Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 8C (2013) 18 – 24

Procedia Technology

www.elsevier.com/locate/procedia

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

A New Levenberg Marquardt Based Back Propagation Algorithm Trained with Cuckoo Search

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

[a]*Faculty of Computer Science and Information Technology (FSKTM), University Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, Malaysia.*

Abstract

Back propagation training algorithm is widely used techniques in artificial neural network and is also very popular optimization task in finding an optimal weight sets during the training process. However, traditional back propagation algorithms have some drawbacks such as getting stuck in local minimum and slow speed of convergence. This research proposed an improved Levenberg Marquardt (LM) based back propagation (BP) trained with Cuckoo search algorithm for fast and improved convergence speed of the hybrid neural networks learning method. The performance of the proposed algorithm is compared with Artificial Bee Colony (ABC) and the other hybridized procedure of its kind. The simulation outcomes show that the proposed algorithm performed better than other algorithm used in this study in term of convergence speed and rate.

*Keywords:*Artificial Neural Network, back propagation, local minima, Levenberg Marquardt, Cuckoo Search

1. Introduction

Artificial Neural Networks (ANN) is known for its capability in providing main features, such as: flexibility, competency of learning by instances, and capability to generalize and solve problems in pattern classification, optimization, function approximation, pattern matching and associative memories [1, 2]. Due to their powerful capability and functionality, ANN provides an unconventional approach for many engineering problems that are

* Corresponding author. Tel.: +601-07-4533603
E-mail address: nazri@uthm.edu.my

difficult to solve by normal methods, and extensively used in many extents such as control, speech production, signal processing, speech recognition and business [1]. Among numerous neural network models, the Multilayer Feed-Forward Neural Networks (MLFF) have been primarily used due to their well-known universal estimation proficiencies [3]. For MLFF training back-propagation (BP) algorithm and Levenberq-Marquardt (LM) which are based on gradient descent are mostly used [4]. Different techniques have been used in finding an optimal network performance for training ANNs such as Partial Swarm Optimisation (PSO), Differential Evolution (DE), Evolutionary Algorithms (EA), Genetic Algorithms (GA), and Back Propagation (BP) algorithm [5, 7, 8].

Therefore, a variety of ANN models have been proposed. The most frequently used method to train an ANN is based on BP [9-10]. The BP learning has become the standard method and process in adjusting weights and biases for training an ANN in many domains [11]. Unfortunately, the most regularly used Error Back Propagation (EBP) algorithm [12, 13] is neither prevailing nor fast. It is also not easy to find the suitable ANN architectures. Moreover, another limitation of gradient-descent is that it requires a differentiable neuron transfer function. Also, as ANN generate multifaceted error-planes with multiple local minima, the BP fall into local minima instead of a global minimum [14].

In recent years, many improved learning algorithms have been proposed to overcome the flaws of gradient descent based systems. These algorithms include a direct enhancement method using a poly tope algorithm [14], a global search procedure such as evolutionary programming [15], and genetic algorithm (GA) [16]. The standard gradient-descent BP is not path driven, but population driven. However, the improved learning algorithms have explorative search topographies. Therefore, these approaches are expected to evade local minima often by promoting exploration of the search space. The Stuttgart Neural Network Simulator (SNNS) [17], which was developed in the recent past use many different algorithms including Error Back Propagation [13], Quick prop algorithm [18], Resilient Error Back Propagation [19], Back percolation, Delta-bar-Delta, Cascade Correlation [20] etc. All these algorithms are derivatives of steepest gradient search, so ANN training is relatively slow.

For fast and efficient training, second order learning algorithms have to be used. The most effective method is Levenberg Marquardt (LM) algorithm [21], which is a derivative of the Newton method. This is quite multifaceted algorithm since not only the gradient but also the Jacobian matrix is calculated. The LM algorithm was developed only for layer-by-layer ANN topology, which is far from optimal [22]. LM algorithm is ranked as one of the most efficient training algorithms for small and medium sized patterns. LM algorithm is coined as one of the most successful algorithm in increasing the convergence speed of the ANN with MLP architectures [23]. It is a good combination of Newton's method and steepest descent [24]. It Inherits speed from Newton method but it also has the convergence capability of steepest descent method. It suits specially in training neural network in which the performance index is calculated in Mean Squared Error (MSE) [25] but still it is not able to avoid local minimum [26, 27].

In order to overcome these issues this study proposed a new method that combines Cuckoo Search (CS) and Levenberg Marquardt algorithm to train neural network for XOR data sets. The proposed method reduces the error and improved the performance by escaping from local minima. The Cuckoo Search (CS) developed in 2009 by Yang and Deb [28, 29], is a new meta-heuristic algorithm replicating animal behaviour. The optimal solutions obtained by the CS are far better than the finest solutions found by particle swarm optimizers and genetic algorithms [28].

The remaining of the paper is organized as follows; Section 2 describes the Cuckoo Search algorithm. In Section 3 and 4, the implementation of CSLM and the experiments are discussed and finally, the paper is concluded in section 5.

2. The Cuckoo Search (CS) algorithm

Xin-Shen Yang [28] proposed a metaheuristic Cuckoo Search (CS) algorithm based on the forceful parasitic behaviour of some cuckoo species by laying their eggs in the nests of other bird species. Sometimes the host bird cannot differentiate between its own and cuckoo eggs. But, if an egg is discovered by the host bird as not its own, then it either throw these unknown eggs away or simply leave its nest. Some species in cuckoo are very specialized in the impersonating the colour and pattern of the eggs of the host species. This reduces the chances of their eggs being abandoned and thus increases the chances of their survival. The CS algorithm follows the three basic rules:

- Each cuckoo lays one egg at a time, and put its egg in randomly chosen nest;
- The best nests with high quality of eggs will carry over to the next generations;
- 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].

In this case, the host species can either throw the egg away or build a completely new nest somewhere else. The last assumption can be approximated by the fraction *Pa* of the *n* nests that are replaced by new nests (with new random solutions). For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function. In this algorithm, each egg in a nest represents a solution, and a cuckoo egg represents a new solution, the aim is to use the new and potentially better solutions (cuckoos) to replace a not so good solution in the nests. Based on these three rules, the basic steps of the Cuckoo Search (CS) can be summarized as the following pseudo code:

Step 1: Generate initial population of N host nest $i = 1...N$
Step 2: ***While (*** $f\min < MaxGeneration$) or (stop criterion)
Step 3: ***Do*** Get a Cuckoo randomly by Levy flights and evaluate its fitness F_i .
Step 4: Choose randomly a nest j among N .
Step 5: ***If*** $F_i > F_j$ ***Then***
Step 6: Switch j by the new solution, ***End If***
Step 7: A segment (p_a) of worse nest are abandoned and new ones are built.
Step 8: Keep the optimal solutions (or nest with quality solutions).
Step 9: Rank the solutions and find the current best.
Step 10: **end while**

When creating new solutions x^{t+1} for a cuckoo i , a Levy flight is performed

$$x^{t+1} = x_i^{\,t} + \alpha \oplus levy(\lambda) \tag{1}$$

Where $\alpha > 0$; is the step size which should be related to the scales of the problem of interest. The product $\oplus$ means entry wise multiplications. The random walk via Levy flight is more effective in exploring the search space as its step length is much longer in the long run. The Levy flight essentially provides a random walk while the random step length is drawn from a Levy distribution.

$$Levy \sim u = t^{-\lambda}, 1 < \lambda \le 3 \tag{2}$$

This has an infinite variance with an infinite mean. Here the steps essentially construct a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by Levy walk around the best solution obtained so far, this will speed up the local search. However, a substantial fraction of the new solutions should be generated by far field randomization whose locations should be far enough from the current best solution. This will make sure the system will not be trapped in a local minimum.

3. The proposed CSLM algorithm

The proposed method known as Cuckoo Search based Levenberg-Marquardt (CSLM) algorithm is given in Figure-1. Cuckoo Search (CS) is a metaheuristic algorithm that starts with a random initial population. It works with three basic rules i.e. selection of the best source by keeping the best nests or solutions, replacement of host eggs with respect to the quality of the new solutions or cuckoo eggs produced based randomization via Levy flights globally (exploration) and discovering of some cuckoo eggs by the host birds and replacing according to the quality of the local random walks (exploitation) [30]. In the figure, 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, p_a [0, 1].

In this algorithm, each best nest or solution represents a best possible solution (i.e., the weight space and the corresponding biases for NN optimization in this study) to the considered problem and the size of a food source 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 was replaced with a new best nest. The main idea of this combined algorithm is that CS is used at the beginning stage of searching for the optimum. Then, the training process is continued with the LM algorithm. The LM algorithm incorporates the Newton method and gradient descent method. The flow diagram of CSLM is shown in Fig. 1. In the first stage CS algorithm finish its training then LM algorithm starts training with the weights generated by CS algorithm and the LM train the network until the stopped condition is satisfied.

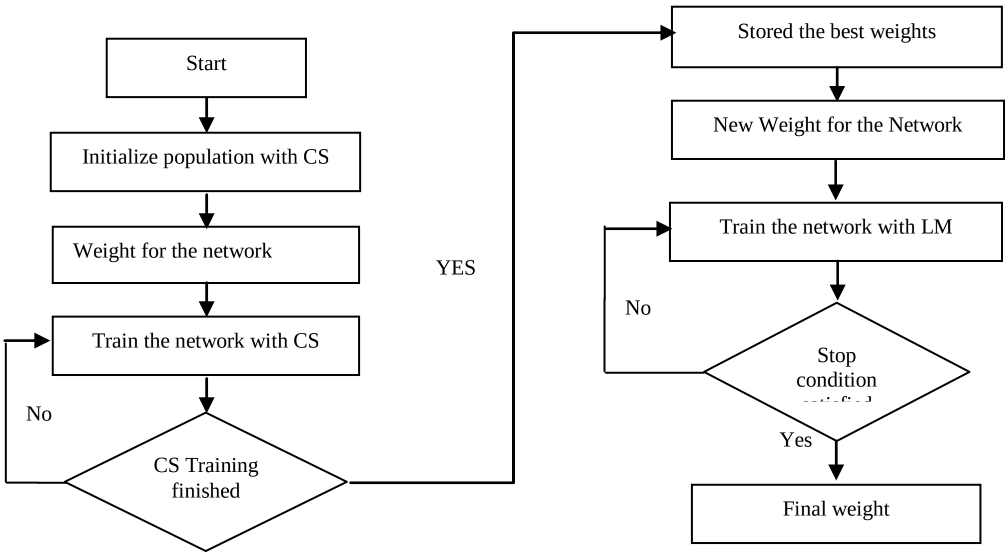


Fig. 1. The proposed CSLM algorithm.

4. Results and discussions

4.1 Preliminaries

In ordered to illustrate the performance of the proposed **CSLM** algorithm, it is trained on 2-bit parity problems. The simulation experiment were performed on a 1.66 GHz AMD E-450 APU with Radeon and 2 GB RAM using MATLAB 2009b software. The proposed CSLM algorithm is compared with artificial bee colony Levenberg Marquardt algorithm (ABC-LM), Artificial Bee Colony Back Propagation (ABC-BP) algorithm and simple back propagation neural network (BPNN) based on MSE and maximum epochs was set to 1000. The three layer feed forward neural network are used for each problem; i.e. input layer, one hidden layer, and output layers. The number of hidden nodes is formed of five and ten neurons. In the network structure the bias nodes are also used and the log sigmoid activation function is placed as the activation function for the hidden and output layers nodes. For each algorithm, 20 trials are repeated.

4.2 Two bit Exclusive-OR problem

The first test problem is the Exclusive-OR (XOR) which consists of two binary inputs and single binary output.

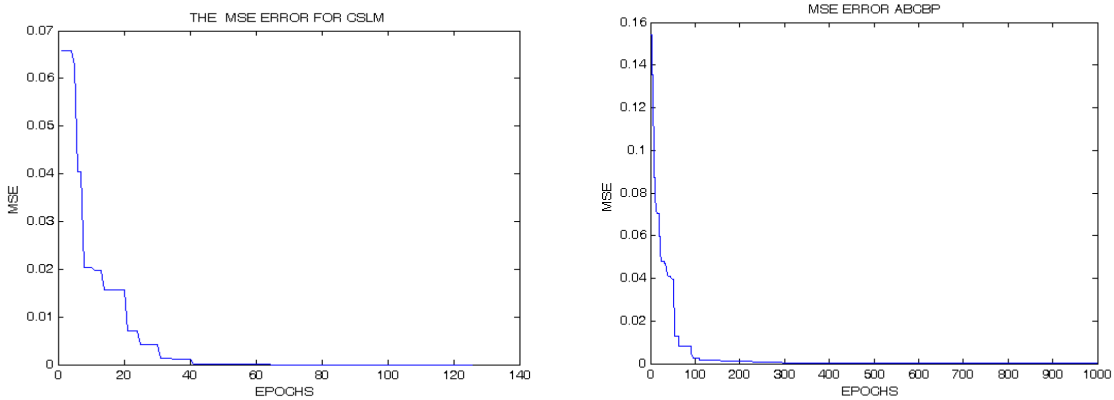
In the simulation we used 2-5-1, 2-10-1 feed forward neural network structure for two bit XOR problem. Table 1 and Table 2 show the CPU time, number of epochs and the MSE for the 2 bit XOR test problem with five and ten hidden neurons. Fig. 2 and Fig. 3 show the MSE of the proposed CSLM algorithm and ABC-BP algorithm for the 2-5-1 network structure.

Table 1. CPU time, Epochs and MSE error for 2-5-1 NN Structure

Algorithm	ABC-BP	ABC-LM	BPNN	CSLM
CPU TIME	12.25	123.94	42.64	15.80
EPOCHS	1000	1000	1000	145
MSE	2.39 x e^{-4}	0.1251	0.2286	0

Table 2. CPU time, Epochs and MSE error for 2-10-1 NN Structure

Algorithm	ABC-BP	ABC-LM	BPNN	CSLM
CPU TIME	197.34	138.90	77.63	18.61
EPOCHS	1000	1000	1000	153
MSE	8.39 x e^{-4}	0.1257	0.1206	0



(a) (b)

Fig. 2. (a) MSE for CSLM of 2-5-1 NN; (b) MSE for ABC-BP of 2-5-1 NN.

5. Conclusion

In this paper, an improved back propagation algorithm known as CSLM is proposed. The proposed CSLM algorithm is based on cuckoo search algorithm, which is simple and generic to train feed-forward artificial neural networks on the 2-bit XOR benchmark problems. The proposed CS algorithm is combined with the LM algorithm where CS algorithm trains the network and LM continues the training by taking the best weight selected in CS algorithm in minimizing the training error. The simulation results show that the proposed CSLM algorithm has better performance than other algorithms such as ABC-LM, ABC-BP and BPNN algorithms.

Acknowledgements

The Authors would like to thank Office of Research, Innovation, Commercialization and Consultancy Office (ORICC), Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) Malaysia for financially supporting this Research under Fundamental Research Grant Scheme (FRGS) vote no. 1236.

References

[1] J. Dayhoff, Neural-Network Architectures: An Introduction.New York: Van Nostrand Reinhold, 1990.
[2] M. K., M. C., and R. S., Elements of Artificial Neural Networks. Cambridge, MA: MIT Press, 1997.
[3] S. Haykin, Neural Networks a Comprehensive Foundation. Prentice Hall, New Jersey, 1999.
[4] Ozturk, C., &Karaboga, D. (2011). Hybrid Artificial Bee Colony algorithm for neural network training.2011 IEEE Congress of Evolutionary Computation (CEC) (pp. 84–88). IEEE. doi:10.1109/CEC.2011.5949602
[5] Leung, C., Member, Chow, W.S.: A Hybrid Global Learning Algorithm Based on Global Search and Least Squares TechniquesNetworks,1997. In: International Conference on, vol. 3, pp. 1890–1895(1997)
[6] Yao, X.: Evolutionary artificial neural networks. International Journal of Neural Systems 4(3), 203–222 (1993)
[7] Mendes, R., Cortez, P., Rocha, M., Neves, J. Particle swarm for feed forward neural network training. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 1895–1899 (2002)
[8] N. M. Nawi, R. G h a z a l i , and M.N.M Salleh: "Predicting Patients with Heart Disease by Using an Improved Back- propagation Algorithm", Journal of Computing. Volume 3, Issue 2, February 2011,pp :53-58
[9] Abid, S., Fnaiech, F. and Najim, M.: A fast feedforward training algorithm using a modified form of the standard back propagation algorithm, IEEE Transactions on NeuralNetworks 12 (2001), 424–430.
[10] Yu, X., Onder Efe, M. and Kaynak, O.: A general back propagation algorithm for feed forward neural networks learning, IEEE Transactions on Neural Networks 13 (2002), 251–259.
[11] Chronopoulos AT, Sarangapani J, (2002) A distributed discrete time neural network architecture forpattern allocation and control, Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02), Florida, USA pp,204-211.
[12] Wilamowski, B.M, Neural Networks and Fuzzy Systems, chapter 32 in Mechatronics Handbook edited by Robert R. Bishop, CRC Press, pp. 33-1 to 32-26, 2002.
[13] Rumelhart, D. E., Hinton, G. E. and Wiliams, R. J, "Learning representations by back-propagating errors", Nature, vol. 323, pp. 533-536, 1986.
[14] Gupta JND, Sexton RS, Comparing back propagation with a genetic algorithm for neural network training, The International Journal of Management Science, 1999;Vol. 27,pp 679-684.
[15] Salchenberger L, et al, Neural Networks: A New Tool for Predicting Thrift Failures. Decision 1992;
[16] Sexton RS, Dorsey RE, Johnson JD, Toward Global Optimization of Neural Networks: A Comparison of the Genetic Algorithm and Backpropagation. Decision Support Systems, 1998;Vol. 22, pp 171–186.
[17] SNNS (Stuttgart Neural Network Simulator) - http://wwwra. informatik.uni tuebingen.de/SNNS/
[18] Scott E. Fahlman. Faster-learning variations on back propagation: An empirical study. In T. J. Sejnowski G. E. Hinton and D. S. Touretzky, editors, 1988 Connectionist Models Summer School, San Mateo, CA,. Morgan Kaufmann 1988.
[19] M. Riedmiller and H Braun. A direct adaptive method for faster back propagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks 1993 (ICNN93), 1993.
[20] S.E Fahlman, .and C. Lebiere, "The cascade- correlation learning architecture" in D. S.Touretzky, Ed. Advances in Neural Information Processing Systems 2, Morgan Kaufmann, San Mateo, Calif, 1990; 524-532.
[21] M.T. Hagan. M.B. Menhaj. "Training feed forward networks with the Marquardt algorithm." IEEE Trans. on Neural Networks; NN-5:989-993. ciences, 1994;Vol.23, pp 899–916.
[22] Bogdan M. Wilamowski, Nicholas Cotton, OkyayKaynak "Neural Network Trainer with Second Order Learning Algorithms" INES 2007 11th International Conference on Intelligent Engineering Systems Budapest, Hungary 2007 IEEE.
[23] M. T. Hagan and M. B. Menhaj, "Training feed forward networks with the Marquardt algorithm," IEEE Trans. Neural Netw., 1994;vol. 5, no. 6, pp. 989–993.
[24] CAO Xiao-ping, HU Chang-hua, ZHENG Zhi-qiang, LV Ying-jie "Fault Prediction for Inertial Device Based on LMBP Neural Network", Electronics Optics & Control, 2005; Vol.12 No.6, pp.38-41.
[25] Simonhaykin, neural networks, China Machine Press, Beijing, 2004.
[26] Qing Xue, and all col "Improved LMBP Algorithm in the Analysis and Application of Simulation Data" 20I0 International Conference on Computer Application and System Modeling (ICCASM 2010).
[27] Jun YAN, Hui CAO, Jun WANG, Yanfang LIU, Haibin ZHAO "Levenberg-Marquardt algorithm applied to forecast the ice conditions in Ningmeng Reach of the Yellow River" Fifth International Conference on Natural Computation. 2009.
[28] Yang XS, Deb S, Cuckoo search via Lévy flights, Proceedings of World Congress on Nature & Biologically Inspired Computing, India, 2009.pp 210-214.
[29] Yang XS, Deb S.. Engineering optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation 2010;1: 330–343.
[30] X.-S. Yang and S. Deb, "Cuckoo search via levy flights," in Nature Biologically Inspired Computing, 2009.NaBIC 2009.World Congress

on, 2009.p. 210 –214.
[31] T. Lee, Back-propagation neural network for the prediction of the short-term storm surge in Taichung harbor, Taiwan, J. Applications of Artificial Intelligence, 2008;21(1):10.