Verifying the Accuracy of GDAM algorithm on Multiple Classification Problems

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

[1]Software and Multimedia Centre, 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, zrehman862060@gmail.com, hi100010@siswa.uthm.edu.my

Abstract

Back-Propagation Neural Network (BPNN) algorithm is a widely used technique implemented in many engineering disciplines. Despite solving several practical problems around the globe, BPNN still faces problems like slow convergence, network stagnancy and convergence to local minima. Many alternative ways of improving the convergence rate in BPNN are suggested by previous researchers such as the careful selection of input weights and biases, learning rate, momentum, network topology, activation function and value for 'gain' in the activation function. This research propose an algorithm for improving the working performance of back-propagation algorithm which is 'Gradient Descent with Adaptive Momentum (GDAM)' by keeping the gain value fixed during all network trials. The performance of GDAM is compared with 'Gradient Descent with fixed Momentum (GDM)' and 'Gradient Descent Method with Adaptive Gain (GDM-AG)'. The results show that GDAM is a better approach than previous methods and shows better accuracy on selected classification problems like Wine Quality, Mushroom, Thyroid disease Breast Cancer, IRIS, Australian Credit Card Approval, Pima Indian Diabetes, and Heart Disease.

Keywords: neural networks, gradient descent, adaptive momentum, adaptive gain.

1. Introduction

Artificial Neural Networks (ANNs) are systematic techniques sculpted on the neurological process of the brain. 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 suitable for solving complex time critical problems such as; biological modeling, NIHL prediction, decision modeling, control systems, manufacturing, ocean and space exploration etc. [1-3, 11].

Back-Propagation Neural Network (BPNN) is one of the most novel supervised-learning Artificial Neural Network (ANN) [4]. The BPNN learns by calculating the errors of the output layer to find the errors in the hidden layers. This qualitative ability makes it highly suitable to be applied on problems in which no relationship is found between the output and the inputs. Due to this prowess, it has been successfully implemented in wide range of applications [5]. Despite providing successful solutions BPNN has some limitations. Since, it requires careful selection of parameters such as network topology, initial weights and biases, learn-

ing rate, activation function, and value for the gain in the activation function. An improper use of these parameters can lead to slow network convergence or even stagnancy. Previous researchers have suggested the use of learning rate and momentum to stop network failure and to speed-up the convergence to global minima. These two parameters are frequently used in the control of weight adjustments along the steepest descent and for controlling oscillations [6].

2. BPNN with Momentum Coefficient

Momentum-coefficient (α) is based on the observation that convergence might be improved if the oscillation in the trajectory is smoothed out, by adding a fraction of the previous weight change [4,7]. So the addition of momentum-coefficient helps to smooth-out the descent path by preventing extreme changes in the gradient due to local anomalies [8]. In this case, it is essential to suppress any oscillations that results from the changes in the error surface [10].

In earlier studies, static momentum-coefficient was found to be beneficial for the convergence to global minima but in later studies it was revealed that Back-propagation with Fixed Momentum (BPFM) shows acceleration results when the current downhill of the error function and the last change in weights are in similar directions, when the current gradient is in an opposing direction to the previous update, BPFM will cause the weight direction to be updated in the upward direction instead of down the slope as desired, so in that case it is necessary that the momentum-coefficient should be adjusted adaptively instead of keeping it static [9,10].

In 1994, Simple Adaptive Momentum (SAM) [11] was proposed to enhance the convergence capability of BPNN to global minima. Although SAM was found as a better alternative to Conjugate Gradient Descent and conventional BPNN but its success and failure rate was found to be same as conventional BPNN. In 2002, C. Yu and B. Liu [17] introduced a more efficient Back Propagation and Acceleration Learning (BPALM) method, to answer the convergence failure problem in a much better way by adding some momentum to the adjustment expression. In 2009, R. J. Mitchell suggested adjusting the momentum-coefficient in SAM [11] by considering all the weights in the Multi-layer Perceptrons (MLP). This technique of global adjustment of weights was found much better than the previously used SAM [11] and helps improve the convergence rate to global minima [12].

In 2007, Nazri *et al*. [13] proposed that by varying the gain value adaptively for each node can radically progress the training time of the network. Based on Nazri *et al.* [13] research, this paper proposes a further improvement on the algorithm that will use adaptive momentum and will keep the gain value fixed for all trials.

3. Gradient Descent Adaptive Momentum (GDAM) Algorithm

For each epoch,
For each input vector,
Step-1:
Calculate the weights and biases using the previous momentum value
Step-2:
Use the weights and biases to calculate new momentum value.
End input vector
IF Gradient is increasing, increase momentum
ELSE decrease momentum
End IF
Repeat the above steps until the network reaches the desired value.
End epoch

The proposed GDAM algorithm adaptively changes the momentum while it keeps the gain and learning rate fixed for each training node. Mean Square Error (MSE) is calculated after each epoch and

compared with the target error. The training continues until the target error is achieved.

3.1. Derivation of GDAM Algorithm

The gradient descent method is utilized to calculate the weights and adjustments are made to the network to minimize the output error. The output error function at the output neuron is defined as;

$$E = \frac{1}{2}\sum_{k=1}^{n}(t_k - o_k)^2 \qquad (1)$$

Here, Log-sigmoid activation function is used to find the output on the j^{th} node;

$$O_j = \frac{1}{1+e^{-a_{net_j}}} \qquad (2)$$

where activation function is calculated as;

$$a_{net,j} = \left[\sum_{i=1}^{l} w_{ij} O_i\right] + \theta_j \qquad (3)$$

The weight and bias update expressions for the links connecting to the output nodes are;

$$\Delta w_{jk} = (t_k - O_k)O_k(1-O_k)\alpha_k O_j \qquad (4)$$

$$\Delta\theta_k = (t_k - O_k)O_k(1-O_k)\alpha_k \qquad (5)$$

For adjusting the α adaptively, gradient path is selected. When the gradient (g_s), is increasing, increase the α else decrease the α. So, all the momentum adjustment is done with respect to an increase or a decrease in g_s. The momentum coefficient on each training at epoch (s + 1) is calculated in Equation 6.

$$\alpha_{s+1} = \begin{cases} \alpha_{s-p}, if & g_s < o \\ \alpha_{s+p}, if & g_s > o \end{cases} \qquad (6)$$

where;

$$0.2 < p < 0.9$$

Here in the Equation (6), '*p*' represents the highest and the lowest value in the sigmoid interval [18]. After calculating the α in each epoch, the weight update expression for the input node links becomes:

$$\Delta w_{ij} = \left[\sum_k \alpha_k w_{jk}(t_k - O_k)O_k(1-O_k)\right]\alpha_j O_j(1-O_j)O_i \qquad (7)$$

The bias update expression for hidden nodes will be like this;

$$\Delta\theta_j = \left[\sum_k \alpha_k w_{jk}(t_k - O_k)O_k(1-O_k)\right]\alpha_j O_j(1-O_j) \qquad (8)$$

After calculating biases, hidden and output layer weight, the net weight becomes;

$$w_{net} = \Delta w_{ij} + \Delta w_{jk} + \Delta\theta_j + \Delta\theta_k \qquad (9)$$

Finally, the network weights are calculated after being updated with momentum coefficient, the net weight becomes;

$$w_{net+1} = w_{net} + w_{net_{l-1}} \quad l = 0,1,2,... \qquad (10)$$

4. Results and Discussions

For performing simulations, eight classification problems from UCI machine learning repository are selected to verify the accuracy of the GDAM algorithm. Gradient Descent with Momentum (GDM), Gradient Descent Method with Adaptive Gain (GDM-AG), and the proposed GDAM algorithms are analyzed and simulated on the problems using the MATLAB 2010 software. Three layer back-propagation neural networks is used for training and testing purposed, hidden layer is kept fixed to 5-nodes while output and input layers nodes vary. Global learning rate of 0.4 is selected for the entire tests and gain is kept fixed to 0.3 and target error is set to 0.01[19]. While log-sigmoid activation function is used, the momentum term is varied adaptively between the range of [0, 1]. For each problem, a total of 30 trials are run for each

momentum value and one trial is limited to 3000 epochs.

4.1. Wine Quality

Red wine sub-dataset contains 1599 instances [14]. The best momentum values for GDM and GDM-AG is 0.6 and 0.2 respectively. While for GDAM, the best performance is found in the momentum interval of $[0.6-0.8]$. The highlighted area in Table 1, demonstrate that the superior performance of the proposed GDAM algorithm.

4.2. Mushroom

In mushroom dataset, each mushroom is identified as definitely edible, definitely poisonous, or of unknown edibility [15]. For mushroom, the best momentum value for GDM and GDM-AG is 0.7 and 0.3 respectively. While for GDAM, the best performance is achieved in the momentum interval of $[0.5-0.8]$.From the Table 1, it is easily seen that GDAM algorithm is better in performing convergence with a percentile accuracy of 96.34 than GDM and GDM-AG.

4.3. Thyroid

For thyroid disease, GDM and GDM-AG have the best momentum values at 0.8 and 0.2 respectively [16]. GDAM works best on the momentum interval of $[0.5-0.9]$. It is evident from the Table 1, that GDM-AG has 2 failed trials, while GDAM shows success-rate throughout.

4.4. Breast Cancer

This problem deals with the classification of breast cancer as benign or malignant [20]. Table 1 shows that GDAM algorithm shows far better performance in reaching the desired target error value.

4.5. IRIS

IRIS dataset consists of 150 instances and deals with length and width of sepal and petals of three selected species [21]. For Iris, the best momentum value for GDM and GDM-AG is 0.2. While for GDAM, the best momentum interval is established in $[0.6-0.8]$.

4.6. Australian Credit Card Approval

Each instance in this dataset represents a real credit card application and the output describes whether the bank will grant the credit card or not [22]. GDM and GDM-AG both have the same best momentum values at 0.4. GDAM works best on the momentum interval of $[0.7-0.8]$. It is apparent from the Table 1, that GDAM is giving a percentile accuracy of 96.60.

4.7. Pima Indian Diabetes

This dataset deals with the diabetes problem in female's body [23]. For Pima Indians Diabetes, GDM and GDM-AG have the best momentum values of 0.3 and 0.2 respectively. GDAM works best on the momentum interval of $[0.6-0.8]$. As seen from the Table 1, GDAM is showing a percentile accuracy of 75.73.

4.8. Heart Disease

This dataset deals with all the symptoms of a heart disease present in a patient on the basis of information given as input [24]. For Heart Disease, GDM and GDM-AG have the best momentum values of 0.4 and 0.7 respectively.

5. Conclusions

The back-propagation neural network (BPNN) is one of the most capable supervised-learning algorithms deployed successfully in all engineering fields. Regardless of its high success rate at provid-

ing many practical solutions, it has a problem of slow convergence and network stagnancy, which still needs to be answered. This paper tries to answer the problem of slow convergence and network failure in BPNN and proposes a further improvement on the current working algorithm by Nazri [18]. The proposed 'Gradient Descent Method with Adaptive Momentum (GDAM)' works by adaptively changing the momentum and keeping the 'gain' parameter fixed for all nodes in the neural network. The performance of the proposed GDAM is compared with 'Gradient Descent Method with Adaptive Gain (GDM-AG)' and 'Gradient Descent with Simple Momentum (GDM)' in this paper. The performance of GDAM is verified by means of simulations on the eight classification problems taken from UCI machine learning repository. The final results show that GDAM works better than the previous methods and it converge to global minima successfully with high accuracy and without showing any failed trials.

Acknowledgments

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

References

[1] M. Z. Rehman et al., "Studying the effect of adaptive momentum in improving the accuracy of GDAM on classification problems," *IJMPCS*, 1(1), 2012.

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

[3] V.M. Krasnopolsky *et al.*, "Some Neural Network application in environmental sciences, Part II: advancing computational efficiency of environmental numerical models," *NN*, 16, pp. 3-4, 2003.

[4] D. E. Rumelhart *et al.*, "Learning Internal Representations by error Propagation," Journal of Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1986.

[5] K. Lee *et al.*, "Comparison of Supervised and Unsupervised Neural Networks in Predicting Bankruptcy of Korean Firms," *ESA*, 29, 2005.

[6] Y. H. Zweiri *et al.*, "Stability Analysis of a Three-term Back-propagation Algorithm," *NN*, 18(10), 2005.

[7] M. A. Fkirin *et al.*, "Change Detection Using Neural Network in Toshka Area," NSRC, Cairo, Egypt, pp. 1-10, 2009.

[8] Y. J. Sun *et al.*, "Improved BP Neural Network for Transformer Fault Diagnosis," China University of Mining Technology, 17, 2007.

[9] H. Shao, & H. Zheng, "A New BP Algorithm with Adaptive Momentum for FNNs Training," *GCIS-2009*, Xiamen, China, 2009.

[10] M. Z. Rehman et al., NIHL Prediction in Humans Using a Modified Back Propagation Neural Network," *IJASEIT*, 1(1), pp. 185-189, 2011.

[11] D. J. Swanston *et al.*, "Simple adaptive momentum: New algorithm for training multilayer Perceptrons," Electronic Letters, 30, 1994.

[12] R. J. Mitchell, "On Simple Adaptive Momentum," CIS-2008, London, UK, pp.01-06, 2006.

[13] N. M. Nawi *et al.*, "An improved Conjugate Gradient based learning algorithm for back propagation neural networks," *CI*, 4, 2007.

[14] P. Cortez *et al.*, "Modeling wine preferences by data mining from physicochemical properties," *DSS*, 47(4), 2009.

[15] J. S. Schlimmer, "Concept Acquisition through Representational Ad-

justment," *Doctorial Disseration*, UC, Irvine, 1987.
[16] J. R. Quinlan *et al.*, "Inductive knowledge acquisition: A case study," *2nd ACAES*, 1986.
[17] C. Yu and B. Liu, "A Backpropagation algorithm with adaptive learning rate and momentum coefficient," *IJCNN'02*, Honolulu, USA, pp.1218-1223, 2002.
[18] N. M. Nawi, M. Z. Rehman, M. I. Ghazali, "Noise-Induced Hearing Loss Prediction in Malaysian Industrial Workers using Gradient Descent with Adaptive Momentum Algorithm," *IRECOS*, 6 (5), 2011.
[19] M. Z. Rehman, N. M. Nawi, "The Effect of Adaptive Momentum in Improving the Accuracy of Gradient Descent Back Propagation Algorithm on Classification Problems," *CCIS*, 179(6), pp.380-390, 2011.
[20] W. H. Wolberg *et al.*, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *NAS*, 87, pp. 9193-9196, 1990.
[21] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annual Eugenics*, 7, pp. 179-188, 1936.
[22] J. R. Quinlan, "Simplifying Decision Trees," *Journal of Man-Machine Studies*, 27, pp. 221-234, 1987.
[23] J. W. Smith *et al.*, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," *SCAMC'88* IEEE, pp. 261-265, 1988.
[24] R. Detrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *American Journal of Cardiology*, 64, pp. 304-310, 1989.

TABLE 1. Selected Classification Problems for GDAM Algorithm

Classification Problem (s)	Algorithm	Mean No. of Epochs	Standard Deviation	Accuracy
Wine Quality	GDM	2801	6.21	71.99%
	GDM-AG	481	5.23	75.70%
	GDAM	22	1.07	90.48%
Mushroom	GDM	3000	1.21	50.96%
	GDM-AG	730	12.57	89.39%
	GDAM	65.47	0.36	96.34%
Thyroid Disease	GDM	1452	1.58	94.32%
	GDM-AG	960	10.01	86.38%
	GDAM	1201	0.53	95.73%
Breast Cancer	GDM	684.33	0.77	93.93%
	GDM-AG	419.60	6.98	94.06%
	GDAM	706.13	0.19	94.71%
IRIS	GDM	1481	0.23	93.85%
	GDM-AG	738.30	4.28	91.93%
	GDAM	655.67	1.09	94.09%
Australian Credit Card Approval	GDM	1293	1.15	94.28%
	GDM-AG	1635.87	11.87	91.05%
	GDAM	2355.33	0.53	96.60%
Pima Indian Diabetes	GDM	2093	2.07	70.81%
	GDM-AG	2797.07	3.63	67.81%
	GDAM	2000	0.54	75.73%
Heart Disease	GDM	1857.40	9.71	88.77%
	GDM-AG	1819.87	15.11	83.97%
	GDAM	1479.93	3.57	91.76%