Effect of varying neurons in the hidden layer of neural network for simple character recognition

Urmi Jadhav
Computer Engineering
VES Institute of Technology
Mumbai, India
urmi.jadhav@ves.ac.in

Ashwija Shetty
Computer Engineering
VES Institute of Technology
Mumbai, India
ashwija.shetty@ves.ac.in

Abstract — Numerous advancements, which have been made in the development of intelligent system, are inspired by the biological activities. Neural network is one such complex *adaptive* system that is able to capture and represent complex input/output relationships. This paper gives an overview of the neural network and further highlights the effect of noise and the number of neurons on the performance of the neural network for simple character recognition application. The number of neurons in the hidden layer is varied during simulation and the effect is observed in the percentage of errors in the recognition of characters

Keywords—*neural networks, ANN, neurons, hidden layer, feedforward network*

*****

I. INTRODUCTION

The field of computation has a lot to gain from neural networks. The capability of neural networks to learn by example makes them extremely adaptable and robust. Moreover there is no need to design an algorithm to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very relevant for real time systems because of their parallel architecture that ensures fast response and computational time.

A neural network typically involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. A neural network is initially "trained" or fed large amounts of data and rules about data relationships. The network is guided by the program on how to behave in response to an external stimulus or can initiate activity on its own.

Neural Networks attempt to bridge the gap between the capabilities of computers and human brain by modelling aspects of information in the brain in a highly simplified way.

Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.

The motivation for research in neural network is due to two prime desires: to obtain a better understanding of the human brain and to develop computers that can deal with abstract and poorly defined problems.

II. ANN

An Artificial Neural Network (ANN) is inspired from the biological nervous system i.e. the brain. ANN is an information processing model composed of a large number of interconnected processing nodes (neurons) working together to solve specific problems. ANNs, like the human brain, learn by example. An ANN is designed for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones as shown in figure 1.

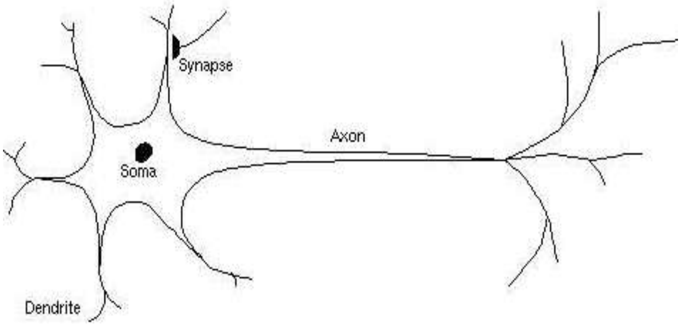

Figure 1.A biological neuron [2]

The perceptron is a mathematical model of a biological neuron. Figure 2 illustrates the working of the perceptron. While in actual neurons the dendrite receives electrical signals from the axons of other neurons, in the perceptron these electrical signals are represented as numerical values. At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. This is also modelled in the perceptron by multiplying each input value by a value called the weight [2].

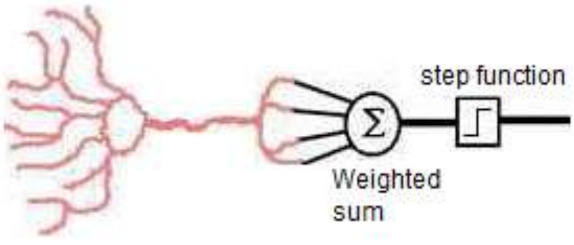

Figure 2.An artificial neuron (perceptron) [2]

III. STRUCTURE OF ANN

Artificial neural networks consist of several neurons. These neurons receive information from neurons of the previous layer and pass them on to the next layer. Neurons are affiliated with each other by edges. The intensity of the connection between two neurons is represented by a weight value and this weight value stores the knowledge of the neural network. The learning procedure of a neural network is defined by its change of weight values between the neurons. How this change of weights happens depends on the used learning rule. Before the learning process begins the weight values are randomized [3]. Figure 3 shows the structure of a basic ANN which includes an input layer, a weight matrix and the output layer.

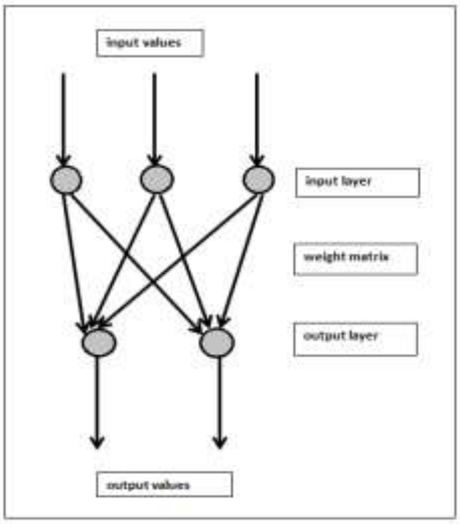

Figure 3.Structure of a basic ANN

The architecture of an ANN consisting of a hidden layer can be seen in figure 4. This ANN incorporates a hidden layer of neurons which process the data passed on from the input layer and forward the information to the output layer.

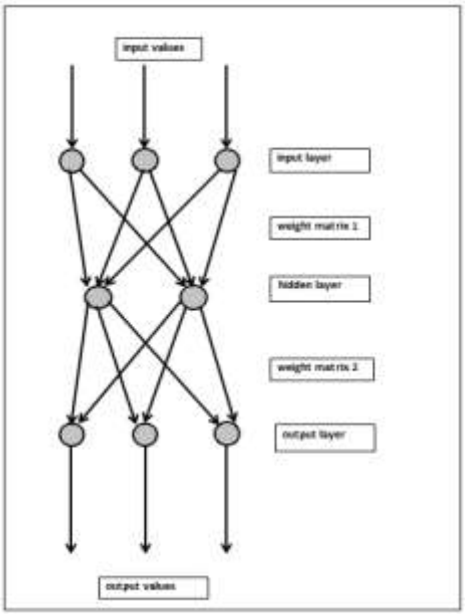

Figure 4.Structure of ANN with a hidden layer

The Input neurons receive the signals like stimuli or patterns. Every Input processes the signals by a separate weight value. The weight values get passed to the hidden layers which contain an internal representation of the environment. The signals pass through the layers and the Output neurons pass the output signals to the environment.

IV. FEEDFORWARD NETWORK

Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when the network acts as a classifier, there is no feedback between layers. Hence, such a network is called a feedforward neural network.

Feed-forward networks have the following characteristics:

1. Perceptrons are arranged in layers, with the first layer taking in inputs and the last layer producing outputs. The middle layers have no connection with the external world, and hence are called hidden layers.
2. Each perceptron in one layer is connected to every perceptron on the next layer. Hence information is constantly "fed forward" from one layer to the next, and this explains why these networks are called feed-forward networks.
3. There is no connection among perceptrons in the same layer [8].

Figure 5 illustrates an example of a 4-layered network with, from top to bottom: an output layer with 4 units, 3 *hidden* layers with 3, 2 and 3 units, respectively. The network has 4 input units.

The 4 inputs are shown as circles and these do not belong to any layer of the network (although the inputs sometimes are considered as a virtual layer with layer number 0). Any layer that is not an output layer is a *hidden* layer. This network therefore has 3 hidden layer and 1 output layer. The figure also shows all the connections between the units in different layers.

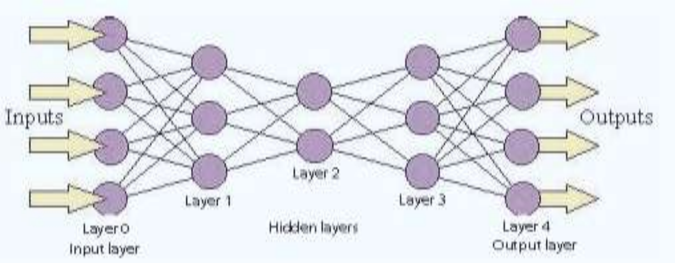

Figure 5.A 4-layered feedforward network [8]

The operation of this network can be divided into two phases:

1. The learning phase

During the learning phase the weights in the feedforward-network will be modified. All weights are modified in such a way that when a pattern is presented, the output unit with the correct category will have the largest output value.

2. The classification phase

In the classification phase, the weights of the network are fixed. A pattern, presented at the inputs, will be transformed from layer to layer until it reaches the output layer. Now classification can occur by selecting the category associated with the output unit that has the largest output value [4].

V. TRAINING THE NEURAL NETWORK

In this paper we have considered a neural network that performs simple character recognition. Two neural networks Network 1 and Network 2 are simulated using the Neural Networks toolbox in Matlab. Both Network 1 and Network 2 use a feedforward neural network setup for pattern recognition with 25 hidden neurons. Network 1 is trained without noise. Network 2 is trained with noisy copies of the letters so that it will not only recognise perfectly formed letters but also noisy versions of the letters. The bitmap pattern generated by Network 1 and Network 2 is as shown in fig. 6 and fig. 7.

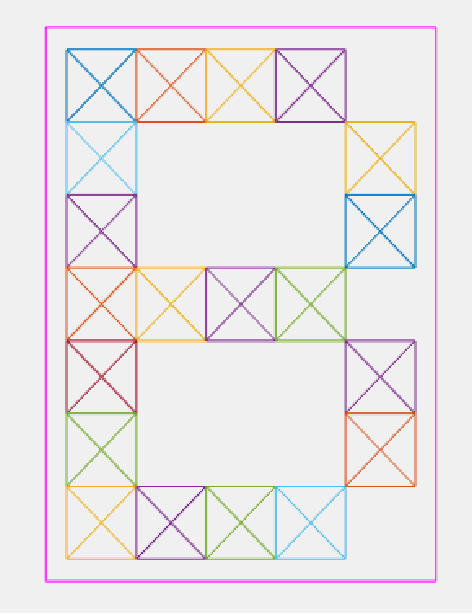

Figure 6.Bitmap pattern of letter B (without noise)

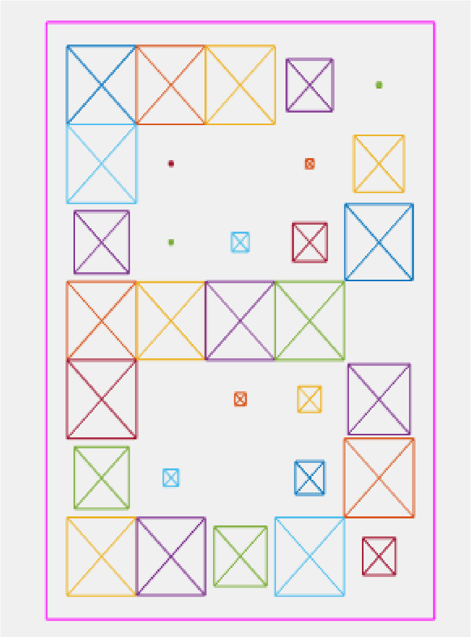

Figure 7 Bitmap pattern of letter B (with noise)

The training set is used to update the network; the validation set is used to stop the network before it over fits the training data, thus preserving good generalization. The test set acts as a completely independent measure of how well the network can be expected to do on new samples. Training stops when the network is no longer likely to improve on the training or validation sets [7].

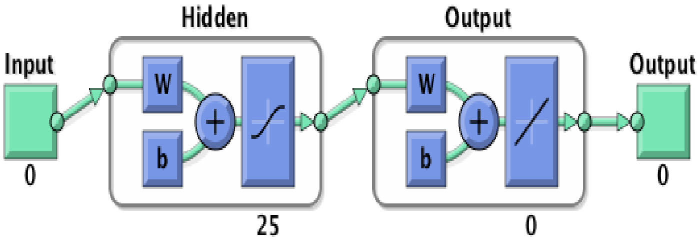

Figure 8.A two-layer feedforward network with sigmoid hidden neurons and linear output neurons. [7]

After simulation, it is found that Network 1 which was trained without noise has more errors as compared to Network 2 which was trained with noise. This result is expected because Network 1 was trained to recognise only perfectly formed letters so it does not have the ability to generalise and identify the letters which differ slightly from the standard bit pattern. While Network 2 is able to identify a wider array of input letters since it was trained with noisy letters and hence produces less recognition errors. This is further verified by the graph (figure 8) that shows that the error rate for Network 1 is greater than the error rate for Network 2.

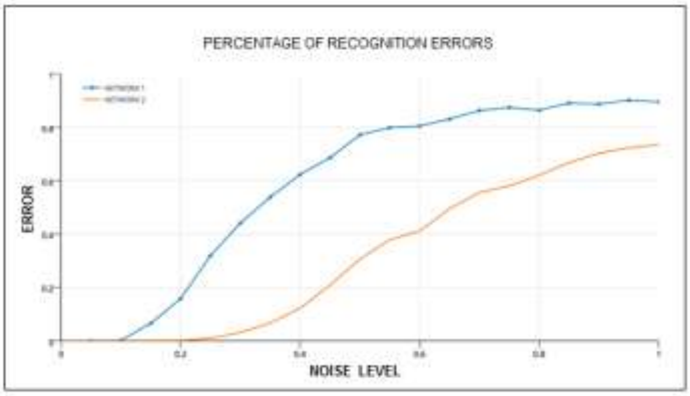

Figure 9.Error VS Noise Level for Network 1 and Network 2

VI. FACTORS AFFECTING THE PERFORMANCE OF NEURAL NETWORK

There are a number of factors that affect the performance of neural network such as the number of neurons in the hidden layer, the number of hidden layer, learning rate etc. In this paper we have carried out the simulation that depicts the effect of the number of neurons in the hidden layer on the neural network.

A. EFFECT OF THE NUMBER OF NEURONS IN THE HIDDEN LAYER

We have studied the effect of the number of neurons in the hidden layer. The network was tested with 5, 10, 15, 20, 25 and 50 neurons in single hidden layer. The figure 10 shows that the network did not coincide with a stable point when the network was simulated with neurons up to 20. The reason for this is that the number of neurons that take the data from the input grid were less, and hence they failed to convey the correct information to the next layers. The results were acceptable with 20 neurons in the hidden layer, however by increasing the number of neurons further, no further improvement was observed in the reduction of error.

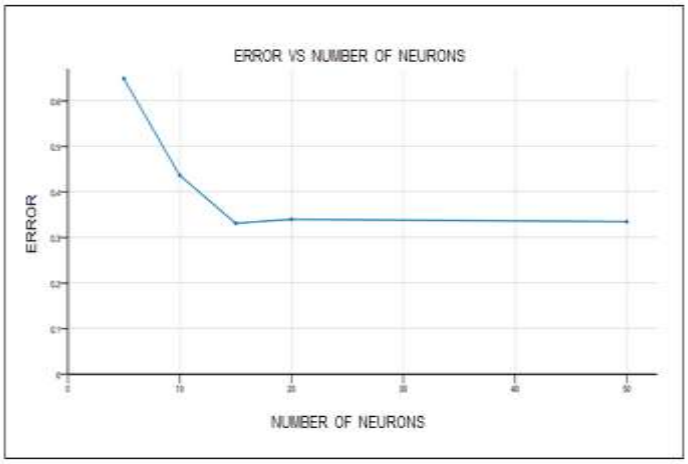

Figure 10.Error VS Number of neurons in the hidden layer

The figure 11 illustrates the impact of the number of neurons in the hidden layer on the error in the network with respect to noise. When the number of neurons in the hidden layer is only 5 the network is susceptible to high rate of errors in the recognition of characters. As the number of neurons in the hidden layer increases, the error rate drops accordingly with 50 neurons having the lowest error rate.

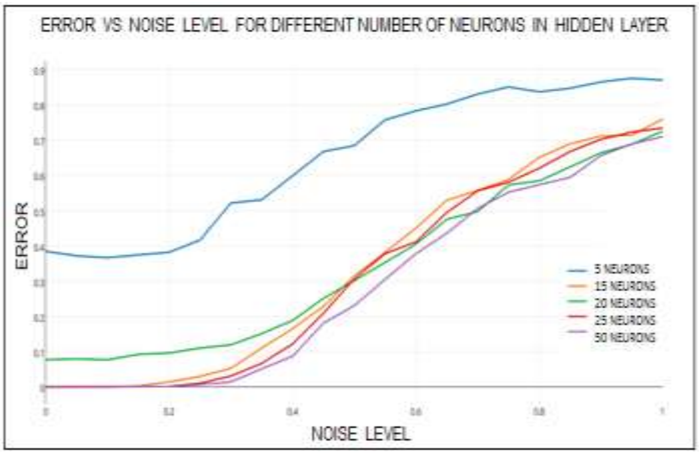

Figure 11.Error VS Noise Level for varying number of neurons in hidden layer

VII. CONCLUSION

The character recognition application in neural networks has received a lot of attention. It is simple enough that many approaches achieve partial success, but complex enough that there are no perfect solutions. Many successful commercial products have been based on this problem, such as reading the addresses on letters for postal routing, document entry into word processors, etc. The simulation results presented in the paper indicate that the neural network architecture composed of single hidden layer with 20 neurons is able to identify both perfect as well as noisy characters effectively with minimum error in last epoch as seen in figure 10. Increasing the number of neurons further only seems to increase the complexity of the network as well as the training time for the network, and is found to be unsuitable exhibited by visual findings in figure 11. The effect of these parameters in other applications will be instrumental in improving the efficiency of various neural network applications.

VIII. REFERENCES

[1] C. Stergiou, “What is a Neural Network”, https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/cs11/article1.html

[2] https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

[3] http://www.cogsys.wiai.unibamberg.de/teaching/ws1112/km/practice/ANNs.pdfhttp://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html

[4] http://www.fon.hum.uva.nl/praat/manual/Feedforward_neural_networks.html

[5] S.N.Sivanandam, S.N.Deepa "Principles of Soft Computing" Second Edition, Wiley Publication

[6] http://www.dspguide.com/ch26.html

[7] http://in.mathworks.com/products/neural-network/features.html#training-algorithms

[8] http://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html