

# ML-2 Assignment

Vinit Ramesh Gore (M20CS064)

2) Solution:

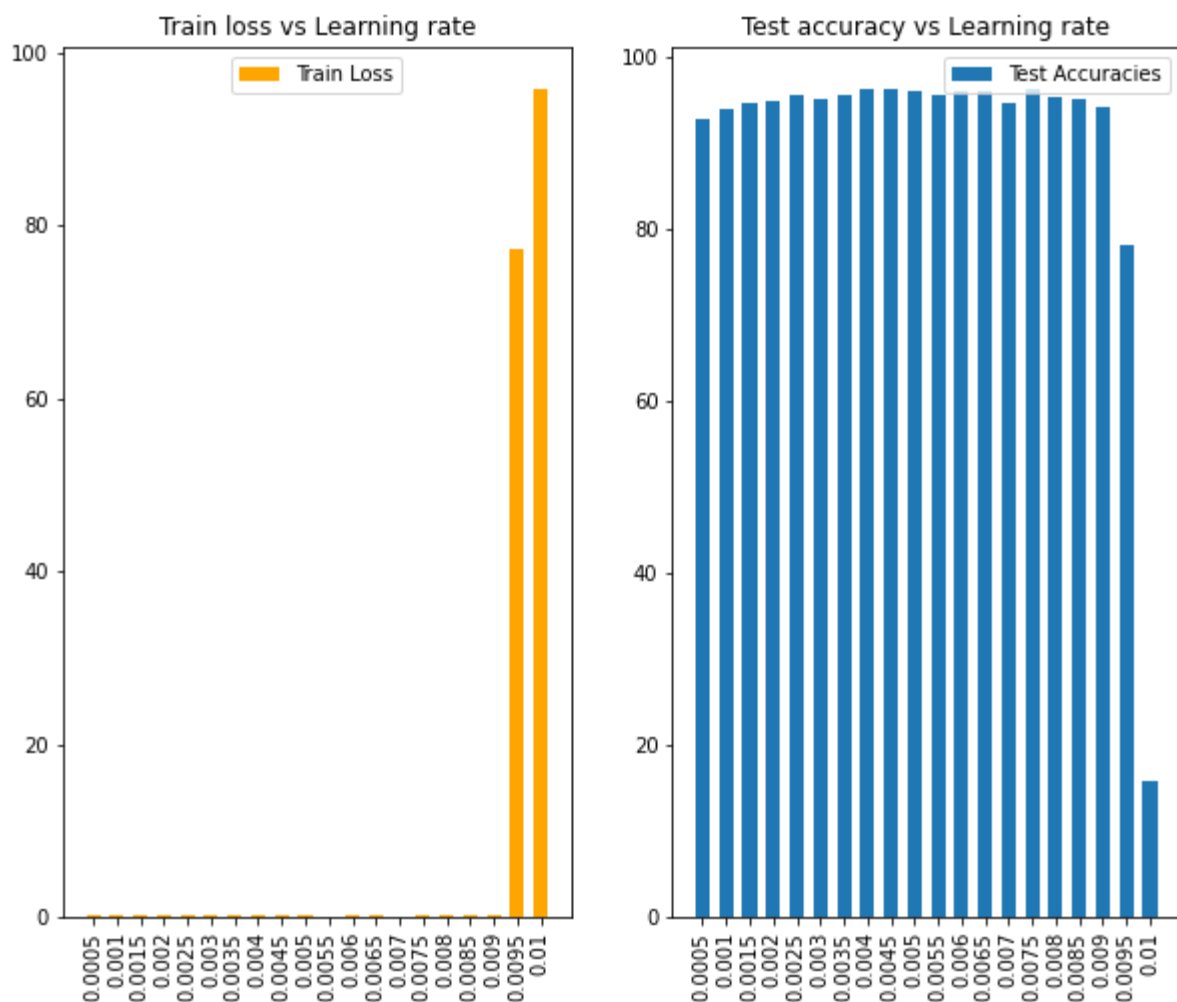
<https://colab.research.google.com/drive/11DC5vJrT2q6ZKi1pSJ6wl-EIZiLjTeoP?usp=sharing>

While doing this question, I learned the depths of Python and more or less, got familiar with the process of creating Python libraries.

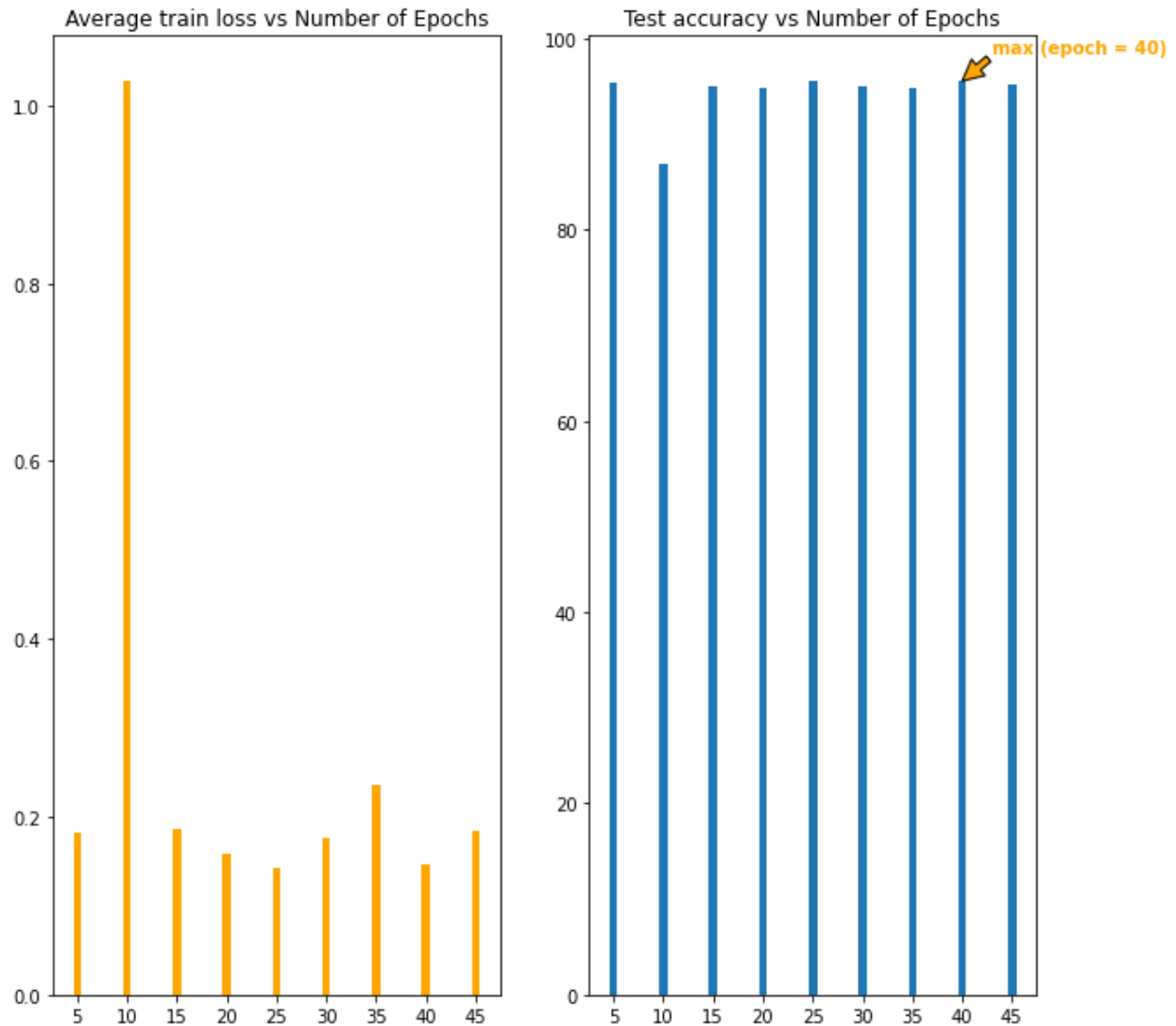
I used the learnings from the ML-1 course here. That is, used cross entropy loss and softmax along with affine transformation.

3) Solution:

<https://colab.research.google.com/drive/1kPBBSXv5WXYb7rJ2Z-ts1wIU9ImUo0fm?usp=sharing>



The above plot was created to identify optimal learning rate by keeping the number of epochs as 2. From the test accuracy vs learning rate graph, **optimal learning rate** was found to be **0.0075**.



The above plot was created to identify optimal learning rate by keeping the learning rate as 0.0075. From the test accuracy vs learning rate graph, there are very slight differences in the test accuracies with respect to number of epochs.

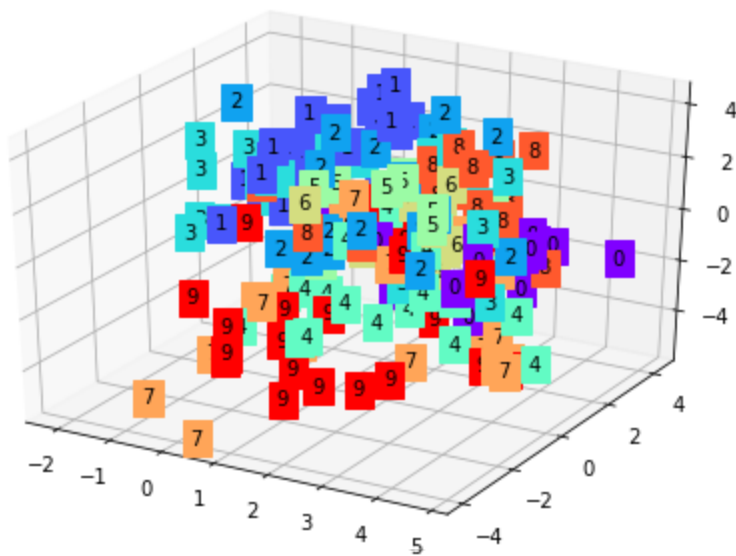
4) Solution:

[https://colab.research.google.com/drive/1FFp\\_8phH9PBL9urOubWyFRSfcCMzfGJd?usp=sharing](https://colab.research.google.com/drive/1FFp_8phH9PBL9urOubWyFRSfcCMzfGJd?usp=sharing)

#### Visualization of Feature Maps:

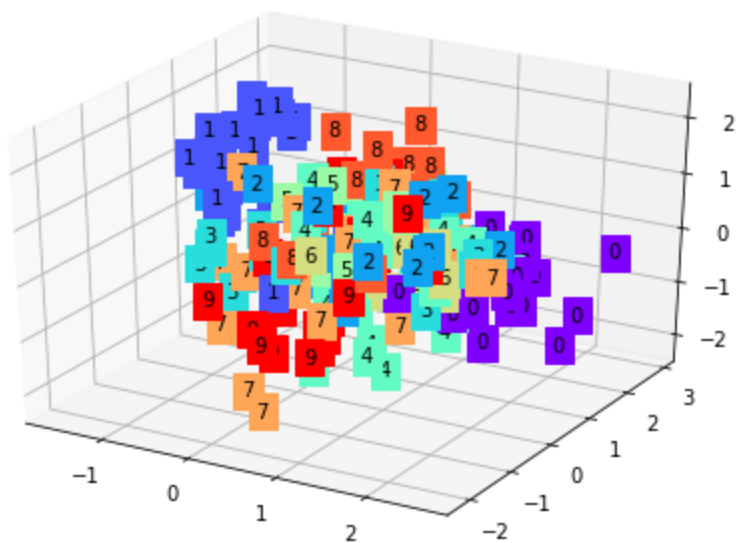
AutoEncoder without activation functions during encoding and decoding

Epochs : 2, Lr: 0.005



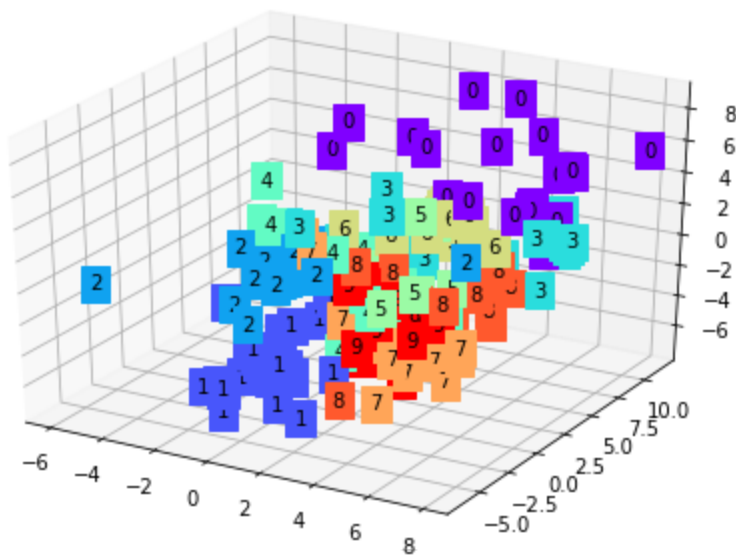
Observations: Quite well clustered

AutoEncoder without activation functions during encoding and decoding  
Epochs : 10, Lr: 0.005



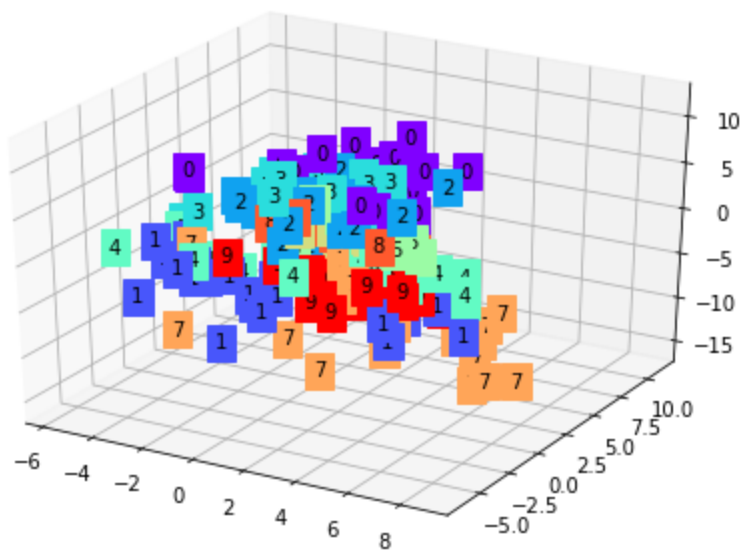
Observations: Well clustered for most of the classes

AutoEncoder with Leaky ReLU activation functions during encoding and decoding  
Epochs : 2, Lr: 0.005



Observations: Not well clustered

AutoEncoder with Leaky ReLU activation functions during encoding and decoding  
 Epochs : 10, Lr: 0.005



Observations: Quite well clustered.

### **Classification:**

Hyperparameters used:

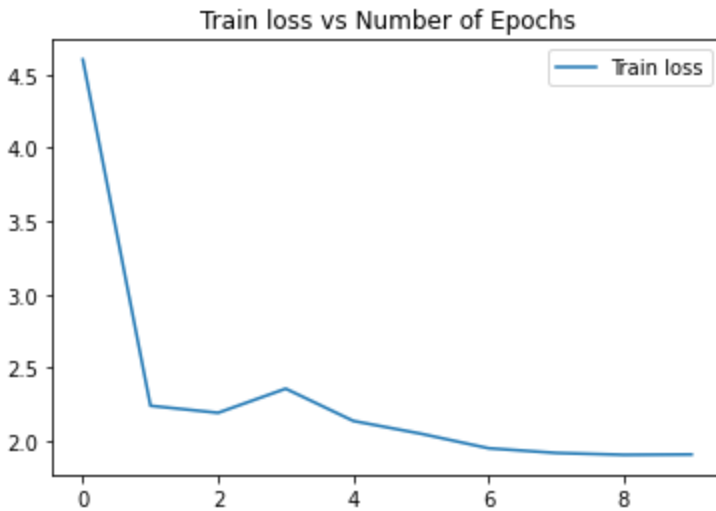
Number of epochs = 10

Batch size= 200

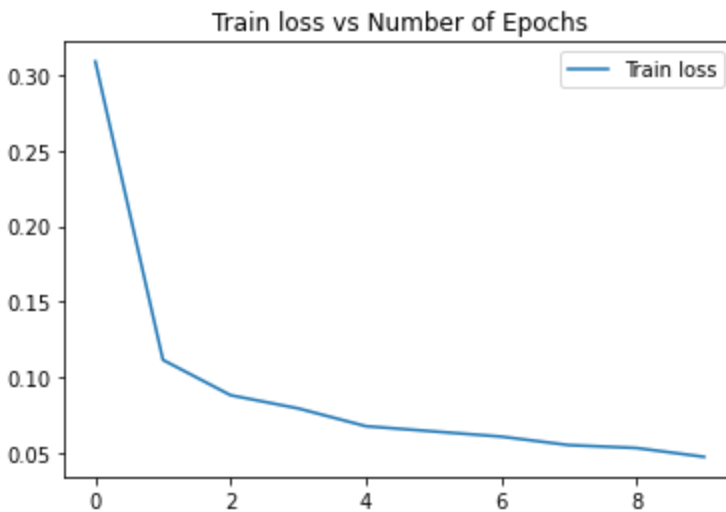
Learning Rate = 0.005

When activation function LeakyRelu was not used during encoding and decoding, the test accuracy came to be **31%**

Train loss curve is as follows:



Observations when activation functions were used:



**Test accuracy** of the model with learning rate as 0.005 on the 10000 test images is **97%**.