CS671 - Deep Learning and its Applications
Programming Assignment 1

Sabin Kafley (D18032)
Dhanunjaya Varma (S18023)
Hritik Gupta (B16097)

March 15, 2019

Contents

1. Problem Statement **2**

2. Dataset **2**

2.1 MNIST Dataset 2

2.1.1 Learning Curves 2

2.2 Variations Tried 2

2.3 Inferences 2

1. Problem Statement

To make a simple fully connected network to classify the MNIST dataset and the dataset you made of lines. The caveat is that the layer APIs provided by Tensorflow cannot be used. Hence a dense layer API needs to be coded with exhaustive features

2. Dataset

1. MNIST

   No. of Classes : 10

   Train Images : 60000

   Test Images : 10000

2. Lines

   No. of Classes : 96

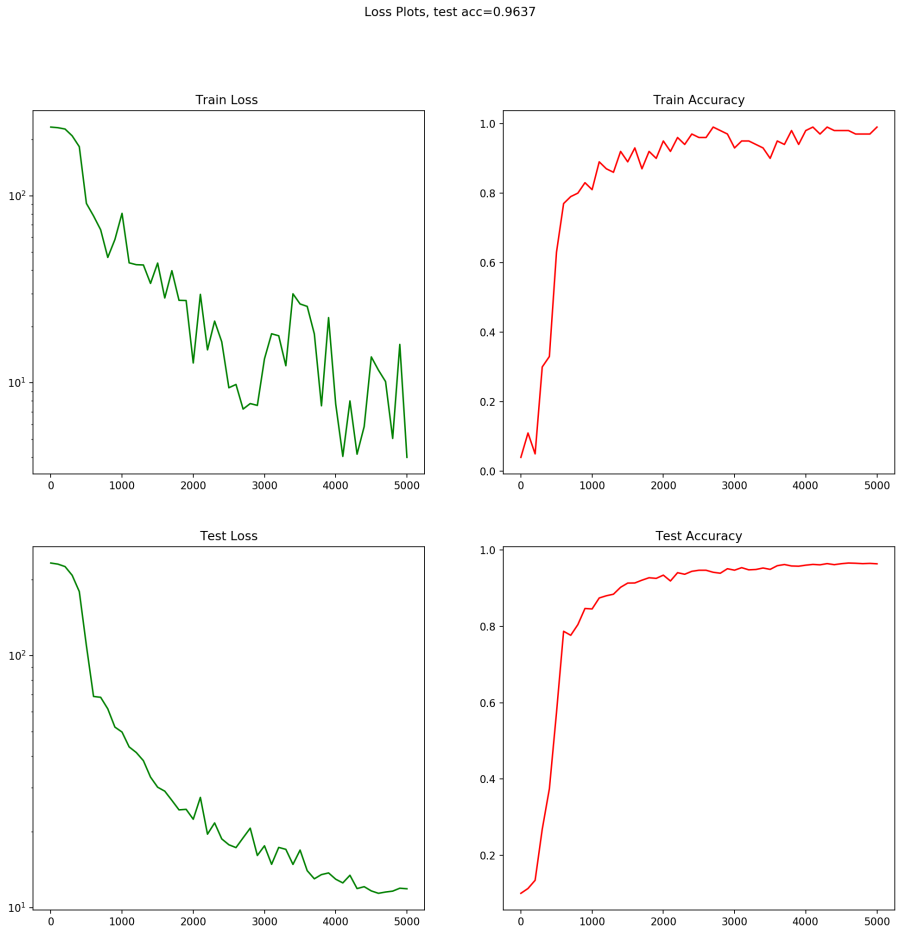
   No. of items in each Class : 1000

   Train : 850 for each Class

   Test : 150 for each Class

2.1 MNIST Dataset

2.1.1 Learning Curves



2.2 Variations Tried

1. Increasing Number of Layers
2. Increasing Number of neurons in each layer
3. Changing the Activation Function (Sigmoid/ReLu)
4. Adding Dropout

2.3 Inferences

1. Increasing the layers increased the accuracy, but to an extent. After we kept on increasing the layers, the model started to overfit and the accuracy on test data was highly reduced

2. ReLu converged faster than Sigmoid, as we started getting better accuracy in lesser number of epochs. But ReLu also suffers from Dying ReLu problem, where neurons with outputs lesser than 0 are simply ignored

3. Increasing the number of neurons also helped in increasing the accuracy. Following a similar pattern as that of the number of layers, accuracy is affected if the number of neurons is very high. This is due to the fact that the dimension of the weight matrix becomes high, hence the model starts to overfit

4. Adding dropout to the layer, ignores output of randomly chosen neuron during TRAINING.It is a kind of regularization that prevents overfitting of the model.

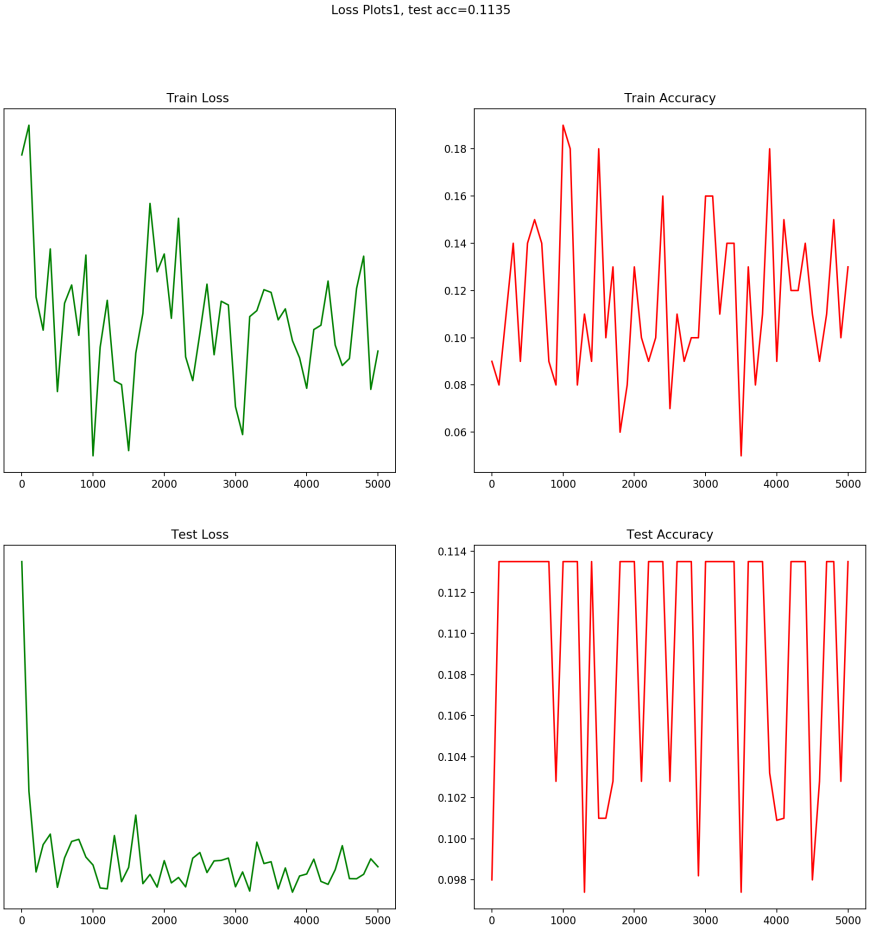


Figure 1: Reduced accuracy by adding 2 more layers

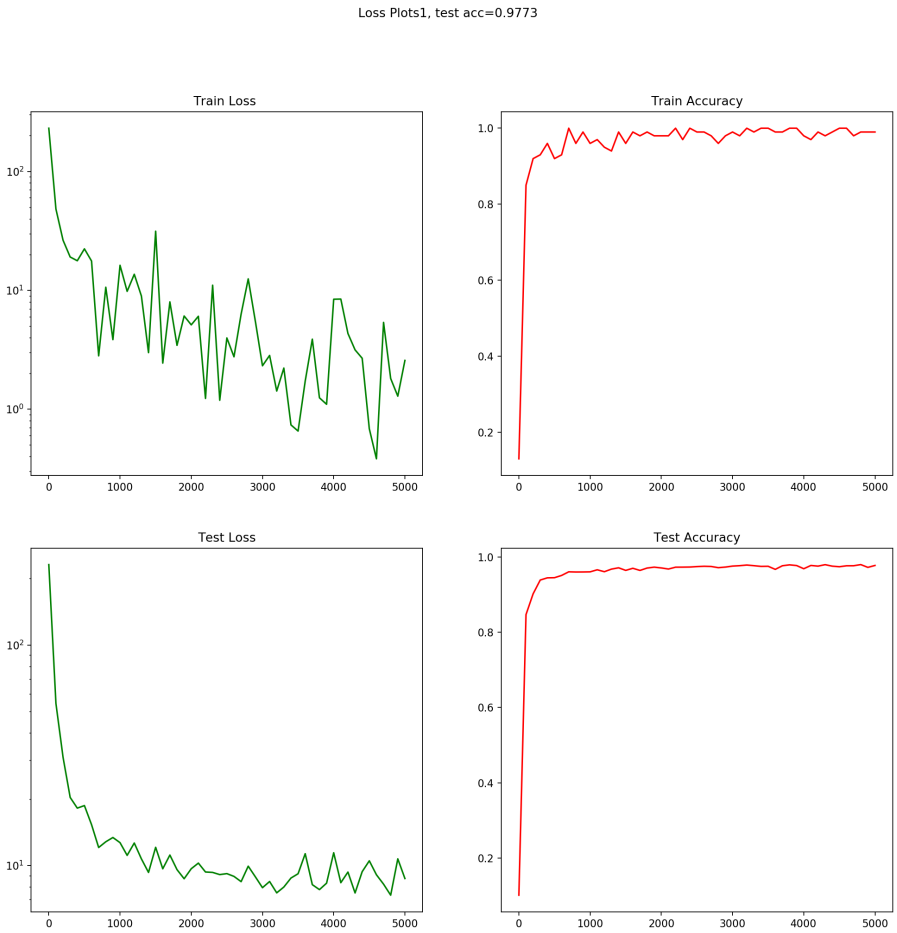


Figure 2: Using ReLu as an Activation Function