2022 Fall EE5183 FinTech - Homework 1

Deep Learning Model: Deep Neural Network

Due: Oct 04, 2022

INSTRUCTIONS

1. Please only use PyTorch and scikit-learn to build the model.
2. You should write your codes independently. Plagiarism is strictly prohibited.
3. Report can only be written in English
4. All the figures are just examples. You do not need to be the same as the figures.
5. You must turn in hw1_student_ID.pdf and hw1_student_ID.ipynb and zip the files as hw1_student_ID.zip. TAs will grade your python code by Google Colab. Please make sure your code can be run on the Google Colab GPU environment. The wrong format will not be graded.

Note: If you are new to Colab, you can watch the videos and tutorials below.

- Get started with Google Colaboratory(youtube) https://www.youtube.com/watch?v=RLYoEyIHL6A
- PyTorch Tutorial(youtube) https://www.youtube.com/watch?v=kQeezFrNoOg
- PyTorch Official Tutorials: https://pytorch.org/tutorials/

PROBLEMS

In this homework, Dataset from Kaggle Credit Card Fraud Detection is utilized to build classification models. The datasets were already preprocessed. The features V1, V2, ..., and V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount.' Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount. Feature 'Class' is the label, and it takes value 1 in case of fraud and 0 otherwise.

1. (100%) Classification: In this exercise, you will implement a DNN model for binary classification using *creditcard_sample.csv*. This exercise aims to train ML models and a neural network to recognize fraudulent credit card transactions. *You need to split the data into training (80%) and test (20%) data.*

(i) (15%) Logistic regression (LR), support vector machine (SVM), and random forest (RF) are supervised machine learning algorithms that can be used for classification tasks. Please briefly describe the concept of these three algorithms and build three models to learn the binary classification task.

(ii) (5%) Please plot the confusion matrices for (i), as shown in Figure 1.

(iii) (15%) Precision, recall, and F1-score are ways to evaluate model performance. For each class, please calculate the corresponding Accuracy, Precision, Recall, and F1-Score on the test set in your report.
(*hint: use sklearn.metrics.classification report*). Which metric do you think is more suitable for this dataset? Please explain why.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ score\ =\ 2\ *\ \frac{Precision*\ Recall}{Precision+\ Recall}$$

(iv) (40%) Please construct a DNN model with Pytorch for binary classification according to the cross-entropy error function.

$$J(\theta) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{C} t_{mi} log S_i$$

where t_{mi} is the ith target of the mth batch, M is the batch size, $C = 2$ is the binary class for each sample, and S_i is the sigmoid activation of neural networks output function. Minimize the error function $J(\theta)$ by running the error backpropagation algorithm using the Adam Optimizer. You should decide the following hyperparameters: number of hidden layers, number of hidden units, learning rate, number of iterations, and mini-batch size. Please try to perform a grid search over the variables mentioned above and show the best-performing setting for your model in the report. You also have to show your (a) training accuracy, (b) test accuracy, (c) training loss, (d) test loss (Figure 2 is an example.), (e) confusion matrices, and (f) Accuracy, Precision, Recall, and F1-Score in the report.

(v) (10%) You have to plot the receiver operating characteristic curve (ROC, as shown in Figure 3) and precision-recall curve (PRC, as shown in Figure 3) with their area-under-curve (AUROC and AUPRC) for DNN, LR, SVM, and random forest on the test set.

(vi) (5%) Please use the metric you choose in (iii) to compare these four models (LR, SVM, RF, DNN) and briefly describe what you found.

(vii) (10%) Let's observe the number of fraud vs. non-fraud cases in the dataset. The dataset is imbalanced, with most of the transactions being non-fraud. This bias in the training dataset can influence many machine learning algorithms, leading to ignoring the minority class entirely. The two main approaches: (1) Undersampling: Randomly resampling an imbalanced dataset are to delete examples from the majority class (2) Oversampling: Duplicate examples from the minority class. First, please use the random undersampling method to deal with your training dataset and bring the non-fraud transactions to the same amount as fraud transactions (we want a 50/50 ratio). Second, use this training dataset to train the LR, RF, SVM, and DNN again. Whether the performance improves after resampling in your case?

(viii) (Bonus 10%) Follow (vii). Do you know other methods to deal with imbalanced data? (you can explain your ideas or directly implement them, the latter gets more points.)

Figure 1 : Example confusion matrix.

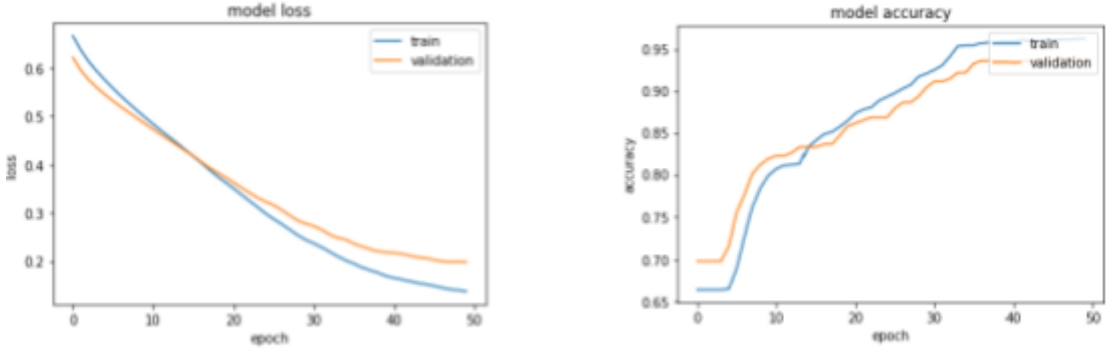

Figure 2: Example of loss and accuracy curve

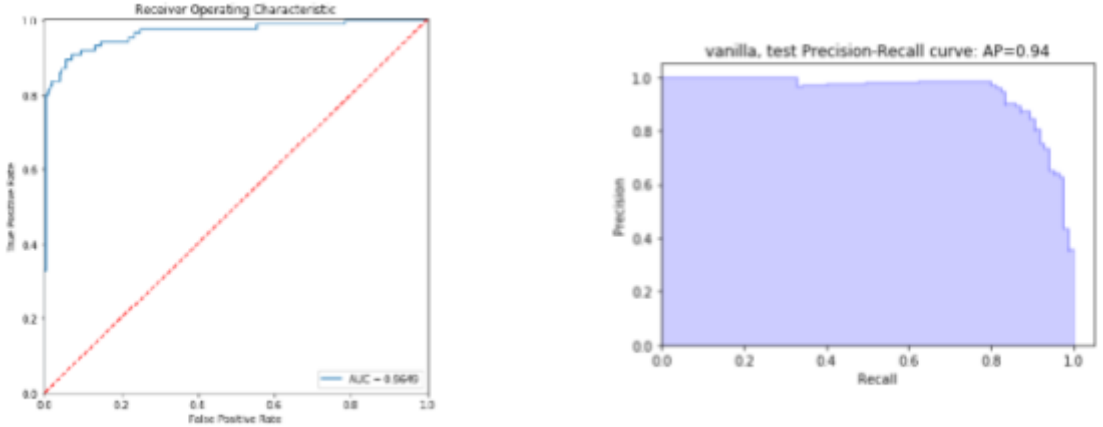

Figure 3: ROC curve and Precision-recall curve.