ML assignment 3

Group 55
Members :
Vinit Raj (19CS10065)
Aryan Agarwal (19CS30005)

Building a Multi Layer Perceptron Classifier for the Statlog dataset

To run, install the dependencies mentioned in the requirments.txt file and then enter python3 questions.py in any terminal

We have used the pytorch package for building the neural network and implemented principal component analysis using numpy.

File structure:

model.py

- We declare our main neural networks(all variations of the hidden layers) in this file.

utils.py

-here we have all the utility functions that help in reading the data and interacting with the model.

questions.py

-here we run all the variations of the models and plot the results as needed

Implementation details

class dataset

- inherits from the Dataset class of pytorch, this lets us define custom dataset and data loaders for pytorch models.

class *PCA*

- uses the train set to learn the projection matrix. This projection matrix is used for dimensionality reduction of both train and test dataset.

class *MLP_0*

- defines the model with no hidden layers.

class *MLP_1*

- defines the model with one hidden layer.

class *MLP_2*

- defines the model with two hidden layers.

function *read_data*

- reads the statlog dataset and also normalizes it around the mean of the dataset.

function *fit*

- given a model and its hyperparameters and training set it trains the model.

function *test*

- given a trained model it finds its accuracy on the test dataset.

function find_*accuracy*

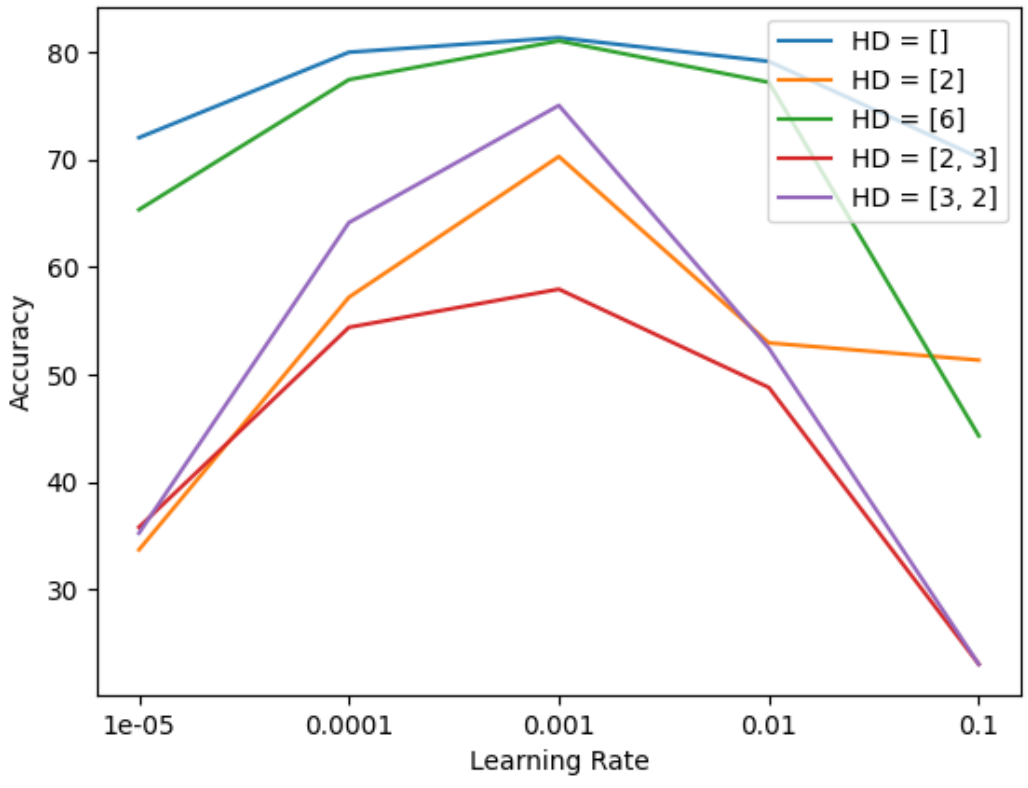
- given a model it trains it using various learning rates and then finds its test accuracy as well.

Question 1:

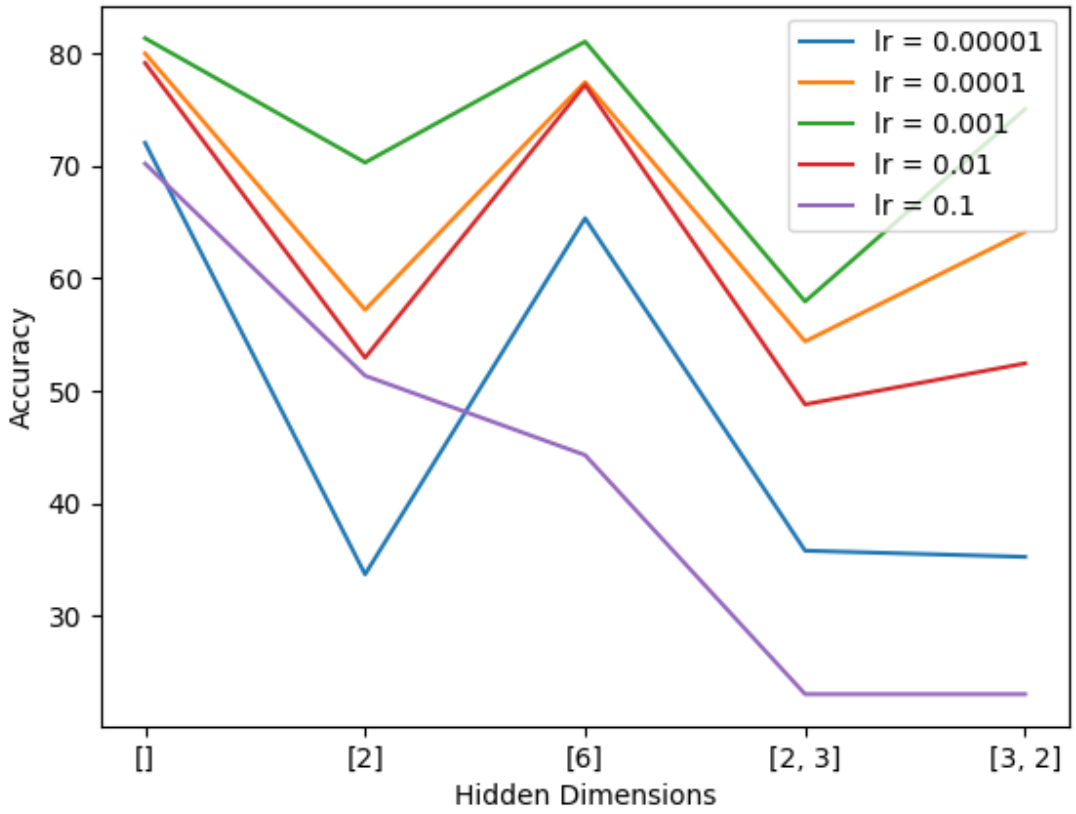
The number of nodes in the input layer will be the same as the dimension of each data sample, 36 in this case and the output layer will have 7 layers(one for each class of label).
The learning rate is varied as given in the assignment and the number of epochs is fixed to 20 for all cases to maintain uniformity.

Question 3:

> Accuracy vs Learning rate for all models, HD = [2, 3] means two hidden layers with dimensions 2 and 3 respectively.



> Accuracy versus model for each learning rate, [6] means the model with one hidden layer of dimension 6.

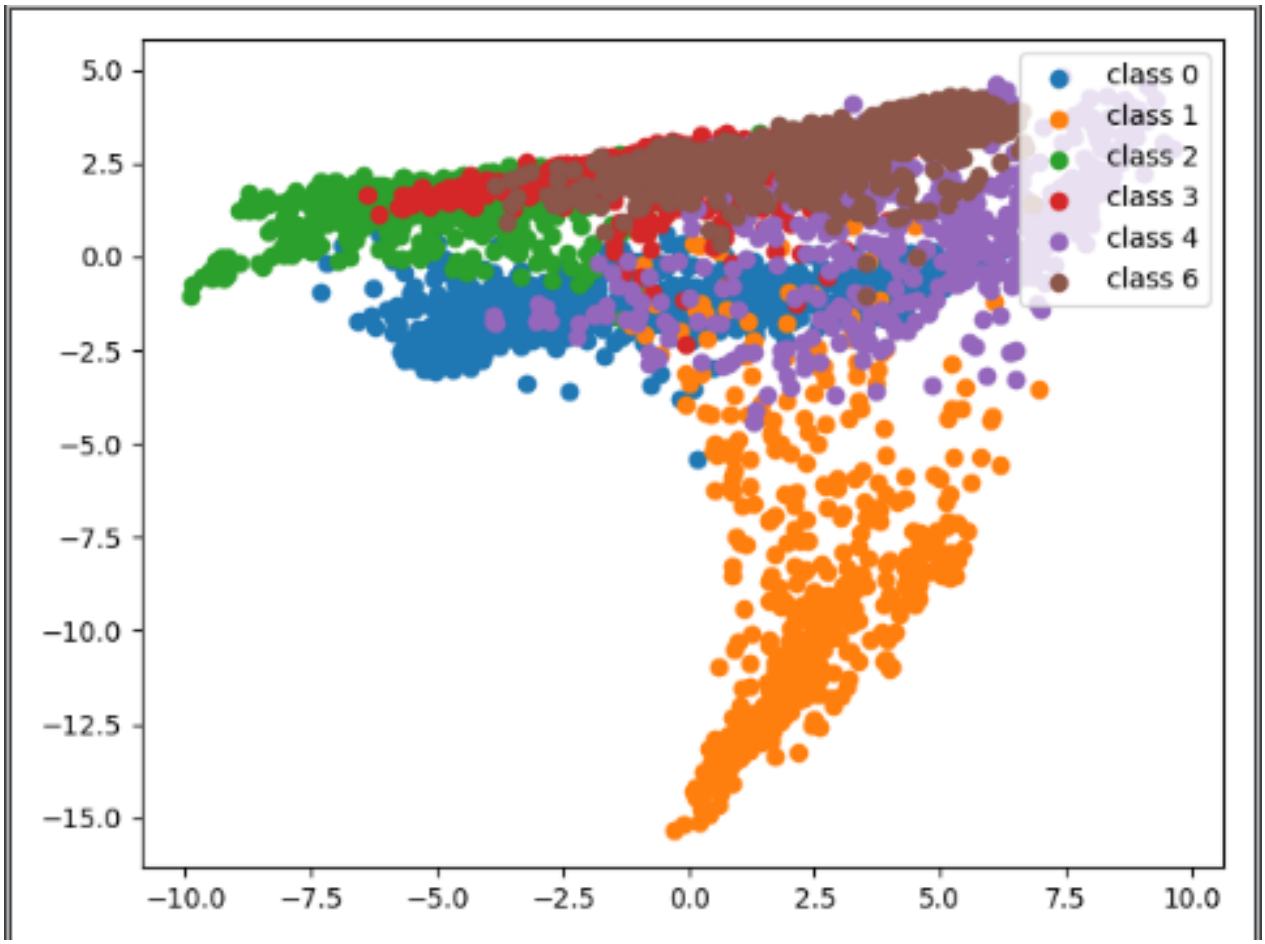


Question 4:

The best result is obtained for the model with no hidden layers when trained with a learning rate of 0.001 with resulting accuracy of 81%. The simplest model in this case provides the best result as claimed by Occam's razor, Moreover the learning rate is in the middle so small learning rate is slowing down the gradient descent whereas a large one is not allowing the loss to converge at a minima and bounce around.

Question 5:

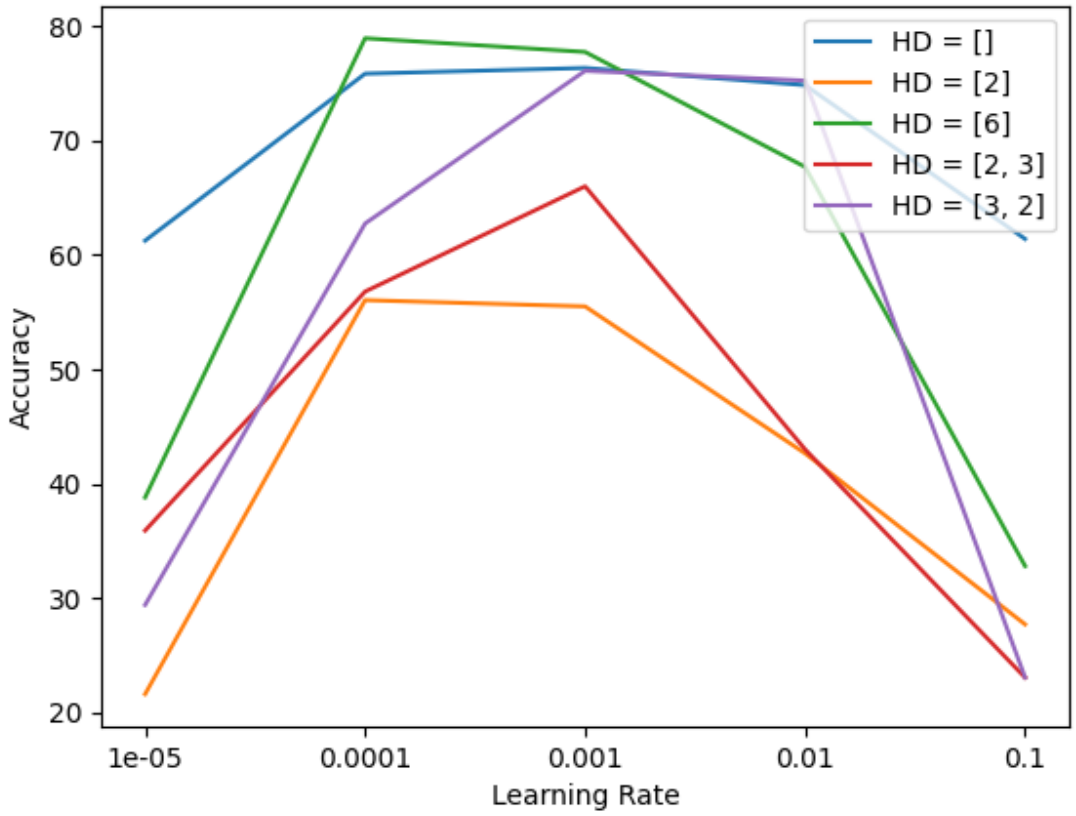
After applying PCA the training set looks like the following The dataset does not contain any sample for class 5



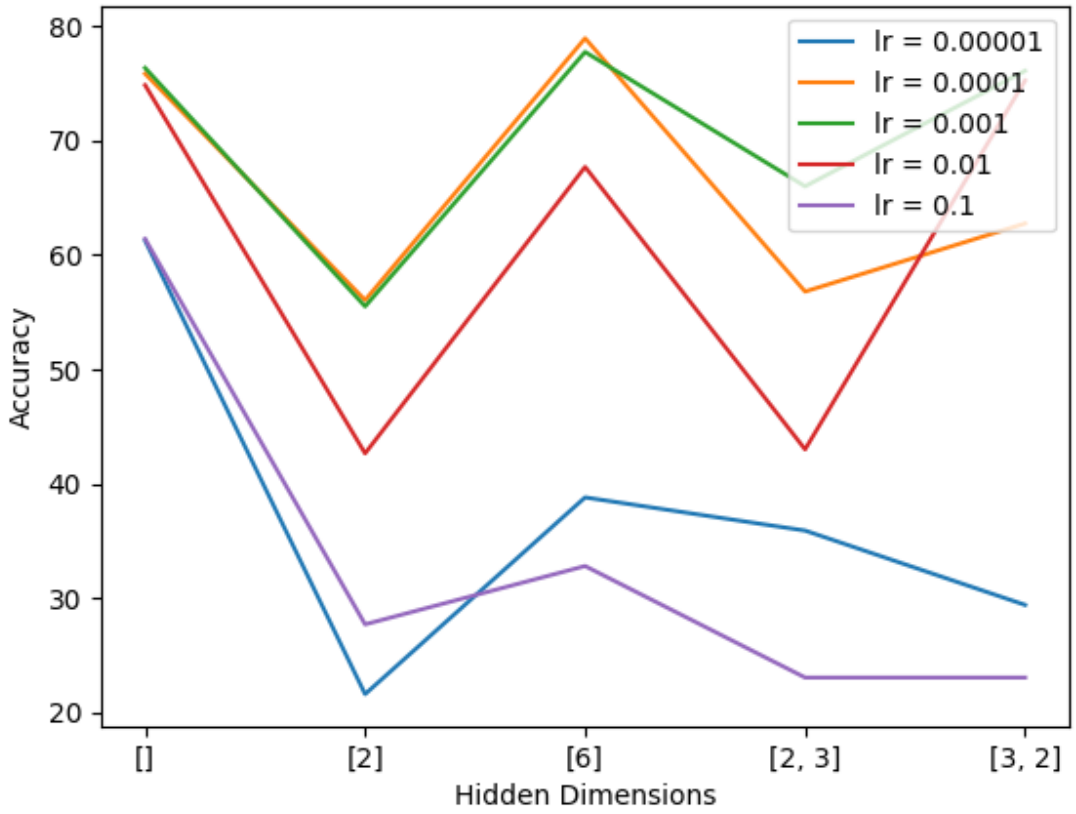
Question 6:

Running the same experiment on the reduced dimension data

> Accuracy vs Learning rate for all models



> Accuracy versus model for each learning rate



The best classification in this is obtained by model with one hidden layer of dimension 6 on learning rate = 0.0001, A more complex model performs better because the dataset has lost some information and is thus compensated by the models complexity.
The best accuracy obtained is 79%, only 2% percent lower than training with all dimensions intact. The model however runs much faster now and the slight reduction in accuracy is worth it.