Individual Final Report

Zhilin Wang
netid: zhw
1st, May 2020

1. Introduction

The project is about classifying whether a planet is an exoplanet or not based on the light intensity we observed from earth by time. Our team has three members. Zixuan, Renee and I collaborated on the data preprocessing part and did three models based on the preprocessed data. And then compare the result our models. I did the CNN model.

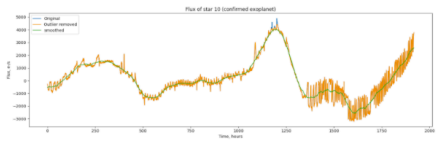
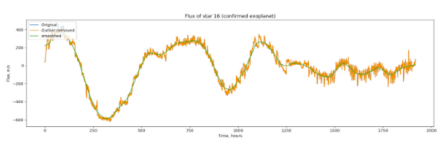
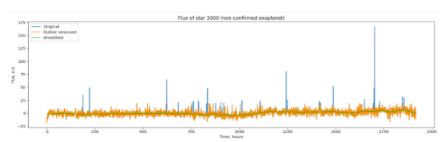
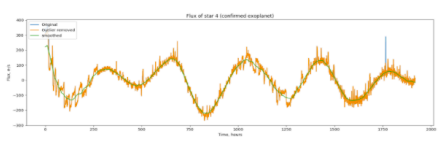
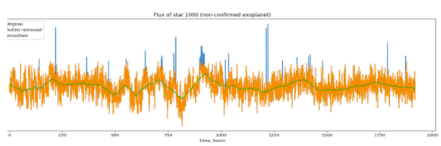
2. Description of my work

I use Keras as the framework. And the CNN model is based on model sequential. A convolution network is a multilayer feedforward network that has two- or three-dimensional inputs. The principal layer type for convolution networks is the convolution layer. A pooling (or subsampling) layer often follows a convolution layer and consolidates r × c elements in the input image to 1 element in the output image. I basically used max pooling for the pooling layer.

$$z_{i,j} = max\left\{v_{r(i-1)+k,c(j-1)+l} | k = 1, ..., r; l = 1, ..., c\right\}$$

3. The portion of the work I did

As how the work is shared, I did part of the data preprocessing. I did train, test, validation splitting, smoothing data(which turns out it is not necessary) and data visualization. I plot the

Exoplanet	Non-Exoplanet
	
	
	

data out to see how our outlier removing works. I did the research on the method of detecting exoplanet. And I did the CNN model tuning that is elaborated below.

4. Results

The final CNN model I choose has 5 convolutional layers with channels from 8 to 128 and with max pooling layers and batch normalization layers connected each layer, and the kernel sizes are all 11. After model flatten, I got 2 dense layers both with 64 neuron(s) and dropout 0.5 and 0.25 in case overfitting. The activation function for the layers above are all â€˜reluâ€™; and the final layer takes sigmoid activation function.

I tested different convolutional layer amounts, kernel size, output amount, pooling method, dense layer amount, number of neurons in a dense layer. Some of the results are listed below.

	Best Model (training all 0 samples)	**Kernel size = 13 for the first two convolutional layers**	**Using AvgPooling before flatten**	**2 more convolutional layers and more neurons in dense layers**
F1-score	**0.8**	**0.75**	**0.5**	**0.29**

At first, I used “Adam” as the optimizer and the validation loss converges quickly but it fluctuates then. So I applied the early stopping and model stops at 18 epochs with a bad f1-score. Then I let early stopping have patience for the epochs and added model checkpoint as callbacks to save the best model. F1-score improves but not that big. Then I tried “SGD” as the optimizer with more epochs(200) and the loss curve becomes smooth after it converges. And actually after around epoch 40, the loss becomes almost flat. So I choose the number of epochs to be 50.Â

Since f1-score is an important reference of the goodness of the model, and in order to balance recall and precision, I tried to change the threshold that classifies the model output to zero or one. Sometimes, lowering the threshold helps improve the recall of one. But then I realized the test set is so imbalanced that it contains only five ones. And the improvements brought by the threshold could be luck. Then the threshold was reset to be 0.5.

Each observation of a planet is a list of flux intensity by time. We wonder if measuring the overall trend and fluctuation of the light intensity is more efficient. I tried to remove the noise by smoothing the data like we usually do in time series problems. However, both precision and recall for positive samples become 0. I assumed that the data is "over-smoothed", then I tried increasing the weights of the original data. It neither works. Actually, we are using a time-series dataset but not solving a time-series problem; we are not forecasting or predicting the future light intensity. So considering the oscillation as features, we do not erase any noises; instead, we keep all the changes of light intensity by time.

Preprocessing data plays a significant role in the performance of the model; especially for sampling in dealing with imbalanced data. We rotate the flux to generate dummy exoplanet data, the number of the fake data generated matters. Since the gap of the amount of two categories is huge, it is not optimal to generate a lot of dummy exoplanets to fill the gap. Then a simple undersampling is helpful in balancing the data. In general, we drop similar data in undersampling; that is finding the nearest neighbors and drop a few. However, for each planet, there are over 3000 dimensions and it could be time-consuming to find the similarity. So I just take the first 3500 observations from the dataset knowing that exoplanets(1s) are all arranged in the front rows. And it actually improves the model by balancing the data. Below are some of the approaches.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	565
1	0.75	0.60	0.67	5
accuracy			0.99	570
macro avg	0.87	0.80	0.83	570
weighted avg	0.99	0.99	0.99	570

Model 1: *200 rotations (200*29 more 1's) | all non exoplanets (0's)*

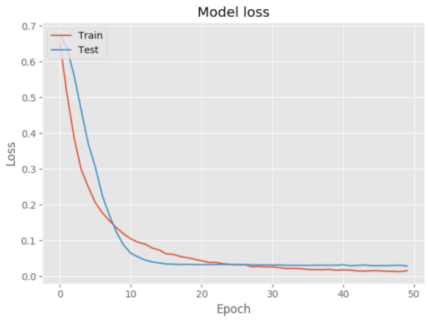
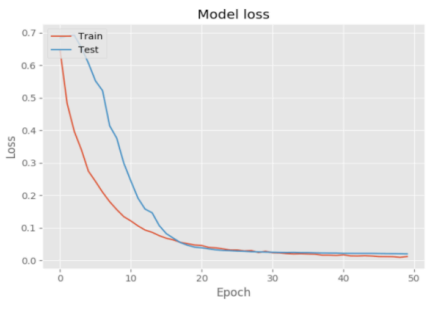
	precision	recall	f1-score	support
0	1.00	1.00	1.00	565
1	0.80	0.80	0.80	5
accuracy			1.00	570
macro avg	0.90	0.90	0.90	570
weighted avg	1.00	1.00	1.00	570

Model 2: *100 rotations (100*29 more 1's) | all non exoplanets (0's)*

	precision	recall	f1-score	support
0	1.00	1.00	1.00	565
1	0.83	1.00	0.91	5
accuracy			1.00	570
macro avg	0.92	1.00	0.95	570
weighted avg	1.00	1.00	1.00	570

Model 3: *100 rotations (100*29 more 1's) | 3521 (1587 less) exoplanets (0's)*

Moreover, f1-score is not our unique consideration on model evaluation. Even though the validation loss of *Model 3* converges slower, it reaches the bottom level as *Model 2* does. With the obvious difference in f1-score and similar loss, we finally take *Model3*.

Model 2 Loss Plot	**Model 3 Loss Plot**
	

5. Summary

I have learned that imbalanced data may bring huge problem to the model. There are multiple ways to handle imbalanced data like resampling. And the evaluation metrics for the model should not be limited to accuracy, especially for imbalanced data. We have to look at the precision, recall, loss and more.

For the future improvements, I would try ROC curve and compare the results of our three values. And grabbing more samples from NASA to add more variables in the training set and test set will improve the model.

6. Percentage of the outside code

21%

7. References

Muonneutrino. (2017, April 04). Exoplanet Data Visualization and Exploration. Retrieved May 01, 2020, from https://www.kaggle.com/muonneutrino/exoplanet-data-visualization-and-exploration

Toregil. (2017, June 27). Mystery Planet (99.8% CNN). Retrieved May 01, 2020, from https://www.kaggle.com/toregil/mystery-planet-99-8-cnn