International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

Performances of Various Back-propagation Learning Algorithms of Neural Network Using Matlab

Md. Ashek-Al-Aziz[a]*, Abdullah-Hil Muntakim[b]

[a]*Associate Professor, University of Development Alternative (UODA), Dhaka-1209, Bangladesh*
[b]*Assistant Professor and Assistant Director, University of Development Alternative (UODA), Dhaka-1209, Bangladesh*
[a]*Email: ashek3000@gmail.com*
[b]*Email: faculty.ahmuntakim@gmail.com*

Abstract

There are plenty of back-propagation learning algorithms of artificial neural network. Performances of various back-propagation learning algorithms have been checked using few portions of Australian Rain Dataset. Polak-Ribiere conjugate gradient back-propagation and Levenberg-Marquardt back-propagation have showed good performance than others.

Keywords: Neural Network; Back-propagation; Training; Testing.

1. Introduction

Artificial Neural Networks are the artificial mimic of human brain [1]. Human beings learn with the presence of teacher or guide which is a common learning paradigm [2]. Whatever the inputs received by the receptor of human being, another person tells him/her what the objects should be that is output is defined by the teacher. While this paradigm is subject to be mimicked artificially, the target output is assigned by supervisor for each corresponding inputs to the neural network. Computed output also called actual output is not same as given output or target output or desired output because inputs are multiplied by some random weight values in the neural network. In that case, weight values are changed by back-propagation [3].

----

* Corresponding author.

The process is repeated until computed output becomes same as target output. There are plenty of back-propagation learning algorithms in neural networks which are listed in Table 1.

Table 1: Back-propagation learning algorithms of neural network

Algorithm
Batch training with weight & bias learning rules
BFGS quasi-Newton back-propagation
Bayesian regularization
Cyclical order incremental training w/learning functions
Powell -Beale conjugate gradient back-propagation
Fletcher-Powell conjugate gradient back-propagation
Polak-Ribiere conjugate gradient back-propagation
Gradient descent back-propagation
Gradient descent with momentum back-propagation
Gradient descent with adaptive lr back-propagation
Gradient descent w/momentum & adaptive lr back-propagation
Levenberg-Marquardt back-propagation
One step secant back-propagation
Random order incremental training w/learning functions
Resilient back-propagation (Rprop)
Sequential order incremental training w/learning functions
Scaled conjugate gradient back-propagation

Al-Aziz and his colleagues (2021) simulated neural network using Matlab to check the performance of different neural networks with different number of hidden layers and tried to obtain the behavior pattern for different number of hidden layers. No regular behavior pattern found by them [4]. It is already mentioned that there are various back-propagation learning algorithms which can be studied for same data inputs. The objective of this research work is to check the performances of these back-propagation learning algorithm for same given dataset as input for training and testing.

2. Method

A neural network consists of 10 neurons in a hidden layer, 1 neuron in input and 1 neuron in output layer has been considered for this experiment. The network is fully connected. The neural network is shown in Figure 1. Each and every edge is connecting the input layer nodes and hidden layer nodes also hidden layer nodes and output nodes consist of weight values. The weights are updated by various back-propagation learning algorithms.

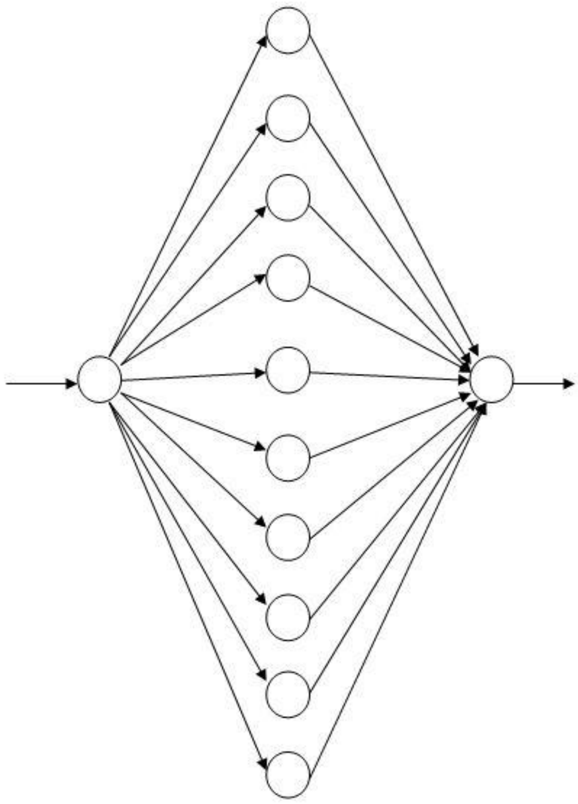

Figure 1: Neural network with input layer, output layer and hidden layer with 10 neurons

3. Experiment

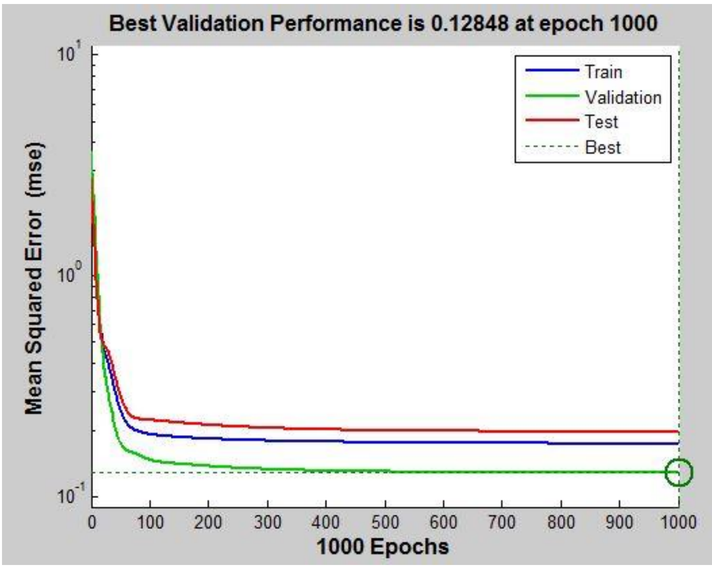

Figure 2: Best validation performance of Batch training with weight & bias learning rules

The experiment has been conducted using Matlab. The neural network is created using 'nnfeedforward' built-in function. The parameter specifications for the different back-propagation algorithms are 'trainb', 'trainbfg', 'trainbr', 'trainc', 'traincgb', 'traincgf', 'traincgp', 'traingd', 'traingdm', 'traingda', 'traingdx', 'trainlm', ‘trainoss’, ‘trainr’, ‘trainrp’, ‘trains’, ‘trainscg’. Experimental results are depicted from Figure 2 to Figure 52 of direct neural network execution. Summary of the number of iterations executed, best validation and neural network performance is shown in Table 2 also in Figure 53, Figure 54 and Figure 55. All back-propagation learning algorithms are given same dataset as input for training and testing. 800 data have been used for training and 200 data for testing from Australian Rain Dataset.

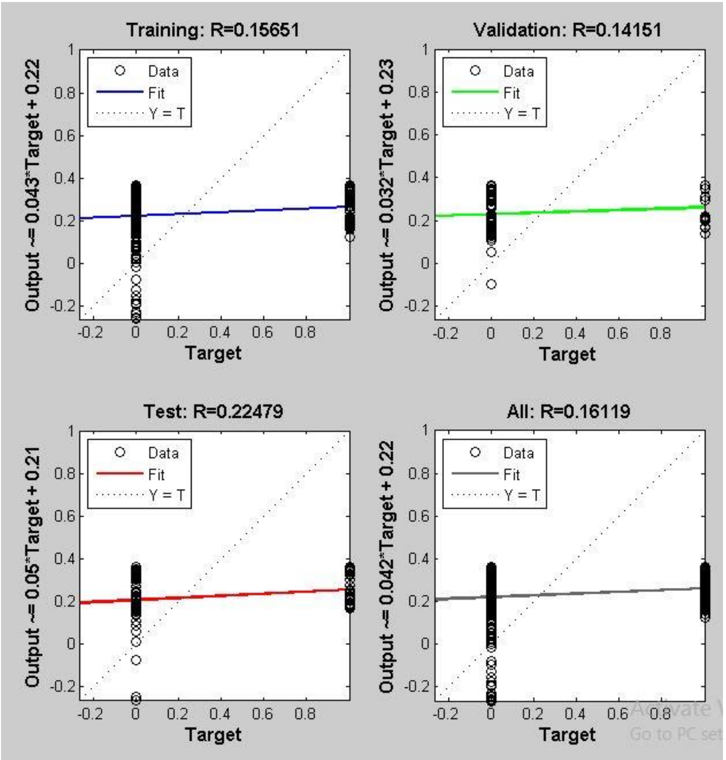

Figure 3: Regression of Batch training with weight & bias learning rules

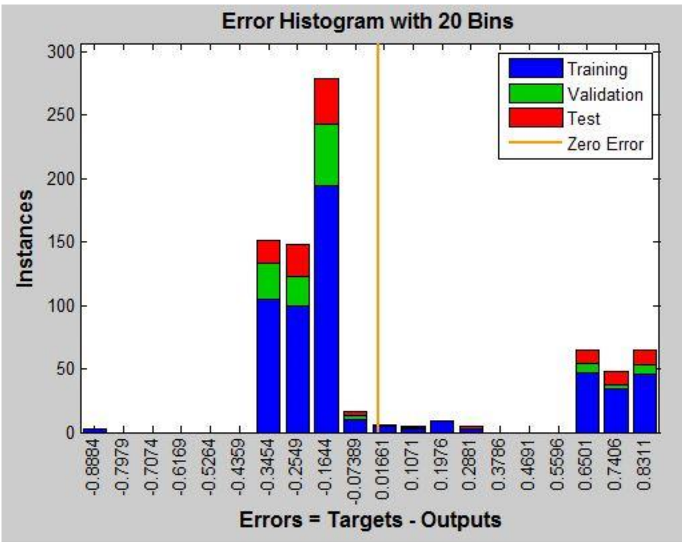

Figure 4: Error histogram of Batch training with weight & bias learning rules

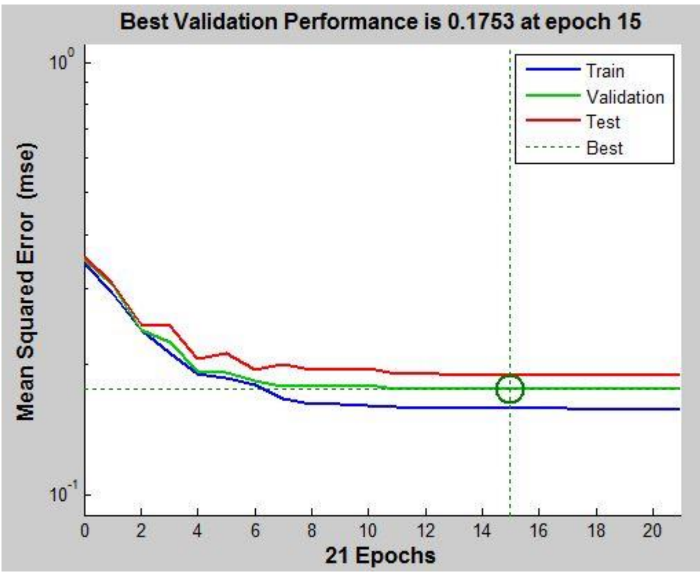

Figure 5: Best validation performance of BFGS quasi-Newton back-propagation

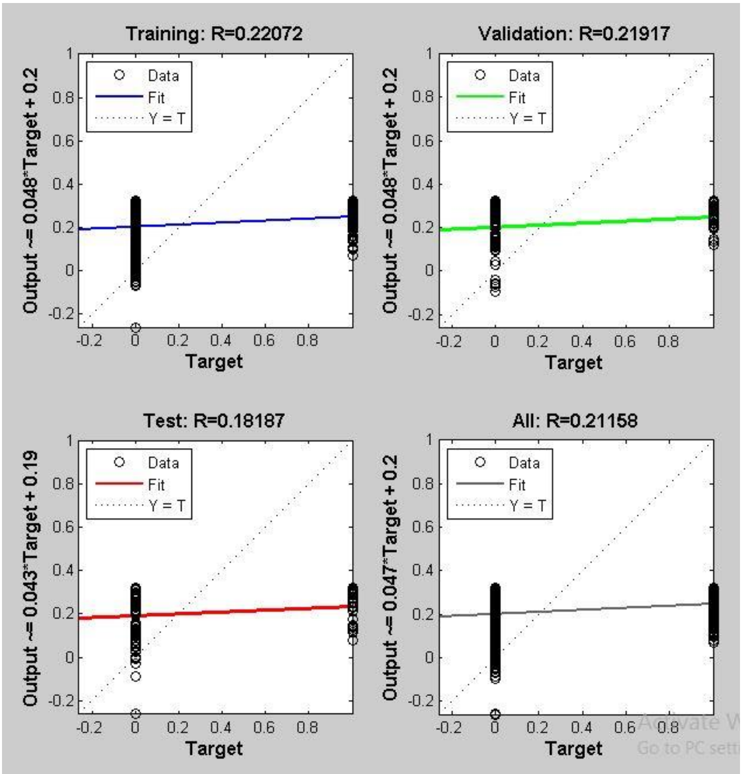

Figure 6: Regressions of BFGS quasi-Newton back-propagation

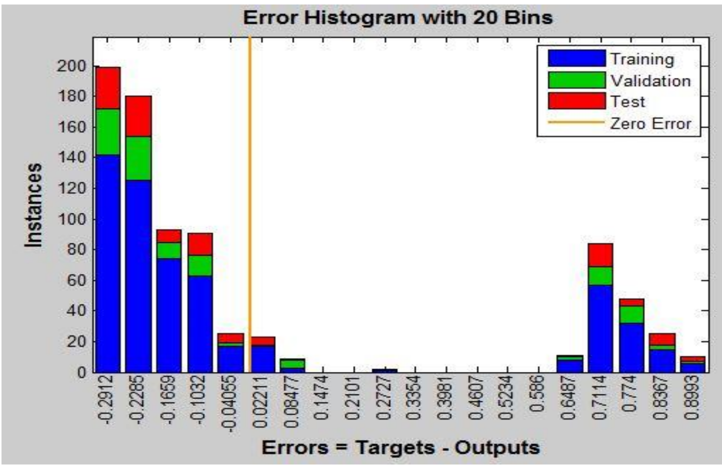

Figure 7: Error histogram of BFGS quasi-Newton back-propagation

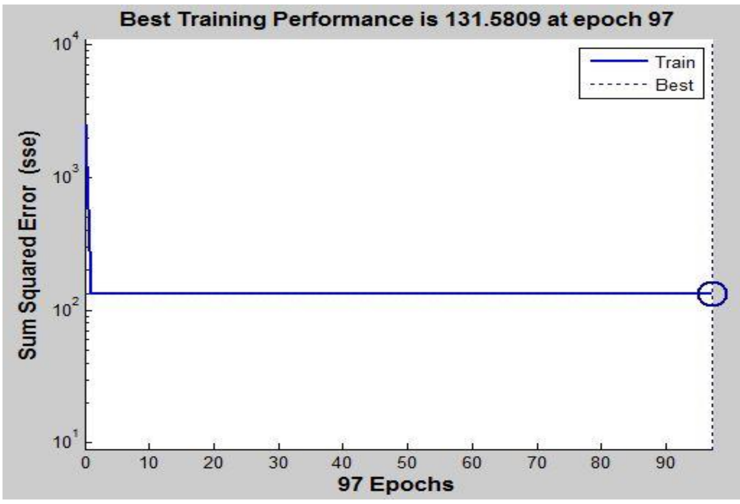

Figure 8: Best validation of Bayesian regularization

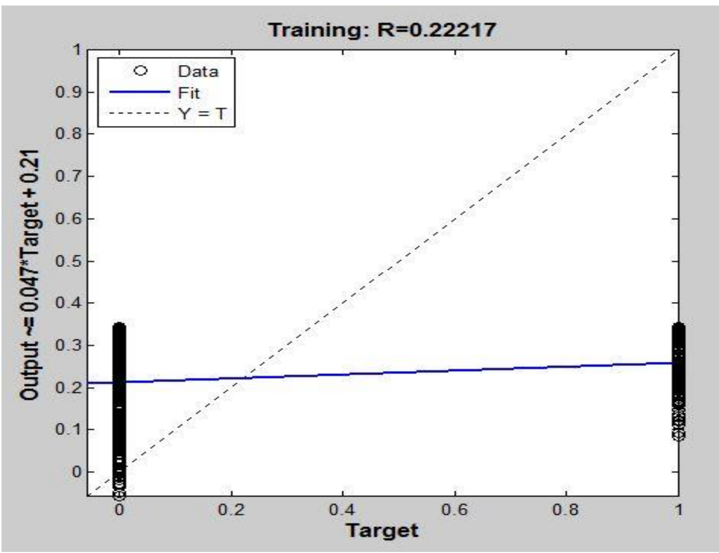

Figure 9: Regression of Bayesian regularization

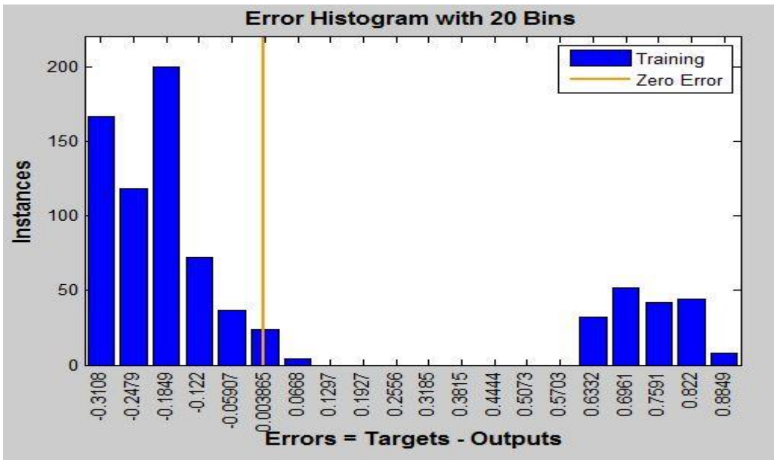

Figure 10: Error histogram of Bayesian regularization

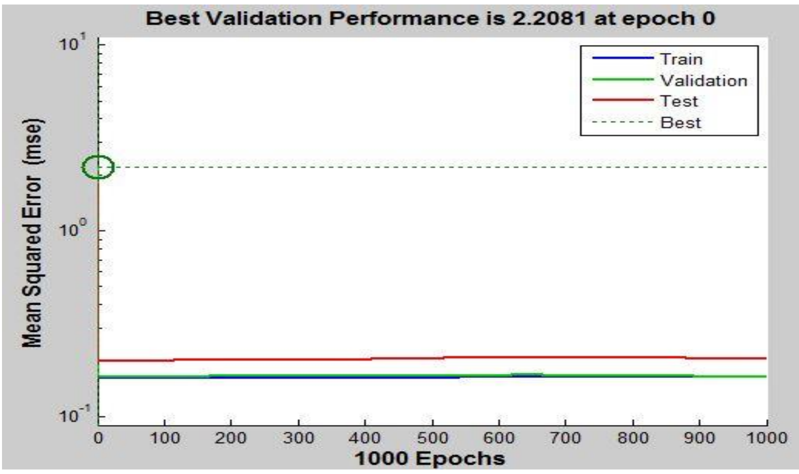

Figure 11: Best validation of Cyclical order incremental training w/learning functions

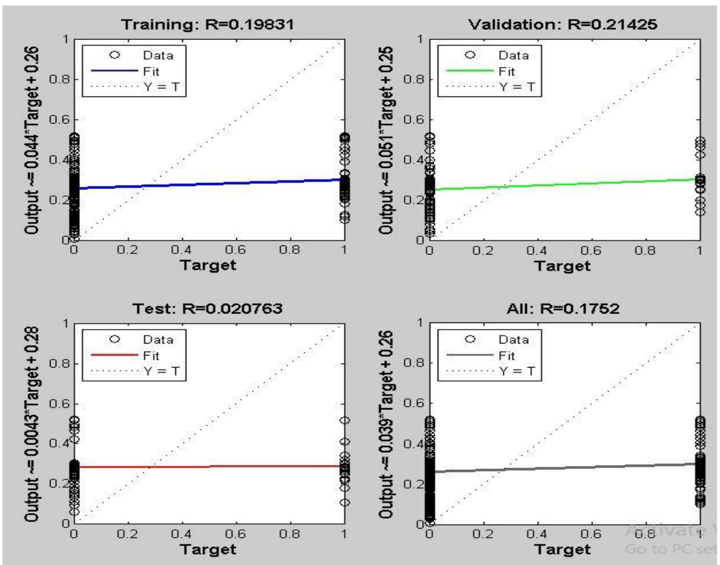

Figure 12: Regression of Cyclical order incremental training w/learning functions

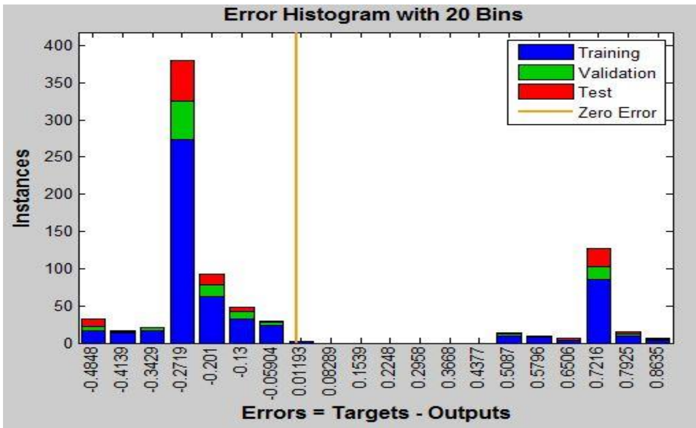

Figure 13: Error histogram of Cyclical order incremental training w/learning functions

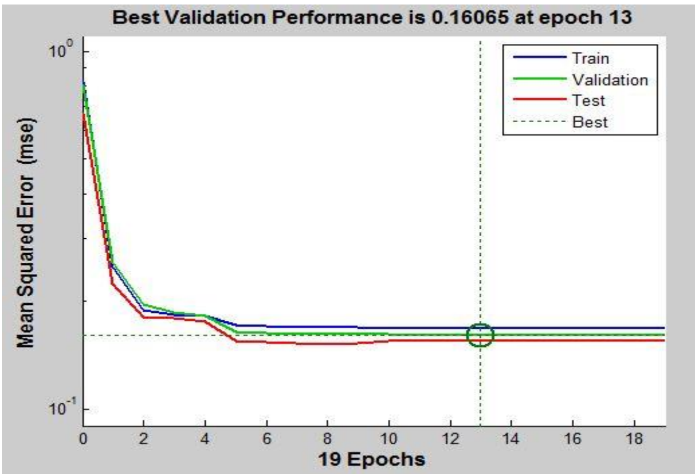

Figure 14: Best validation of Powell -Beale conjugate gradient back-propagation

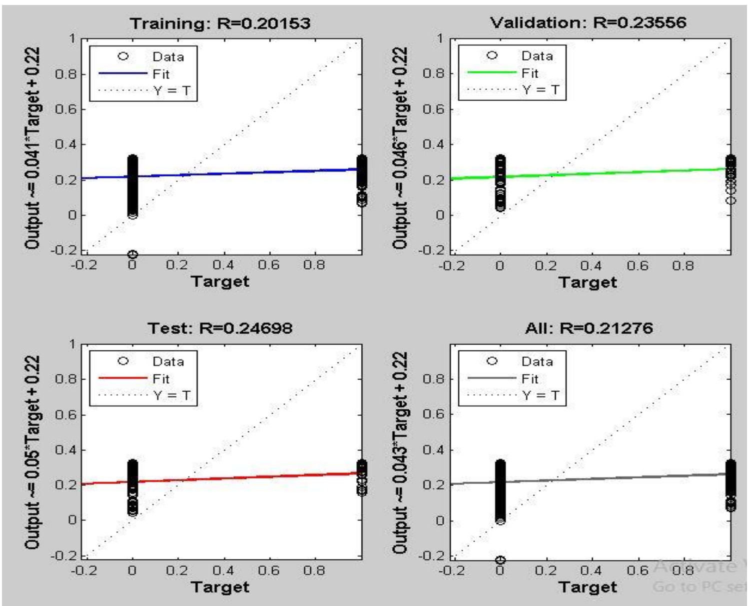

Figure 15: Regression of Powell -Beale conjugate gradient back-propagation

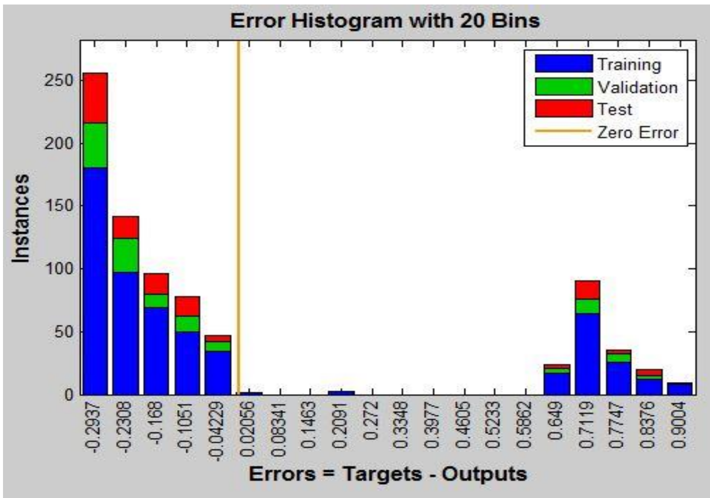

Figure 16: Error histogram of Powell -Beale conjugate gradient back-propagation

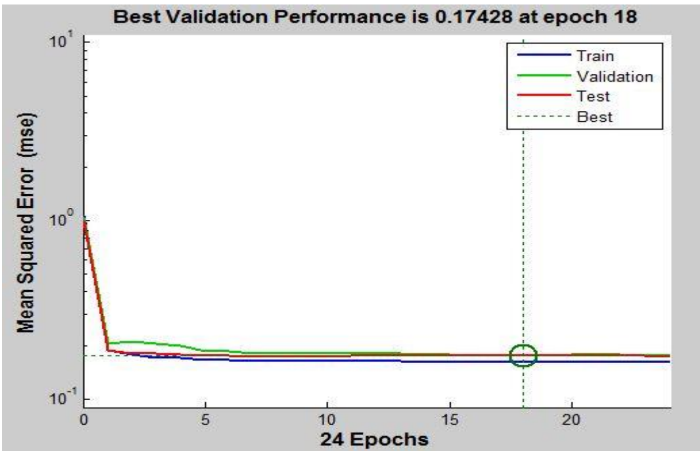

Figure 17: Best validation of Fletcher-Powell conjugate gradient back-propagation

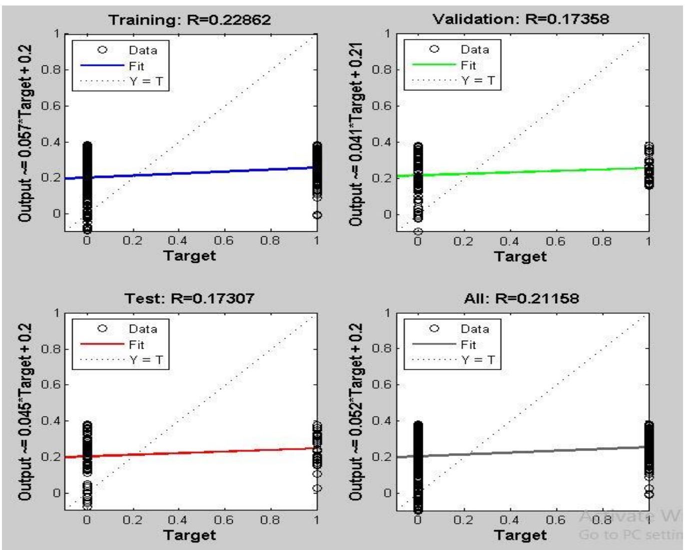

Figure 18: Regression of Fletcher-Powell conjugate gradient back-propagation

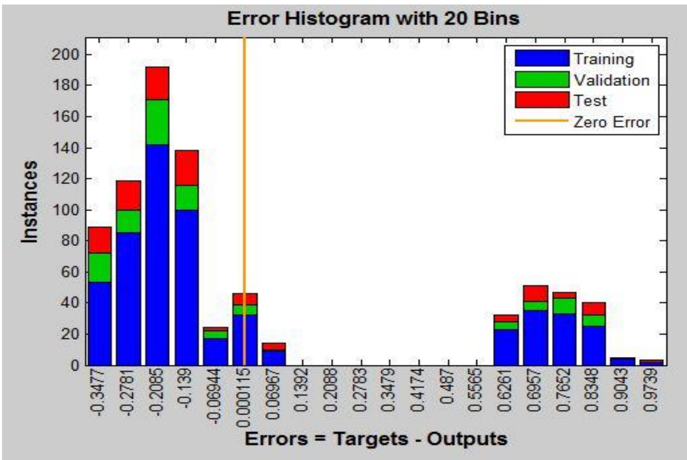

Figure 19: Error histogram of Fletcher-Powell conjugate gradient back-propagation

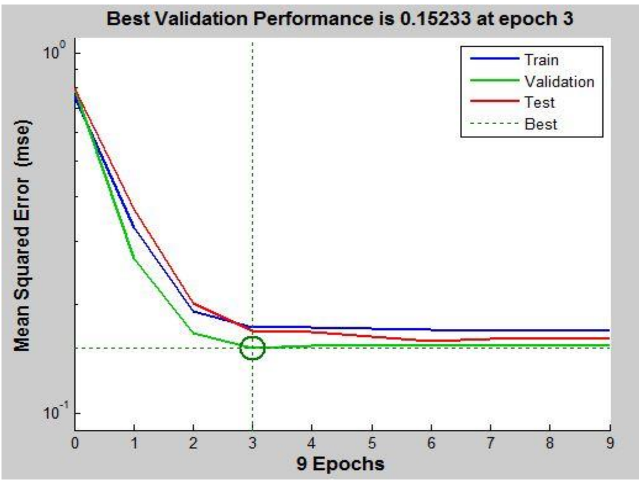

Figure 20: Best validation of Polak-Ribiere conjugate gradient back-propagation

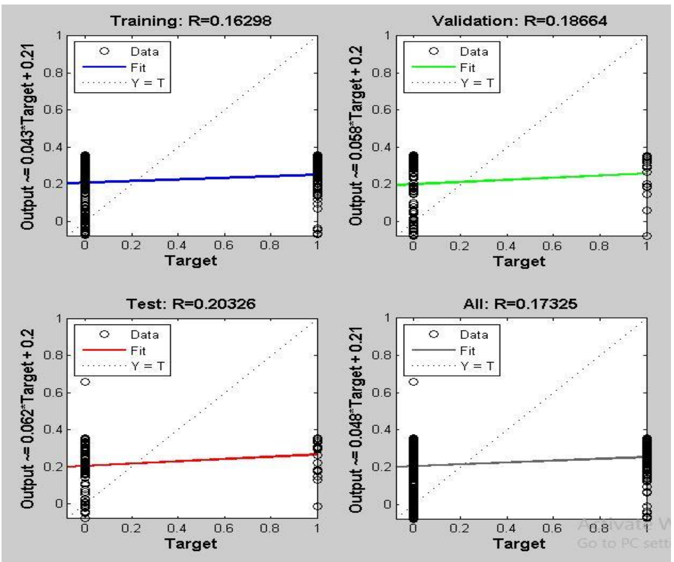

Figure 21: Regression of Polak-Ribiere conjugate gradient back-propagation

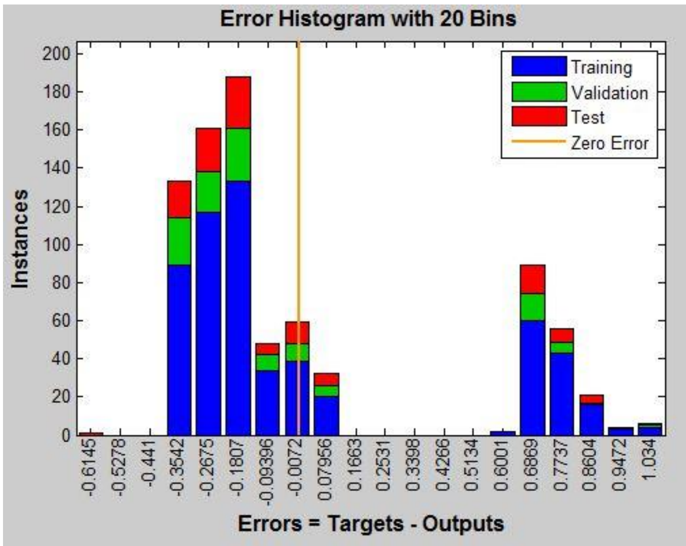

Figure 22: Error histogram of Polak-Ribiere conjugate gradient back-propagation

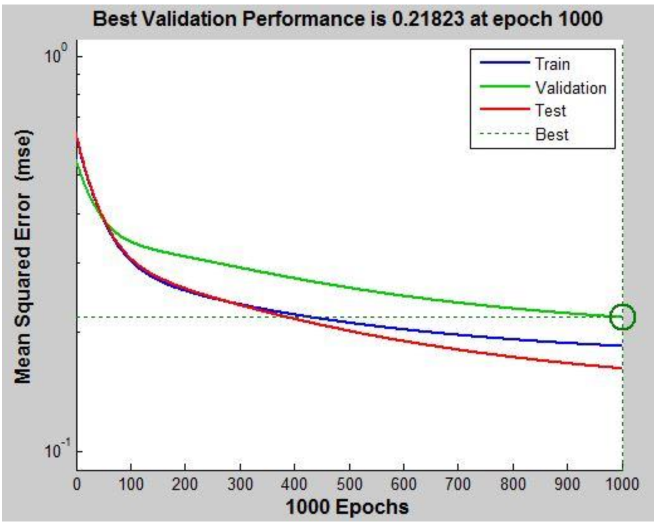

Figure 23: Best validation of Gradient descent back-propagation

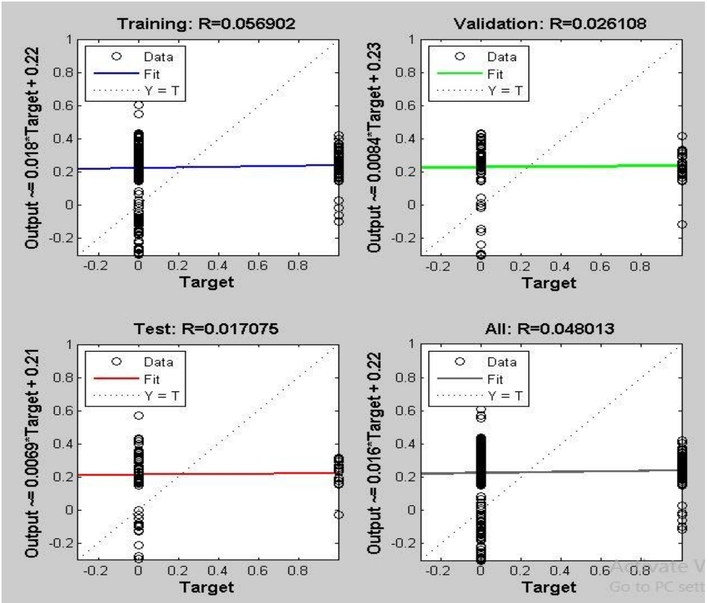

Figure 24: Regression of Gradient descent back-propagation

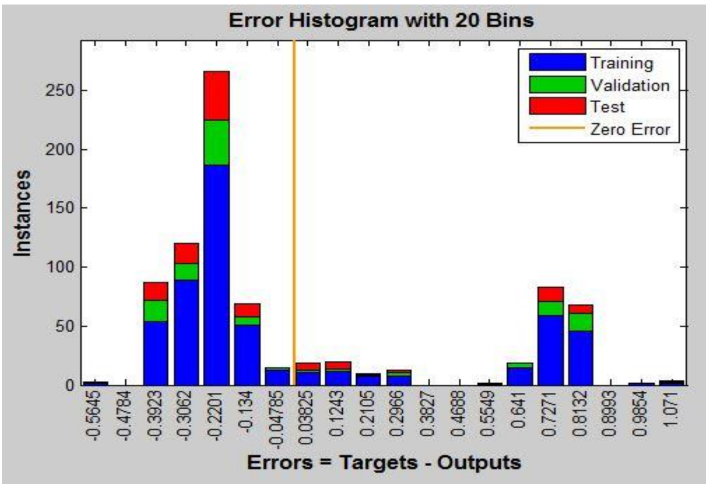

Figure 25: Error histogram of Gradient descent back-propagation

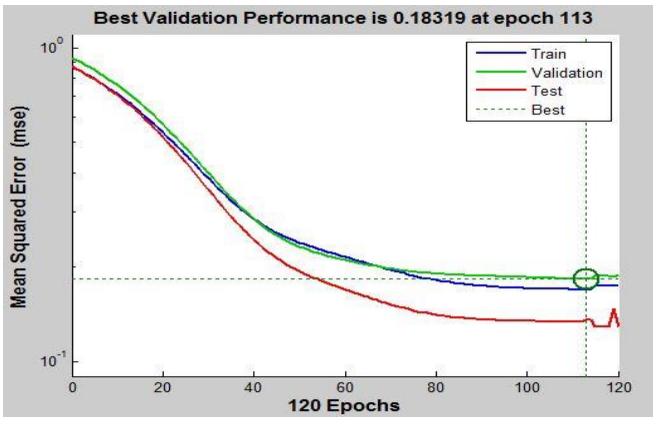

Figure 26: Best validation of Gradient descent with adaptive lr back-propagation

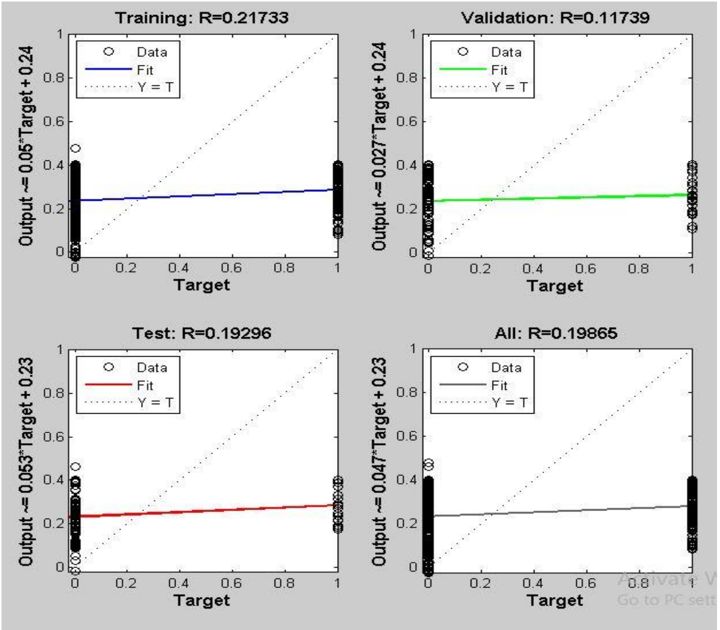

Figure 27: Regression of Gradient descent with adaptive lr back-propagation

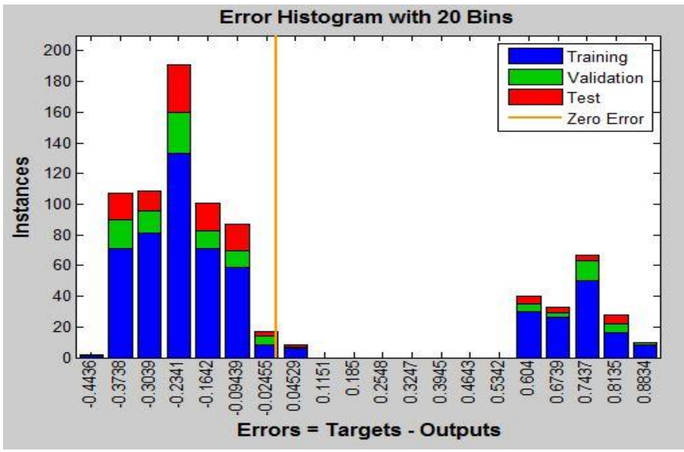

Figure 28: Error histogram of Gradient descent with adaptive lr back-propagation

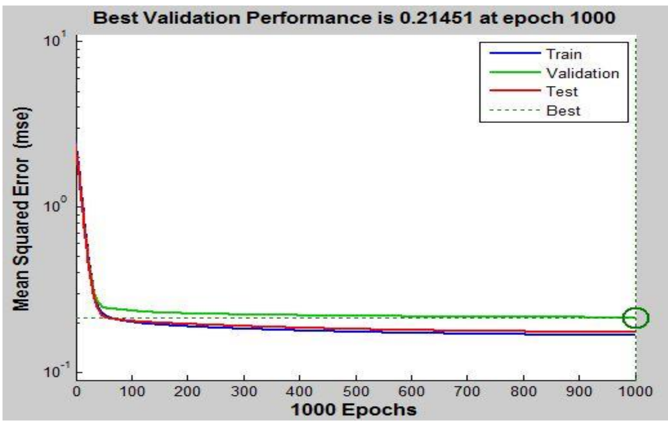

Figure 29: Best validation of Gradient descent with momentum back-propagation

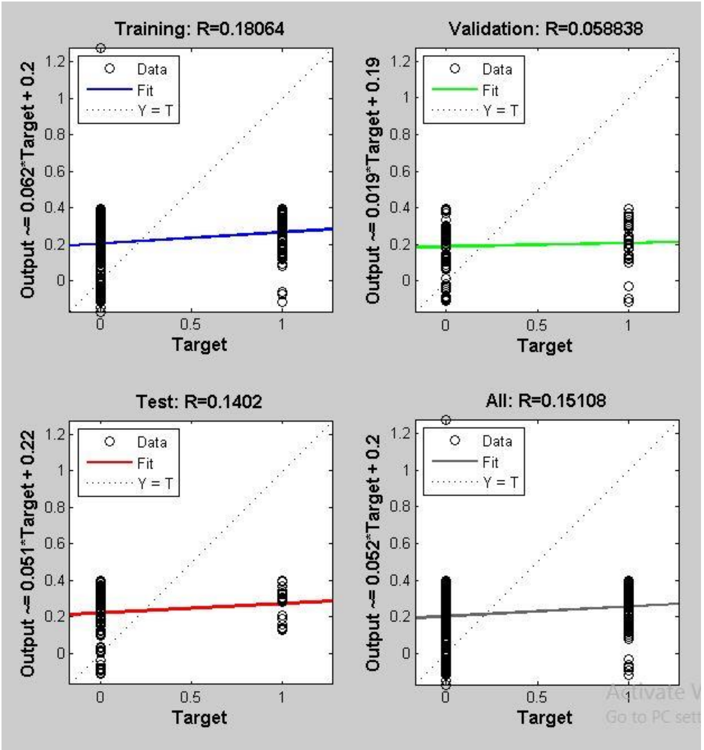

Figure 30: Regression of Gradient descent with momentum back-propagation

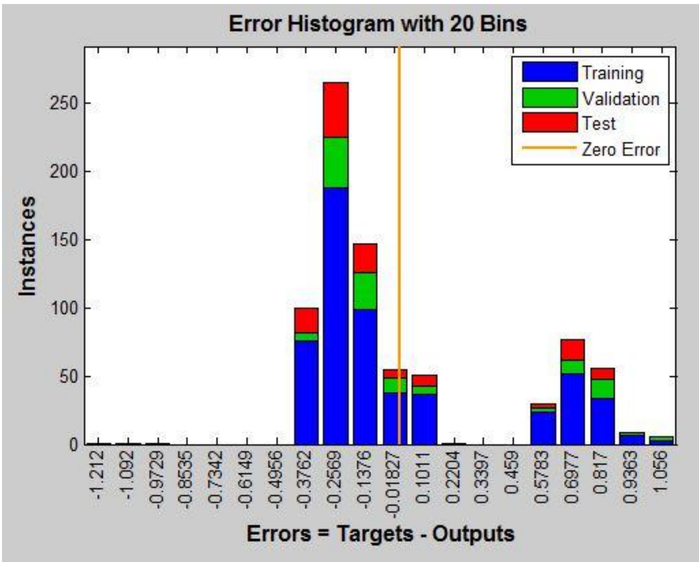

Figure 31: Error histogram of Gradient descent with momentum back-propagation

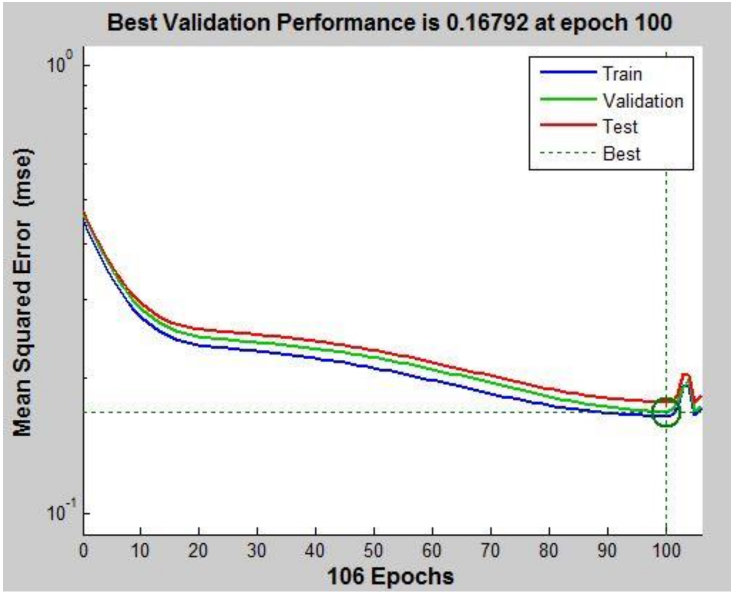

Figure 32: Best validation of Gradient descent w/momentum & adaptive lr back-propagation

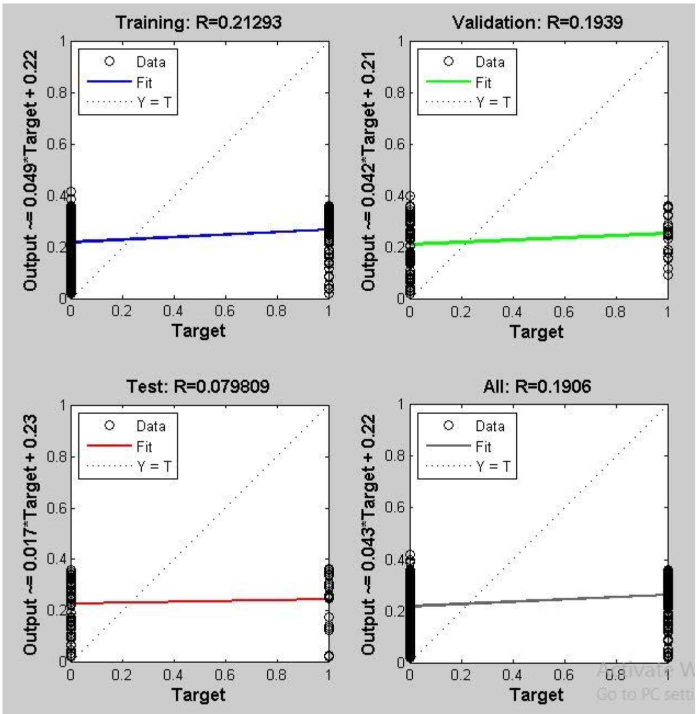

Figure 33: Regression of Gradient descent w/momentum & adaptive lr back-propagation

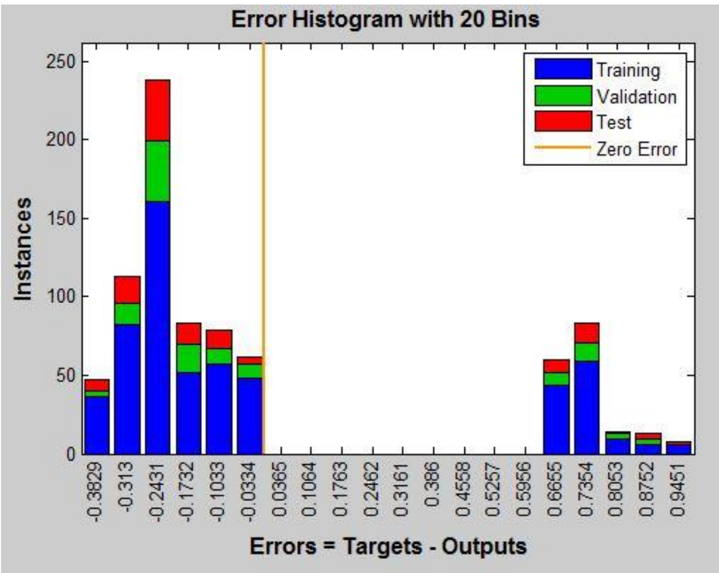

Figure 34: Error histogram of Gradient descent w/momentum & adaptive lr back-propagation

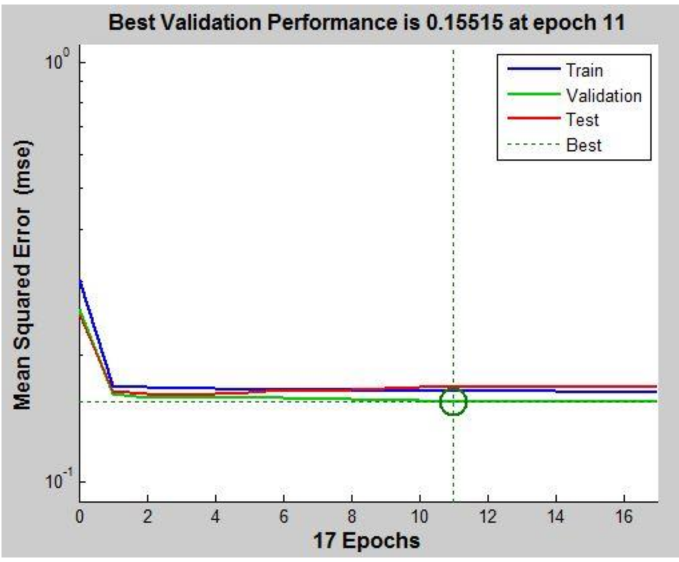

Figure 35: Best validation of Levenberg-Marquardt back-propagation

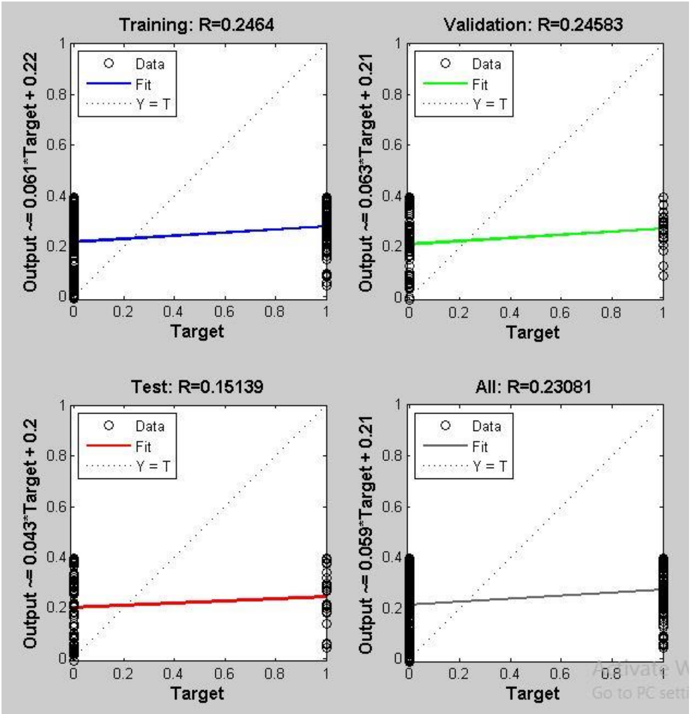

Figure 36: Regression of Levenberg-Marquardt back-propagation

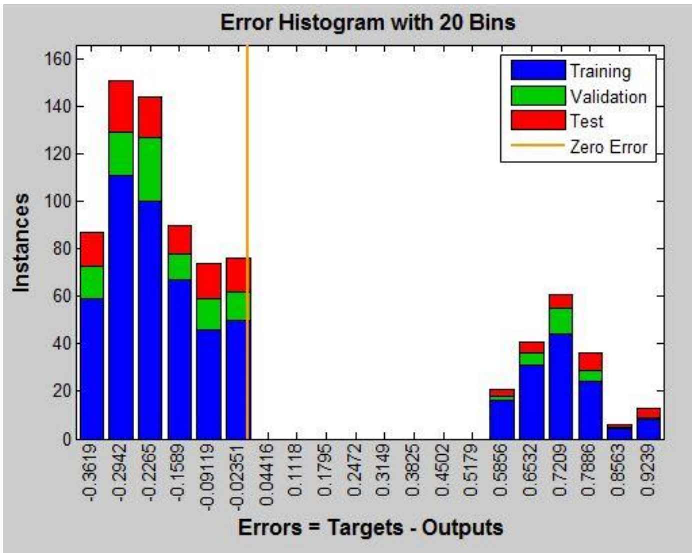

Figure 37: Error histogram of Levenberg-Marquardt back-propagation

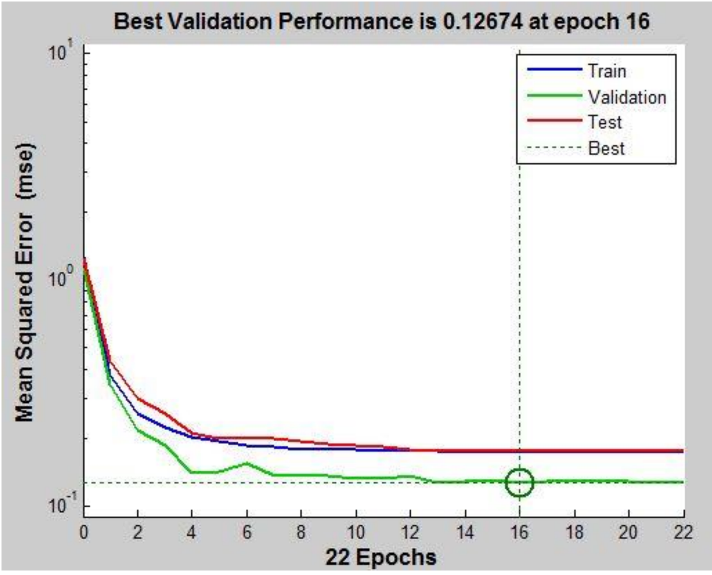

Figure 38: Best validation of one step secant back-propagation

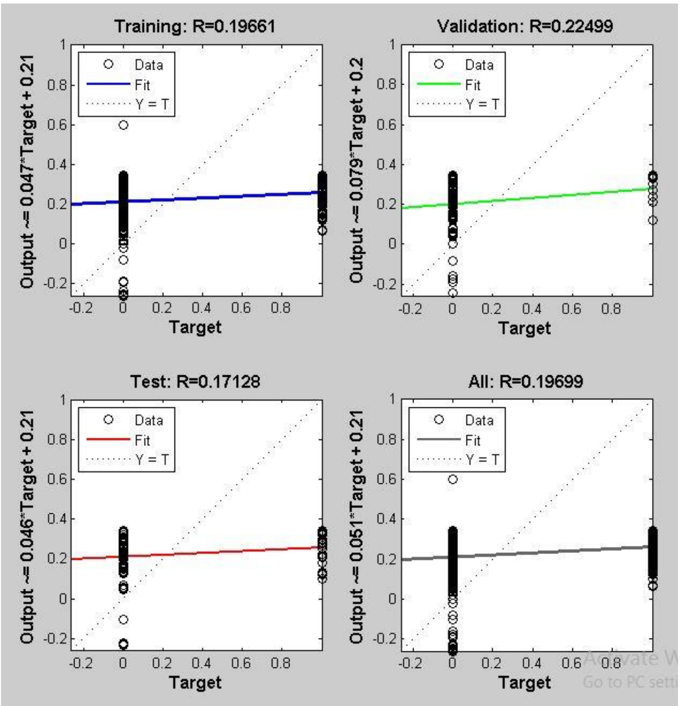

Figure 39: Regression of one step secant back-propagation

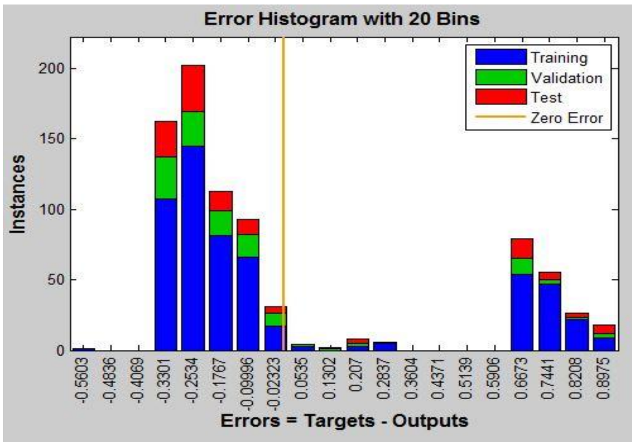

Figure 40: Error histogram of one step secant back-propagation

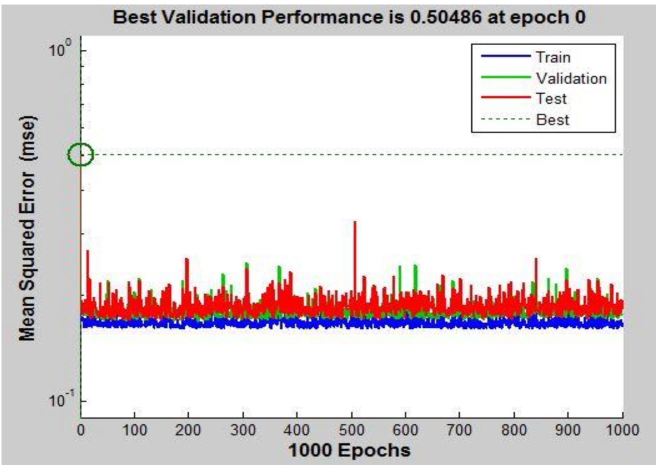

Figure 41: Best validation of Random order incremental training w/learning functions

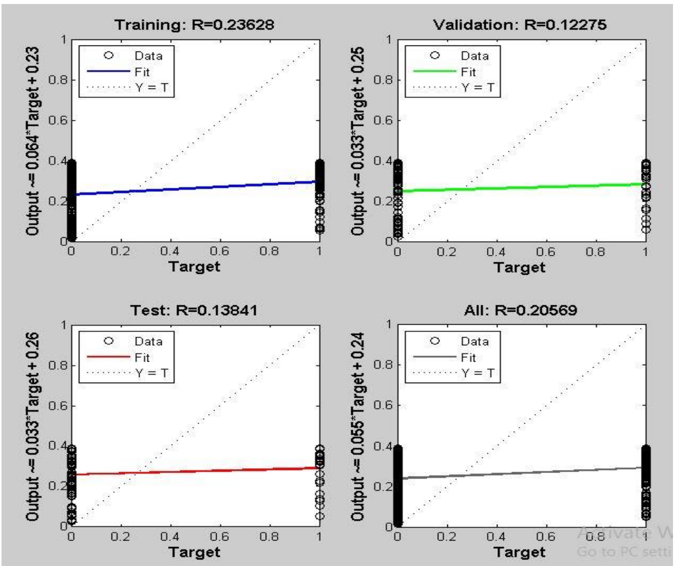

Figure 42: Regression of Random order incremental training w/learning functions

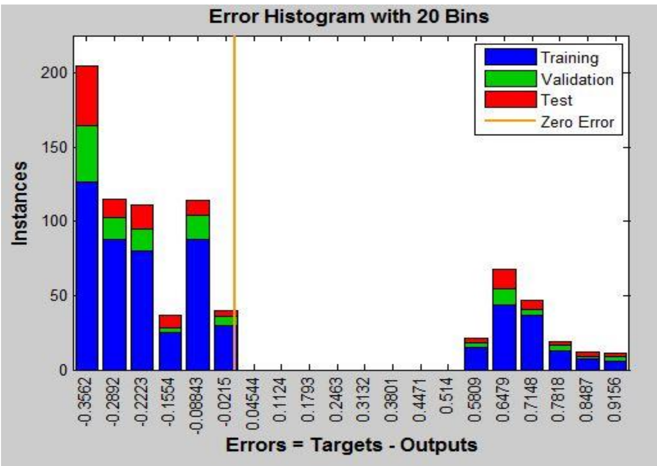

Figure 43: Error histogram of Random order incremental training w/learning functions

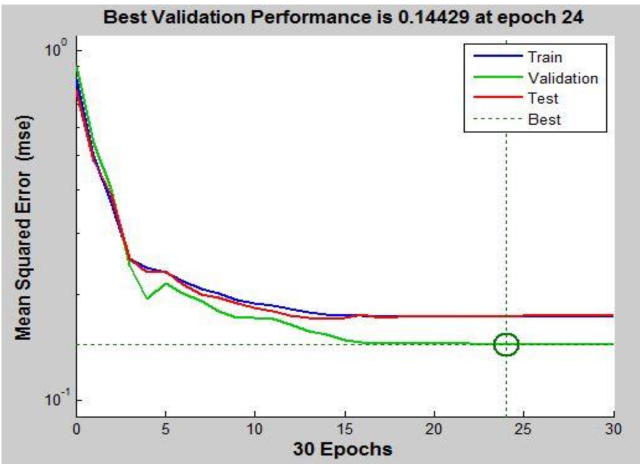

Figure 44: Best validation of Resilient back-propagation (Rprop)

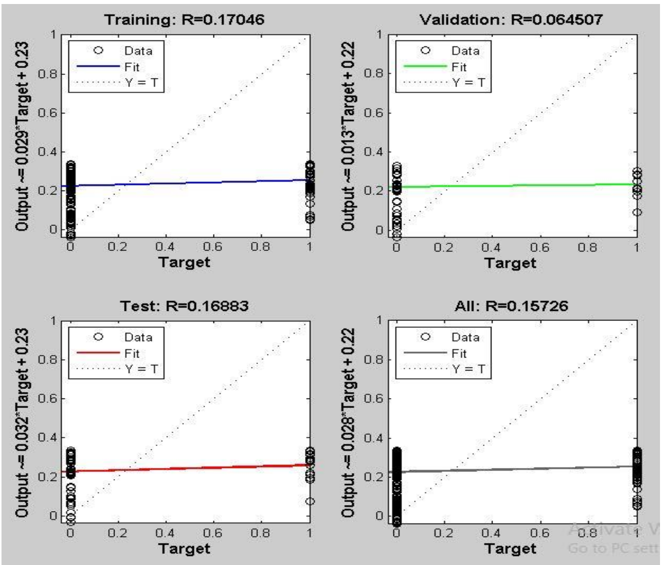

Figure 45: Regression of Resilient back-propagation (Rprop)

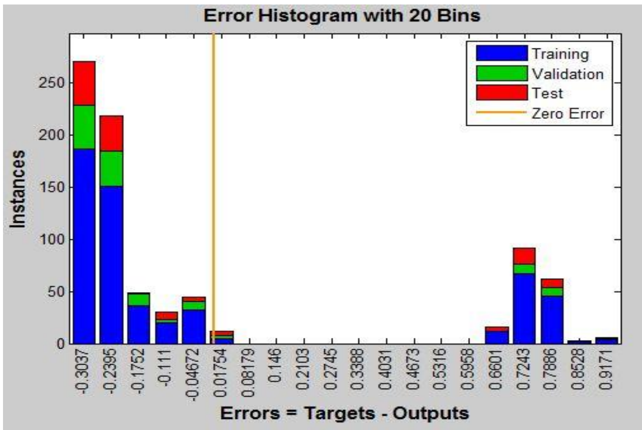

Figure 46: Error histogram of Resilient back-propagation (Rprop)

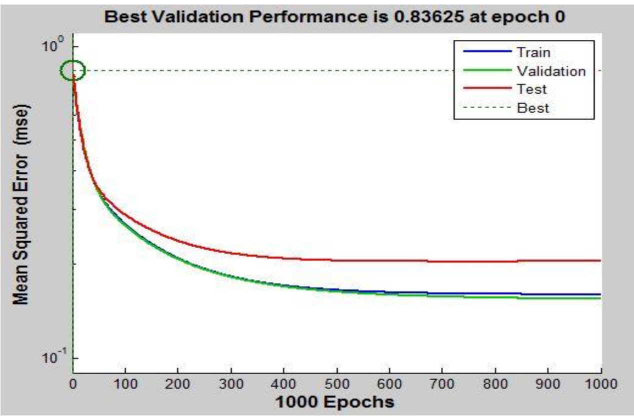

Figure 47: Best validation of Sequential order incremental training w/learning functions

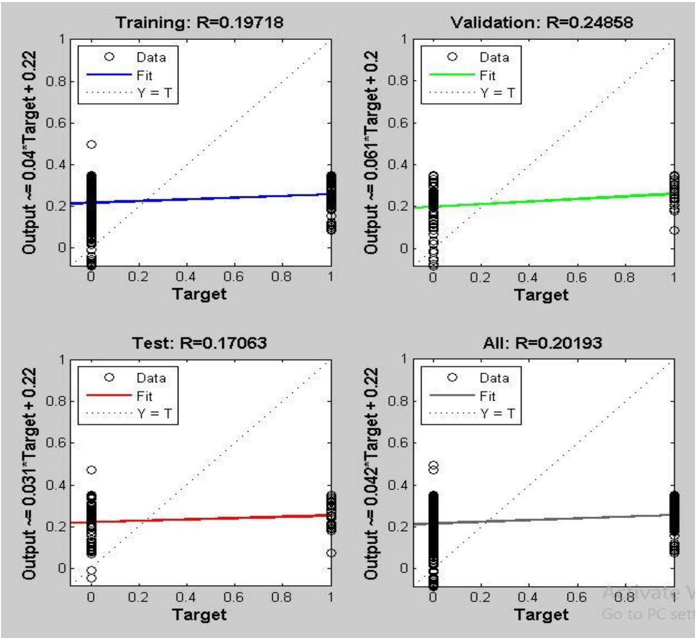

Figure 48: Regression of Sequential order incremental training w/learning functions

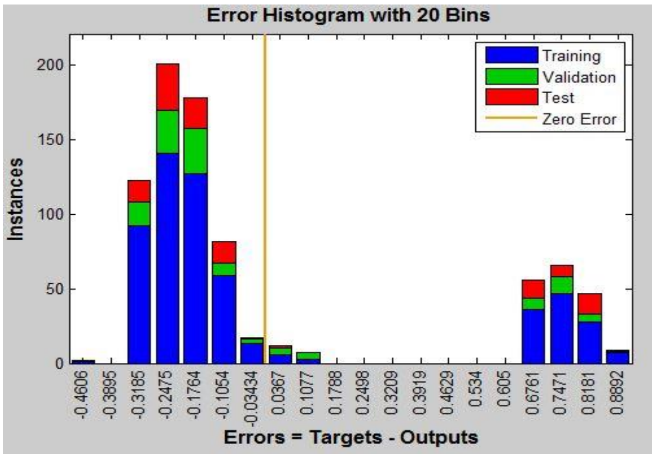

Figure 49: Error histogram of Sequential order incremental training w/learning functions

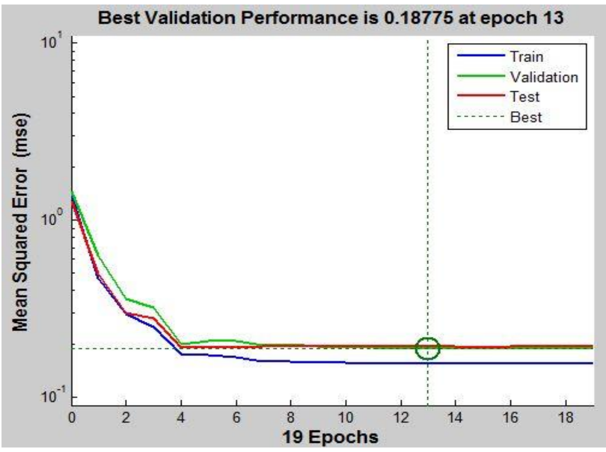

Figure 50: Best validation of Scaled conjugate gradient back-propagation

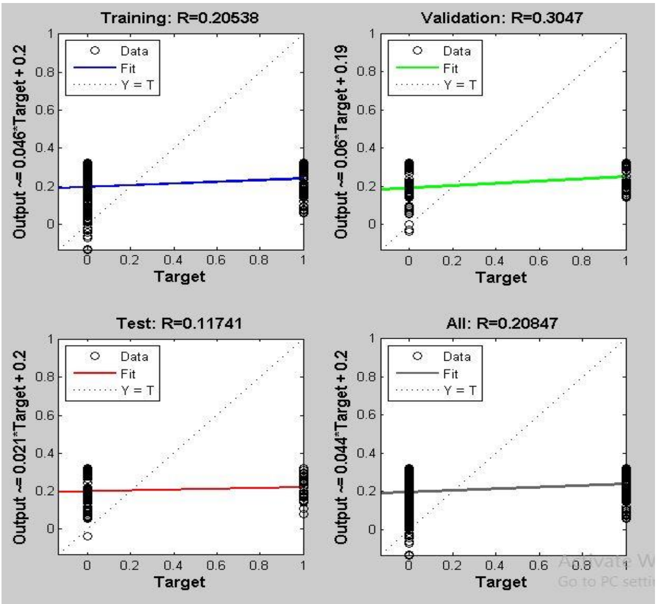

Figure 51: Regression of Scaled conjugate gradient back-propagation

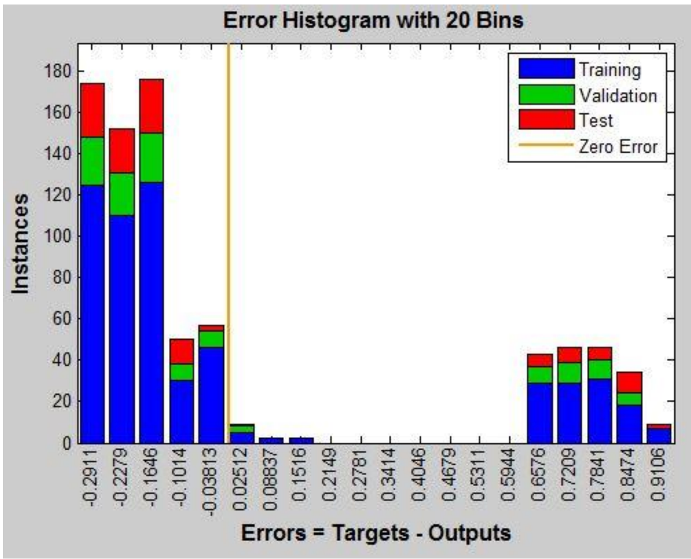

Figure 52: Error histogram of Scaled conjugate gradient back-propagation

Table 2: Summary of three parameters of various back-propagation algorithms

	Iterations	Performance	Best validation
trainb	1000	0.173	0.12848
trainbfg	21	0.158	0.1753
trainbr	97	132	131.5809
trainc	1000	0.164	2.2081
traincgb	19	0.168	0.16065
traincgf	24	0.161	0.17428
traincgp	9	0.169	0.15233
traingd	1000	0.185	0.21823
traingda	120	0.17	0.18319
traingdm	1000	0.167	0.21451
traingdx	106	0.165	0.16792
trainlm	17	0.164	0.15515
trainoss	22	0.173	0.12674
trainr	1000	0.162	0.50486
trainrp	30	0.173	0.14429
trains	1000	0.16	0.83625
trainscg	19	0.155	0.18775

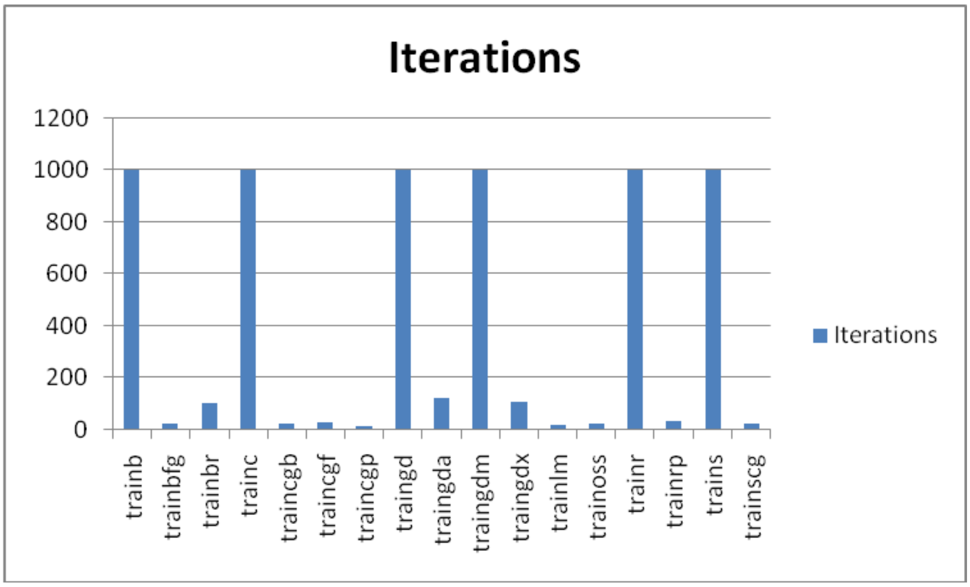

Figure 53: Iterations required for training of various back-propagation algorithms

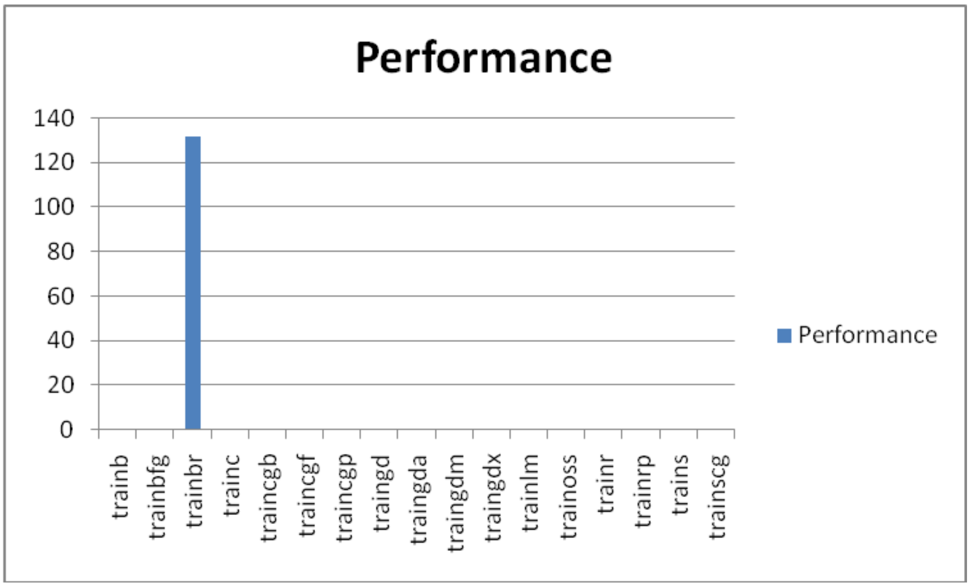

Figure 54: Performance statistics of various back-propagation learning algorithms

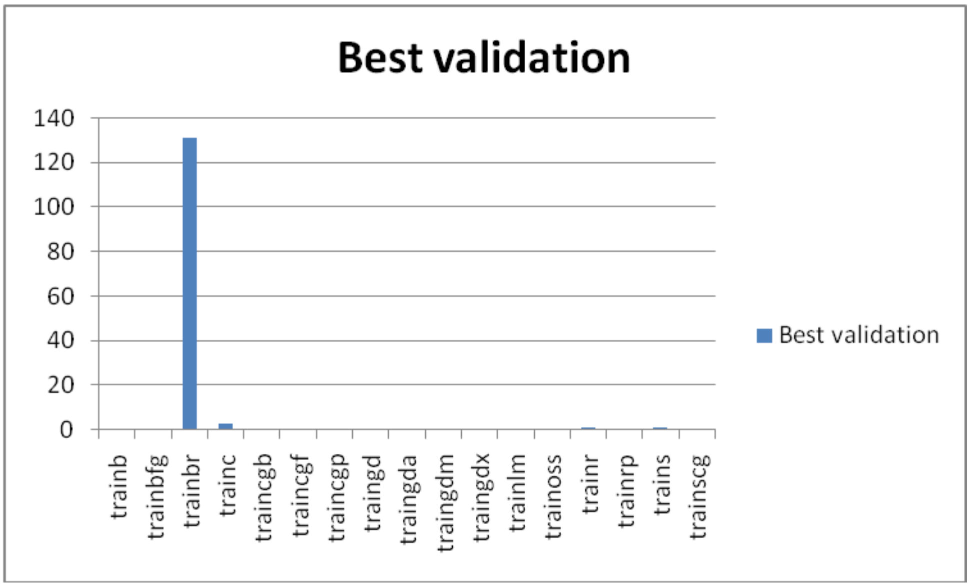

Figure 55: Best validation results of various back-propagation learning algorithms

4. Conclusion

Among all the back-propagation learning algorithms, Polak-Ribiere conjugate gradient back-propagation has shown the least number of iterations executed for training. Immediately after that Levenberg-Marquardt back-propagation has shown next minimum iterations executed for training. Except Bayesian regularization, all other back-propagations have close mean-square errors and best validation performance. Since the algorithms have been checked for same dataset for training, Polak-Ribiere conjugate gradient back-propagation and Levenberg-Marquardt back-propagation can be used for any experiment and research. Testing results showed 22% accuracy all algorithms. The activation function at the output layer was 'purelin'. Instead of that 'hardlim' could have been different result. This is the next plan of research.

Acknowledgements

We would like to thank our friends who have inspired us to gather knowledge about neural network.

References

[1]. Kapil Nahar, Artificial Neural Network, Compusoft, International Journal of Advanced Computer Technology, Vol. 1, No. 1, 2012

[2]. Stuart J. Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, ISBN 9780136042594, 2010

[3]. Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016). Back-Propagation and Other Differentiation Algorithms, Deep Learning. MIT Press. pp. 200–220. ISBN 9780262035613.

[4]. Md. Ashek-Al-Aziz, Abdullah-Hil Muntakim, Md. Kawshik Ahmed, No Regular Behavior Pattern in Neural Network Execution – A Matlab Experience, International Journal of Computer Applications, Vol. 174, No. 19, February 2021