

No Regular Behavior Pattern in Neural Network Execution – A Matlab Experience

Md. Ashek-Al-Aziz
Associate Professor
University of Development
Alternative, Dhaka, Bangladesh

Abdullah-Hil Muntakim
Asst. Prof. & Asst. Dir.
University of Development
Alternative, Dhaka, Bangladesh

Md. Kawshik Ahmed
Lecturer
University of Development
Alternative, Dhaka, Bangladesh

ABSTRACT

Neural Networks are designed and simulated using Matlab and performance of the networks are computed by Matlab simulation software. In this research work, 11 number of neural networks with different number of hidden layer (from 0 to 10) with two neurons in each hidden layer designed and simulated with XOR gate input, trained by XOR gate output target values and performances have been checked by using Matlab simulation software. No regular behavior pattern has been found in those performances of different neural network for XOR gate operation.

General Terms

Neural Network

Keywords

Neural network, Feed-forward, Back-propagation, XOR gate, Training

1. INTRODUCTION

Neural network is an artificial mimic of the human brain [1]. Using artificial neural networks, it is possible to predict future expectations, classify data values into groups which enable decisions to be made using artificially intelligent machines [2, 3]. Like human beings, artificial neural networks are trained by supervisors as per supervised learning paradigm [4, 5]. Feed forward and back-propagation is the usual process to train an artificial neural network. Neural networks receive inputs in input terminals, inputs are multiplied by random weight values, goes through activation functions and tempered by bias values, hence calculated output is obtained – this is the process of feed forward computation [6]. As per supervised learning paradigm, the supervisor knows the target output for each and every corresponding input or input combinations, weight values are adjusted by back-propagation learning rule [7]. Such iteration is repeated until the network reaches a condition where the difference between the computed output and target output is sufficiently minimized [8]. It is not irrelevant to mention that an artificial neural network usually has input terminals in input layer, output node(s) in output layer and some other neurons consist of hidden layer(s) [9]. Number of hidden layers is completely arbitrary, even the number of neurons in each layer. An artificial neural network with no hidden layer is a simple perceptron [10] and it is a linear classifier.

Exclusive OR gate in short XOR gate is a member of basic gates of digital logic. XOR gate produces output 1 only when the inputs are different e.g. 0 and 1 or 1 and 0 for two input XOR gates. Otherwise it produces 0 for the same inputs of 0 and 0 or 1 and 1 [11]. If it is subject to classify the XOR gate principle into two groups among which one group producing 1 output and another group producing 0 outputs, it is said that

such classification is not achievable by any linear classifier rather it requires non-linear classification [12] for which hidden layer in the neural network is essential. Classification is not a motivation in this research work rather it is subject to check the correct results production using different perceptron models and find out common characteristics in training/learning process.

The objective of this research work is to check the behavior pattern of neural networks with different numbers of hidden layers in the training aspect using XOR gate principle and deduce a conclusion.

2. METHOD

Neural Network model:

- Two input terminals in input layer
- One neuron in output layer with bias (bias is not shown in figure)
- n number of hidden layers, n=0,1,2,3,4,5,6,7,8,9,10
- Each hidden layer consists of two neurons (considered as constant for varying hidden layers), each neuron has bias (bias is not shown in figure)
- All activation functions are same, purelin (a linear activation function with threshold zero)

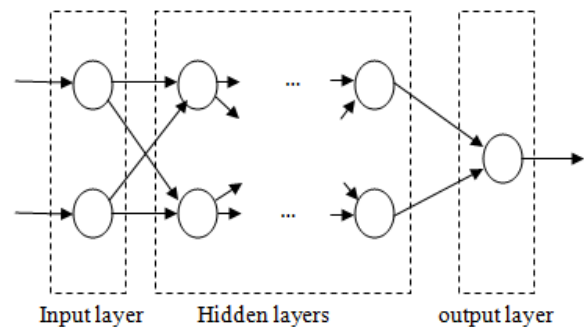


Fig 1: Neural Network model with hidden layers

XOR gate:

Table 1: Truth table of XOR gate

Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

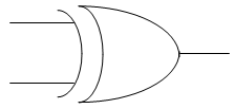


Fig 2: XOR gate symbol

3. EXPERIMENT

A computer machine with Intel Pentium P 6200, 2.13 Gigahertz Microprocessor and 2 Gigabyte DDR 3 RAM is used in this experiment. Also Matlab Version 7.12.0.635 (R2011a) has been used for neural network simulation. The inputs and outputs of XOR gate are entered in transposed matrix form and neural networks have been simulated using nntool of Matlab.

- Net 0: Neural network without any hidden layer, a simple perceptron
- Net 1: One input layer, one output layer with one hidden layer
- Net 2: One input layer, one output layer with two hidden layers
- Net 3: One input layer, one output layer with three hidden layers
- Net 4: One input layer, one output layer with four hidden layers
- Net 5: One input layer, one output layer with five hidden layers
- Net 6: One input layer, one output layer with six hidden layers
- Net 7: One input layer, one output layer with seven hidden layers
- Net 8: One input layer, one output layer with eight hidden layers
- Net 9: One input layer, one output layer with nine hidden layers
- Net 10: One input layer, one output layer with ten hidden layers

All activation functions: Purelin (A linear activation function)

Training method: Levenberg-Marquardt
Performance: Mean Squared Method

Fig 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 and 23 depict the performance result of the neural network with corresponding varying number of hidden layers. Similarly, Fig 4, 6, 8, 10, 12, 14, 16, 18, 20, 22 and 24 depict regression lines. Results are summarized in Table 2. Fig 25 depicts the curve fitting for best validation performance versus number of hidden layers i.e. relationship between best performance and number of hidden layers. It is seen that the relationship is sinusoidal (non-linear) but if it is considered for linear relationship then it $y = -0.062x + 0.751$ where y : best validation performance and x : number of hidden layers. Fig 26 shows the number of iterations executed for different number of hidden layers. Fig 27 depicts a comparison between testing accuracy and neural networks with different number of hidden layers.

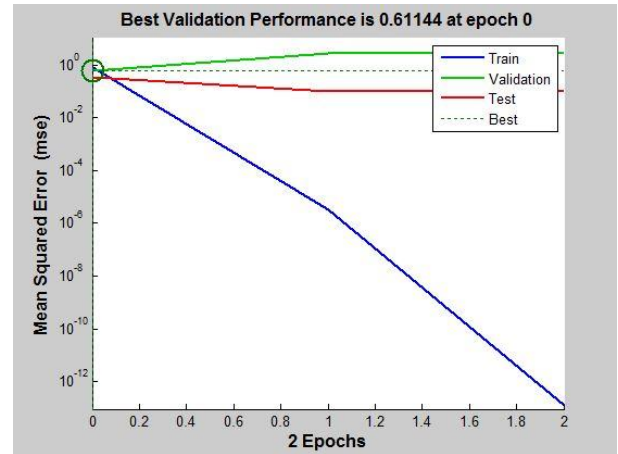


Fig 3: Performance of Net 0

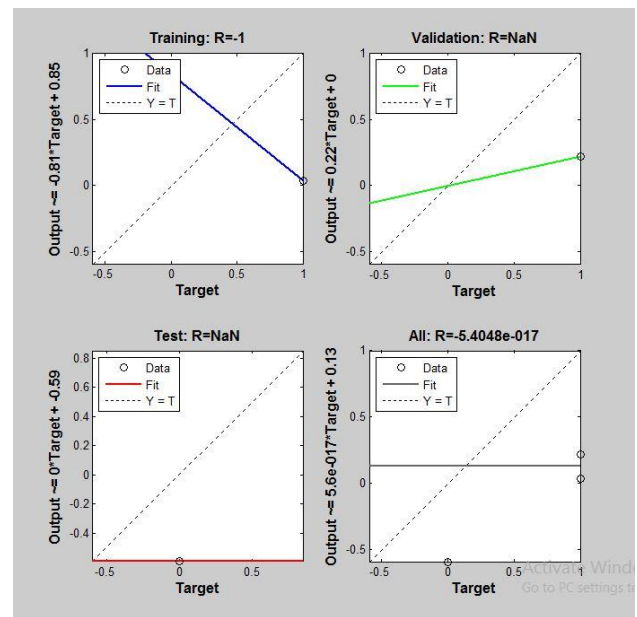


Fig 4: Regression line of Net 0

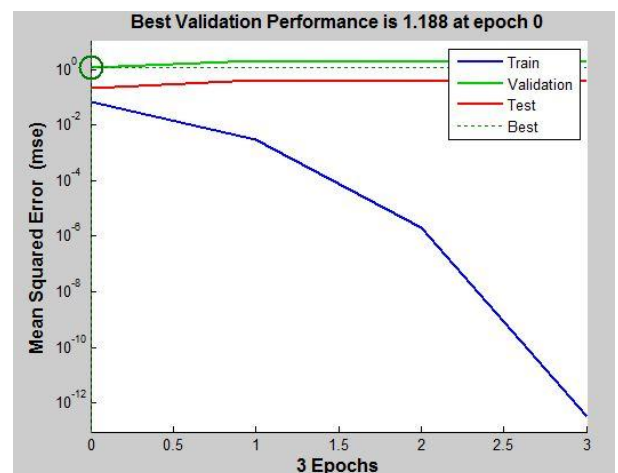


Fig 5: Performance of Net 1

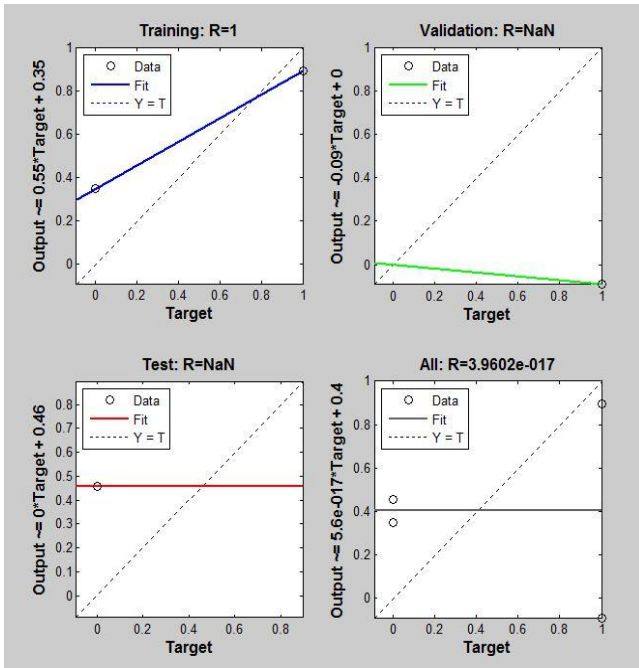


Fig 6: Regression line of Net 1

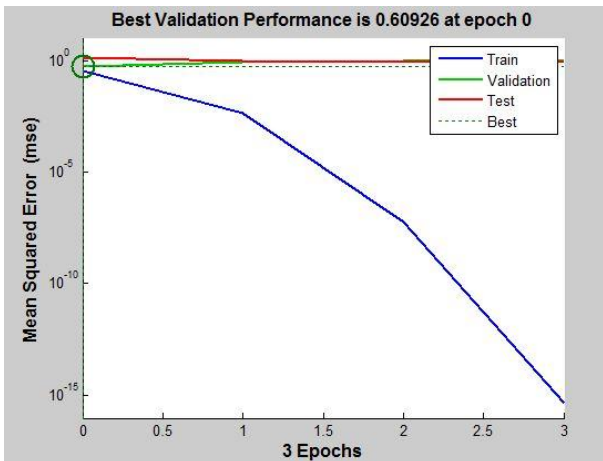


Fig 7: Performance of Net 2

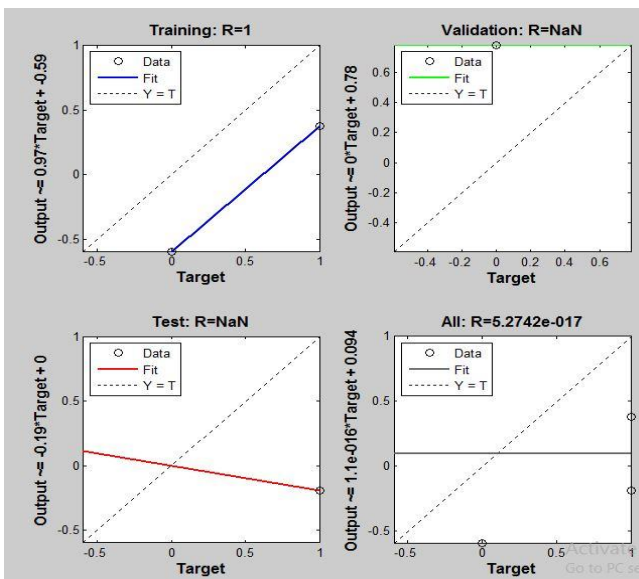


Fig 8: Regression line of Net 2

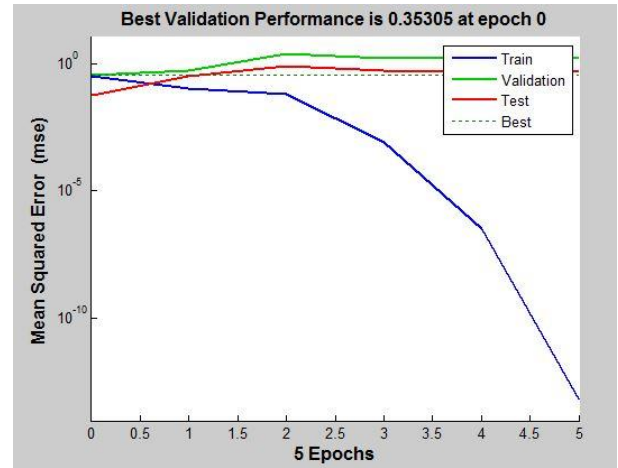


Fig 9: Performance of Net 3

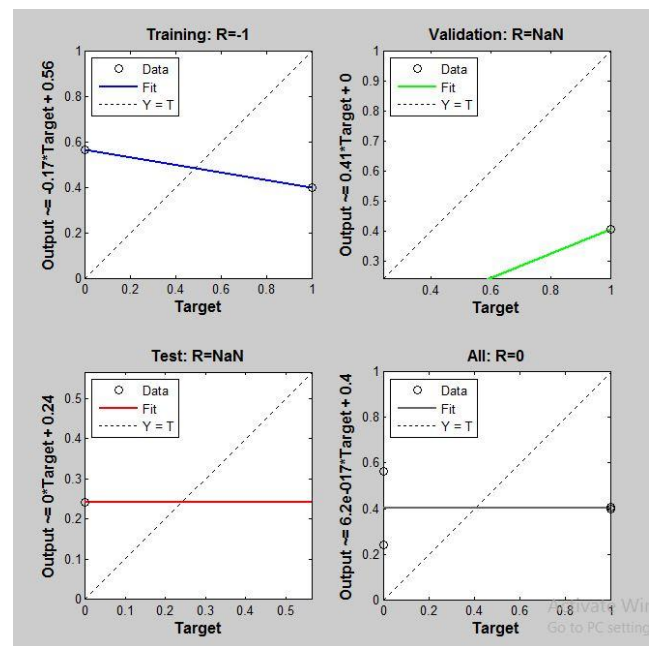


Fig 10: Regression line of Net 3

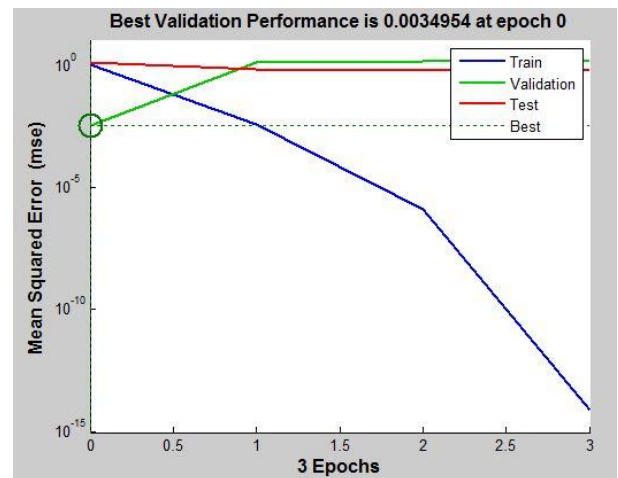


Fig 11: Performance of Net 4

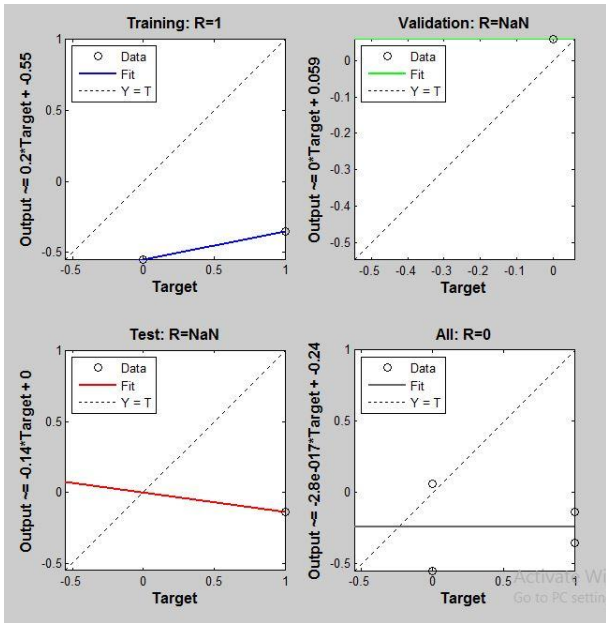


Fig 12: Regression line of Net 4

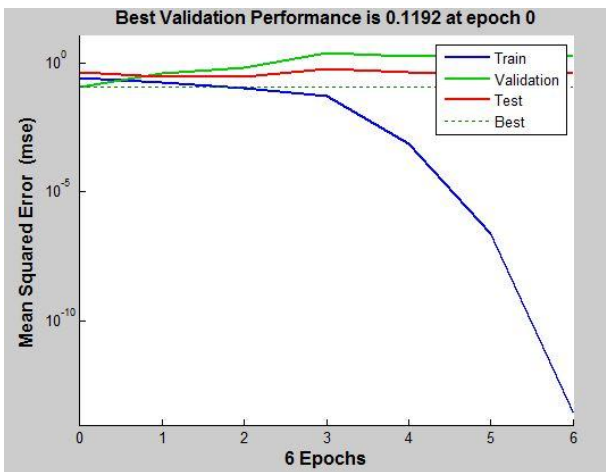


Fig 13: Performance of Net 5

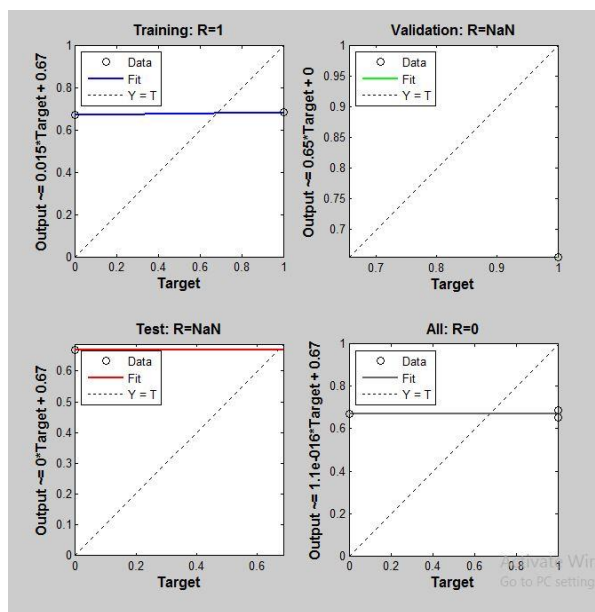


Fig 14: Regression line of Net 5

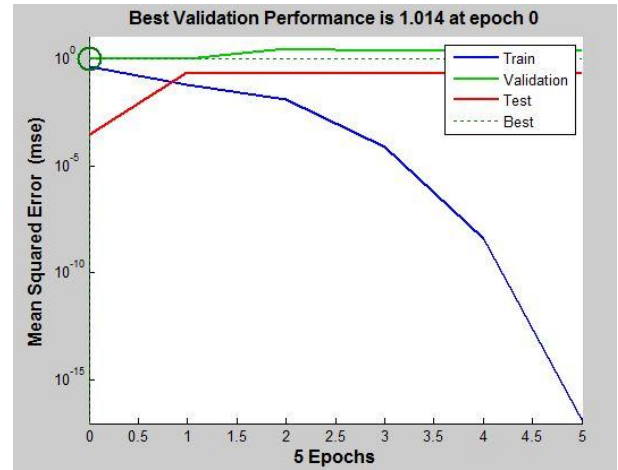


Fig 15: Performance of Net 6

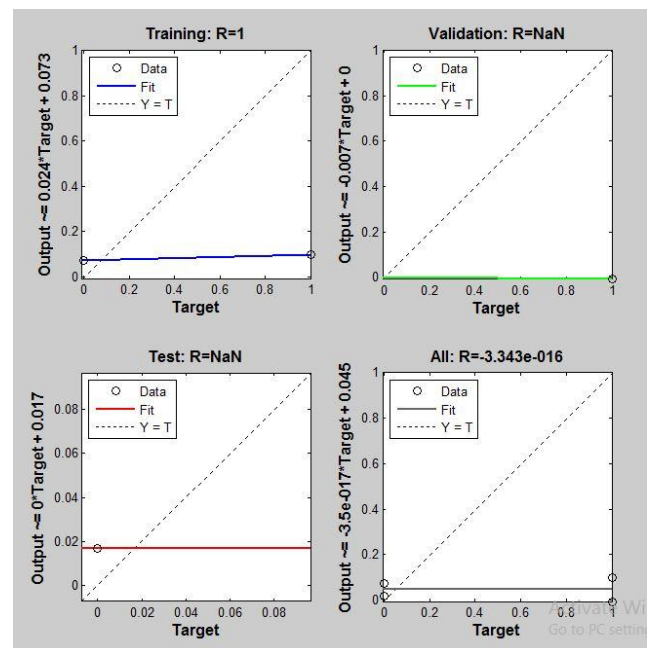


Fig 16: Regression line of Net 6

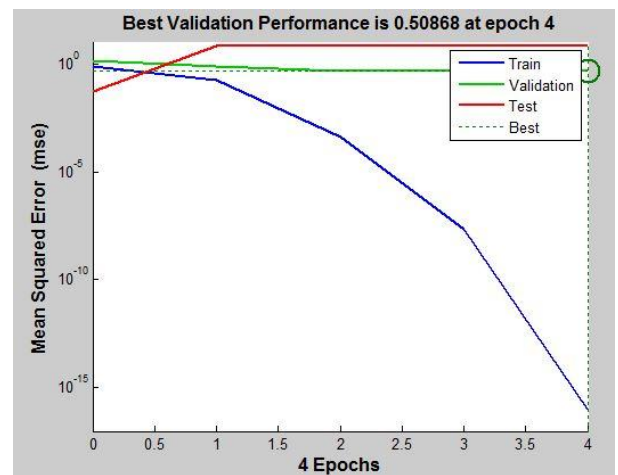


Fig 17: Performance of Net 7

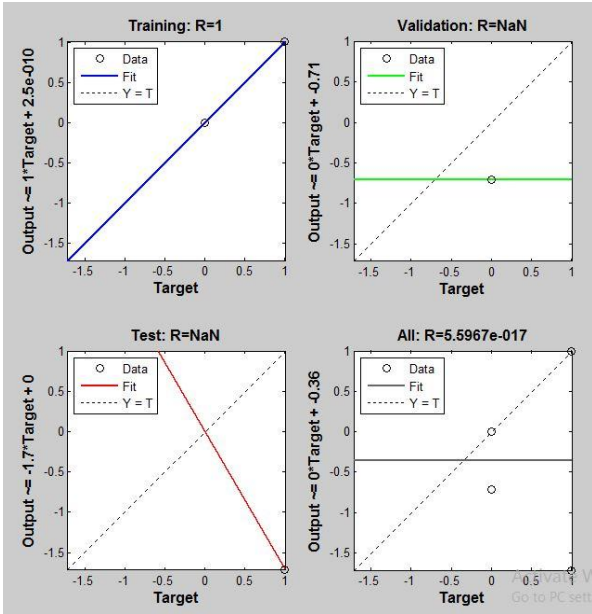


Fig 18: Regression line of Net 7

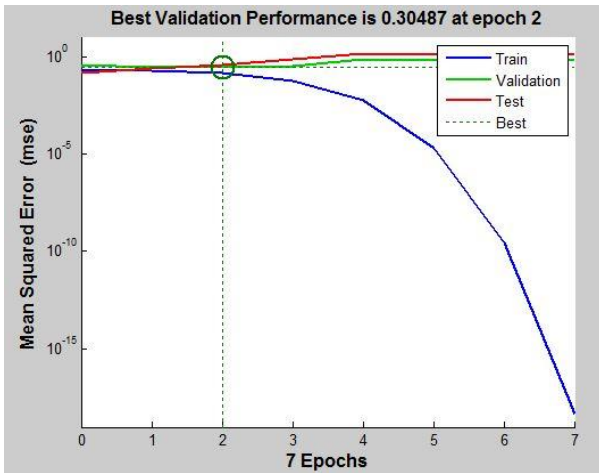


Fig 19: Performance of Net 8

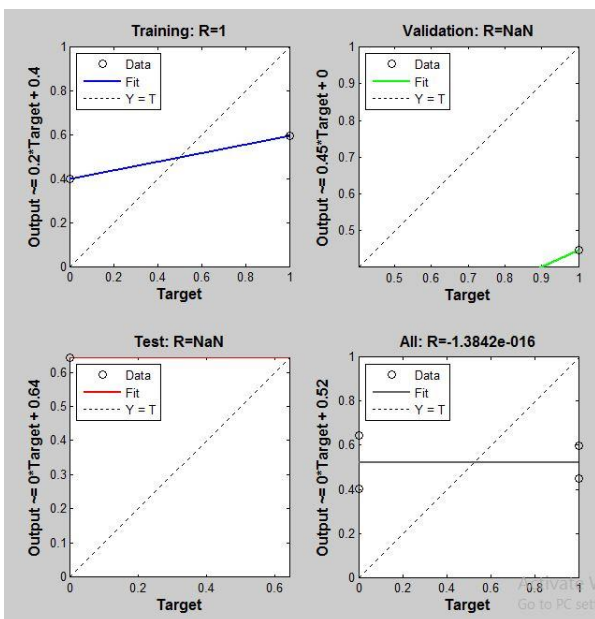


Fig 20: Regression line of Net 8

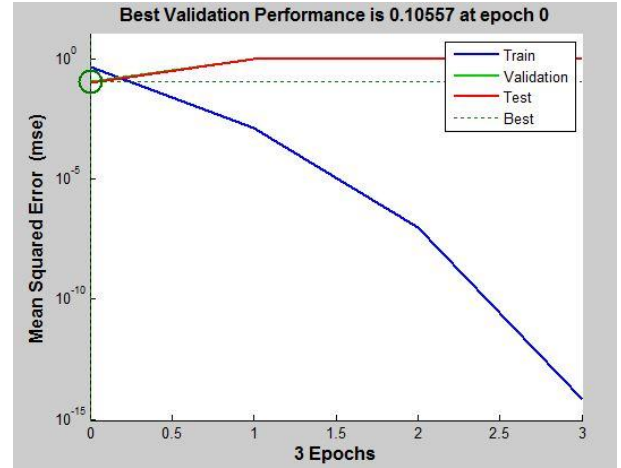


Fig 21: Performance of Net 9

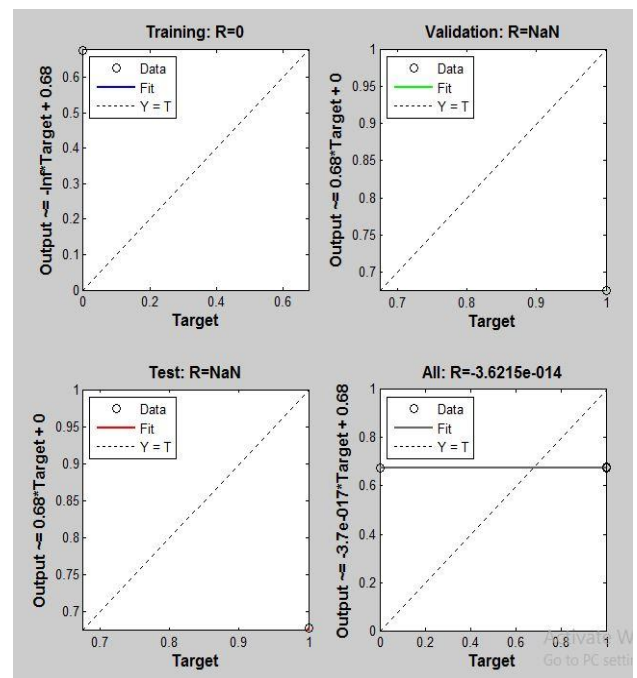


Fig 22: Regression line of Net 9

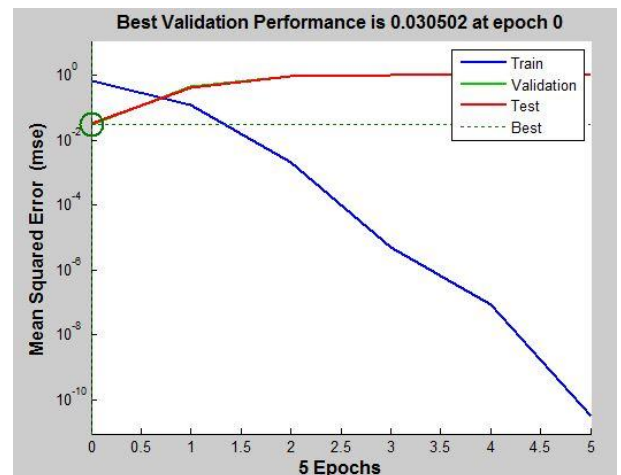


Fig 23: Performance of Net 10

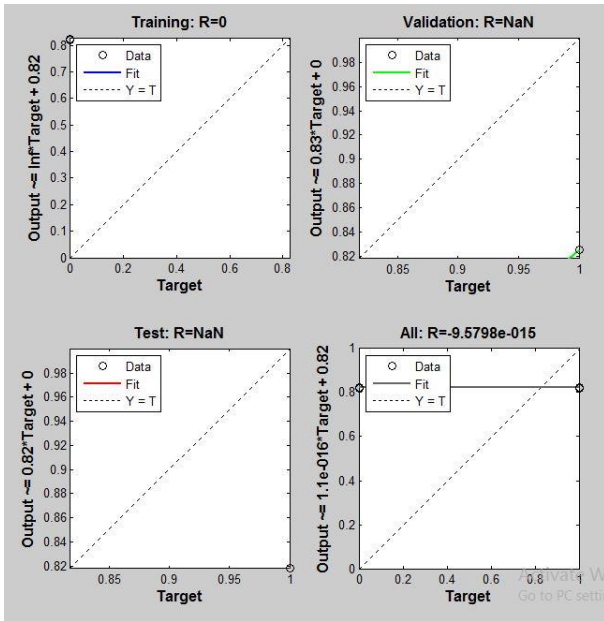


Fig 24: Regression line of Net 10

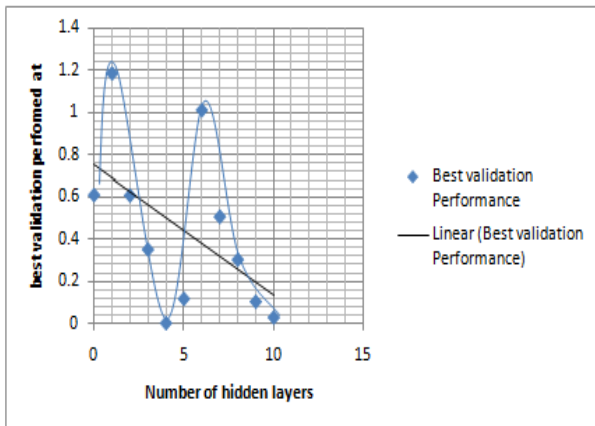


Fig 25: Relationship between best performance and number of hidden layers

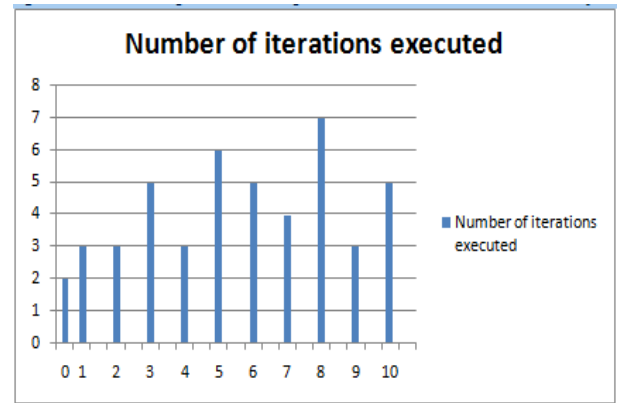


Fig 26: Number of iterations executed for different hidden layers (0-10)

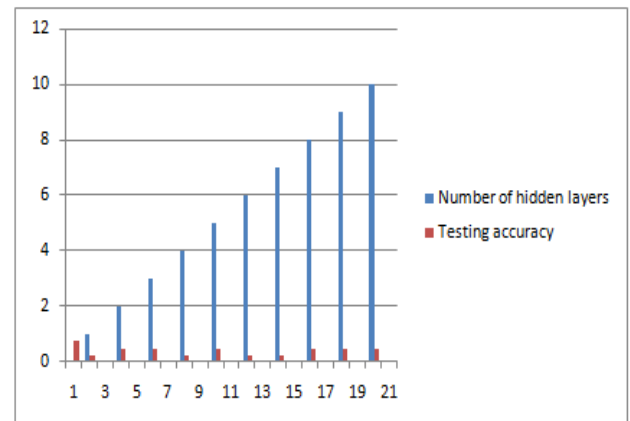


Fig 27: Comparison of Testing Accuracy of different neural network

4. CONCLUSION

Table 2 holds the whole picture of the experiment. Few analyses of the experimental data have been carried out. It has been tried to extract relationships between different neural networks of various numbers of hidden layers and best validation performances. A sinusoidal curve has been with downward tendency indeed Microsoft Excel auto detect a linear curve $y = -0.062x + 0.751$ with huge dispersion of data distant apart vertically from the line ($R^2 = 0.273$). There is no relationship established between 11 different neural networks with varying number of hidden layers from 0 to 10 and number of iterations executed. Even more, there is no pattern found in increment and decrement of testing accuracy for such different neural networks with different numbers of hidden layers (0-10). However, it can be concluded that there is no regular pattern in the behavior of neural networks for varying numbers of hidden layers and their performances.

Table 2: Summary of Training parameters of neural network

Neural Networks	Number of hidden layers	Output values	Weight values	Error values	Best validation Performance	Number of iterations executed	Testing accuracy
Net 0	0	[-0.59353 0.21806 0.035919 0.8475]	[0.62945 0.81158]	[0.59353 0.78194 0.96408 - 0.8475]	0.61144 (at epoch 0)	2	75%
Net 1	1	[0.34851 - 0.089932 0.89529 0.45685]	[0.093763 0.92978; 0.91501 - 0.68477]	[-0.34851 1.0899 0.10471 - 0.45685]	1.188 (at epoch 0)	3	25%
Net 2	2	[-0.59311 0.38032 - 0.19287 0.78055]	[0.62945 - 0.74603; 0.81158 0.82675]	[0.59311 0.61968 1.1929 - 0.78055]	0.60926 (at epoch 0)	3	50%
Net 3	3	[0.56414 0.40582 0.39852 0.2402]	[-0.49783 - 0.053422; 0.23209 - 0.29668]	[-0.56414 0.59418 0.60148 - 0.2402]	0.35305 (at epoch 0)	5	50%
Net 4	4	[0.059122 - 0.35188 - 0.13618 - 0.54719]	[-0.37757 - 0.6687; 0.057066 0.20396]	[-0.059122 1.3519 1.1362 0.54719]	0.0034954 (at epoch 0)	3	25%
Net 5	5	[0.67165 0.65475 0.6863 0.6694]	[-0.64825 - 0.053028; 0.44352 - 0.69456]	[-0.67165 0.34525 0.3137 - 0.6694]	0.1192 (at epoch 0)	6	50%
Net 6	6	[0.072559 0.096408 - 0.006968 0.016881]	[0.55143 - 0.12828; -0.026417 - 0.10643]	[-0.072559 0.90359 1.007 -0.016881]	1.014(at epoch 0)	5	25%
Net 7	7	[2.5074e-010 1 -1.7132 - 0.71322]	[-0.58076 0.27653; -0.79537 0.50867]	[-2.5074e-010 -1.3485e-008 2.7132 0.71322]	0.50868 (at epoch 4)	4	25%
Net 8	8	[0.64299 0.44785 0.59682 0.40168]	[-0.32622 - 0.67086; 0.39846 - 0.95136]	[-0.64299 0.55215 0.40318 - 0.40168]	0.30487 (at epoch 2)	7	50%
Net 9	9	[0.67593 0.67721 0.67508 0.67636]	[-0.96903 - 0.66566; 0.96813 - 0.78757]	[-0.67593 0.32279 0.32492 - 0.67636]	0.10557 (at epoch 0)	3	50%
Net 10	10	[0.8236 0.81808 0.82535 0.81983]	[-0.47854 - 0.95497; 0.18871 - 0.14948]	[-0.8236 0.18192 0.17465 - 0.81983]	0.030502 (at epoch 0)	5	50%

5. REFERENCES

- [1] Nahar K., Artificial Neural Network, An international journal of advanced computer technology (Compusoft), Vol. 1, No. 2, Dec 2012
- [2] Spangler W. E., Jerrold H. May, And Luis G. Vargas, Choosing Data-Mining Methods for Multiple Classification: Representational and Performance Measurement Implications for Decision Support, Journal of Management Information Systems, , Vol. 16, No. I, pp. 31-62, r 1999
- [3] Garcia-Laencina P. J., Sancho-Gomez J. L. and Figueiras-Vidal A. R., Pattern classification with missing data: a review, Neural Computing & Applications, Springer, Vol. 19, pp 263–282, 2010
- [4] Zhou Z. H., A brief introduction to weakly supervised learning, National Science Review, Volume 5, No 1, pp 44–53, January 2018

- [5] Schwenker F. and Trentin E., Pattern classification and clustering: A review of partially supervised learning approaches, *Pattern Recognition Letters*, Elsevier, Vol. 37, pp 4-14, February 2014
- [6] Bebis G. and Georgiopoulos M., Feed-forward neural networks, *IEEE Potentials*, Vol. 13 No. 4, Oct.-Nov. 1994
- [7] Svozil D., Kvasni Eka V., Pospichal J, Introduction to multi-layer feed-forward neural networks, *Chemometrics and Intelligent Laboratory Systems*, Elsevier, Vol. 39, pp 43-62, 1997
- [8] Rojas R., *The Backpropagation Algorithm*, Neural Networks, Springer, pp 149-182, 1996
- [9] Mehrotra K., Mohan C. K., Sanjay Ranka, *Elements of Artificial Neural Networks*, Penram International Publishing (India), Vol. 1, pp 1-41, 1997
- [10] Pitts W. and McCulloch W. S., How we know universals: The perceptron of auditory and visual forms, *Bulletin of Mathematical Biophysics*, Vol. 9, pp 127-147, 1947
- [11] Fletcher, William, *An engineering approach to digital design*. Prentice-Hall. p. 98, 1980.
- [12] Rajasekaran S., Vijayalakshmi Pai G. A., *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, Prentice-Hall of India Private Limited, p. 25, 2007.