Jurnal
Teknologi

Full Paper

An Accelerated Particle Swarm Optimized Back Propagation Algorithm

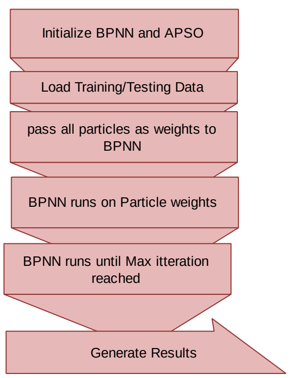
Nazri Mohd. Nawi*, Abdullah Khan, M. Z. Rehman

Soft Computing and Data Mining Centre (SMC), Faculty of Computer Science and Information Technology (FSKTM), Universiti Tun Hussein Onn Malaysia (UTHM), 86400, Parit Raja, Johor, Malaysia

Article history
Received
1 June 2015
Received in revised form
13 August 2015
Accepted
29 September 2015

*Corresponding author
nazri@uthm.edu.my

Graphical abstract



Abstract

Recently, accelerated particle swarm optimization (APSO) derived from particle swarm optimization (PSO) algorithm's principle is becoming a very popular method in solving many hard optimization problems particularly the inherent weight problem in back propagation (BP). Therefore, this paper proposed an accelerated particle swarm optimized back propagation neural network (APSO-BP) algorithm in order to overcome the problems faced in BP algorithm. By using APSO to optimize the weights at each iterations of BP algorithm, the proposed APSO-BP is able to increase the convergence speed and avoids local minima. The simulation results demonstrates that the proposed algorithm outperforms the traditional BP method and achieves the objectives of this research, which contributes to artificial intelligence field.

Keywords: Back propagation, particle swarm optimization, metaheuristics, optimal weight, local minima

Abstrak

Kajian terkini terhadap accelerated particle swarm optimization (APSO) yang dihasilkan menggunakan prinsip particle swarm optimization (PSO) algoritma telah menjadi satu kaedah yang sangat popular dalam menyelesaikan banyak masalah pengoptimuman keras terutamanya masalah penentuan pemnberat yang optimum dalam back propagation (BP). Oleh itu, kertas kerja ini pemncadangkan particle swarm optimized back propagation neural network (APSO-BP) algoritma untuk mengatasi masalah yang dihadapi dalam algoritma BP. Dengan menggunakan APSO untuk mengoptimumkan pemberat pada setiap kitaran BP algoritma, kaedah yang dicadangkan APSO-BP mampu untuk meningkatkan tahap convergence dan mengelakkan minima tempatan. Keputusan simulasi menunjukkan bahawa algoritma yang dicadangkan menghasilkan keputusan yang lebih baik berbanding kaedah BP tradisional disamping mencapai objektif kajian ini, yang sekaligus menyumbang kepada bidang artificial intelligence.

Kata kunci: *Back propagation, particle swarm optimization, metaheuristics, optimal weight, local minima*

1.0 INTRODUCTION

Artificial Neural Networks (ANN) refers to the nonlinear system, which consists of small processing units known as Artificial Neurons that works by processing information like biological neurons in the brain [1-4]. ANN intends to imitate some basic characteristics of human brain, such as self-adaptability, self-organization, high parallelism, robust and fault-tolerant etc., and it is very important for describing the non-linear problem correctly [5]. An Artificial Neuron can be trained to store, recognize, estimate and adapt to new patterns without having the prior information of the function it receives. This ability of learning and adaption has made ANN superior to the conventional methods used in the past. Due to its capability to solve complex time critical problems, it has been widely used in the engineering fields such as biological modelling, financial forecasting, weather forecasting, decision modelling, control systems, manufacturing, health and medicine, ocean and space exploration [6-7]. ANN are usually classified into several categories on the basis of supervised and unsupervised learning methods and feed backward architectures [8].

This paper will only focus on supervised learning methods and the most popular algorithm used in supervised learning method, i.e., BP. The remaining paper is organized as follows: Section 2.0 explains BP and APSO algorithms. Section 3.0 discusses the proposed APSO-BP algorithm to overcome the poor weight approximation in simple BP algorithm and the simulation results are discussed in Section 4.0. Finally the paper is concluded in the Section 5.0.

2.0 BACK PROPAGATION ALGORITHM

Back-Propagation (BP) Neural Network is one of the most novel supervised learning ANN algorithm proposed by Rumelhart, Hinton and Williams in 1986 [9]. BP is a multilayer perceptron (MLP) that consists of 3 layers of neurons connected in parallel, i.e.; an input layer, i hidden layer, j and output layer, k. Each layer in MLP is fully connected to each other and each layer is referred as nodes with nonlinear activation function. The process of back propagation comprised of two stage which are feed forward and back-forward and the process will continue until it reach near the actual output. The processes as proposed in [10] starts with random weight initialization, which are usually set in the range of -0.5 to 0.5. Each iteration involves first feeding data through neural network from inputs to the outputs. When errors occurs, output layer will feeding back to input layers while making changes to the weights of nodes which give the algorithm its name, back propagation.

The algorithm process will repeat itself in this way until the output produced for the training data is close to the desired value [11]. BPNN learns by calculating the errors of the output layer to find the errors in the hidden layers. Due to this ability of back propagating, it is highly suitable for problems in which no relationship is found between the output and inputs. The gradient descent method is utilized to calculate the weights and adjustments are made to the network to minimize the output error. The BPNN algorithm has become the standard algorithm used for training MLP. It is a generalized least mean squared (LMS) algorithm that minimizes a criterion equals to the sum of the squares of the errors between the actual and the desired outputs. The principal of LMS algorithm as discussed by [12] is given as:

$$E_p = \sum_{i=1}^{j} (e_i)^2 \tag{1}$$

P defined as P^{th} pattern and j is the number of output units. Where the nonlinear error signal is:

$$e_i = d_i - y_i \tag{2}$$

d_i and y_i is defined as the desired and the current outputs for i^{th} unit respectively. The gradient descent method is as follow:

$$w_{ki} = -\mu \frac{\partial E_p}{\partial w_{ki}} \tag{3}$$

where w_{ki} is the weight of the i^{th}in the $(n-1)^{th}$ layer to k^{th} unit in the n^{th} layer, μ is defined as the learning rate. BP calculates errors in output layer, ∂_l, and the hidden layer, ∂_j are using the two formulae proposed by [13-14] below:

$$\partial_l = \mu(d_i - y_i) f'(y_i) \tag{4}$$

$$\partial_j = \mu \sum \partial_l w_{lj} f'(y_i) \tag{5}$$

d_i and y_i are the desired output of the i^{th} output neuron and the actual output layer respectively. k is the adjustable variable in the activation function. The weights, w_{lj} and biases, b_l are adjusted using the following formulae:

$$w_{ij}(k+1) = w_{ij}(k) + \mu \partial_j y_i \tag{6}$$

$$w_{lj}(k+1) = w_{lj}(k) + \mu \partial_j x_l \tag{7}$$

$$b_i(k+1) = b_i(k) + \mu \partial_j \tag{8}$$

As beneficial the BP algorithm is, it is bound to have limitations such as local minima and slow convergence rate. In order to overcome the limitation of BP, researcher explore other algorithms and hybridize as well as implements other algorithms with BP such as artificial bee colony back propagation (ABCBP) and particle swarm optimization back propagation (PSOBP). This paper introduced accelerated particle swarm optimization (APSO) to hybridize with BP algorithm.

2.0 PARTICLE SWARM OPTIMIZATION (PSO)

Accelerated Particle Swarm Optimization (APSO) is one of variants introduced in Particle Swarm Optimization (PSO) where in PSO, particles move in the search space area of the problem to find the main solution by changing its velocity accordingly and shares information with each other until they reach optimized solution that have all the information needed to optimize the best solution. The principle of

APSO is derives from PSO's principle that focus in fasten the convergence rate. As mention in [15], this is equivalent to introduce an implicit mass to stabilize the motion of the agents, and thus the algorithm is likely to converge more quickly. As APSO improvised on changing the velocity to obtain the optimized solution, it is suitable to be hybridized with BP by changing the velocity to find the best optimized weight for BP algorithm.

3.1 The Proposed APSO-BP Algorithm

The APSO is a population based optimization algorithm, and like other meta-heuristic algorithms, it starts with a random initial population. In the proposed APSO-BP algorithm, each best particle represents a possible solution (i.e., the weight space and the corresponding biases for BPNN optimization in this study). The weight optimization problem and the size of a population represent the quality of the solution. In the first epoch, the best weights and biases are initialized with APSO and then those weights are passed on to the BPNN. The weights in BPNN are calculated and compared in the reverse cycle. In the next cycle APSO will updated the weights with the best possible solution and APSO will continue searching the best weights until the last cycle epoch of the network is reached or either the MSE is achieved. The error can be calculated as:

$$\mathrm{E} = (\mathrm{T_t} - \mathrm{Y_t}) \tag{9}$$

The performance index for the network is calculated as:

$$\mathrm{V(x)} = \frac{1}{2}\sum_{\mathrm{t=1}}^{\mathrm{R}}(\mathrm{T_t} - \mathrm{X_t})^{\mathrm{T}}(\mathrm{T_t} - \mathrm{Y_t}) \tag{10}$$

$$V_F(x) = \frac{1}{2}\sum_{t=1}^{R} E^T.E \tag{11}$$

The proposed method the MSE as the performance index is calculated as follows:

$$V_\mu(x) = \frac{\sum_{j=1}^{N} V_F(x)}{P_i} \tag{12}$$

where, Y_t is the output of the network when the t^{th} input net_i is presented. And $E = (T_t - Y_t)$ is the error for the t^{th} input, $V_\mu(x)$ is the average performance, $V_F(x)$ is the performance index, and P_i is the number of particle population in i^{th} iteration. At the end of each epoch the list of average MSE of i^{th} iteration can be calculated as:

$$MSE_i = \{V_\mu{}^1(x), V_\mu{}^2(x), V_\mu{}^3(x) \ldots\ldots V_\mu{}^n(x)\} \tag{13}$$

The APSO is replicating the MSE and it is found when all the inputs are processed for each population of the particle. So, the particle x_i is calculated as:

$$x_i = Min\{V_\mu{}^1(x), V_\mu{}^2(x), V_\mu{}^3(x) \ldots\ldots V_\mu{}^n(x)\} \tag{14}$$

And the rest of the MSE is considered as other particle. A new solution $x_i{}^{t+1}$ for particle i is generated according to the following Equation:

$$x_i{}^{t+1} = x_i{}^t + v_i{}^{t+1} \tag{15}$$

So, the movement of the other particle can be drawn from Equation (16):

$$v_i{}^{t+1} = v_i{}^t + \alpha\varepsilon_n + \beta(g^* - x_i{}^t) \tag{16}$$

The particle swarm optimization can move through velocity $v_i{}^{t+1}$ can be written as:

$$\nabla v_i{}^{t+1} = \begin{cases} v_i{}^t + \alpha\varepsilon_n + \beta(g^* - x_i{}^t) \\ \mathrm{x_i} \quad \mathrm{else} \end{cases} \tag{17}$$

where, ∇V_i is a small movement of the particle. The weights and bias for each layer is then adjusted as;

$$W_x{}^{n+1} = W_x{}^n - \nabla v_i{}^{t+1} \tag{18}$$

$$B_x{}^{n+1} = B_x{}^n - \nabla v_i{}^{t+1} \tag{19}$$

The pseudo code of the proposed APSO-BP algorithm is given as;

Step 1: *Initialize APSO and BPNN*
Step 2: *Load the training data*
While *MSE<stopping criteria*
   Step 3: *Initialize all Particle*
   Step 4: *Pass the particle as weights to network*
   Step 5: *Feed forward neural network runs using the weights initialized with APSO*
   Step 6: *Calculate the error backward*
   Step 7: *APSO keeps on calculating the best possible weight at the start of each epoch until the network is converged*
End While

4.0 RESULTS AND DISCUSSION

To test the performance of the proposed APSO-BP algorithm, some benchmark classification datasets such as Breast Cancer, Australian Credit Card, Iris, and Pima Indian Diabetes were used. The simulations were performed on a 2.00 GHz Intel processor with 2GB of RAM. MATLAB 8 was used to perform the simulations. The proposed APSO-BP algorithm is compared with traditional back propagation neural network algorithm (BPNN), artificial bee colony neural network algorithm (ABCNN), artificial bee colony back propagation algorithm (ABC-BP) and artificial bee colony Levenberg Marquardt algorithm (ABC-LM) based on mean squared error (MSE), accuracy and epochs. The three layer feed forward neural network are used for each problem. In the network structure, the log sigmoid activation function is used as the activation function for the hidden and output layers nodes. For each problem, trial is limited to 1000 epochs. The MSE is set to 0.0001. A total of 30 trials are run for each case. The network results are stored in the result file for each trial.

4.1 Breast Cancer dataset

The first test problem is the breast cancer dataset was obtained from the University of Wisconsin, Madison by Dr. William H. Wolberg [16]. The dataset consists of 9 inputs and 2 outputs with 699 instances. From Table 1, we can see that the proposed APSO-BP algorithm performs well on breast cancer dataset. The APSO-BP achieves an average MSE of 0.002 within an average accuracy of 99.78%. While other algorithms fall behind in-terms of MSE and accuracy.

Table 1 Epochs, MSE, SD and Accuracy for Breast Cancer dataset

	BPNN	ABCNN	ABC-BP	ABC-LM	APSO-BP
Epochs	1000	1000	1000	1000	1000
MSE	0.271	0.132	0.174	0.056	0.003
SD	0.154	0.022	0.043	0.006	7.9E-20
Accuracy	88.89	76.79	89.99	77.78	99.75

4.2 Australian Credit Card Approval dataset

The second test problem is the Australian credit card approval dataset that predicted the approval or non-approval of a credit card to a customer [17]. Descriptions of each attribute name and values were not enclosed for confidentiality. The dataset consists of 51 inputs and 2 outputs with 690 instances. Table 2 show the proposed APSO-BP succeeds in achieving much lower average MSE error of 0.003 with higher average accuracy of 99.75% than the BPNN, ABCNN ,ABC-BP, ABC-LM algorithms which have MSE of (0.271, 0.132, 0.174, 0.056), and SD of (0.154, 0.022, 0.043, 0.006). From the simulations, it can be seen that the proposed APSO-BP has the capability to converge to global minima with better performance than the other compared algorithms.

Table 2 Epochs, MSE, SD and Accuracy for Australian Credit Card Approval dataset

	BPNN	ABCNN	ABC-BP	ABC-LM	APSO-BP
Epochs	1000	1000	1000	1000	1000
MSE	0.271	0.015	0.184	0.014	0.002
SD	0.017	0.0002	0.046	0.001	0.0004
Accuracy	90.71	88.97	92.02	93.83	99.78

4.3 Iris dataset

The third test problem is the Iris dataset, a classical classification dataset made famous by Fisher, who used it to illustrate principles of discriminate analysis [18]. There are 150 instances, 4 inputs, and 2 outputs in this dataset. From the result shown in Table 3, we can conclude that APSO-BP is performing well, and achieve 99.80% of accuracy with MSE of 0.002 and SD of 0.0004 respectively. For the Iris classification benchmark problem as shown in the Table 4, the proposed algorithm has 12.60% better performance than the other compared algorithm.

Table 3 Epochs, MSE, SD and Accuracy for Iris Dataset

	BPNN	ABCNN	ABC-BP	ABC-LM	APSO-BP
Epochs	1000	1000	1000	1000	1000
MSE	0.321	0.049	0.155	0.058	0.002
SD	0.022	0.049	0.023	0.006	0.0004
Accuracy	87.20	80.24	86.88	79.56	99.80

4.4 Pima Indian Diabetes dataset

The next identified test problem is the Pima Indian Diabetes dataset that was selected from a larger data set held by the National Institutes of Diabetes and Digestive and Kidney Diseases. The constraint of this dataset are all the patients are Prima-Indian women, at least 21 years old and must be living near Phoenix, Arizona, USA [19].This dataset consists of 8 inputs and 2 outputs with 768 instances. From Table 4, the average MSE recorded is 0.0012, and SD of 4.4E-19 with average accuracy of 99.80% which is much higher than the compared algorithms, such as BPNN, ABCNN, ABC-BP, ABC-LM algorithms. From the simulation results as shown in the Table 4, it is clear that the proposed APSO-BP algorithm has 5.95 percent better performance than the other algorithms in terms of accuracy.

Table 4 Epochs, MSE, SD and Accuracy for Diabetes dataset

	BPNN	ABCNN	ABC-BP	ABC-LM	APSO-BP
Epochs	1000	1000	1000	1000	1000
MSE	0.270	0.132	0.201	0.161	0.0012
SD	0.027	0.022	0.002	0.883	4.4E-19
Accuracy	86.96	68.09	88.16	60.95	99.80

5.0 CONCLUSIONS

Conventional back propagation (BP) algorithm has some inherent problems such as getting stuck in local minima and slow speed of convergence due to poor random weight selection. Meta-heuristic algorithms provide derivative-free solution to optimize complex problems. A new meta-heuristic search algorithm called accelerated particle swarm optimization (APSO) is proposed to train the weights in BP to achieve fast convergence rate and to minimize the training error as well as to avoid the local minima. The performance of the proposed APSO-BP algorithm are compared with BPNN, ABCNN, ABC-BP and ABC-LM by means of simulation on four datasets such as breast cancer, Australian credit card, Iris, and Pima Indian Diabetes. The simulation results show that the proposed APSO-BP is far better than the previous methods in terms of mean squared error (MSE) and accuracy.

Acknowledgement

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. This research is also supported by Gates IT Solution Sdn. Bhd under its publication scheme.

References

[1] Nawi, N. M., Khan, A., and Rehman, M. Z. 2013. A New Levenberg Marquardt based Back Propagation Algorithm Trained with Cuckoo Search. *Procedia Technology*. 11. 18-23.

[2] Haikun WEI. 2005. *Theory and Method of Neural Network Structure*. National Defence Industry Press.

[3] Nawi, N. M., Rehman, M. Z., and Khan, A. 2014. Hybrid Bat-BP: A New Intelligent Tool for Diagnosing Noise-Induced Hearing Loss (NIHL) in Malaysian Industrial Workers. *Applied Mechanics and Materials*. 465. 652-656.

[4] Nawi, N. M., Khan, A., and Rehman, M. Z. 2013. *A New Back-Propagation Neural Network Optimized With Cuckoo Search Algorithm. Computational Science and Its Applications.* Springer Berlin Heidelberg. 7971. 413-426.

[5] Zhang, J., MI, X. 1996. *Neural Network and its Engineering Application*. Mechanism Press.

[6] Kosko, B. 1994. *Neural Network and Fuzzy Systems*. 1st edition. Prentice Hall of India.

[7] Lee, T. 2008. Back-propagation Neural Network For The Prediction Of The Short-Term Storm Surge In Taichung Harbor, Taiwan. *Engineering Applications of Artificial Intelligence*. 21(1). 63-72.

[8] Deng, W. J., Chen W. C., and Pei, W. 2008. Back-Propagation Neural Network Based Importance-Performance Analysis For Determining Critical Service Attributes. *Expert Systems with Applications*. 34(2). 1115-1125.

[9] Rumelhart, D. E., Hinton, G. E., Williams, R. J. 1986. Learning Internal Representations by error Propagation. *Nature*. 323. 533-536.

[10] Lowery, A. J., Miller, N., Devaney, A., McNeill, R. E., Davoren, P., Lemetre, C., Benes, V., Schmidt, S., Blake, J., Ball, G., and Kerin, M. J. 2009. MicroRNA Signatures Predict Oestrogen Receptor, Progesterone Receptor and HER2/neuReceptor Status In Breast Cancer. *Breast Cancer Resource*. 11(3). R27.

[11] Coppin, B. 2004. *Artificial Intelligence Illuminated*. 1st edition. Sudbury, Massachusetts: John And Bartlett Publishers, Inc.

[12] Nawi, N. M., Khan, A. and Rehman, M. Z. 2003. A New Back-Propagation Neural Network Optimized with Cuckoo Search. *ICCSA 2013*. Ho Chi Minh City, Vietnam. 413-426.

[13] Cichocki, A., Unbehauen, R. 1993. *Neural Network for Optimization and Signal Processing*. Wiley, Chichester.

[14] Lippman, R. P. 1987. An Introduction To Computing With Neural Networks. *ARIEL 209*. 115-245.

[15] Fergany, 2013. Accelerated Particle Swarm Optimization-based Approach to the Optimal Design of Substation Grounding Grid. *Przegląd Elektrotechniczny*. 89(7). 30-34.

[16] Wolberg, W. H., and Mangasarian, O. L. 1990. Multisurface Method Of Pattern Separation For Medical Diagnosis Applied To Breast Cytology. *Proceedings of the National Academy of Sciences*. 87. 9193-9196.

[17] Quinlan, J. R. 1987. Simplifying Decision Trees. *Int. J Man-Machine Studies*. 27. 221-234.

[18] Fisher, R. A. 1936. The Use Of Multiple Measurements In Taxonomic Problems. *Annals of Eugenics*. 7(II). 179-188.

[19] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R.S. 1988. Using the ADAP Learning Algorithm To Forecast The Onset Of Diabetes Mellitus. *Proceedings of the Symposium on Computer Applications and Medical Care*. IEEE Computer Society Press. 261-265.