

Training Wavelet Neural Networks Using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for System Identification

Navid Razmjoooy* and Mehdi Ramezani

Department of Electrical Engineering, University of Tafresh, Tafresh, Iran

Phone Number: +98-914-4539067

**Corresponding Author's E-mail: navid.razmjoooy@hotmail.com*

Abstract

System identification is mainly the process of improving a mathematical modeling of a physical system using experimental data. In this paper, a new hybrid wavelet neural network is proposed for the system identification purposes. The Gravitational Search Algorithm (GSA) is a new evolutionary algorithm which recently introduced and has a good performance in different optimization problems. The GSA inspired by the law of gravity and mass interactions. The only disadvantage of GSA is that suffers from slow searching speed in the last iterations. In this paper the hybridization of the defined algorithms (GSAPSO) is proposed for constructing and training wavelet neural networks. The difference of the conventional neural network and wavelet neural network is that the activation function of the original WNN is based on wavelet transformation. This algorithm is based on the optimal selection of network weights dynamically during the training process. The suggested method determines the optimal value of the weights and solves the optimization problem of wavelet neural network structure. The problem of finding a good neural model is then discussed through solutions achieved by wavelet neural networks trained by PSO based and GSA based algorithms. Experimental results show that this method can improve the performance of the wavelet based neural network significantly.

Keywords: *Neural Network, Wavelet, Learning neural network, Gravitational search algorithm, Particle swarm optimization, System Identification, Nonlinear System.*

1. Introduction

The system physical representation and identification of linear and nonlinear dynamic systems by using the measured experimental data has become a problem of considerable importance in engineering. System identification, which is based on the method of least square fit to identify system parameters, may be classified into two categories: one in a deterministic manner and the other in a statistical manner. Recently, Artificial Neural Networks as the nonparametric identification method has become a popular technique for the purposes of identification. Neural networks due to its characteristics like: adaptability, massive parallelism, the inherent capability to handle nonlinear systems and robustness have been widely used in complex nonlinear function mapping [1-4]. The defined networks suffer from the lack of strong constructive technique. Since, they require describing the neuron's parameters, as well as choosing network's architecture. Multilayer perceptron networks (MLP) define a large variety of feed forward neural networks [5]. Conventional MLP is a static network with a forward direction of signal flow and wit no feedback loops, and is usually constructed with sigmoid neurons and trained with the back-propagation (BP) algorithm [6]. Due to multilayered structure of these kinds of networks and the greedy nature of the BP algorithm, the training process often stick in an incorrect local minimum of the error surface and converges too slowly. This process makes The MLP construction to be very time consuming since the optimal number of hidden neurons

is not known in advance, and it must be described by trial and error. To rectify the considered problems, in this paper a new algorithm for constructing and training wavelet neural network (WNN) is proposed.

The idea of using wavelet in neural network was proposed by Zhang and Benveniste [7]. Based on wavelet theory, WNN achieves the best function approximation ability. Since the constructing model algorithm unlike to the conventional BP neural network algorithm, can effectively overcome intrinsic defect in the common neural network. WNN have been presented as an effective universal tool for function approximation, which shows high performance in solving the poor convergence and divergence affront in other neural networks, as it can increase the convergence speed. WNNs have a construction similar to the Radial-basis Function (RBF) neural network [8] except that in the WNN, the radial basis functions are replaced by radial wavelets.

As it can be seen, by designing an optimal methodology for WNNs, we can improve the network efficiency more and more. Most of researches in the neural networks optimization are based on BP networks, like: Genetic Algorithm (GA) [9], Back Propagation (BP) [10], Pruning Algorithm [11], Simulated Annealing [12] and Particle Swarm Optimization [13] have shown significant role in this regard. There still is no good works on the WNNs optimization.

In this paper we focused only on weight optimization of ANN, and propose an evolutionary algorithm to system identification purposes; for this case, we employ a new hybrid evolutionary algorithm for optimizing the weights of Multilayer Perceptron (MLP) ANNs called GSAPSO Algorithm. This optimization algorithm is inspired by the law of gravity and mass interactions have shown great performance in both convergence rate and better global optima achievement [14]. In the present work, we propose hybrid GSAPSO for optimizing the initial weights of WNN. For decreasing the random variation of the proposed algorithm, each experiment has been run 15 times and the mean is presented. The method is tested against ANNs.

2. Wavelet Neural Network and Learning Scheme

In terms of wavelet transformation theory, wavelets in the form below is a family of functions generated from one single function $\psi(x)$ by the operation of dilation and translations.

$$\Psi = \{\Psi_i = |a_i|^{-\frac{1}{2}} \psi\left(\frac{x-b_i}{a_i}\right) : a_i, b_i \in R, i \in Z\} \quad (1)$$

$$x = (x_1, \dots, x_N)$$

$$a_i = (a_{i1}, \dots, a_{iN})$$

$$b_i = (b_{i1}, \dots, b_{iN})$$

Where $\psi(x)$, which is localized in both the time space and the frequency space, is called a mother wavelet. The parameters a, b are defined as the scalar and translation parameters, respectively. In the standard form of wavelet neural network, the output of a WNN is given by:

$$f(x) = \sum_{i=1}^M w_i \Psi_i(x) = \sum_{i=1}^M w_i |a_i|^{-\frac{1}{2}} \psi\left(\frac{x-b_i}{a_i}\right) \quad (2)$$

where $\Psi_i(x)$ presents the wavelet activation function of ith unit of the hidden layer and w_i defines the weight connecting the ith unit of the hidden layer to the output layer unit. It is obviously, the localization of the ith units of the hidden layer is characterized by the scalar parameter a_i and the translation parameter b_i . Due to the previous works, the value of these two parameters can either be achieved by the wavelet transformation theory or be by a training algorithm. Note that the presented wavelet neural network comprises a basis function neural network in the sense of that the wavelets consist of the basis function.

Note that the main feature of the presented basis function networks is the localized activation of the hidden layer, so that the connection weights associated with the units can be viewed as locally precise piecewise constant models whose validity for a given input is illustrated by the activation functions. WNNs Compared to the multilayer perceptron neural network, provides sensible profits like: the learning efficiency and the structure transparency. However, the basis function networks disadvantages is also led by it.

Due to the incorrect local approximation, a large number of basis function units have to be used to approximate a given system. A disadvantage of the wavelet neural network is that for higher dimensional problems many hidden layer units are required.

In order to take profit of the local capacity of the wavelet basis functions while not having to have a great deal of hidden layers, an alternative type of wavelet neural network is used. Its output of the kth unit in the output layer is given by:

$$y_k = \sum_{i=1}^M (w_{i0} + w_{i1}x_1 + \dots + w_{iN}x_N) \Psi_i(x) \tag{3}$$

where, a linear model is replaced by the straightforward weight w_i (piecewise constant model):

$$w_{i0} + w_{i1}x_1 + \dots + w_{iN}x_N \tag{4}$$

Due to the activities of the linear models v_i ($i=1,2,\dots,M$) are illustrated by the associated locally active wavelet functions Ψ_i ($i=1,2,\dots,M$), v_i is locally valid. Since, the proposed local linear wavelet neural network can be defined as below:

$$f(x) = \sum_{i=1}^M (w_{i0} + w_{i1}x_1 + \dots + w_{iN}x_N) \Psi_i(x) \tag{5}$$

$$= \sum_{i=1}^M (w_{i0} + w_{i1}x_1 + \dots + w_{iN}x_N) \left| a_i \right|^{-\frac{1}{2}} \psi\left(\frac{x-b_i}{a_i}\right)$$

3. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) algorithm is a population based evolutionary approach which is proposed by Kennedy and Eberhart. PSO is inspired from the social behavior of birds flocking and fish schooling. Formal PSO algorithm starts by owning swarms (as population candidates) to find solution particles (solution). The particles are moved around in the search-space based on definite formulae. The movement of the particles is followed by their own best known position in the search-space as well as the entire swarm's best known position. After finding the proper positions, these will then come to guide the motions of the swarm. The process is iterated and by accomplishing so it is hoped, but not ensured, that a satisfactory solution will finally be figured out. The swarm regulates as the following two equations:

$$v_i^{t+1} = w.v_i^t + c_1r_1(p_i^t - x_i^t) + c_2r_2(g_i^t - x_i^t) \tag{6}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{7}$$

, $i = 1, 2, \dots, n$

where n defines the number of particles, w is the weighted inertia, $C1$ and $C2$ are the positive constants, $r1$ and $r2$ are two random numbers in the interval $[0,1]$, t describes the iteration number, P_i is the best position of the i -th particle and g_i illustrates the best particle among the group members.

Using eq. 6, the particle updates its velocity based on its former velocity and the distances to its current position from its own best historical position and the best positions of its neighbors in every iteration step, and then it moves around a new position specified by (7).

4. The Gravitational Search Algorithm (GSA)

Gravitational Search Algorithm (GSA) has been proposed by Rashedi *et al.* in 2009 [14]. The GSA inspired from the Newtonian laws of gravitation and motion where all objects move as a result of attraction with each other by gravitational forces. Objects with heavier mass have higher attraction and move faster than the objects with relatively smaller mass. Assume each individual position in the searching space can be represented by

$$X_i = (x_i^1, x_i^2, \dots, x_i^n) \quad (8)$$

$, i = 1, 2, \dots, n$

where n defines the number of individuals (called agents) and the position of i -th agents in the d th dimension can be represented as x_i^d . The mass of each agent is computed for updating each i -th agent with reference to the equality of gravitational and inertia mass assumption as below:

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (9)$$

$$m_j(t) = \frac{fit_j(t) - worst(t)}{best(t) - worst(t)} \quad (10)$$

Where:

$$best(t) = \min_{j \in \{1, 2, \dots, n\}} fit_j(t) \quad (11)$$

$$worst(t) = \max_{j \in \{1, 2, \dots, n\}} fit_j(t) \quad (12)$$

The fitness value, $fit_i(t)$, affects the mass value of i -th agents, which corresponds to the position of the particle in the search space.

The optimization process begins by positioning the agents randomly with random velocity values and initialization of gravitational constant. In the next step, the fitness value, $fit_i(t)$, is evaluated for each agent due to the objective function.

Next, the gravitational constant, $G(t)$, is updated according to the effect of decreasing gravity as below:

$$G(t) = G(t_0) \times \left(\frac{t_0}{t} \right)^\beta, \beta < 1 \quad (13)$$

Mass, M , for each agent is calculated by employing Eq. (9) and Eq. (10), and acceleration, α , is achieved by using:

$$\alpha_i^d(t) = F_i^d(t) \quad (14)$$

where the force acting is achieved by the below formula:

$$F_i^d(t) = \sum_{\substack{j=1 \\ j \neq i}}^N rand_j F_{ij}^d(t) \tag{15}$$

$$F_{ij}^d(t) = G(t) \frac{M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \tag{16}$$

where $M_{aj}(t)$ describes the active gravitational mass related to agent j , is a small constant, $R_{ij}(t)$ is the distance between agent i and j and $rand_j$ is a uniform random variable in the interval $[0,1]$.

Then, the velocity and position of i -th agents can be calculated as follows:

$$v_i^d(t+1) = rand_i.v_i^d(t) + \alpha_i^d(t) \tag{17}$$

$$x_i^d(t+1) = x_i^d + v_i^d(t+1) \tag{18}$$

$, i = 1, 2, \dots, n$

The updating process is continued as long as the stopping criterion is not satisfied. Rashedi et al. compared their GSA with some well-known heuristic optimization algorithms like PSO. The results showed that GSA has worthy in the optimization field. However, GSA suffers from slow searching speed in the last iterations. In this paper we used a hybrid of this algorithm with PSO, called PSO-GSA to improve this weakness.

5. The hybrid PSO-GSA algorithm

As mentioned, the basic purpose in hybridization of PSO and GSA is to combine the ability for social thinking (gbest) in PSO with the local search capability of GSA. In order to combine these algorithms, (4.1) is proposed as follows:

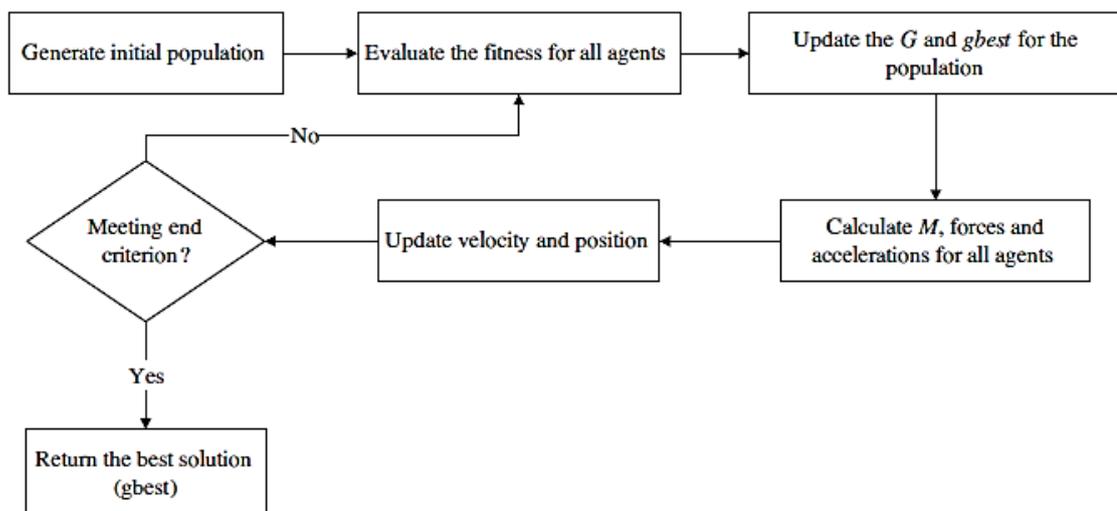


Figure 1: Steps of PSO-GSA [16]

$$v_i^{t+1} = w.v_i^t + c_1' \times rand \times ac_i^t + c_2' \times rand \times (gbest - X_i(t)) \quad (19)$$

where v_i^t defines the velocity of agent i at iteration t ; c_j' is the acceleration coefficient, w describes a weighting function, $rand$ is a random number in the interval $[0, 1]$, ac_i^t is the acceleration of agent i at iteration t , and $gbest$ defines the best solution so far. The positions of agents are updated for all iterations as follows:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (20)$$

$, i = 1, 2, \dots, n$

In PSOGSA, like every other stochastic algorithm, all agents are first randomly initialized. Generated agent is considered as a candidate solution. After initialization, the gravitational force, gravitational constant and resultant forces among agents are calculated using eq. (8) to eq. (16) respectively. The accelerations of particles can also be defined from eq.14.

The best solution so far should be updated each iteration. After achieving the accelerations and updating the best solution so far, the velocities of all agents can be described by eq. (19). Finally, the positions of agents are updated by eq. (20).

The process of velocities and positions updating will be stopped when the criterion purposes is reached. The steps of PSOGSA are represented in Fig. 1.

To illustrate how PSOGSA is efficient, the following remarks are noted [16]:

- In PSOGSA, the fitness quality (solutions) is considered in the updating procedure.
- The agents near good solutions try to absorb the other agents which are exploring different parts of the search space.
- Acceding all agents to a good solution, they move very slowly. In this case, $gbest$ helps them to exploit the global best.
- PSOGSA employs a memory ($gbest$) to save the best solution found so far, so it is accessible at any time.
- Each agent can observe $gbest$ (the best solution) and tend toward it.
- By adjusting c_1' and c_2' , the abilities of global searching and local searching can be balanced.
- The above considered remarks make PSOGSA strong enough to solve a wide range of optimization problems [17].

6. Learning Algorithm

Optimal value selection of weights in neural networks can be formulated as an exploration search problem wherein the architecture of the neural network is reconstructed and fixed during the evolution. The value of the weights can be defined as being train with specific length and the total network is encoded by interpolation of all weights values of the network in the agent. A heuristic concerning the order of the interpolation is to put connection weights to the same node together. The general flow chart of PSOGSA for optimizing a local linear wavelet neural network can be described as follows:

- 1) *Generation of initial position of each agent:* Initial searching points (x_i^0) and velocity (V_i^0) of each agent are usually generated randomly within the allowable range. Note that the dimension of search space is consists the weight parameters used in the local linear wavelet neural network as shown in equation (21). The current searching position is set to $pbest$ for each agent. The best-evaluated value of $pbest$ is set to $gbest$ and the agent number with the bset value is stored.

- 2) *Calculate GSAPSO parameters*: The objective function value is calculated for each agent and the forces and accelerations are achieved for each agent.
- 3) *Evaluation of searching points of each agent*: If the new position (solution) is better than the current *pbest* of the agent, the *pbest* value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and the agent number with the best value is stored.
- 4) *Modification of each searching point*: The current searching point of each agent is updated.
- 5) *Checking the exit condition*: If the critic values reaches the predetermined number, then exit. Otherwise, go to step 2.

In order to express the ANN, consider a two layered network which is formulated as formula (5):

$$\sum_{i=1}^H w_i \sigma \left(\sum_{j=1}^d w_j x_j + b \right) \quad (21)$$

where H defines the number of neurons in the hidden layer, w is the weights of the network, b includes the bias values and σ is the activation function of each neuron which in this case is considered as wavelet function.

The wavelet network is trained by applying optimization the value of the weights for node interconnections and bias terms; until the values output at the output layer neurons are as close as possible to the actual outputs. The mean squared error of the network (MSE) can be represented as below:

$$MSE = \frac{1}{2} \sum_{k=1}^g \sum_{j=1}^m (Y_j(k) - T_j(k))^2 \quad (22)$$

where m describes the number of output nodes, g is the number of training samples, $Y_j(k)$ is the desired output, and $T_j(k)$ is the real output [18].

7. Identification Problem

Neural networks are expedient tool for the systems identification typically encountered in the structural dynamics fields. Neural networks were originally improved simulate the function of the human brain or neural system. This technique really does not solve problems in a strictly mathematical sense, but they can provide an approximation to solve problems. Different neural network techniques have been used in the system identification like: back-propagation network, Hopfield network and Kohonen network and wavelet networks. In the present paper, a new hybrid optimization based on PSO and GSA is introduced. The proposed method is applied on a wavelet neural network technique for adapting it for the systems identification purposes. In this paper, two different problems are introduced for validating the proposed method. The first problem is a linear (static) model of a submarine and the other is a nonlinear (dynamic) model of spring oscillations. The results show that the proposed method has a good performance for the identification of the static and dynamic systems.

8. Numerical Examples

For the following experiments, the used mother wavelet is morlet as below:

$$\Psi = e^{-(x^2/2)} \times \cos(5x) \quad (23)$$

And the utilized objective function is root mean square error:

$$RMSE = \sqrt{\frac{1}{n-1} \sum_{i=1}^n |T_i - Y_i|^2} \tag{24}$$

Where n is the number of data, T_i and Y_i are the network target and the output respectively.

8.1. Submarine linear model

A simple static model of submarine is analyzed in this section. The state space of the system can be described as below:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix} \tag{25}$$

$$B = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}$$

$$C = [1 \ 0]$$

Note that the system is strictly proper, since D state is zero. We generate a set of 127 data points equally spaced as the training dataset by characterizing the step response of the system. The considered system is approximated by the proposed optimized WNN with 1 hidden layer (1 wavelet). The evolved local linear wavelet neural network was obtained at iteration 364 with a mean value error of 6.14×10^{-9} for training data and 0.0133 for validation data set, respectively.

Fig. 2 present training error and error histogram. It is clear that the error value is desirable and we reached the suitable value. Fig.3 illustrates the outputs of actual static system and its approximations by the standard and optimized WNNs. The mean error for the standard WNN is about 0.102. Therefore, we have a 10 times high accuracy by the proposed PSO-GSA WNN technique toward the standard WNN.

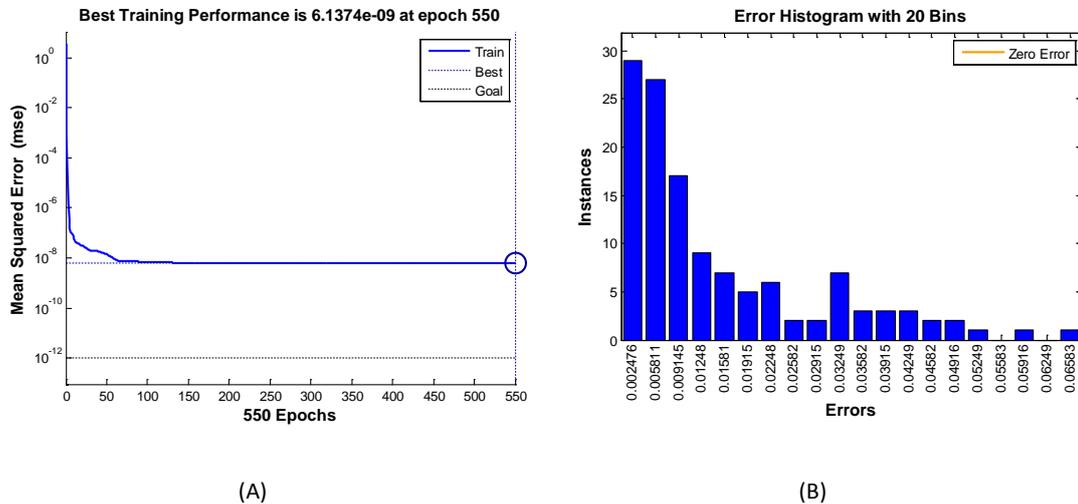


Figure 2: (A) Training Performance and (B) Error histogram for the submarine linear model.

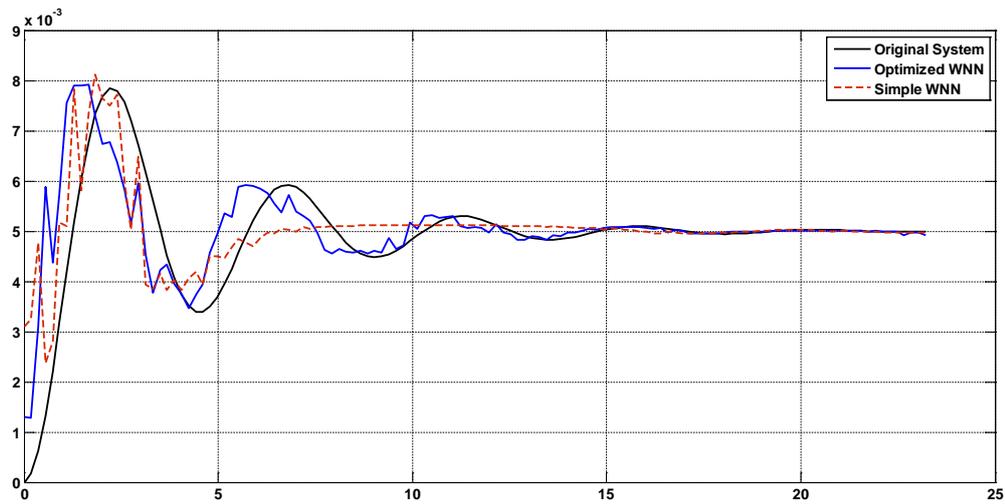


Figure 3: Step response the main system (Solid-Black), Simple WNN (Dashed-Red) and Optimized WNN (Solid-Blue)

8.2. Static Nonlinear Function Approximation

The nonlinear system to be identified is given by the:

$$P = \sin(2\pi \times k \times T_s / 5) + \frac{1}{3} \sin(2\pi \times k \times T_s / 50) \tag{24}$$

$$k = [1, 2, \dots, 500]$$

$$T = \sin(P) + 0.2 \times rand$$

Where rand is a random number with the same length of P and T_s is 0.1.

Standard wavelet neural network trained by PSO-GSA algorithm was convergent at iteration 72 with the root mean square error (RMSE) 0.033276 for training data set and the RMSE for test data is 0.3490 whereas standard WNN has 0.034543 and 0.3703 RMSE for training data set and test data in this system. The used local linear wavelet neural network has 1 hidden unit (1 wavelet) again.

Fig.4 gives the comparison result of the output of the nonlinear system, the output of the standard and Optimized hybrid PSO-GSA wavelet neural network and the identification error. From Fig.4, it is clear that the proposed optimized wavelet neural network works better for identification of nonlinear systems rather than the standard WNN.

Conclusion

In this paper, a new approach for structure optimization of WNN, based on PSO-GSA optimization during the training process is proposed. Two benchmark problems: submarine static model and spring nonlinear model are employed to evaluate the performance of the proposed new learning algorithm. The results are compared with standard WNN. For the analyzed benchmark problems, WNN-PSO-GSA shows better efficiency in terms of convergence rate and avoidance of local minima.

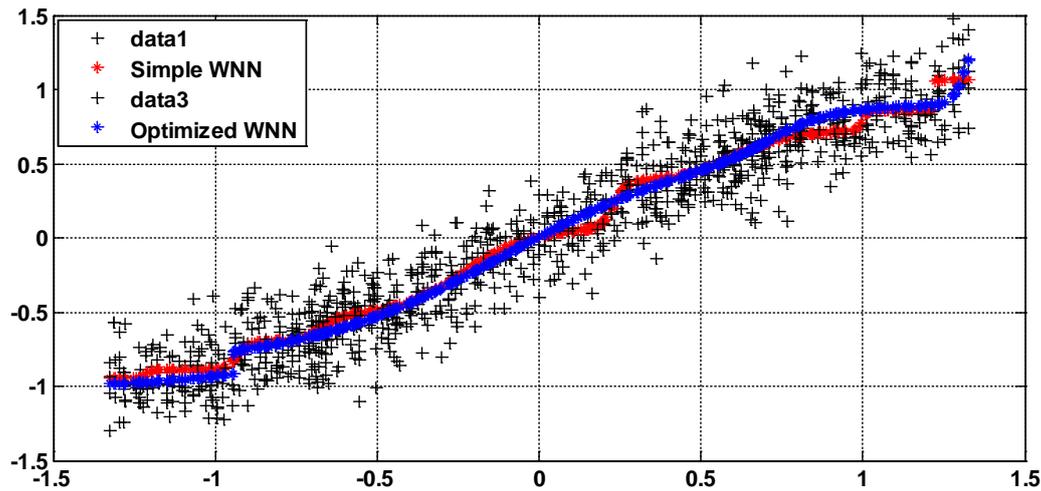


Figure 1: Output for the main system (Solid-Black), Simple WNN (Sashed-Red) and Optimized WNN (Solid-Blue)

References

- [1] Chu, S. R., Shoureshi, R., and Tenorio, M. (1990). "Neural networks for system identification." *IEEE Control Systems Magazine*, Apr., 31-34.
- [2] Chandra, Bala, and K. V. Naresh Babu. "Classification of gene expression data using spiking wavelet radial basis neural network." *Expert systems with applications* 41.4 (2014): 1326-1330.
- [3] C.zhang, H.Shao, Y.Li, "Particle swarm optimization for evolving artificial neural network", 2000 *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Vol. 4, pp.2487 – 2490, Oct. 2000.
- [4] D. Guyer, X. Yang, "Use of genetic artificial neural networks and spectral imaging for defect detection on cherries", *Computers and Electronics in Agriculture*, vol. 29, pp. 179-194, 2000.
- [5] Masri, S.F., Chassiakos, A. G., and Caughey, T. K. (1993). "Identification of nonlinear dynamic systems using neural networks." *J. Appl. Mech.*, 60, 123-133.
- [6] M.C.P.de Souto, A.Yamazaki, T.B.Ludernir, "Optimization of neural network weights and architecture for odor recognition using simulated annealing", *Proc. 2002. Intl. Joint Conf. on Neural Networks*, vol.1, 2002, p.547– 552.
- [7] Mirjalili, SeyedAli, Siti Zaiton Mohd Hashim, and Hossein Moradian Sardroudi. "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm." *Applied Mathematics and Computation* 218.22 (2012): 11125-11137.
- [8] Peláez, J. I., et al. "Ischemia classification via ECG using MLP neural networks." *International Journal of Computational Intelligence Systems* 7.2 (2014): 344-352.
- [9] Pribram, Karl H., ed. *rethinking neural networks: Quantum fields and biological data*. Psychology Press, 2014.
- [10] P.Tang and Z. Xi, "The Research on BP Neural Network Model Based on Guaranteed Convergence Particle Swarm Optimization", *Second Intl. Symp. on Intelligent Information Technology Application, IITA '08*, Vol.2, pp.13 – 16, Dec. 2008.
- [11] Q. Zhang, and A. Benveniste, "Wavelets networks", *IEEE Trans. on Neural Network*, 3, 6, pp.889-898, 1992.
- [12] R.Reed, "Pruning algorithms—a survey", *IEEE Trans. Neural Networks* Vol.4, pp.740–747, 1993.
- [13] Razmjoo, Navid, B. Somayeh Mousavi, and Fazlollah Soleymani. "A hybrid neural network Imperialist Competitive Algorithm for skin color segmentation." *Mathematical and Computer Modelling* 57.3 (2013): 848-856.
- [14] Rashedi, Esmat, and Aliakbar Zarezadeh. "Noise filtering in ultrasound images using Gravitational Search Algorithm." *Intelligent Systems (ICIS), 2014 Iranian Conference on. IEEE*, 2014.
- [15] R. C. Eberhart, Y. Shi, 1999, "Empirical Study of Particle Swarm Optimization", *Proceedings IEEE*, pp. 1945 –1950.
- [16] S. Mirjalili and S.Z. Mohd Hashim, A New Hybrid PSO-GSA Algorithm for Function Optimization, in: *International Conference on Computer and Information Application (ICCIA 2010)*, 2010, pp. 374-377.
- [17] X. Qu1, J. Feng, W. Sun, "Parallel Genetic Algorithm Model Based on AHP and Neural Networks for Enterprise Comprehensive Business", *IEEE Intl. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp.897-900, 2008.

Authors



Navid Razmjooy born in 1988; he received his B.Sc. in electrical engineering from the Islamic Azad University of Ardabil Branch, Ardabil, Iran in 2009 and M.Sc. with honor in Mechatronics Engineering from Islamic Azad University, Majlesi branch, Isfahan, Iran in 2011. He is now pursuing a PhD in the field of Control engineering at the Tafresh University and is the brilliant member of young researchers club in the IA University (Majlesi Industrial Unit). His research interests include: Control; Optimization; Image Processing, Machine Vision, Soft Computing, Data Mining, Evolutionary Algorithms and System Control. He published 56 papers in Farsi and English in peer-reviewed journals and conferences and is now a reviewer in the national and international journals. navid.razmjooy@hotmail.com



Mehdi Ramezani received the PhD degree in Mathematics from Amirkabir university of Technology in Tehran, Iran. He is Currently assistant professor with the department of mathematics, Tafresh university, Iran. ramezani@aut.ac.ir