CS583A: Course Project

Zhengyan Zhuo, Guanghua Zha

December 1, 2019

1 Summary

We participate an inactive competition with late submission of classifying computers that might be infected by malware. Each of the member build their own model using keras and other related libraries. Guanghua implemented a multi-layer neural network with skip connections with learned from Resnet and run the code on a laptop with i7 CPU, 16 GB memory and Nvidia 1660ti GPU. Zhengyan utilized lightgbm model to deal with this problem, explore data, do feature engineering then train the model. Code was ran on Google Colab with TPU. Performance is evaluated on the classification accuracy. In the public leaderboard, our best score is 0.69259; we rank 919 among the 2412 teams. In the private leaderboard, our score is 0.63415 from; we rank 1050 among the 2412 teams.

2 Problem Description

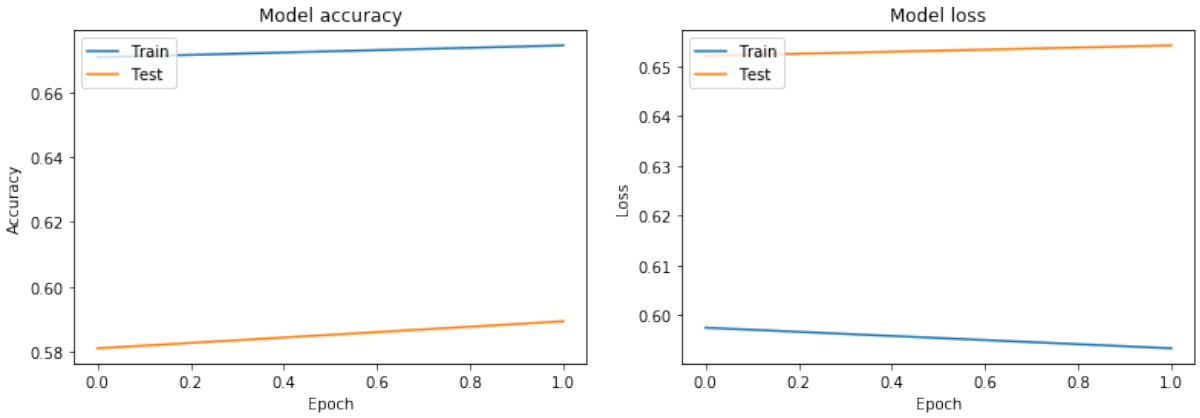
Problem. The problem is to classify computers that is affected by malware. This is a binary classification problem. The competition is at `https://www.kaggle.com/c/microsoft-malware-prediction`.

Data. The data contains 8,921,482 computer information including their manufacture, OS edition, firmware information and much more. Since not all of the information are related to our classification, we selected several columns from the original data to be considered as training set.

Challenges. The first challenges we have encountered is the data set is too big and we had many memory error when processing the data. We have to reduce the number of columns of information to make sure there is enough memory for training. Another challenge is we do not know how deep should our model be to avoid under-fitting since this data set is quite large, we have to try many times with different model structure to see how well the model performance.

3 Solution

Model. Guanghua build a multi-layer neural network with skip connection which learned form Resnet structure. He also tried the classic fully connected neural network but found out it has worse performance compare to the current model. Zhengyan utilized lightgbm model to implement the prediction algorithm, the most challenging part is feature engineering. After figuring out the model is likely to overfit, he improved the model performance by making categorical features sparse.



(a) The classification accuracy on the training set and validation set.

(b) The loss on the training set and validation set.

Figure 1: The convergence curves.

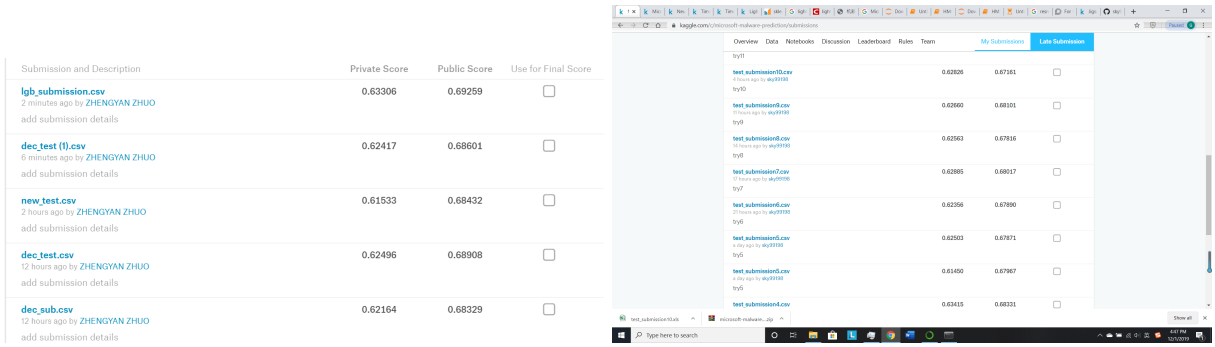
Implementation. Guanghua implement the model using Keras with TensorFlow as the backend. Zhengyan utilize pandas package to do data manipulation and lightgbm/xgboost package to train the model. Our code is available at `https://github.com/sky99190/CS583Final`. Guanghua run the code on a lenovo laptop with one Intel i7 CPU, 16 GB memory and NVIDIA 1660ti GPU. It takes 1 hours to train the model and 1 hour to generate the submission. Zhengyan using the kernel provided by the Google Colab to run his code.

Settings. For Guanghua The loss function is categorical cross-entropy. The opitmizer is adam. validation split is 0.1. And shuffle data set to True. For Zhengyan he is using the function LGBMClassifier in LightGBM to train his model with learning rate equals 0.05 and number of leaves equals 2**12-1.

Cross-validation. Guanghua tune the parameters using validation split set to 0.1.Zhengyan using a 5 fold validation method to tune the parameter.

4 Compared Methods

Method comparison between team members. Since we have two team member, each of us do their own work on this project. Guanghua uses a traditional way to build a neural network with Keras and tensorflow, Zhengyan allied advanced library LightGBM to achive his approch. Zhengyan's method is a gradient boost framewrk with optimized performance. The outcome is showing Guanghua has a little better performance on private leader board while Zhengyan's method has better performance on public leader board. Zhengyan using a folding method to tune the parameter of the model while Guanghua tried different model structure. However, both of us encounter the boundary of our method which happened when we reach the accuracy near 0.68. No matter what we tried, the results are not going better and sometimes even go worse. Both of us



(a) Zhengyan result.

(b) Guanghua result.

Figure 2: Figure 2 Result.

having problem of running out of memory so we have to shrunk our training data set to avoid such problem.

5 Outcome

We participated in an inactive competition. Our score is 0.69259 in public leaderboard and 0.63415 in the private leaderboard. We rank 919/2412 in the public leaderboard and 1050/2412 in the private leaderboard. The screenshots are in Figure 2.