

Deep Learning Report 2

Rishabh Ravindra Meshram (16149)

M.Tech CSA

1 Objective

This project is to implement neural network and convolutional neural network for the task of classification. The classification task will be that of recognizing an image and identify it as one of ten classes. You are required to train the classifiers using Fashion-MNIST clothing images. Following are the two tasks to be performed:

1. Build multi-layer Neural Network with open-source neural-network library (pytorch/tensorflow) on Fashion-MNIST dataset.
2. Build Convolutional Neural Network with open-source neural-network library (pytorch/tensorflow) on Fashion-MNIST dataset.

2 Multi-layer Neural Network

2.1 Architecture

Input layer Size : $28 * 28$ (784)

Number of Hidden Layer : 3 has 180 neurons each

Output Layer Size :10

Activation Functions : Relu in hidden layers

2.2 Optimizer and Loss function

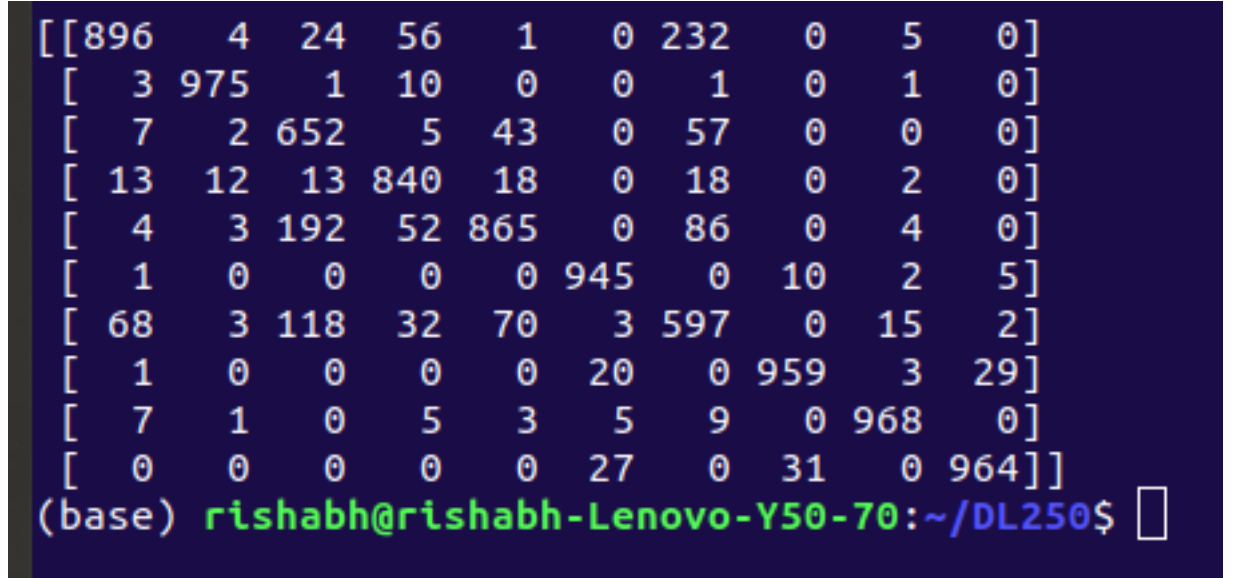
Optimizer : Adam with learning rate = 0.001

Loss function : Cross Entropy

2.3 Result

Accuracy:86.61 % Loss: 1.543490

Figure 1: Confusion Matrix



3 Convolutional Neural Network

3.1 Architecture

Model Architecture :

No. of convolution layer : 2

Input channel for layer 1 : 1

Output channel for layer 1 : 16

Kernel size (filter size) : 5

Input channel for layer 2 : 16

Output channel for layer 2 : 32

Output Layer Size : 10

Activation Functions : Relu

Pooling : Max Pooling

Optimizer: Adam

Loss function: Cross Entropy

Batch size: 32

No. of epoch: 20

3.2 Result

Accuracy: 89 %

Loss : 0.2765

```

➡ [[793    2   11   14    1    0   72    0    1    0]
   [  0 987    1    7    1    0    2    0    1    0]
   [ 16    1 840    9   41    0   51    0    1    0]
   [ 13    5    3 897   21    0   15    0    3    0]
   [  4    1   74   23 878    0   73    0    4    0]
   [  1    0    0    0    0 983    0    6    2    8]
   [167    4   71   48   57    0 783    0    4    1]
   [  0    0    0    0    0   15    0 975    3   29]
   [  5    0    0    2    1    0    4    0 981    0]
   [  1    0    0    0    0    2    0   19    0 962]]

```

Figure 2: Confusion Matrix