CSLMEN: A New Optimized Method for Training Levenberg Marquardt Elman Network Based Cuckoo Search Algorithm

Nazri Mohd. Nawi[1], Abdullah Khan[1], M. Z. Rehman[1]

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia (UTHM).
P.O. Box 101, 86400 Parit Raja, Batu Pahat, Johor Darul Takzim, Malaysia
nazri@uthm.edu.my, hi100010@siswa.uthm.edu.my, zrehman862060@gmail.com

Abstract

RNNs have local feedback loops within the network which allows them to shop earlier accessible patterns. This network can be educated with gradient descent back propagation and optimization technique such as second-order methods; conjugate gradient, quasi-Newton, Levenberg-Marquardt have also been used for networks training [14, 15]. But still this algorithm is not definite to find the global minimum of the error function since gradient descent may get stuck in local minima, Nature inspired meta-heuristic algorithms provide derivative-free solution to optimize complex problems. This paper proposed a new meta-heuristic search algorithm, called cuckoo search (CS), based on cuckoo bird's behavior to train Levenberg Marquardt Elman network (LMEN) in achieving fast convergence rate and to avoid local minima problem. The proposed Cuckoo Search Levenberg Marquardt Elman network (CSLMEN) results are compared with artificial bee colony using BP algorithm, and other hybrid variants. Specifically OR and XOR datasets are used. The simulation results show that the computational efficiency of BP training process is highly enhanced when coupled with the proposed hybrid method.

Keywords: Back propagation neural network, cuckoo search algorithm, local minima, artificial bee colony algorithm.

1 Introduction

Artificial Neural Network (ANN) is a unified group of artificial neurons that uses a mathematical or computational form for information processing based on a connectionist move towards calculation. In most cases, ANN is an adaptive construction that changes its formation based on outer or inner information that flows in the network. They can be used to find patterns in data [1]. Among some possible network architectures the ones usually used are the feed forward and the recurrent neural networks (RNN). In a feed forward neural network the signals move only in one direction, starting from the input layer, through the hidden layers to the output layer. While RNNs have local feedback loops inside the network which allows them to store earlier accessible patterns. The ability makes this type of neural networks progressive than the conventional feed forward neural networks in modelling active systems because the network outputs are functions of both the present inputs as well as their inner states [2-3].

Many types of RNNs have been proposed and they can be moderately recurrent or fully recurrent networks. RNNs can carry out especially non-linear elastic mappings and have been used in interesting applications such as associative memories, spatiotemporal pattern classification, manage optimization, forecasting and simplification of pattern sequences [4-8]. Fully recurrent networks are still complex when dealing with complicated problems, therefore partially recurrent networks are used, whose connections are mainly feed forward, but they comprise of carefully selected set of feedback associations. The reappearance allows the system to memorize past history from the precedent without complicating the learning extremely [9]. One example of such a network is an Elman RNN which in rule is set up as a usual feed forward network [10]. However, certain property of RNN makes many of algorithms less efficient, and it often takes an enormous amount of time to train a network of even a reasonable size. In addition, the complex error surface of the RNN network makes many training algorithms more flat to being intent in local minima. Thus the main disadvantage of the RNN is that they require substantially more connections, and more memory in simulation, than standard back propagation network, thus resulting in a substantial increase in the computational time.

The Elman network can be refined with gradient descent back propagation and optimization techniques. The back propagation has several inherent problems. The algorithm is not definite to find the global minimum of the error function since gradient descent may get stuck in local minima, where it may stay indefinitely [11-13]. In recent years, a number of research studies have attempted to conquer these problems and to improve the convergence of the back propagation were proposed. Optimization methods such as second-order methods conjugate gradient, quasi-Newton, Levenberg-Marquardt have also been used for networks training [14, 15]. To overcome these drawback many evolutionary computing technique have been used. Evolutionary computation is often used to train the weights and parameters of the networks. In recent years, many improved learning algorithms have been proposed to overcome the weaknesses of RNN.

In this paper, we propose a new meta-heuristic search algorithm, called Cuckoo Search Levenberg Marquardt Elman Network (CSLMEN). Cuckoo search (CS) is developed by Yang and Deb [16] which imitates animal behavior and is constructive for global optimization [17-19]. The CS algorithm has been applied independently to solve several engineering design optimization problems, such as the design of springs and welded beam structures [20], and forecasting [21]. In this paper, the convergence performance of the proposed Cuckoo Search Levenberg Marquardt Network (CSLMEN) algorithm is analyzed on XOR and OR datasets. The results are compared with artificial bee colony using back-propagation (ABCBP) algorithm, and similar hybrid variants. The main goals are to decrease the computational cost and to accelerate the learning process using a hybridization method.

The remaining paper is organized as follows: Section 2 gives literature review on Levenberg Marquardt algorithm. Section 3 explains Cuckoo Search via levy flight and the proposed CSLMEN algorithm is discussed in Section 4. The simulation results are discussed in section 5. Finally, the paper is concluded in the Section 6.

2 Levenberg Marquardt Algorithm

To speed up convergence, Levenberg Marquardt (LM) is selected as the training algorithm. The LM algorithm is an

approximation of Newton's method to get faster training speed. The advantage of applying LM algorithm over variable learning rate and conjugate gradient method was reported in [22]. The LM algorithm is developed through Newton's method
Assume the error function is:

$$E(t) = \frac{1}{2}\sum_{i=1}^{N} e_i{}^2(t) \tag{1}$$

Where, $e(t)$ is the error; N *is* the number of vector elements, then:

$$\nabla E(t) = J^T(t)e(t) \tag{2}$$

$$\nabla^2 E(t) = J^T(t)J(t) \tag{3}$$

Where, $\nabla E(t)$ is the gradient; $\nabla^2 E(t)$ is the Hessian matrix of E (t) and J (t) is Jacobian matrix;

$$J\,(t) = \begin{bmatrix} \frac{\partial v_1(t)}{\partial t_1} & \frac{\partial v_1(t)}{\partial t_2} & \ldots\ldots & \frac{\partial v_1(t)}{\partial t_n} \\ \frac{\partial v_2(t)}{\partial t_1} & \frac{\partial v_2(t)}{\partial t_2} & \ldots\ldots & \frac{\partial v_2(t)}{\partial t_n} \\ & \cdot & & \\ & \cdot & & \\ & \cdot & & \\ \frac{\partial v_n(t)}{\partial t_1} & \frac{\partial v_n(t)}{\partial t_2} & \ldots\ldots & \frac{\partial v_n(t)}{\partial t_n} \end{bmatrix} \tag{4}$$

For Gauss-Newton Method:

$$\nabla w = -[J^T(t)J(t)]^{-1}J(t)e(t) \tag{5}$$

For the Levenberg-Marquardt algorithm as the variation of Gauss-Newton Method:

$$w(k+1) = w(k) - [J^T(t)J(t) + \mu I]^{-1}J(t)e(t) \tag{6}$$

Where $\mu > 0$ and is a constant; I is identity matrix. So that the algorithm will approach Gauss- Newton, which should provide faster convergence. Note that when parameter λ is large, the above expression approximates gradient descent (with learning rate 1/λ) while for a small λ, the algorithm approximates the Gauss-Newton method.

3 Cuckoo Search Algorithm

Cuckoo Search (CS) algorithm is a novel meta-heuristic technique proposed by Xin-Shen Yang [16, 17]. This algorithm was motivated by the obligate brood parasitism of some cuckoo species in which they lay their eggs in other birds nest. Some host birds are not able to differentiate between their eggs and the cuckoo's. But, if the host bird find an eggs as not their own, they either throw these eggs away or simply abandon its nest and build a new nest elsewhere. The CS algorithm follows the three idealized rules:

1) Each cuckoo lays one egg at a time, and put its egg in randomly chosen nest;
2) The best nests with high quality of eggs will carry over to the next generations;
3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa [0, 1].

The common used equation of the CS algorithm is based on the universal equation of the random-walk algorithm, which is given in Equation 7;

$$X_{k+1\,i} = X_{ki} + \alpha \otimes levy(\lambda) \tag{7}$$

Where k indicates the number of the current generation (k = 1, 2, 3,..., max-cycle and max cycle determined maximum created number). In the CS algorithm, the initial values of the j^{th} attributes of the i^{th} pattern, $p_k = 0; i = [x_k = 0; j, i\,]$, have been determined by using Equation 8;

$$X_k = 0;\, j, i = rand \cdot (up_i - low_i) + low_i \tag{8}$$

Where low_i and up_i are the lower and upper bound limits of j^{th} attributes, respectively. The CS algorithm controls the boundary conditions in each calculated steps. Therefore, when the value of an attribute overflows the allowed search space limits, then the value of the related attribute is updated with the value of the closer limit value to the related attribute. Before starting to iterative search process, the CS algorithm detects the most successful pattern as X_{best} pattern. The iterative evolution phase of the pattern matrix begins with the detection step of the σ_u by using Equation 9;

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta).\sin(\tau.\beta/2)}{\Gamma\left[\left[\frac{(1+\beta)}{2}\right].\beta.2^{\frac{(\beta-1)}{2}}\right]} \right\}^{\frac{1}{\beta}} \quad (9)$$

4 Proposed CSLMEN Algorithm

The proposed framework of the CSLMEN algorithm is given in Figure1. In the Figure 1, each cycle of the search consists of several steps initialization of the best nest or solution, the number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa $\epsilon[0, 1]$. In this algorithm, each best nest or solution represents a possible solution (i.e., the weight space and the corresponding biases for ERNN optimization in this study) to the considered problem and the size of a solution represents the quality of the solution. The initialization of weights was compared with output and the best weight cycle was selected by cuckoo. The cuckoo would continue searching until the last cycle to find the best weights for networks. The solution that was neglected by the cuckoo birds was replaced with a new best nest.

The main idea of this combined algorithm is that CS is used at the begging stage of searching for the optimum. Then, the training process is continued with the LM algorithm. The LM algorithm interpolate between the Newton method and gradient descent method. The LM algorithm is the most widely used optimization algorithm. It outperforms simple gradient conjugate descent and gradient methods in a wide other variety of problems [23]. The flow diagram of CSLMEN is shown in Figure 1. In the first stage CS algorithm finished its training, then, LM Algorithm start training with the weight of CS algorithm and the LM train the network till the stopped condition satisfied.

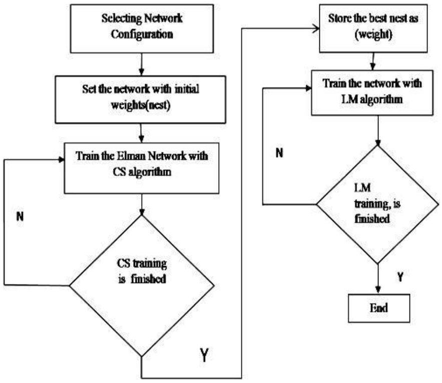

Fig1. Proposed Framework for (CSLMEN)

5 Results and Discussions

The first test problem used is the 2-bit XOR Boolean function consisting of two binary inputs and a single binary output. In the simulations, we used 2-5-1, network for two bit XOR dataset. From the Table 1, we can see that the proposed CSLMEN method performs well on 2-bit XOR dataset. The CSLMEN converges to global minima in 3.011 seconds with 10 epochs, an average accuracy of 100% and achieves 0 MSE. While the other algorithms stay behind and take more CPU times and epochs to converge.

Table 1 CPU time, Epochs and MSE error for 2- bit XOR dataset with 2-5-1 ANN architecture

Algorithm	ABC-BP	ABC_LM	BPNN	CSLMEN
CPU TIME	172.38	123.95	42 .6	3.01
EPOCHS	1000	1000	1000	10
MSE	2.4e-4	0.125	0.2206	0
SD	6.7e-5	1.5e-6	0.010	0
Accuracy (%)	96.39	71.19	54.61	100

The second test problem is the 4-bit OR Boolean function. In the simulations, we used 4-5-1, network for four bit OR dataset. For four bit OR dataset, if all inputs are 0, the output is 0, otherwise the output will be For the 4-5-1 network architecture, it has twenty five connection weights and six biases. Tables 2, confirms the CPU time, number of epochs, the mean square error, and accuracy for the 4 bit OR test problem with five hidden neurons. The proposed CSLMEN converged to a MSE of 0 within 12 epochs. While the ABC_LM algorithm has an MSE of 1.8E-10 the ABC-BP has the MSE of 1.91E-10 with 99.99 and 99.97 % of accuracy, and the simple BPNN still remain behind and need more epochs and CPU time to converge.

Table 2 CPU time, Epochs and MSE error for 4- bit OR dataset with 4-5-1 ANN architecture

Algorithm	ABC-BP	ABC_LM	BPNN	CSLMEN
CPU TIME	162.44	118.72	63.280	3.81
EPOCHS	1000	1000	1000	12
MSE	1.9E-10	1.82E-10	0.0527	0
SD	1.5E-10	2.11E-11	0.0084	0
Accuracy (%)	99.97	99.99572	89.83	100

6 Conclusions

Elman Recurrent Neural Network (ERNN) is one of the most widely used and a popular procedure of the feed-foward neural network training. The Elman network can be refined with gradient descent back propagation and optimization technique. In this a new meta-heuristic search algorithm, called cuckoo search (CS) is proposed to train LMEN to achieve fast convergence rate and to minimize the training error. The performance of the proposed CSLMEN algorithm is compared with the ABC_LM, ABC-BP and BPNN algorithms. The appearance of the proposed model is verified by means of simulation on two datasets such as 2-bit XOR and 4-bit OR. The simulation results show that the proposed CLMEN is far better than the previous methods in terms of MSE, convergence rate and accuracy.

Acknowledgments

The Authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research under the Fundamental Research Grant Scheme (FRGS) Vote No. 1257.

References:

[1] M. Firdaus, S. Mariyam, & A. Razana, “Enhancement of Particle Swarm Optimization in Elman Recurrent Network with bounded Vmax Function,” *3rd AISMS*, 2009.

[2] X. G. Peng, K. Venayagamoorthy, K. A. Corzin, “Combined Training of Recurrent Neural Networks with Particle Swarm Optimization and Back propagation Algorithms for Impedance Identification,” *IEEE SIS*, pp. 9-15, 2007 .

[3] S. Haykin, “Neural Networks, A Comprehensive Foundation”, 2nd Edition, pp. 84-89, 1999.

[4] E. D. Ubeyli, “Recurrent neural networks employing Lyapunov exponents for analysis of Doppler ultrasound signals”, *Expert Systems with Applications*, 34(4), pp. 2538-2544, 2008.

[5] E. D. Ubeyli, “Recurrent neural networks with composite features for detection of electrocardiographic changes in partial epileptic patients”, *Computers in Biology and Medicine*, 38(3), pp. 401-410, 2008.

[6] E. W. Saad *et al.*, “Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks”, *IEEE Transac-*

tions on Neural Networks, 9(6), pp.1456-1470, 1998.

[7] L. Gupta, & M. McAvoy, “Investigating the prediction capabilities of the simple recurrent neural network on real temporal sequences”, *Pattern Recognition*, 33(12), 2000.

[8] L. Gupta *et al.*, “Classification of temporal sequences via prediction using the simple recurrent neural network”, *Pattern Recognition*, 33(10), pp. 1759-1770, 2000.

[9] G. F. Nihal *et al.*, “Recurrent neural networks employing Lyapunov exponents for EEG signals classification”, *Expert Systems with Applications*, 29, pp. 506-514, 2005.

[10] J. L. Elman, “Finding structure in time”, *Cognitive Science*, 14(2), pp. 179-211, 1990.

[11] N. M. Nawi *et al.*, “BPGD-AG: A New Improvement of Back-Propagation Neural Network Learning Algorithms with Adaptive Gain” *JST*, 2 (2), 2011.

[12] W. Ahmed *et al., “*Modified Back Propagation Algorithm for Learning Artificial Neural Networks”, *18th NRSC*, pp. 345-352, 2001.

[13] J. Wen *et al.*, “The Improvements of BP Neural Network Learning Algorithm”, *5th WCCC-ICSP*, pp.1647-1649, 2000.

[14] Y. Tanoto *et al.*, “Levenberg-Marquardt Recurrent Networks for Long- Term Electricity Peak Load Forecasting”, *Telkomnika*, 9(2), pp. 257-266, 2011.

[15] C. Peng *et al.*, “Nonmonotone Levenberg–Marquardt training of recurrent neural architectures for processing symbolic sequences”, *Journal of Neural Computing & Application*, 20, pp. 897-908, 2011.

[16] X. S. Yang & S. Deb, “Cuckoo search via Lévy flights”, *WCNBIC*-09, India, pp. 210-214, 2009.

[17] X. S. Yang, & S. Deb, “Engineering Optimisation by Cuckoo Search”, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1(4), pp. 330-343, 2010.

[18] M. Tuba *et al.*, “Modified cuckoo search algorithm for unconstrained optimization problems”, *ECC ’11*, pp. 263-268, 2011.

[19] M. Tuba, *et al.*, “Performance of a Modified Cuckoo Search Algorithm for Unconstrained Optimization Problems” *Faculty of Computer Science*, 11(2), pp. 62-74, 2012.

[20] X. S. Yang, and S. Deb, “Engineering optimisation by cuckoo search”, *Int. Journal of Mathematical Modelling and Numerical Optimisation*, 1 (4), pp. 330-343, 2010.

[21] K. Chaowanawate, & A. Heednacram, “Implementation of Cuckoo Search in RBF Neural Network for Flood Forecasting”, *4th International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 22-26, 2012.

[22] M.T. Hagan, M. B. Menhaj, “Training Feedforward Networks with the Marquardt Algorithm”, *IEEE Transactions on Neural Networks*, 5(6), pp. 989-993, 1999.

[23] E. Nourani *et al.*, “Forecasting Stock Prices using a hybrid Artificial Bee Colony based Neural Network,” *ICIMTR-2012*, Malacca, Malaysia, pp. 21-22, 2012.