.0025
OBDE	3	2000	0.0132
DEGL	3	5000	0.0069
SADE	3	5000	0.005
DE-SDLS(Proposed)	**3**	**200**	**0.000704**
FWNN-S[5]	32	5000	0.009771
FWNN-R[5]	32	5000	0.009688
FWNN-M[5]	32	5000	0.009635
FWN[7]	3	200	0.019736
FWN[7]	5	200	0.018713
FWNN-GA[8]	2	5000	0.0044
Type-2 FWNN [16]	3	200	0.01667
Type-2 FWNN+FCM [16]	4	200	0.01462

TABLE 4. Comparison of DE-SDLS with FWNN-GA in terms of average number of fitness calls.

Method	Number of fitness function evaluation: Ex. 1	Ex. 2	Ex. 3
FWNN-GA [8]	475800	410712	310168
DE-SDLS	**43913**	**20626**	**28270**

4. 4. Evaluation of DE-SDLS Time Complexity

Time complexity is one of the important issues in considering effectiveness of algorithms. In EAs, the time complexity on a given problem refers to the number of times which the EA evaluates the fitness function before an acceptable solution to be found. In this section the proposed method is compared with FWNN-GA method which is proposed in the literature [8] in terms of time complexity. This comparison is given in Table 4. We counted number of fitness function evaluations for 20 runs of DE-SDLS and the average of them reported in Table 4. It is clear from Table 4 that the DE-SDLS has lower time complexity and therefore is executed so faster than FWNN-GA. The following values are used as error threshold to stop the fitness evaluation counting: Ex.1:0.03, Ex. 2:0.03, and Ex. 3: 0.007.

5. CONCLUSIONS

In this paper, a MA was proposed for training FWNN. DE was used as global search and a new method based on spatial distribution was employed as local search. Proposed local search guides the solution toward better ones without requirement of adjusting any parameter. Selection operator of DE was modified such that worse solution in the first generations has a chance to be selected, like selection procedure in simulated annealing. Additionally, an adaptive selection method was introduced to find appropriate individuals for local refinement and tuning local search frequency during the MA. The proposed DE-SDLS leads to a good tradeoff between exploration and exploitation abilities. Despite fewer number of rules and hence simpler model and lower computational cost of learning algorithm, the performance was improved in terms of the obtained RMSE values.

DE-SDLS is a derivative free optimization method. This important property makes the FWNN to be very suitable for complex modeling problems where no differentiable function may be used. The obtained experimental results over some non-linear modeling benchmark problems confirm the efficiency of DE-SDLS in comparison to the other reported works about FWNN training and different versions of DE. Furthermore, the results justify the suitable computational cost of DE-SDLS.

6. REFERENCES

1. Ho, D.W., Zhang, P.-A. and Xu, J., "Fuzzy wavelet networks for function learning", ***Fuzzy Systems, IEEE Transactions on***, Vol. 9, No. 1, (2001), 200-211.

2. Mousavi, S.H., Noroozi, N., Safavi, A.A. and Ebadat, A., "Modeling and control of nonlinear systems using novel fuzzy wavelet networks: The output adaptive control approach", ***Communications in Nonlinear Science and Numerical Simulation***, Vol. 16, No. 9, (2011), 3798-3814.

3. Ebadat, A., Noroozi, N., Safavi, A.A. and Mousavi, S.H., "New fuzzy waveletnetwork for modeling and control: The modeling approach", ***Communications in Nonlinear Science and Numerical Simulation***, Vol. 16, No. 8, (2011), 3385-3396.

4. Wei, H.-L., Billings, S.A., Zhao, Y. and Guo, L., "An adaptive wavelet neural network for spatio-temporal system identification", ***Neural Networks***, Vol. 23, No. 10, (2010), 1286-1299.

5. Yilmaz, S. and Oysal, Y., "Fuzzy wavelet neural network models for prediction and identification of dynamical systems", ***Neural Networks, IEEE Transactions on***, Vol ,۲۱ .No. 10, (2010), 1599-1609.

6. Zainuddin, Z. and Pauline, O., "Modified wavelet neural network in function approximation and its application in prediction of time-series pollution data", ***Applied Soft Computing***, Vol. 11, No. 8, (2011), 4866-4874.

7. Abiyev, R.H. and Kaynak, O., "Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study", ***Industrial Electronics, IEEE Transactions on***, Vol. 55, No. 8, (2008), 3133-3140.

8. Tzeng, S.-T., "Design of fuzzy wavelet neural networks using the ga approach for function approximation and system identification", ***Fuzzy Sets and Systems***, Vol. 161, No. 19, (2010), 2585-2596.

9. Davanipoor, M., Zekri, M. and Sheikholeslam, F., "Fuzzy wavelet neural network with an accelerated hybrid learning algorithm", ***Fuzzy Systems, IEEE Transactions on***, Vol. 20, No. 3, (2012), 463-470.

10. Bodyanskiy, Y. and Vynokurova, O., "Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification", ***Information Sciences***, Vol. 220, No., (2013), 170-179.

11. Cao, J., Lin, Z. and Huang, G.-B., "Composite function wavelet neural networks with differential evolution and extreme learning machine", ***Neural Processing Letters***, Vol. 33, No. 3, (2011), 251-265.

12. Jun, W. and Hong, P., "Constructing fuzzy wavelet network modeling", ***International Journal of Information Technology***, Vol. 11, No. 6, (2005), 68-74.

13. Lee, C.-H. and Teng, C.-C., "Identification and control of dynamic systems using recurrent fuzzy neural networks", ***Fuzzy Systems, IEEE Transactions on***, Vol. 8, No. 4, (2000), 349-366.

14. Zekri, M., Sadri, S. and Sheikholeslam, F., "Adaptive fuzzy wavelet network control design for nonlinear systems", ***Fuzzy Sets and Systems***, Vol. 159, No. 20, (2008), 26.۲۶۹۵-۶۸

15. Zekri, M., Sadri, S. and Sheikholeslam, F., "Approximation of feedback linearization control input based on fuzzy wavelet neural networks", ***Iranian Journal of Science & Technology Transaction B, Engineering,***, Vol. 31, No. 6, (2007), 651–662.

16. Abiyev, R.H., Kaynak, O. and Kayacan, E., "A type-2 fuzzy wavelet neural network for system identification and control", ***Journal of the Franklin Institute***, Vol. 350, No. 7, (2013), 1658-1685.

17. Zhang, Z., "Learning algorithm of wavelet network based on sampling theory", ***Neurocomputing***, Vol. 71, No. 1, (2007), 244-269.

18. Piotrowski, A.P. and Napiorkowski, J.J., "A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling", ***Journal of Hydrology***, Vol. 476, No., (2013), 97-111.

19. Khedri, P.A., Eftekhari, M. and Maazallahi, R., "Comparing evolutionary algorithms on tuning the parameters of fuzzy wavelet neural network", Vol., No.

20. Neri, F. and Cotta, C., "Memetic algorithms and memetic computing optimization: A literature review", ***Swarm and Evolutionary Computation***, Vol. 2, No., (2012), 1-14.

21. Sudholt, D., "The impact of parametrization in memetic evolutionary algorithms", ***Theoretical Computer Science***, Vol. 410, No. 26, (2009), 2511.۲۵۲۸-

22. Jia, D., Zheng, G. and Khurram Khan, M., "An effective memetic differential evolution algorithm based on chaotic local search", ***Information Sciences***, Vol. 181, No. 15, (2011), 3175-3187.

23. Zhang, C., Chen, J. and Xin, B., "Distributed memeticdifferential evolution with the synergy of lamarckian and baldwinian learning", ***Applied Soft Computing***, Vol. 13, No. 5, (2013), 2947-2959.

24. Bao, Y., Hu, Z. and Xiong, T., "A pso and pattern search based memetic algorithm for svms parameters optimization", ***Neurocomputing***, Vol. 117, No., (2013), 98-106.

25. Neri, F. and Tirronen, V., "Recent advances in differential evolution: A survey and experimental analysis", ***Artificial Intelligence Review***, Vol. 33, No. 1-2, (2010), 61-106.

A Differential Evolution and Spatial Distribution based Local Search for Training Fuzzy Wavelet Neural Network

H. A. Bazoobandi, M. Eftekhari

Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

PAPER INFO

Paper history:
Received 27 October 2013
Received in revised form 15March2014
Accepted in 17 April 2014

Keywords:
Fuzzy Wavelet Neural Network (FWNN)
Memetic Algorithm
Differential Evolution
Spatial Distribution Local Search
Adaptive Selection Strategy

چکیده

الگوریتم‌های بسیاری تاکنون برای تنظیم پارامترهای شبکه‌های عصبی فازی موجک معرفی شده‌اند. نبود یک ساختار مناسب، همگرایی به بهینه‌های محلی و سرعت پائین الگوریتم‌های یادگیری را می توان از مهم‌ترین اشکالات مطالعات گذشته در مورد شبکه‌های عصبی فازی موجک دانست. در این مقاله یک الگوریتم ممتیک برای رفع این اشکالات در یادگیری پیشنهاد شده است. در روش پیشنهادی الگوریتم تکامل تفاضلی به عنوان جستجو کننده سراسری استفاده شده است. مهم ترین نوآوری‌های این مقاله عبارتند از: 1) در این مقاله یک الگوریتم جستجوی محلی براساس توزیع فضایی معرفی شده است که در آن گام حرکت به صورت وفقی تنظیم می‌شود. 2) یک روش انتخاب جدید برای انتخاب افراد مناسب از جمعیت برای اعمال جستجوی محلی ارائه شده است. این ویژگی باعث می‌شود تا فرکانس جستجوی محلی به صورت وفقی تنظیم شود و هزینه محاسباتی الگوریتم کاهش یابد. 3) عملگر انتخاب در روش جستجوی تکامل تفاضلی به گونه ای بهبود داده شده است که برای فرزندان نامناسب نیز شانس انتخاب وجود داشته باشد. این استراتژی باعث می‌شود تا فشار انتخاب در مراحل الگوریتم به خوبی کنترل گردد و از افتادن در بهینه‌های محلی پیشگیری شود. الگوریتم ممتیک پیشنهادی با چندین الگوریتم یادگیری دیگر روی توابع محک مقایسه شده است. نتایج عملی بدست آمده بهبود نرخ همگرایی و دقت بالاتر مدل‌های بدست آمده را در مقایسه با سایر روش‌های یادگیری نشان می‌دهد.

doi:*10.5829/idosi.ije.2014.27.08b.04*