The Impact of Training Iterations on ANN Applications Using BPNN Algorithm

Qamar Abbas, Waqas Haider Bangyal, and Jamil Ahmad

***Abstract*—Training Artificial Neural Network (ANN) has attracted many researchers for a long time. This paper investigates the impact of training iterations of ANN using Backpropagation Neural Network (BPNN) algorithm. The two sets of adjustable parameters, i.e., the learning rate and number of hidden nodes in the hidden layer are used to analyze the impact of training iterations of ANN applications is used. The applications that are used in this research are XOR problem and Digit Recognition. The efficacy of the results using BPNN algorithm is shown through an analysis of the impact of training iterations and by presenting simulation results from two different applications.**

***Index Terms*—XOR, digit, training iterations, BPNN**

I. Introduction

Artificial Neural networks can resolve problems in lots of applications of assorted areas such as medicine engineering, medical and so on. This paper investigates the impact of training iterations of ANN using Backpropagation Neural Network (BPNN) algorithm. In this research single-hidden-layer feedforward neural networks architecture is used for two non linear applications [1]-[3]. The two applications that are used in this research paper are the XOR problem and the Digit Recognition.

Parallel stream of information among neurons and layers formulates ANN more gorgeous and practical for real life applications [1], [2], [4]. ANN can fabricate numerous outputs in parallel behavior as fashioned in these paper applications. The fundamental rationale of the ANN is the intelligent computing like human aptitude. ANN uses neurons or nodes for the application dispensation. In ANN layers are used to carry information from input to output to converge the application. Layer is actually a collection of neurons. Neurons in reality stores the existing knowledge gained from the network [3]. BPNN is a non linear, supervised learning algorithm. BPNN uses multilayer architecture [5], [6]. Multilayer architecture has one input layer, one output layer and one or more hidden layer(s). Some of the important parameters that ANN uses to solve applications could be learning rate, number of training iterations, number of hidden neurons in the hidden layer, type of architecture used, accuracy of the target results [3], [7], [8]. Learning in fact acts to administrate the process of varying parameters used in ANN model [3], [9]. The most important thing in the ANN models is the adjustment of the weights [1], [10], [11]. The number of training iterations in weight adjusting is a core parameter.

The residue of the paper is structured as follows: in Section II, literature survey is discussed. Section III depicts proposed architecture. Section IV explains training and testing while section V presents simulation results of experiments correlated to XOR problem and digit recognition. In Section VI, conclusion of the paper is presented.

II. Literature Survey

Training of the ANN for any application is most important to consider [3], [12]. It engages a lot of the exclusive features that are quite different from the common practices of ANN. Training iterations has been used habitually in ANN but not analyzed for any application. In this research impact of Training Iterations on ANN Applications using BPNN is analyzed. In Backpropagation neural network, the training iterations parameter can have a considerable effect on generalization accuracy [9], [13], [14]. The selection of small or large number of training iterations affects the generalization accuracy and training the neural network architecture directly [1], [3].

Huynh [1] uses Singular Value Decomposition on single Hidden Layer Feedforward Neural Networks is used and compared with BP algorithm and some other algorithms and its results are proved to be better. The activation function that is used in this paper is tangent sigmoid function.

Yanling [13] discusses the network structures and methods of single-layer Perceptron and multi-layer Perceptron. XOR is a non linear problem that cannot be solved by single-layer Perceptron. The analysis presented in the paper proposes to be many solutions of the XOR problem. The solution of the XOR problem is designed by the multi-layer Perceptron.

Yan [10] describes that classical algorithms have some problems like premature limitation and low level convergence.

Considering XOR problem Ma [15] has discussed RBF and fuzzy output functions using multi-layered artificial neural networks. The function approximation and the XOR problem are compared with various models that differ in the number of trained output functions. The results that are given in the paper suggests that RBF and fuzzy output models are quicker in learning time than the conventional.

Wei [16] has used new ENN (evolutionary neural network) that uses three layer feed forward artificial neural network can approximate with precision by mapping from input space to output space. The architecture of the artificial neural network should be as simple as possible to improve

Manuscript received December 19, 2012; revised February 25, 2013.
The authors are with Iraq University, Islamabad, Pakistan (e-mail: qamar.bhk@gmail.com, waqas_bangyal@hotmail.com, jamil.ahmad@abasyn.edu.pk).

DOI: 10.7763/IJFCC.2013.V2.228

the generalization of artificial neural network for XOR problem.

Mohammed [3] has used tangent vectors for humanizing handwritten digit recognition precision. And then the results are compared with the innovative results and 2% improvement in the precision is noted.

Cousineau [17] has used XOR problem when the noise and redundancy exists. Perceptron and race networks are compared. By the addition of the redundant inputs the uncertainty could be manipulated by within channel by adding noise and between channels by adding redundant inputs. Redundancy must be low for the Perceptron model if noise is high, otherwise Perceptron model does not learns.

III. Proposed Architecture, Training and Testing

The proposed architecture for analysis used in this paper has 3 layers. The three layers are Input layer, output layer and hidden layer. The numbers of input neurons for XOR problem are 2, number of output neurons for XOR problem is 1 and the number of hidden neurons for XOR problem contains 4,6,8,10,12 and 15 neurons. Similarly the number of input neurons for digit recognition are 100, number of output neurons for digit recognition are 10 and the number of hidden neurons for digit recognition contains 6,8,10,12 and 15 neurons. The activation function used in both the applications is sigmoid function at all the layers. Sigmoid function is selected because this function is continuous and differentiable, and so less chance of local minima problem.

The XOR problem used is the simple problem of computing. And digit recognition problem uses digits that are written in the form of matrix of size 10x10. The data sets for XOR problem are 4. The library containing 1500 different handwritten data sets is prepared written by different peoples. The library is prepared by the process of preprocessing on the digits written on paper in matrix form of 10x10. The library has data in the binary form where binary 1 shows the presence of data and 0 otherwise. Out of 1500 data sets, 1000 are training input data sets, 200 are the validation data sets and 300 are the testing data sets.

TABLE I: Results of Training Iterations for XOR Problem

Training Iterations		Learning Rate					
		0.1	0.2	0.4	0.6	0.8	0.9
Number of hidden Neurons	4	70000	35000	20000	15000	11000	10000
	6	54000	30000	15000	10000	7000	5900
	8	50000	26000	13000	8000	5800	5300
	10	45000	24000	12000	7800	5800	5300
	12	44000	22000	11000	7500	5600	4900
	15	40000	20000	10000	7000	5200	4800

IV. Results and Analysis

The simulation results are shown in the form of tables and graphs. The two applications used in this paper are XOR and digit recognition. The parameters that are used in the simulation results are the learning rate, number of hidden neurons in the hidden layer, accuracy of results and number of training iteration. The impact of the training iterations then is analyzed from the simulation results. The simulation results are obtained using the BPNN algorithm.

A. XOR Problem Application

The results of Table I are obtained using simple BPNN algorithm. The accuracy of results is maintained to 98% to get these simulation results of training iterations.

TABLE II: Corresponding Accuracy of Results of Table I

Accuracy Of results		Learning Rate					
		0.1	0.2	0.4	0.6	0.8	0.9
Number of hidden Neurons	4	98.06%	98.04%	98.11%	98.12%	98.02%	98.04%
	6	98.02%	98.14%	98.14%	98.14%	98.07%	98.04%
	8	98.09%	98.13%	98.14%	98.06%	98.03%	98.06%
	10	98.06%	98.03%	98.03%	98.01%	98.01%	98.04%
	12	98.01%	98.01%	98.02%	98.05%	98.05%	98.03%
	15	98.03%	98.03%	98.04%	98.09%	98.08%	98.10%

B. Digit Recognition Application

The results of Table II are obtained using BPNN algorithm. The accuracy of results is maintained to 98% to get these simulation results of training iterations

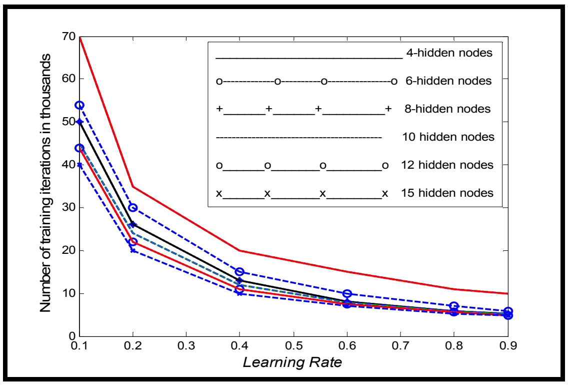

Fig. 1. Training iterations against learning rate according to table I

TABLE III: Results of Training Iterations for Digit Recognition Application

Training Iterations		Learning Rate					
		0.1	0.2	0.4	0.6	0.8	0.9
Number of hidden Neurons	6	600	350	200	200	125	100
	8	500	260	180	140	90	75
	10	350	230	160	110	80	75
	12	600	200	130	110	70	55
	15	470	220	125	80	65	55

The explanation and analysis of the simulation results is described as. Table I and table III show the number of training iterations for XOR problem and digit recognition application. The table II and table IV show the corresponding accuracy of results of Table I and Table III respectively. The Fig. 1 and Fig. 2 show the impact of training iterations against learning rate for different number of hidden neurons. It is important to note that the number of training iterations are noted by hit and trial method that is number of training iterations are noted based on the accuracy of results as shown in results.

TABLE IV: CORRESPONDING ACCURACY OF RESULTS OF TABLE III

Accuracy of results		Learning Rate					
		0.1	0.2	0.4	0.6	0.8	0.9
Number of hidden Neurons	6	98.06%	98.01%	98.12%	98.02%	98.02%	98.07%
	8	98.07%	98.10%	98.04%	98.08%	98.13%	98.19%
	10	98.06%	98.10%	98.07%	98.00%	98.19%	98.08%
	12	98.13%	98.08%	98.01%	98.03%	98.05%	98.04%
	15	98.05%	98.03%	98.01%	98.06%	98.10%	98.03%

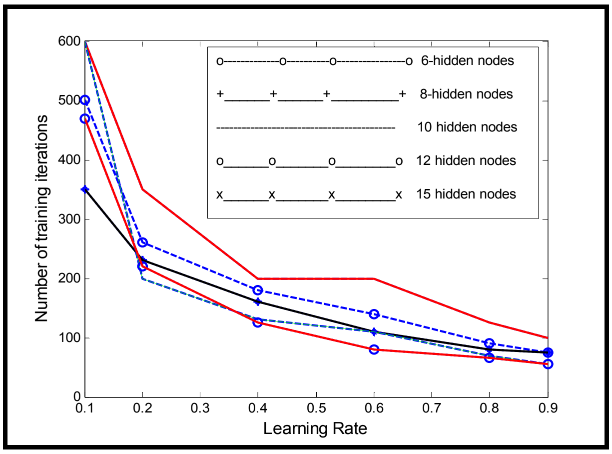

Fig. 2. Training iterations against learning rate according to Table II

V. CONCLUSION

This paper concludes that training iterations have great impact on the accuracy of results for XOR problem and digit recognition application of ANN using BPNN. Different parameters like learning rate, number of hidden neurons in the hidden layer, accuracy of results are used to analyze the number of training iterations impact on two ANN applications in this paper. It was observed that the behaviour of BPNN for XOR problem and digit recognition application, as we decrease the learning rate the learning rate required to be increase and also as we decrease the number of training iterations number of hidden neurons required to be increased. So it can be concluded that number of training iterations has impact on ANN applications using the BPNN algorithm. The future work of this paper could be to find the relationship in mathematical form between number of training iteration, learning rate and number of hidden neurons in the hidden layer to improve the BPNN algorithm performance.

REFERENCES

[1] H. T. Huynh and Y. Won, "Training single hidden layer feedforward neural networks by singular value decomposition," in *Proc. Fourth IEEE International Conference on Computer Sciences and Convergence Information Technology*, pp. 1300-1304, 2009.

[2] Z. Yanling, D. Bimin, and W. Zhanrong, "Analysis and study of perceptron to solve XOR problem," in *Proc. 2nd IEEE International Workshop on Autonomous Decentralized System*, pp. 168-173, 2002.

[3] T. Yan, Y. Tao, and D. Cui, "Research on handwritten numeral recognition method based on improved genetic algorithm and neural network," in *Proc. IEEE International Conference on Wavelet Analysis and Pattern Recognition*, pp. 1271 - 1276, 2007.

[4] L. Ma, H. Miyajima, and N. Shigei, "Multi-Layered neural networks with learning of output functions," *International Journal of Computer Science and Network Security*, vol. 6, no. 3A, 2006.

[5] H. Nemmour and Y. Chibani, "New technique in the use of tangent vectors for a robust handwritten digit recognition," in *Proc. IEEEACS International Conference on Computer Systems and Applications*, pp. 915 – 916, 2008.

[6] G. W. G. Wei, "Study on evolutionary neural network based on ant colony optimization," in *Proc. International IEEE Conference on Computational Intelligence and Security Workshops*, pp. 3-6, 2007.

[7] L. Fausset, *Fundamental of Neural Networks by Laurene Fausset*, Third Edition, 2008.

[8] Y. Zhou, D. He, and Z. Nong, "Application of functional network to solving classification problems" *World Academy of Science, Engineering and Technology*, vol. 7, pp. 390-393, 2007.

[9] A. Ahmad and H. Maen, "Recognition of on-Line Handwritten Arabic Digits Using Structural Features and Transition Network," *Informatica*, 2006.

[10] J. Maher, B. M. Ginley, P. Rocke, and F. Morgan, "Intrinsic hardware evolution of neural networks in reconfigurable analogue and digital devices," in *Proc. IEEE Symposium on FieldProgrammable Custom Computing Machines*, pp. 321-322, 2006.

[11] A. A. Salah, E. Alpaydin, and L. Akarun, "A selective attention based method for visual pattern recognition with application to hand wriiten digit recognition and face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol. 24, no. 3, pp. 420 – 425, 2002.

[12] L. Chen and B. Shi, "CMOS PWM VLSI Implementation of Neural Network," in *Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pp. 485 - 488, 2000.

[13] M. A. Otair and W. A. Salameh, "Speeding up back-propagation neural networks," in *Proc. of the 2005 Informing Science and IT Education Joint Conference,* pp. 167-173, 2005.

[14] P. J. G. Lisboa, T. A. Etchells, and D. C. Pountney, "Minimal MLPs do not model the XOR logic," *Neurocomputing*, vol. 48, pp. 1033–1037, 2003.

[15] D. Cakmakov, V. Radevski, Y. Bennani, and D. Gorgevik, "Decision fusion and reliability control in handwritten digit recognition system," *Journal of Computing and Information Technology*, vol. 10, no. 4, pp. 283-293, 2002.

[16] D. Cousineau, "Learning of an XOR problem in the presence of noise and redundancy," in *Proc. IEEE International Joint Conference on Neural Networks*, pp. 2111-2116, 2005.

[17] J. Peng, K. Li, and G. W. Irwin, "A new jacobian matrix for optimal learning of single-layer neural networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 119-129, 2008.

Q. Abbas received the BSc degree in mathematics and Statistics in 2002 from Bahauddin Zakariya University Multan, Pakistan, MSc degree in Computer Science in 2004 from Gomal University D.I.Khan, Pakistan, He completed his MS degree in computer science in 2008 from Iqra University Islamabad, Pakitsan. Currently he is a PhD (computer science) student at the Department of Computing and Technology Iqra University Islamabad, Pakistan. His research interests include pattern recognition, neural networks and evolutionary algorithms. He is an active member of several professional Societies.

W. H. Bangyal is a research scholar and Ph.D (Computer Science) Candidate at the Department of Computing and Technology Iqra University Islamabad, Pakistan.