Utilizing Machine Learning to Search for LIGO Sources: Interim Report

Sarah McCarthy[1]

July 6, 2018

[1]Grinnell College,
1115 8th Ave. Grinnell, IA 50112

1. Introduction

In 1916, Albert Einstein predicted the existence of gravitational waves, ripples created by accelerating masses that would propagate through spacetime [1,2]. However, he believed they would be too small for human detection. On September 14, 2015 at 09:50:45 UTC, nearly one century after Einstein's prediction, both detectors of the Laser Interferometric Gravitational-wave Observatory (LIGO) simultaneously observed a transient gravitational-wave signal [3]. This was the first direct detection of gravitational waves, as well as the first detection of a binary black hole merger. The two LIGO observatories are in Livingston, LA and Hanford, WA. Each instrument is a dual-recycled Michelson interferometer with 4 km arms [4]. LIGO's discoveries were made possible by a factor of 10 sensitivity improvement in the frequency regime around 100 Hz.

One of the challenges for LIGO is differentiating between signal and noise. Transient noise arises as short $O(seconds)$ glitches in the data that can mimic true transient astrophysical gravitational wave signals including binary black hole mergers. One powerful method to identify signals with waveforms that are well predicted by Einstein's relativity, including neutron star and black hole binaries, is matched filtering, which calculates the cross-correlation between modeled templates and the noisy gravitational wave detector data. The PyBC pipeline, which identified all LIGO discoveries to date, employs matched filtering to calculate the signal-to-noise ratio (SNR) between all modeled templates considered in a gravitational wave search. To help make the pipeline more robust to noise, PyCBC uses a χ^2 test to downweight the SNR of events where the data does not match the modeled template well. Any times where the re-weighted SNR is above threshold are saved as "triggers" [5]. Any given trigger will have multiple associated modeled templates with non-zero re-weighted SNR, each with individual re-weighted SNR tends to be densely clustered in total mass and effective spin.

One of the outputs of this process is a set of parameters, such as total mass, effective spin, and maximum re-weighted SNR [6]. These parameters can be plotted for a given stretch of data. For gravitational wave signals, the plots typically have a compact area of high maximum re-weighted SNR, with a relatively low re-weighted SNR over other values of total mass and effective spin. If the event is noise, typically maximum re-weighted SNR is not well localized in these template parameters.

Our approach to improving the PyCBC pipeline performance by limiting the impact of transient noise is to use a convolutional neural network and image classification. A neural network is a biology-inspired computer program in which a computer learns a specified task from a series of provided 'training' examples. Neural networks have successfully been used to classify images of LIGO data in the past [7]. We will build an image classifier that takes as input a plot representing the distribution of templates associated with a trigger time in total mass and effective spin. We expect that the much more well-localized appearance of true signals in this parameter space will serve as a powerful distinguishing feature for our machine learning image classifier.

2. Objectives

The aim of my summer research project is twofold:
1) Create a convolutional neural network that will differentiate signal and noise in LIGO data
2) Test this algorithm's performance on increasingly large data sets

3. Approach

I will first design and build a simple convolutional neural network (CNN) algorithm that can intake images of the total mass and effective spin distribution of PyCBC triggers and output some likelihood that the trigger belongs to the 'signal' class or the 'noise' class. Next I will need to develop a training set to train the CNN to make accurate classifications. I will inject a series of simulated gravitational wave signals into data from Advanced LIGO's second observing run and flag each of these as part of the 'signal' class. For the glitch class, I will use glitch examples identified by Gravity Spy.

After I have trained the CNN with known examples of both the signal and glitch classes, I will test it with a test data set. To ensure this data is independent from the training set I will use a different period of Advanced LIGO data, and I will follow the same method as above to inject signal examples and identify glitch examples. I will evaluate the performance of the CNN by producing a confusion matrix for the test data which will calculate the fraction of mis-classified PyCBC triggers for each class. I anticipate tuning the CNN based on these results.

At this point, I will be ready to expand my training set and data set to an extended set of O2 data. Then, I will produce and analyze my results. All simulations will be run from my personal laptop using Caltech's LDAS computing cluster for computational power.

4. Project Schedule

I will follow the timeline below:

Weeks 1-2: Learn LIGO software and computing clusters, assemble basic code
Weeks 3-5: Tune algorithm using isolated test cases
Weeks 6-8: Run on extended data set
Weeks 9-10: Combine results, prepare final report and presentation

5. Current Work

Upon my initial arrival at Caltech, I began to familiarize myself with LIGO data through Gravity Spy. I practiced categorizing different types of glitches so that I could develop intuition categorizing noise and distinguishing it from signals. This work was important for the potential to need to troubleshoot further in the project when building the training and test data sets. I then designed the data feature that I intend to feed into the machine learning algorithm, studying how the SNR varied with certain parameters like end time, mass, and spin, to name a few. I more strictly defined the information to feed into the classifier so that I can reliably represent the data that I am putting into the classifier, and I modified the existing software to automate plots for an inputted time.

Once I familiarized myself with promising feature sets for a signal, I then tested the plots on noise. I used Python to find the times that had maximum SNR and were not artificial. I reran the same density plots, this time using noisy data, and searched for more interesting behavior. I decided that the plots of reduced χ^2 v. template duration and end time v. template duration exhibited the most distinct behavior. Once I had done a preliminary analysis of the plots, I modified the code so that it would display the data with more confined axis limits and only analyze that portion of the data. Part of this required that I modify the binning of the data so that it would best display meaningful information. Examples of the impact of binning can be seen below in Figure 1.

Alongside my work on the aforementioned tasks, I also read about CNNs in Keras and TensorFlow and tested some sample scripts to familiarize myself with the steps of creating a CNN so I would be prepared for the next step of injecting this data into a CNN. The next step for me is to write code to automate this generation on the top 100 or so SNR times within any given data set to see if similar shapes hold with another data set.

6. Challenges

One of the difficulties that arose thus far was determining what made the plots interesting enough to consider them as viable for training the CNN. I struggled to develop an understanding of what combinations of data would provide a meaningful plot, especially since a large portion of that entailed learning how to quickly yet effectively interpret code written by someone else. A similar problem that arose from that issue was doing unnecessary work. It took me a

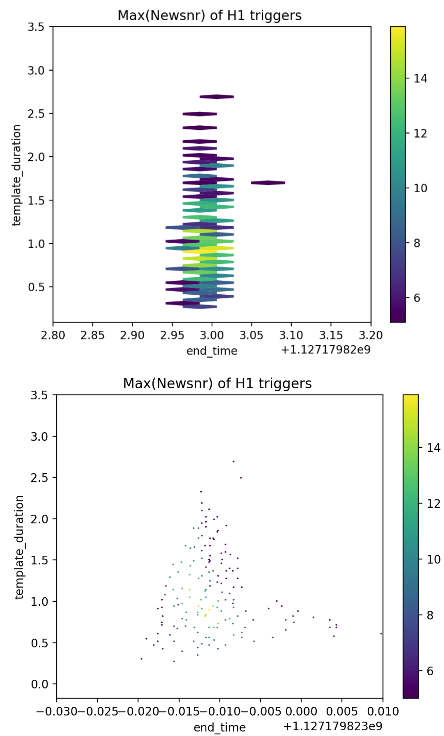

Figure 1: The top figure is an example of bad binning, where the bins are too wide to effectively distinguish the shape of the distribution. On the bottom, the binning has been adjusted so that a more distinct feature space can be seen.

long time to find out that code that I had been writing was already part of the code I was using, just hidden under different names or deeper in source code.

Another difficulty was determining how to constrain the plots that would allow high levels of clarity for data sets with differing distributions. This required studying the behavior that signals had and finding variations across different noise data sets to look for different patterns. Changing the binning was a significant step towards remedying this issue, and utilizing the potential to run the CNN over different data constraints will help in the future.

Potential issues that I could see arriving in the future include needing to modify the windows for CNN to use to determine if the data is signal or noise, perhaps not seeing as clear of patterns with other data sets and thus needing to modify my search parameters, and difficulty in defining strict parameters as to what constitutes a signal for these two sets of plots and what happens if it passes by one test but fails by another. There is also the potential for the CNN to only work well with either low or high mass systems, not the both. This would require me developing a second way to test for the opposite case. Regardless, the CNN will not be able to distinguish between signal and noise at low SNR because they become indistinguishable. Moving onto newer data sets also raises the potential for issues to arise. Many of the problems that may arise will not surface until I am able to run the CNN. I also will not know if the CNN is efficient or effective until I attempt to use it to classify information.

Acknowledgements

I would like to thank Dr. TJ Massinger and Dr. Jess McIver for their guidance and support throughout the summer.

References

1. A. Einstein, Sitzungsber. K. Preuss. Akad. Wiss. 1, 688 (1916).
2. A. Einstein, Sitzungsber. K. Preuss. Akad. Wiss. 1, 154 (1918).
3. B.P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration). "Observation of Gravitational Waves from a Binary Black Hole Merger." *Phys. Rev. Lett. 116. 061102* (2016)
4. LIGO Scientific Collaboration. "Advanced Ligo." *arXiv:1411.4547* (2014)
5. B.P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration). "Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914." *arXiv:1602.03844* (2016)
6. B P Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration). "Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run." *Class. Quantum Grav. 35 065010* (2018)
7. M Zevin et al. "Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science." *Classical and Quantum Gravity, Volume 34, Number 6.*(2017)