ELL409

Machine Intelligence and Learning

Assignment – 1 Report

Submitted by:

Karan Mittal(2017ME20671)

Handwritten digits recognition using Neural Network

Implemented a neural network using numpy in Python

Features of the neural network model (by hit and trial):

80% of the train data set was used to train the model

Remaining part of the set was the validation set which was used to calculate the accuracy of the model

No. of units in the input layer -> 28*28 = 784

No. of units in the hidden layer -> 75

No. of units in the output layer -> 10

Activation function for layer 1 weights -> Sigmoid

Activation function for layer 2 weights -> Softmax

Learning Rate: 0.01

Initial weights and biases: Between -0.001 and 0.001

Epochs: 60

Batch Size: 1

Accuracy Obtained: 90.714%

By Changing the Activation Function for layer 1 weights:

Activation Function -> Tanh

Accuracy Obtained: 92.42%

Activation Function -> ReLU

Accuracy Obtained: 91%

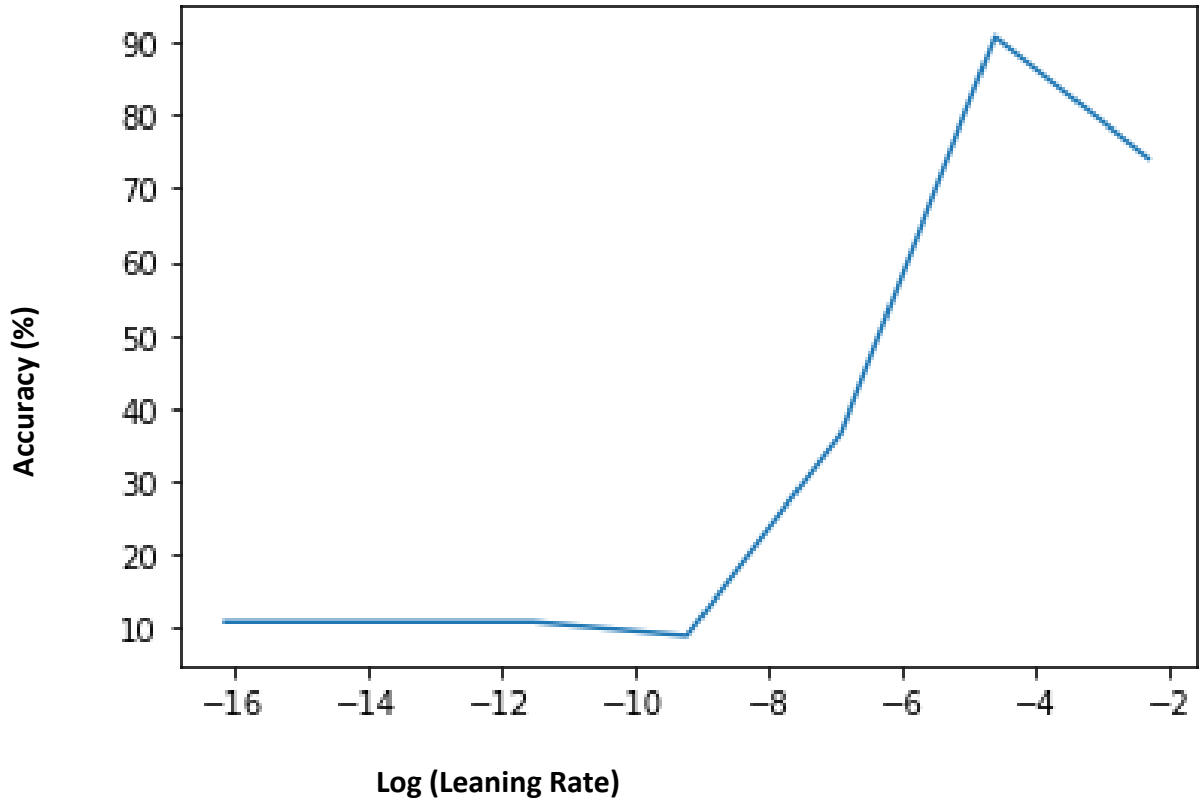
Change in accuracy* by varying the hyperparameters

*Calculation of accuracy has been done on the validation set

Plotted the Graphs using matplotlib

Learning Rate

Plotted Accuracy v/s Log (Learning Rate)

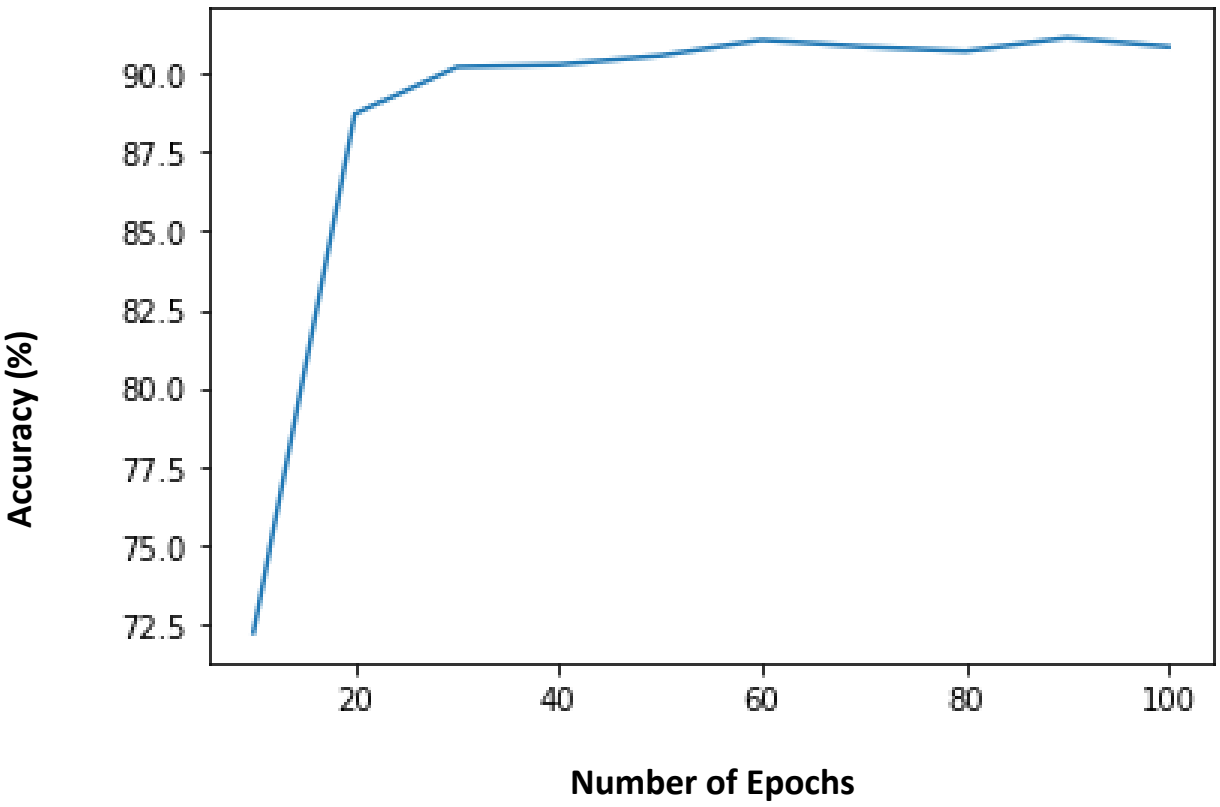


Accuracy increases with the increase in Learning Rate but falls down after a certain point because large value of Learning Rate causes oscillations as it jumps over the minima during every iteration.

Optimum value of Learning Rate for Neural Network is 0.0001

Number of Epochs

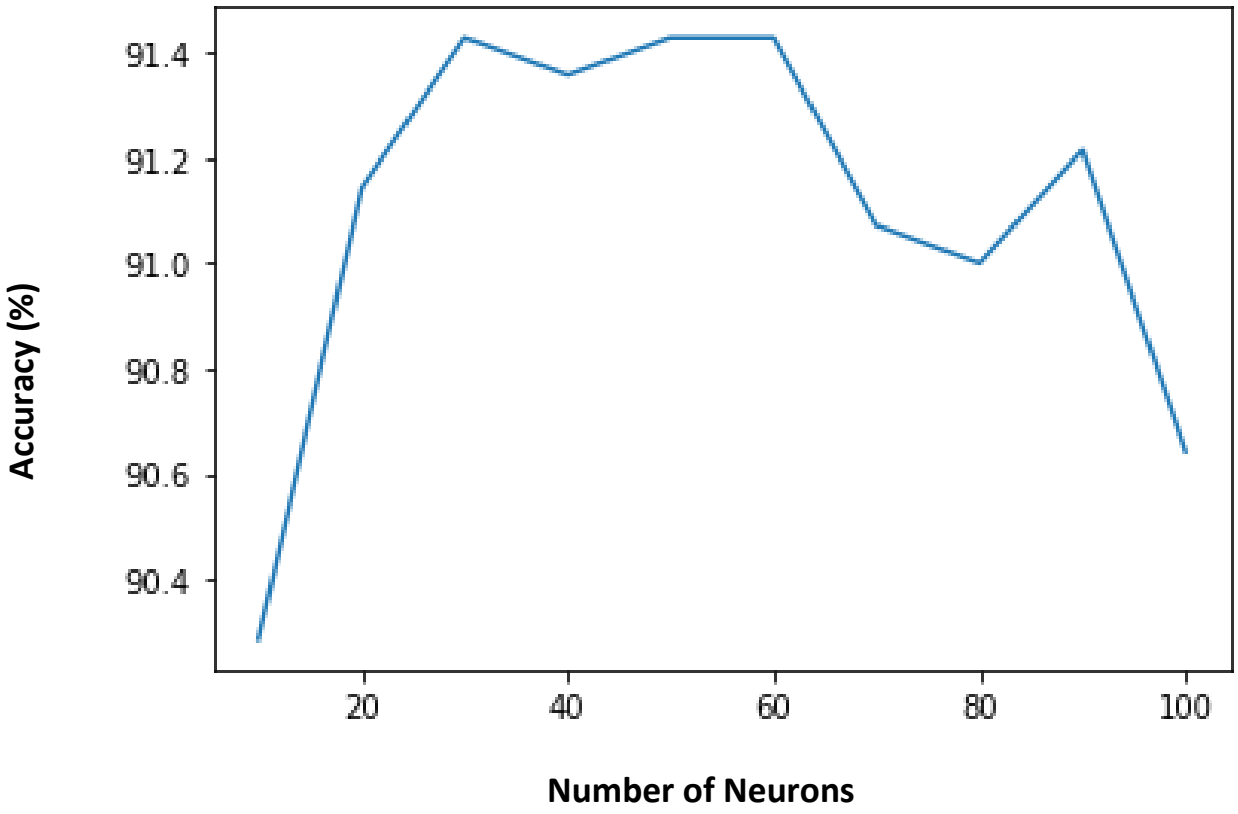
Plotted Accuracy (%) v/s Number of Epochs



Accuracy increases with the increase in number of epochs till the number reaches 20. After 20 epochs the, the test accuracy kind of saturates. It would be wrong to say that this lies in the overfitting region, since test accuracy never stops. Hence we can use convergence criterion anywhere between 20 – 100 epochs

Number of Neurons

Plotted Accuracy (%) v/s Number of Neurons



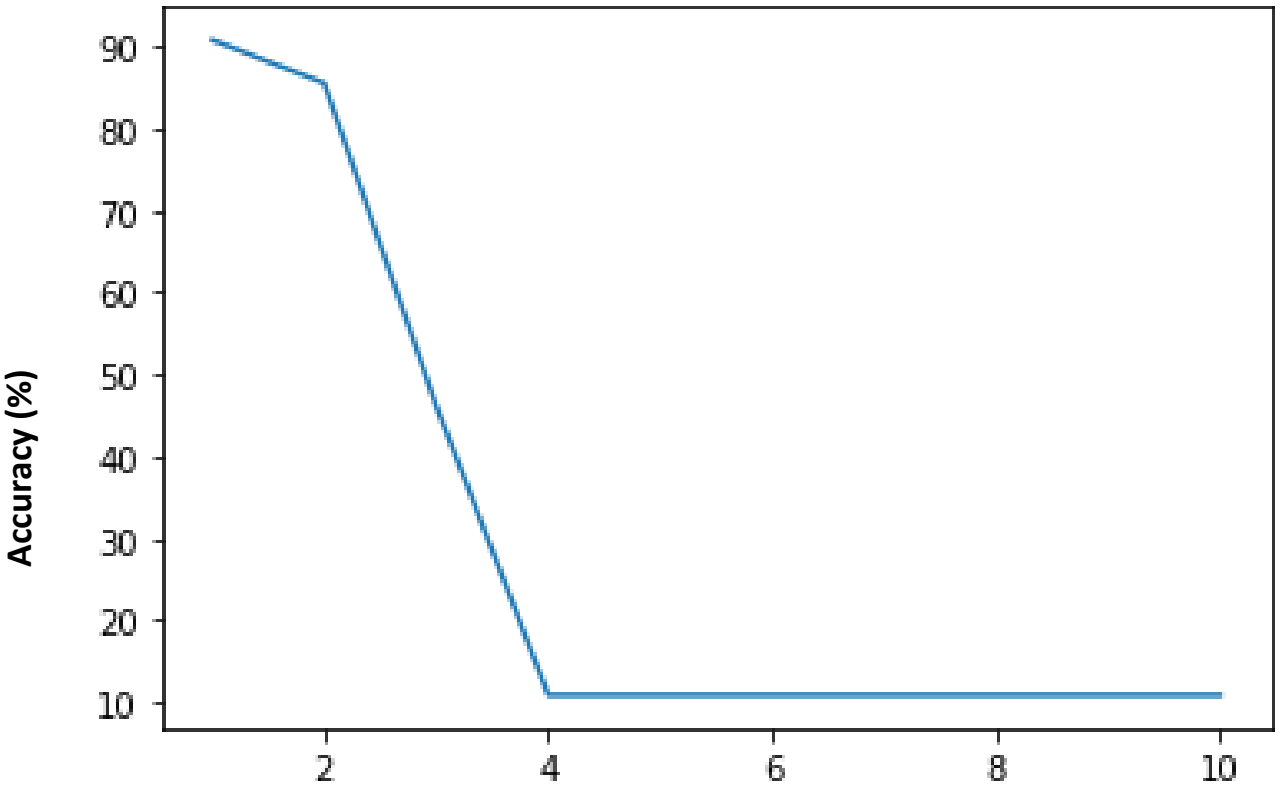
Optimal value of number of neurons is 40-60

I have used 70 neurons in the hidden layer

Min 90% accuracy is obtained when the number of neurons chosen is between 20 and 100

Batch Size

Plotted Accuracy (%) v/s Batch Size



The graph obtained is for some particular values of no. of epochs, neurons and learning rate. Only the batch size is variable. Because of this, the plot obtained does not seem right.

One can obtain a better accuracy with a large Batch Size by tweaking the initial weights and biases and the learning rate.

For very less number of epochs, **Stochastic Gradient Descent** is found to perform better (as expected). The accuracy increases for larger batch size as we increase the number of epochs but cannot match that to stochastic gradient descent

The best accuracy with Stochastic Gradient Descent is reported

Variation with number of hidden layers

Implemented two hidden layers instead of one but the accuracy did not improve. This implies that, for the given data, even one hidden layer is also sufficient to classify data.

Activation function for layer 1 weights -> Sigmoid

Activation function for layer 2 weights -> Sigmoid

Activation function for layer 3 weights -> Softmax

No. of neurons in 1^{st} hidden layer -> 50

No. of neurons in 2^{nd} hidden layer -> 25

Accuracy obtained: 87.35%

Conclusion

With the initial weights and biases -> (-0.01,0.01) and a learning rate of 0.01, using only one hidden layer it is possible to obtain a Neural Network Classifier with good accuracy (close to 90%) for the given problem. Unnecessary complicating the Neural Network might not give a better accuracy.