DL Assignment-1 [700 Level]

NN from Scratch, Denoising Autoencoder and CNN

By:- Kwanit Gupta
B19EE046

Question-1

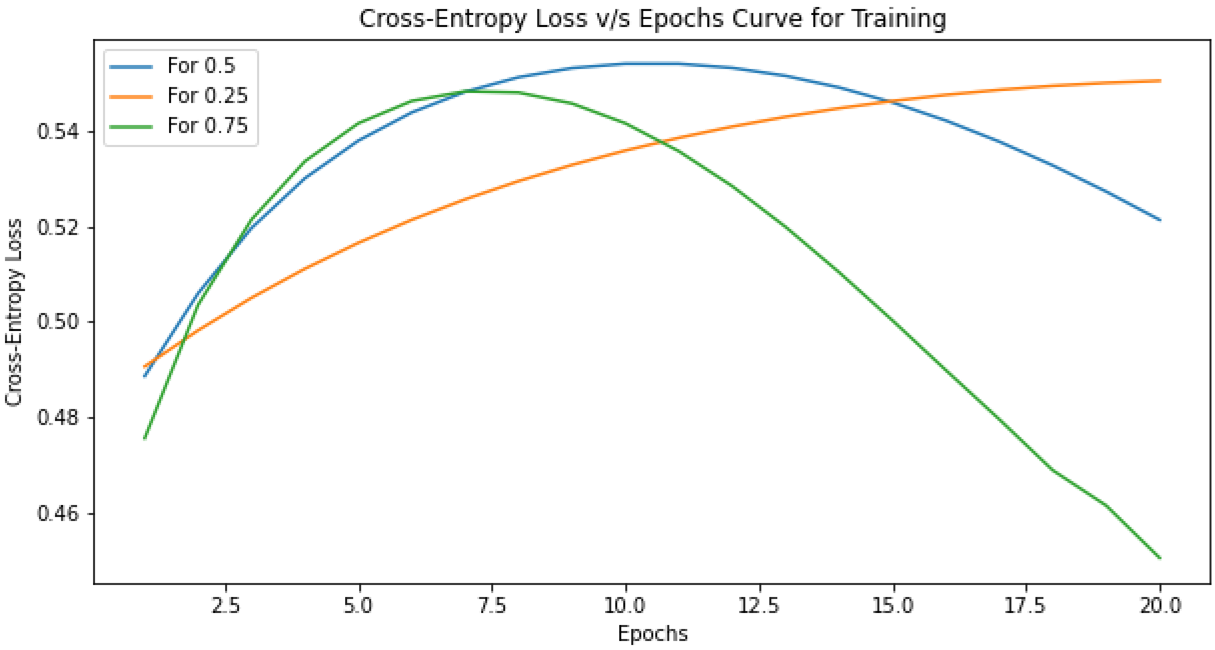
Implement back propagation neural network from scratch on **Letter Recognition datase**t with following configurations:-

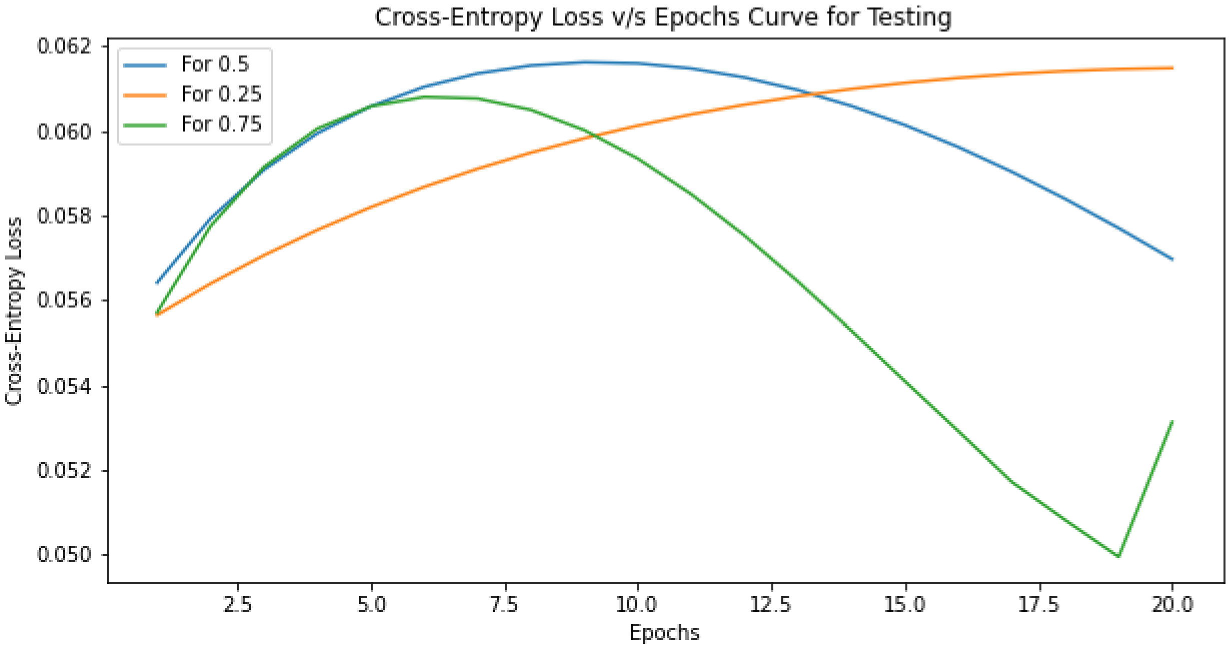
a. Vary learning rate
b. Vary number of epochs
c. Xavier weight initialisation
d. Use Adam optimizer
e. Using activation functions: tanh, ReLU

Use cross-entropy loss. Plot the loss and accuracy curves on the training and test sets. Find the best configuration for your network (a combination of best learning rate, best number of epochs, and activation function), support your claim.

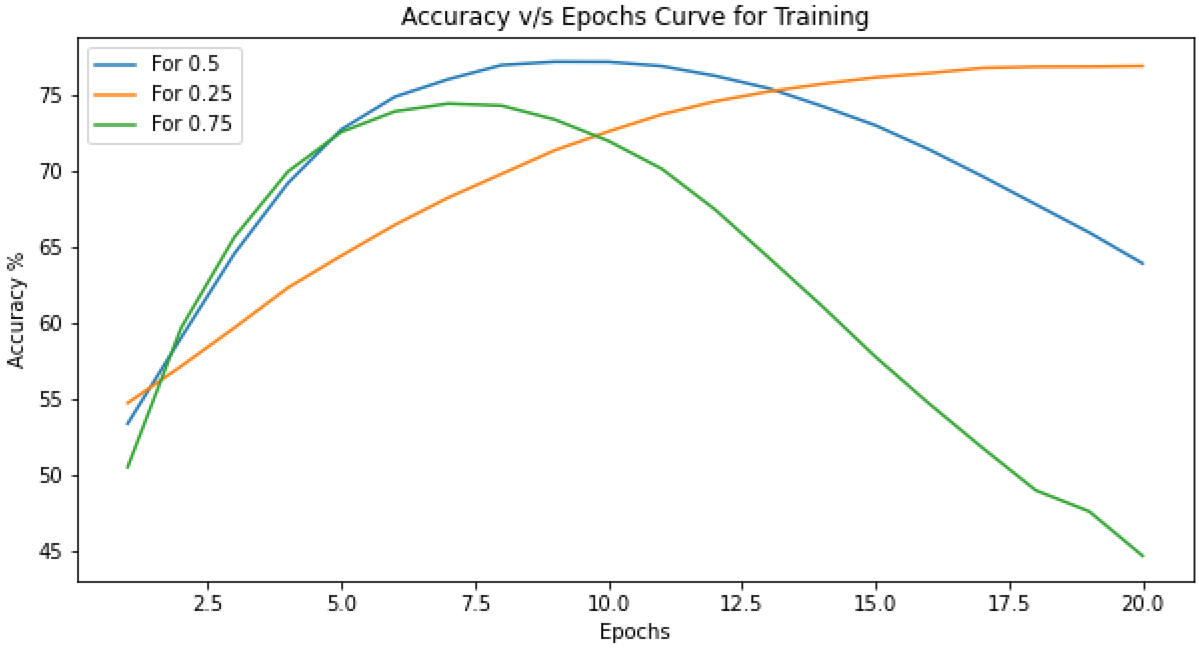
Soln 1.

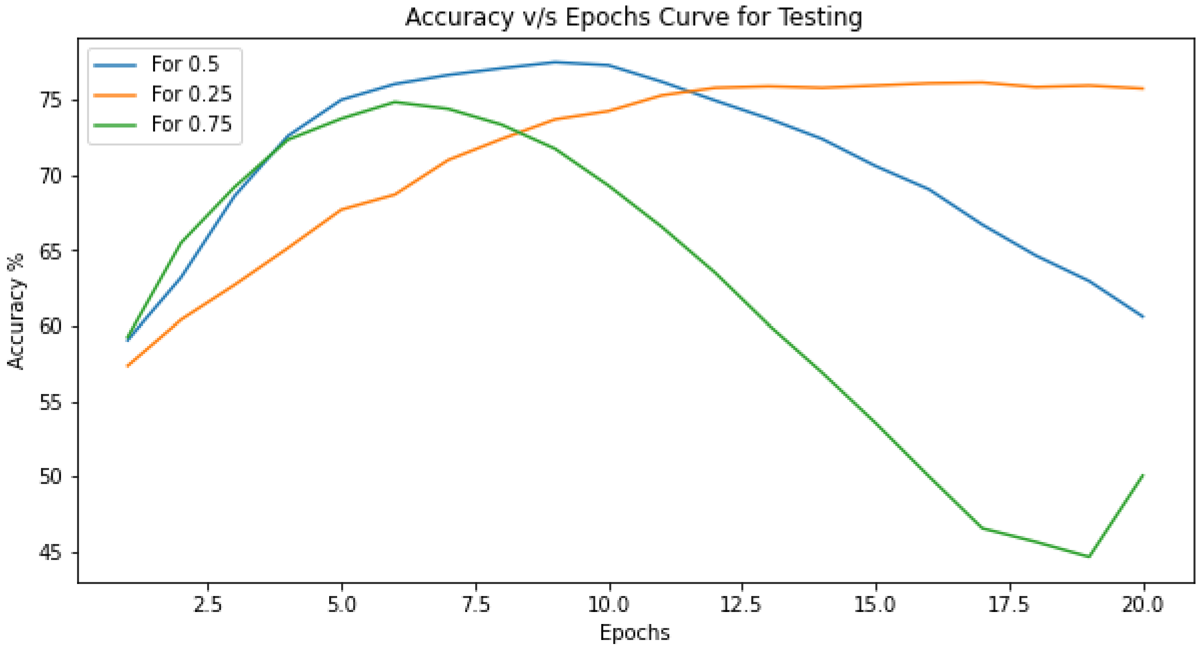
- **Loss Variation for varying lr**



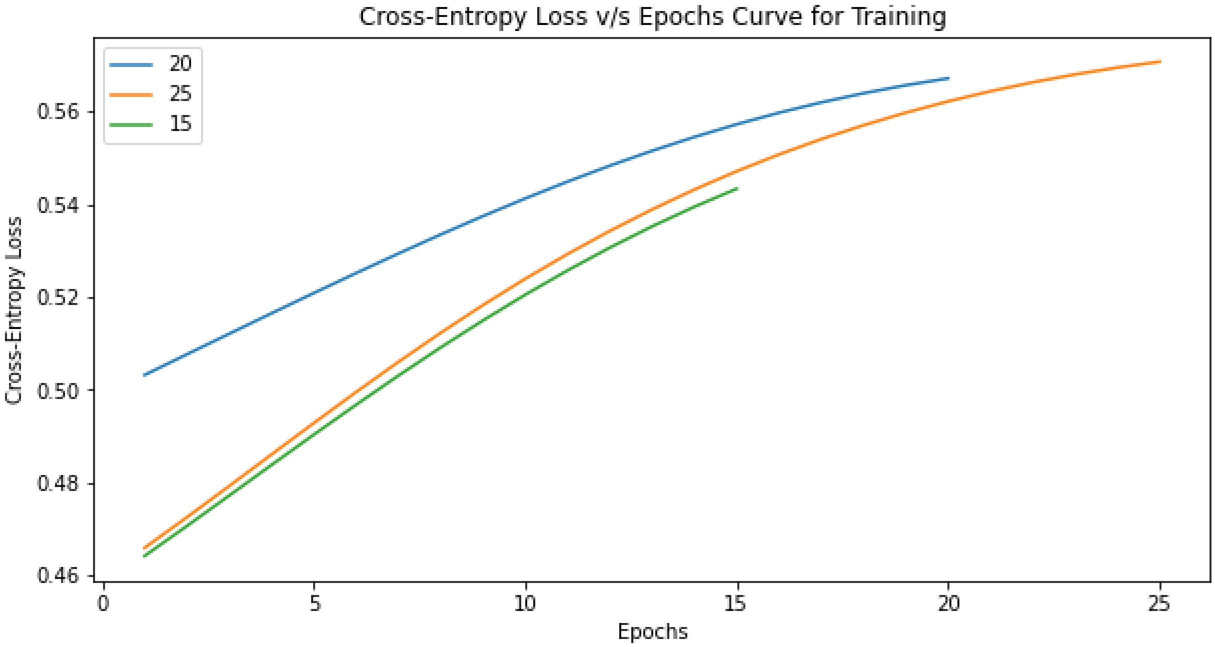


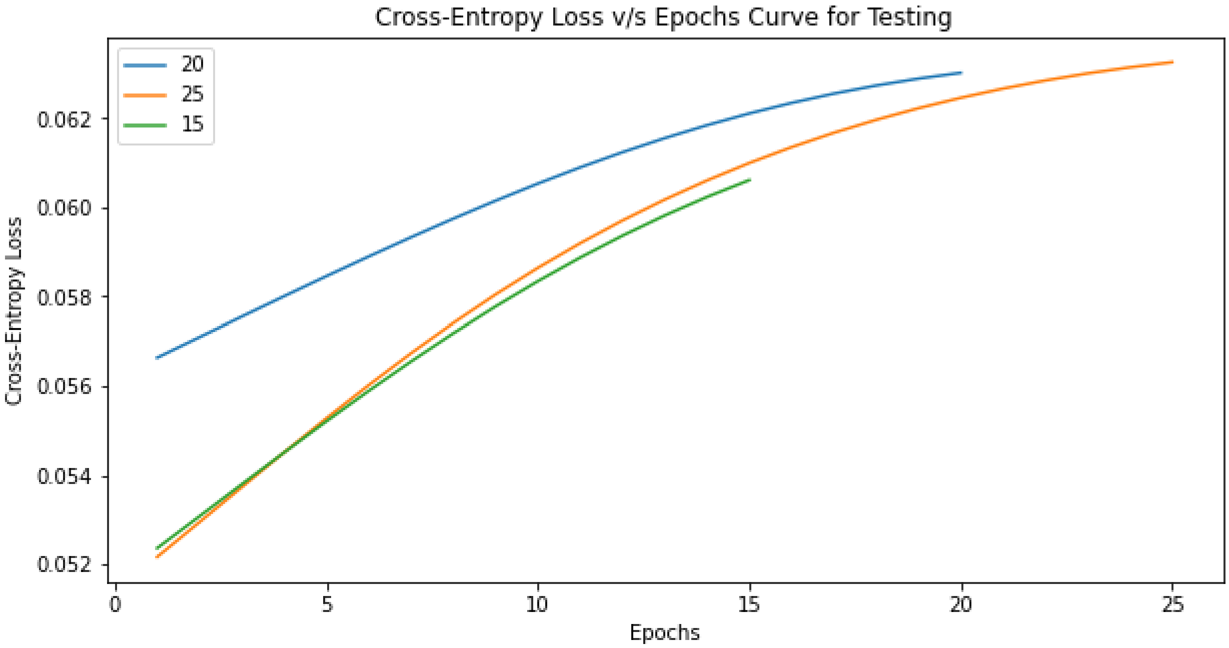
- **Accuracy Variations for varying lr**



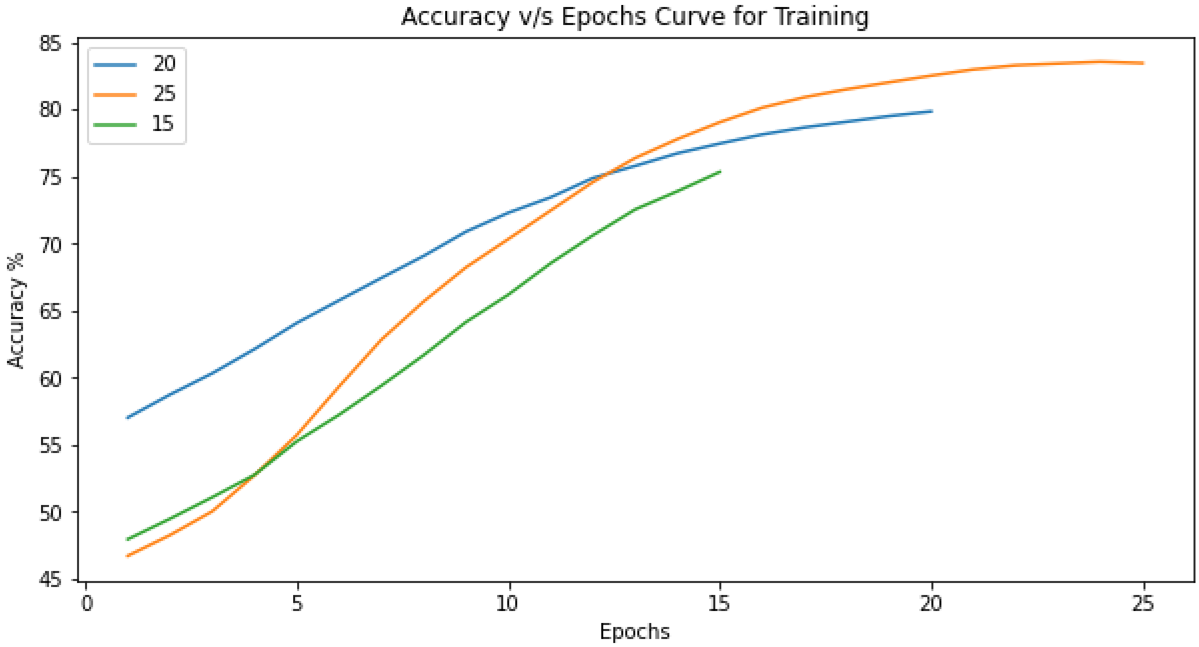


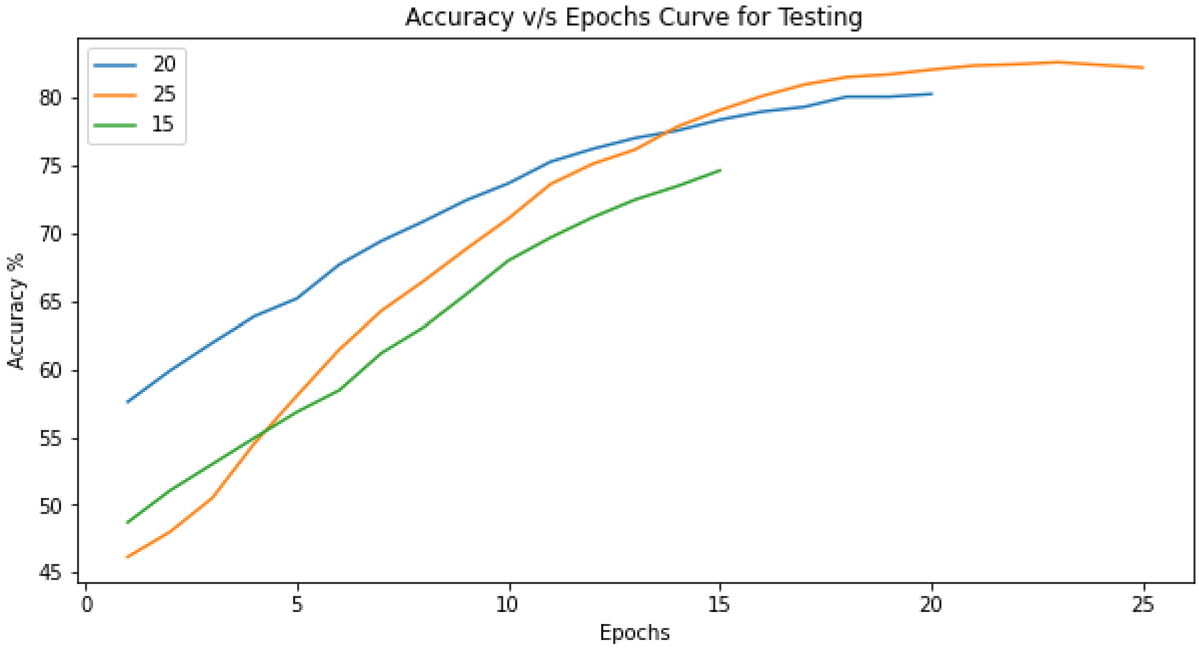
- **Loss Variation for varying epochs**



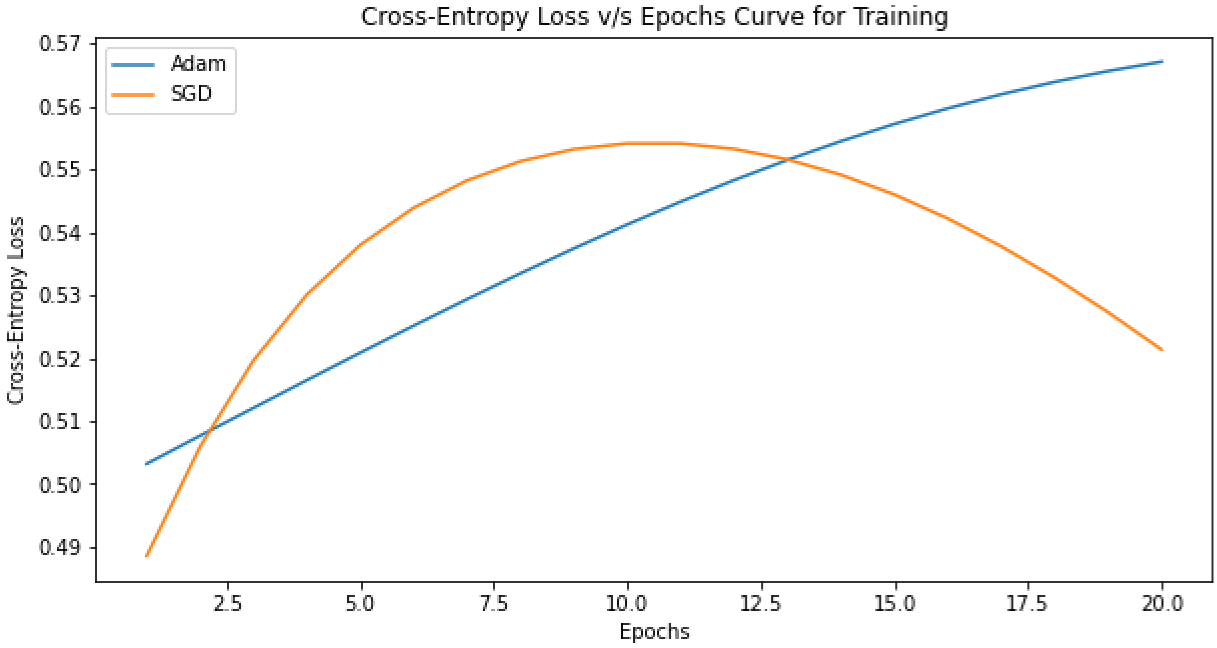


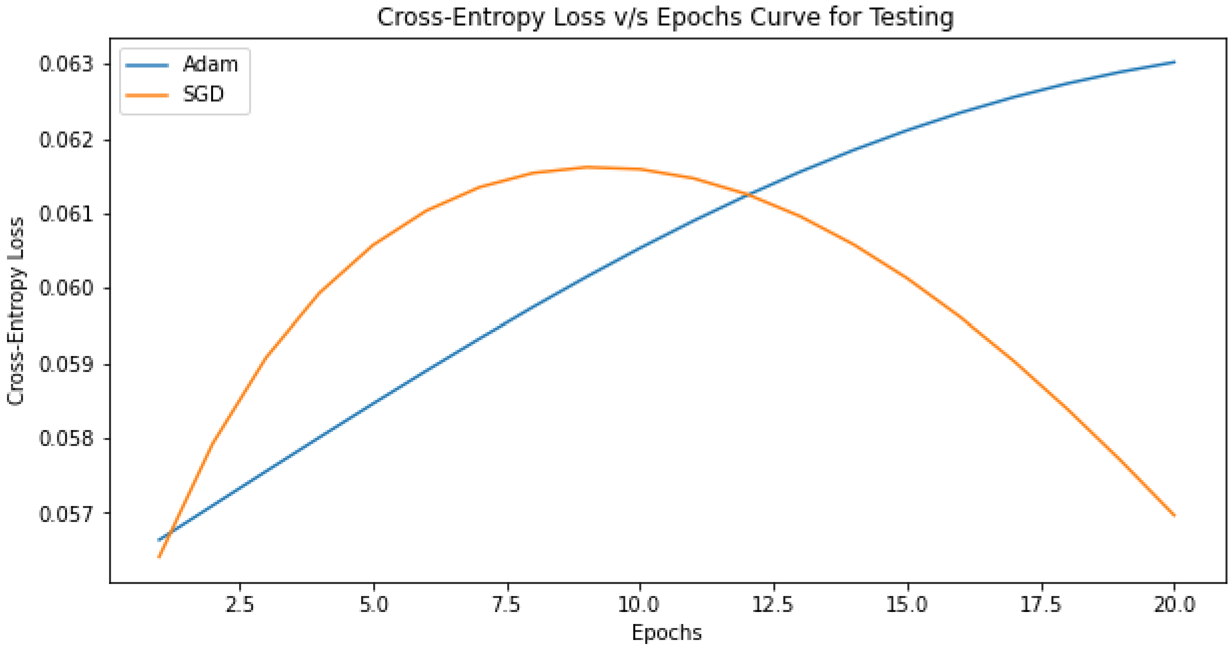
- **Accuracy Variations for varying epochs**



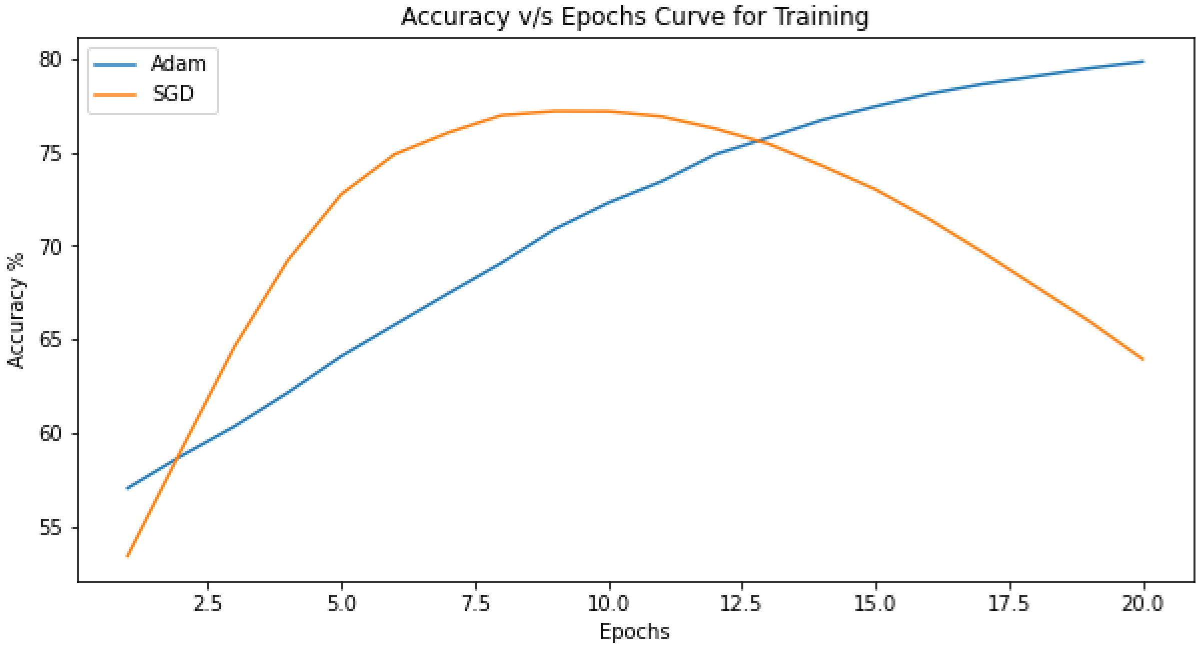


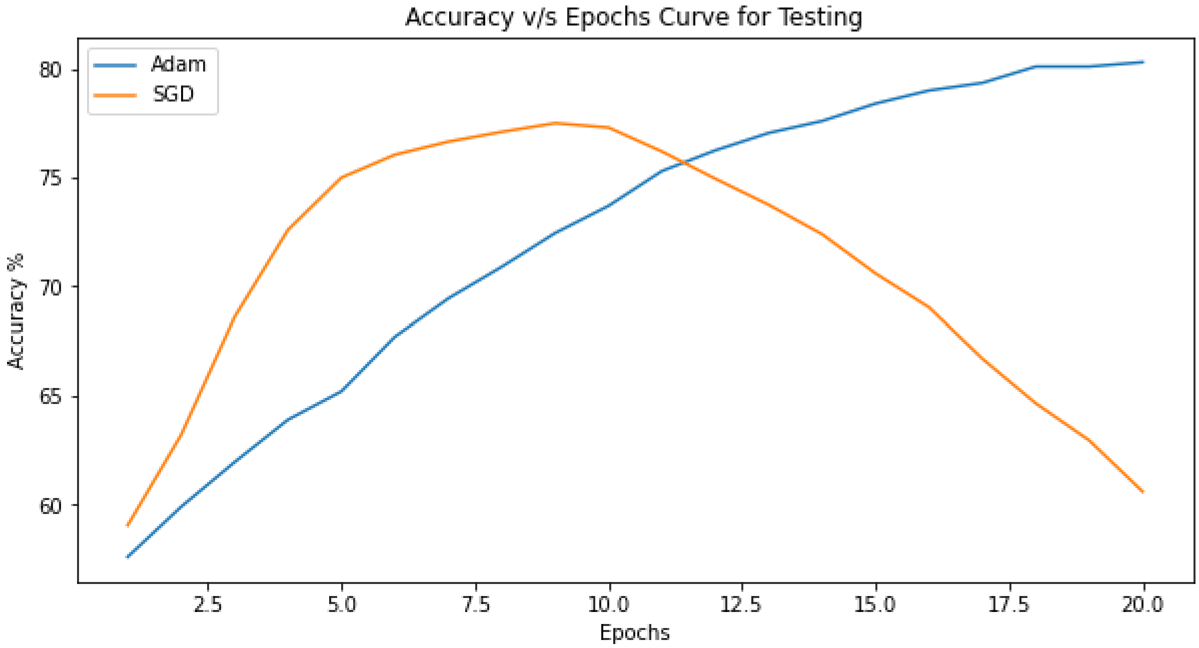
- **Loss Variation for varying optimizer**



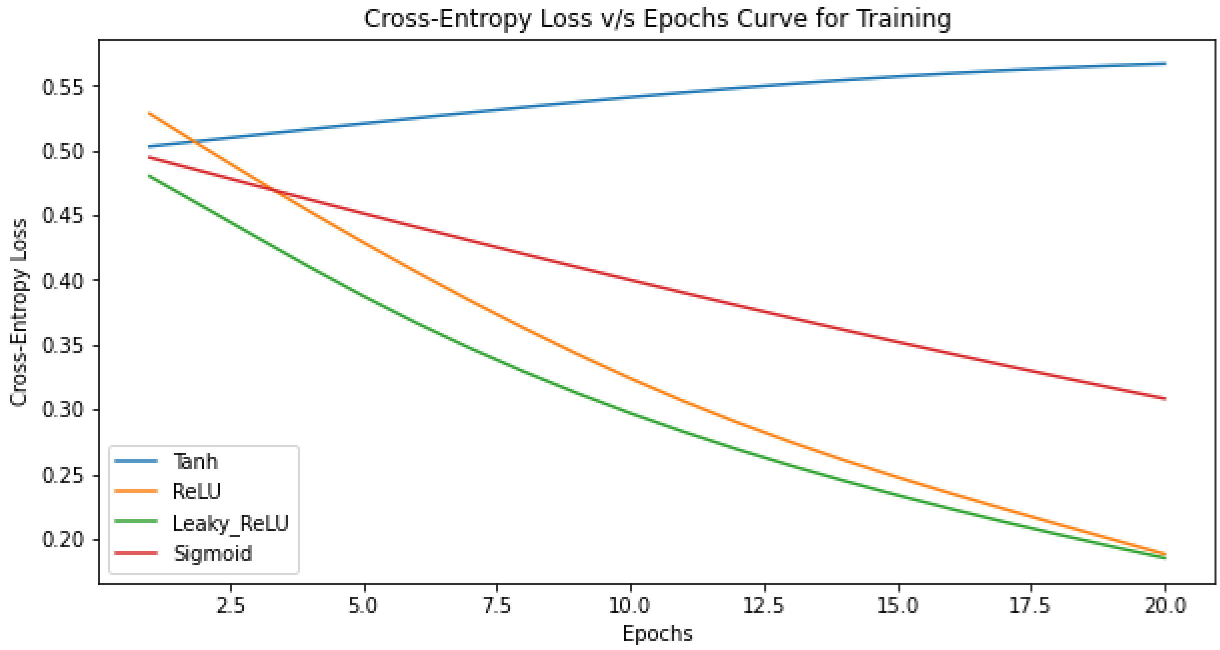


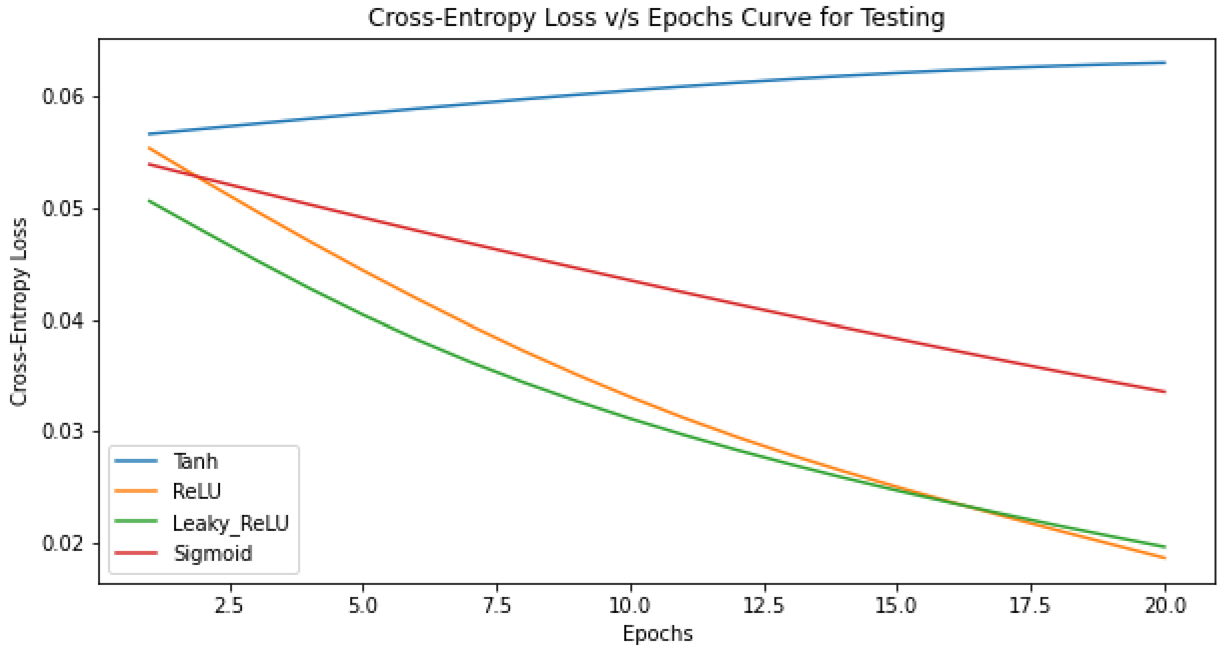
- **Accuracy Variations for varying optimizer**



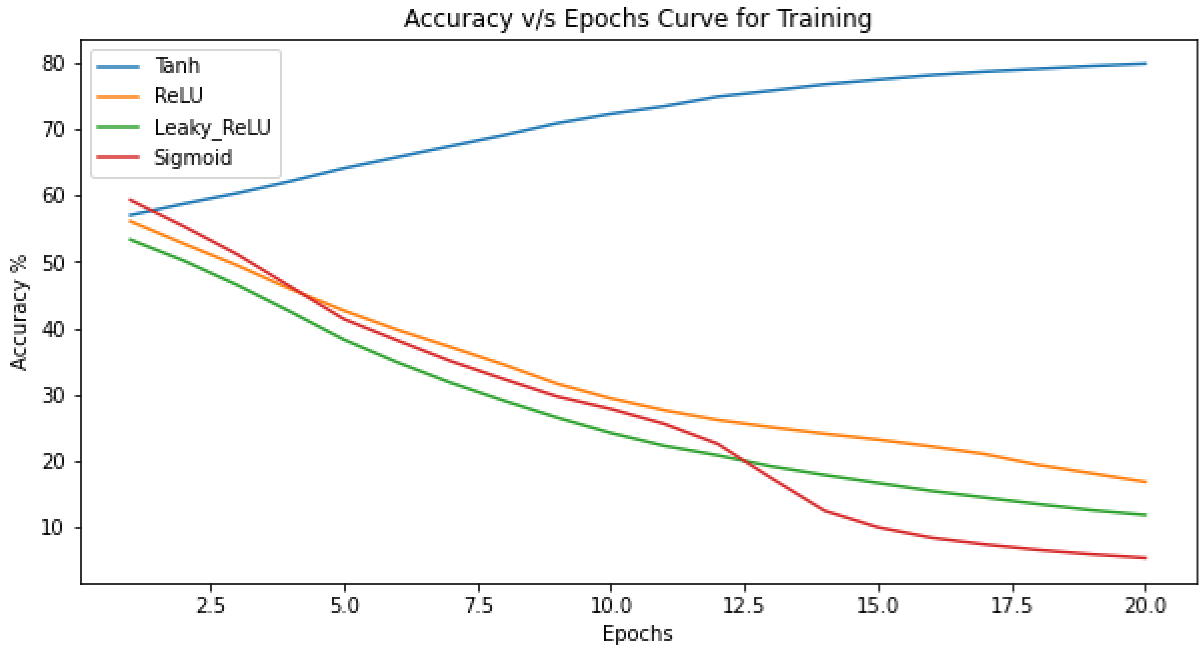


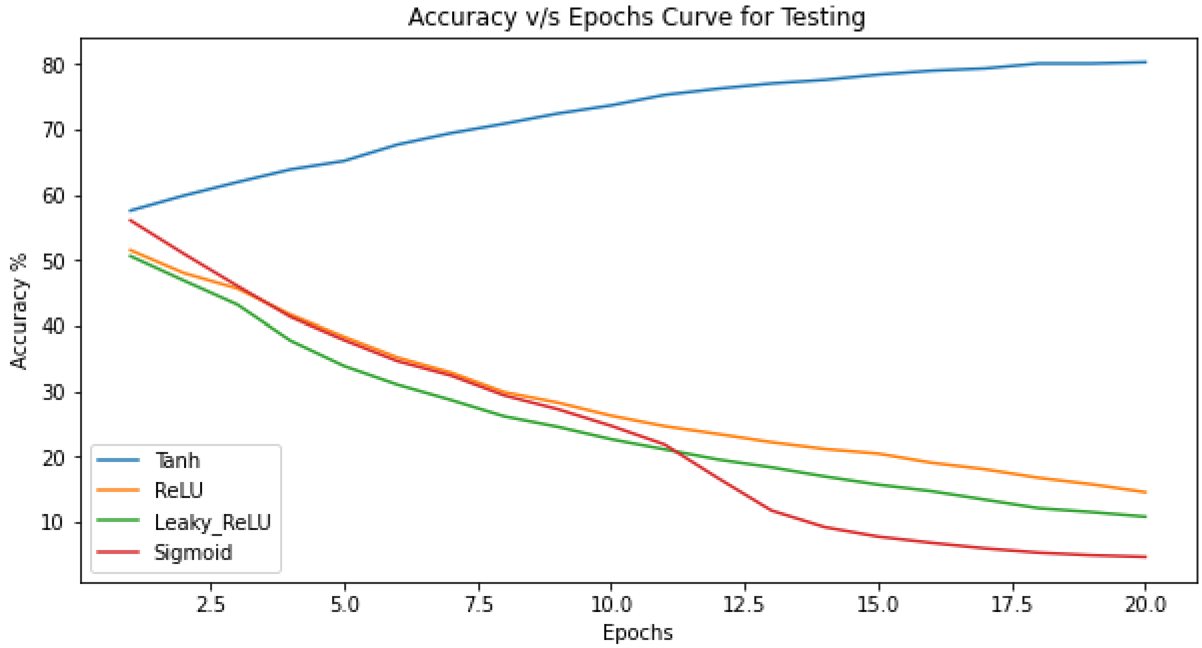
- **Loss Variation for varying activation functions**


Cross-Entropy Loss v/s Epochs Curve for Training


Cross-Entropy Loss v/s Epochs Curve for Testing

- **Accuracy Variations for varying activation functions**





Question-2

Implement a Denoising Autoencoder using three encoding and three decoding layers on the **MNIST dataset** for construction and reconstruction of the image. Use MSE as the loss function:-

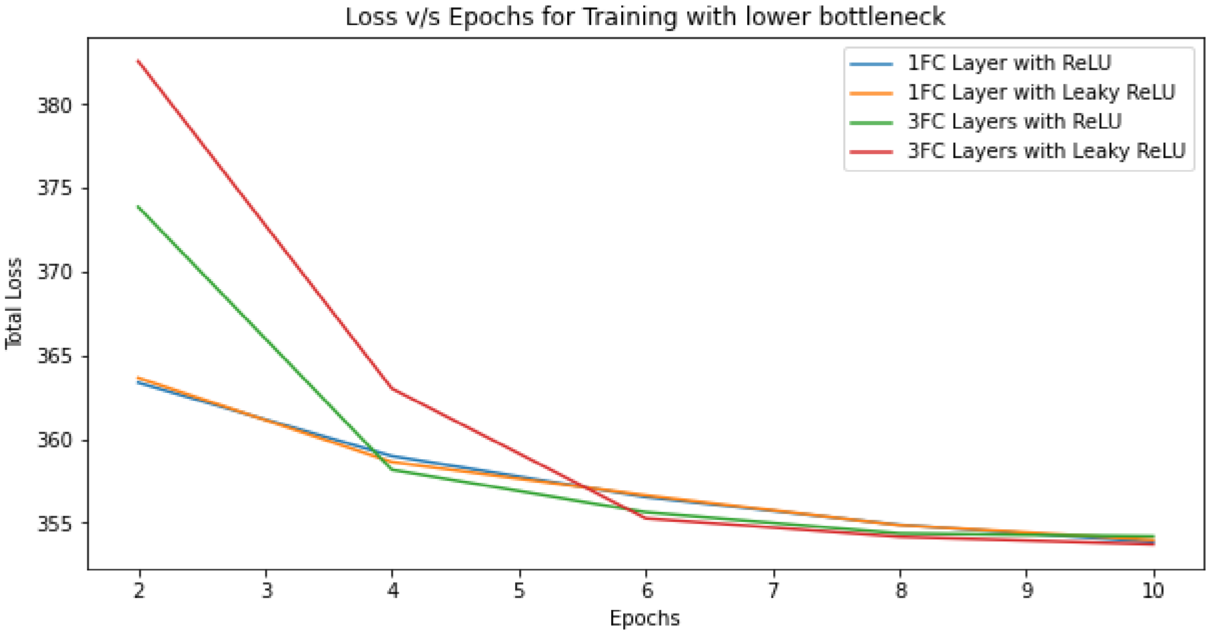
a. Use 1 FC layer and 2 different activation functions of your choice for 10-class classification

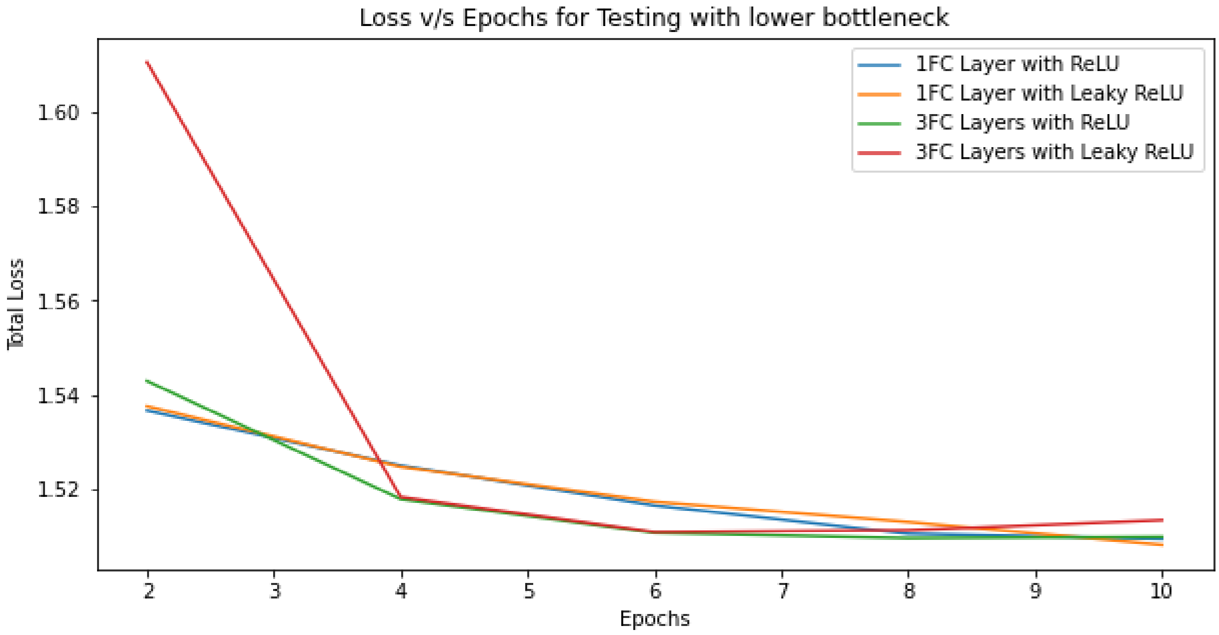
b. Use 3 FC layers and 2 different activation functions of your choice for 10-class classification

c. Compare the performance between 1FC and 3 FC layer results and report the accuracy on test set and plot loss curves on training and test dataset

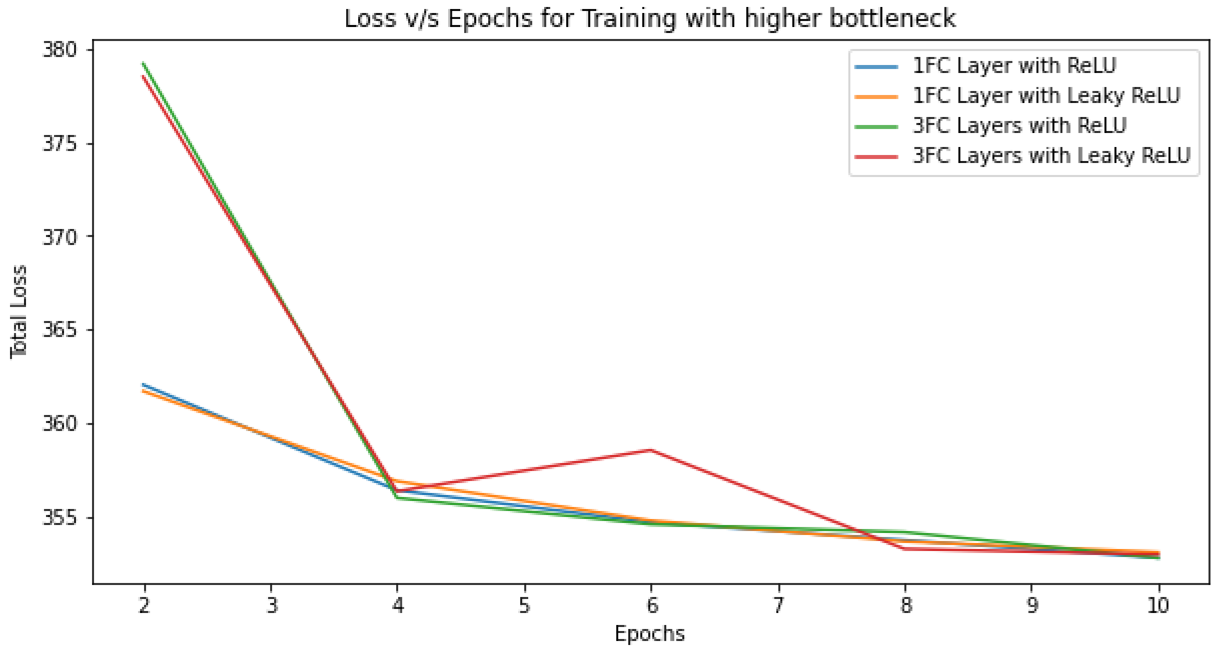
Soln 2.

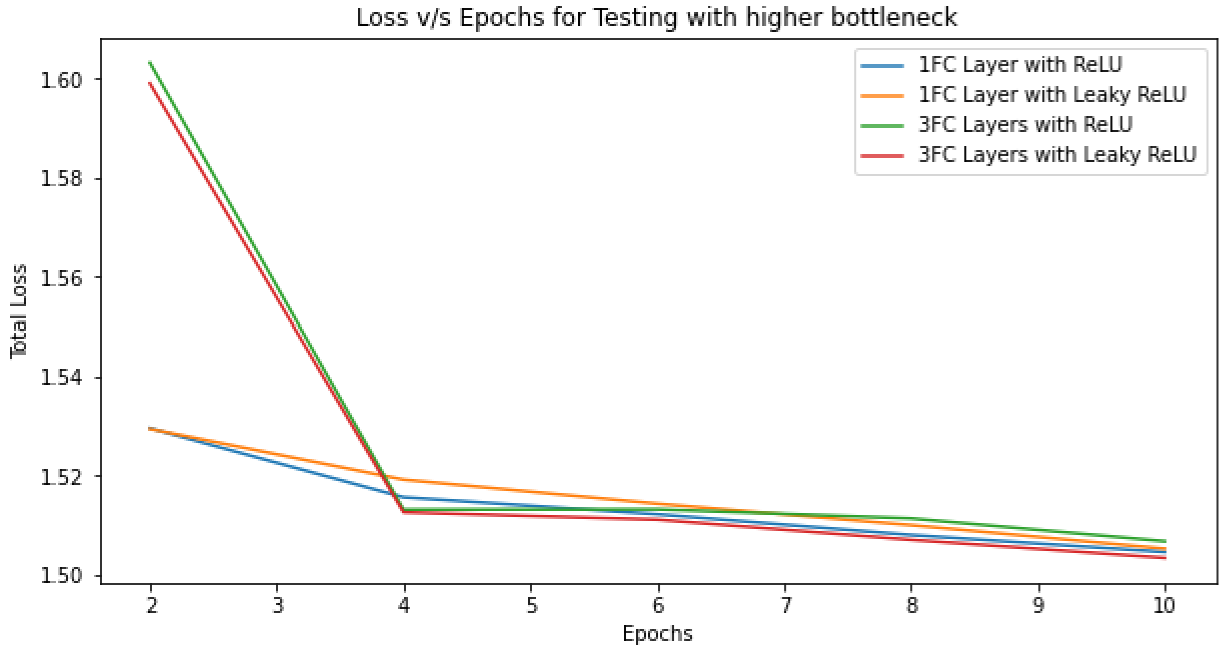
- **Graphs showing how loss varies if we increase the number of epochs for 1 FC layer and 3 FC layers with different activation functions, namely ReLU, and Leaky ReLU for lower bottleneck size.**



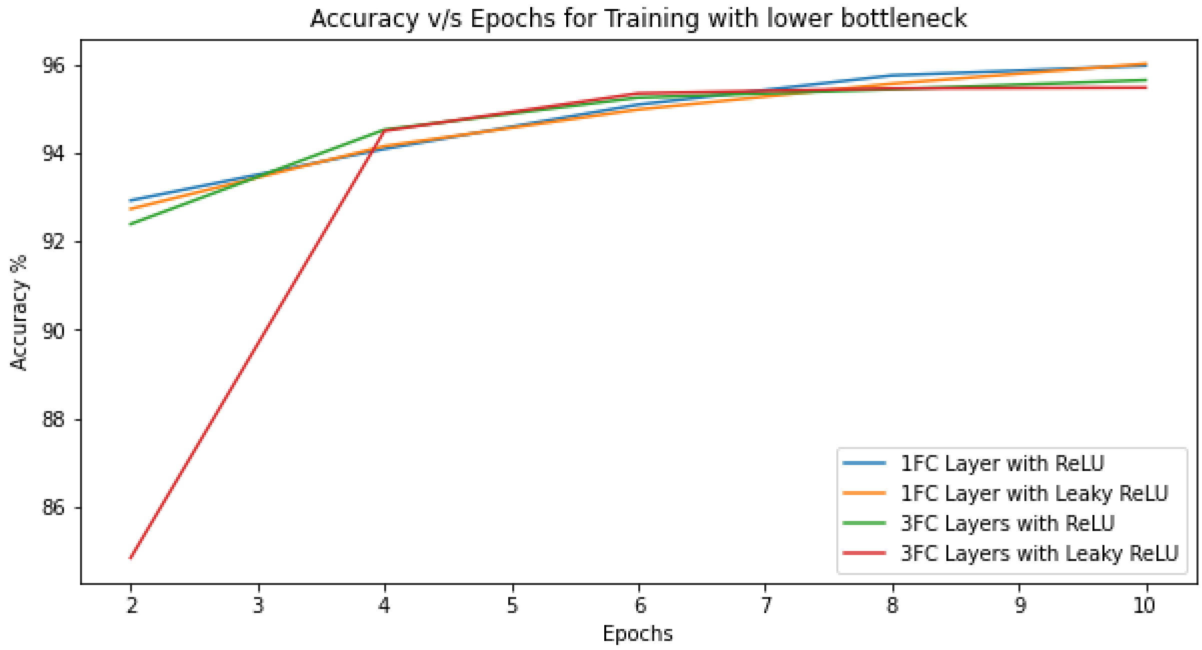


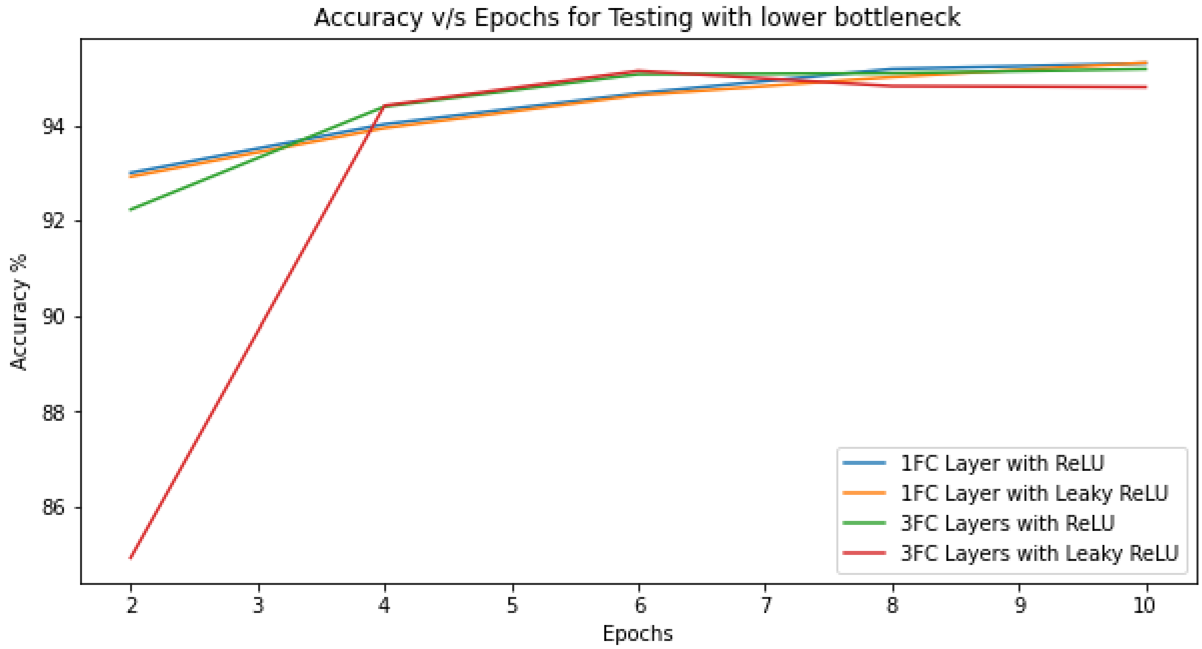
- **Graphs showing how loss varies if we increase the number of epochs for 1 FC layer and 3 FC layers with different activation functions, namely ReLU, and Leaky ReLU for higher bottleneck size.**



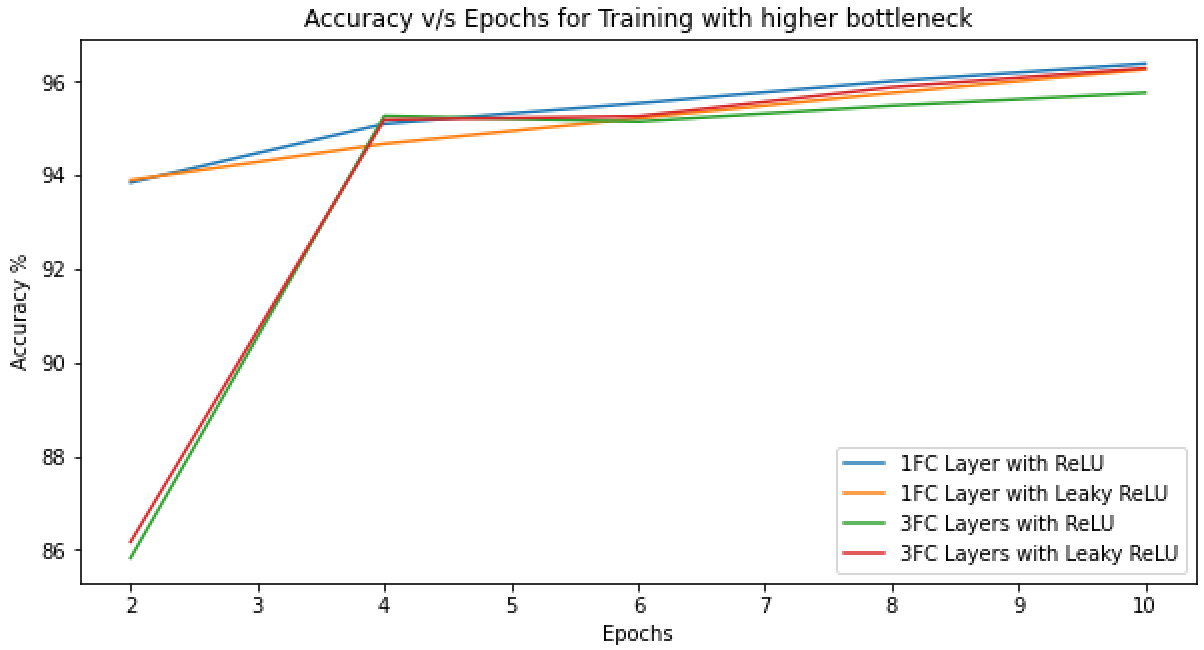


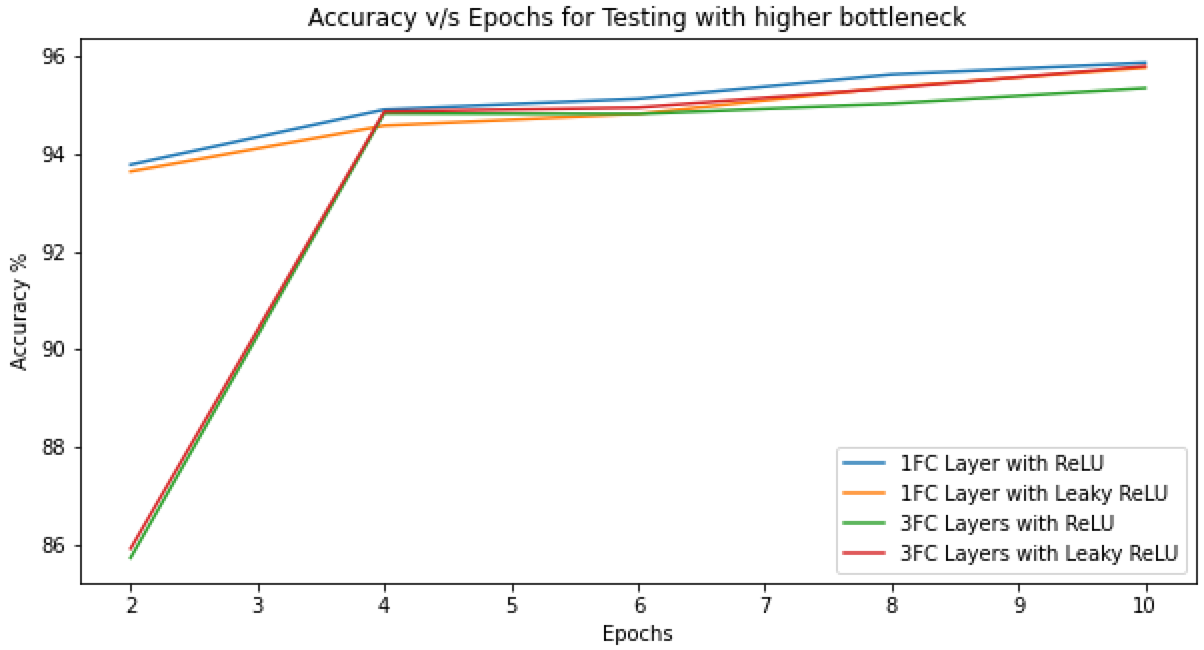
- **Graphs showing how accuracy varies if we increase the number of epochs for 1 FC layer and 3 FC layers with different activation functions, namely ReLU, and Leaky ReLU for lower bottleneck size.**


Accuracy v/s Epochs for Training with lower bottleneck


Accuracy v/s Epochs for Testing with lower bottleneck

- **Graphs showing how accuracy varies if we increase the number of epochs for 1 FC layer and 3 FC layers with different activation functions, namely ReLU, and Leaky ReLU for higher bottleneck size.**





Question-3

Construct following CNN architectures. Use the **CIFAR-10 dataset** for all the analysis:-

1. Conv-Conv-Pool-Conv-Conv-Pool-FC-FC
2. Conv-Pool-BatchNormalization-ReLU-Conv-Pool-BatchNormalization-ReLU-FC

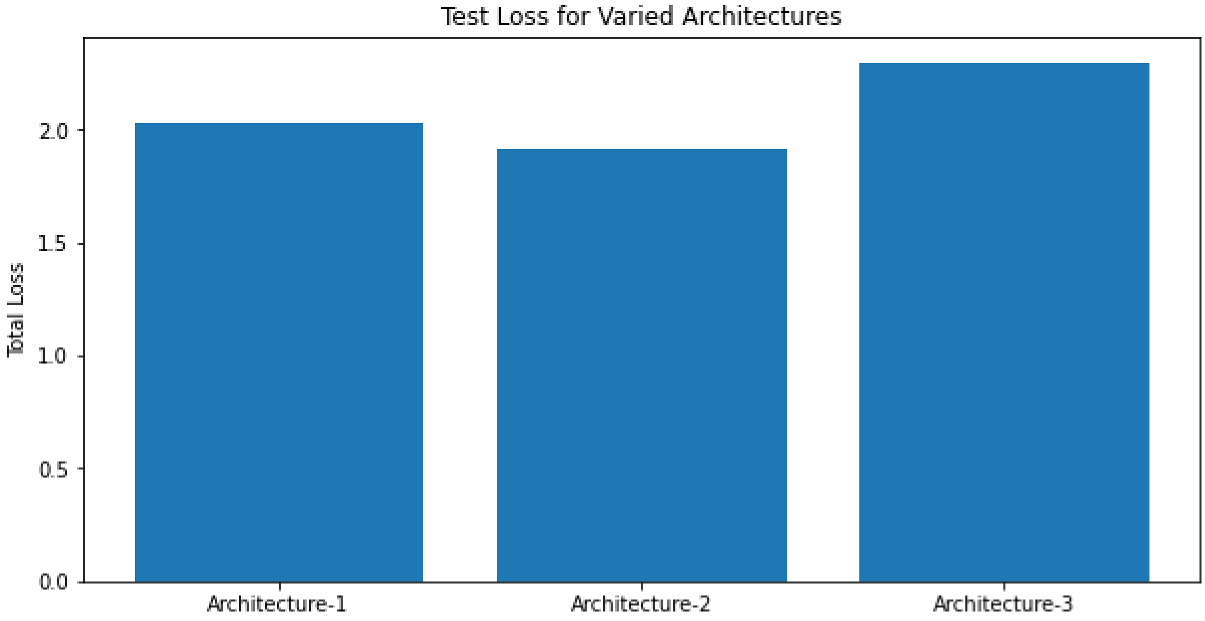
3. Conv-BatchNorm-ReLU-Conv-BatchNorm-ReLU-FC

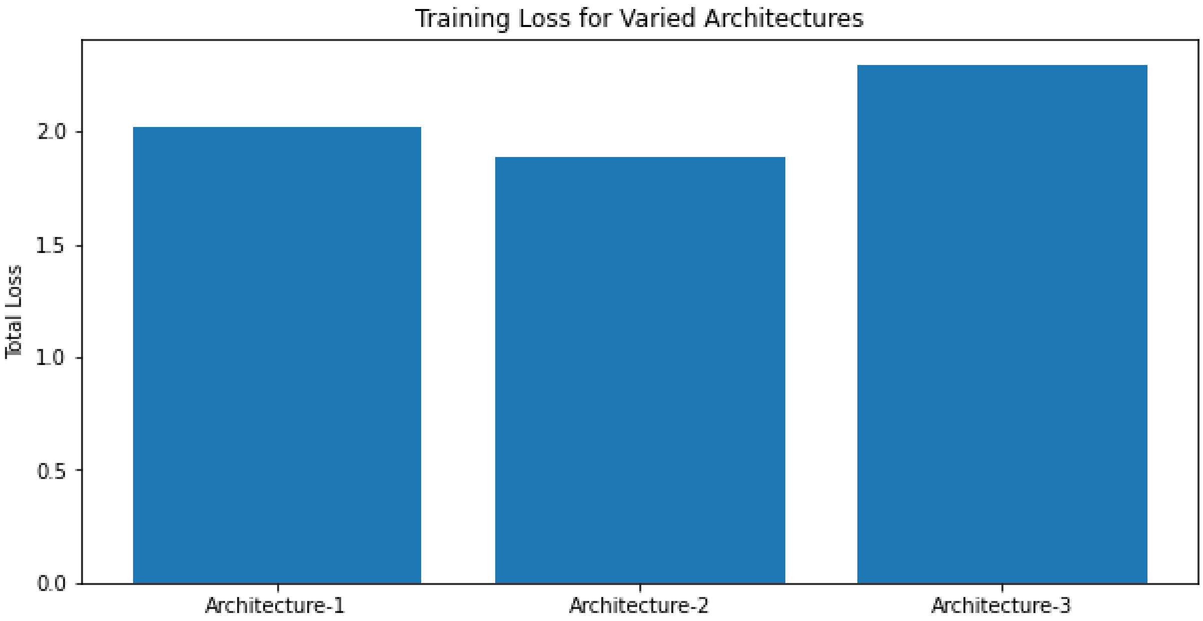
FC refers to Fully Connected Layer here, provide with the following analysis:-

a. How does changing network size affect the performance ?
b. Experiment with different sizes and types of pooling like (Max Pooling, Average Pooling and Stochastic Pooling(optional)) and do a detailed analysis of pooling size on the network.
c. How the presence of one or more fully connected layers changes the accuracy.
d. Change the stride size from 1 to 2 and find the differences in the output size.

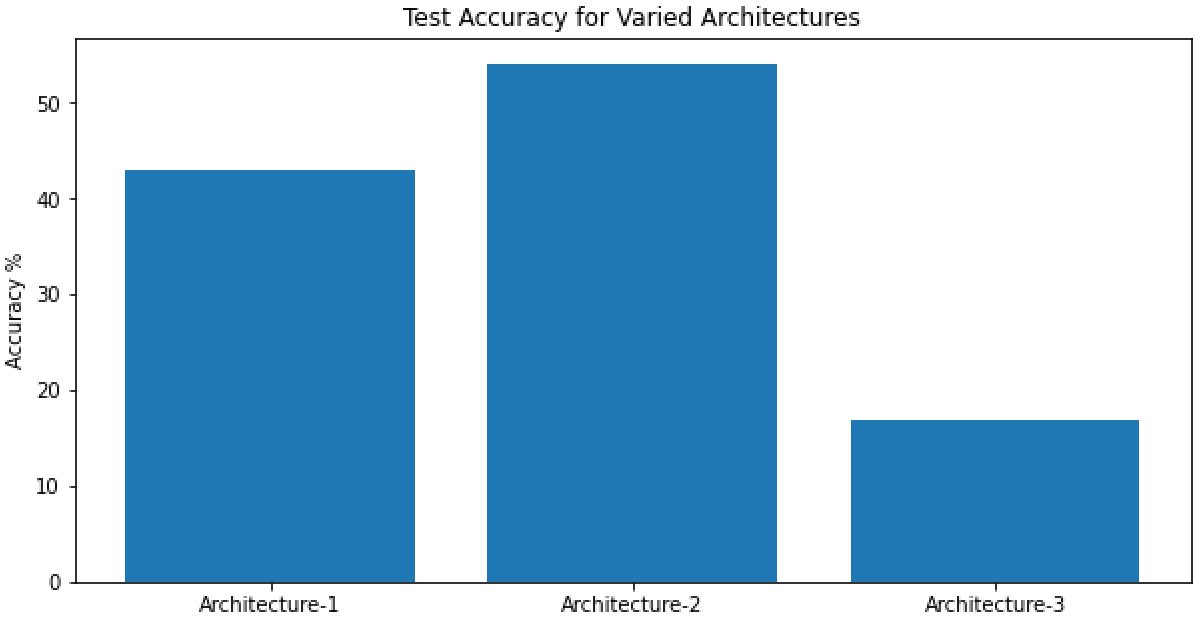
Soln 3.

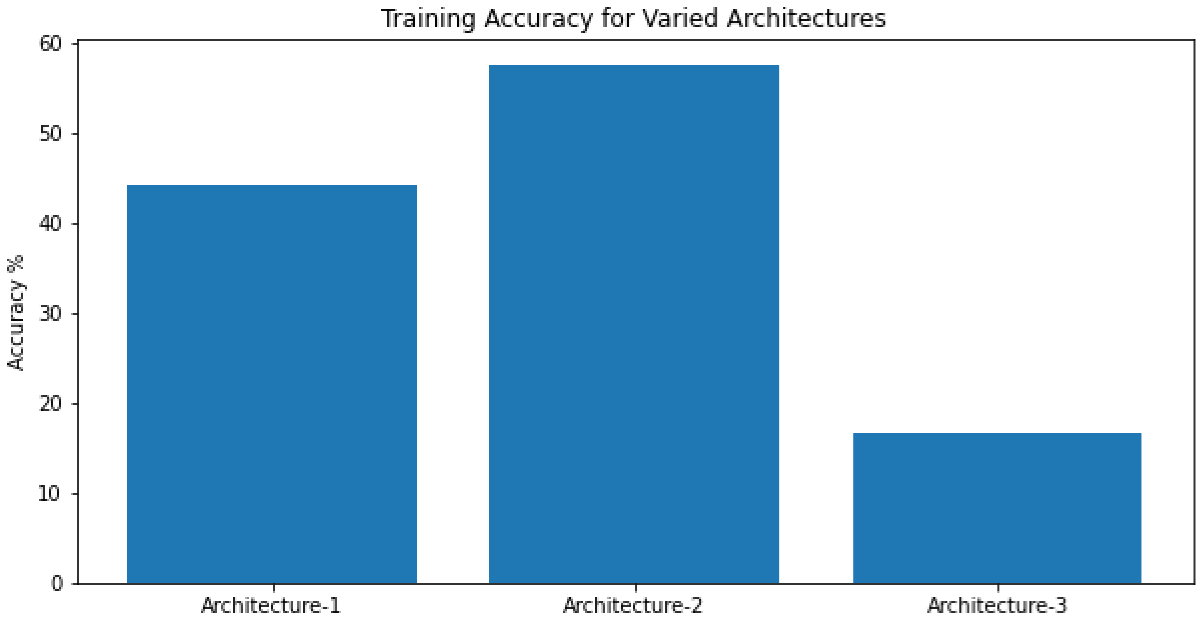
- **Graphs showing how loss varies if we vary the architectural choices**



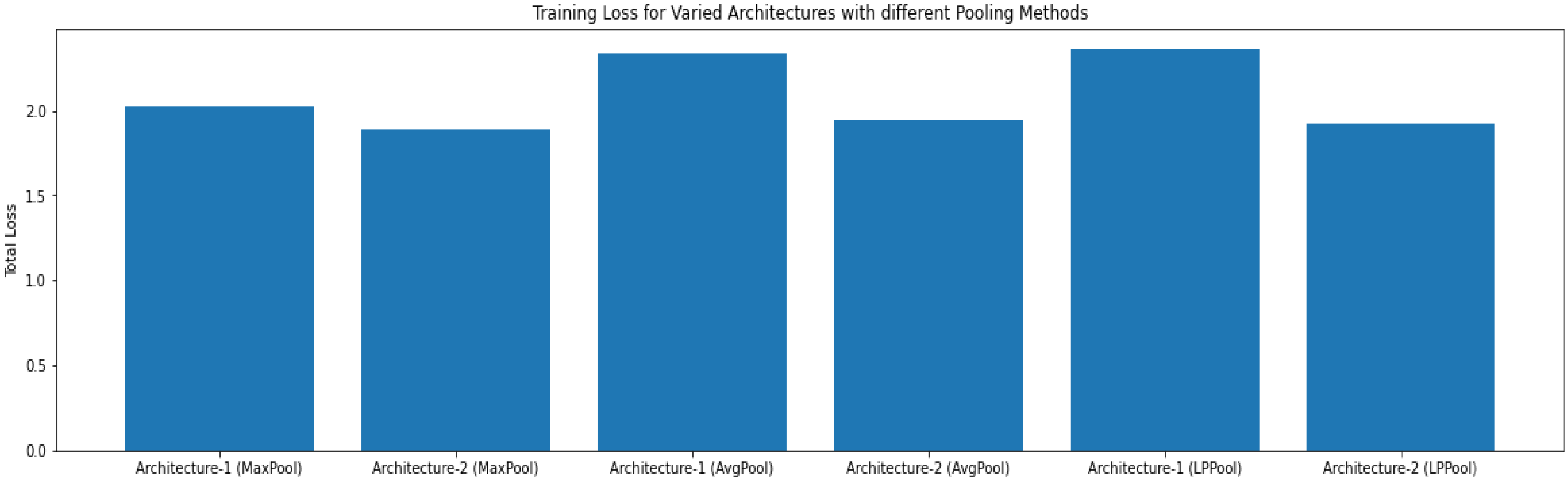


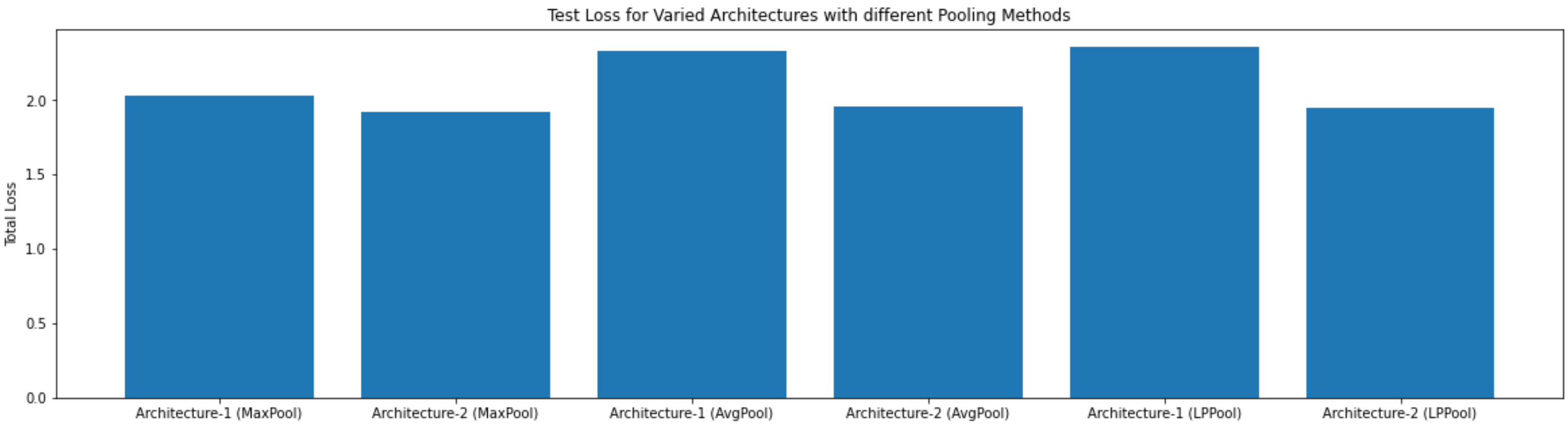
- **Graphs showing how accuracy varies if we vary the architectural choices**



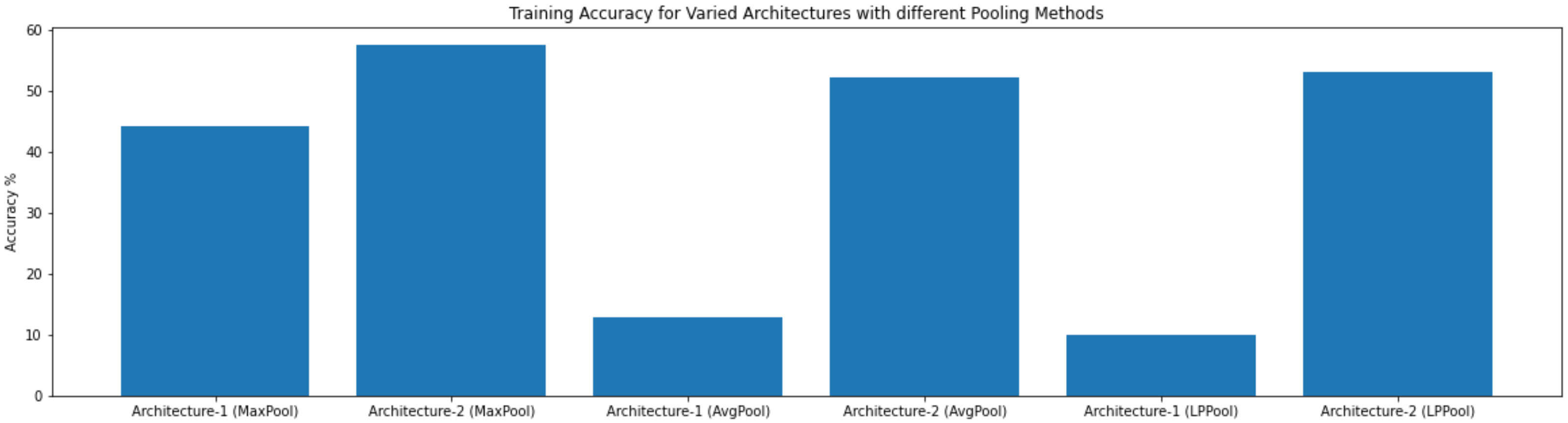


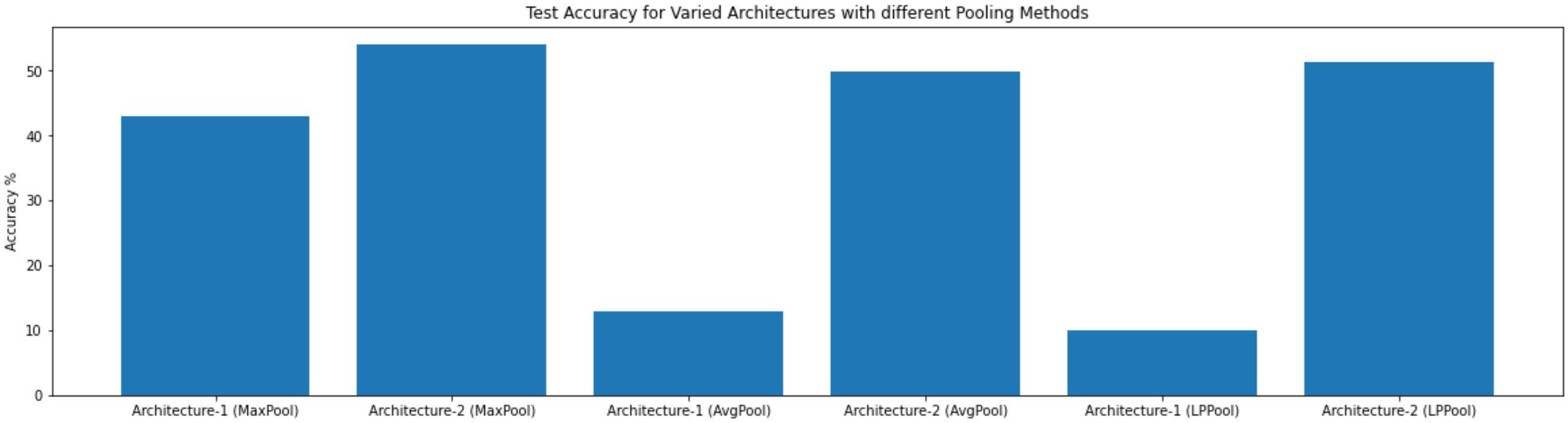
- **Graphs showing how loss varies if we vary the pooling choices**



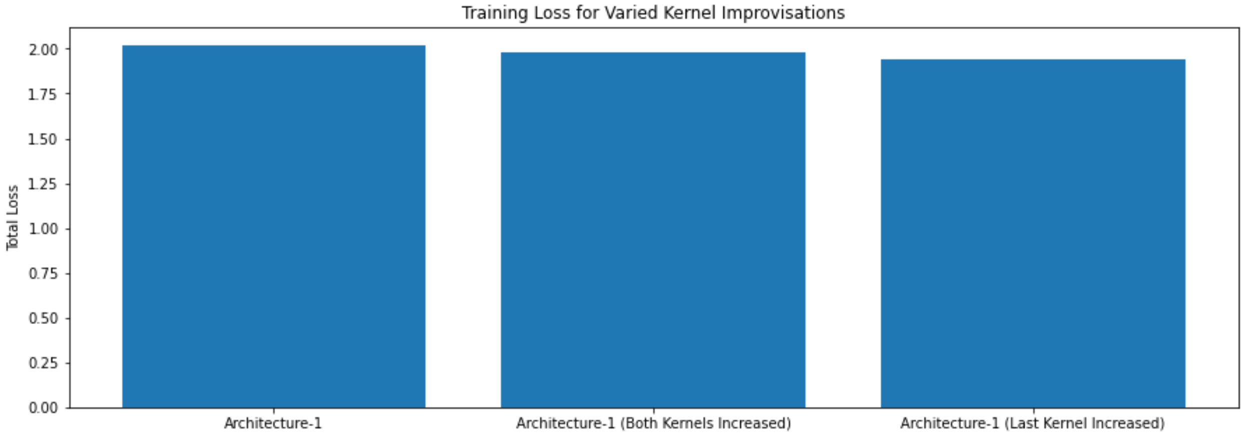


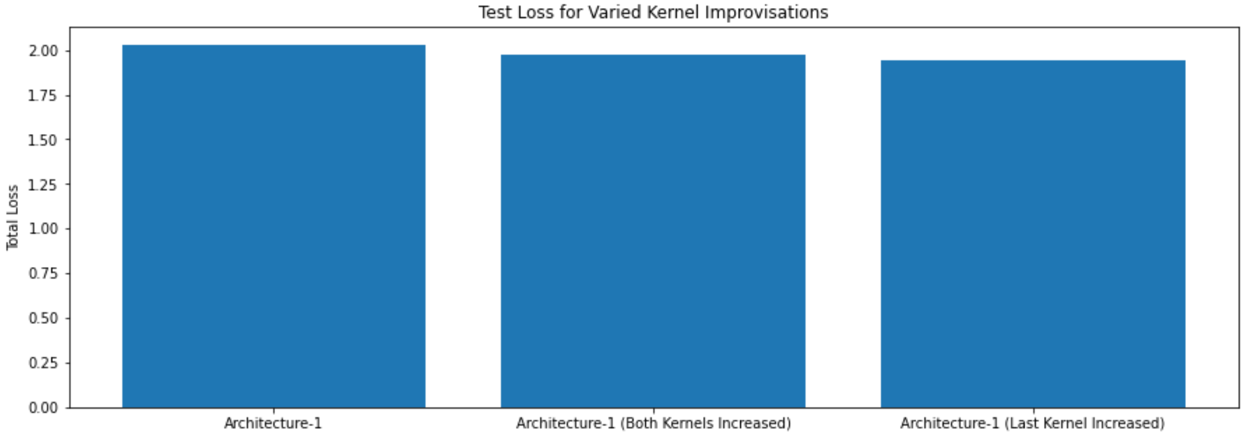
- **Graphs showing how accuracy varies if we vary the pooling choices**



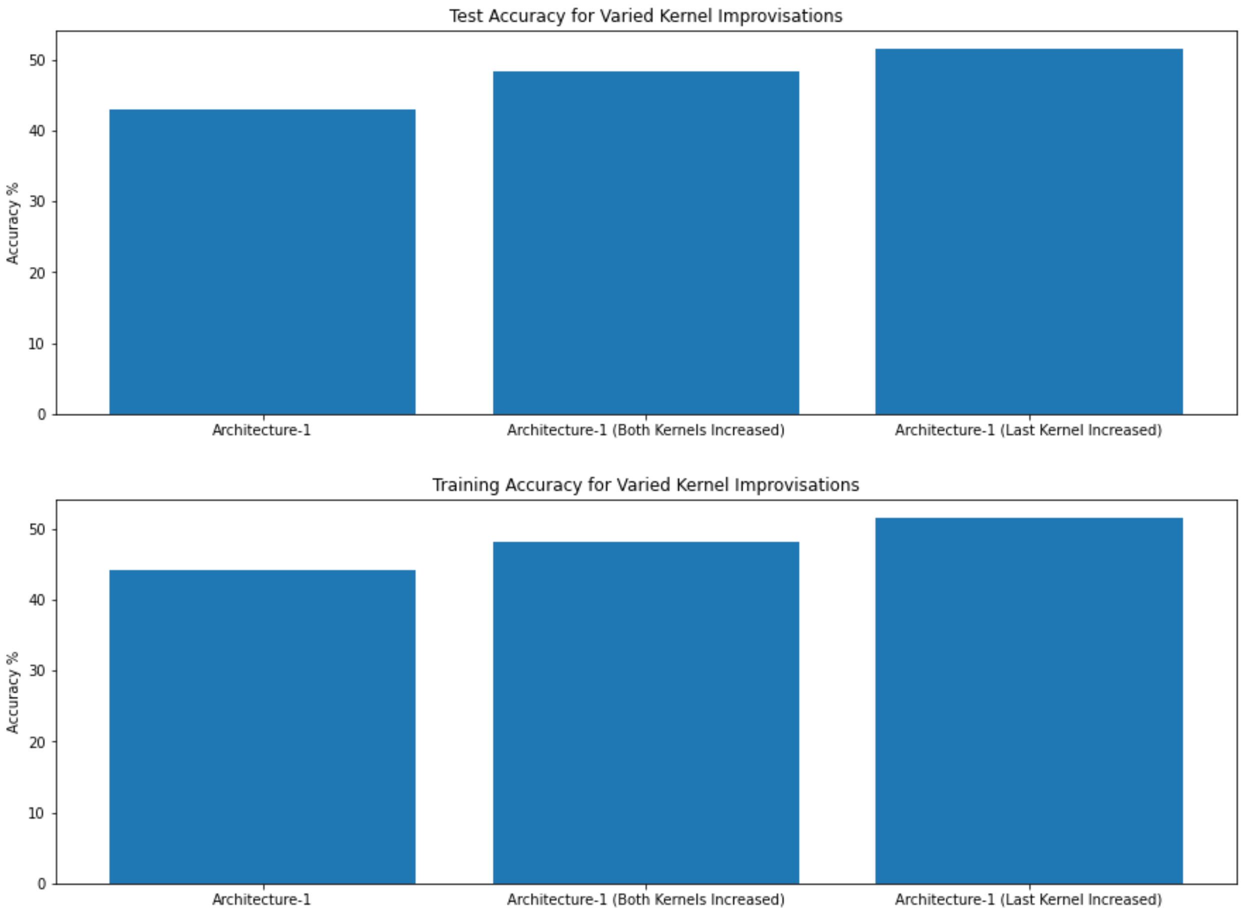


- **Graphs showing how loss varies if we vary the convolution kernel size choices**

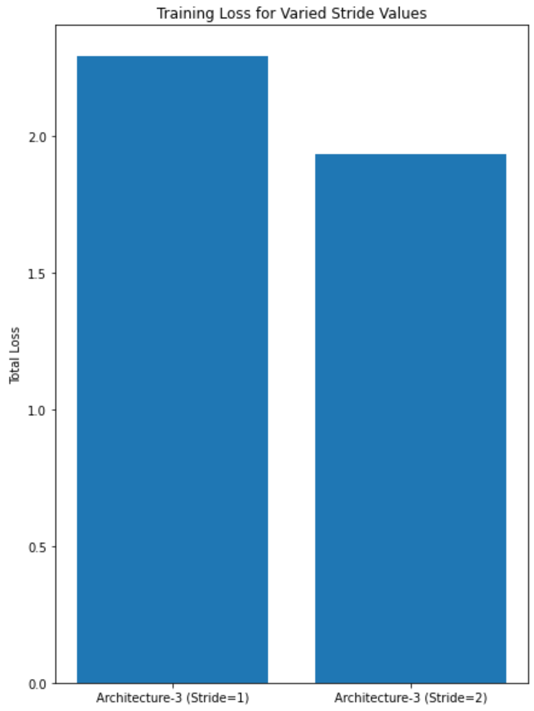


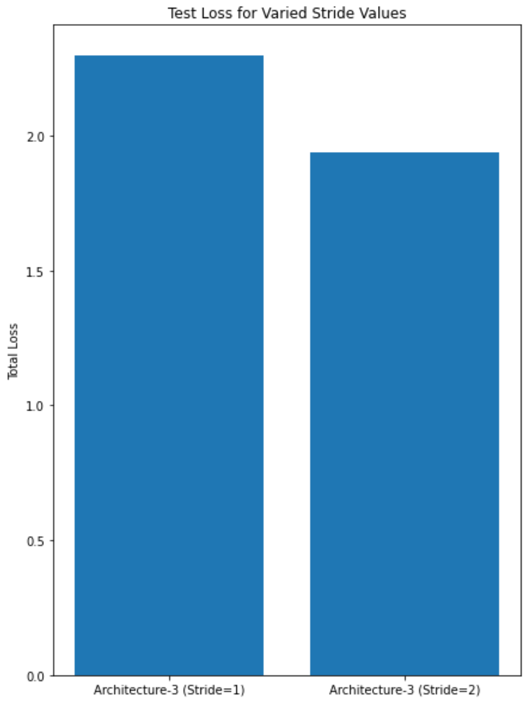


- **Graphs showing how accuracy varies if we vary the convolution kernel size choices**

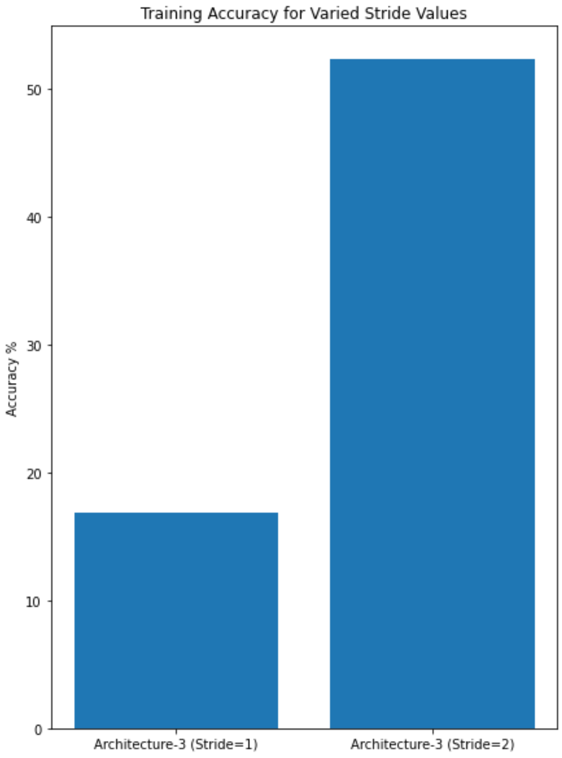


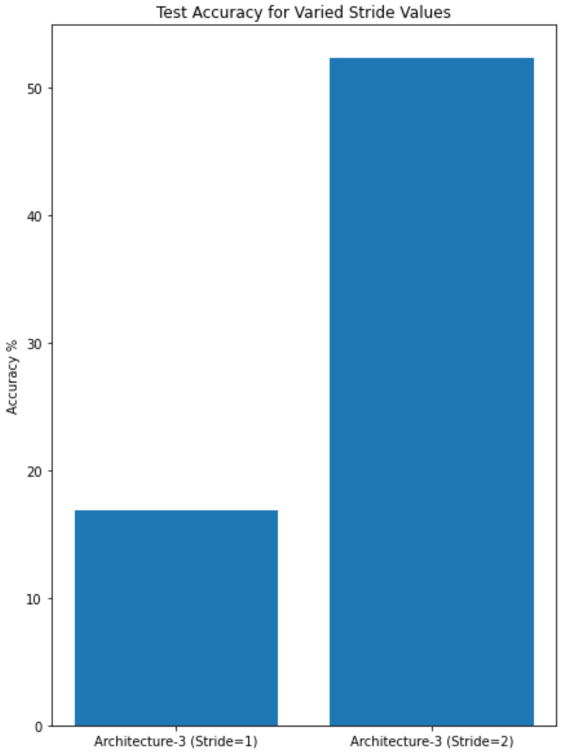
- **Graphs showing how loss varies if we vary the stride from 1 to 2**





- **Graphs showing how accuracy varies if we vary the stride from 1 to 2**





References

For Q-1 :-

- https://github.com/jiexunsee/Adam-Optimizer-from-scratch/blob/master/adamoptimizer.py
- https://towardsdatascience.com/the-ultimate-beginners-guide-to-implement-a-neural-network-from-scratch-cf7d52d91e00

For Q-2:-

- https://www.geeksforgeeks.org/implementing-an-autoencoder-in-pytorch/
- https://ai.plainenglish.io/denoising-autoencoder-in-pytorch-on-mnist-dataset-a76b8824e57e
- https://gist.github.com/kdubovikov/eb2a4c3ecadd5295f68c126542e59f0a
- https://nextjournal.com/gkoehler/pytorch-mnist

For Q-3:-

- https://shonit2096.medium.com/cnn-on-cifar10-data-set-using-pytorch-34be87e09844
- https://pytorch.org/docs/stable/index.html