Searching for Exoplanets using Artificial Intelligence

Kyle A. Pearson
Lunar and Planetary Laboratory
University of Arizona
Tucson, AZ 85719
pearsonk@lpl.arizona.edu

Leon Palafox
Lunar and Planetary Laboratory
leonp@lpl.arizona.edu

Caitlin A. Griffith
Lunar and Planetary Laboratory
griffith@lpl.arizona.edu

Abstract

In the last decade, over a million stars were monitored to detect transiting planets. Manual interpretation of potential exoplanet candidates is labor intensive and subject to human error, the results of which are difficult to quantify. Here we present a new method of detecting exoplanet candidates in large planetary search projects which, unlike current methods uses a neural network. Unlike past transit detection algorithms deep nets learn to recognize planet features instead of relying on hand-coded metrics that humans perceive as the most representative. Our convolutional neural network is capable of detecting Earth-like exoplanets in noisy time-series data with a greater accuracy than a least-squares method. Our deep net can generalize to different time-series after interpolation without compromising performance. As validated by our deep net analysis of Kepler light curves, we detect periodic transits consistent with the true period without any light curve fitting. Our study indicates that machine learning will facilitate the characterization of exoplanets in future analysis of large astronomy data sets.

1 Introduction

Transiting exoplanets provide a remarkable opportunity to detect planetary atmospheres through spectroscopic features. During primary transit, when a planet passes in front of its host star, the light that transmits through the planet's atmosphere reveals absorption features from atomic and molecular species. Currently 3,513 exoplanets have been discovered from space missions (Kepler (Borucki u.a., 2010), K2 (Howell u.a., 2014) and CoRoT (Auvergne u.a., 2009)) and from the ground (HAT/HATnet (Bakos u.a., 2004), SuperWASP (Pollacco u.a., 2006), KELT (Pepper u.a., 2007)). Future planet hunting surveys like TESS, PLATO and LSST plan to increase the thresholds that limit current photometric surveys by sampling brighter stars at faster cadences and over larger field of views (LSST Science Collaboration u.a. 2009; Rauer u.a. 2014). Kepler's initial four-year survey revealed $\sim$15% of solar type stars have a 1–2 Earth-radius planet with an orbital period between 5–50 days (Fressin u.a. 2013; Petigura u.a. 2013). The detection of such small Earth-sized planets are difficult because the transit depth, $\sim$100 ppm for a solar type star, reaches the noise limit of current photometric surveys and is below the average stellar variability. Stellar variability is present in over 25% of the 133,030 main sequence Kepler stars and ranges between $\sim$950 ppm (5th percentile) and $\sim$22,700 ppm (95th percentile) with periodicity between 0.2 and 70 days (McQuillan u.a., 2014). The analysis of data in the future needs to be both sensitive to Earth-like planets and robust to stellar variability.

Workshop on Deep Learning for Physical Sciences (DLPS 2017), NIPS 2017, Long Beach, CA, USA.

Classical techniques to find planets maximize the correlation between data and a simple transit model via a least-squares optimization, grid-search, or matched filter approach (Kovács u.a. 2002; Jenkins u.a. 2002; Carpano u.a. 2003; Petigura u.a. 2013). A least-squares optimization aims to minimize the mean-squared error (MSE) between data and a model. Since the transit parameters are unknown a priori, a simplified transit model is constructed with a box function. Least-square optimizers are susceptible to finding local minima when trying to minimize the MSE and, thus, can result in inaccurate transit detections unless the global solution can be found. When individual transit depths are below the scatter, as is the case for Earth-like planets currently, constructively binning the data can increase the signal-to-noise (SNR). Grid-searches utilize binning by performing a brute-force evaluation over different periods, epochs and durations to search for transits either with a Least-squares approach (Kovács u.a., 2002); or matched-filter (Petigura u.a., 2013). A matched filter approach tries to optimize the signal of a transit by convolving the data with a hand-designed kernel/filter to accentuate the transit features.

The ideal algorithm for detecting planets should be fast, robust to noise and capable of learning and abstracting highly non-linear systems. A neural network trained to recognize planets with simulated data provides the ideal platform. Deep nets are composed of layers of "neurons", each of which are associated with different weights to indicate the importance of one input parameter compared to another. Our neural network is designed to make decisions, such as whether or not an observation detects a planet, based on a set of input parameters that treat, e.g. the shape and depths of a light curve, the noise and systematic error, such as star spots. The discriminative nature of our deep net can only make a qualitative assessment of the candidate signal by indicating the likelihood of finding a transit within a subset of the time series. The advantage of a deep net is that it can be trained to identify very subtle features in large data sets. This learning capability is accomplished by algorithms that optimize the weights in such a way as to minimize the difference between the output of the deep net and the expected value from the training data. Deep nets have the ability to model complex non-linear relationships that may not be derivable analytically. The network does not rely on hand designed metrics to search for planets, instead it will *learn* the optimal features necessary to detect a transit signal from our training data.

This paper summarizes our previous work (Pearson u.a., 2017) where we design various deep learning algorithms to recognize planetary transit features from a training data set. Below, we briefly explain the architecture of our deep learning algorithm. Afterwards, we evaluate time series data of known planets in the Kepler mission. Finally, we compare our algorithm to a classical detection method and summarize our findings.

2 Training Data and Detection Algorithm

Simulated training data is used to teach our deep net how to predict single planetary transits in noisy photometric data. The simulated data is similar to what we would expect from a real planetary search survey. The training data are computed using the analytic expressions of Mandel u.a. 2002 to generate transit light curves. Parameters such as the transit depth and orbital period are varied to produce different light curve shapes for a total of 311040 training samples. Additionally, we add in a quasi-periodic sinusoid that varies in amplitude and frequency reminiscent of instrumental systematics and stellar variability. Each synthetic light curve has a non-transit sample using the same systematic parameters but newly generated noise of the same distribution shape and size. This allows our deep net to differentiate between transit and non-transit signals. The synthetic data are normalized to unit variance and have the mean subtracted off prior to input in the deep net. Various light curves and systematic trends are shown in Figure 1. After the deep net is trained, we use the network to assess the likelihood of potential planetary signals in data it has not seen before. The test dataset (933120 samples) consists of the same parameters used in our training data except the range of noise is much larger so that we can estimate its detection sensitivity.

2.1 Convolutional Neural Network

The photometric measurements in a light curve are correlated to one another through time. Convolutional neural networks (CNN 1D; LeCun u.a. (1999)) utilize convolutions and down sampling to compute local properties of the data when features are correlated to one another. Planet finding techniques in the past have used convolutions via a matched filter approach however the filters are

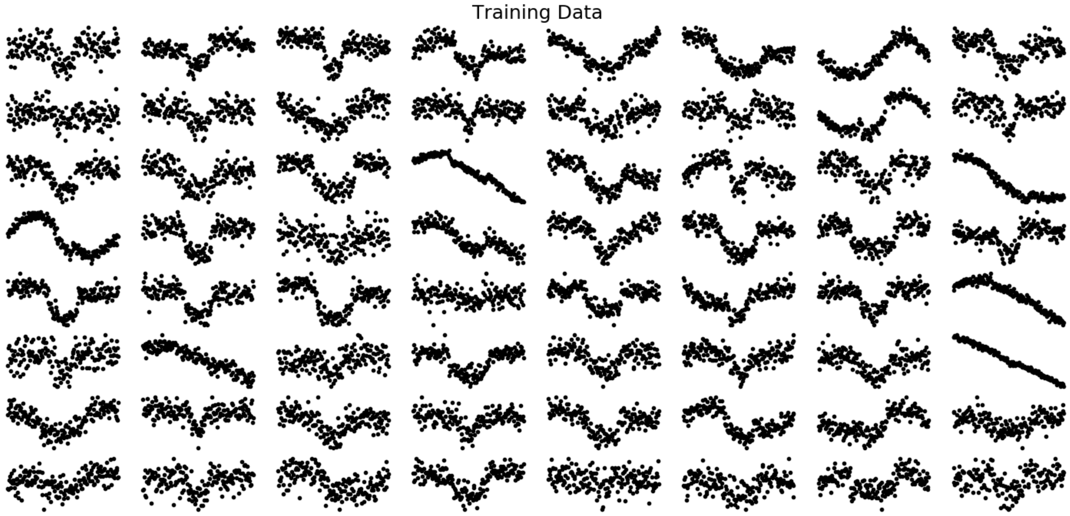

Figure 1: A random sample of our training data shows the differences between light curves and systematic trends. Each transit was calculated with a 2 minute cadence over a 6 hour window for a total of 180 time-ordered input features and the transit parameters vary based on the grid in Table 1 of Pearson u.a. 2017. The y-axis is relative flux/brightness of the host star and the dip at the center of each light curve corresponds to when the planet passes in front of the star.

hand designed and only one is used. Our CNN 1D uses 4 filters each containing 6 weights that are optimized using the training data. After the data has been convolved with a filter we down sample by averaging every three data points together to reduce the number of features for the next layer. We use an average pooling layer to help reduce the scatter from sources of noise. The average pooling layer mimics binning observational measurements in time. After the input data are convolved and down sampled, we concatenate each light curve and use it as the input to a fully connected network with a layer size of 64,32,8,1. ReLU (Hahnloser u.a. (2000); Nair u.a. (2010)) is used as the activation function in every layer of our network except for the last layer which uses a sigmoid function. We initialize the neuron weights following a method in He u.a. (2015) found to help networks (e.g. 30 convolutional/fully connected layers) converge and prevent saturation. Optimization of the weights for our deep net is done by minimizing the loss function of our system, the cross-entropy. The weights in each layer are optimized using a backward propagation scheme (Werbos, 1974). We employ the use of Nesterov momentum to modify the weight update by predicting the gradient at a new position and correcting the gradient at the current position (Nesterov, 1983). We use the common technique of stochastic gradient descent (SGD), whereby we determine the gradient of the loss function using subsets of the training data (here 128 samples) (Bottou, 1991). Additionally, when we train, we cycle through all of the training data 30 times but at each epoch the samples in the batches are randomized. We employ the use of dropout with 25% of our neurons on the first layer of the fully connected network (Srivastava u.a., 2014). The neural network relies on a handful of parameters that define the architecture (e.g. the hidden layer size and learning rate) which affect the performance. Tuning of these parameters was accomplished from a grid search where we trained over 1000 different neural works and chose the configuration that yielded the best performance. The optimal parameters are a regularization weight of 0, learning rate of 0.1, momentum of 0.25 and a decay rate of 0.0001. We employ the use of TensorFlow to design our neural network (Abadi u.a., 2015). The code for our project is provided online[1]

3 Time Series Evaluation

The Kepler exoplanet survey was first launched in 2009 and has acquired over 22 million light curves of transiting exoplanets that range in size from Earth to Jupiter and above. We only use one quarter of the Kepler data without any pre-processing to validate our neural network. Due to constraints

[1] https://github.com/pearsonkyle/Exoplanet-Artificial-Intelligence

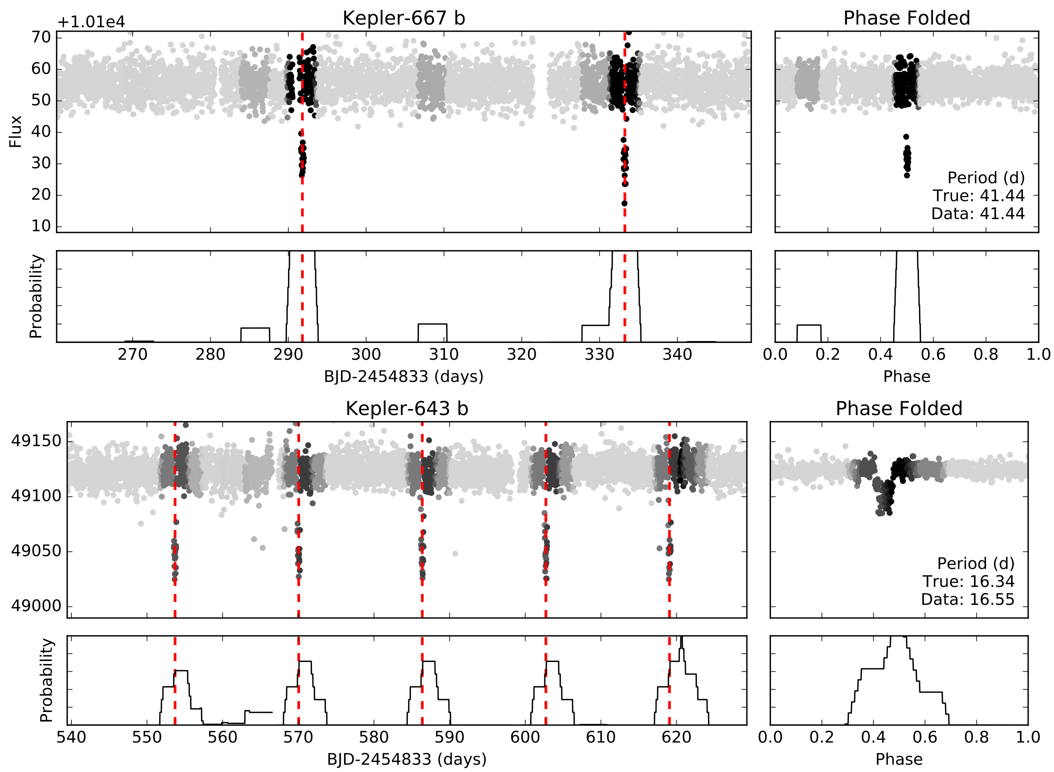

Figure 2: The color of the data points are mapped to the probability of a transit being present. The red lines indicate the true ephemeris for the planet taken from the NASA Exoplanet Archive. The period labeled “Data” is estimated by finding the average difference between peaks in the probability time plot (bottom left) and is then used to compute the phase folded light curve. This estimated period in most cases is similar to the true period and differs if the planet is in a multi-planet system or has data with strong systematics.

regarding the transit duration within our time series window we limit transits to ∼7–15 hours with an orbital period greater than 90 hours. Detecting individual transit events with our algorithm requires a certain amount of features in-transit to yield an accurate prediction. We chose random targets with an SNR larger than 1.5 and a transit period greater than 90 hours so that we can acquire multiple transit events in a single quarter of Kepler data. Figure 2 shows the results from a small analysis of Kepler targets. The probability of detecting a transit is mapped to the transparency of the dots in each light curve. From the probability-time plot we can estimate the period of the planet by finding the average difference between peaks. We can recover 99% of the Kepler transits above an SNR of 1.5.

4 Conclusion

In the era of “big data” manual interpretation of potential exoplanet candidates is a labor intensive effort and difficult to do with small transit signals (e.g. Earth-sized planets). Exoplanet transits have different shapes, as a result of, e.g. the stellar activity and planet size and orbital period. Thus a simple template does not suffice to capture the subtle details, especially if the signal is below the noise or strong systematics are present. We use an artificial neural network to learn the photometric features of a transiting exoplanet. Deep machine learning is capable of processing millions of light curves in a matter of seconds. The discriminative nature of neural networks can only make a qualitative assessment of the candidate signal by indicating the likelihood of finding a transit within a subset of the time series. We validate our deep nets on light curves from the Kepler mission and detect periodic transits similar to the true period without any model fitting. We find that our CNN 1D algorithm in the worst SNR regime (0.8) was able to achieve an accuracy of ∼60% compared to an accuracy of ∼1% for the classical least-squares algorithm in Foreman-Mackey u.a. 2015. Machine learning techniques provide an artificially intelligent platform that can learn subtle features from large data sets in a more efficient manner than a human. In the future we would like to explore the use of more deep learning techniques (e.g. long short-term memory or PReLU) to increase the detection robustness to noise. Additionally, active research is currently being done in machine learning to optimize the network architecture and have it adapt to specific problems. Adding a pre-processing step has the potential to greatly improve the transit detection performance by removing systematics from the time series (e.g. Aigrain u.a. (2017)).

References

References

Werbos, P. J.(1974): *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.

Nesterov, Yurii(1983): *A method of solving a convex programming problem with convergence rate O (1/k2)*In: Soviet Mathematics Doklady, 2: 372–376.

Bottou, Léon(1991): *Stochastic Gradient Learning in Neural Networks*In: Proceedings of Neuro-Nîmes 91.

LeCun, Y. / Haffner, P. / Bottou, L. / Bengio, Y.(1999): *Object Recognition with Gradient-Based Learning*In: Feature Grouping.

Hahnloser, Richard H. R. / Sarpeshkar, Rahul / Mahowald, Misha A. / Douglas, Rodney J. / Seung, H. Sebastian(2000): *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*947 EP -.

Kovács, G. / Zucker, S. / Mazeh, T.(2002): *A box-fitting algorithm in the search for periodic transits*369-377.

Mandel, K. / Agol, E.(2002): *Analytic Light Curves for Planetary Transit Searches*L171-L175.

Jenkins, J. M. / Caldwell, D. A. / Borucki, W. J.(2002): *Some Tests to Establish Confidence in Planets Discovered by Transit Photometry*495-507.

Carpano, S. / Aigrain, S. / Favata, F.(2003): *Detecting planetary transits in the presence of stellar variability. Optimal filtering and the use of colour information*743-753.

Bakos, G. / Noyes, R. W. / Kovács, G. / Stanek, K. Z. / Sasselov, D. D. / Domsa, I.(2004): *Wide-Field Millimagnitude Photometry with the HAT: A Tool for Extrasolar Planet Detection*266-277.

Pollacco, D. L. u.a.(2006): *The WASP Project and the SuperWASP Cameras*1407-1418.

Pepper, J. u.a.(2007): *The Kilodegree Extremely Little Telescope (KELT): A Small Robotic Telescope for Large-Area Synoptic Surveys*923-935.

LSST Science Collaboration u.a.(2009): *LSST Science Book, Version 2.0*.

Auvergne, M. u.a.(2009): *The CoRoT satellite in flight: description and performance*411-424.

Borucki, W. J. u.a.(2010): *Kepler Planet-Detection Mission: Introduction and First Results*977.

Nair, Vinod / Hinton, Geoffrey E.(2010): *Rectified Linear Units Improve Restricted Boltzmann Machines*In: Proceedings of the 27th International Conference on Machine Learning (ICML-10)807-814.

Petigura, E. A. / Marcy, G. W. / Howard, A. W.(2013): *A Plateau in the Planet Population below Twice the Size of Earth*69.

Fressin, F. u.a.(2013): *The False Positive Rate of Kepler and the Occurrence of Planets*81.

Srivastava, Nitish / Hinton, Geoffrey / Krizhevsky, Alex / Sutskever, Ilya / Salakhutdinov, Ruslan(2014): *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*1929-1958.

McQuillan, A. / Mazeh, T. / Aigrain, S.(2014): *Rotation Periods of 34,030 Kepler Main-sequence Stars: The Full Autocorrelation Sample*24.

Howell, S. B. u.a.(2014): *The K2 Mission: Characterization and Early Results*398-408.

Rauer, H. u.a.(2014): *The PLATO 2.0 mission*249-330.

Foreman-Mackey, D. / Montet, B. T. / Hogg, D. W. / Morton, T. D. / Wang, D. / Schölkopf, B.(2015): *A Systematic Search for Transiting Planets in the K2 Data*215.

He, K. / Zhang, X. / Ren, S. / Sun, J.(2015): *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*.

Aigrain, S. / Parviainen, H. / Roberts, S. / Reece, S. / Evans, T.(2017): *Robust, open-source removal of systematics in Kepler data*.

Pearson, K. A. / Palafox, L. / Griffith, C. A.(2017): *Searching for Exoplanets Using Artificial Intelligence*.

Abadi, Martín u.a. (2015): *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* , Software available from tensorflow.org.