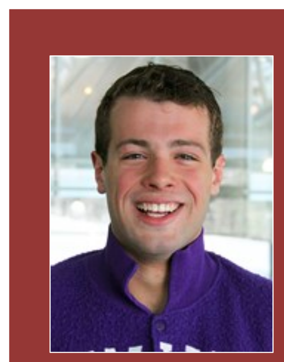
Using Artificial Neural Networks for Gravitational-Wave Glitch Identification in Advanced LIGO



Donald A. Moffa III ‘18, Kyle Rose ‘19, Leslie Wade, Madeline Wade
Kenyon College Summer Science 2017

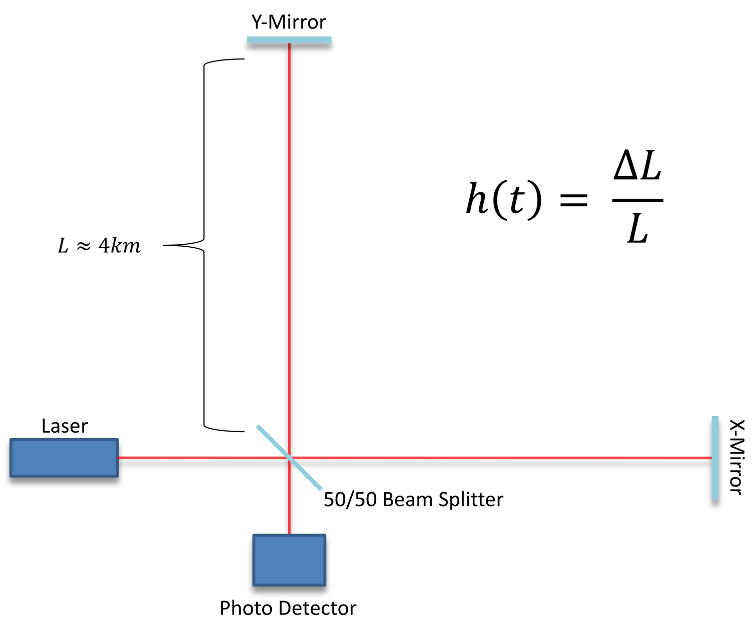
Abstract

In an effort to vet gravitational-wave data faster, the LIGO collaboration is beginning to incorporate machine learning algorithms into the vetting process. Vetting data faster will allow LIGO to veto unclean data in real time, which could improve the chance to confidently detect gravitational waves that have associated electromagnetic counterparts. In this work, we studied the effectiveness of Google’s artificial neural network software package, Tensor Flow, for identifying unclean data segments. Tensor Flow, as implemented, had comparable effectiveness to other machine learning algorithms but showed more robustness to increased feature sets.

LIGO Background

The Laser Interferometer Gravitational-Wave Observatory (LIGO) detected its first gravitational wave in 2015. Since then there have been three other gravitational wave detections from merging black holes. LIGO is essentially a larger Michelson Interferometer capable of measuring the small ripples in spacetime radiated from coalescing black holes and potentially other gravitating systems. A Michelson interferometer splits coherent light down two perpendicular arms where it is then reflected off mirrors. The light is then recombined to create destructive interference at a photodetector. These ripples cause a change in the detector’s arm length resulting in a change in the interference pattern at a photodetector. Along with this measurement, auxiliary channels record changes in all of the instrument’s components.

Figure 1: A Michelson interferometer. Light travels from the laser and is split at the 50/50 beam splitter. The split beam then travels down each beam arm, reflects off the mirrors and is recombined at the 50/50 beam splitter. The interference pattern from the recombined light shows up at the photo detector.



$$h(t) = \frac{\Delta L}{L}$$

Artificial Neural Networks

Artificial neural networks (ANN) are a concept that has been around since the 1950’s. Recently, thanks to an increase in computational power, ANNs have made great strides in computer science. The concept of an artificial neural network comes from how neurons in a brain work. When a neuron receives electric signals from other neurons, charge builds up until it surpasses a threshold, then the neuron will fire. The more a neuron is used, the stronger the action potential becomes.

In an artificial neural network, neurons are mathematically modeled by three parts: weights, a summation, and an activation function. The weights represent action potentials connecting neurons, the summation represents incoming electrical signals, and the activation function represents the threshold of the neuron. Mathematically, this can be expressed as a matrix multiplication.

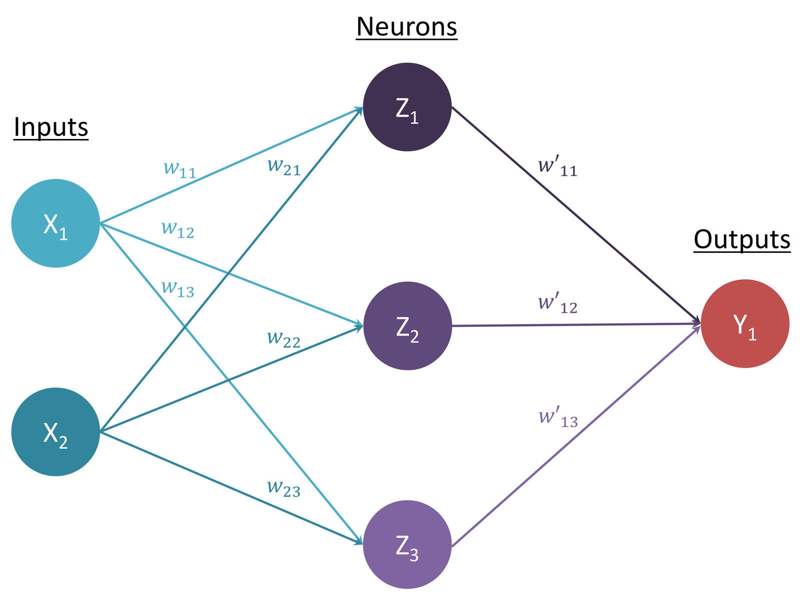


Figure 2: A typical shallow neural network. This model is called a perceptron, and is one of the simplest neural networks.

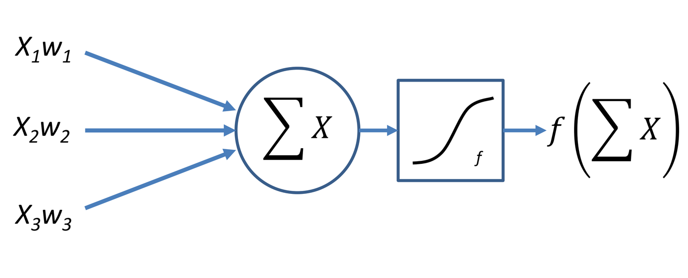
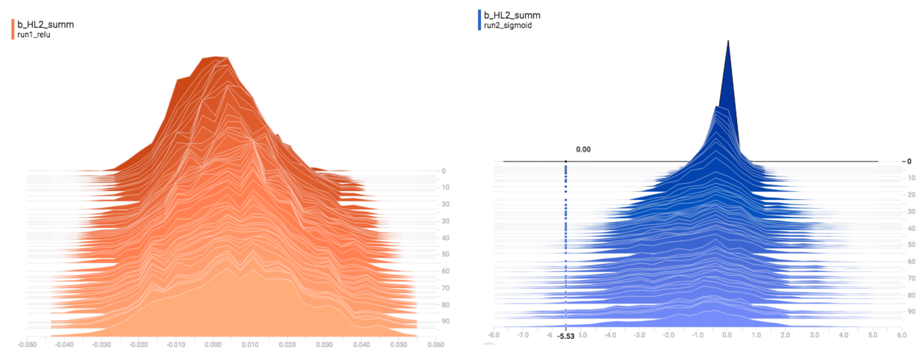


Figure 3: The flow of information through a single neuron (node). The activation function ‘f’ is represented here by a sigmoid function, but there are several other commonly used activation functions.

Training Neural Networks on LIGO Data

Training artificial neural networks requires training data and evaluation data. By looking at the expected output for a set of inputs the ANN adjusts weights so that in the future it can make accurate predictions on new data. Once trained, we use an evaluation set of data to cross validate our predictions and build a receiver operator characteristic (ROC) curve.

By looking at triggers in auxiliary channels that are safe from gravitational waves we are able to build both training and evaluation data sets. These data sets are built by taking data that is free of gravitational waves and sampling random times in the segment of data. These times will either have a glitch (unclean) or not have a glitch (clean). Glitches are identified and decomposed by match-filtering with a half-sine Gaussian template.



Figures 4 & 5: The above pictures are overlapped histograms that show how weights change with training. This is evidence that our neural network was in fact, “learning.”

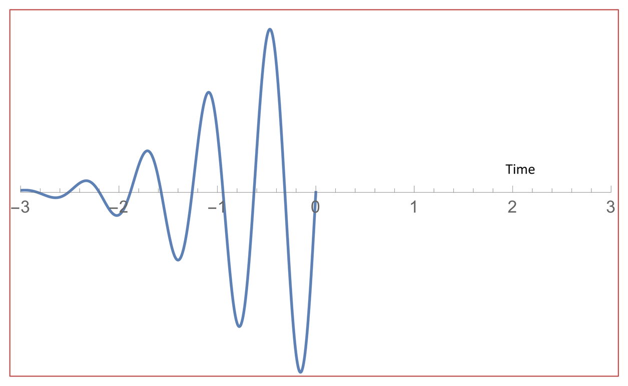


Figure 6 (above): A half-sine Gaussian used to identify non-causal glitches

Trigger: Any time we filter with a half-sine Gaussian that has a signal to noise ratio (SNR) that is above some threshold.
Glitch: A time with a trigger in the main gravitational wave.

Preliminary Findings

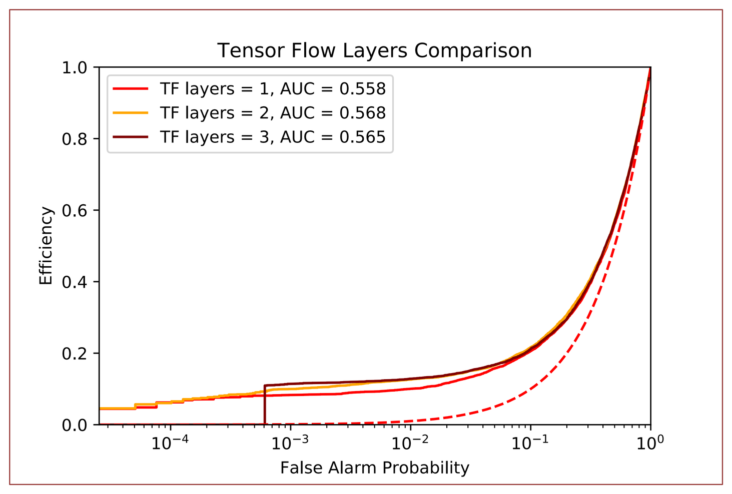


Figure 7: Comparison of shallow neural networks and deeper neural networks. Each layer contained 1000 nodes

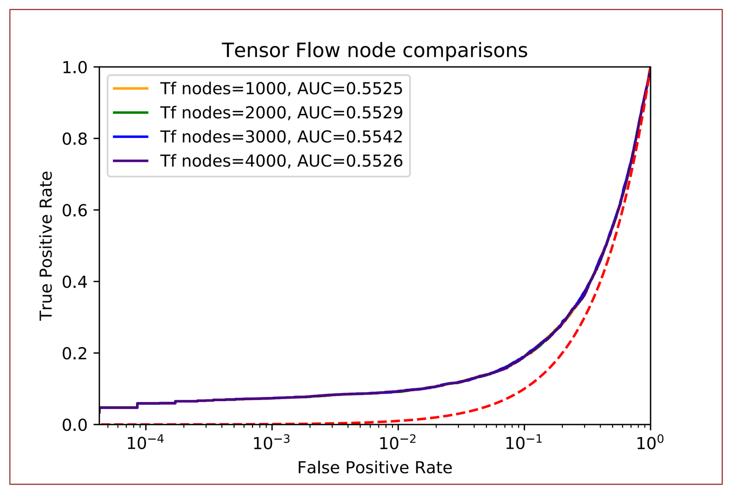


Figure 8: Comparison of how nodes effect identification of glitches

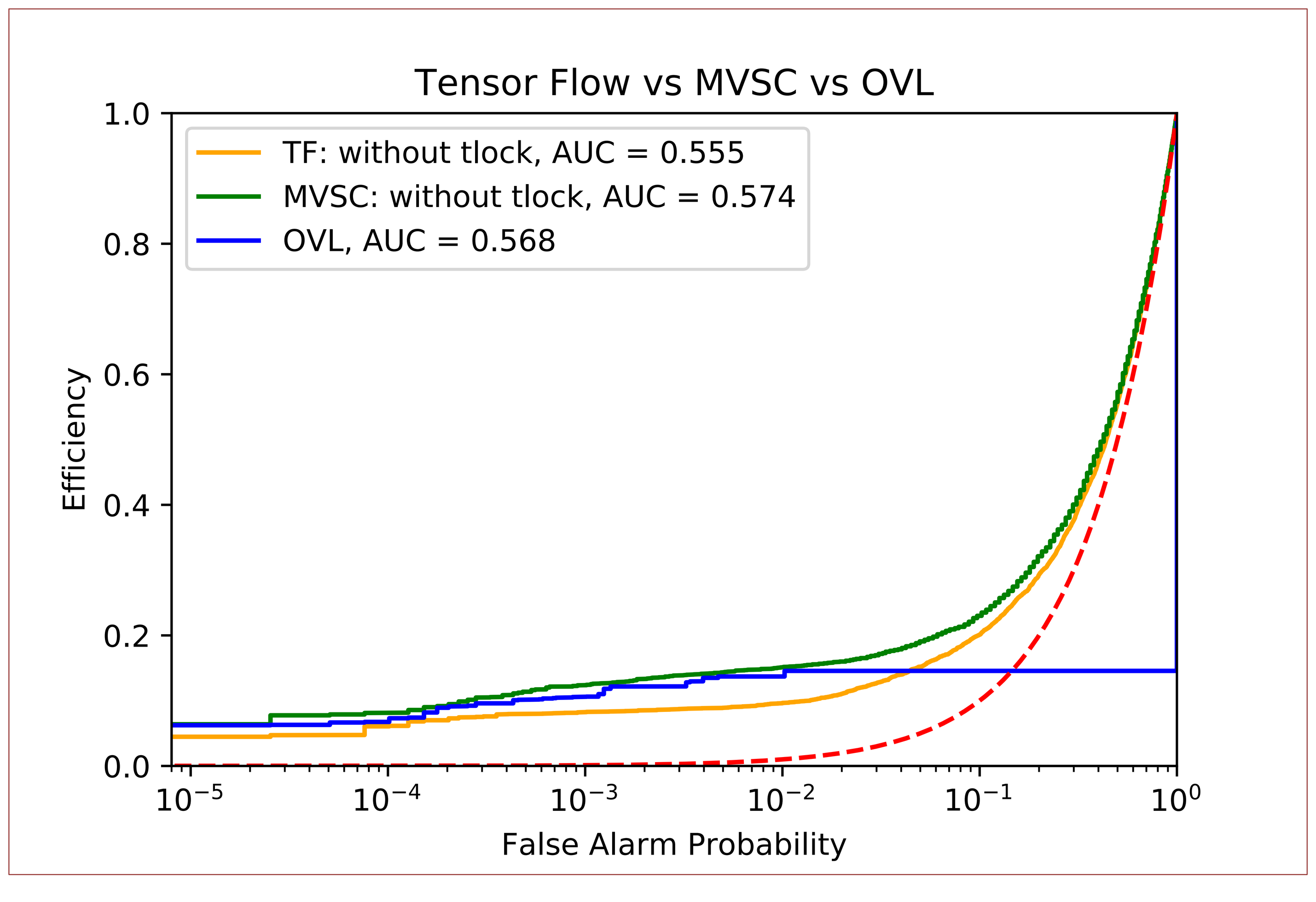


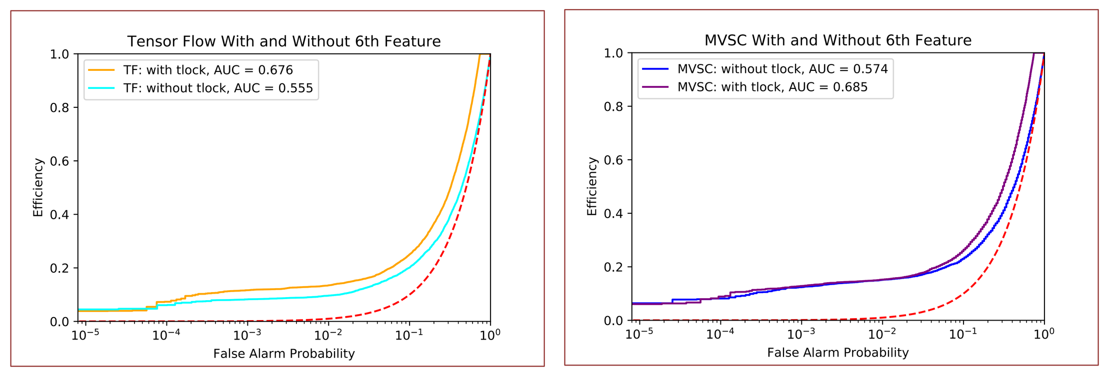
Figure 9: Tensor Flow ROC Curve compared to OVL and MVSC

$$False\ Alarm\ Probability = \frac{Number\ of\ Clean\ at\ or\ Above\ Rank}{Total\ Number\ of\ Cleans}$$

$$Efficiency = \frac{Number\ of\ Glitches\ at\ or\ Above\ Rank}{Total\ Number\ of\ Glitches}$$

Conclusions

Through this project we were able to create an artificial neural network that is able to train on auxiliary channel data and perform comparably to already existing machine learning algorithms. Due to the inherent randomness of glitches, the current methods of defining a glitch may not be capturing the whole picture and our methods can be refined. Tensor Flow showed more improvement than a comparable machine learning algorithm when the feature set of a glitch was expanded, which suggests it may be more robust to additional expansions of the feature set.”



Figures 10 & 11: Using Kyle Rose’s 6th feature, time since lock, Tensor Flow was able to more accurately identify glitches from auxiliary channels. In comparison, MVSC’s efficiency did not change in the 10^{-2} range of false alarm probability. This shows that a more accurate definition of a glitch could lead to a better classifier

Future Work

The next step in this project is to begin experimenting with different artificial neural networks. One network structure particularly interesting is known as a Recurrent Neural Network (RNN). A recurrent neural network cell, in a sense, gives a short term memory to the network. By incorporating RNN nodes, our ANN will have access to a history of events leading up to any given trigger and might therefore be more effective at predicting whether or not the trigger is correlated with a glitch in the gravitational wave channel. This method is as if the neural network is able to make predictions based on a short video rather than a single picture.

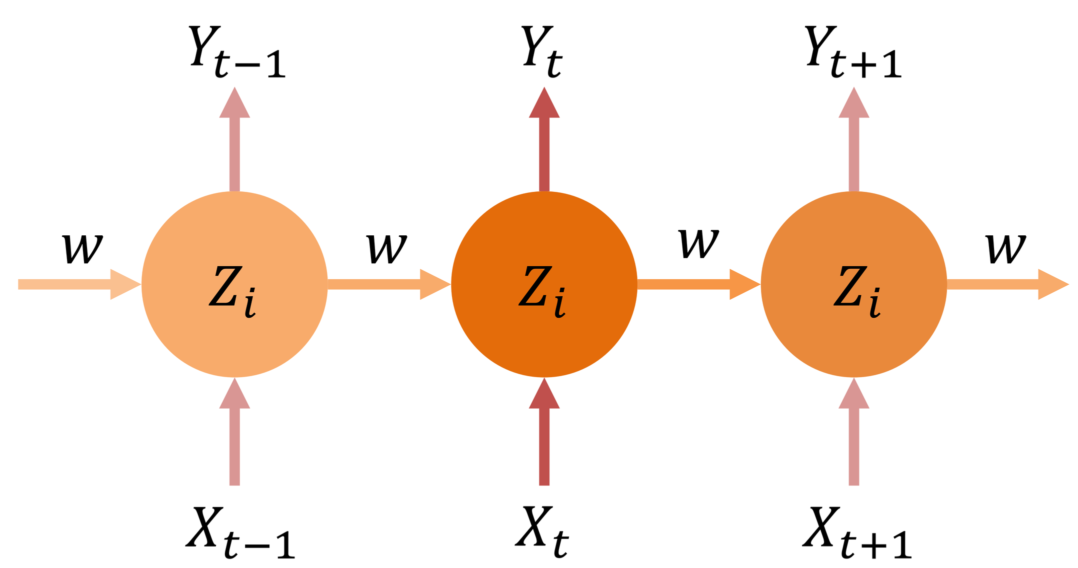


Figure 10: A single neuron from a recurrent neural network. Information at each time step is passed on to itself in a future time step allowing each neuron to have a short term memory.

Acknowledgments

I would like to thank Leslie Wade, Madeline Wade, collaborator Kyle Rose, the rest of the Kenyon LIGO lab members, the Kenyon Physics department, the Penn State LIGO group, Reed Essick, the LIGO scientific collaboration, and the summer science program for their help and support throughout this project.

References

Marsland, Stephen: Machine Learning an Algorithmic Perspective, Second Edition. Boca Raton:Taylor and Francis group, 2015. Print.
Hartle, J. B., Gravity: An Introduction to Einstein's General Relativity. San Francisco: Addison-Wesley, 2003. Print