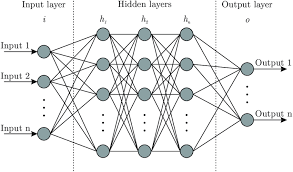
**Artificial Neural Networks**

**Iype Eldho(2018AAPS0339H)**

**CH V SS Chaitanya(2018AAPS0393H)**

**Abhipal Sharma(2019A7PS0161H)**

Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems vaguely inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs.



In the model developed in this assignment, the gradient descent algorithm has been used to optimize the weights for the training data. These calculated weights have then been used to test the accuracy of the model for the testing data.

1. One hidden layer:

Number of nodes in input layer = 6

Number of nodes in hidden layer = 15

Number of nodes in output layer = 10

Activation Function for hidden layer = Relu

Activation Function for output layer = Softmax

Learning rate = 0.10

Number of training iterations = 5000

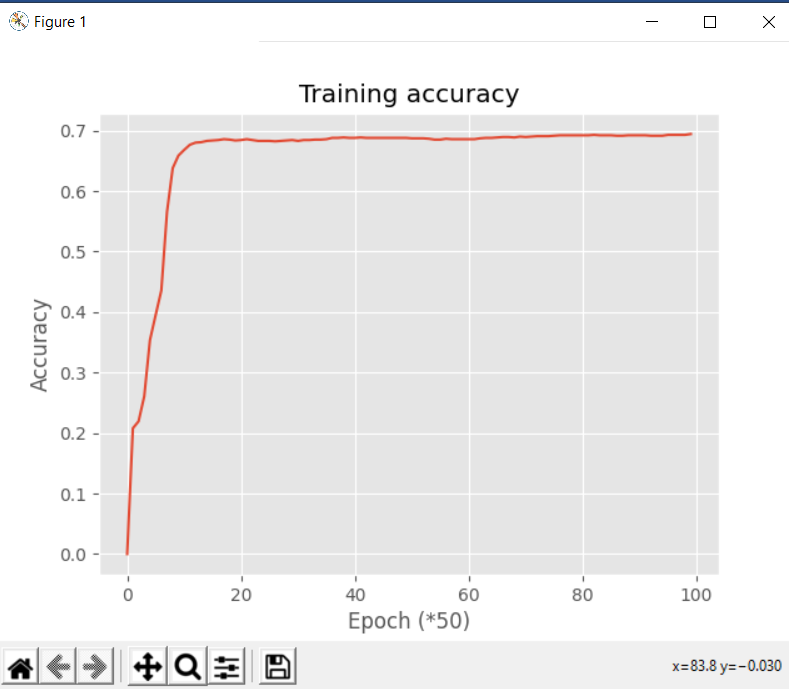
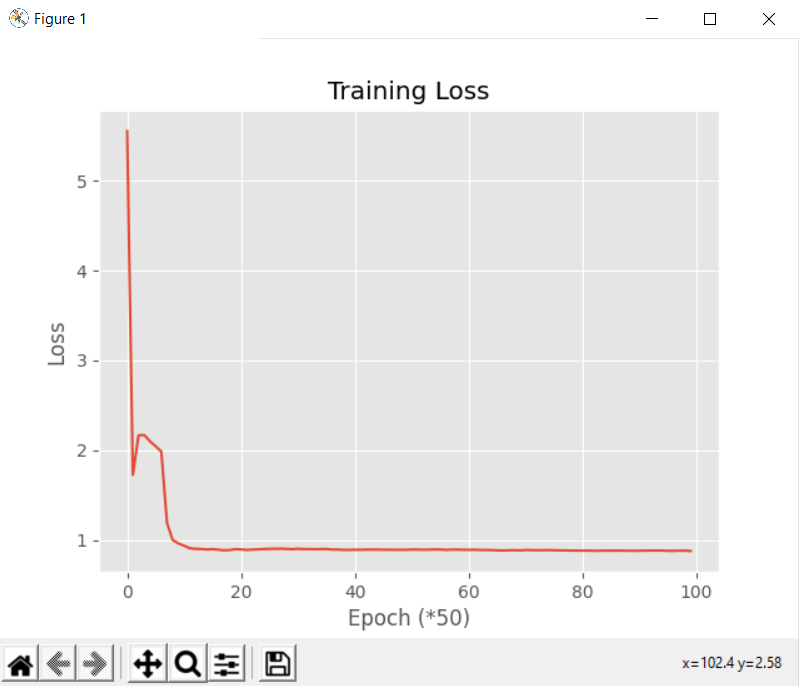
Accuracy achieved in training data = 69.43%

Loss encountered while training = 0.878

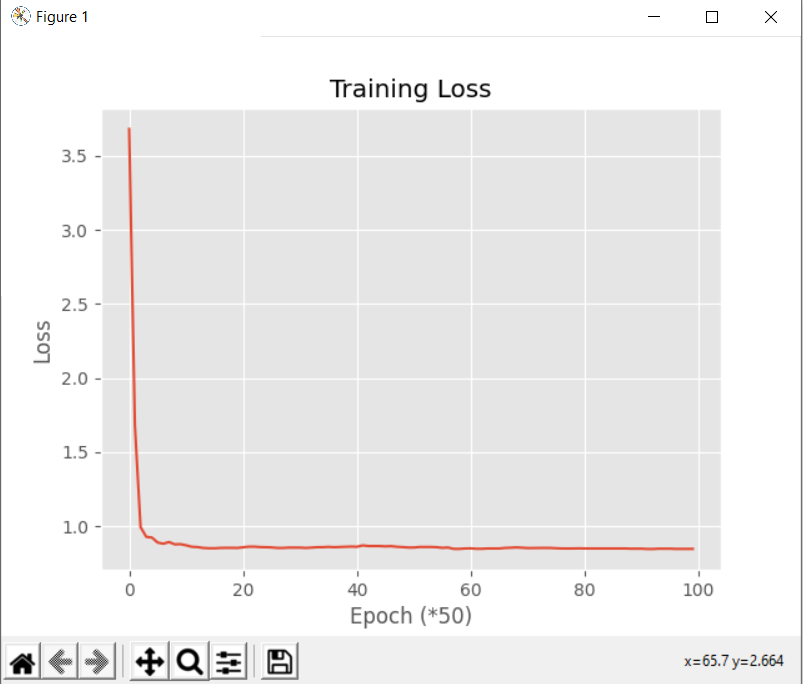
Accuracy achieved in testing data = 70.45%

Loss encountered in testing data = 0.846

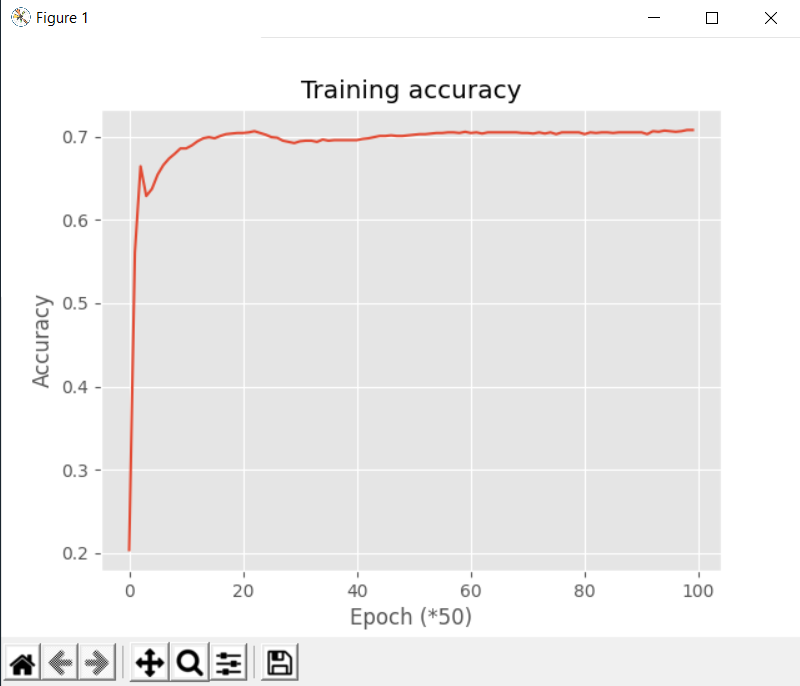
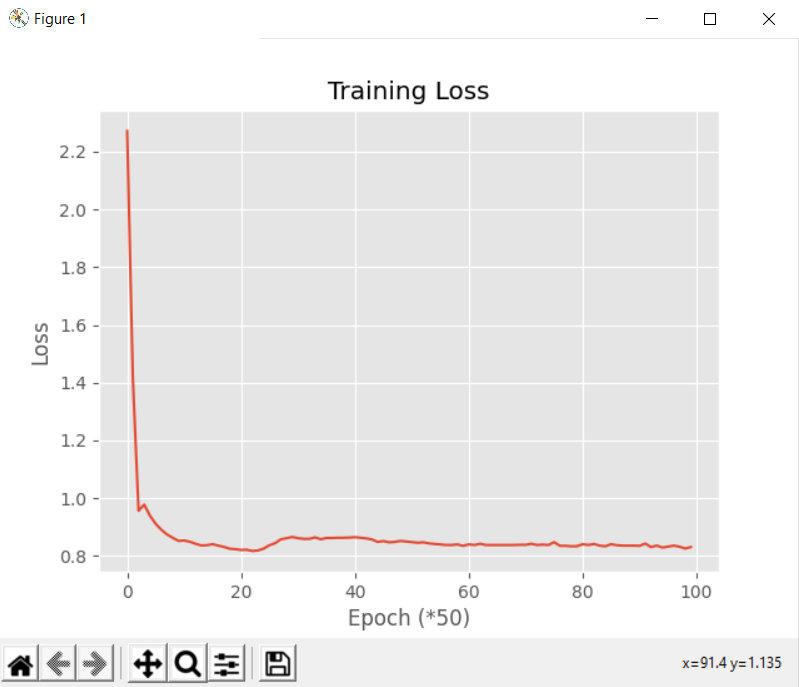
Plots for learning rate = 0.10

Plots for learning rate = 0.25



Plots for learning rate = 0.5

1. Two hidden layers:

Number of nodes in input layer = 6

Number of nodes in 1st hidden layer = 20

Number of nodes in 2nd hidden layer = 15

Number of nodes in output layer = 10

Activation Function for both hidden layers = Relu

Activation Function for output layer = Softmax

Learning rate = 10e-2

Number of training iterations = 20000

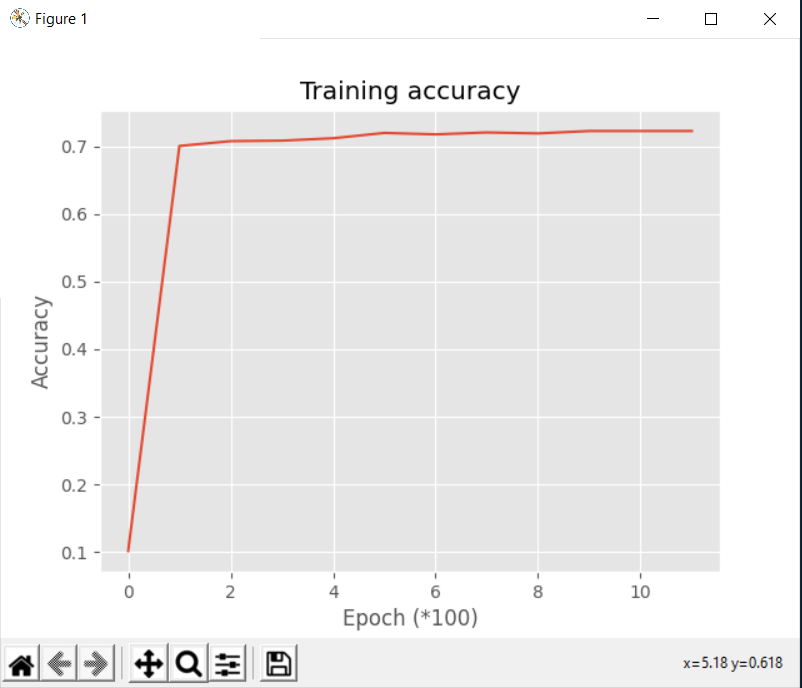
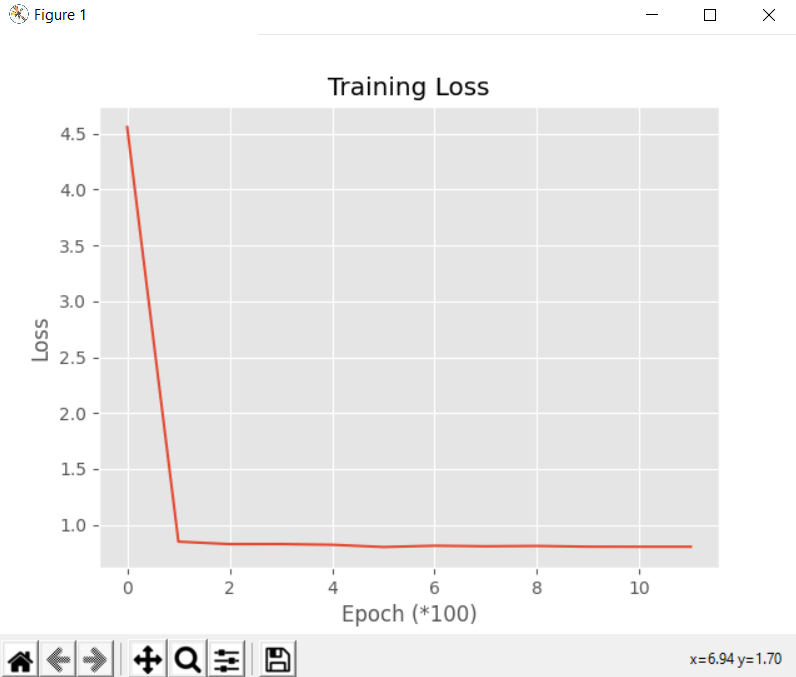
Accuracy achieved in training data = 71.50%

Loss encountered while training = 0.823

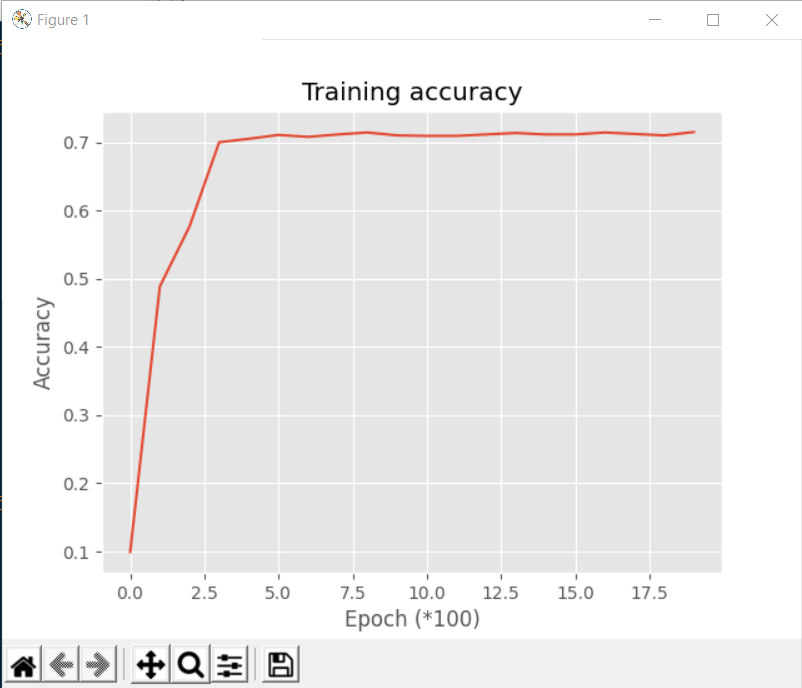
Accuracy achieved in testing data = 69.28%

Loss encountered in testing data = 0.875

Plots for learning rate = (10e-3)/2

Plots for learning rate = 10e-2

Plots for learning rate = 10e-3

