ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(31), DOI: 10.17485/ijst/2016/v9i31/91847, August 2016*

Performance Evaluation of Feed Forward Neural Network for Wired Equivalent Privacy/Wi-Fi Protected Access Protocols

Menal[1]* and Sumeet Gill[2]

[1]Department of Computer Science, Maharaja Surajmal Institute, Janakpuri, New Delhi - 110058, India; menaldahiya@gmail.com
[2]Department of Mathematics, Maharashi Dayanand University, Rohtak - 124001, Haryana, India

Abstract

Objective: Millions of people use wireless devices in their day to day diligences without knowing the security facets of Wireless Technology. The aim of our research is to enhance the execution of widely used wireless devices's protocols by examining their behavior with Feed Forward Neural Network. Fundamentally, Neural Network is a multilayer perceptron network. It processes the records one at a time and "learn" by comparing the obtained output with the actual output. Hidden layer neurons play a cardinal role in the performance of Back Propagation. The process of determining the number of hidden layer neurons is still obscure. The work is focused on performance evaluation of the hidden layer neurons for WEP (Wired Equivalent Privacy) and WPA (Wi-Fi Protected Access) protocols. **Methods/ Statistical Analysis:** For this work, three network architectures have been picked out to perform the analysis. The research work is carried out by using Back Propagation Algorithm in Neural Network Toolbox on the data captured by using Wireshark tool. **Findings**: The behavior of various unlike hidden neurons is evaluated through simulation technique. Network performance is also diagnosed with the help of epochs and Mean Square Error (MSE). The performance of Neural Network is evaluated and outcomes indicate that hidden layer neurons affect the functioning of the network. **Improvement**: We would like to work with the parameter and learning of the Neural Network to achieve best results.

Keywords: Back Propagation, Feed Forward Neural Network, Hidden Layer, Mean Square Error, Wi-Fi Protected Access, Wired Equivalent Privacy

1. Introduction

Artificial Intelligence technologies like Neural Network are vastly used technology in pattern recognition. Pattern recognition is an approach which recognizes patterns and categorize data based on preliminary information or deduced numerical data from the patterns[1]. It is all about learning the behavior of machines in observing environments, discriminating patterns and building rational conclusion of the class of the patterns. This technique will be very useful for business goals, in engineering fields or in analyzing the data etc[2,3]. Pattern recognition is the more applicable answer to all the worries related to the acknowledgement and classification of patterns like look recognition, medical examinations, speech recognition and codification of handwritten letters, figures, symbols etc. This new approach provides excess benefit or extra help in solving "real world" problems in many fields[4].

Due to the rapid advancement in the world of electronics, the wireless devices are used by millions of people

*Author for correspondence

daily. Widely adaptation of Wireless Technology depends on the standards, security levels and the flexibility of the Wireless Technology. In WLANs privacy is accomplished by protection of data contents with encryption. In current scenario WEP and WPA are used for the security of wireless network environment. When security measures are enabled in Wi-Fi devices, an encryption protocol is usually used called WEP, it is the default encryption process introduced in the first IEEE 802.11 standard. It is based on RC4 stream cipher encryption algorithm and developed with the aim of to offer security through 802.11 wireless network[5]. WPA is a standard based, interoperable security specification introduced in 2003 by Wi-Fi Alliance and its purpose is to enhance the level of security for wireless LANs and overcome the lapse that's in WEP. WPA has some good security features such as WPA encryption process and WPA authentication mechanism.

Wireless networks are becoming more and more popular today. Therefore, the protection of wireless networks is essential means protection of confidentiality, integrity and availability[6]. Neural Networks caters the simple mechanism of approaching problems.

Generally Artificial Neural Network made up of processing units which are interconnected by weights and here the output of one layer acts as an input to the next layer. MLP a network consists of neurons called perceptrons. Artificial Neural Network (ANN) commonly relates as Multilayer Perceptron Network[7]. In 1958, Rosenblatt presented the idea of single perceptrons to the world. Mainly, the Neural Network has three layers such as input, hidden and output layers. Initially in the development phase, hidden layer was not added in the problems having linearly separable domain. But when a function uses uninterrupted mapping between limited intervals, one has to make use of single hidden layer directly which does not involve in the network but it affects the final output[8]. Therefore, training the network one has to choose carefully the number of hidden layers and the neurons in that hidden layer. Adoption of excessive neurons in the processing units may conclude in overfitting of the network and using the fewest number of neurons in the processing units leads to underfitting[9,10].

The proposed experiments,analyze the behavior of Feed Forward Neural Network for the wireless security protocols by increasing the amount of neurons in hidden layer for the given set of problems. Section 2 gives the overview of the supervised Feed Forward Network used in the problem. Section 3 describes the various topologies used for designing the network. Section 4 shows the outcome and discussion of the proposed work while the last section concludes the paper.

2. Supervised Feed Forward Network

Artificial Neural Network consists of many nodes and each node has a node function associated with it. Artificial Neural Network contains neurons called processing elements which are large in numbers in the network. Neurons are connected by communication links with associated weights w_1, w_2, w_3…… $w_{n..}$ The weight associated with links increases as the approaching signal reaches out to the Neural Network. Neural Network uses weights as a key information to solve a problem and initialization of weights acts as an important aspect for the Neural Network[11]. Weights may be fired or randomized at the beginning of the network. An equation for computing the weighted sum as below:

$$\text{Net input} = \sum_{1}^{n} x_i \, w_i$$

From the input, applying the activation function the output signal may be obtained[12]. The output of any node will be calculated by utilizing the binary sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Artificial Neural Network comes in many different patterns. The Neural Network is a layered architecture having three forms of layer. A network which organized the neurons in the simplest form of structured network where every node of each layer is connected to the each other node of the forwarding layer. In Feed Forward Network architecture single layer of input neurons is directly linked with the output units and sometimes many hidden layers are inserted in between these two main layers. The multilayer Feed Forward Network is advanta-

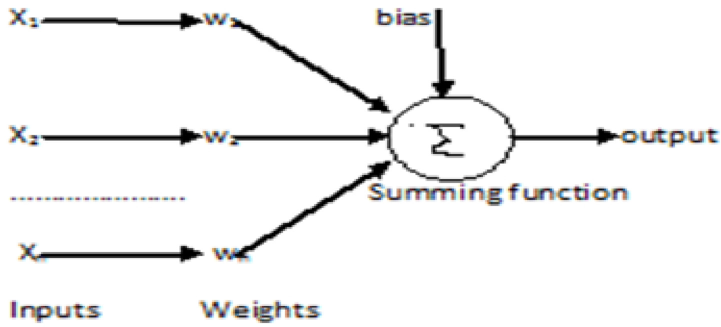

Figure 1. Neural Network Architecture.

geous over single layer because many statistical problems can be solved by using these types of networks. In this type of network, hidden layer plays beneficial part of the network where computational process is done in the layer. The sum is calculated by considering every node in the hidden layer as an input from the previous layer otherwise sigmoid function can lead to a feasible change in the range[13].

The Feed Forward Neural Network is based on supervised learning algorithm: This type of learning is performed with the help of a teacher or an advisor and he/she trained or supervised the learner. Similarly, a target set which represents the desired output is required by each input node in the Artificial Neural Network. During training the neurons are fed into the network, which results in an output vector.This is the actual output vector which is compared with the result that the user wants. After computation if the network found any difference, then an error signal is produced by the network[14]. Then, this error signal is propagated back from the output layer to the input layer and used for adjustments of weights. For this type of studies, we take for granted that the correct "target" values are known for each input vector.

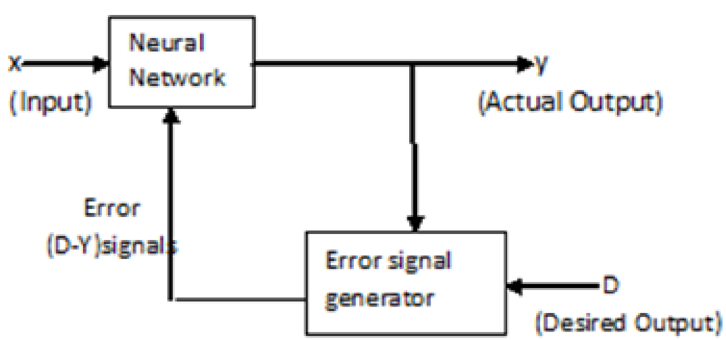

Figure 2. Supervised Learning.

3. Implementation and Simulation Designing

For this experiment, we have taken three types of architecture by taking a different volume of neurons in the hidden layer of the Feed Forward Network. The first implementation was a two layer MLP (10 neurons in the hidden layer). This structure is referred to as 32-10-32. The second implementation was on the structure 32-20-32. The third implementation was on the structure 32-30-32.

3.1 Neural Network Architecture

A fully linked multilayer perceptron Feed Forward Architecture is used in the Neural Network concept. In a multilayer Feed Forward Back Propagation Network, different layers are interconnected by links called weights such as a first layer (called as input layer) is connected to the second layer (called as hidden layer) and hidden layer is in contact with output layer and so on. Each node of the input layer collects the input signal x_i, and broadcasts the signals to each and every node of the hidden layer. The purpose of each hidden nodes z_j is to first calculate the sum of weighted input signals and applying the activation function on the computed result and then send this signal to further connected layers. This process continues upto the output with y_k of the network[15].

The objective of the Back Propagation algorithm is that the network can be trained to determine the mapping of an input-output pattern pair. Method follows a Gradient Descent approach to achieve the optimum weights. The weights are adjusted in such a way that the squared difference between the desired and actual output obtained is minimized at the output layer. In this algorithm, the change in weight was based on the error calculated after every train process. Following equation shows the weight updation and change in error in the network.

Weight updation for output unit is:

$$\Delta w_{jk} = \propto \delta_k z_j$$

Weight updation for the hidden unit is:

$$\Delta v_{ij} = \propto \delta_j x_i$$

$$E = \frac{1}{2}\Sigma_K [t_k - y_k]^2$$

A gradient descent method is used in the Back Propagation network during training of the network to minimize the total squared error of net result concluded by the network[16,17].

3.2 Experimental Setup

The IP addresses of three wireless devices are used for the experiment. The addresses are converted into binary for input data. Here we have used Back Propagation learning algorithm for experiments. Neural Network training

Table 1. The parameters used for experimentation 1

Parameter	Value
Training Algorithm	Back Propagation
No. of input neurons	32
No. of hidden layer	1
No. of hidden neurons	10
No. of output neurons	32
Learning Rate	0.1
Minimum error exists in the network	0.0001
Initial Weights and Biased term values	between 0 and 1

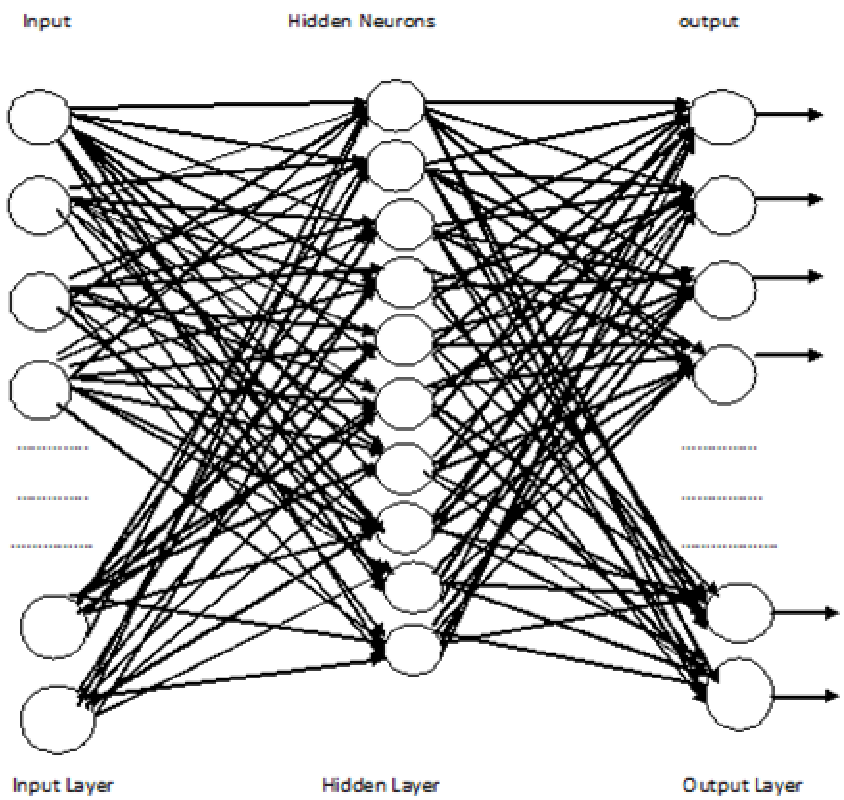

Figure 3. Neural network architecture for network topology 32-10-32.

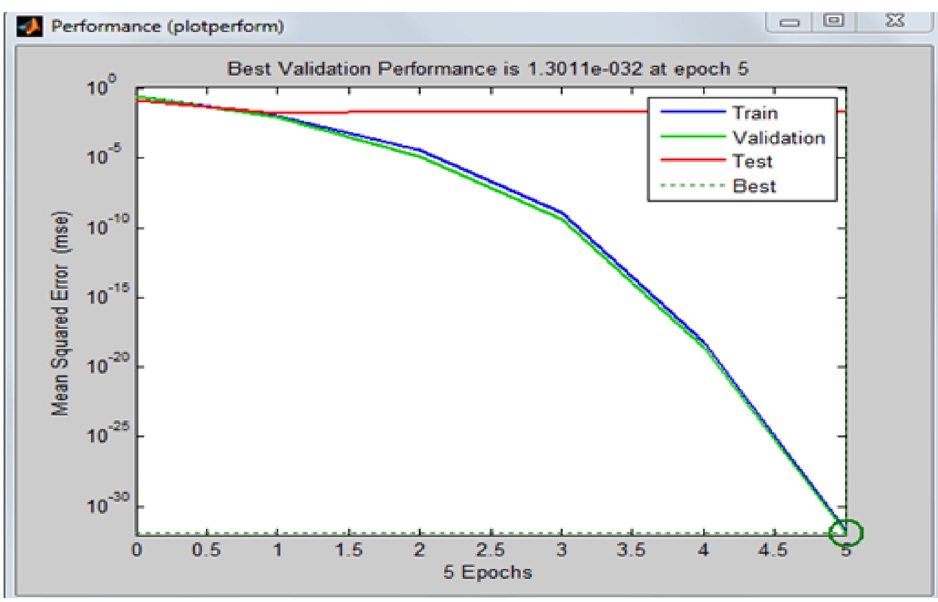

Figure 4. Performance plot of feed-forward network with topology 32-10-32.

0.1925	-0.1312	0.0682;
-0.1031	-1.0005	0.2155;
0.0058	-1.0007	0.0437;
0.0711	-0.3332	0.8273;
0.8550	0.4498	0.6903;
0.0675	-0.0323	0.0813;
0.6433	-0.0749	-1.0006;
-1.0079	0.0062	0.9412;
0.0291	0.0373	-1.0409;
0.0462	1.0002	0.0110

Weights in layer 1 to hidden layer.

-0.6114 -0.6722 0.1373 0.1714 0.3602 0.2642 -0.2325 -0.8605 -0.4275 -0.5064
-0.2334 0.2313 0.4725 -0.1005 0.1645 -1.094 -0.5651 -0.1366 0.5212 0.8832
-0.2449 0.9806 -0.1546 0.7441 -0.2935 0.0404 0.1326 -0.2037 -0.2526 0.6757

Weights in hidden layer to output layer.

is done through 5 trials in each experiment. Here in the paper we show only one. Neural Network architecture with 32 input neurons with 10 hidden neurons is diagrammatically described in Figure 3. Figure 4 shows a performance plot of feed-forward network with topology 32-10-32.

The second implementation was on the structure {32-20-32}. Figure 5 shows a performance plot of Feed Forward Network with topology 32-20-32.

Table 2. The parameters used for experimentation 2

Parameter	Value
Training Algorithm	Back Propagation
No. of input neurons	32
No. of hidden layer	1
No. of hidden neurons	20
No. of output neurons	32
Learning Rate	0.1
Minimum error exists in the network	0.0001
Initial Weights and Biased term values	between 0 and 1

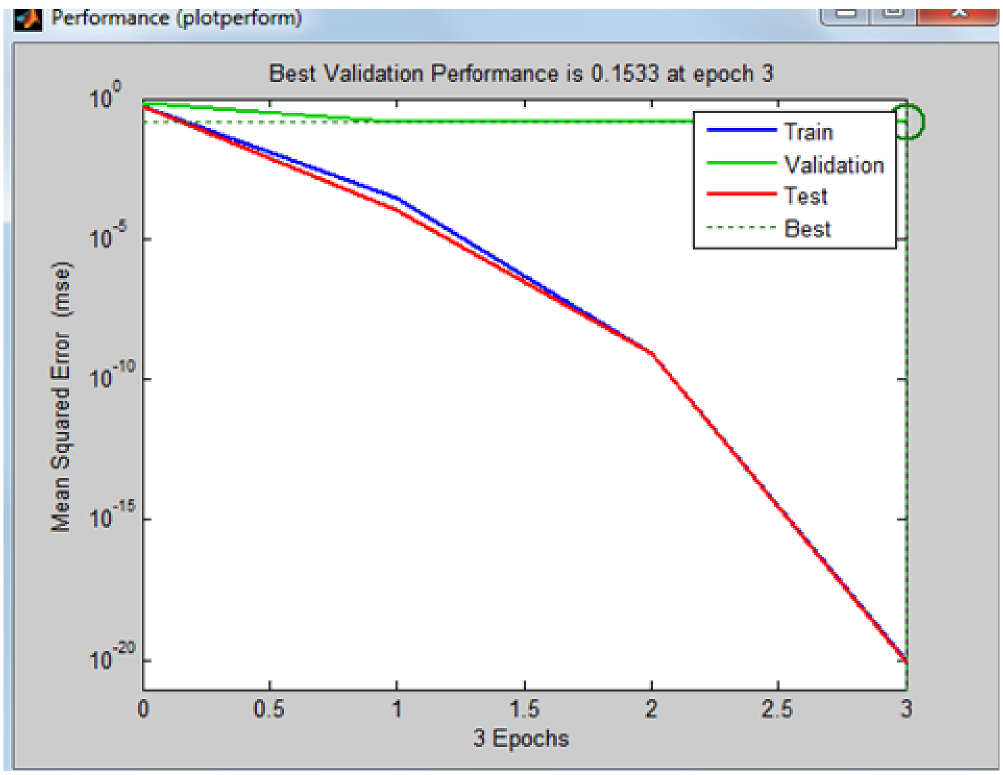

Figure 5. Performance plot of feed-forward network with topology 32-20-32.

0.0230	-1.1187	0.9793;
-0.0012	0.7008	0.3902;
-0.3057	-0.0211	0.6612;
0.4103	0.7931	0.4705;
0.0880	0.4054	0.0027;
0.2692	1.0046	-1.0352;
-0.6154	0.0553	0.1804;
0.6399	-1.0005	0.6405;
0.0273	0.6881	0.0092;
0.9035	0.2057	0.0732;
0.0204	0.7542	0.0881;
-0.4701	0.7609	0.2123;
0.0405	0.0748	-1.0902;
-1.0866	0.1115	0.3913;
0.0014	0.1404	0.6236;
-0.1235	0.0356	-1.0508;
0.0553	0.0546	0.0393;
0.6505	0.3011	0.1002;
0.3183	0.2868	-1.0030;
0.0268	-0.8365	0.0222

Weights from layer 1 to hidden layer.

-0.2080 -0.1322 -0.672 0.4264 0.3325 0.6572 -0.2725 -0.2211 0.5955 -0.2202 -0.0230 0.4525 -0.323 0.2344 -0.3042 -0.0331 -0.0515 0.8036 0.1305 0.0805;
-0.3632 -0.5502 -0.3527 0.0135 0.5573 0.6001 0.0511 0.0332 0.3249 0.0326 0.1940 0.7244 -0.4044 0.6536 -0.5498 0.5036 0.3558 1.000 0.0621 -0.6681;
-0.6103 0.2203 -0.0685 0.5527 -0.6477 -0.9809 0.4121 0.2687 -0.4376 -0.8221 0.3332 0.7786 0.2252 0.0410 0.6068 -0.0478 -0.0976 0.1343 0.0113 0.7166;

Weights in hidden layer to output layer.

The third implementation was on the structure {32-30-32}. Figure 6 shows a performance plot of Feed Forward Network with topology 32-30-32.

Table 3. The parameters used for experimentation 3

Parameter	Value
Training Algorithm	Back Propagation
No. of input neurons	32
No. of hidden layer	1
No. of hidden neurons	30
No. of output neurons	32
Learning Rate	0.1
Minimum error exits in the network	0.0001
Initial Weights and Biased term values	between 0 and 1

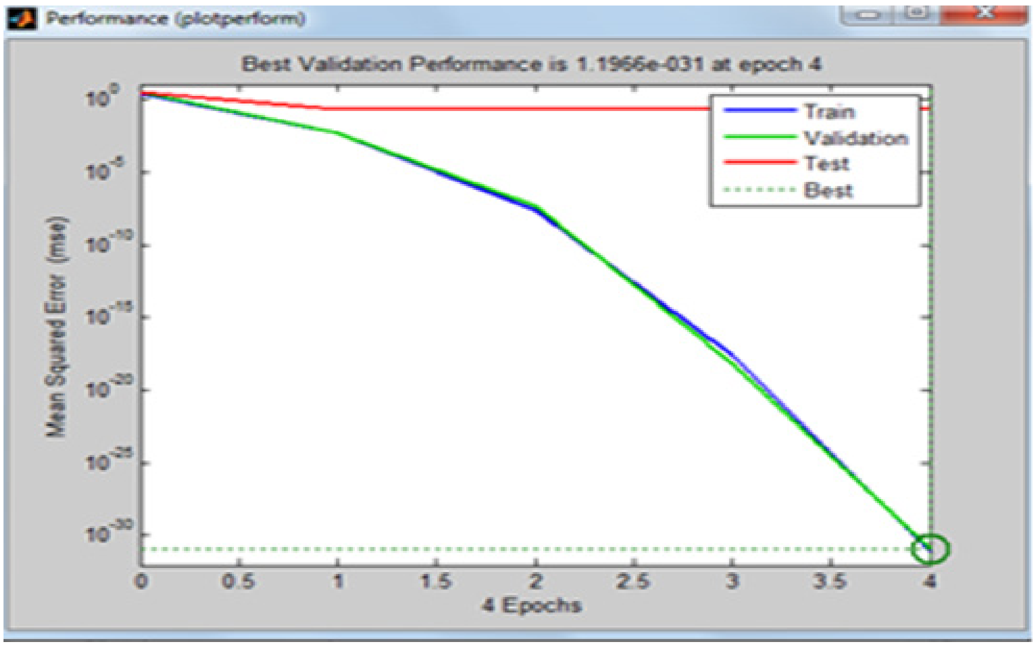

Figure 6. Performance plot of feed-forward network with topology 32-30-32.

-0.0697	-0.3934	0.0424;
-0.0379	-0.3164	0.0726;
-1.0009	-1.0437	0.0037;
-1.0074	0.7919	0.0216;
0.7202	0.2264	0.2418
0.5491	0.0606	0.3424;
-0.0488	-0.4512	1.0076;
1.0009	0.0043	0.4774;
0.0272	0.0656	0.6511;
0.2121	0.0368	-0.4211;
0.7909	-1.0365	-1.0605;
0.2755	0.8254	0.4247;
0.6427	0.0221	0.0072;
0.3649	0.5216	-0.9327;
0.6237	0.7206	0.6491;
0.0551	-0.8739	-0.3287;
0.3222	-0.4489	0.8839;
0.8183	-0.0358	0.0295;
0.1199	-1.0048	-0.4022;
-0.4659	0.1373	0.2308;
-1.0356	0.7916	-1.0096;
1.0007	-1.0414	-0.0062;
0.0866	1.0001	0.2608;
-0.9339	-1.0053	0.5745;
0.4243	0.5063	0.1053;
0.2304	1.0009	0.0993;
0.0721	-1.0059	0.2166;
-1.0068	0.6828	0.0778
0.3536	0.2244	0.1555;
0.4346	0.1961	-1.0003

Weights in layer 1 to hidden layer.

0.0595 -0.8082 -0.0485 0.6735 -0.5361 0.7926 0.8064 0.5266 -0.11610.6942 0.1852 -0.0111 -0.0598 -0.5115 0.5271 -0.2522 0.4763 0.3531 -0.4812 -0.8861 0.0312 0.3779 0.6132 -0.5597 0.3602 -0.2347 0.7811 -1.0010 0.6071 0.6237;
0.1423 -0.1412 0.0465 0.6377 -0.2364 0.0275 0.2143 -0.3227 -0.8427 -0.4445 0.4134 -0.6347 0.1527 0.7668 0.0124 0.4621 0.7121-0.1794 -0.6451 0.7611 0.4704 -0.2730 -0.818 0.2084 -0.5533 -0.5620 -1.0062 -0.2455 0.2023 0.5048;
-0.6364 0.7014 -0.6460 -0.0087 0.6552 -0.0335 0.6013 -0.3466 -0.7358 -0.0262 0.5125 0.0551 0.4932 0.6337 0.2263 0.3430 0.5245 0.4141 0.0234 -0.2257 0.5260 -0.0124 -0.2782 -0.3044 -0.5305 0.8776 0.0025 -0.2143 -0.3214 -0.1713;

Weights in hidden layer to output layer.

3.3 Analyze the Behavior of Hidden Neurons

3.3.1 *Based on Mean Square Error (MSE)*

In Table 4, results are arrived from the hidden neurons. The Mean Square Error (MSE) is calculated for each trail and it varies as the volume of hidden neurons varies. On the basis of table data, a graphis plotted between the average MSE and different volume of hidden neurons as shown in Figure 7.

Table 4. MSE for hidden neurons (10, 20, 30)

Simulation	No. of hidden neurons		
	10	20	30
1	0.000910	0.000942	0.000101
2	0.000918	0.000113	0.000020
3	0.000905	0.000900	0.000136
4	0.000923	0.000980	0.000150
5	0.000917	0.000143	0.000018
Average	0.000914	0.000615	0.000085

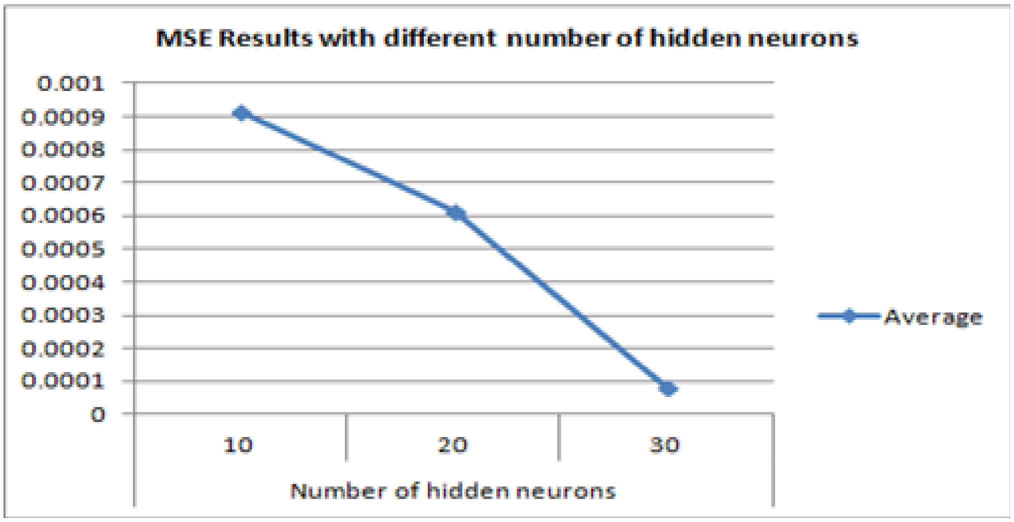

Figure 7. MSE with different number of hidden neurons.

3.3.2 Based on Epochs

Table 5 Shows the different epochs from the network for training the hidden neurons (10, 20, 30) and a graph between the average number of epochs and different number of hidden neurons shown in Figure 8.

that as we increase the hidden neurons as the number of epochs come down. Experimental value shows that 3 epochs (lowest among all the epochs) were used to train the network by implementing 30 hidden neurons.

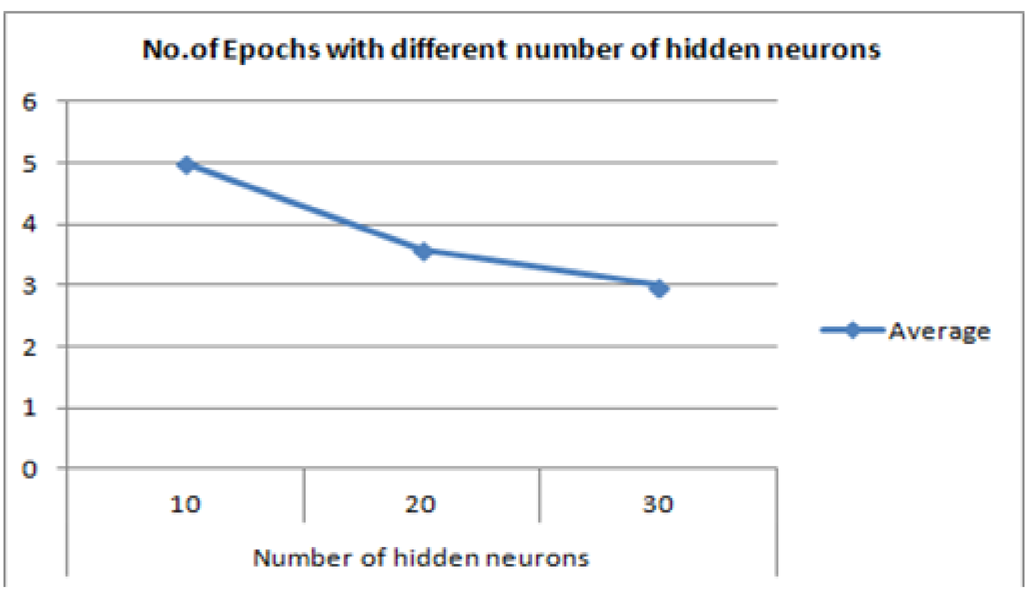

Figure 8. Number of Epochs with different number of hidden neurons.

Table 5. Number of epochs for hidden neurons (10, 20, 30)

Simulation	No. of hidden neurons		
	10	20	30
1	5	4	4
2	5	3	3
3	6	3	3
4	5	4	3
5	4	4	2
Average	5	3.6	3

4. Result and Discussion

The simulated results of experiments that shows in graphs (Figures 4-6) depicts the analysis of the performance of the three networks. The above brought up outcome indicates that the performance of each network varies depend upon weights. For network topology (32- 10-32) the best result was attained in a training session that was stopped at 5th epoch. For network topology (32-20-32), the validation and training dataset do not overlap each other. For network topology (32-30-32), the two curves validation and training overlap each other. The simulation result depicts that the network topology {32-30-32} has given the best training and performance validation.

The comparison of the performance of the networks on the basis of MSE and number of epochs shows in graphs (Figure 7-8). Table 4 consists of MSE of three networks for five different trials. Computed result shows that the differences of Mean Square Error for different topologies was small. Further, we calculate the average MSE result which depicts that the obtained output value is very much similar to the target value i.e. below 0.0001. Using 30 hidden neurons in hidden layer has given the best performance and the result was 0.000085.

Table 5 consists of number of epochs carried out by three networks for five different trials. We come to an end

5. Conclusion

In this work, The network is trained through a Back Propagation algorithm by constructing a Feed Forward Neural Network. Network performance totally depends upon the neurons volume used in the hidden layers. It also perceives that the flexibility of the network is directly proportional to the neurons volume used in the hidden layer, but on the other hand it demands more storage capacity and requires a lower implementation period to converge iterations.

6. References

1. Jain A. Statistical pattern recognition: A review. IEEE Transaction on Pattern Analysis and Machine Intelligence. 2000; 22(1):4–37.
2. Friedman M, Kandel A. Introduction to pattern recognation: Statistical, strutural, neural and fuzzy logic approaches. World Scientific; Singapore. 1999.
3 Goldberg DE. Genetic algorithms in search, optimization and machine learning. Pearson Education; 2001.
4. Drucker H, Schapire R, Simard P. Boosting performance in Neural Networks. Int J Pattern Recogn Artif Intell. 1993; 7(4):61–75.
5. Stallings W. Cryptography and network security principles and practice. 5th ed. Prentice Hall; 2011.

6. Zhang Y, Lee W, Huang YA. Intrusion detection techniques for mobile wireless networks. Wireless Networks. 2003 Sep; 9(5):545–56.
7. Zeng X, Wu S, Han L. Sensitivity-based adaptive learning rules for binary Feed Forward Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. 2012; 23(3):480–91.
8. Zurada JM. Introduction to Artificial Neural Systems. 10th ed. Jaico Publication House; 2006.
9. Shinike K. A two phase method for determining the number of neurons in the hidden layer of a 3-layer Neural Network. SICE Annual Conference; 2010 Aug. p. 18–21.
10. Zebardast B, Maleki I, Maroufi A. A novel multilayer perceptron Artificial Neural Network based recognition for Kurdish Manuscript. Indian Journal of Science and Technology. 2014 Mar; 7(3):343–51.
11. Shrivastava S, Singh MP. Performance evaluation of Feed Forward Neural Network with soft computing techniques for hand written English alphabets. Applied Soft Computing. 2011;(11): 1156–82.
12. Yegnarayana B. Artificial Neural Networks. Prentice Hall of India; 2004.13. Prochazka A, Pavelka A. Feed-Forward and recurrent Neural Networks in signalprediction. IEEE 4th International Conference on Computational Cybernetics; 2007. p. 93–6.
14. Bahl S, Dahiya D. Enhanced intrusion detection for detecting rare class attacks using correlation based dimensionality reduction technique. Indian Journal of Science and Technology. 2016 Mar; 9(11).
15. Demuth H, Beale M. Neural Network toolbox for use with MATLAB: User's Guide. Version 4. Apple Hill Drive Natick, MA: The Math Works Inc; 2002.
16. Sivanandam SN, Deepa SN. Principals of soft computing. New Delhi; Wiley-India; 2008.
17. Haykin S. Neural Networks - A comprehensive foundation. 2nd ed. Prentice Hall; 2008.