Financial Technology: Assignment 2

Nov. 2020

B07703014 財金三 蔡承翰

[Problem 1]

(*i*) Following the instruction, we have constructed a DNN model using *PyTorch* package. We set up a reasonable search space for each of the criteria requiring grid search. The search spaces are stored in corresponding lists. The spaces are shown in the table below:

CRITERIA	SEARCH SPACE
# HIDDEN LAYERS	[3, 4, 5]
# HIDDEN UNITS	[16, 32, 64]
LEARNING RATE	[0.01, 0.001]
# ITERATIONS	[1, 25, 50]
MINI-BATCH SIZE	[128, 256, 512]

Notice that in our model, all neurons are fully connected from layer to layer, and the activated functions are all set to ReLu. To construct a binary classifier, the output layer is composed of only one neuron and a sigmoid function is applied before the output. To compare among candidate models, we simply choose the model with the highest accuracy on the validation set as the "best performing model" is asked for the assignment. Grid search has provided us with the following result:

CRITERIA	OPTIMAL VALUE
# HIDDEN LAYERS	4
# HIDDEN UNITS	16
LEARNING RATE	0.001
# ITERATIONS	50
MINI-BATCH SIZE	128

The optimal model is capable of giving us training accuracy of 0.987, testing accuracy of 0.964, training loss of 0.393 and testing loss of 0.336. Before we introduced the loss curve and the accuracy curve, it is worth-mentioning that our method to pick up the optimal model is of controversy. The main issue is that, though we can filter the model that is more fitted, it is of high probability that the model is overfitted. If that is the case, the loss of the training set and the validation set might not match together. Solely determined by testing accuracy of final result means that the "optimal model" we have chosen may achieve high accuracy by chance in the final epoch. The interpretability of our model may be weak per se. Now we introduce the loss curve and the accuracy curve.

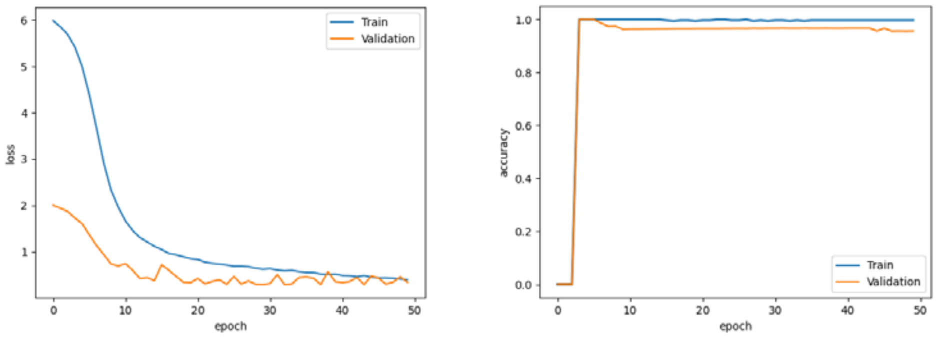

Figure 1: loss curve (left) and accuracy curve (right)

It turns out that our model performs well, the loss and accuracy on validation set follow the patterns of training set. Through these diagram, though the overfitting issue exists, but it is reasonable for us to believe that the current model is good enough.

(*ii*) To plot the following matrices and diagrams, we import *sklearn* package. The confusion matrices are shown as follows:

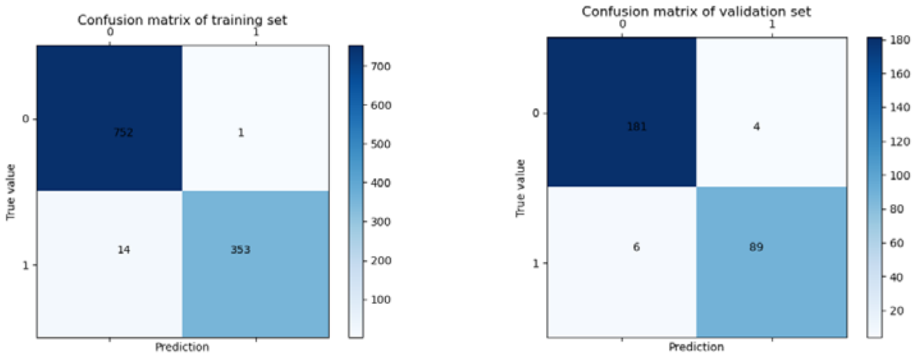

Figure 2: confusion matrices

(*iii*) We directly give the result. For training set,

CLASS: TRAIN	VALUE
PRECISION	0.997
RECALL	0.962
F-1 SCORE	0.979

For validation set,

CLASS: VALIDATION	VALUE
PRECISION	0.957
RECALL	0.937
F-1 SCORE	0.947

And the average values of these indexes are

CLASS: AVERAGE	VALUE
PRECISION	0.989
RECALL	0.957
F-1 SCORE	0.972

(*iv*) The main difference between decision tree and random forest is that random forest is in fact a combination of multiple decision trees (that's why it is called "forest"). A decision tree contains only a single decision sequence and a single subset of data (or the full data) to do prediction, while a random forest model predicts the target from outputs of several different trees, all with different structures (subset of data, decision sequence, etc.). In general, given a large dataset, when the interpretability is not our major concern, random forest models should outperform decision tree models.

(*v*) Fitting the decision tree and random forest models has given us the following result (the accuracies are evaluated by the validation set):

MODEL: DECISION TREE	VALUE
ACCURACY	0.986
PRECISION	0.979
RECALL	0.979
F-1 SCORE	0.979

As for random forest,

MODEL: DECISION TREE	VALUE
ACCURACY	0.993

PRECISION	0.979
RECALL	1.0
F-1 SCORE	0.99

(*vi*) We present the diagrams directly.

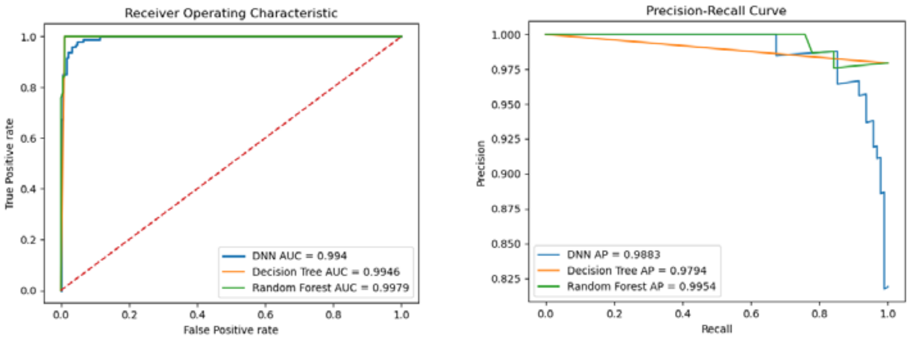

Figure 3: ROC (left) and PRC (right)

[Problem 2]

Now we present the lift curve.

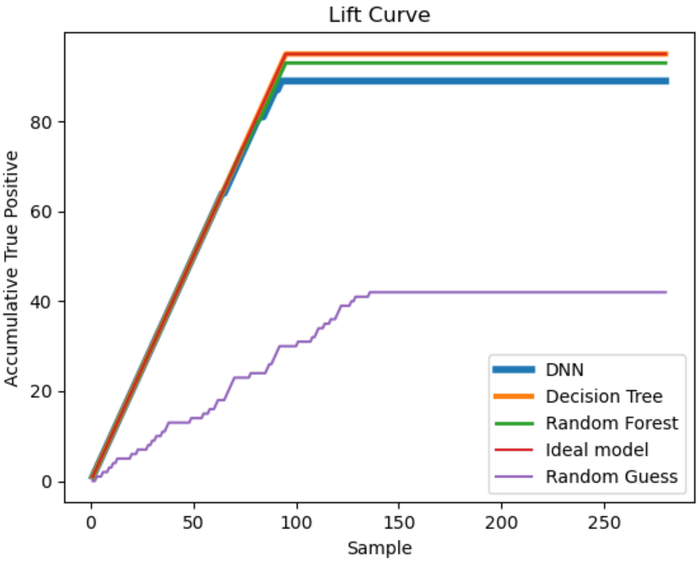

Figure 4: Lift curve