Question 2: (NOTE: Module names are for Pytorch, you are free to use Keras,Tensorflow or any other library, the steps will be same)

Use the MNIST dataset for all the experiments. In this question, you will implement a fully functioning under complete autoencoder for feature learning. Use a 70:30 data split.

Perform the following tasks:

(a) Load the dataset using " torchvision.datasets" library, preprocess the dataset using "torchvision.transforms" library and finally create a train loader and test loader of some batch size using "torch.utils.data" library.
(b) Create an autoencoder class with decoder and encoder architecture and a hidden neuron representation.
(c) Initialise an optimizer using "torch.optim" library and you can also use a learning rate scheduler using "torch.optim.lr_scheduler" library.
(d) Write a training loop over epochs and over the batches of the train set and calculate the average loss between the reconstructed image and the input image.
(e) Write a testing loop over the batches of the test set and calculate the loss between the reconstructed image and the input image.
(f) You can save the best performing model using "torch.save".

Report the test reconstruction loss and plot the train loss vs epochs for the following cases:

1) Perform the experiment using different optimizers like Adam, RMSProp, SGD with momentum, SGD without momentum.
2) Plot the test reconstruction loss vs hidden neurons for all the above optimizers.
3) Perform PCA reconstruction and compare with the Autoencoder reconstruction using test reconstruction loss (you can use inbuilt PCA).