



OBSERVATION ON TRAINING NEURAL NETWORK FOR DIAGNOSING SCHIZOPHRENIA

Moumita Paul

School of Bioscience and Engineering
Jadavpur University
Kolkata, India - 700032

Dr. Monisha Chakraborty

Associate Professor, School of Bioscience and Engineering
Jadavpur University
Kolkata, India-700032

Abstract: Artificial neural networks may be deployed to diagnose illnesses including mental illness. One such mental illness is schizophrenia which is characterized by persistent delusions, hallucinations, disorganized speech, highly disorganized or catatonic behaviour and negative symptoms. It is commonly believed that one or two hidden layers in a neural network are sufficient to classify data, and that more hidden layers may be avoided because of longer times taken for the network to converge. However, we demonstrate that beyond a certain size of the hidden layer(s), it is harmful to deploy more than one layer because not only will it take longer for the network parameters to converge, the classification performance deteriorates sharply with more than one hidden layers.

Keywords: schizophrenia, artificial neural network, training, layers, validation, performance

I. INTRODUCTION

Schizophrenia is a serious mental disorder [1] that may be diagnosed by doing a psychometric test on the subject. The results of the test are captured on a thirty point scale known as the Positive and Negative syndrome Scale (PANSS) [2]. Originally, each item on the scale was assigned a possible rating in the range of one to seven; but because of certain disadvantages of the system, the rating is now assigned between zero to six. [3]

Artificial intelligence is increasingly being used to diagnose diseases like diabetes, chest disease and urological dysfunction [4] [5] [6] [7]. It may also be used to diagnose mental illness like schizophrenia. At the heart of the diagnostic system is the artificial neural network, which can classify subjects based on their thirty-dimensional PANSS ratings as either schizophrenic or otherwise. A simple perceptron may be deployed to classify data that is linearly separable, but in case the classes are not linearly separable, the data must be cast into a higher dimension [8]. This is where the multi-layer perceptron (MLP) comes in. The MLP has one input layer, one output layer and one or more hidden layers. The number of nodes in the input layer is equal to the dimension of the input data, which in our case is thirty; the number of nodes in the output layer is one, and the number of nodes in the hidden layer as well as the number of hidden layers may be varied to get optimum classification performance out of the MLP.

Researchers have worked towards establishing the ideal number of hidden layers as well as the ideal number of nodes in each hidden layer. If N_t is the number of training samples, N_i is the number of input nodes, N_h is the number of neurons in the hidden layer and N_o is the number of output nodes, then the various values of N_h may be arrived at by the following methods:

According to Li, Chow and Yu's method [9] [10] [11],

$$N_h = \frac{\sqrt{1 + 8N_i} - 1}{2} \quad (1)$$

According to Tamura and Tateishi's method [9] [10] [12],

$$N_h = N_i - 1 \quad (2)$$

According to Xu and Chen's method [9] [10] [13],

$$N_h = \frac{1}{2} \frac{N_t}{N_i \log N_t} \quad (3)$$

According to Shibata and Ikeda's method [9] [10] [14],

$$N_h = \sqrt{N_i N_o} \quad (4)$$

According to Sheela and Deepa's method [9] [10],

$$N_h = \frac{4 N_i^2 + 3}{N_i^2 - 8} \quad (5)$$

According to Trenn [9] [10],

$$N_h = \frac{(N_i + N_o - 1)}{2} \quad (6)$$

Apart from the above, there are several rules of thumb [15] [16], viz.

- Size of hidden layer must be between sizes of input layer and output layer.
- Size of hidden layer must be two thirds the size of the input layer plus the size of the output layer.
- Size of hidden layer must be less than twice the size of the input layer.

As for the number of the hidden layers, the common consensus is that one or two layers are adequate for most situations [15].

II. METHOD

We created and trained multi-layer perceptrons with the MATLAB neural network toolbox. We have on hand a set of 960 training samples. The training samples were generated synthetically with the help of a fuzzy expert system and some custom MATLAB code. Different PANSS readings were provided as inputs to the fuzzy expert system for diagnosing schizophrenia, and the output for each reading was noted. The

reading was categorized as typical of a schizophrenic or otherwise depending on the output which was assessed by a qualified psychiatrist. The MATLAB software randomly distributed these samples into training, validation and testing sets. The training algorithm used was Levenberg-Marquardt algorithm and mean square error was considered as the error criterion.

Various different models of neural networks with varying number of hidden nodes were created. First, the training was done on a neural network with a certain number of nodes in a single hidden layer. Then the training was repeated on a neural network with two hidden layers and the same number of nodes as above in each hidden layer.

The same training pairs were used to train all models of neural network.

III. RESULTS AND DISCUSSION

The validation error obtained in each training instance is tabulated below:

Table 1: Validation Performance with Various Numbers of Nodes in One and Two Hidden Layers

No. of nodes in each hidden layer	Best Validation Performance	
	No. of hidden layers = 1	No. of hidden layers = 2
4	5.3428e-15	4.0688e-15
5	1.2837e-18	5.4858e-15
7	6.7224e-16	8.7213e-16
15	2.3247e-15	3.018e-15
20	2.1164e-15	9.2449e-16
29	4.5492e-16	9.8195e-11
30	8.478e-16	6.7969e-16
35	7.5571e-16	4.4422e-16
40	2.362e-11	1.6613e-05
60	1.7206e-15	2.9723e-04
80	4.8873e-13	4.5297e-04
100	4.4468e-05	2.3377e-03

It is observed that if the number of neurons in the hidden layer is greater than 35, the validation performance deteriorates sharply if a second hidden layer is added. It is commonly believed that adding more hidden layers is overkill in the sense it does not improve performance. But our work has demonstrated that adding more hidden layers not only does not improve performance, it will cause the performance to decline sharply.

For example, the training performance graphs and error histograms for $N_h = 40$ are given below:

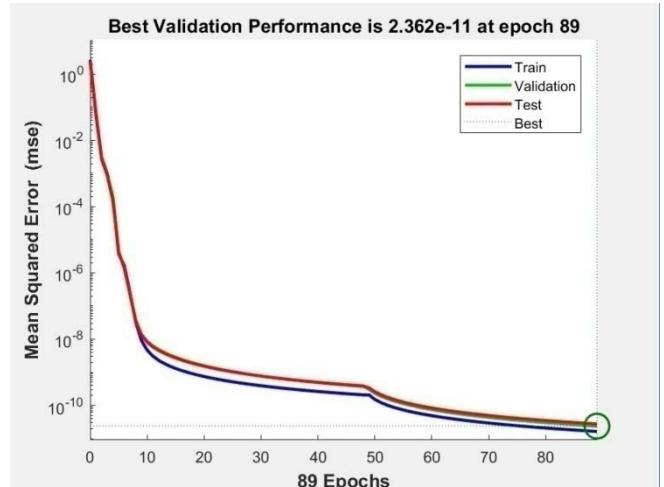


Figure 1: Validation Performance with One Hidden Layer Containing 40 neurons

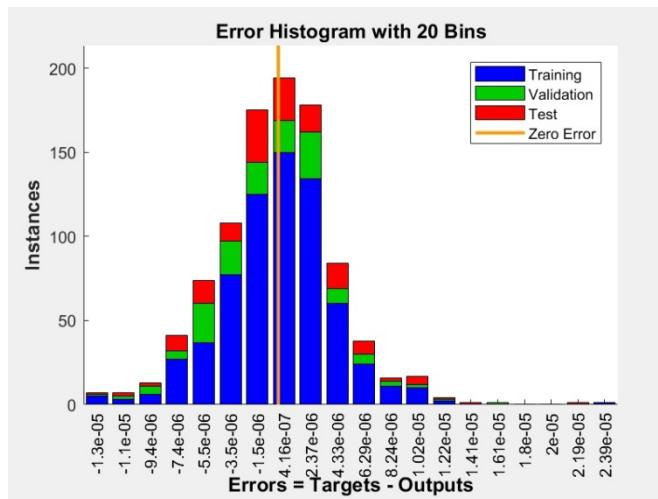


Figure 2: Error Histogram with One Hidden Layer Containing 40 Neurons

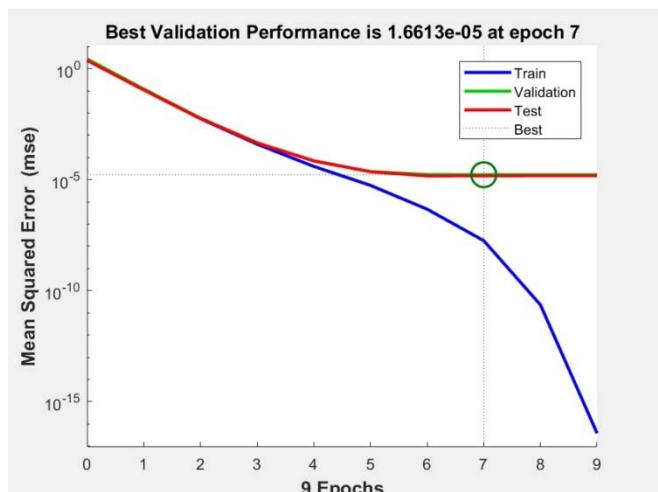


Figure 3: Validation Performance with Two Hidden Layers, Each Containing 40 Neurons

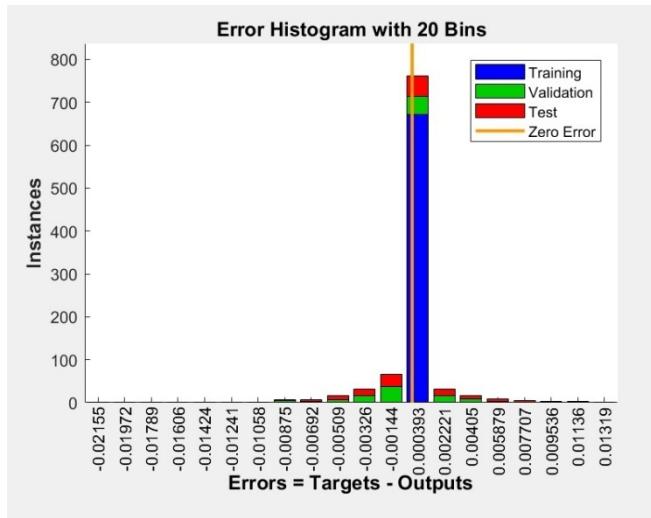


Figure 4: Error Histogram with Two Hidden Layers, Each Containing 40 Neurons

As can be observed from the figures above, the neural network's ability to generalize declines sharply if a second hidden layer is added and each hidden layer contains 40 neurons.

On the contrary the neural network's ability to generalize is not impacted significantly if the number of neurons in the hidden layer is relatively low. For example, the training performance curves and the error histograms for $N_h = 4$ are given below:

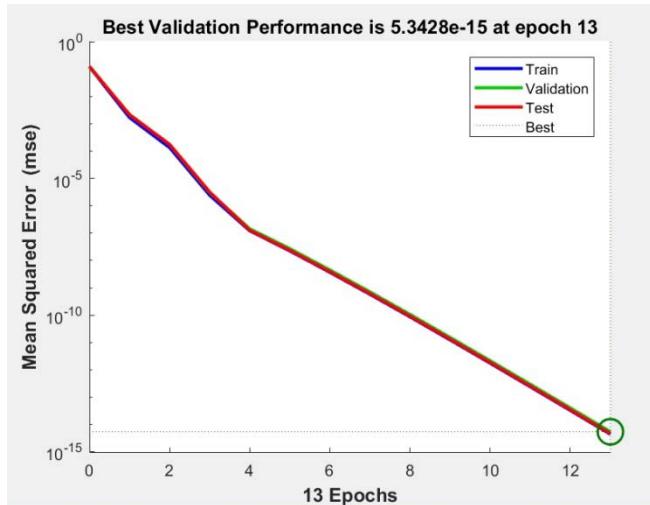


Figure 5: Validation Performance with One Hidden Layer Containing 4 Neurons

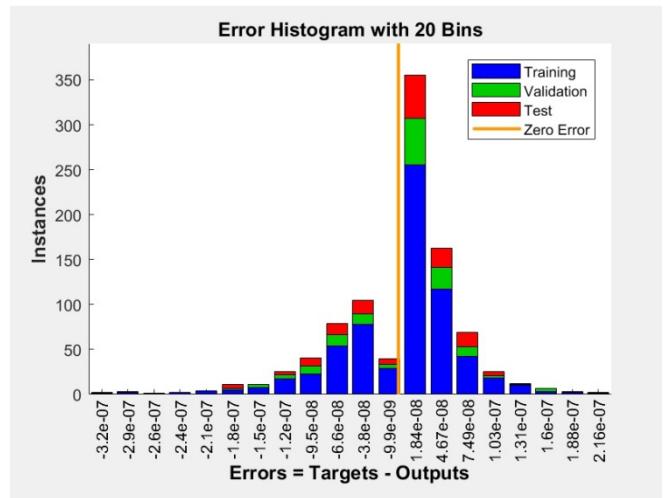


Figure 6: Error Histogram with One Hidden Layer Containing 4 Neurons

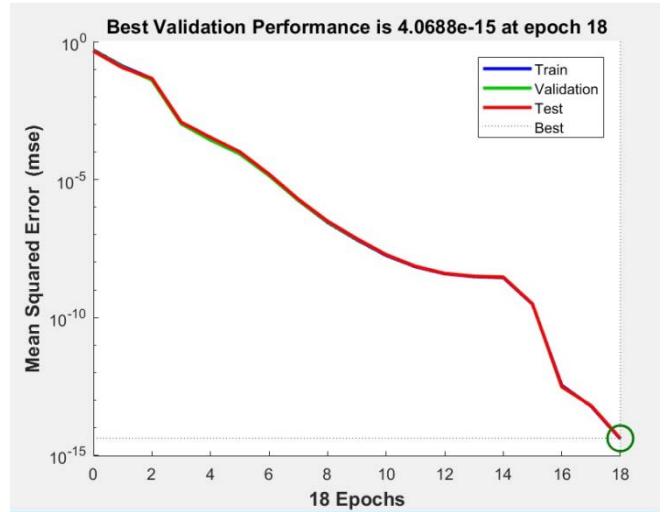


Figure 7: Validation Performance with Two Hidden Layers, Each Containing 4 Neurons

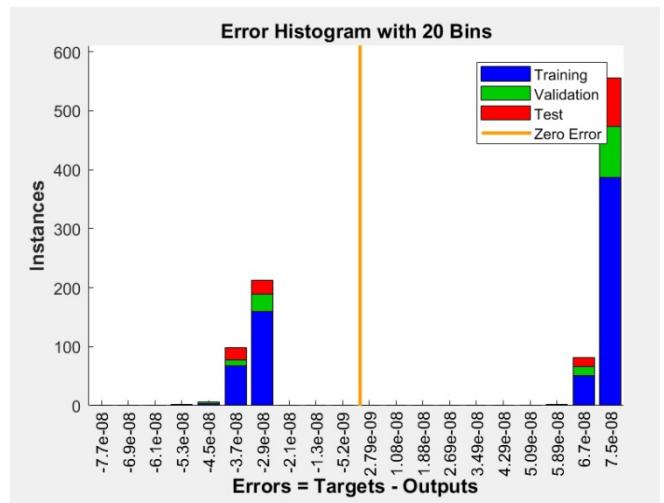


Figure 8: Error Histogram with Two Hidden Layers, Each Containing 4 Neurons

IV. CONCLUSION

When deploying an artificial neural network in software, the user has the flexibility of adding as many or as few neurons as he wants. However a situation may arise wherein the user must use a hardware implementation of a neural network [16] [17] [18], which does not offer the same flexibility. Under such a circumstance, the user may continue to get good performance out of the neural network even if the number of neurons in the hidden layer is larger than what is recommended, provided the neural network has just one hidden layer. However, the validation performance will be unacceptably poor if multiple hidden layers are present. When it comes to classifying data for schizophrenia patients, the user must avoid using neural networks with more than one hidden layer.

V. ACKNOWLEDGMENT

We would like to thank Dr. Subhadip Bharati, MD, for his role in helping us obtain the training data.

No special funding was received for this project.

VI. REFERENCES

- [1] Benjamin J. Sadock, Virginia A. Sadock, and Pedro Ruiz, *Synopsis of Psychiatry: Behavioral Sciences/Clinical Psychiatry*.: Wolters Kluwer, 2014.
- [2] S.R. Kay, A. Fiszbein, and L.A. Opler, "The Positive and Negative Syndrome Scale (PANSS) for Schizophrenia," *Schizophrenia Bulletin*, pp. 261-276, 1987.
- [3] M. Obermeier, "Should the PANSS be Rescaled?," *Schizophrenia Bulletin*, 36, pp. 455-460, 2010.
- [4] H. Temurtas, N. Yumusak, and F. Temurtas, "A comparative study on diabetes disease diagnosis using neural networks," *Expert Systems With Applications*, vol. 36, pp. 8610-8615, 2009.
- [5] O. Er, N. Yumusak, and F. Temurtas, "Chest diseases diagnosis with artificial neural networks," *Expert Systems with Applications*, vol. 37, pp. 7648-7655, 2010.
- [6] D. Gil, M. Johnsson, J.M.G. Chamizo, A.S. Paya, and D.R. Fernandez, "Application of artificial neural networks in the diagnosis of urological dysfunctions." *Expert Systems with Applications*, vol. 36, pp. 5754-5760, 2009.
- [7] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Marcia J. Horton, Ed. New Jersey, United States of America: Prentice Pearson Hall, 2009.
- [8] K.G. Sheela and S.N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks," *Mathematical Problems in Engineering*, vol. 2013, May 2013.
- [9] T. Vujicic, T. Matijevic, J. Ljucovic, A. Balota, and Z. Sevarac, "Comparative Analysis of Methods for Determining Number of Hidden Neurons in Artificial Neural Network," , Varazdin, Central European Conference on Information and intelligent Systems, pp. 219-223.
- [10] J.Y. Li, T.W.S. Chow, and Y.L. Yu, "The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network," in *Proceedings, IEEE Internation Conference on Neural Networks*, 1995, Perth, 1995.
- [11] S. Tamura and M. Tateishi, "Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 251-255, March 1997.
- [12] S. Xu and L. Chen, "A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining," in *5th International Conference on Information Technology and Applications*, 2008, pp. 683-686.
- [13] K. Shibata and Y. Ikeda, "Effect of Number of Hidden Neurons on Learning in Large-Scale Layered Neural Networks," in *ICROS-SICE International Joint Conference*, Fukuoka, 2009, pp. 5008-5013.
- [14] S. Karsoliya, "Approximating Number of Hidden Layer Neurons in Multiple Hidden Layer BPNN Architecture," *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714-717, 2012.
- [15] F. Panchal and M. Panchal, "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Networks," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 11, pp. 455-464, November 2014.
- [16] A. Dinu, M.N. Cirstea, and S.E. Cirstea, "Direct Neural-Network Hardware-Implementation Algorithm," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1845-1848, May 2010.
- [17] A. Gomperts, A. Ukil, and F. Zurfluh, "Development and Implementation of Parameterized FPGA-Based General Purpose Neural Networks for Online Applications," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 78-89, February 2011.
- [18] N.J. Cotton and B.M. Wilamowski, "Compensation of Nonlinearities Using Neural Networks Implemented on Inexpensive Microcontrollers," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 733-740, March 2011.