An Accelerated Particle Swarm Optimization Based Levenberg Marquardt Back Propagation Algorithm

Nazri Mohd Nawi[1], Abdullah Khan[1], M.Z. Rehman[1], Maslina Abdul Aziz[2], Tutut Herawan[3], and Jemal H. Abawajy[2]

[1]Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
86400 Parit Raja, Johor, Malaysia
[2]School of Information Technology, Deakin University
Waurn Ponds, Geelong, VIC, Australia
[3]University of Malaya, 50603 Kuala Lumpur, Malaysia
[4]AMCS Research Center, Yogyakarta, Indonesia
nazri@uthm.edu.my, hi100010@siswa.uthm.edu.my,
zrehman862060@gmail.com, tutut@um.edu.my,
{mabdula,jemal.abawajy}@deakin.edu.au

Abstract. The Levenberg Marquardt (LM) algorithm is one of the most effective algorithms in speeding up the convergence rate of the Artificial Neural Networks (ANN) with Multilayer Perceptron (MLP) architectures. However, the LM algorithm suffers the problem of local minimum entrapment. Therefore, we introduce several improvements to the Levenberg Marquardt algorithm by training the ANNs with meta-heuristic nature inspired algorithm. This paper proposes a hybrid technique Accelerated Particle Swarm Optimization using Levenberg Marquardt (APSO_LM) to achieve faster convergence rate and to avoid local minima problem. These techniques are chosen since they provide faster training for solving pattern recognition problems using the numerical optimization technique.The performances of the proposed algorithm is evaluated using some bench mark of classification's datasets. The results are compared with Artificial Bee Colony (ABC) Algorithm using Back Propagation Neural Network (BPNN) algorithm and other hybrid variants. Based on the experimental result, the proposed algorithms APSO_LM successfully demonstrated better performance as compared to other existing algorithms in terms of convergence speed and Mean Squared Error (MSE) by introducing the error and accuracy in network convergence.

Keywords: Artificial Neural Networks, Particle Swarm Optimization, Levenberg Marquardt Back Propagation, Meta-heuristic optimization, Nature inspired algorithms.

1 Introduction

Artificial Neural Networks (ANN) is one of the best approaches in Machine Learning. An ANN is modeled and designed based on the actual human brain concept with

C.K. Loo et al. (Eds.): ICONIP 2014, Part II, LNCS 8835, pp. 245–253, 2014.

interconnected neurons. It simulates the exact way of a processing information. Unlike other conventional techniques, the key element of an ANN is known as supervised learning, in which a set of input/output complex patterns is analyzed and classified [1-7]. As a multilayer perceptron feed-forward network, an ANN is used for random nonlinear function approximation and information processing which other techniques do not have [8]. There are many different types of ANNs depending on their structure and training model, but we will only focus on the most basic one is the Back Propagation Neural Network (BPNN) [9]. A Back-Propagation (BP) algorithm is designed to reduce error between the actual output and the desired output and adjust the ANN weights (and biases) of the network in a gradient descent manner.

However, despite of its reputation, simple architecture and easy to understand learning process, the BPNN has few limitations. The limitations are the risk of getting trapped in a local minima [10-12], possibility of overshooting the minima of the error surface [13-16], slow rate of convergence and so on. Therefore to get faster and more efficient trainings process, second order learning algorithms have to be used.

The Levenberg Marquardt (LM) algorithm is one of the most successful algorithm in speeding up the convergence rate of the ANN with Multilayer Perceptron (MLP) architectures [17]. It is ranked as one of the most efficient training algorithms for small and medium sized patterns. The LM algorithm was developed only for layer-by-layer ANN topology, which is far from optimal [18]. It combines Gauss–Newton Algorithm (GNA) with gradient descent [19]. 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). Unfortunately, along with its benefits, LM algorithm can also suffer the problem of local minimum entrapment [20-23].

Alternatively, there is no one size fits all solution exist. Researchers have been trying to find the optimal solution by proposing different approaches and robust algorithm. Therefore, our focus is finding the best and efficient algorithm(s) of optimizing the neural network using genetic algorithms. In order to overcome the drawback of the Levenberg Marquardt Back Propagation (LMBP), the solution will be converged to the meta-heuristic algorithm. Among the examples of the meta-heuristic nature inspired algorithms are Evolutionary Algorithm (EA) [24], Ant Colony Optimization (ACO) Algorithm [25], Artificial Bee Colony (ABC) Algorithm [26], Genetic Algorithm (GA) [27], Cuckoo Search (CS) Algorithm [6], Bat Algorithm (BA) [28], Particle Swarm Optimization (PSO) Algorithm [29].

Specifically for this paper, The LMBP is combined with APSO, which was originally proposed by Yang in 2008 [30]. In this paper, the convergence behavior and performance of the proposed APSO_LM algorithm is analyzed on selected benchmark classification datasets obtained from UCI machine learning repository. This method is based on the imitation of the social behavior of bird flocking and fish schooling. The LM and scaled conjugate gradient based back-propagation training algorithms are used to train the network. These two training algorithms have been chosen since they provide faster training for solving pattern recognition problems using the numerical optimization technique [13]. Their classification performances with different network architecture are reported in the result section. The results are compared with Artificial Bee Colony (ABC) Algorithm using BPNN algorithm, and other similar hybrid variants. The objective of the optimization is to minimize the computational cost and to accelerate the learning process using a hybridization method.

The outline of this article is as follows. In section 2, the proposed APSO_LM algorithm is explained, and simulation results are discussed in Section 3. Finally, the conclusion of this work is presented in Section 4.

2 The Proposed APSO_LM Algorithm

The APSO is a population based optimization global search algorithm, which has strong ability to find global optimistic result, the LM algorithm has the strong ability to find local optimistic result, but its ability to find the global optimistic result is weak. By combining the APSO with LM, a new algorithm referred to as APSO_LM hybrid algorithm is formulated. Similar to many other meta-heuristic algorithms, APSO starts with a random initial population. The searching process is also started from initialization a group of random particle. First all particle are update according to the Equation (5), and (6), until a new generation set of particle are generated, and then those new particle are used to search the global best position in solution space. Finally the LM algorithm is used to search around the global optimum. In this way the hybrid algorithm may find as optimum more quickly.

In the proposed Accelerated Particle Swarm Optimized Levenberg-Marquardt (APS_LM) algorithm, each best particle or solution represents a 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 population represents the quality of the solution. The initialization of weights is compared with output and the best weight cycle is selected by APSO. The APSO will continue searching until the last cycle to find the best weights for the network. The main idea of this combined algorithm is that APSO algorithm is used at the beginning stage of searching for the optimum to select the best weights. Then, the training process is continued with the LM algorithm using the best particle as weights of APSO algorithm. The LM algorithm interpolate between the Newton method and gradient descent method. The pseudo code for the ASPO-LM algorithm is given as follow.

1. **Initialized** APSO population size, dimensions, and NN structure.
2. Evaluate each initialized particle is fitness value, and x_i is set as the position of the current particle, while g^* is set as the best position of initialized particle.
3. Load training data
4. While (MSE<Stopping criteria)
5. Pass the current best particle as weights to the network.
6. Present all inputs to the network and compute the corresponding network outputs and errors using Equation (1) over all inputs. And compute sum of square of error over all input.

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

7. The sensitivity of one layer is calculated from its previous one and the calculation of the sensitivity start from the last layer of the network and move backward.

8. Compute the Jacobin matrix using Equation (2).

$$J(t) = \begin{bmatrix} \frac{\partial v_1(t)}{\partial t_1} & \frac{\partial v_1(t)}{\partial t_2} & \cdots & \frac{\partial v_1(t)}{\partial t_n} \\ \frac{\partial v_2(t)}{\partial t_1} & \frac{\partial v_2(t)}{\partial t_2} & \cdots & \frac{\partial v_2(t)}{\partial t_n} \\ & \vdots & & \\ \frac{\partial v_n(t)}{\partial t_1} & \frac{\partial v_n(t)}{\partial t_2} & \cdots & \frac{\partial v_n(t)}{\partial t_n} \end{bmatrix} \tag{2}$$

9. Solve Equation (3) to obtain ∇t .

$$\nabla t = -[J^T(t)J(t) + \mu I]^{-1} J(t)e(t) \tag{3}$$

10. Recomputed the sum of squares of errors using Equation (3) if this new sum of squares is smaller than that computed in Step 6, then reduce μ by λ=10, update weight using $w(k+1) = w(k) - \nabla w$ and go back to Step 6. If the sum of squares is not reduced, then increase μ by λ =10 and go back to Step 8.
11. The algorithm is assumed to have converged when the norm of the gradient Equation (4) is less than some prearranged value, or when the sum of squares has been compact to some error goal.

$$\nabla E(t) = J^T(t)e(t) \tag{4}$$

12. Chose the particle with the best fitness value of all the particle as gbest
13. For each particle
14. Calculate particle velocity according Equation (5)

$$v_i^{t+1} = v_i^{t} + \alpha \varepsilon_n + \beta(g^* - x_i^{t}), \tag{5}$$

15. Update particle position according Equation (6)

$$x_i^{t+1} = x_i^{t} + \quad v_i^{t+1}. \tag{6}$$

    End
16. APSO keep on calculating the best possible weight at each epoch until the network is converged.
    End while

3 Results and Discussion

Basically, the main focus of this paper is to compare the performance of different algorithms introducing the error and accuracy in network convergence. Some simulation results, tools and technologies, network topologies, testing methodology and the classification problems used for the entire experimentation will be discussed further in the this section.

3.1 Wisconsin Breast Cancer Classification Problem

This problem tried to diagnosis of Wisconsin breast cancer by trying to classify a tumor as either benign or malignant based from continues clinical variable. This dataset consist of 9 inputs and 2 outputs with 699 instances. The input attribute are, for instance, the clump thickness, the uniformity of cell size, the uniformity of cell shape, the amount of marginal adhesion, the single epithelial cell size, frequency of bare nuclei, bland chromatin, normal nucleoli, and mitoses. The selected network architecture is used for the breast cancer classification problem is consists of 9 inputs nodes, 5 hidden nodes and 2 output nodes.

Table 1. Summary of algorithms performance for breast cancer classification problem

Breast Cancer Benchmark Classification Problem			
Algorithms	**Accuracy**	**MSE**	**SD**
ABC-BP	92.02	0.184	0.459
ABC-LM	93.83	0.0139	0.001
ABCNN	88.96	0.014	0.0002
BPNN	90.71	0.271	0.017
APSO_LM	99.9	2.40E-06	2.80E-06

Table 1, illustrate that the proposed algorithm (APSO_LM), shows superior performance than BPNN, ABC-BP, ABCNN, and ABC-LM. The proposed models such as APSO_LM, have achieve small MSE (2.4E-06) and SD (2.8E-06) with 99.95 percent of accuracy. While the other algorithms such as ABCNN, BPNN, ABC-BP, and ABC-LM fall behind of the proposed algorithms with large MSE (0.014, 0.271, 0.184, and 0.013), and SD (0.0002, 0.017, 0.459, and 0.001) and low accuracy. Similarly, Figure1 shows the performances of MSE convergence for the used algorithms. The proposed APSO_LM algorithm convergences only in 3 epochs. While the other algorithm take more epochs for their convergence. From the simulation results its can easily understand that the proposed algorithms such as APSO_LM shows better performance than the BPNN, ABC-BP, and ABC-LM, algorithms in term of MSE, SD and accuracy.

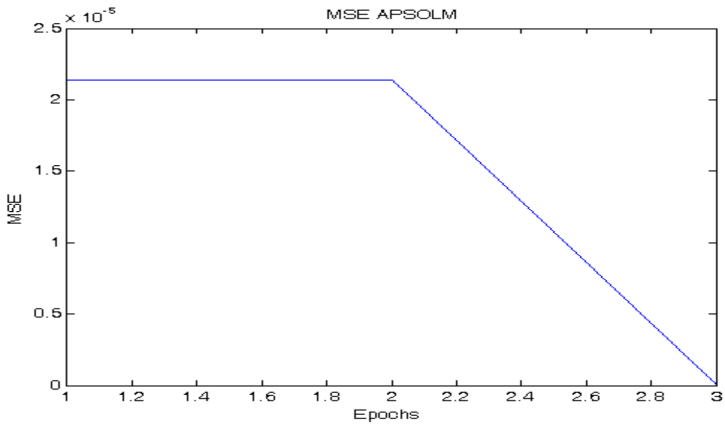

Fig. 1. MSE via Epochs Convergence for breast cancer classification problem

3.2 IRIS Classification Problem

The Iris classification dataset was created by Fisher. Who used it to demonstrate the values of differentiate analysis. This is maybe the best famous database to be found in the pattern recognition literature. There were 150 instances, 4 inputs, and 3 outputs in this dataset. The classification of Iris dataset involving the data of petal width, petal length, sepal length, and sepal width into three classes of species, which consist of Iris Santos, Iris Vermicular, and Iris Virginia. The selected network structure for Iris classification dataset is 4-5-3. Which consist of 4 inputs nodes, 5 hidden nodes and 3 outputs nodes. 75 instances are used for training dataset and the rest as for testing dataset.

Table 2. Summary of algorithms performance for Iris Benchmark Classification Problem

Iris Benchmark Classification Problem			
Algorithms	**Accuracy**	**MSE**	**SD**
ABC-BP	86.87	0.155	0.022
ABC-LM	79.55	0.058	0.0057
ABCNN	80.23	0.048	0.004
BPNN	87.19	0.311	0.022
APSO_LM	99.99	1.21E-05	1.84E-06

Table 2 shows the comparison performances of the proposed algorithm such as APSO_LM, with the BPNN, ABCNN, ABC-BP, ABC-LM algorithms in term of MSE, SD, and accuracy. From the table 2 it's clear that the proposed APSO_LM models have better performances achieved less MSE, SD, and high accuracy than the BPNN, ABCNN, ABC-BP, ABC-LM algorithms. Meanwhile, the Figure 2 illustrates the MSE's convergence performances of the algorithm. Form these figure it's clear that the proposed algorithms show high performances than the other algorithms in term of MSE, Standard deviation (SD), and accuracy.

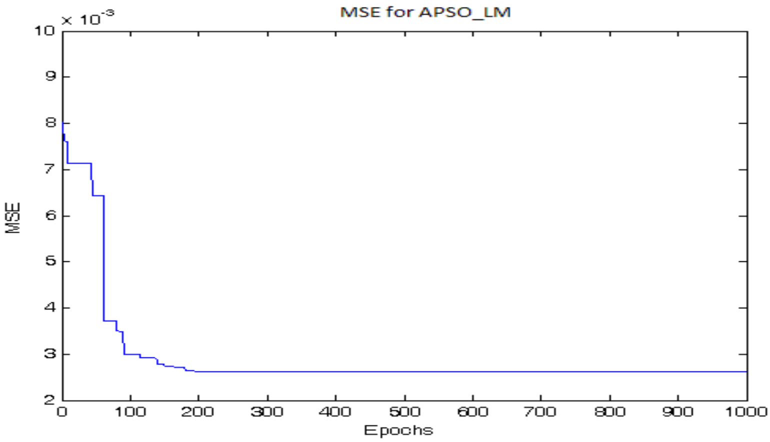

Fig. 2. MSE via Epochs Convergence on Iris Benchmark Classification Problem

4 Conclusion

APSO algorithm is one of the latest addition among the meta-heuristic nature inspired algorithms, which provide derivative-free solutions to solve complex problems. This paper studies the data classification problem using the dynamic behavior of LMBP, trained by nature inspired meta-heuristic APSO algorithm, in-order to achieve fast convergence rate and to avoid local minima problem. The performances of the proposed models APSO_LM is compared with the Artificial Bee Colony using BPNN algorithm, and other hybrid variants. Specifically, 7-Bit Parity, and some selected benchmark classification datasets are used for training and testing the network. The simulation results show that the proposed APSO_LM is far better than the previous methods in terms of convergence rate, and achieved higher accuracy and less MSE on all the designated datasets.

Acknowledgement. This work is supported by University of Malaya High Impact Research Grant no vote UM.C/625/HIR/MOHE/SC/13/2 from Ministry of Higher Education Malaysia.

References

1. Zheng, H., Meng, W., Gong, B.: Neural Network and its Application on Machine fault Diagnosis. In: ICSYSE 1992, pp. 576–579 (1992)
2. Kosko, B.: Neural Network and Fuzzy Systems, 1st edn. Prentice Hall of India (1992)
3. Basheer, I.A., Hajmeer: Artificial Neural Networks: fundamentals, computing, design and application. Journal of Microbiological Methods 43(1), 3–31 (2000)
4. Krasnopolsky, V.M., Chevallier: Some Neural Network applications in environmental sciences. Part II, advancing computational efficiency of environmental numerical models. Neural Networks 16(3), 335–348 (2003)
5. Coppin, B.: Artificial Intelligence Illuminated. Jones and Bartlet illuminated Series, USA, pp. 291–324 (2004)
6. Nawi, N.M., Khan, A., Rehman, M.Z.: A new cuckoo search based levenberg-marquardt (CSLM) algorithm. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013, Part I. LNCS, vol. 7971, pp. 438–451. Springer, Heidelberg (2013)
7. Nawi, N.M., Khan, A., Rehman, M.Z.: A new back-propagation neural network optimized with cuckoo search algorithm. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013, Part I. LNCS, vol. 7971, pp. 413–426. Springer, Heidelberg (2013)
8. Contreras, J., et al.: ARIMA models to predict next-day electricity prices. IEEE Transactions on Power Systems 18, 1014–1020 (2003)
9. Leung, C.T., Chow, T.: A hybrid global learning algorithm based on global search and least squares techniques for back propagation networks. In: International Conference on Neural Networks, pp. 1890–1895 (1997)
10. Zhang, J.-R., et al.: A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation 185, 1026–1037 (2007)

11. Ahmed, W.A.M., Saad, E.S.M., Aziz, E.S.: Modified Back Propagation Algorithm for Learning Artificial Neural Networks. In: Proceedings of the Eighteenth National Radio Science Conference, pp. 345–352 (2001)
12. Nawi, N.M., et al.: Countering the Problem of Oscillations in Bat-BP Gradient Trajectory by Using Momentum. In: Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng 2013), pp. 103–110. Springer Singapore (2014)
13. Jin, W., et al.: The improvements of BP neural network learning algorithm. In: Proceedings of 5th International Conference on Signal Processing WCCC-ICSP 2000, vol. 3, pp. 1647–1649 (2000)
14. Nawi, N.M., Ransing, R.S., Salleh, M.N.M., Ghazali, R., Hamid, N.A.: An Improved Back Propagation Neural Network Algorithm on Classification Problems. In: Zhang, Y., Cuzzocrea, A., Ma, J., Chung, K.-i., Arslan, T., Song, X. (eds.) DTA and BSBT 2010. CCIS, vol. 118, pp. 177–188. Springer, Heidelberg (2010)
15. Mohd Nawi, N., et al.: BPGD-AG: A New Improvement Of Back-Propagation Neural Network Learning Algorithms With Adaptive Gain. Journal of Science and Technology 2 (2011)
16. Nawi, N.M., Ghazali, R., Salleh, M.N.M.: The Development of Improved Back-Propagation Neural Networks Algorithm for Predicting Patients with Heart Disease. In: Zhu, R., Zhang, Y., Liu, B., Liu, C. (eds.) ICICA 2010. LNCS, vol. 6377, pp. 317–324. Springer, Heidelberg (2010)
17. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks 5, 989–993 (1994)
18. Wilamowski, B., et al.: Neural network trainer with second order learning algorithms. In: 11th International Conference on Intelligent Engineering Systems, INES 2007, pp. 127–132 (2007)
19. Lourakis, M.I.: A brief description of the Levenberg-Marquardt algorithm implemented by levmar (2005)
20. Xue, Q., et al.: Improved LMBP algorithm in the analysis and application of simulation data. In: 2010 International Conference on Computer Application and System Modeling (ICCASM), vol. 6, pp. 545–547 (2010)
21. Yan, J., et al.: Levenberg-Marquardt algorithm applied to forecast the ice conditions in Ningmeng Reach of the Yellow River. In: Fifth International Conference on Natural Computation, ICNC 2009, pp. 184–188 (2009)
22. Ozturk, C., Karaboga, D.: Hybrid artificial bee colony algorithm for neural network training. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 84–88 (2011)
23. Nawi, A.K.N.M., Rehman, M.Z.: A New Levenberg-Marquardt based Back-propagation Algorithm trained with Cuckoo Search. In: ICEEI 2013 Procedia Technology 8C, pp. 18–24 (2013)
24. Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1, 1–23 (1993)
25. Blum, C., Socha, K.: Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In: Fifth International Conference on Hybrid Intelligent Systems, HIS 2005, p. 6 (2005)
26. Karaboga, D., Akay, B., Ozturk, C.: Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 318–329. Springer, Heidelberg (2007)
27. Montana, D.J., Davis, L.: Training Feedforward Neural Networks Using Genetic Algorithms. In: IJCAI, pp. 762–767 (1989)

28. Nawi, N.M., Rehman, M.Z., Khan, A.: A New Bat Based Back-Propagation (BAT-BP) Algorithm. In: Swiątek, J., Grzech, A., Swiątek, P., Tomczak, J.M. (eds.) Advances in Systems Science. AISC, vol. 240, pp. 395–404. Springer, Heidelberg (2014)
29. Zhang, J.Z.T.L.J., Lyu, M.: A hybrid particle swarm optimization back propagation algorithm for neural network training. Applied Mathematics and Computation 185, 1026–1037 (2007)
30. Yang, X.-S.: Nature-inspired metaheuristic algorithms. Luniver Press (2010)