International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4723-4728

Analysis on Multiple Hidden Layer Complexity of BPNN

Stuti Asthana[1], Dinesh Goyal[2] and Amitkant Pandit[3]

[1] Ph.D. Research Scholar, CSE Department,
[2] Principal, Suresh Gyan Vihar University, Jagatpura, Jaipur, Rajasthan, India.

[3] SECE, Shri Mata Vaishno Devi University, Katra, Jammu, India.

[1]Orcid: 0000-0003-4572-710X

Abstract

Artificial Neural Network plays a vital role in Pattern Matching. This analysis is beneficial for pruning the neural network architecture additionally as factors governing the training time and overall performance of Multiple hidden layer back propagation neural network. Now a days, artificial intelligence have terribly very important role in day to day life. whether it's Finger print recognition or Face detection, artificial intelligence found its usability. what is more accuracy is becoming a chief criteria of competition among numerous corporations that develop applications and product based on artificial intelligence. Multiple hidden layers ar used wherever the matter complexness is high. Also, it's troublesome to make a decision the amount of neurons in each hidden layer. Here we have a tendency to created associate degree approach in pruning a multiple hidden layer fulfilling all "Rule of Thumb" [2] until utilized in making associate degree optimized design in neural network. additionally time complexness analysis of multiple hidden layers with single/dual hidden layer design is bestowed in this paper.

Keywords: BPNN , Multiple hidden layers, Neural Network, Time Complexity.

INTRODUCTION

Artificial Neural Network plays an important role in machine learning. A common architecture of neural network include input layer, hidden layer and output layer. Basically input layer consist of 'N' number of Neurons where N is the number of inputs received by the neural network. Output neurons are directly proportional to the type of expected outcomes. For example, in character recognition system, the number of parameters which uniquely detects the symbol, can be considered as main criteria for selecting number of input neurons. In the same manner, number of output neurons may be 26 if English language is chosen if any newbie programmer would like to design the neural network.

Still the number of neurons in hidden layer is taken by estimation and there is not any sure shot formula which can calculate the exact Number of Neurons. For a better output and less training time, the number of neurons in all the three layers must be assumed perfectly elsewise Neural Network will undergo Overfitting or Underfitting Problem. In Overfitting, A network which is too closely fitted to the data will tend to have a large variance andhence give a large expected generalization error.

If the variance is decreased by smoothing the network outputs, then another issue raises in which the expected generalization error is large again. This problem is termed as Underfitting. The Overfitting problem can be easily resolved by stopping the Training data early.

But this is to be done manually. In order to Avoid Underfitting, enough hidden units to represent to required mappings. So pruning appropriate number of Neurons is most important in all the three layers.

As mentioned by many researchers that one or two hidden layers are sufficient to resolve any complex problem but in this paper, an approach is made in adding five hidden layer in neural network architecture with proper proportionality among then and then checking the resulting outcomes. The numbers of hidden layers neurons are also important in the whole architecture because they revert the output quick and accurate. Previously Hidden layer Units were selected by following thumb rule. The no. of H/L neurons lies in 2/3 (nearly 70% up to 90%) of size of input layer. If this is insufficient then number of output layer neurons can be added later on [3].

Second rule, the number of hidden layer neurons should be less than twice of the number of neurons in input layer. [8] Last rule the size of the hidden layer neurons is between the input layer size and the output layer size [9]. Beside all these thumb rule, If the training data is linearly divisible than there's no ought to use hidden because the activation perform may be enforced to input layer which might solve the problem. But just in case of issues that deals with absolute call boundary to absolute accuracy wit rational activation functions then one needs to use 2 or 3 hidden layer.

The Artificial Neural Network (ANN) is a mathematical model composed of interconnected artificial neurons. The pruning of neural network architecture depends on the number

of neurons in all layers and this is the most important part of the system. The performance of the multilayer artificial neural network greatly depends on the architecture of hidden neurons. Many rule-of-thumb methods have been suggested for determining the number of neurons in hidden layers.

Researches have been made to train the artificial neural network with lesser number of hidden layers but this cause "Underfitting" problem. Also if more numbers of neurons are used unnecessarily, then "Overfitting" may arise. In both cases, accuracy will be effected negatively effecting training time badly. In order to get solution of this problem, proportionality among hidden layers including Input and Output layers, plays a vital role in achieving high accuracy. This paper focuses on pruning neural network architecture by hidden layers proportionality.

Neural networks are the interconnected complex structure of neurons which have the capability of performing various tasks in the same manner as human mind can do. They can also perform computation, logical inference & pattern recognition. Neural networks can be of any type based on the number of hidden layers, input & output neurons, the criteria of perceptrons etc. In very simple terms the technology which is based on the computation of neurons of brain is called as Artificial Neural Network or Neural Network.

Among the 23 official languages spoken in India, apart from Maharashtra Marathi is also spoken in Daman and Diu, Dadar and Nagar Haveli. The recognition rate of this language in this method is 97% [2]. 60 phonemic letters are used by this language and they are divided into 3 groups swear, vyanjan & ank i.e vowels (13 letters), consonants (38 letters) & numbers (10 digits). Developing offline & online OCR technology for Marathi handwritten characters & numbers recognition is a very difficult task for researchers because handwriting of each person is mimetic.

Properties of Neural Networks:

- The NNs are very much capable for display mapping i.e they can map input patterns to the desired output patterns.
- Neural Networks have learning capability also i.e they learn by examples. They can recognize new instances with the help of example.
- NNs may predict new outcomes from the past trends i.e they possess the ability to generalize.
- NNs model real world problems , they are very much robust & fault tolerant. Noisy patterns or any type of disturbance doesn't affect the result of NN.
- NNs process the result in undistributed manner and in parallel at a high speed.

Characteristics of Neural Networks:

Neural network basically refers to the interconnection between neurons present in different layers. It is a very complex structure and initially three layers are present in the neural network input layer, hidden layer and output layer. The value of the weight of the neurons is stored in 'synapses'. The input layer transfers the input neurons via synapses to the hidden layer. The hidden layer then transfers the hidden neurons to the output layer with the help of some more synapses.

An ANN can be defined based on the following three characteristics:

- The first one is its Characteristics i.e the number of nodes & the number of layers.
- For updating the weights of the neurons a learning mechanism is applied.
- The activation function which is been used by various neurons.

Ravindra S. Hegadi & Parshuram M. Kamble [2] recognized Marathi Handwritten Numerals using Multilayer Feedforward Neural Network. In their work they came out with a technique to recognize the handwritten Marathi numerals. In this method the noise is been removed from the input image while pre-processing in the form of tiny dots. By applying morphological dilation operation, they had taken care of the breakages at the low gradient parts of the numerals due to edge detection. Mr. Hegadi & Mr. Kamble introduced resizing method used in this work that will provide better representation to every pixel in the original numeral image.

The recognition accuracy of proposed work is 97%. In the future work they had decided to test this algorithm on large database by including some additional features in training the network.

COMPLEXITY ANALYSIS OF MULTIPLE HIDDEN LAYER

M Time complexity plays a vital role while recognition as well as training with the data set. For calculating the effective time taken for training the data set by the neural network with different increasing hidden layers, a 1000 Random Multiscript Numerals are taken in to account. Before starting with the training time calculation lets check how many "weights" (interlinks between the neurons nodes) need to be updated or balanced for any dataset.

For Single Hidden Layer:

No. of Weights = 150 (Input layer) x 102(hidden layer i.e. 70% of Input) + 102 x 16 (O/P Layer)

= 15,300 + 1,632

= 16,932 Weights need to be balanced in an iteration.

For Two Hidden layer Approach

No. of Weights = 150 (Input layer) x 102 (First hidden layer i.e. 90% of Input) + 102 x 74 (Second Hidden layer) + 74 x 16 (O/P Layer)

= 15,300 + 7548 + 1184

= 24, 032 Weights need to be balanced in an iteration.

For Proposed Five Hidden Layers Approach

No. of Weights = 150 (Input layer) x 102 (First hidden layer i.e. 90% of Input) + 102 x 74 (Second Hidden layer) + 74 x 49(Third) + 49 x 37(Forth) + 37 x 29(fifth) + 29 x16 (O/P Layer)

= 15,300 + 7548 + 3626 + 1813+1073 + 464

= 29,824 Weights need to be balanced in an iteration.

TIME COMPLEXITY CALCULATION:

For Two Hidden Layer training with 70 Numerals : 10.5 seconds

Number of weight adjusted : 24, 032

Average Time to adjust one neuron weight :0.00004369 or 43.6 micro sec

For Five Hidden Layer training with 70 Numerals : 14 Seconds (Matlab 7.0 platform)

Number of weight adjusted : 29,824

Average Time to adjust one neuron weight :0.00004694 Sec or 46.9 micro sec

From the above calculations this has been observed that here by increasing the number of hidden layers up to five, we had tremendously increased the overall neural network's architecture complexity. This also led to increased in training time for the data set. If we estimate this time complexity with respect to the number of weights (29,824 weights overall in five hidden layer) updated in the process then as compared to normal architecture which contain dual hidden layer (with less accuracy) then approximately 20% weights need to be adjusted additionally. This increased weights calculation will required additional 25% (approx..) time during the Training Phase . However, in the Testing Phase, the time taken to detect the symbol remains same as in traditional Analysis.

This factor will beneficial in implementing multiple languages detecting characters/symbols hardware firmware devices because they had been trained during their making in the electronic industry. This factor will beneficial in implementing multiple languages detecting characters/symbols hardware firmware devices because they had been trained during their making in the electronic industry.

Table 1: Overall Complexity Analysis

	Criteria	Dual Hidden Layer	Five Hidden Layer
1.	Number of Weights Adjustment per Iteration	**24, 032**	**29,824**
2.	Time taken to train 70 numeral data set	10.5 Sec	14 Sec
3.	Accuracy	Up to 97%	98.2% - 98.7%
4.	Require Ideal Condition[2]	Yes	Even work on Handwritten no.
5.	Multiscript Recognition	less Accurate	Good Recognition
6.	Inclusion of more scripting language possible	No	Yes

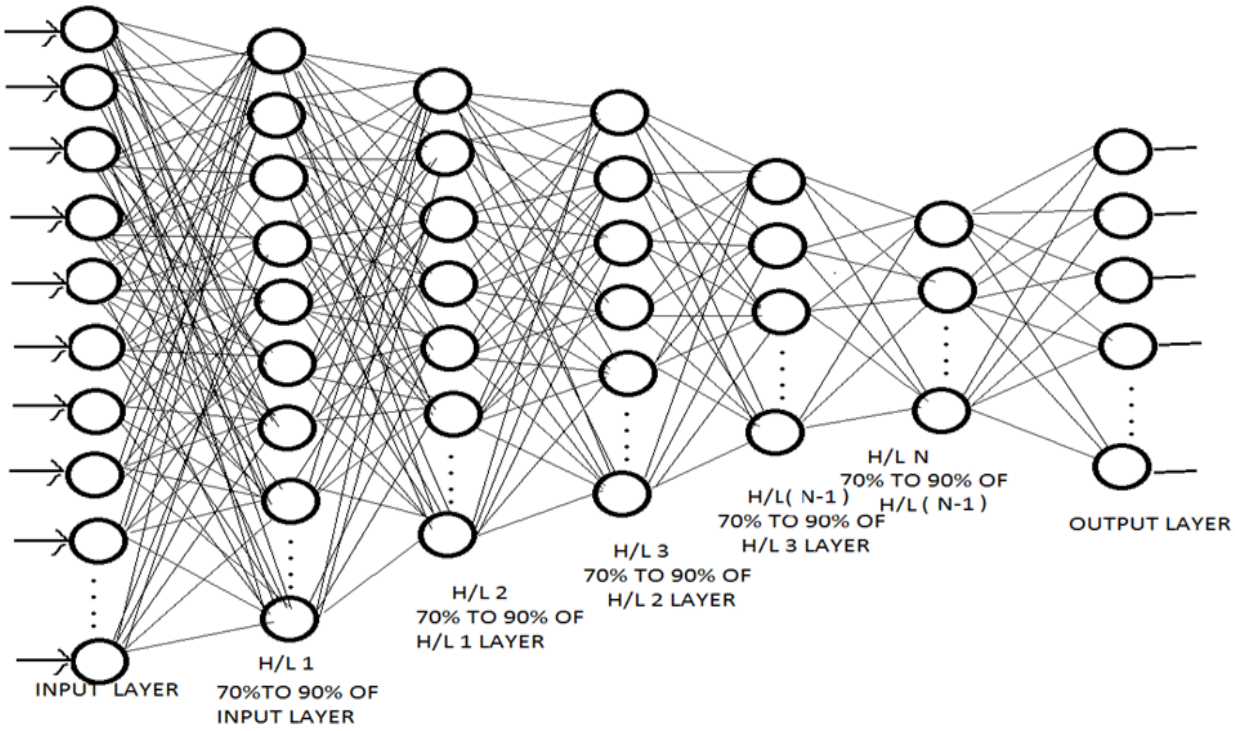

Figure 1: Optimized Architecture of Multiple Hidden Layer

In the below sample figure 2(a), three different scripting numeral viz. Hindi, Tamil and English (656565) were scanned and fed to the proposed work as input. It has been observed that all Numerals are identified correctly using five hidden layer.

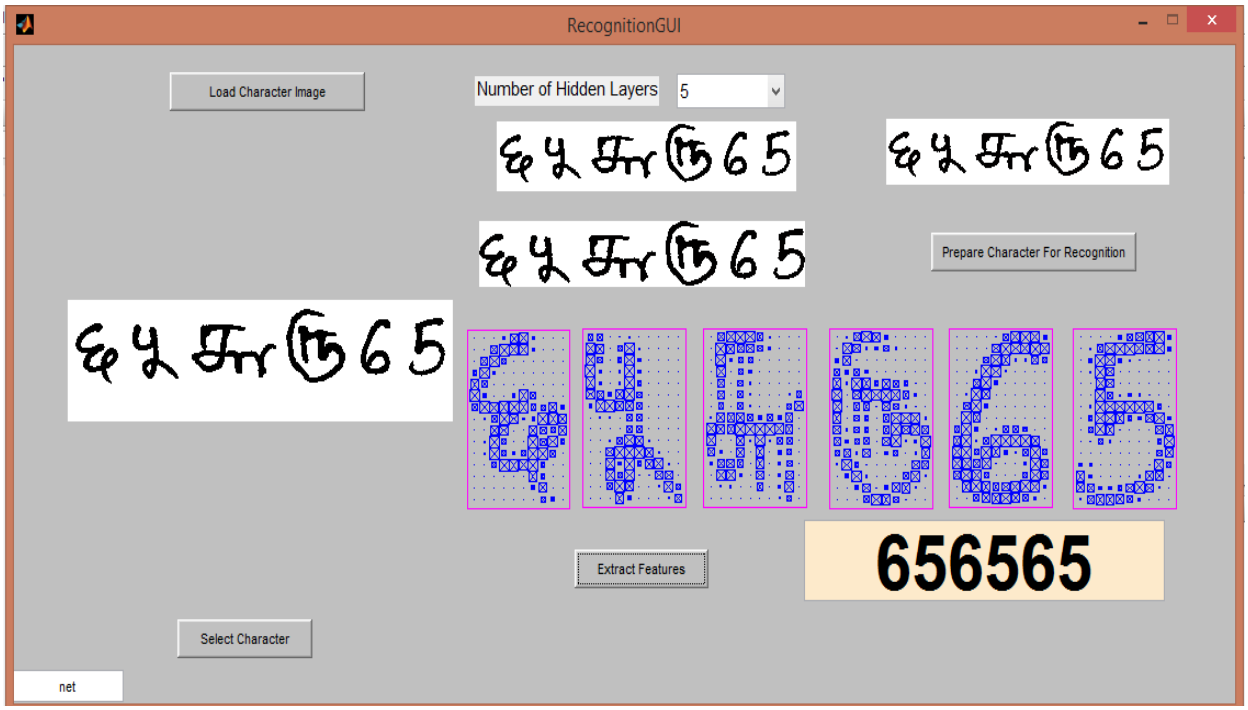

Figure 2(a): MATLAB software output having five hidden layers

As compared to double hidden layer approach, the neural network architecture get confused totally while recognizing three different language yielding wrong output Figure 2(b).

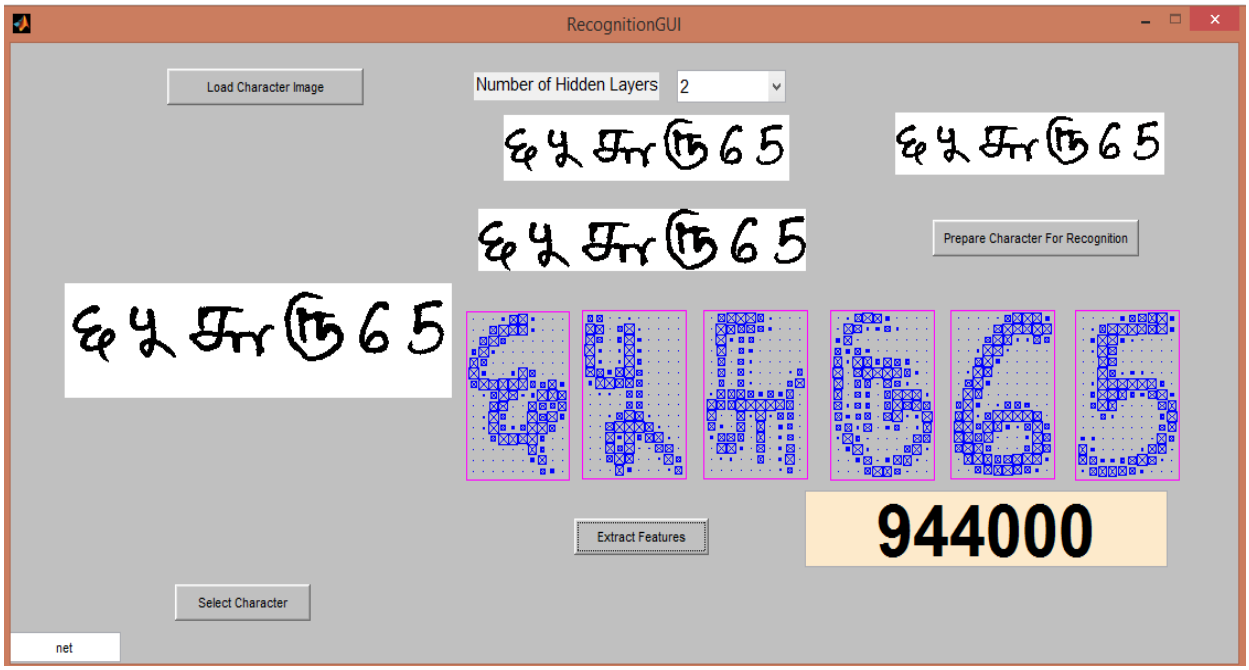

Figure 2(b): MATLAB software output having two hidden layers

PROPOSED OPTIMISED MULTIPLE HIDDEN LAYER BPNN ARCHITECTURE

The five hidden layer neural network architecture is shown in the Fig.2. The architecture shown is five hidden layers fully interconnected with different weights. The first layer is hidden layer and the input to this layer is present in binary form viz 0, 1. It is assumed that every numeral has the size of 15*10 so the input layer consists of 150 neurons. Now seen the architecture, the input pattern will have the form as shown. Say for character 'C' the pattern will be [1,1], [1,2], [1,3], [1,4], [1,5], [2,1], [3,1], [4,1], [5,1], [6,1], [7,1], [7,2], [7,3], [7,4], [7,5] will contain 1 bit and the rest will contain 0. In this way the input values are framed & fed to the network as input. The output layer has 16 neurons in order to support Unicode. As specified in the thumb rule, 3/4th number of neurons must be in hidden layer and the first hidden layer has 70-75 % neurons. Similarly in second hidden layer, the first hidden layer will act as input layer for it. Hence 75% of 102 is nearly equals 74 and so on for third, fourth & fifth layer. A very minute difference of 2 or 3 neurons can be adjusted while training, so this helps in better architecture of the network.

70-75% neurons of input layer = first hidden layer

70-75% neurons of first hidden layer = second hidden layer

70-75% neurons of second hidden layer = third hidden layer

70-75% neurons of third hidden layer = fourth hidden layer

70-75% neurons of fourth hidden layer = fifth hidden layer

Total neurons in hidden layers = 291 (less than twice of input)

This is proved theoretically if we consider rule of thumb that the network is pruned. But many researchers suggest that least number of hidden layers must be used to save training time & complexity but this will not result correctly when accuracy is the major criteria.

CONCLUSION

Hidden layer plays an important role in the performance of Back Propagation Neural Network especially in the case where database is large and training time need to be reduced. A lot of work has already been done with single hidden layer and two hidden layer and it has been found that increasing the hidden layer also increase the accuracy performance slightly. But the drawback of increasing hidden layer is that it also increase the network complexity.

The proposed method for the Neural Network Architecture using multilayer perceptron algorithm showed the remarkable enhancement in the performance when five hidden layers are used in proper proportionality. If the accuracy of the results is a critical factor for an numeral recognition application, then the network having many hidden layers should be used but if training time is a critical factor then the network having single hidden layer (with sufficient number of hidden units) should

be used. The number of hidden layers is also proportional to the number of epochs. This means that as the number of hidden layers is increased, the training process of the network slows down because of additional branching of weight adjustment. However, the training of the network is more accurate if more hidden layers are used. This accuracy is achieved at the cost of network training time. In Back propagation neural network, hidden layer plays the most crucial role. It is been concluded that optimizing the number of hidden layers in a neural network more is the accuracy and overall performance.

ACKNOWLEDGEMENT

Author would like to thanks Dr. Amitkant Pandit and Dr. Dinesh Goyal for their suggestions, guidance and support.

REFERENCES

[1] Rakesh Kumar Bhujade, Amitkant Pandit and Savita Shiwani, "Improving Accuracy in Back Propagation Neural Network by Pruning Proportionality in Multiple Hidden Layers", Ciência e Técnica Vitivinícola Journal (ISSN: 0254-0223), Volume 31, Issue 2, Feb 2016, PORTUGAL.

[2] Ravindra S. Hegadi & Parshuram M. Kamble, 'Recognition of handwritten Marathi numerals using Multilayer feed forward neural network', 2014 World Congress on Computing & Communication Technologies.

[3] Rakesh Kumar Bhujade, Amitkant Pandit and Naveen Hemrajani, Stuti Asthana "A Novel Approach of Five Hidden Layer Proportionality in Back Propagation Neural Network", Fifth International Joint Conference on Advances in Engineering and Technology, AET 2014.Kochi, India.

[4] Rakesh Kumar Bhujade, Amitkant Pandit, "An Analysis of Effect of Proportionality among Hidden Layers in Standard Back Propagation Neural Network", International Journal of Computer Technology and Electronics Engineering, FLORIDA (An ISO 9001:2008 Certified Journal, ISSN 2249-6343(Online)) Volume 5, Issue 4, August 2015.