International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 13, Issue 8 (August 2017), PP.74-81

Predictive Data Mining with Normalized Adaptive Training Method for Neural Networks

Parveen Sehgal[1], Surendra Pal Singh[2], Dharminder Kumar[3], Sangeeta Gupta[4]
[1]Research Scholar, Department of Computer Science & Engineering,
NIMS University, Jaipur, Rajasthan-303121, India
[2]Associate Professor & Head, Department of Computer Science & Engineering,
NIMS University, Jaipur, Rajasthan-303121, India
[3]Professor, Department of Computer Science & Engineering,
Guru Jambheshwar University of Science & Technology, Hisar, Haryana-125001, India
[4]Professor, MERI, Guru Gobind Singh Indraprastha University, New Delhi-110058, India
[1]parveensehgal@gmail.com
[2]drspsingh2511@gmail.com
[3]dr_dk_kumar_02@yahoo.com
[4]sangeet_gju@yahoo.co.in

Abstract:- *Predictive data mining is an upcoming and fast-growing field and offers a competitive edge for the benefit of organization. In recent decades, researchers have developed new techniques and intelligent algorithms for predictive data mining. In this research paper, we have proposed a novel training algorithm for optimizing neural networks for prediction purpose and to utilize it for the development of prediction models. Models developed in MATLAB Neural Network Toolbox have been tested for insurance datasets taken from a live data warehouse. A comparative study of the proposed algorithm with other popular first and second order algorithms has been presented to judge the predictive accuracy of the suggested technique. Various graphs have been presented to analyse the convergence behaviour of different algorithms towards point of minimum error.*

Keywords:- *Adaptive Learning Methods, ANN, Gradient based Training Algorithms, KDD, Predictive Data Mining, Second Order Optimization.*

I. INTRODUCTION

In today's demanding age of fierce competition, Knowledge Discovery in Databases (KDD) is gaining popularity and becoming vital as a strong analytical solution for deriving useful information from bulk volume of data. [1] In particular, Predictive Data Mining (PDM) is gaining importance and has been successfully applied in many domains and important research areas. Predictive modelling is a well-defined process which uses data mining techniques and probability to forecast outcomes based upon historical data and trends. These forecasts can be beneficial in deciding for the company policies, medical diagnosis, weather forecasts or deciding the market future trends etc. Development of predictive models is based upon a number of predictor variables, which are likely to influence future results. Once data has been collected for relevant predictors, a statistical model can be formulated [2]. But it is hard to build traditional statistical methods based prediction models as it requires a number of initial hypotheses and with these older techniques it is difficult to handle the large amount of non-linear data which is generally present in real life situations. A variety of traditional techniques like regression, decision trees, naïve bayes, support vector machine, nearest neighbourhood methods etc. [3] have been used earlier for development of decision support systems (DSS) and for the predictive data mining purpose. But nowadays, intelligent techniques and algorithms like neural networks, genetic algorithms, fuzzy techniques, evolutionary algorithms, hybrid techniques etc. are upcoming techniques and are in the development stage.

In recent years, researchers have shown renewed interest in artificial neural networks (ANN) for development of predictive data mining models, mainly because of invent and developments of new training algorithms, and their suitability to deal with large volume of non-linear data and relationships present among real life data. Also, ANNs are flexible and non-parametric modelling tools, which can model any complex function mapping with higher accuracy [4], [5]. These new techniques prove to be more accurate and efficient in comparison to the traditional statistical techniques in certain areas of data mining like prediction, sequencing, clustering etc.

In this research paper, we have suggested a new gradient based algorithm to train the neural networks, and this novel technique is better than existing adaptive gradient based techniques. With this new technique, we can reach the point of target error gradient in lesser time during training of neural network and achieve better convergence behaviour. We have suggested a variation of adaptive gradient based training method and we have named this novel variation as 'normalized adaptive gradient method'. We have also presented a comparative investigation for the suggested technique with existing popular gradient based training techniques of first and second order to test the relative improvements in training speed and convergence behaviour. Prediction models based upon ANN have been constructed in MATLAB Neural Network Toolbox [6] and deployed to test the results on insurance data sets.

II. PREDICTIVE DATA MINING WITH ANN

Inspired from biological neural networks (BNN), artificial neural network based models are artificial intelligence models designed to replicate the human brain’s learning process and behaviour. ANN based models have proved superior to the traditional models because of their intelligent nature of learning and adapting to complex and non-linear data present in real life situations. They have been successfully used in important prediction based applications in the fields of research and engineering, forecasting, control systems, manufacturing, decision modelling, business problems, health and medicine, agriculture, biological modelling, ocean and space exploration etc. [7], [8].

A neural network is an interconnected network of artificial neurons with a weight update rule to adjust the strength or weights of the connections between the units in response to training data as shown below in Figure 1. The model consists of three main layers: input data layer (to input predictor attributes), hidden layer(s) (commonly referred as “black box”), and output layer (for predicted output).

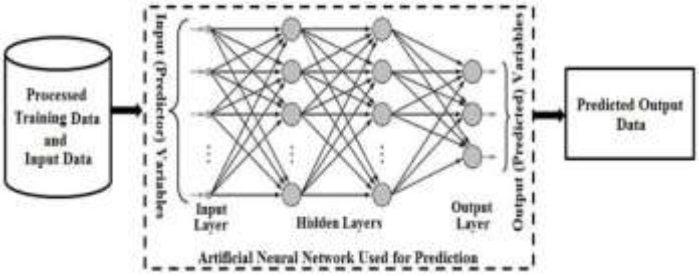

Processed Training Data and Input Data
Input (Predictor) Variables
Input Layer
Hidden Layers
Output Layer
Output (Predicted) Variables
Predicted Output Data
Artificial Neural Network Used for Prediction

Fig. 1: Prediction modelling based on ANN. [12]

ANN is trained with supervised learning method to optimize the network weights and thus to minimize the error energy towards the desired target value. With help of a suitable training algorithm, we descend towards the point of minimum error on a multidimensional error surface by moving in small steps until we achieve the point of minimum of the error function.

III. BACGROUND STUDY AND RELATED WORKS

In recent times a variety of gradient based algorithms have been developed by researchers which have been used to train a variety of neural network architectures. Most popular among them are gradient descent [9], Levenberg Marquardt algorithm, Newton’s method, quasi-Newton methods, some variations of conjugate gradient based methods, scaled conjugate gradient method etc. [10], [11]. These techniques and algorithms basically depend upon computation of step size or learning rate parameter and finally optimize the synaptic weights in ANN to minimize the error gradient function. Here we review some important gradient based techniques of first and second order.

In simple gradient descent this method, we move near to the point of minimum error in small steps on the multidimensional error surface, in a direction opposite of the error gradient to search for the point of minima on the error surface. New weight matrix is computed according to the following eqn. (1) [12].

$$w_{ji_{next}} = w_{ji_{prev}} + \eta(T_j - O_j)I_i \qquad (1)$$

Parameter η known as learning rat parameter varies the speed at which we move toward the set target for minimum of error gradient on the error surface.

Unfortunately, even for mildly non-linear cases, this algorithm shows a poor convergence and is not useful for real life situations. Instead, second order techniques which are more powerful than simple gradient such as conjugate gradient and Quasi–Newton techniques [11]-[14] are preferred for training of the neural network in non-linear cases.

In conjugate gradient based algorithms, we move toward the point of minimum error using the error gradient and move along successive non-interfering directions [15]. The scaled conjugate gradient algorithm bypasses the time consuming line search along the conjugate directions and is one of the fastest among the well-known gradient search algorithms for larger networks [16]. Levenberg Marquardet algorithm actually interpolates between Gauss–Newton Algorithm (GNA) and the method of gradient descent and is a trust region modification to Gauss–Newton Algorithm. Main advantage of Quasi–Newton methods is that the hessian of second derivatives is not evaluated directly but an approximation for hessian is computed [17].

IV. ADAPTIVE GRADIENT BASED TECHNIQUES

Adaptive gradient based techniques like adaptive learning rate and adaptive momentum based techniques have also been very popular for training of the neural networks [8], [18]. The fundamental reasons behind adaptive learning rate techniques are the optimal control of learning rate parameter to achieve a fast training along with guaranteed convergence towards set target. Here we briefly review some important adaptive gradient based techniques for training of ANNs.

A. Gradient descent with adaptive learning rate (GDA)

In this adaptive algorithm, we keep on varying the learning rate in each iteration and try to maintain the step size large enough to increase the speed of convergence but to achieve a stable learning. If an increase in error is observed, then newly computed weight matrix is rejected and, we reduce the rate of learning by multiplying the current learning rate value with a fractional value which is slightly less than 1. If some decrease in error is observed then newly computed weight matrix is retained and the learning rate is slightly increased by multiplying the current value with a fractional value slightly greater than 1 to enhance speed of learning [19].

B. Gradient descent with adaptive momentum (GDM)

This method is not only adapts the learning rate parameter according to the changes in error gradient but also tries to optimize the training speed according to the recent changes in learning rate parameter or the learning rate momentum parameter. If we don’t consider momentum term during training, then training can stop due to presence of a small narrow local minimum on the error surface, and we will not be able to reach the desired target. Momentum constant controls the effect of preceding iterations on the current iterations and hence it can be considered to work like a second order algorithms [20].

By considering the momentum parameter the weight update rule can be rewritten as shown in eqn. 2. [18].

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}\ (n-1) \qquad (2)$$

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

The algorithm presents a combination of adaptive learning rate and adaptive momentum as discussed above [21-23]. Here, weight update rule considers both the parameters i.e. adaptive learning rate and adaptive momentum parameters.

All the adaptive methods discussed above are available as built-in functions in MATLAB [6] and have been tested on the insurance datasets and used in development of prediction models. A comparative study of these methods with the proposed algorithm is also presented.

V. PROPOSED NORMALIZED ADAPTIVE GRADIENT BASED TRAINING ALGORITHM

In gradient descent algorithm with error back propagation, new weight matrix is calculated from the previous weight matrix according to eqn. (1) and learning rate parameter η is kept at constant value. In adaptive gradient descent methods, to make training of neural network more efficient, learning rate parameter η of eqn. (1) is varied by multiplying with small fractional values slightly lesser or greater than 1 [24], depending upon we are proceeding in the right direction or moving away from error minimization on the error gradient surface. But there are no fixed limits for increase or decrease of learning rate parameter which can create problem for final convergence of algorithm.

In our proposed algorithm, a new adaptive factor depending upon the variation of learning rate in successive iterations or epochs is computed directly proportional to the difference between the gradients or ‘*perf’* values. The ‘*perf’* value in MATLAB neural network toolbox is computed based on MSE (Mean squared error performance function).

Also, the difference in the ‘*perf’* values of successive epochs is normalized to avoid learning rate going beyond limits and helping in a steady and controlled descent on the error gradient surface. In order to avoid vary high or low values of learning rate; the performance difference is normalized between fixed bounds. The boundary values can be varied and in our case the learning parameter can vary from 0.5 to 1.5 depending upon the value of current performance difference.

Depending upon the situation, the boundary limits can be changed to get a narrower or a broader span for the learning rate parameter, but this can cause learning rate to fall very low resulting in poor speed of convergence or go very high and cause oscillations near point of final solution. In the proposed algorithm, new normalized values of learning rate parameter are computed with the help of the eqn. 3 shown below.

$$\eta_{new} = \eta_{prev} + \Delta\eta_{normalizd} \qquad (3)$$

Where

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

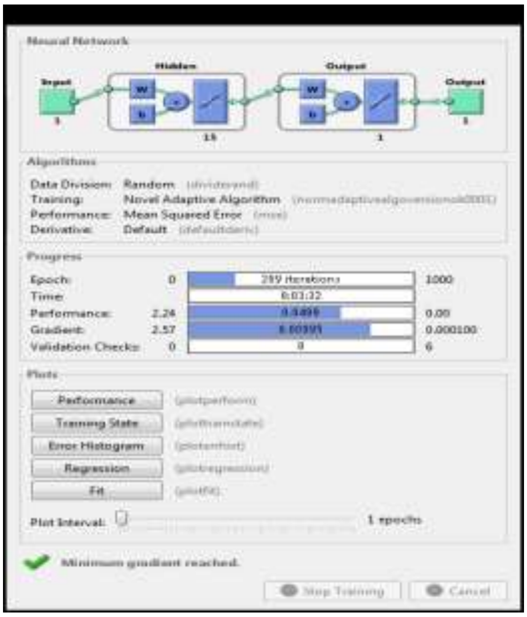
These improvements have been suggested keeping in view the goal of speeding up the training of neural network and in turn reducing the number of iterations required to reach the point of minimum error. The new algorithm is coded in MATLAB Neural Network Toolbox [6] under function name '*normadaptiveversion0001*' and has been successfully applied and tested with a variety of data sets.

VI. OBSERVATIONS AND RESULTS: A COMARATIVE INVETIGATION

We have investigated predictive performance of various models trained with first and second order algorithms including the adaptive algorithms and compared with the results obtained from the proposed algorithm. A large number of simulations have been trained and tested in MATLAB Neural Network Toolbox [6] with different configurations of neural network by changing parameters like number of neurons in hidden layers, initial synaptic weights matrix, various training functions, transfer functions and datasets etc. while employing all the above mentioned algorithms.

Different types of graphs like performance plots, error gradient curves, regression plots, confusion matrix, histogram plots, ROC plots etc. and have been plotted and judged to understand the performance and convergence behavior of the models developed. The performance curves and the error gradient graphs have been displayed in the figures 3 and 4 shown below. Datasets have been taken from a live data warehouse in insurance sector and have been divided into training, validation, and testing datasets for developing the models.

The term '*perf*' in algorithm *(defined by the network performance function 'net.performFcn')* actually represents computation of mean square error (MSE) for the given dataset and has been used for the training of the networks. After experimental investigations, the best results obtained for different algorithms including newly developed normalized variation have been presented in Table I shown below.



Neural Network
Hidden
Output
Input
W
b
W
b
Output
1
15
1
1

Algorithms

Data Division: Random (dividerand)
Training: Novel Adaptive Algorithm (normadaptivealgoversion0001)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	289 iterations	1000
Time:		0:03:32	
Performance:	2.24	0.0499	0.00
Gradient:	2.57	0.00999	0.000100
Validation Checks:	0	0	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)

Plot Interval: 1 epochs

✓ Minimum gradient reached.

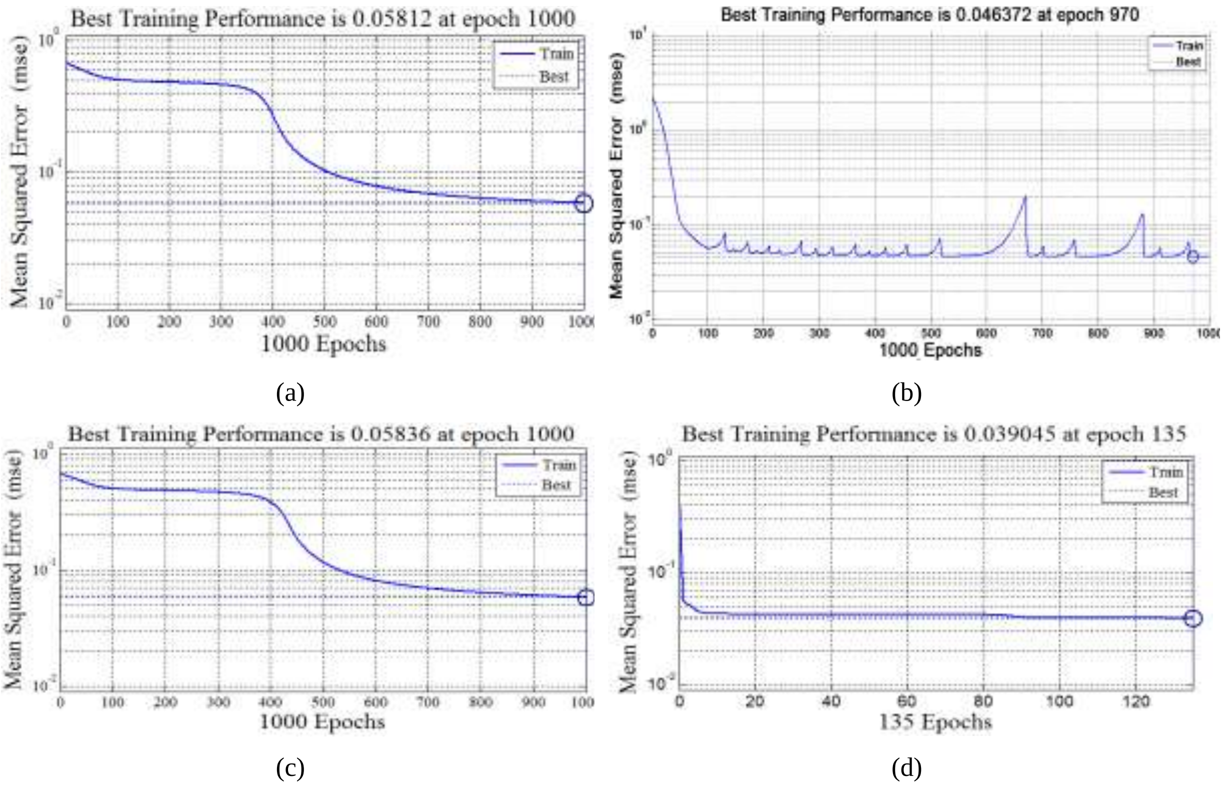
Fig. 2: Neural network training window

Table –I: A comparative investigation of results after employing various gradient based algorithms and the proposed algorithm

Training Algorithm	Training Function	Min. gradient	Neurons in hidden layer	Final epochs	Training time	Training performance	Starting gradient value	Final gradient value
Gradient descent	traingd	0.0001	15	1000	0:19:06	0.0581	0.447	0.0421
Gradient descent with adaptive learning rate	traingda	0.0001	15	1000	0:14:04	0.0464	2.57	0.0816
Gradient descent with adaptive momentum	traingdm	0.0001	15	1000	0:14:24	0.0584	0.447	0.0427
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
Levenberg Marquardet	trainlm	0.0001	15	71	0:01:39	0.0138	0.447	7.86e-05
Normalized adaptive (Proposed Method)	*Norm-adaptiveversion0001*	0.0001	15	209	0:03:41	0.0499	2.57	9.95e-03

In figure 3, we can clearly observe from the performance graphs that second order algorithms are able to converge toward final solution in a better way than first order algorithms. Best training performance value and the epoch for each algorithm has been mentioned in the graph. Convergence behaviour of the algorithms has been tested for error gradient target of 0.0001. It can be clearly seen from results that second order methods are able to achieve the set target of error gradient of the order of 10^{-4} e.g. Conjugate gradient method (CGM) has found the solution in 135 epochs (perf value = 0.0390) and Scaled conjugate gradient method (SCGM) has converged in 119 epochs (perf value = 0.0375) and Levenberg Marquardet algorithm has proved the best to achieve it only in 71 epochs (perf value = 0.0138).

On the other side, gradient decent and the adaptive methods are not able to achieve the set target even in 1000 epochs (perf values are more than 0.500) but the newly proposed normalized adaptive method is able to converge toward set target in 209 epochs (perf value = 0.499). It has been found that the performance of the newly proposed algorithm is better than other adaptive methods but it is still below the performance of second order algorithms.

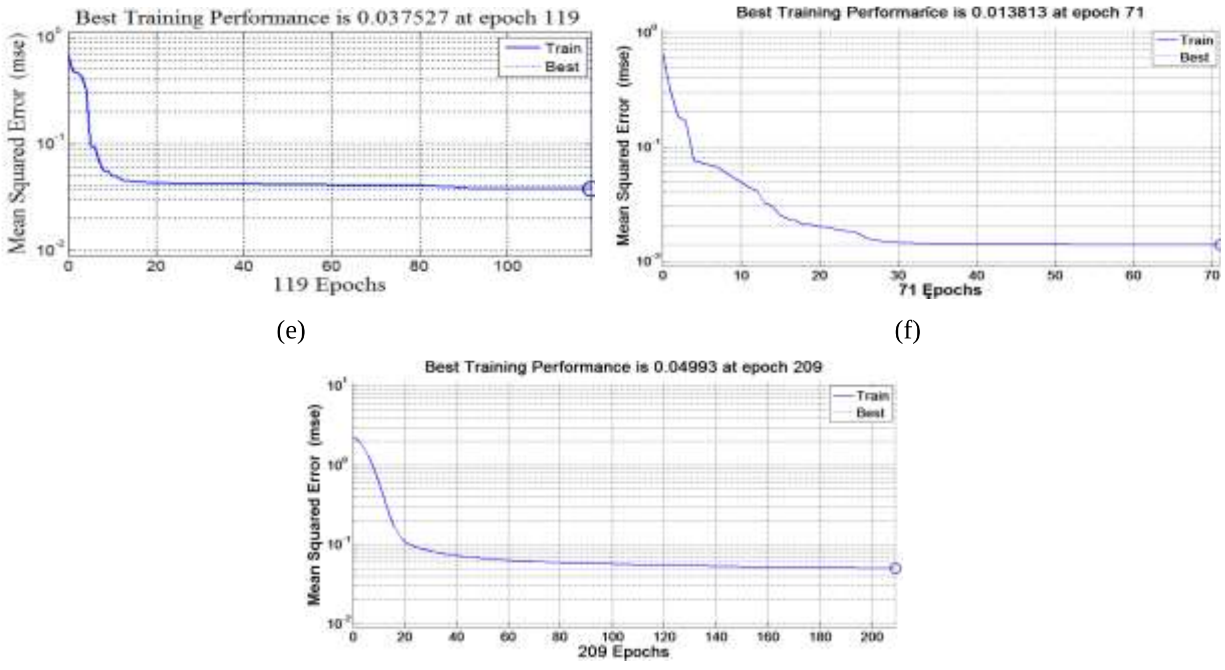


(a)

(b)

(c)

(d)



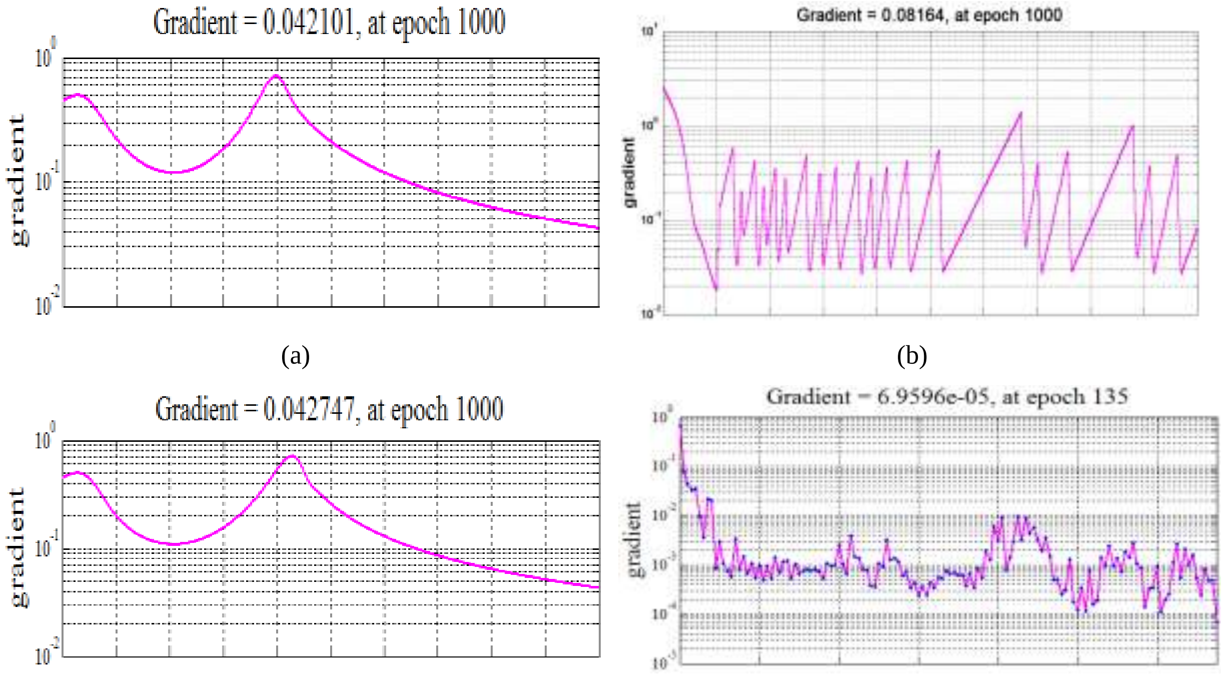
(e)

(f)

(g)

Fig. 3: Performance graph with (a) Gradient descent learning (b) Gradient descent adaptive learning (c) Gradient descent adaptive momentum learning (d) Conjugate gradient learning (e) Scaled conjugate gradient learning (f) Levenberg Marquardett learning algorithm (g) Normalized adaptive learning algorithm

In Figure 4, we can observe in the error gradient graphs that second order methods were able to reach the set target of error gradient the order of 10^{-5}. Levenberg Marquardt algorithm was able to accomplish the best minimum error gradient value of 7.86e-05 and in the minimum time of 0:01:39 minutes in 71 epochs. First order methods and adaptive techniques have failed to achieve the minimum of error gradient even in 1000 epochs. But the newly proposed algorithm has been able to accomplish the set target and achieve the final gradient value of 9.95e-03 and it converged in 0:03:41 minutes in 209 epochs. Its final gradient value is not near to second order methods but far better than first order and other adaptive techniques.

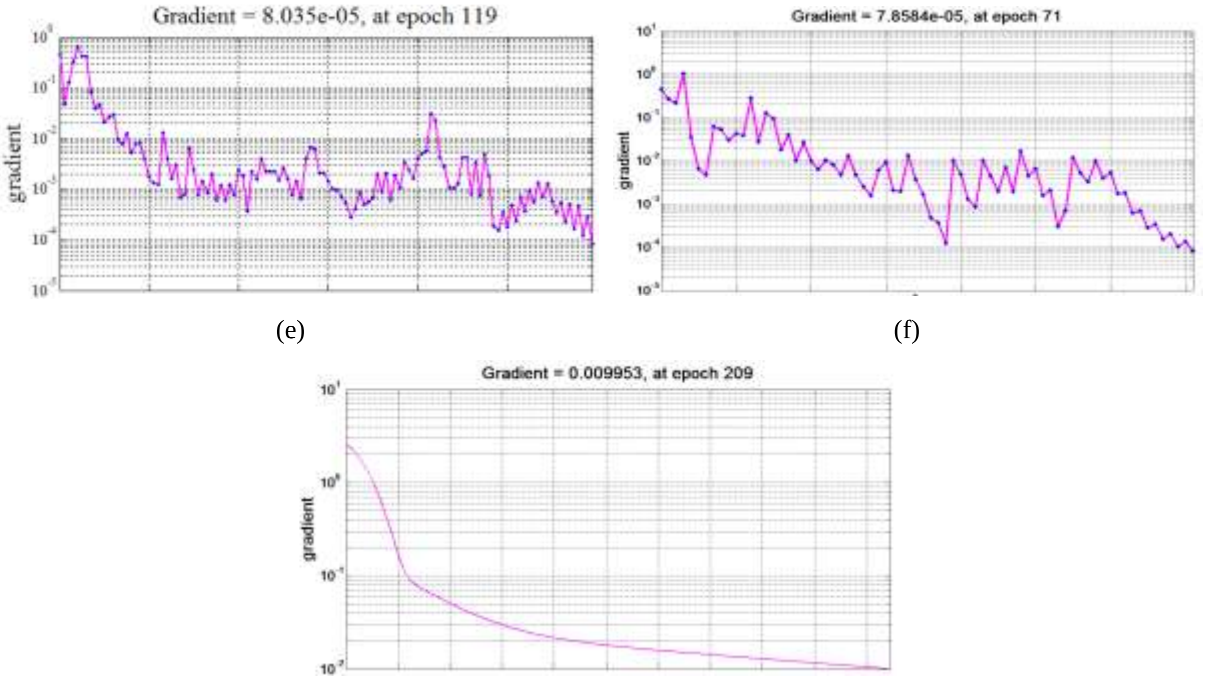


(a)

(b)

(c)

(d)



(e)

(f)

(g)

Fig. 4: Error gradient graph with (a) Gradient descent learning (b) Gradient descent adaptive learning (c) Gradient descent adaptive momentum learning (d) Conjugate gradient learning (e) Scaled conjugate gradient learning (f) Levenberg Marquardet learning algorithm (g) Normalized adaptive learning algorithm

VII. OBSERVATIONS AND RESULTS: A COMARATIVE INVETIGATION

In our research work, we have aimed at the development of an intelligent prediction model to be used in decision support systems (DSS) for an early prediction in the insurance sector based upon artificial neural networks. We have proposed a novel adaptive gradient based algorithm and accomplished an improved convergence behavior of the newly developed adaptive learning method. A comparative investigation of the proposed technique along with existing algorithms has been conducted.

After observing and comparing the performance plots, error gradient and other graphs for different algorithms, we can conclude that second order algorithms have shown the best performance and the first order methods have shown very poor performance, while performance of the newly proposed technique is found in between the first and second order algorithms. We have found the best performance value of 0.0138 with second order Levenberg Marquardet algorithm in 71 epochs and a very good performance of 0.0499 with newly proposed normalized adaptive algorithm in 209 epochs, but first order methods and other adaptive methods couldn’t converge toward set target even in 1000 epochs. Error gradient plots have also shown that second order techniques like conjugate gradient, scaled conjugate gradient and Levenberg Marquardet algorithm were able to realize an accuracy level of the order of 10^{-5} and have proved much better in terms of convergence training time. But first order methods like gradient descent and simple adaptive techniques like GDA and GDM have not been able to achieve target of the order of 10^{-4} even in 1000 epochs. On the other side, we were able to attain an error gradient value of 9.95e-03 with the newly proposed algorithm. It is also observed from regression plots that regression coefficient *‘R’* values for second order methods are greater than 0.9 and for the newly proposed method this is coming as 0.89451, which is a very good fit for the predictive accuracy of the newly developed method. In histograms plots (error bars are high near the line of zero error) and in ROC curves (plot curve hugs the left and top edges of the plot). Confusion matrix also proves the predictive accuracy for the proposed method to be excellent. Regression plots, histogram plots, confusion matrix, and ROC plots are not shown in this paper for the sake of simplicity.

From the experimental results, we can clearly conclude that newly proposed normalized adaptive algorithm is a better approach than previously existing adaptive methods.

ACKNOWLEDGMENT

We are extremely thankful to ‘KOTAK Life Insurance’ for providing the datasets for our research work from the company’s live data warehouse and giving their kind consent to utilize the datasets for testing the models. We are also grateful for facilitating to explain the metadata.

REFERENCES

[1]. P. Mittal, and N. S. Gill, “Study and Analysis of Predictive Data Mining Approaches for Clinical Dataset”, International Journal of Computer Applications, Volume 63, No. 3, pp. 35-39, February 2013.

[2]. D. Mishra, A. K. Das, Mausumi and S. Mishra, “Predictive Data Mining: Promising Future and Applications”, Int. J. of Computer and Communication Technology, pp. 20-27, Vol. 2, No. 1, 2010.

[3]. https://en.wikipedia.org/wiki/Predictive_modelling.

[4]. 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.

[5]. 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.

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

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

[8]. 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.

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

[10]. 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.

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

[12]. 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.

[13]. J. A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. vol. 97, ch. 2, P. M. Pardalos, and D. W. Hearn, Ed. New York: Springer, 2005,.

[14]. S. S. Rao, Engineering Optimization Theory & Practice. 4th ed., ch. 6, New Jersey: John Wiley and Sons, 2009.

[15]. E. K. P. Chong, and S. H. Zak, An Introduction to Optimization. 2nd Ed., John Wiley and Sons, 2001, pp. 151-164.

[16]. J. Lund´en, and V. Koivunen, “Scaled conjugate gradient method for radar pulse modulation estimation,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Honolulu, Hawaii, 2007, vol. 2, pp. 297–300.

[17]. S. M. A. Burney, T. A. Jilani and C. Ardil, “A comparison of first and second order training algorithms for artificial neural networks,” Int. J. Comput. Intell., vol. 1, no. 3, pp. 176-182, Summer 2005.

[18]. 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.

[19]. 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.

[20]. 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.

[21]. 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.

[22]. 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.

[23]. 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.

[24]. P. Sehgal, S. Gupta and D. Kumar, “Predicting for Sustainable Insurance with Adaptive Gradient Methods”, BIJIT - BVICAM’s International Journal of Information Technology BharatiVidyapeeth’s Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA), Vol. 7 No. 2; ISSN 0973 – 5658, pp. 896-902, BIJIT – 2015, July - December, 2015.

*Parveen Sehgal. “Predictive Data Mining with Normalized Adaptive Training Method for Neural Networks.” International Journal of Engineering Research and Development, vol. 13, no. 08, pp. 74–81.