Machine Learning Classification of Gravitational Waves

Tom Casaletto  
*xxx*  
*Lockheed Martin*xxx, USA  
email  
  
  
  
Anthony Fisher  
*xxx*  
*Lockheed Martin*Sunnyvale, USA  
email  
  
  
  
Phillip Shirts  
*Solar and Astrophysics Laboratory*  
*Lockheed Martin*Palo Alto, USA  
phillip.g.shirts@lmco.com  
  
  
Joel Wiser  
*xxx*  
*Lockheed Martin*location  
email

*Abstract*— Several data preprocessing, analysis and machine learning techniques were employed in our group’s effort to identify synthetic gravitational waves from data provided by the [Kaggle.com](http://kaggle.com/) G2NetGravitation Wave Detection Research Prediction Competition, “Find gravitational wave signals from binary black hole collisions.”

Keywords—gravitational waves, machine learning, Convolutional Neural Net (CNN), Q-Transforms, data pre-processing, continuous wavelet transform, spectrograms, Kaggle contest

# Introduction

Gravitational waves are “ripples” in space-time caused by the movement of massive objects. The faster the movement and the more massive the object, the stronger the gravitational wave created. Albert Einstein’s 1915 general theory of relativity predicted the existence of gravitational waves, and they were detected for the first time 100 years later, in 2015, by twin Laser Interferometer Gravitational-wave Observatory (LIGO) observatories. The gravitational waves that were detected were generated by colliding black holes.  
The gravitational wave signals are very weak, of order 10‑21. Convolutional Neural Network (CNN) approaches are a promising avenue for identifying the weak gravitational wave signals. In order to encourage development of CNN approaches to Gravitational wave detection, [Kaggle.com](http://kaggle.com/) is hosting a competition, “G2Net Gravitational Wave Detection, “Find gravitational wave signals from binary black hole collisions” with $15,000 in prizes for top signal detectors using CNN. (<https://www.kaggle.com/c/g2net-gravitational-wave-detection>). The final submission deadline is September 20, 2021.

# Continuous Wavelet Transform

In this section we describe the approach to take advantage of the success of Convolutional Neural Networks (CNN) in processing images for classification. This has been applied across multiple domains (references?). Here we propose to transform the 1-dimensional time series from each detector site into a 2-dimensional image using the Continuous Wavelet transform (CWT). This is accomplished by taking the spectrogram of the CWT.

The CWT is given by transforming a time signal into its frequency (psi) and amplitude (a) components as shown in Fig. 1.

A diagram of the pre-processing is shown in Fig. 2. A training example where there is a gravitational wave is shown on the upper left (Signal). Similarly, a training example where there is no gravitational wave (just detector noise) is shown in the upper right (Noise). The first row of diagrams shows detector output for each of the 3 sites for 2 seconds (4096 samples). From the naked eye it is hard to tell if these represent time series from a gravitational wave or just noise. Note the scale of the Y-axis is 1e-20. The second row of diagrams shows the result of the first step of pre-processing: normalizing the signals by the z-transform. Again it is difficult to tell the difference between detector output with a gravitational wave and with just noise. The third row of diagrams shows the CWT spectrogram of the normalized signal after applying a bandpass filter to remove frequencies below 15 Hz and above 500 Hz. It was given in the problem statement that typical gravitational waves are around 350 Hz.

Fig. 3 shows a diagram of the CNN used to process the spectrogram images. Each image is 300x97 pixels. As shown the CNN consists of multiple convolutional and pooling layers for the feature extractor. The output layer classifies the image as 1 for gravitational wave detected or 0 for no detection.

Fig. 5 shows the summary description of the Tensorflow model.

Fig. 6 shows the initial results of running 1000 training examples through the model using a 60/20/20 training/test/validation split. During the training, the accuracy increased with each epoch, eventually reaching 0.795 after 30 epochs. The divergence of the accuracy and validation shows that overfitting is happening. Though it appears our model is getting more accurate, running the test data through achieves a score of 0.43. If this were the entirety of the training set we should actually take the opposite of whatever our model predicts! However, the low score is probably due to the small amount of training/test samples used for this analysis.

Fig. 4 shows the results of running 10000 training examples through the model using a 60/20/20 training/test/validation split. Here we use regularization in training the model and see that overfitting is not occurring. However, our test accuracy is 0.505 so our model is not much better than flipping a coin. This indicates our pre-processing is not doing much to differentiate detector output with gravitational waves from those with just noise. So more work needs to be done on the pre-processing step. Some ideas include:

1) Changing the bandpass filter parameters to allow more or less data through.

2) Making larger/smaller images to pass into the CNN.

In summary, an approach was developed to perform pre-processing on detector output using the Continuous Wavelet Transform. The data was processed through a Convolutional Neural Network using Tensorflow. Results were inconclusive regarding the ability of the model to detect gravitational waves and ideas for further study were presented.

# Q-Transforms

For pre-processing in this approach, we transformed each time-series data sample into a spectrogram using a Q-transform. The resulting spectrogram is an image showing the power of different frequencies within the signal as it moves through time. The resulting images were also resized to a smaller resolution to be more manageable.

Each sample included readings from three different gravitational wave detectors. Each reading was transformed into a spectrogram and then the three were stacked together as different channels, much like the red, green, and blue channels that make up an RGB image. This stack of three spectrograms was used as the input to a convolutional neural network.

Because each sample was fairly large, and there were a huge number of samples in the dataset, it was impossible to hold all samples necessary for training in memory at once. So, to train the network (or make test predictions), we wrote a custom generator class derived from keras’ Sequence class. This allowed only a small batch of data to be loaded into memory at once. The generator would read in a batch of data (either the raw time-series or pre-processed data, if available), perform the necessary transformations (if desired), and send the result to the neural network model. It would repeat this for as many batches as were necessary.

The model was configured as shown in Fig. 7. The spectrogram was resized so that the values data\_shape[0], data\_shape[1], and data\_shape[2] were 128, 120, and 3 respectively.

We initially began training the convolutional neural network model on smaller amounts of data, 1,000 or fewer samples. With this small amount of data, the model appeared to overfit the data, similar to as was seen in the wavelet approach. Over 10-20 epochs, the training accuracy would increase nearly to 1 while training loss decreased. But, the validation accuracy stayed nearly the same (around 0.55) or decreased in later epochs. We attempted to counter this overfitting effect by using a larger number of data samples.

The best results for the Q-transform method were obtained by using the first 200,000 data samples for training and validation with a random 80% train, 20% validation split. After five training epochs, the model was able to achieve 66% accuracy on the validation set.

# Approach 3

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**