Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 7
TPCEE 2024

Development of a neural network based on pyTorch for the identification of planetary candidates among TESS mission targets

Zichen Huang*

Suzhou North America High School, Suzhou, China

Abstract. This paper Since NASA launched the Transiting Exoplanet Survey Satellite (TESS) in 2018, many exoplanets beyond the solar system have been discovered. The TESS mission has collected a magnificent amount of photometric data for scientists to analyze and its instrument observes more than 200,000 target stars in its mission. Scientists identify and analyze the patterns of light curve data for each target and determine if it is a planetary candidate or false positive due to an eclipsing binary star system or instrumental noise. We present a tool to automatically analyze the data of target stars through machine learning and a neural network built with pyTorch. When the Tess satellite finds stars that may have transits, the observation data can be downloaded through computer programs or archived. This will be a simple and time-saving tool that allows people to distinguish real transit planets from eclipsing binaries and observation equipment noise among the targets. The neural network will use the timing data of planet luminosity as input to output the probability that the galaxy really contains exoplanets. By training and testing the neural network, we find that the recall rate and accuracy rate of the network are a and B respectively. Moreover, in the case of X percent, the probability of recognition by this neural network is greater than that of non-planets.

Keywords: Exoplanet detection, Transit, Neural network, Deep learning, Data analysis.

1. Introduction

Since the first exoplanet was discovered in 1995, there has been broad participation in further searches: both ground and space-based telescopes, including WasP, Gaia and the Kepler K2 mission. In 2009, NASA launched the Kepler telescope in search of the density of exoplanet in star systems close to the sun. The Kepler mission brought a significant number of exoplanet discoveries. After its two reaction wheels got impaired, NASA launched the Transiting Exoplanet Surveying satellite for a designated mission to search for exoplanet, typically earth-liked exoplanets. Equipped with photometric observers, TESS satellite observes more than 200,000 target stars during its survey so that astronomers are able to examine the data product and dis- cover the potential candidates for new exoplanets. For the target stars that may have transit, this is called a threshold crossing event. These planets will be generated a timing data by Tess, and the brightness of this star will be recorded at each time instant. From the beginning of Tess mission in 2018 to now, people have found more than 2000 planets and a total of 5000 possible planets. From the beginning of Tess mission in 2018 to the present, people have found more than 2000 planets and a total of 5000 possible planet candidates. These data, called lightcurve, are analyzed manually. At this time, a problem in the research of exoplanet exploration has emerged. With the accumulation of fits images and lightcurve data from satellite optical instruments, we must find a way to quickly process these massive data in order to find exoplanets more efficiently or do exoplanet density re- search. All the data of Tess are from stars with threshold crossing event (TCE), which represents the photometric changes caused by the transit of suspected exoplanets. People usually use Python and other programs to visualize these photometric curves before analysis, discussion and subsequent confirmation (data collection of ground telescopes or shooting in other periods). All luminosity curves caused by transits have visual rules to follow. Since the periodic transits of planets will block a certain amount of starlight, there should be periodic downtime on the luminosity curve. However, most TCES are false positive events, either affected by other light sources, or mechanical noise, etc. Our research is to use the lightcurve data product generated by Tess task as input, and use the neural network based on pytorch to get its classification

between false positive and planar candidate. The training will use the existing artificially confirmed planets and false positive data.

2. Deep Learning Model

Machine learning is a science of artificial intelligence. It automatically induces logic or rules from data by selecting appropriate algorithms, and predicts according to the inductive results (models) and new data. In most labeled machine learning models, the computer is given features and outputs to get an inter- mediate hidden function and to use this hidden function to get predictions from new inputs. Deep learning is a representation learning, which means that when we feed data to a machine, the machine will learn the input characteristics on its own. The deep neural network is the model we use, and it is also a very widely used model in modern data science. It can analyze varied types of data, such as image classification. Feedforward neural network, which can also be a fully connected network, is the simplest one in which each neuron is arranged in layers, and each neuron is connected only to the neuron in the previous layer. Receives the output from the previous layer and outputs it to the next layer without feedback between the layers. It is one of the most widely used and rapidly developing artificial neural networks. [Image A] is an image description of a fully connected neural network, or, a feed-forward neural network can be described from a matrix calculation. We can convert each layer into a matrix. That is [Result Vector = Activation Function* (Weight Vector of this layer * Connected to the node of the previous layer + bias); where each link in the previous image is a weight calculation.

In a neural network, each neuron learns a more abstract representation of the values of the previous neuron. For example, the first hidden layer learns the features of "edge". The second hid- den layer learns the features of "shape" consisting of "edge". The third hidden layer learns the features of "pattern" consisting of "shape". The last hidden layer learns the features of "target" consisting of "pattern". However, in pictures or timing data, the central feature of the picture body or data may appear on adjacent but different pixels or time points. The fully connected neural network will lead to too many parameters, low efficiency and difficult training. At the same time, a large number of parameters will soon lead to over fitting of the network. In convolution neural network, convolution layer can quickly train some features to avoid unnecessary over training of offset. Convolution neurons in convolution layer are not fully connected with the nodes of the previous layer, but partially connected.

Therefore, our neural network adopts the method of combining full connection layer and convolution layer. Generally, we use gradient descent to train neural networks. In classification problems, the cross-entropy loss function is the most widely used. This loss function is a loss function that can ensure accuracy and low derivative calculation; Cross entropy loss function:

$$L = \frac{-1}{N}\sum_i \sum_{g=1}^{M} y_{ic} log p_{ic} \tag{1}$$

The loss function of classification problem can quantify the deviation between machine classification and actual classification. Training neural network is to minimize the loss function. Gradient descent makes the parameters change according to the gradient by calculating the gradient of the loss function on the parameters. By determining the learning rate, the mathematical process of training is:

$$W_{new} = W_{old} - \phi \times learningrate \tag{2}$$

After repeating this process many times, we get the trained neural network.

3. Neural network Training

3.1. Dataset

For the training of the neural network, a 'human labeled' data set will be present. While these data consist of human vetted dispositions of possible transit event light curves, including more than 5000 of known planets that have been "validated" according to the TESS transit search standard and validation TESS follow up program. All of the TCE's in the training set were already closely examined by human scientists and analysts to determine the categorical disposition the target star. The labels of the data set used in the neural network is decided by the results of human vetting discussion. To maintain the differentiability and unity of labels, we assign each training data or a prediction result in to one of these labels.

From the training set, all the target TCE that have been con- firmed like aforementioned, will be labeled as "True Planet" and "Planetary candidate", which is also a call sign for the TESS follow up observation program (TESSFOP) for further confirmation on the vetting result.

The 'stellar variability' will be belonging to the target TCEs that indicated relatively large difference in odd/even transits or traces secondary eclipse. Nevertheless, it also presents clear transits and raise high BLS max-power ratio. This is likely to be caused by either eclipsing binaries or stellar variability. Clearly labeling of this category is very important since the false positive belonging to 'stellar variability' is hard to distinguish from real planetary candidates. Hence, some of the vague results that S bn one has truly determined whether it is a planet will not be included in to the data set.

For false positives caused by aperture problems and obvious instrumental noises will be categorized in to "False Positive". To prevent overfitting and put it closer to reality, the number of "False Positive" targets will be 2 times more than the "PC" targets in the training set.

Data that cannot effectively eliminate noise and interference and damaged targets will be excluded from the data set.

3.2. Lightcurve imput

In order to obtain these data, we will use Python's crawler function and the module of Tess task – 'lightkurve' to obtain lightcurves objects After that, we will transform lightcurve. First, we will collapse it. The box least square function in the lightcurve module is an analytical technique that approximates fast Fourier transform. It will come to the conclusion that this data is most likely to represent the period of stars, and the folding in front of the neural network will follow this period. After folding, in order to ensure the unity of the input structure and dimension of the neural network, all data points will be combined into the data of 1024 time points through matrix transformation (the input of convolutional neural network can be divided by 2 for many times) Referring to the use of another neural network, in order to maintain the integrity of the possible transit signal, another folded lightcurve will also be used as input, and will be introduced into different, separated convolution layers and pooling layers, but will still be connected by the full connection layer after feature learning. This will still be folded first according to the cycle, but it will be cut according to the transit time and finally averaged into 512 data points. For each lightcurve that has been folded, it will be divided into 1024 or 512 areas. In each area, there will be certain data points in it, and each data point that falls in a certain area will be averaged and used as the new data of that area, and finally a lightcurve with a length of 1024 or 512 data points will be generated.

4. Architectural testing

We tested different machine learning models that have the same input, the first is a multi-variable regression model that do not make the assumption that the data can be separated by a linear function. The second is a feed-forward neural network and we also have a architecture that combines fully

connected layers and convolutional layers. The model that gives the highest accracy and f1 score of an average of 5 runs, the model that performed that best is as shown in Fig.1.

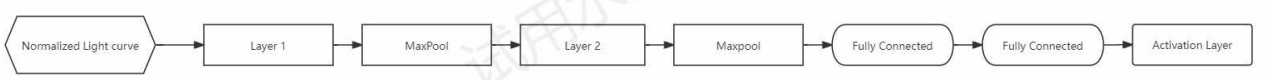

Fig.1 The tradeoff between precision and recall

5. Model Training and Performance Analysis

In our implementation of model training, we used GPU as the processing unit to calculate the parameters on different nodes, GPU is a much more efficient processor than CPU, its processing speed is roughly 50times greater than that of the latter. During the training, we used the Adam's optimization method and cross entropy loss as the loss function. For our neural network, we used the ReLU function as the activation function and sigmoid function before the output. The best performing model and optimal training parameters are passed to model testing stage. As afore- mentioned in the section in which we choose the optimal neural network architecture, we present some important criteria's that evaluates a classification model. Accuracy: One of the most intuitive aspects of a model, it equals to the fraction of data that is correctly classified. Precision: This measures the precent of real planet in what the model classifies as a planet. Recall: This reveals the fraction of true planet that has been classified as planet. In the situation of TESS data, precision and recall, or sometimes AUC score or f1 score could be a better evaluation for the model then accuracy, as it refers to what is said before, that the dataset is not statically balanced and the number of false positives far exceeds the one of true planet or planetary candidate We usually think that 0.5 is a threshold for classification for an output that was activated by the sigmoid function as it gives a result over the range from 0 to 1. However, we can tune this threshold and find a tradeoff between precision and recall.

As 20 percent of the dataset is not involved in the training process, the testing stage is going to be involved with these parts of data because they are not fitted by the functions of the neural network. This testing will get the result of the aforementioned criteria's and therefore evaluate the model on the newly generated data of TESS. Among the data generated by Tess, most of the data are false positive signals rather than transiting exoplanet, which makes the data set unbalanced, as mentioned earlier. This imbalance affects our evaluation of the model to a certain ex- tent. If the model recognizes all input signals as false positive, the accuracy will still be very high. The recall - precision trade off graph we generated will be a better evaluation method for this model. It can be seen from the figure that if we pursue higher precision, recall will be sacrificed. We can select the desired trade off by adjusting the threshold. For example, we provide a method to lower the threshold of sigmoid activation function to obtain a higher recall value. We can see from the Fig.2 that if the recall is lower than 70percent, there will be 100 percent precision, and when the precision is lower than 80 percent, all the planets in the test data set will be classified as planets. In the training, it is also possible to supplement the uneven data set, but this process requires more computation. In addition to recall precision, we also obtained better accuracy-94 percent and AUC score of 97 through the test. Between a large number of false positives and a relatively high recall precision trade off value, a specific thresh- old can be used. This model can greatly reduce the workload of manual curve fitting. Ideally, keep the recall value of the model close to the threshold of 100 percent colleagues' maximum precision, and 0.496 of sigmoid output. The high precision and the relatively small number of planetary candidates reduce the time required for manual analysis of these data classified as planets. When using this threshold, scientists only need to analyze the data classified as planets. In our test, this part only accounts for less than 10 percent of the test set on average, and only 30 percent of them are false positive.

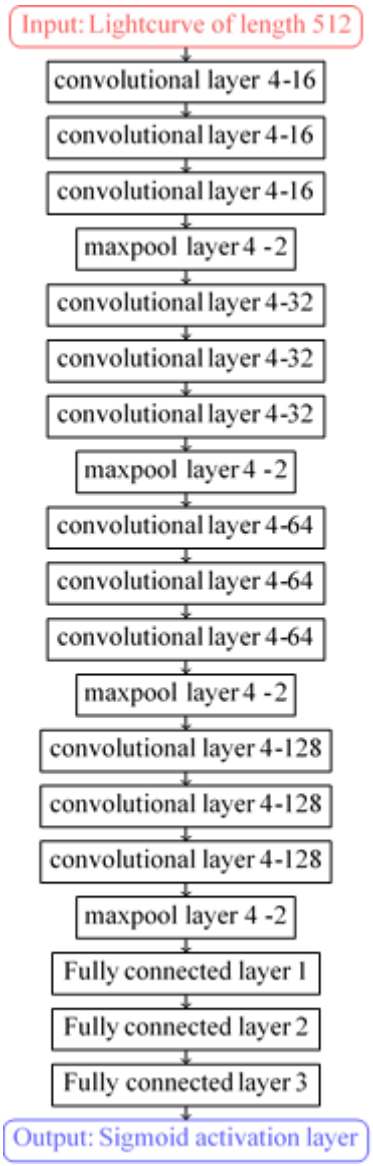

Fig.2 The structure of the neural network

Acknowledgements

Part of this work was supported by the German Deutsche Forschungsgemeinschaft, DFG project number Ts 17/2–1.

References

[1] Ansdell M, Ioannou Y, Osborn H P, et al. Scientific domain knowledge improves exoplanet transit classification with deep learning[J]. The Astrophysical journal letters, 2018, 869(1): L7.

[2] Batalha N M, Borucki W J, Bryson S T, et al. Kepler's first rocky planet: Kepler- 10b[J]. The Astrophysical Journal, 2011, 729(1): 27.

[3] William J Borucki 2016 Rep. Prog. Phys. 79 036901

[4] Burke C J, Christiansen J L, Mullally F, et al. Terrestrial planet occurrence rates for the Kepler GK dwarf sample[J]. The Astrophysical Journal, 2015, 809(1): 8.

[5] Karen A. Collins et al 2018 AJ 156 234

[6] Dattilo A, Vanderburg A, Shallue C J, et al. Identifying exoplanets with deep learning. ii. two new super-earths uncovered by a neural network in k2 data[J]. The Astronomical Journal, 2019, 157(5): 169.

[7] Everett M E, Barclay T, Ciardi D R, et al. High-resolution multi-band imaging for validation and characterization of small Kepler planets[J]. The Astronomical Journal, 2015, 149(2): 55.

[8] Kaltenegger L, Selsis F, Fridlund M, et al. Deciphering spectral fingerprints of habitable exoplanets[J]. Astrobiology, 2010, 10(1): 89-102.

[9] Livingston J H, Endl M, Dai F, et al. 44 Validated Planets from K2 Campaign 10[J]. The Astronomical Journal, 2018, 156(2): 78.

[10] Rowe J F, Bryson S T, Marcy G W, et al. Validation of Kepler's multiple planet candidates. III. Light curve analysis and announcement of hundreds of new multi-planet systems[J]. The Astrophysical Journal, 2014, 784(1): 45.

[11] Shallue C J, Vanderburg A. Identifying exoplanets with deep learning: A five- planet resonant chain around kepler-80 and an eighth planet around kepler- 90[J]. The Astronomical Journal, 2018, 155(2): 94.

[12] Vigan A, Patience J, Marois C, et al. The International Deep Planet Survey-I. The frequency of wide-orbit massive planets around A-stars[J]. Astronomy Astrophysics, 2012, 544: A9.

[13] Wright J T, Gaudi B S. Exoplanet detection methods[J]. arXiv preprint arXiv:1210.2471, 2012.