2019 2019 Fall EE5183 FinTech - Homework 2

Deep learning Model: Deep Neural Network

Due: Nov 7, 2019

INSTRUCTIONS

1. 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, ... 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, this feature can be used for example-dependant cost-senstive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

2. **Please use *Data.csv* to train/test your models and report classification results generated from the hidden test set as *test_no_Class.csv*.**

3. Please only use TensorFlow or Keras to build the model.

4. Name your source code that contains your *main* function as *hw2_StudentID.py* and your report as *hw2_StudentID.pdf*. You should provide your predictions for hidden test set following the format of example submissions (*StudentID_Class.txt*).

5. **You should write your own codes independently. Plagiarism is strictly prohibitted.**

PROBLEMS

1. (90%) Classification:

   In this exercise, you will implement a DNN model for binary classification using *Data.csv*. The objective in this exercise is to create and train a neural network to recognize fraudulent credit card transactions. You need to split the data into training (80%) and validation (20%) data.

   (i) (40%) Please construct a DNN for binary classification according to the cross-entropy error function

   $$E(w) = -\frac{1}{m}\sum_{m=1}^{M}\sum_{i=1}^{C=2} t_{mi} log S_i,$$

   where t_{mi} is the ith target of the mth batch, M is the batch size, C is the classes for each sample, S_i is softmax activation of neural nets output function. Minimize the error function *E(w)* by running the error backpropagation algorithm using the Adam Optimizer.You should decide the following variables: number of hidden layers, number of hidden units, learning rate, number of iterations and mini-batch size. **Please try to perform grid search over your 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) testing accuracy, (c) training loss and (d) testing loss in the report. An example is detailed in Figure 1.**

   (ii) (10%) **please plot confusion matrices** for (i) as example in Figure 2.

   (iii) (10%) Precision, recall, F1-score are other ways to evaluate model performance. For each class, **please record precision, recall and F1-score as well as the averages of those criteria** over all classes in your report.

   $$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

   $$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

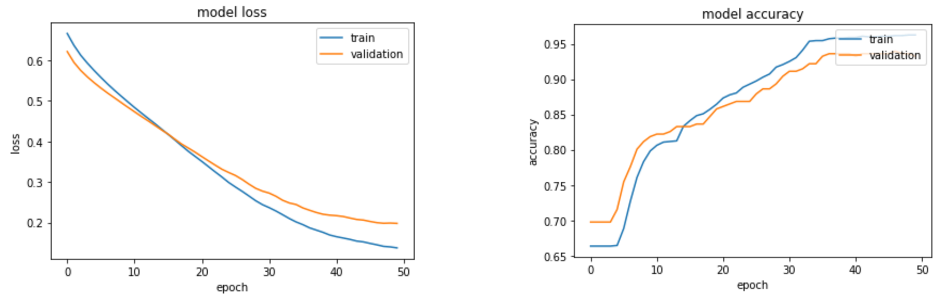

Figure 1: Example of loss and accuracy curve.

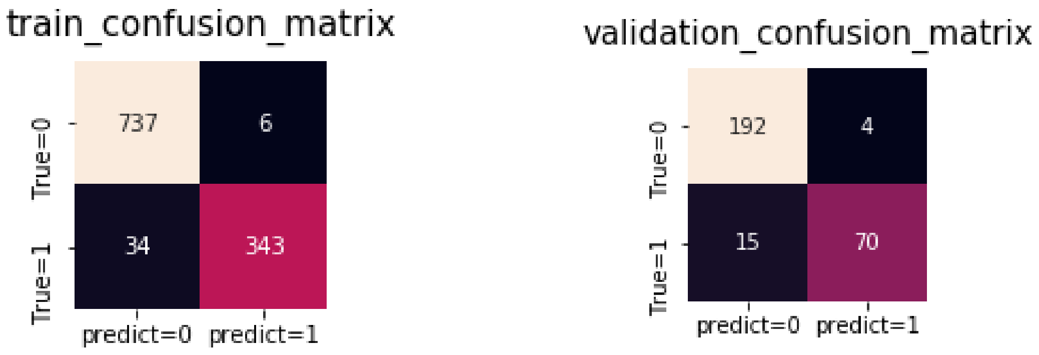

Figure 2: Example confusion matrix.

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

(iv) (10%) **What is difference between decision tree and random forest?**

(v) (10%) **Please use decision tree and random forest to calculate Accuracy, Precision, Recall and F1-Score**.

(vi) (10%) You have to plot learning curve, 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).

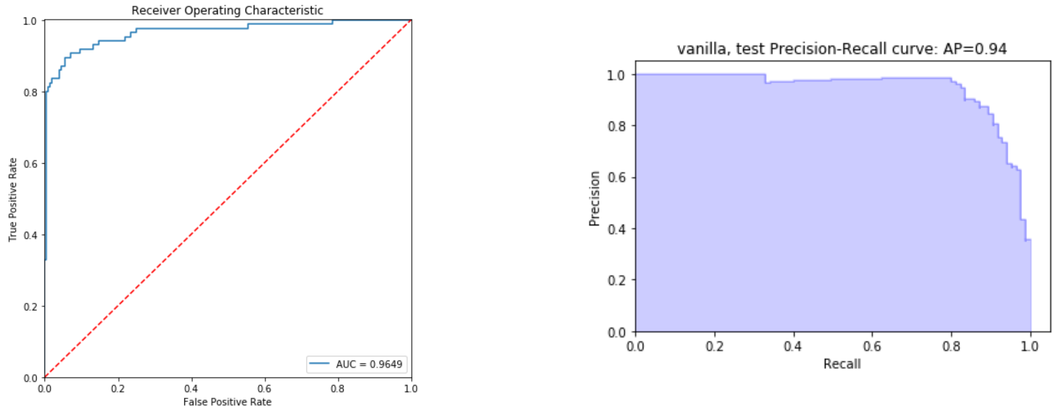

Figure 3: ROC curve and Precision-recall curve.

2. (10%) Hidden Test Set :

   (i) (10%) Apply the model to *test_no_Class.csv* and save your classify results as *StudentID_answer.txt*.