

# A Study on Artificial Neural Network

Mariya Ushshaque<sup>1</sup> and Deependra Pandey<sup>2</sup>

<sup>1,2</sup>Department of Electronics and Communication Amity School of Engineering and Technology Amity University Lucknow Campus  
E-mail: <sup>1</sup>mariya20nic@gmail.com, <sup>2</sup>dpandey@lko.amity.edu

**Abstract**—An information processing paradigm which is influenced by the nervous system is the Artificial Neural Network. The main component of this system are the neurons which are interconnected in large numbers, activation inputs, bias values, weights altogether generates an output which is the trained network. These networks are ingenious technique in neural computing and learning problems. Likely ANNs are mainly aimed for the building up of machine intelligence but after the advent of complex algorithms pattern recognition is the central one. This paper deals with the understanding of proper functioning, implications and learning processes and achieving trained network with minimum error.

**Keywords:** Neurons, activation inputs, bias, trained network, machine intelligence, learning, error.

## 1. INTRODUCTION

Apart from the concept of machine learning, neural networks are mainly exploited in classification of objects and pattern recognition. Artificial Neural Networks are the information processing systems which comprises of multiple inputs called nodes or neurons. The whole system is based on three layers—an input layer, hidden layer and an output layer. The input values are combined by taking the summation with the weighted values. A set of activation values and learning rules determine the network performances. Input units are those units which receive data from the source, output units transmits the data to neural networks which are connected in long chains and the hidden layer is the layer which allows all computation to take over. Neural networks consists of a number of classical network architectures which are utilised as per the demand. Multilayer Perceptron and Radial Basis Function networks are used in function approximations.

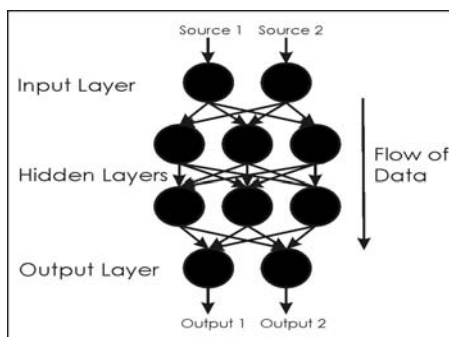


Fig. 1: A feed forward neural network

## 1.1 Topology of Network

The direction in which data is flowing from input to output is a matter of great concern. There are mainly two types of networks feed forward and recurrent. In Feed forward networks the data flow is from input to output and data is processed in multiple layers. But if there is any discrepancy in the output obtained then there is no availability of feedback paths. In recurrent networks, feedback path is available and the activation inputs relax for some time so as to let the network achieve a stable state. Examples of Feed forward networks are the Perceptron and Adaline whereas recurrent networks are Anderson, Kohonen and Hopfield.

## 2. METHODOLOGY OF NETWORK TRAINING

There are many algorithms existing for the determination of the network parameters. Firstly the architecture of network is decided as to which one to opt specifying the number of hidden layers and nodes. The activation function is also chosen and most of the time it is a known value. The activation function could be the sigmoid, hyperlin, step and few more. Weights and biases values are the unknown parameters. The parameters namely error goal, number of epochs are defined pertaining to the training algorithm selected. The network obtained is tested by the measuring input data.

## 3. LEARNING PROCESS AND ALGORITHMS FOR TRAINING THE NEURAL NETWORK

The artificial neural network can be trained in two distinct methods- supervised learning and unsupervised learning. Supervised learning is done by matching the output pattern with the given input values. This method is also known as associative learning. In unsupervised learning output unit is trained to match the input clusters and system itself has to develop its own represented input.

Perceptron network is trained by first starting with random weights for the connections. Then an input vector say 'x' is selected from the set of training samples. If the perceptron gives an incorrect response the weighted values are altered. In the Adaline network learning the delta rule is used that is the net output.

#### 4. CLASSIFICATION OF DATA WHICH ARE LINEARLY SEPARABLE USING PERCEPTRON NETWORK

Linearly inseparable data cannot be classified using single layer perceptron network. Data clusters from two domains are taken as input and they are classified using perceptron. The algorithm undertaken for this is firstly we have initialised input and output and there 50 number of samples .Its plot is produced thereafter which shows two clusters of data at a distance from each other. Then neural network is trained by creating it using Neural Network Toolbox in Matlab. The decision limiting line is also drawn in plot which purely classifies the two clusters.

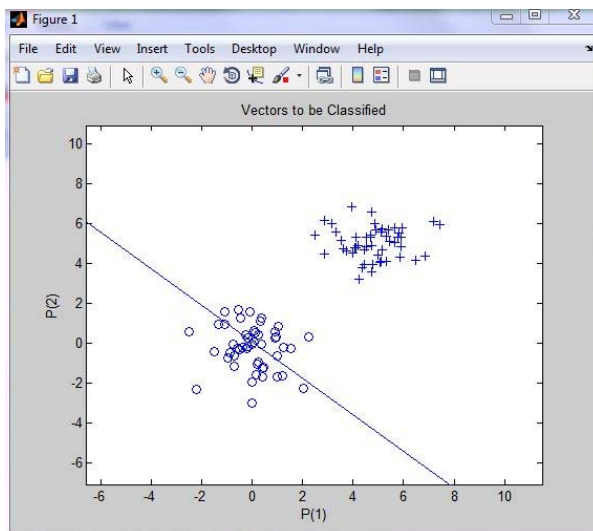


Fig. 2: Classified clusters of data

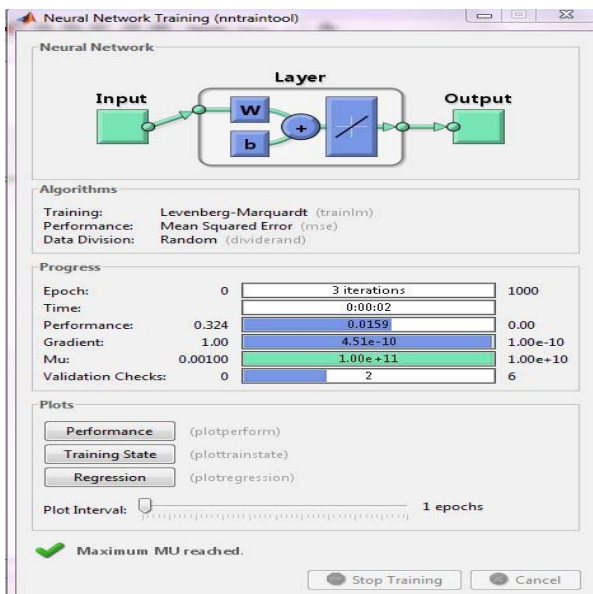


Fig. 3: Neural Network Trained

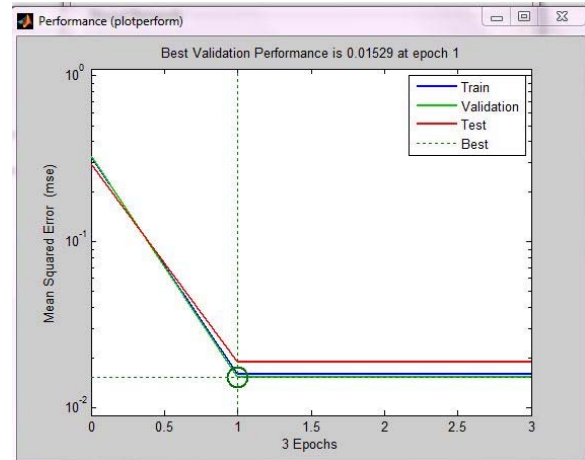


Fig. 4: Performance graph

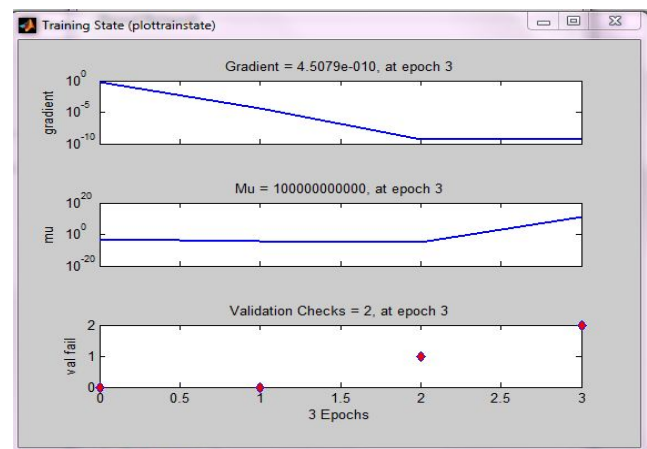


Fig. 5: Training state of the network

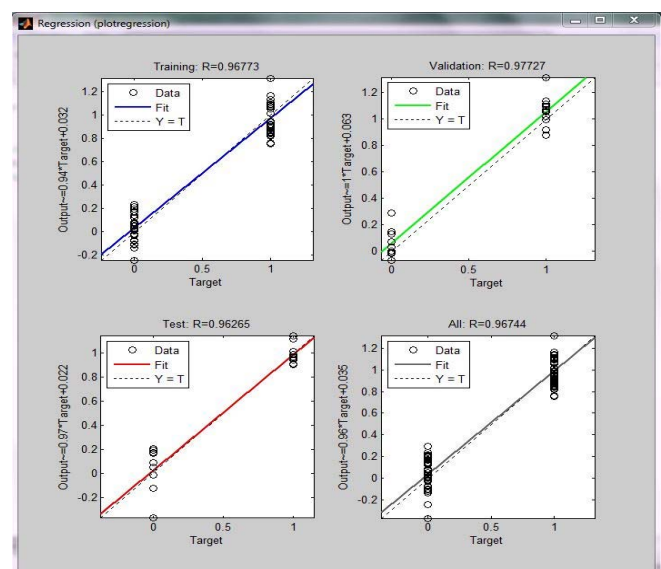


Fig. 6: Regression graph

The neuron what exactly does is computes the value of weighted sum of input data and then compares it with a value known as the threshold value. Accordingly the value of net input when comes out to be less than threshold value, the output value is assigned as -1 and when net input is greater than threshold then net output is +1. There are various activation function for neurons :sigmoid function, sign function, step and linear function.

## 5. MULTI LAYERED PERCEPTRON

Multilayered perceptron network is a feed forward network with one or more hidden layers in it. As the name infers hidden layer hides its desired output and it is not possible to probable the output at that layer. Learning in multilayered perceptron is mainly done using Back Propagation algorithm. Here if there occurs any error then error is backwardly propagated and then weights are adjusted accordingly.

## 6. PROS AND CONS OF NEURAL NETWORK

Neural networks are easy in generating the complex non linear functions but difficult to categorise hypothesis space. Predictions are done quickly but interpretability is slower. Overfitting is done, efficient optimisation and regularisation of parameters.

## 7. ACKNOWLEDGEMENT

This work is accomplished by the support of Amity School of Engineering and Technology for providing IEEE papers access and the co author for valuable effort.

## REFERENCES

- [1] Saurabh Karsoliya, 2012, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture", *International Journal of Engineering Trends and Technology-Volume3Issue6*
- [2] Berry, M.J.A., and Linoff, G. 1997, *Data Mining Techniques*, NY: John Wiley & Sons.
- [3] Kazuhiro Shin-ike, "A Two Phase Method for Determining the Number of Neurons in the Hidden Layer of a 3-Layer Neural Network", SICE Annual Conference 2010, August 18-21, 20 10, The Grand Hotel, Taipei, Taiwan.
- [4] Stuti Asthana and Rakesh K Bhujade, "Handwritten Multiscript Pin Code Recognition System having Multiple hidden layers using Back Propagation Neural Network", *International Journal of Electronics Communication and Computer Engineering*, ISSN(Online): 2249 - 071X, Volume 2, Issue 1, Sep 2011
- [5] Rivals I. & Personnaz L. "A statistical procedure for determining the optimal number of hidden neurons of a neural model", Second International Symposium on Neural Computation (NC'2000), Berlin, May 23-26 2000.
- [6] Fnaiech, F., Fnaiech, N., Najim, M., 2001 "A new feedforward neural network hidden layer neuron pruning algorithm "IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01).
- [7] Yinyin Liu, Janusz A. Starzyk, Zhen Zhu, "Optimizing Number Of Hidden Neurons in Neural Networks", [http://www.ohio.edu/people/starzykj/network/Research/Papers/Recent%20conferences../Hidden%20Neurons%20AIA2007\\_549-204.pdf](http://www.ohio.edu/people/starzykj/network/Research/Papers/Recent%20conferences../Hidden%20Neurons%20AIA2007_549-204.pdf)
- [8] R.Rojas, *Neural Networks*, Springer- Verlag, Berlin 1996.
- [9] Blum, A., 1992, *Neural Networks in C++*, NY: Wiley.