Deep Neural Networks : Implementation and Analysis

Akash Soni (1401047), Jay Joshi (1401005), Mihir Gajjar (1401076), Nishi Shah (1401099), Shruti Agrawal (1644007)

***Abstract*—Our work aims to acquire deep understanding and knowledge about Deep Neural Networks (DNN) by implementing its basic applications using TensorFlow library to generate some practical results. Our work projects the working of Deep neural networks in identifying a data set similar to expected output data set with maximum achievable accuracy based on techniques of recursive cost minimization approach for training neural network for particular number of epoch.**

***Index Terms*—TensorFlow, Deep Neural Network, Epoch**

I. Introduction

Deep neural network has one or more hidden layers consisting of nodes, unlike simple artificial neural network where there is only one hidden layer of many nodes.Nodes of network are referred to as neurons.They store some data within itself and fires output when input satisfies certain threshold. Nodes of these hidden layers assign unique weights to each and every connected node of another layer.Initially, all the nodes are assigned random weights. These weights keeps on changing its value as deep neural network trains itself after every epoch from training input data. More number of hidden layers increases deep learning for the given neural network.

II. Implementation

Deep Neural Network has been implemented using tensorflow with Adam algorithm as well as with gradient descent algorithm to minimize the cost. The neural network is trained using the training examples of the MNIST data set and the classification of the testing examples of the MNIST data set is done using the trained DNN and the accuracy of the classification is shown in the below graphs with varying various parameters.

A single neuron neural network is implemented and trained with training sets containing three elements in a training example and the output is the first element of the training example. And then the network is tested with a testing set.

III. Implementation Result

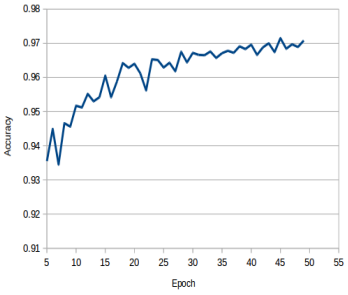

Figure: 1
Epoch vs Accuracy

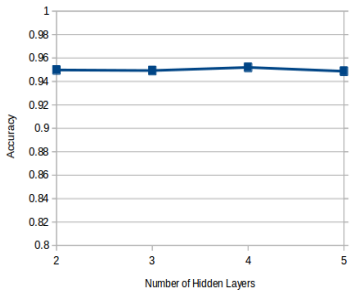

Figure:2
No. of Hidden Layers vs Accuracy

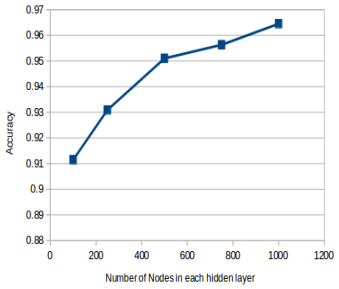

Figure: 3
No. of Nodes vs Accuracy

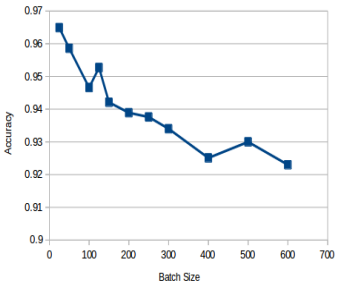

Figure:4
Batch Size vs Accuracy

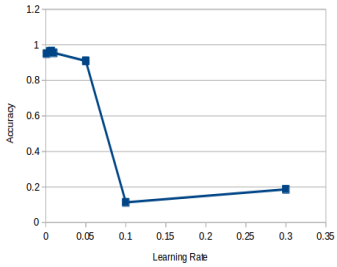

Figure: 5
Learning Rate vs Accuracy

```
Random starting weights:
[[-0.16595599]
 [ 0.44064899]
 [-0.99977125]]
New weights after training:
[[ 9.67299303]
 [-0.2078435 ]
 [-4.62963669]]
Considering new situation [1, 0, 0]:
[ 0.99993704]
```

Figure:6
A Simple Neural Network

IV. Interpretation and Conclusion

While calculating one parameters other parameters are kept constant and Adam Optimizer is used to minimize the cost function.

Defined Parameters :

1. Number of epochs = 10
2. Number of Nodes in a hidden layer = 500
3. batch size = 100
4. Learning rate = 0.001
5. Number of hidden layers = 3

- **Epoch vs Accuracy :** Accuracy increases as number of iterations(epoch) increases because as the forward and backward propagation increases then the weights adjust themselves closer to the optimal values due to which resulting error minimizes.
- **No. of hidden Layers vs Accuracy:** As the number of hidden layers increases then more the number of training pairs will be required to train the network because more connections are established and more nodes needs to be adjusted and hence there is a certain amount of decrease in accuracy as the number of hidden layers increases.
- **No. of nodes in each hidden layer vs accuracy :** Increase in number of nodes in a given hidden layer leads to proportionate increase in accuracy because more the number of nodes, more the weights and more accurate will be the result.
- **Batch size vs accuracy:** Batch size is the size of data fed to network in a given time interval. As batch size increases accuracy decreases for Learning neural network.
- **Learning rate vs accuracy:** Initally when the learning rate is low then the cost function can converge and reach near to the global minima but as the learning rate increases then the cost function diverges and hence the accuracy is very low.
- **A Simple neural network:** The weights are randomly initiated. The neural network is then trained with training pairs containing three values in a training example and the output value is the first element in the training example. After training the modified weights are shown. Then a new example is tested and the output is almost similar to the first element in the training example.

References

[1] Michael A. Nielsen, 'Neural Networks and Deep Learning', 2017. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap6.html. [Accessed: 26- Mar- 2017]

[2] Sentdex, Neural Network Model - Deep Learning with Neural Networks and TensorFlow ', 2016. [Online]. Available: https://youtu.be/BhpvH5DuVu8 [Accessed: 26- Mar- 2017]