Engr421: Machine learning Hw7 Report
Name: Ahmed Masry
ID: 61868

INTRODUCTION

In this assignment, I have implemented an MLP using Pytorch in Python to predict whether a bank customer will delay his credit card bill payment more than 1, 31, or 61 days given a set of information (features) about the customer. So we have 3 binary classification problems. For simplicity, I put all my implementation for one binary classification problem in the main function. Then I call it three times with the target number as an argument.

REQUIREMENTS:

- Python
- Pytorch
- Numpy
- Pandas
- Sklearn
- Matplotlib

DATA PREPROCESSING:

1) Loading the Data:
   - All the training and test data files are loaded using Pandas read_csv function and stored into Pandas frames.
2) Handling NAN values:
   - The Nan values in each column in the training data have been replaced with the mean of the remaining values in the column. In the test set, we do the same but using the means of the columns of the training data as well.
3) Dropping Nun Numeric columns:
   - From my experiments, the AUC value on the validation set is higher when I drop the Nun Numeric columns.
4) Normalizing the data:
   - In order to scale the features values in the training data, I normalized all the columns using the maximum and minimum of the columns. For the test data, I have done the same using the mins and maxs from the training data.
5) Splitting the data into Training and Validation:
   - In order to evaluate my model performance during the training, we need a validation set. So I used the Sklearn train_test_split function to split my training data into two sets: 80% train data and 20% validation set.
   - I have also noticed that the number of instances with each label are not the same in the training data. So during splitting, I made sure that the data is stratified.
6) Pytorch DataLoader:
   - Using Pytorch data.Dataset API, I implemented a class called Dataset for storing and retrieving the data easily.
   - I have also used the Pytorch Data Loader API to mini-batch of size 32 and shuffle my training data.

MODEL DESCRIPTION

In this assignment, I implemented a Multi-Layer Perceptron (MLP) using the Module API from the pytorch library. My model has 5 fully-connected hidden Linear layers each followed by a ReLU activation function in addition to the output layer which is followed by a Sigmoid function (since we are doing binary classification). The number of hidden nodes in each layer is as follows:

- Layer1: 512 hidden nodes

- Layer2: 256 hidden nodes
- Layer3: 128 hidden nodes
- Layer4: 63 hidden nodes
- Layer5: 32 hidden nodes.
- Output layer: 1 node.

TRAINING PROCEDURE:

In order to train the MLP model, I used the following

- Optimizer: Adam
- Learning rate: 0.001
- Loss function: Binary Cross Entropy.
- Epochs: 10

In each epoch, I train my model on all the batches of the training set. Then I calculate and print the following:

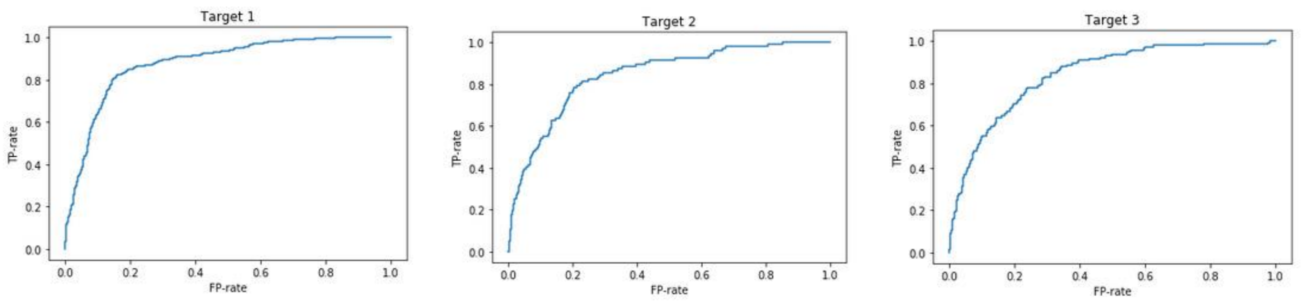
1- The average training loss in the epoch
2- Validation Set Loss.
3- AUC value. (Calculated using the Sklearn roc_auc_score function)

RESULTS:

After finishing the training, I calculated the AUC value for each target and got the following values:

- Target1 ~= 0.88
- Target2 ~= 0.84
- Target3 ~= 0.83

Moreover, I plotted the AUC curve for each target (using Matplotlib and Sklearn) as you can see in the next figure.



CONSLUSION:

As we can see from the AUC values, my MLP model does a good job in these binary classification problems.