

Question 3: (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 MLP classifier for label classification. Use a 70:30 data split. Perform the following tasks:

- (a) Load the dataset using “torchvision.datasets” library as train and test, 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 MLP class with fully connected layers with relevant activation functions.
- (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 train cross entropy loss.
- (e) Write a testing loop over the batches of the test set and calculate the test accuracy.
- (f) You can save the best performing model using “torch.save”.

Part 1:

Report the test accuracy 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) Try different weight initialisations of the neural network like Xavier, uniform and normal.
- 3) Try different learning rates and also perform exponential annealing on the learning rate.

Part 2:

Perform the following experiment using the autoencoder code written in the previous question:

- 1) Train a MLP classifier using the feature representation from under complete autoencoder trained on the train set and test the MLP classifier on the feature representation from under complete autoencoder tested on the test set.
- 2) Use the MLP classifier from Part 1 which shows the best accuracy score.
- 3) Compare both the accuracy scores.