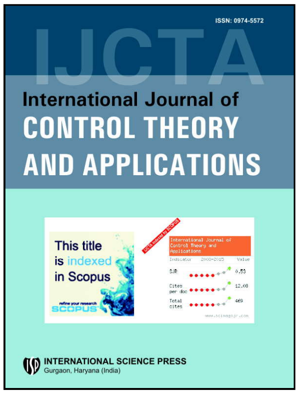


International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 19 • 2017

A Novel Normalized Adaptive Optimization Technique for Neural Networks

Parveen Sehgal[1], Sangeeta Gupta[2] and Dharminder Kumar[3]

[1] Department of Computer Science & Engineering, NIMS University, Jaipur, Rajasthan-303121, India, Email: parveensehgal@gmail.com
[2] Guru Nanak Institute of Management Guru Gobind Singh Indraprastha University New Delhi-110026, India, Email- sangeet_gju@yahoo.co.in
[3] Guru Jambheshwar University of Science & Technology Hisar, Haryana-125001, India, Email- dr_dk_kumar_02@yahoo.com

Abstract: In this paper, a novel technique for the optimization of error energy in the training of neural networks has been proposed. This new technique is a variation of adaptive gradient based optimization; which is applicable in non-linear situations. Neural Networks along with this new adaptive optimization technique have been employed to implement the predictive models for insurance data. The goal of the study is to minimize the error gradient during training of neural network in a faster way and convergence of the new algorithm is compared with existing first and second order algorithms.

Keywords: Adaptive Gradient Methods, Artificial Neural Networks, Non-Linear Optimization, Prediction Modeling, Predictive Data Mining, Supervised Learning

1. INTRODUCTION

Researchers have shown renewed interest in artificial neural networks (ANNs) over the last few decades, mainly because of invent and developments of novel training methods, which are capable of dealing with large-scale learning problems. Also, ANNs are flexible and non-parametric modeling tools, which can model any complex function mapping with higher accuracy [1, 2].

Neural networks can be trained to store, recognize, estimate and adapt to new patterns without having any initial hypothesis of the function it receives and works well in real life situations usually involving complex non-linear relationships present among the data. They are capable of approximating any non-linear function without any prior information about the relationships present among data. The powerful characteristics of learning and adapting to real life situations have made ANNs superior to the traditional techniques used in the past. They have been widely applied in solving complex problems in the fields of engineering, biological modeling, prediction modeling, decision modeling, control systems, manufacturing, business problems, health and medicine, ocean and space exploration etc. [3-5].

Much research has been carried out to develop a variety of training methods; which are used to train different types of neural networks. From the optimization literature, we know that there are many first and second order iterative methods that can minimize the error function during training of the network, like steepest descent [6], Newton's method, Levenberg Marquardt method, conjugate gradient, scaled conjugate gradient etc. [7, 8] These techniques vary according to how they adjust their estimates of the parameter value to minimize the error function. But the main idea remains the same is to achieve the faster rate of convergence while trying to find the point of minimum error [9]. Techniques mentioned above follow a numerical optimization procedure to compute the best values for step size in iterations to reach towards the point of minimum error. Besides the above-mentioned methods, heuristics based adaptive learning rate adaptation and momentum methods are also well-known training techniques for neural networks [4, 10]. The fundamental reasons that justify the importance of adaptive learning rate methods are that the value of the learning rate should be sufficiently large to allow a fast learning process, but should be small enough to guarantee convergence towards minimum gradient. The other reason is that the trial and error search for the best initial values for the parameters can be avoided because the learning rate adaption is able to quickly adapt from any initial values to the proper values [9,11].

The main objective of this work is to suggest a novel technique, which is better than existing adaptive gradient based techniques in terms of convergence and can reach the point of minimum of error gradient in lesser time during training of neural network. Here, we have proposed a new variation of adaptive gradient based training algorithm and we have called this algorithm as a normalized adaptive algorithm. Also, to verify the speed improvement of the novel technique, we have compared the convergence of this method with popular existing first order and second order training techniques. We have applied and tested this new technique and existing techniques to develop prediction models in MATLAB [12] to predict the customer's behavior on insurance data set taken from a live data warehouse. The experimental observations and comparisons have been presented in Sec. III.

2. A REVIEW OF EXISTING ADAPTIVE GRADIENT-BASED METHODS AND PROPOSED ALGORITHM

There are two important reasons that validate the study of adaptive methods. One is that the amount of weight update that can be acceptable is to maximally adapt to the shape of the error surface at each particular situation and second is that the learning rate should be controlled and varied in such a manner that it should be large enough to allow for a fast learning, but small enough to guarantee the convergence towards the desired solution. [13, 14] In this section, we briefly review existing learning rate and/or momentum based adaptive techniques with their main characteristics.

2.1. Gradient descent with adaptive learning rate (GDA)

GDA keeps varying the learning rate in each of iteration and tries to keep the step size large to increase the speed of convergence and tries to adapt the error surface by keeping the learning stable. Instead of keeping the step size fixed, it is varied according to the complexity of the error surface. If the new error exceeds the old error by a set threshold value, then newly calculated weights are rejected. In the case of rejection of the new weight vector, we reduce the rate of learning by multiplying the current value with a fractional value which is slightly less than 1. If there is some decrement in error then newly calculated weight vector is retained and the learning rate is increased by multiplying with a fractional value slightly greater than 1 to enhance learning [15].

In a variation of this technique suggested in Ref. 16, Silva and Almeida proposed the changes in learning rate can be achieved by multiplying the current learning rate by constant fraction values. Momentum term can also be included in the technique for regulating the learning rate but is kept constant and non-adaptive in this method. Backtracking is done to revert back to points of lesser errors achieved during previous iterations if a continuous increase in the error is observed.

2.2. Gradient descent with adaptive momentum (GDM)

This method is sensitive not only to the error gradient but also tries to boost the training speed by varying learning rate according to the latest trends in the error surface. Due to the presence of the momentum term the algorithm is not trapped inside small irregularities present on the gradient surface. In the absence of momentum term, training can come to an end in a small narrow local minimum, and momentum supports the algorithm to slide through such a local minimum. Momentum constant decides the amount of influence of preceding iterations on the current iterations and therefore it can be called to approximate the second order algorithms [17, 18].

The improved delta rule including the momentum term for GDM can be written as shown below in eqn. 1. [10]

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}\ (n-1) \tag{1}$$

The momentum parameter tries to enhance learning rate in smooth areas of the error surface and slows down the search in irregular areas of the error surface. A fixed value for momentum parameter must be avoided because it causes unnecessary acceleration when the current error gradient is in opposite direction to the preceding searches and ultimately retards convergence. If the acceleration due to momentum term remains uncontrolled then it can disturb learning or move down the slope and we will never reach the desired point of minimum error. Therefore, it is absolutely necessary that the momentum term must be adjusted adaptively instead of keeping it as a fixed value [4, 11].

2.3. Gradient descent with adaptive learning and momentum (GDX)

The method combines adaptive learning rate with adaptive momentum discussed above. [19 - 21] Here, the network training function updates weights and bias values taking into consideration both the factors i.e. according to an adaptive learning rate and gradient descent momentum. Adaptive learning helps to move towards minimum error by varying the learning step size and adaptive momentum acts like a low pass filter, to ignore small features on the error surface.

2.4. Other heuristic based adaptive techniques

In addition to adaptive techniques discussed above, other heuristic-based adaptive methods do exist but are not in much use. Methods like adaptation with angle between gradient direction in consecutive iterations, adaptation with the sign of the local gradient in successive iterations, Adaptation according to the evolution of the error, prediction based adaptation, search for zero-points of the error function instead of zero-points of its derivative, adaptation based upon peak values for the learning rate etc. are some valid variations for optimizing better learning.

In angled based adaptation, we consider vector directions of the previous weight update and the present gradient descent and the value of the angle between them in successive iterations can give information about the properties of the error surface. Same direction of these two vectors indicates stability of the search procedure and therefore the value of the learning rate can be increased. But a noticeable difference between their directions denotes the presence of an irregular error surface and in this situation, the learning rate should be decreased.

2.5. Proposed normalized adaptive algorithm based on normalized difference of error gradients in successive epochs

In the simple gradient method, new weight vector can be computed from weight vector in previous iterations as shown in eqn. 2 [22], where learning rate parameter is kept fixed in the starting.

$$w_{ji_{next}} = w_{ji_{prev}} + \eta(T_j - O_j)I_i \tag{2}$$

In the simple adaptive gradient method, learning rate parameter of eqn. 2 is varied by multiplying with small fractional values (1.05 for increment and typically 0.7 for decrement) [12], which are fixed in the beginning for increasing or decreasing the step size, depending upon we are moving in the right direction or moving away from minimization of gradient values. But still there are no bounds for the new learning rates and this can create a problem in successive epochs.

In the proposed method, an adaptive factor based upon the change of learning rate is computed proportional to the difference between the gradients during successive epochs. In addition, the difference in the performance values between successive epochs is normalized to avoid a negative effect on the convergence of the algorithm. To avoid very high or very low values of the performance difference for successive epochs, the difference is normalized to set a lower and upper bound on the values. In our case, it has been set between -50% and 50 % of the initial learning rate.

In case the learning rate adaption tries to go beyond the set limits then it gets normalized to lower or upper bound depending upon the direction of adaptation. Therefore, new learning parameter can now vary from -0.5 to 1.5 depending upon the value of performance difference. If required, the lower bound and upper bound values can be varied to get a narrower or a broader span for the new learning rate according to situation, but this should be avoided so that learning rate should not fall very low resulting in poor speed of convergence and also not to go very high to avoid the oscillations during convergence and never reaching the point of minimum error. Calculation of new learning rate during the algorithm is done with the help of the new formula given below in eqn. 3.

$$\eta_{new} = \eta_{prev} + \Delta\eta_{normalizd} \tag{3}$$

Where

$$\Delta\eta \propto \frac{\partial E(n)}{\partial w_{ji}(n)} - \frac{\partial E(n-1)}{\partial w_{ji}(n-1)}$$

The modifications have been done keeping in view the goal of speeding up the convergence and in turn reducing the number of objective function evaluations required to reach the point of minimum error. The new algorithm has been implemented and tested in MATLAB [12] with a function name '*normadaptiveversion0001*'.

3. EXPERIMENTAL OBSERVATIONS AND RESULTS

3.1. Model optimization and stopping criterion

Selected training methods of first and second order along with the proposed method have been employed to train and test the neural networks based prediction models and datasets have been taken from a live insurance data warehouse. Training datasets have been divided into training, validation, and testing sets. Stopping criteria have been used to decide for stopping the training process as these criteria determine whether the network has been optimally trained toward required target of minimum gradient or the methods tend to diverge from the training path.

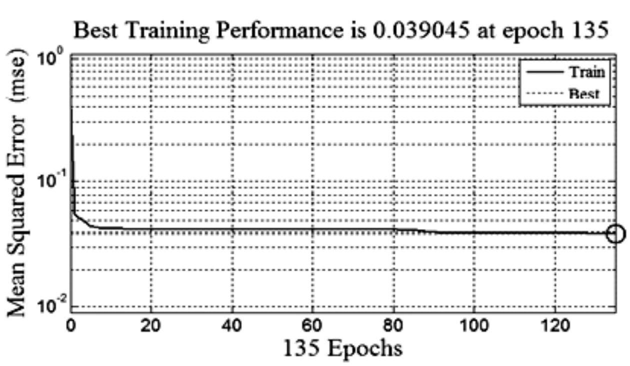
Training performance of these algorithms in terms of Mean Squared Error (MSE) and error gradient graphs have been plotted and observed to analyze their convergence and behavior to achieve the set target values of error gradient. A large number of simulations for predictive models with different configurations of neural networks have been tested in MATLAB [12]; while employing all the above said algorithms.

Table 1
Experiment results of employing different gradient-based learning algorithms of the first and second order including adaptive algorithms and the proposed algorithm

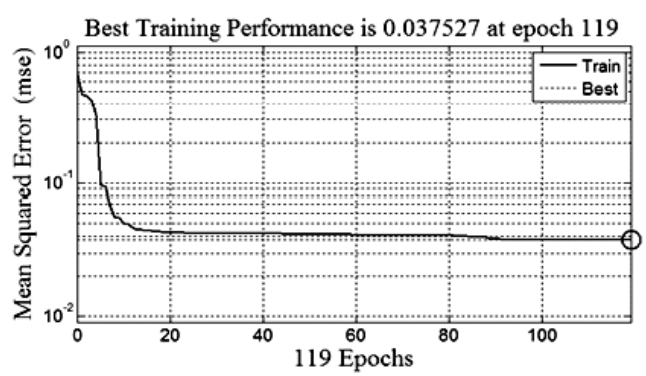
TrainingAlgorithm	*Training Function*	*Min. gradient*	*Neurons in hidden layer*	*Final epochs*	*Training time*	*Training per-formance*	*Starting gradient value*	*Final gradient value*
Conjugate gradient	traincgp	0.0001	15	135	0:07:24	0.0390	0.6760	6.96e-05
Scaled conjugate gradient	trainscg	0.0001	15	119	0:04:41	0.0375	0.447	8.04e-05
Steepest (gradient) descent	traingd	0.0001	15	10^3	0:19:06	0.0581	0.447	0.0421
Gradient descent with adaptive learning rate	traingda	0.0001	15	10^3	0:14:59	0.0428	0.447	0.0473
Gradient descent with momentum	traingdm	0.0001	15	10^3	0:14:24	0.0584	0.447	0.0427
Normalized adaptive	*norm adaptive*	0.0001	15	284	0:08:16	0.0158	0.235	8.37e-05

3.2. Results

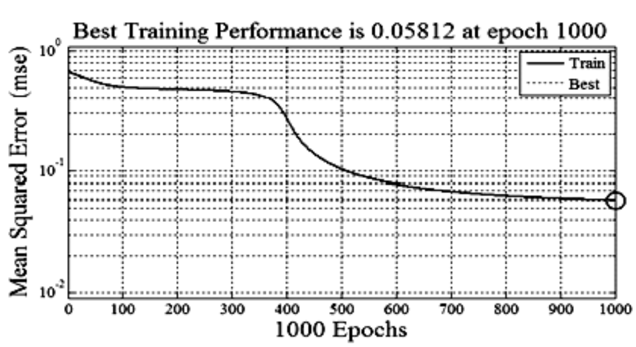
After an experimental investigation, the best results obtained for different algorithms including newly developed normalized variation have been presented in Table 1 shown above. Graphs in figures 1 and 2 illustrate respective performance and convergence behaviors of the selected methods. Figures 1(a) to 1(f) demonstrate graphs for variations in mean square error (MSE) versus numbers of epochs during the training of network with these algorithms. Figures 2(a) to 2(f) represent error gradient graphs during the training process. Convergence of different algorithms is tested for error gradient target of 0.0001. It has been clearly observed that second order methods are able to accomplish the target value of error gradient of the order of 10^{-4} like CGM found the



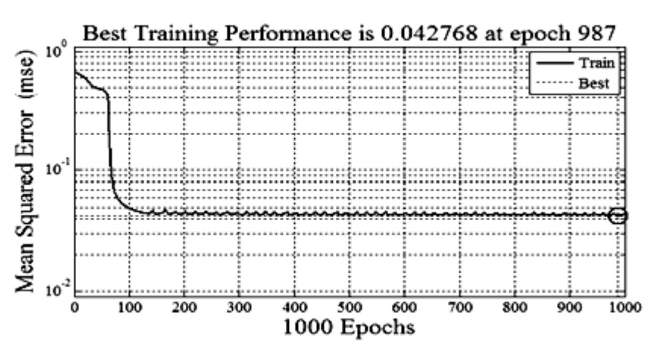
(a)



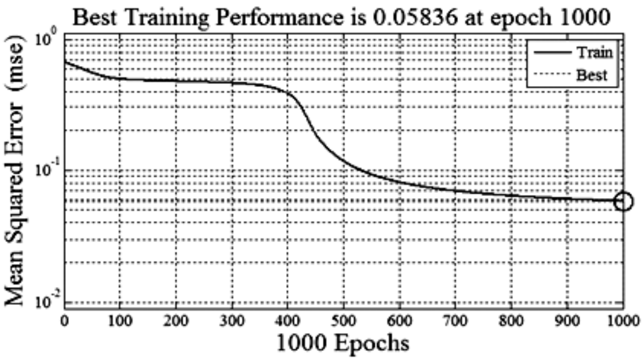
(b)



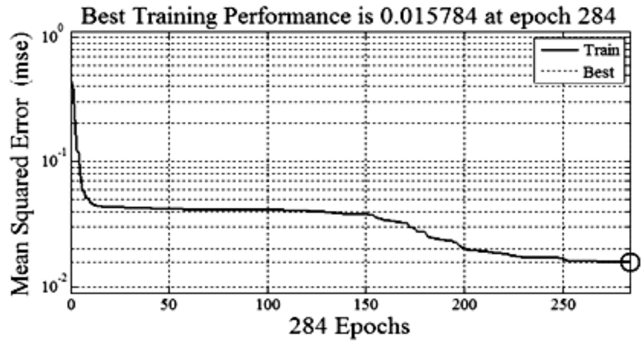
(c)



(d)

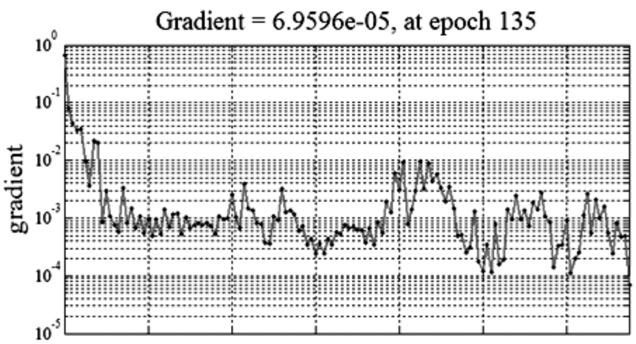


(e)

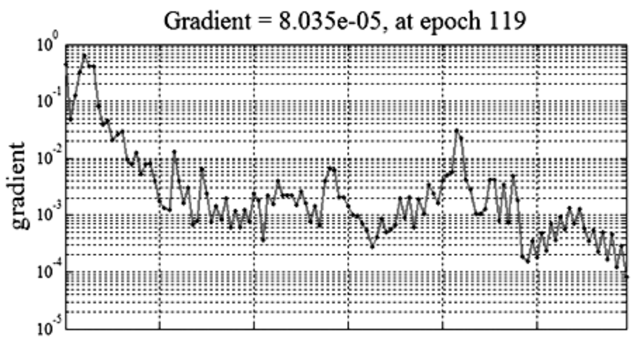


(f)

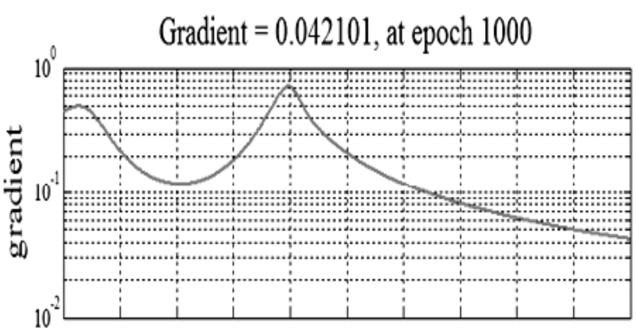
Figure 1: Training performance graph with (a) Conjugate gradient learning (b) Scaled conjugate gradient learning (c) Simple gradient descent learning (d) Gradient descent adaptive learning (e) Gradient descent adaptive momentum (f) Normalized adaptive learning algorithm



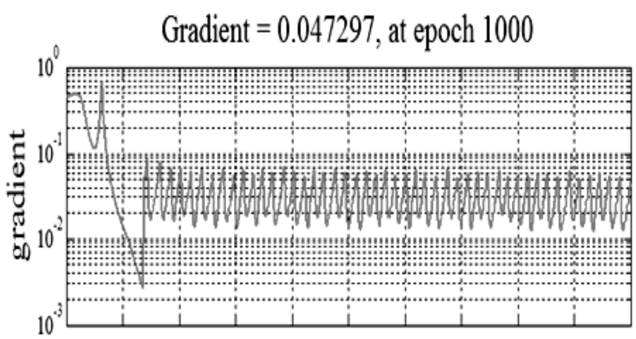
(a)



(b)



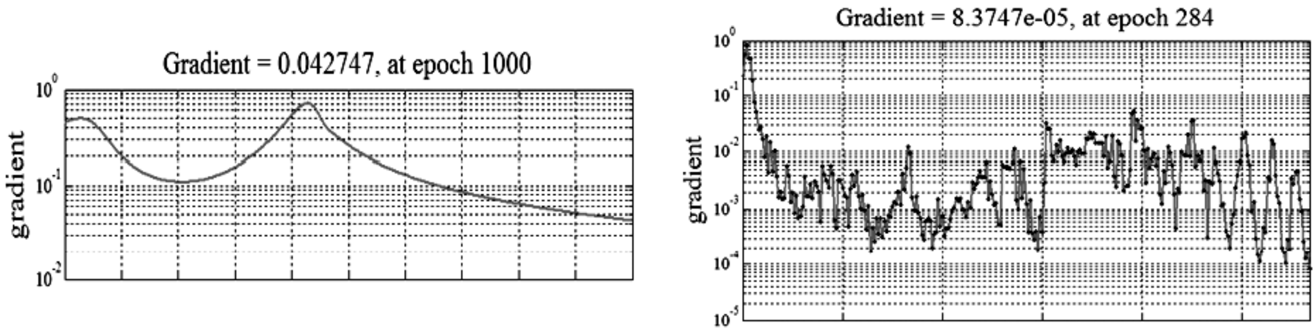
(c)



(d)

solution in 135 epochs and SCGM converged in 119 epochs. On the other hand, steepest decent and existing adaptive methods are not able to achieve the solution even in 1000 epochs but the proposed normalized adaptive algorithm is able to converge in 284 epochs.

Figure 1(f) shown above shows the graph for mean square error versus training epochs for the newly suggested method and it converges in 284 epochs for the set target of error gradient, which could not be achieved with older adaptive techniques. It has been observed that the performance of the proposed algorithm is better than existing adaptive methods but it is found below second order algorithms.



(c) **(d)**

Figure 2. Error gradient graph with (a) Conjugate gradient learning (b) Scaled conjugate gradient learning (c) Simple gradient descent learning (d) Gradient descent adaptive learning (e) Gradient descent adaptive momentum (f) Normalized adaptive learning algorithm ***[Figures1 and 2 have been plotted in MATLAB Neural Network Toolbox R2012a, V.7.14.0.739.]***

4. CONCLUSION

In this paper, we have suggested a novel adaptive gradient based technique and achieved the enhanced convergence speed of the new adaptive learning method. Performance and gradient curves for the new method have been investigated and compared with existing first and second order methods. Convergence behavior of proposed adaptive method in terms of speed and accuracy has been observed. From the results obtained, it is concluded that normalized adaptive method is much better than existing gradient based techniques but still, its performance is found less than second order techniques.

Simple gradient and adaptive techniques have not been able to converge towards set error gradient target of the order of 10^{-4} even in 1000 epochs. But, the suggested normalized adaptive method has achieved the solution for the set target value of minimum gradient. On the other hand, second order techniques have been able to reach an accuracy level of 10^{-4} and 10^{-5} and have proved much better in terms training time and convergence. Simple gradient method (GDM) with constant learning rate and the adaptive methods (GDA, GDM) have shown poor convergence, but the suggested normalized adaptive method falls in between existing first order and second order methods, in terms of convergence towards the minimum error gradient. Second order methods like conjugate and scaled conjugate gradient methods (CGM, SCGM) have shown faster convergence. From the experimental outcomes, we can conclude that normalized adaptive algorithm is a better approach than previously existing adaptive methods.

REFERENCES

[1] E. Trentin and A. Freno, "Unsupervised nonparametric density estimation: a neural network approach", in*Proc. IEEE Int. Joint Conf. on Neural Networks,* pp. 3140-3147, Atlanta, Georgia, USA, 2009.

[2] W. Sibanda and P. Pretorius, "Novel application of multi-layer perceptrons (MLP) neural networks to model HIV in South Africa using seroprevalence data from antenatal clinics", *International Journal of Computer Applications,* Vol. 35, No. 5, pp. 26-31, 2011.

[3] S. Rajasekaran and G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications*, PHI, New Delhi, 2012.

[4] M. Z. Rehman and N. M. Nawi, "Studying the effect of adaptive momentum in improving the accuracy of gradient descent backpropagation algorithm on classification problems",*International Journal of Modern Physics: Conference Series*, *World Scientific,* Vol. 1(1), pp. 1–5, 2010.

[5] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application",*Journal of Microbiological Methods*, *Elsevier*, Vol. 43, No. 1, pp. 3-31, 2000.

[6] J. C. Meza, "Steepest descent", *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2(6), pp. 719-722, 2010.

[7] T. Slavici, S. Maris and M. Pirtea, "Usage of artificial neural networks for optimal bankruptcy forecasting. Case study: Eastern European small manufacturing enterprises", *Springer*, Volume 50, Issue 1, pp. 385–398, January 2016.

[8] R. Fletcher, Practical Methods of Optimization, 2nd edn., *John Wiley and Sons*, Great Britain, 2000.

[9] R. A. Jacobs, "Increased rate of convergence through learning rate adaptation", *Neural Networks, Elsevier,* Vol. 1 (4), pp. 295–307, 1988.

[10] Saduf and M. A. Wani, "Improving learning efficiency by adaptively changing learning rate and momentum",*International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE),* Vol. 1(3), pp. 32-39, August 2014.

[11] M. Moreira and E. Fiesler, "Neural Networks with Adaptive Learning Rate and Momentum Terms",*IDIAP*, 1995.

[12] MathWorks, *Neural Network Toolbox*, *Matlab*, R2012a, V.7.14.0.739.

[13] Bikesh Kumar Singh and Kesari Verma, A. S.Thoke, "Adaptive gradient descent backpropagation for classification of breast tumors in ultrasound imaging", *International Conference on Information and Communication Technologies (ICICT 2014), Elsevier*, Procedia Computer Science, Vol. 46, pp. 1601 – 1609, 2015.

[14] I. Mukherjee and S. Routroy, "Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process",*Expert Systems with Applications, Elsevier*, Vol. 39, pp. 2397–2407, 2012.

[15] Saduf and M. A. Wani, "Comparative study of back propagation learning algorithms for neural networks",*International Journal of Advanced Research in Computer Science and Software Engineering,* Vol. 3(12), pp. 1151-1156, December 2013.

[16] F. M. Silva and L. B. Almeida, "Acceleration techniques for the back propagation algorithm", in*Proc. EURASIP Workshop,* Sesimbra, Portugal, 1990, *Lect. Notes Comput. Sci.*, *Springer-Verlag*, eds. L. B. Almeida and J. C. Wellekens, Vol. 412, pp. 110-119, 1990.

[17] A. J. Shepherd, "Second-Order Methods for Neural Networks: Fast and Reliable Methods for Multi-Layer Perceptrons", ed. J. G. Taylor *Springer-Verlag*, London, 1997.

[18] Y. Bai, H. Zhang and Y. Hao, "The performance of the backpropagation algorithm with varying slope of the activation function", *Chaos, Solitons Fractals, Elsevier,* Vol. 40(1), pp. 69–77, 2009.

[19] G. Tezel and M. Buyukyildiz, "Monthly evaporation forecasting using artificial neural networks and support vector machines", *Theoretical and Applied Climatology, Springer*, Volume 124, Issue 1, pp. 69–80, April 2016.

[20] S. J. Narayanana, R. B. Bhatt and B. Perumala, "Improving the Accuracy of Fuzzy Decision Tree by Direct Back Propagation with Adaptive Learning Rate and Momentum Factor for User Localization", *Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016), Elsevier*, Procedia Computer Science, Vol. 89, pp. 506 – 513, 2016.

[21] N. A. Hamid, N. M. Nawi, R. Ghazali and M. N. M. Salleh, "Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems", *Springer-Verlag* Berlin Heidelberg, *Communications in Computer and Information Science, CCIS* Vol. 151, pp. 559–570, 2011.

[22] D. Kumar, S. Gupta and P. Sehgal, "Improved Training of Predictive ANN with Gradient Techniques",*Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS 2014*, Vol. 1, pp. 394-399, Hong Kong, March 2014.