



## Analysis of Multi layer Perceptron Network

Jatinder Kaur<sup>1</sup>, Dr. Mandeep Singh<sup>2</sup>, Pardeep Singh Bains<sup>3</sup>, Gagandeep Singh<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Electronics and Communication Engineering  
Guru Gobind Singh College of Modern Technology, Kharar.

<sup>2</sup> Associate Professor, Chandigarh University, Gharuan.

<sup>3</sup>Assistant Professor

Chandigarh University, Gharuan.

Research Scholar, Punjabi University, Mohali.

E-mail: [jatinder\\_kor@yahoo.co.in](mailto:jatinder_kor@yahoo.co.in)

**Abstract-**In this paper, we introduce the multilayer Perceptron (feedforward) neural network (MLPs) and used it for a function approximation. For the training of MLP, we have used back propagation algorithm principle. The main purpose of this paper lies in changing the number of hidden layers of MLP for achieving minimum value of mean square error.

**Keywords-** MLP; Tansig; Perceptron; ANN; MSE.



---

## Council for Innovative Research

Peer Review Research Publishing System

**Journal:** INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 7, No 2

[editor@cirworld.com](mailto:editor@cirworld.com)

[www.cirworld.com](http://www.cirworld.com), [member.cirworld.com](http://member.cirworld.com)

# 1. INTRODUCTION

In this light, new schemes have been devised for a better solution of our needs. Recently, out of these schemes one pretty technique based on our brain system is the Artificial Neural Network (ANN). The profound property of a neural network is the ability of the network to learn from its environment and then enhance its performance through learning. Learning means cogitate (focus) a network to yield a particular response to a specific input [1].

Several neural networks have been developed and analyzed over the last few decades. These include self-organizing neural networks [2, 3], the Hopfield network [4, 5], radial basis function networks [6], and multilayer perceptrons (MLPs) [7].

MLPs have evolved over the years as a very powerful technique for solving a wide variety of problems. Much progress has been made in improving performance and in understanding how these neural networks operate.

In this paper we develop a network and train it for a function and then analyzing the results. Simultaneously, we would like to investigate the effects of changing the number of hidden layer processing elements. In this paper, we focus on the most common neural network architecture such as multilayer Perceptron. A multilayer Perceptron is a feedforward artificial neural network model that maps input data samples onto the appropriate number of outputs. An MLP employ a supervised learning technique named as backpropagation for training the network [8]. The backpropagation algorithm is used in layered feed-forward ANNs. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers.

AN MLP includes multiple layers of nodes, with each layer completely connected to the next one. Excluding the input node, each node is a neuron with a non linear activation function. When the input layer receives input, its neurons generate output, which becomes input to neurons of the next layer. The process continues until the output layer is invoked & its neurons fire their output to the external environment. The block diagram view of MLP shown in fig. 1.

A topological structure of three layer network is shown in Fig. 2. Only three layer MLPs will be considered in this work. For the actual three layers MLP, all of the inputs are also fully directly connected to all of the outputs.

The overall performance of the MLP is measured by the mean square error (MSE) expressed by :

$$E = \frac{1}{N} \sum_{p=1}^{N_p} E_p = \frac{1}{N} \sum_{p=1}^{N_p} \sum_{i=1}^M [t_p(i) - y_p(i)]^2$$

Where

$$E_p = \sum_{i=0}^M [t_p(i) - y_p(i)]^2$$

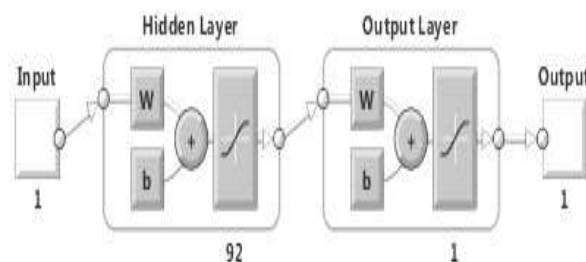
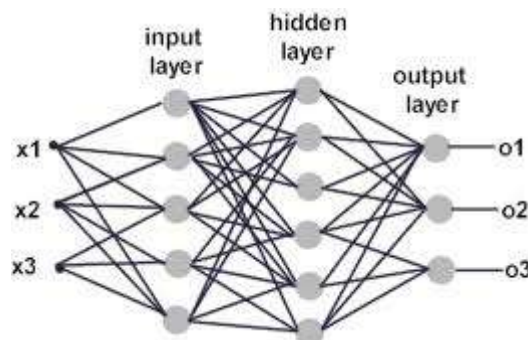


Fig 1: Block diagram view of MLP(for 92 hidden layers)



**Fig 2: Typical Topological view of MLP Network [9]**

$E_p$  Corresponds to the error for the  $P_{th}$  pattern and  $t_p$  is the desired output for the  $P_{th}$  pattern.

## 2. Review of Literature

An attempt has been made to examine the effect of changing both the No of hidden layers of MLPs and the No of Processing elements that exist in the hidden layers on the analyzed properties of Jordan Oil Shale. A compare between the o/p of MLP and the mathematical formula was done and results revealed that o/p performance parameter are best suited with the experimental [9].

A better comparison between the linear and nonlinear neural network predictors for SIR estimation in DS/CDMA

Systems were presented in [10]. One more issue was addressed in this paper was to solve the problem of complexity by using Minimum Mean Squared Error (MMSE) principle. The optimized Adaline predictor was compared to optimized MLP by employing noisy Rayleigh fading signals with 1.8 GHZ carrier frequency in an urban environment [10].

There are two principles to find the optimal predictor. Two criteria, Minimum Mean Squared error (MSE) and Minimum Description Length (MDL) criteria [11], [12] are used for filter design parameter selection.

In [13], a neural network model capable of learning and generating complex temporal patterns by self-organization was proposed. The model actively regenerates the next component in a sequence and compares the anticipated component with the next input. A difference between what the model anticipates and actual input triggers one-shot learning. Although the anticipation mechanism enhances the learning efficiency of this model, it required several training sweeps to learn a sequence.

## 3. Problem Statement & Methodology

### 3.1 Problem Statement

In this paper we create and train a Multilayer Perceptron network to learn to evaluate a function such as:

$$y = x^2 - 3/x + 3,$$

$x$  is the matrix contacting inputs and  $y$  is the matrix which holds the target. The input vector ranges from -10 to 10.

For the Implementation of above function on neural network first we generate the training input output data. Then we create a neural network and train it for the given function. After that we generate test input output data so that we can simulate the model for checking its functionality.

### 3.2 Research Methodology

#### Steps for Research Methodology:

##### Step1: Input/output training data generation:

In order to train the neural network using supervised learning algorithm, input-output training data is required. This data can be easily generated by varying the input within the range for which training is required and obtaining the corresponding outputs by solving the system equation.

##### Step2: Neural network creation and input and target scaling

in this step MLP network is created for 1 input layer and 5 hidden layers also scaling of input and target is done so that they fall within a specified range. We used the 'Tansig' transfer function for hidden and outer layers shown in

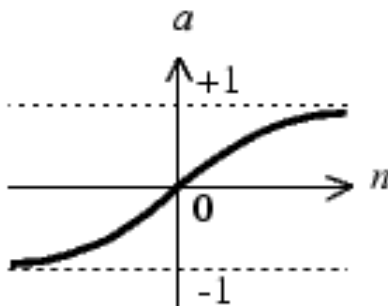


fig.3. 'Tansig' calculate the layer output from its net input. This function has more speed as compared to other functions. This allows a smooth transition between the low and high output of the neuron (close to zero or close to one).

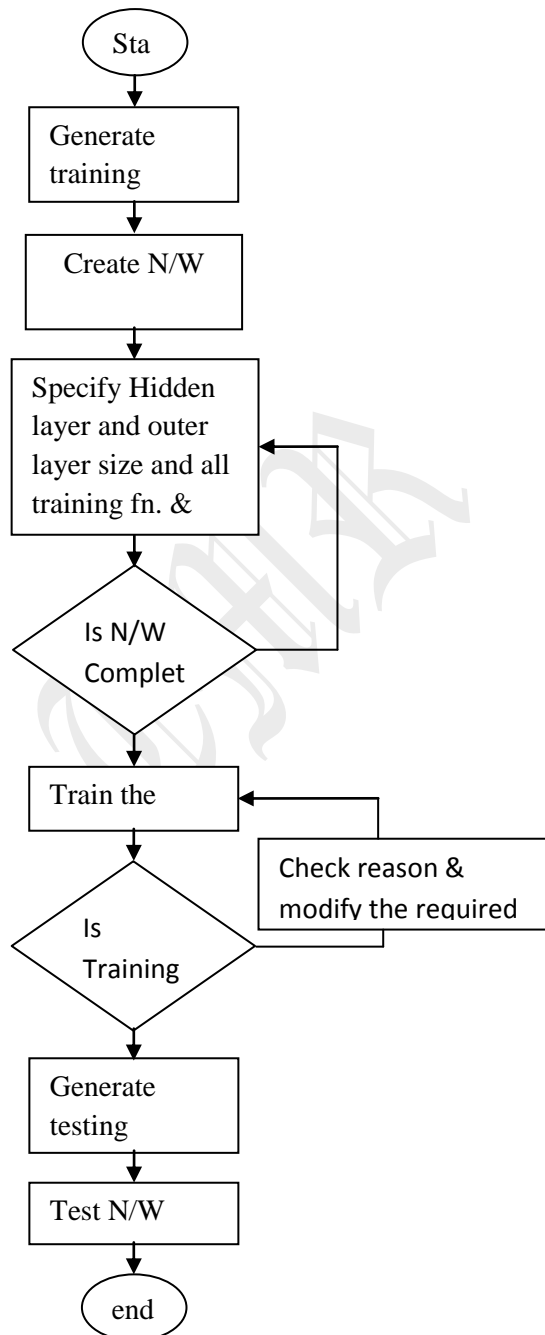
**Fig 3: Tan-sigmoid transfer function**

**Step3: MLP Training**

We have used the `traingd` function for Training our MLP network. Then we check that our training is complete or not. If it stops in-between, we check the reason and modify the parameters which affect the training and discourage it for achieving its best.

**Step4&5: Input-output test data generation& Testing**

After network creation and training, now it needs to be tested with a different set of data. After obtaining test samples network needs to be simulated for these test inputs. At last analyze the results.



**Fig 4: Flow Chart of Research Methodology**

## 4. Results & Discussion

### 4.1 Regression Plot

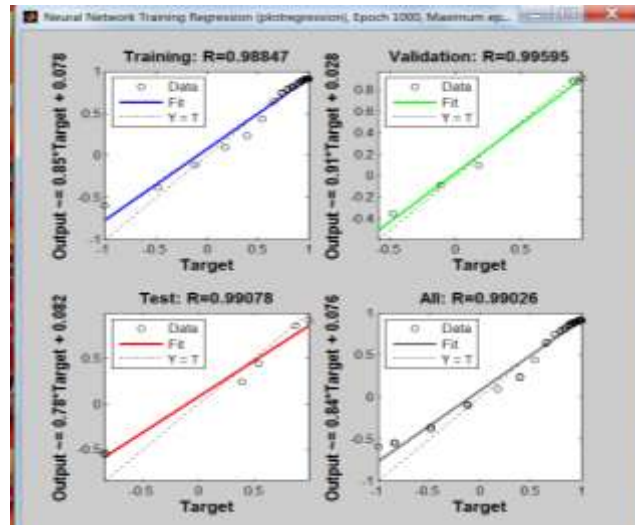


Fig 5: Regression plot values (For 28 hidden layers)

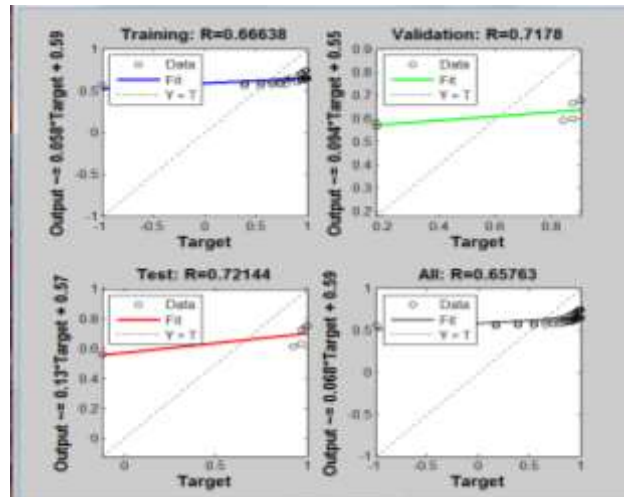
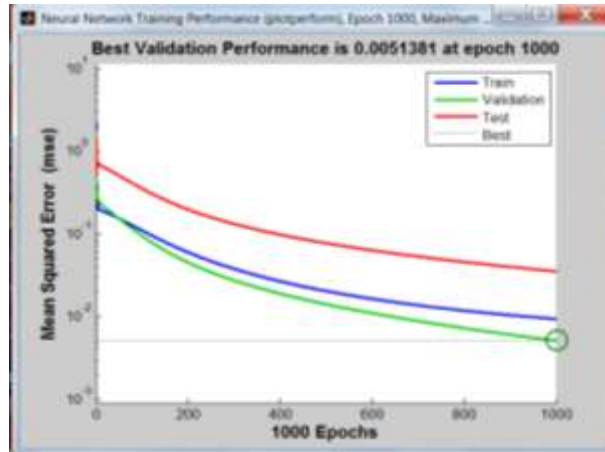


Fig 6: Regression plot values (For 18 hidden layers)

Regression is a process of fitting models to data. From above fig.5., we analyze that our model approximate fitted to our most of samples(data) values. Regression may be linear or non linear. In non linear regression, most of the data values set to zero for fitting the model to that data. From fig. 5 and fig. 6, we conclude that for getting the appropriate regression values, we have to select the number of hidden samples in such a way that it makes our model fitted to most of data samples.. In fig.6., approximate fitted to samples(data) values is not exact.

### 4.2 Performance Plot



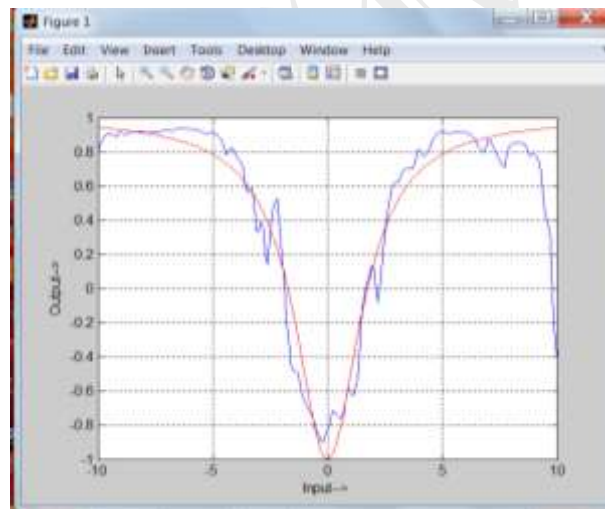
**Fig 7: Performance values**

In performance plot (Fig.7.) we can see that our validation curve approximately achieved our MSE minimization goal. But if we want that our other two curves also reaches at same point then we have to continuously changing the number of hidden layers until we have to achieve our goal.

We obtain the regression and performance plot before the simulation of our network.

#### 4.2 Simulation Result Plot

Fig 8. , shows the plot of the test output and the test input. It is worth noticing that network created is able to approximate the function to reasonably good extent.



**Fig 8: Trained neural Network output**

## 5. Conclusion

When we design neural networks for the function demonstrated in this paper, there are many ways used to check the effects of network structure which refers to the number of hidden samples when the number of inputs and outputs are fixed which in turn affects on the o/p predicted parameters. The results reported in this paper suggest that for making a model to fit to data samples, the one of essential thing is the size of hidden samples.

## REFERENCES

- [1] Jacek M.Zurada, Introduction to Artificial neural systems, jaico publishing house, 1997.
- [2] Lippman, R. P., 1987. An introduction to computing with neural networks. IEEE ASSP Magazine, (April 1987).
- [3] Kohonen,T.,1988. Self Organization and Associative Memory. Springer-Verlag, 2<sup>nd</sup> edition.

- [4] Hopfield, J. J, 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the U.S.A., pp. 2554-2558.
- [5] Galan-Marín, G. and Perez, J. 2001. Design and analysis of maximum Hopfield networks. IEEE Transactions on Neural Networks. vol. 12 no. 2, pp. 329-339, (March 2001).
- [6] Powell, M. J. D., 1985. Radial basis functions for multivariable interpolation: A review. IMA Conference on Algorithms or the Approximation of Functions, pp. 143-167, RMCS, Shrivenham, U. K.
- [7] Chandrasekaran, H., 2000. Analysis and Convergence Design of Piecewise Linear Networks. Ph.D. Dissertation, The University of Texas at Arlington.
- [8] Rosenblatt, Frank., 1961. Principles of neurodynamics: Perceptrons and the theory of Brain Mechanisms. Spartan Books.
- [9] Nazzal, J.M, El-Emary, I.M and Najim, S.A. 2008. Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale. World Applied Sciences Journal. ISSN 1818-4952. Vo. 5, no. 5, 546-552.
- [10] Ardalani, N. Khoogar, A. and Roohi, H. 2005. A Comparison of Adaline and MLP neural Network-based Predictors in SIR Estimation in Mobile DS/CDMA Systems. In Proceedings of World Academy of Science, Engineering and Technology. ISSN 1307-6884, Vol.9(Nov 2005)
- [11] Gao, X.M., Ovaska, S.J. and Hartimo, I.O. 1997. Restoration of noisy speech using an optimal neural network structure. IEEE International Conference on neural networks, vol. 8, no. 6, (November 1997).
- [12] Gao, X.M., Gao, X. Z., Tanskanen, J. M. A., and Ovaska, S.J. 1997. Power prediction in mobile communication systems using an optimal neural network structure. IEEE Trans. on Neural Networks, vol. 8, no. 6, (November 1997).
- [13] Wang, D. and Yuwono, B. 1995. Anticipation-based temporal pattern generation. IEEE Trans. Syst., Man, Cybern., vol. 25, no. 4, (9 Apr. 1995). 615–628.