Assignment 3

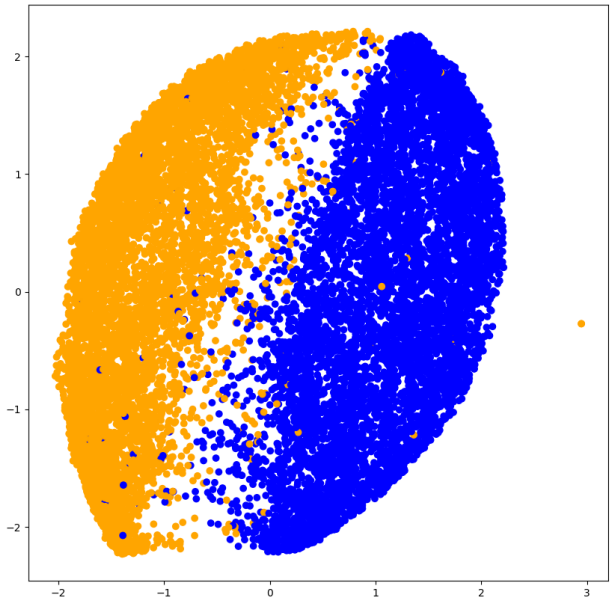
Machine Learning

Submitted By: **Himanshu Aggarwal**
(MT17015)

References:
- For implementing backpropogation: http://neuralnetworksanddeeplearning.com
- For reading MNIST dataset: https://gist.github.com/akesling/5358964
- Implementation discussed with Vani Agarwal

Given datasets visualisation:

MNIST Subset Visualization



MNIST Subset (h5 file)

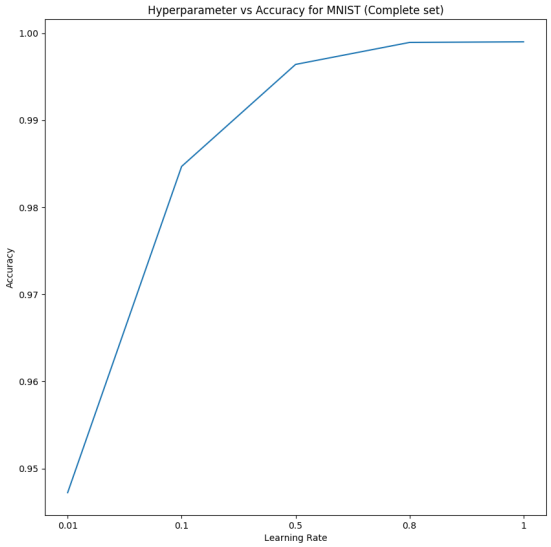
The given subset has two classes which are linearly separable as seen in the above plot. Classes here are the number on the images provided, i.e., number 7 and number 9.

MNIST Complete Set

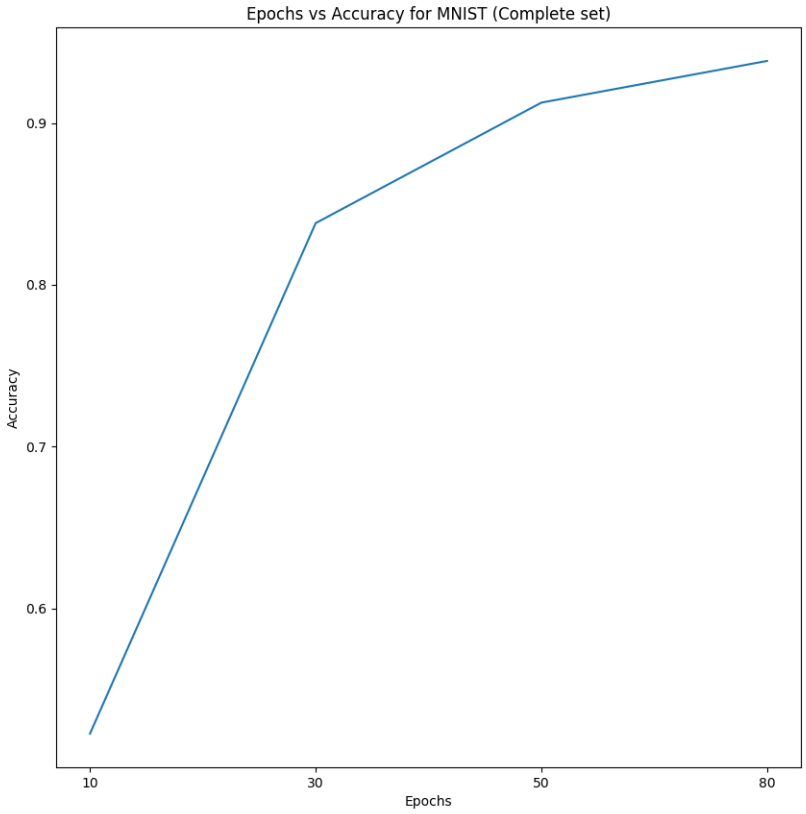
The given dataset has 10 classes representing 10 different numbers as images.

1A. Implementation of Artificial Neural Network with custom Backpropogation on MNIST Subset

Neural network has been implemented with sigmoid function at each layer(including output layer).



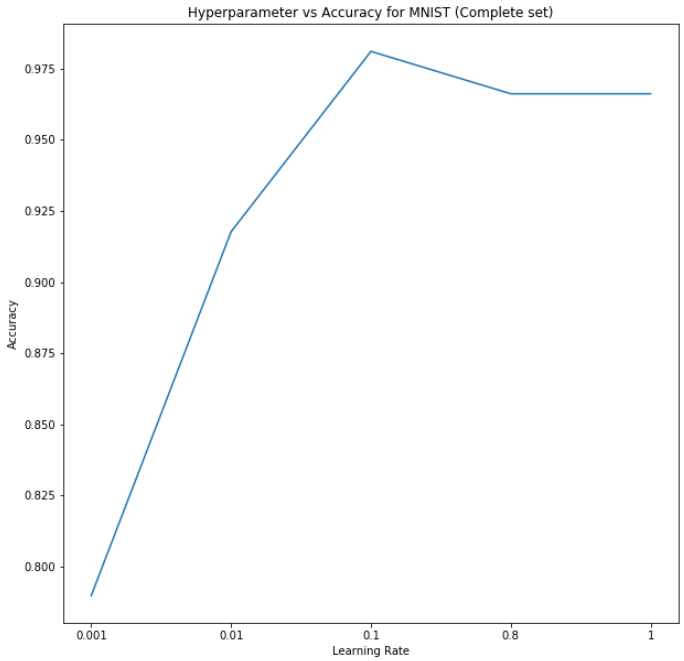
K-fold with K = 3 has been implemented with grid search on hyper parameter as learning rate.



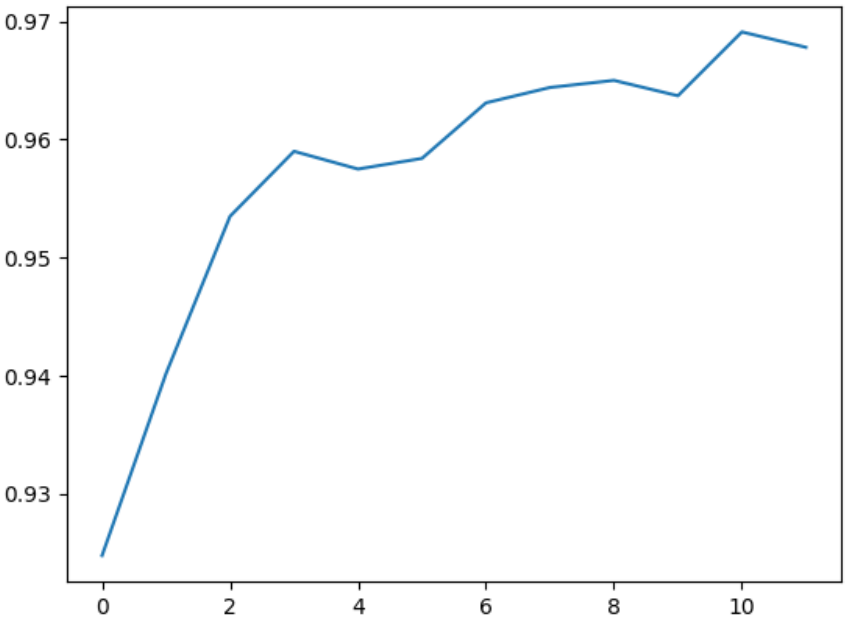
Above is the Epochs vs Accuracy graph for this model. Accuracy is increasing with epochs, depicting learning of the model. Accuracy for this model is 0.9396

1B. Implementation of Artificial Neural Network with custom Backpropogation on MNIST

Neural network has been implemented with activation function as sigmoid function at each hidden layer and softmax function at the output layer.



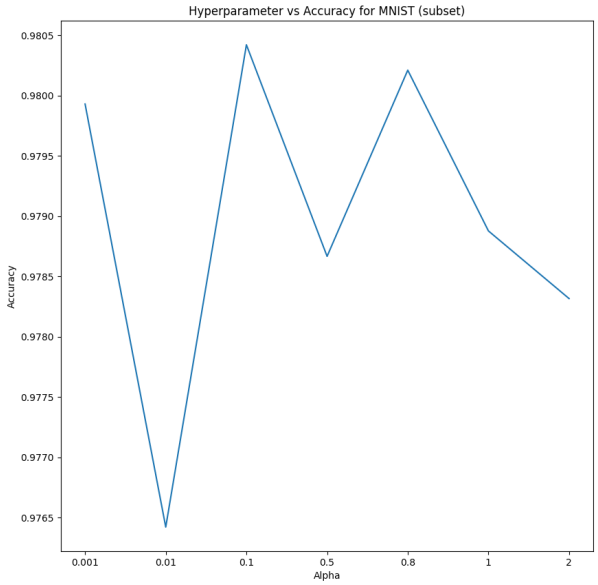
K-fold with K = 3 has been implemented with grid search on hyper parameter as learning rate.

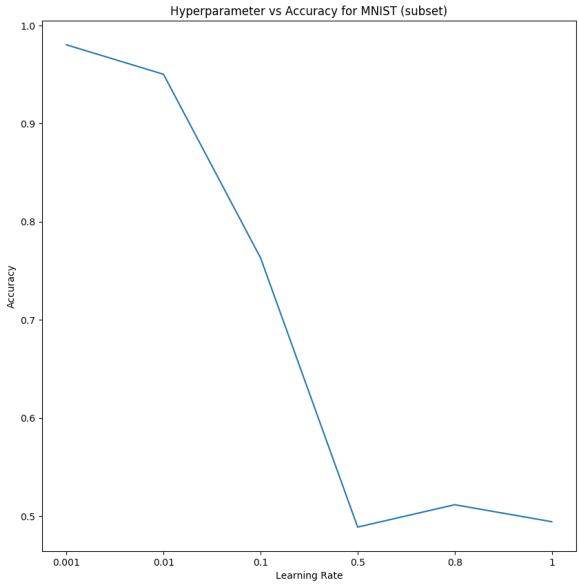


This is the Epochs vs Accuracy graph for this model. With each increasing epoch, the accuracy of the test data is increasing. However, the accuracy converges after a certain epochs, therefore, the graph with varying number of epochs has not been plotted. Accuracy for this model is 0.9663

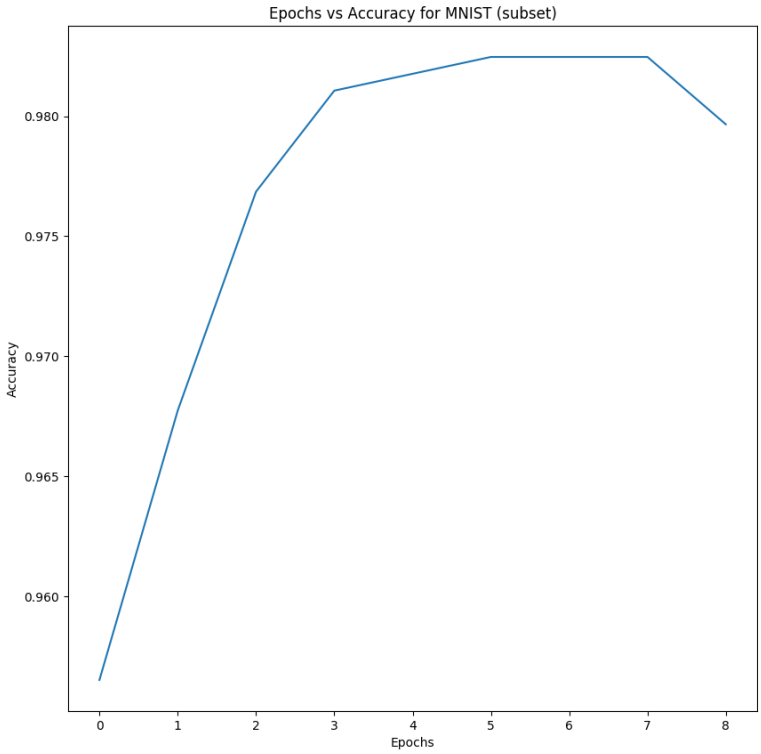
2A. Implementation of Artificial Neural Network with MLPClassifier on MNIST Subset

Neural network has been implemented with sigmoid function at each hidden layer(including output layer)





K-fold with K = 3 has been implemented with grid search on hyper parameter as learning rate and alpha. As can be seen from the graphs above, with increasing alpha, accuracy varies too much. However, with increasing learning rate, accuracy of this model decreases.

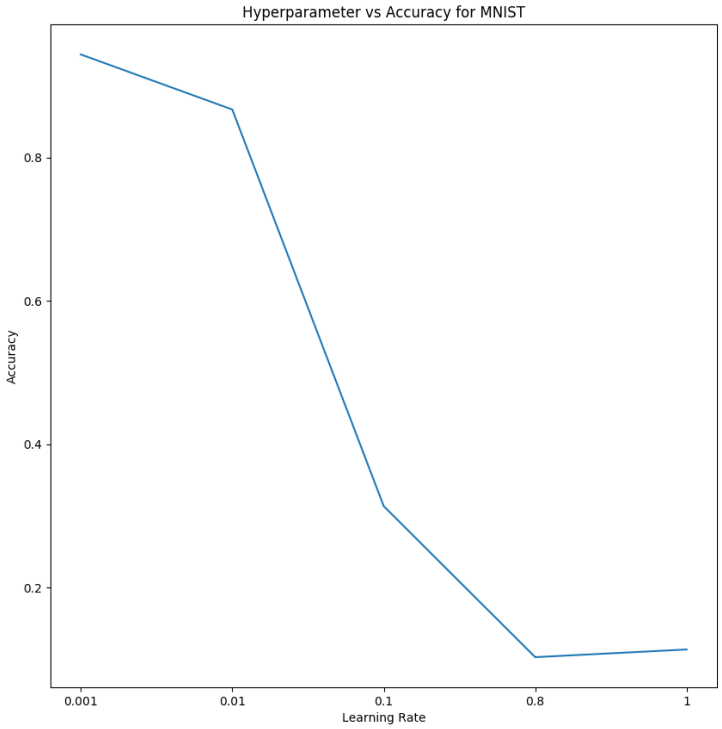


This is the Epochs vs Accuracy graph for this model. Accuracy is increasing with epochs, depicting learning of the model. Accuracy for this model is 0.9788

This model performs better than the above(self implemented) model with the same dataset, possibly because the outer layer of this model uses softmax function(by default).

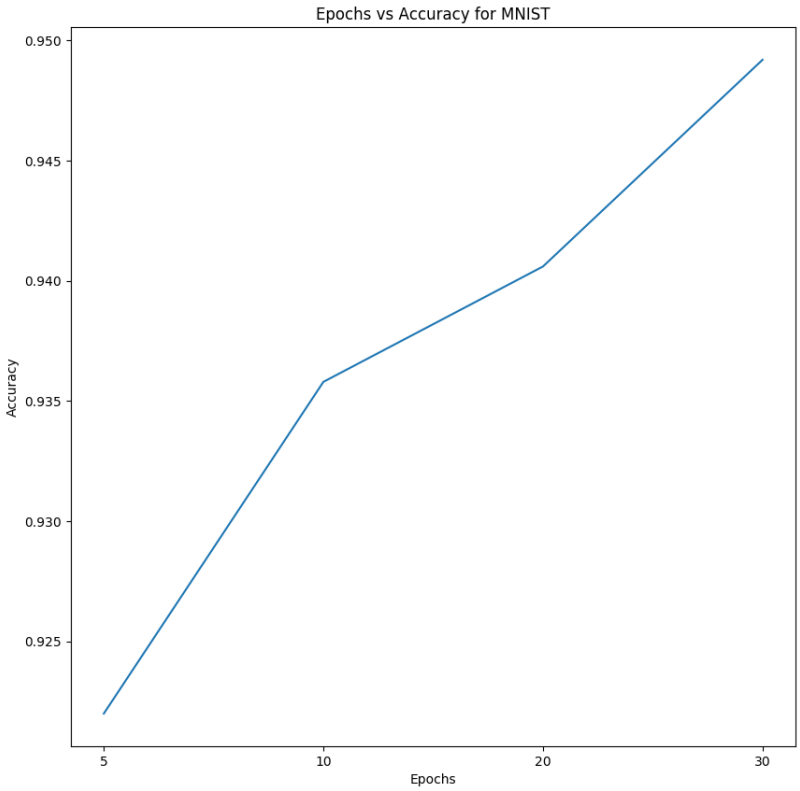
2B. Implementation of Artificial Neural Network with MLPClassifier on MNIST

Neural network has been implemented with sigmoid function at each hidden layer and softmax at outer layer.



K-fold with K = 3 has been implemented with grid search on hyper parameter as learning rate.

With increase in the learning rate, accuracy of the model decreases.

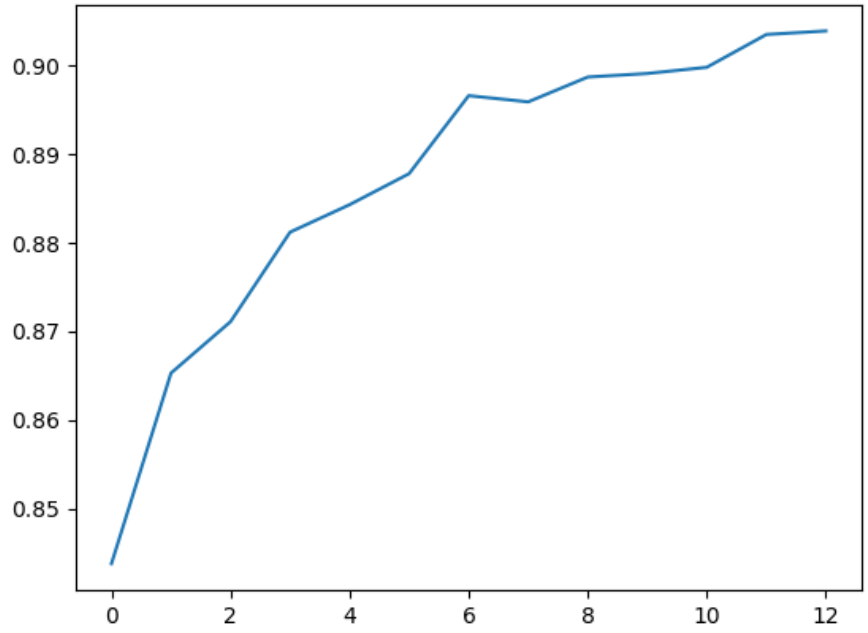


This is the Epochs vs Accuracy graph for this model. Accuracy for this model is 0.9446

Accuracy for this model is almost similar to the self built model above.

1C. Implementation of Artificial Neural Network with custom Backpropogation on MNIST Subset

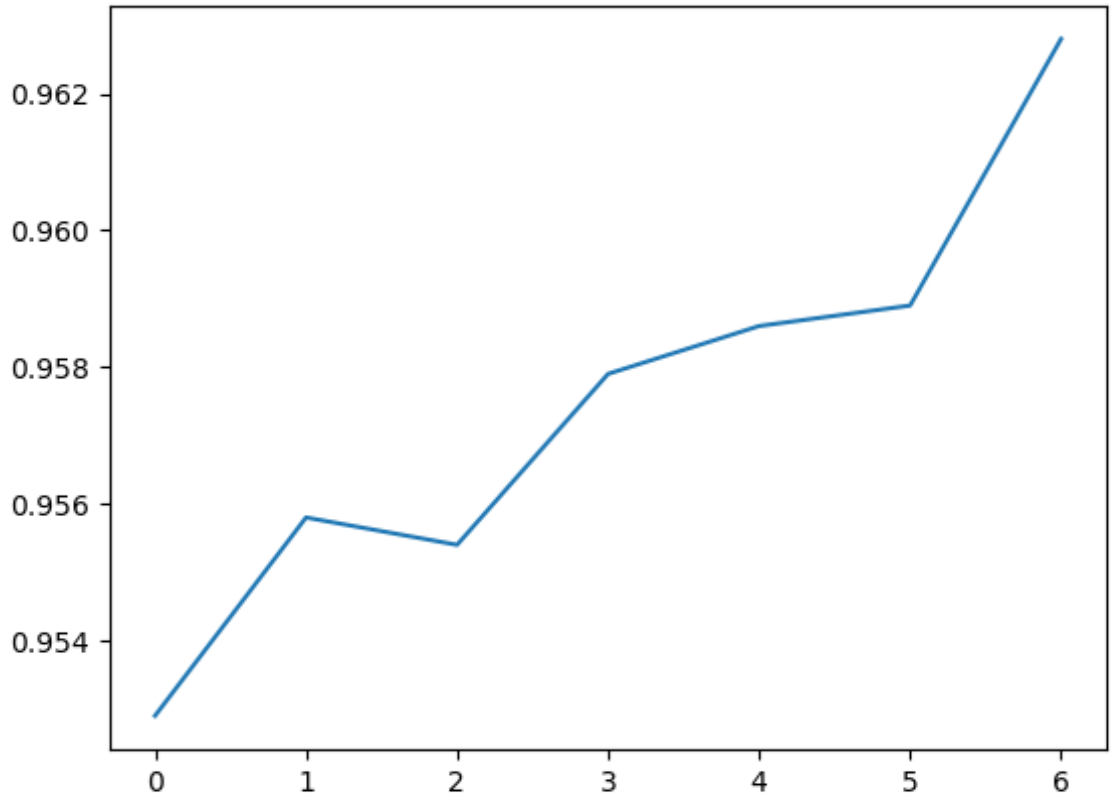
Neural network has been implemented with activation function as ReLU function at each hidden layer and sigmoid function at the output layer.



Above is the Epochs vs Accuracy graph for this model. Accuracy for this model is 0.9628
This is better than the self built model and almost similar to the model in 2(a). This shows that ReLU function performs better than the sigmoid functions in the hidden layer.

1C. Implementation of Artificial Neural Network with custom Backpropogation on MNIST

Neural network has been implemented with activation function as ReLU function at each hidden layer and softmax function at the output layer.



Above is the Epochs vs Accuracy graph for this model. Accuracy for this model is 0.9027
This is worse than the self built model and model in 2(a). This shows that ReLU function does not perform better than the sigmoid functions in the hidden layer with softmax in the output.

3. Implementation of Artificial Neural Network with MLPClassifier on MNIST

The below variations in the architecture have been tried with gives the resultant accuracy better than the previous models.

1. Activation function: ReLU, Hidden Layers Nodes: (900, 400, 100)
2. Activation function: ReLU, Hidden Layers Nodes: (500, 300, 50)
3. Activation function: Logistic, Hidden Layers Nodes: (500, 300, 100, 30)

These structures of the neural nets have been taken at with hit and trial method, keeping in mind that the activation function, number of hidden layers and number of hidden nodes, affect the model's accuracy.
Shown below are the accuracy graphs of the models described above. All of them performs better than most of the previously implemented models in this assignment. However, a comparison can be seen with the experiment (1) where the accuracy decreases at some epochs. This shows that that model is not perfect and may produce less accuracy at odd times.