Hamzah Baig
20100275

Artificial Assignment 2 – Neural Network

Description of Assignment:

I completed the assignment using following steps:

1. I divided the neural network in two steps:
    - train()
    - test()

2. The images are stored in a .txt file in form of pixels and its labels are also stored in .txt file. Both files are passed through command line as arguments.

3. Description of two major functions test and train are given below:

train():

- The train functions takes file name and learning rate as arguments.
- I read the ***train.txt*** file and fill a list of list of ***image pixels***. Each list in this Big list contains 784 elements i.e. image pixels.
- Now I will initialize randomly HiddenWeights [***dimension (784,30)***] and OutputWeights [***dimension (30,10)***] with numbers between -1 and 1.
- Now I take one image multiply with ***hiddenWeights*** and then I activate it using activation function i.e ***sigmoid***,
- Then after activation I multiply it with ***outputWeights*** and then I activate it using activation function.
- Hence the result will be a matrix of ***dimension (1,10).***

- Now, I calculate the error using the labels given and use ***back propagation*** and ***gradient descent*** to find the best ***hiddenWeights*** and ***OutputWeights*** and I continue this training for rest of images i.e. ***60000.***
- When traingis complete I write all the weights in a text file called ***netWeights.txt.***

test():

- The test function takes the file name and ***netweights.txt*** as arguments.
- Now I read weights from ***netWeights.txt*** from ***HiddenWeights*** and ***OutputWeights***.
- After Reading, I take one image pixel and run it on the neural network with ***weights*** I just calculated using the ***train() function.***
- I match the output with the l***abels given*** and calculate the ***accuracy***.
- If image is classified ***correctly*** I do

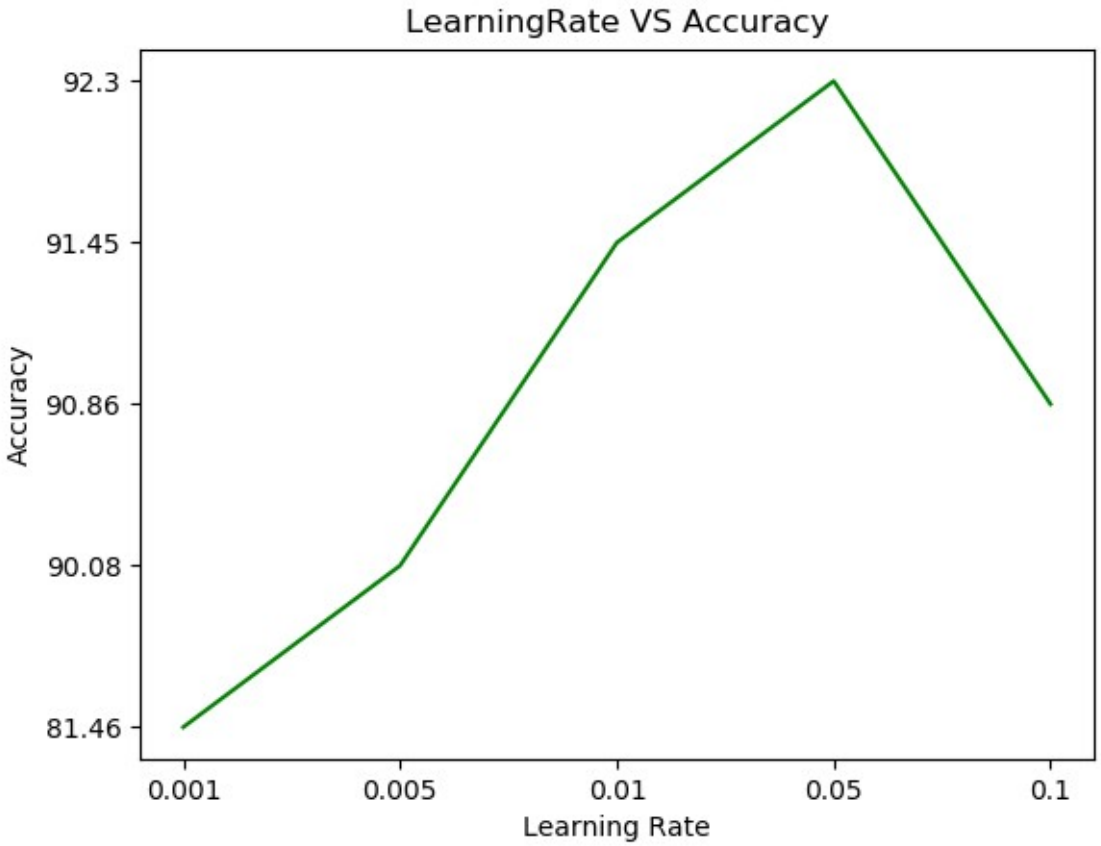
  accuracy = accuracy +1
- I continue this step for all the images ***i.e. 10,000 images.***
- And after testing ***10,000 images***, I use following formulas to find Accuracy and Error:

  Error = (10000-accuracy/100) %

  Accuracy = accuracy/100 %

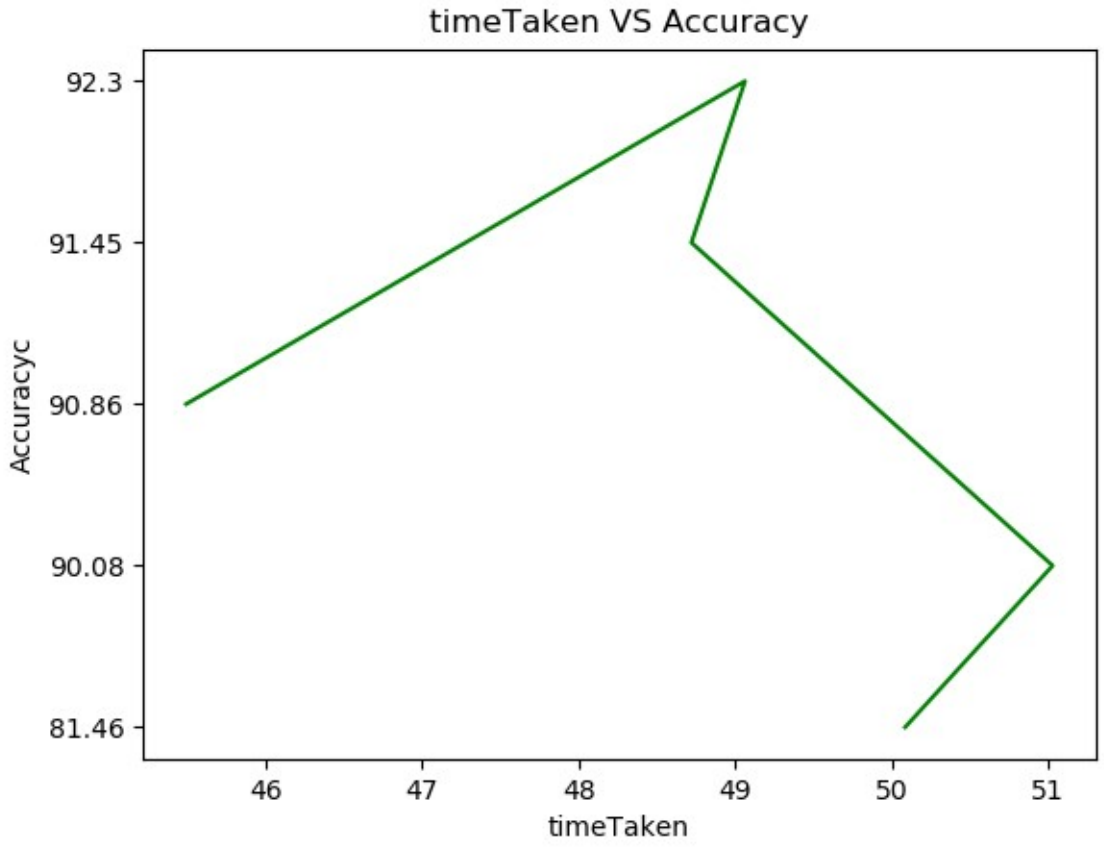
Graphs – Neural Network

1)



**graph showing relation between learningRate and accuracy*

2)



graph showing relationship between **timeTaken by the nueral network and **Accuracy***