
Gravitational-WaveNet: A Feasability Study For Deep Learning On Noisy Time Series

---

Gerardo Salazar

Abstract

An experiment is performed to test the feasability of employing deep learning models to perform binary classification on noisy time-series data, such as that collected at the LIGO gravitational wave experiment. A novel deep learning architecture, *Gravitational*-WaveNet, is introduced and evaluated on synthetic binary-merger gravitational wave data. Experiments show that this approach performs very well for sufficiently different signal and noise spectra, accuracy falls to the 60-70% range when they are visually similar.

1. Introduction

In 2016 the two detectors (LIGO and VIRGO) of the Laser Interferometer Gravitational-Wave Observatory were able to perform the first observation of gravitational waves. The gravitational waves were eminated by a binary black hole system that collapsed millions of lightyears away. This detection of gravitational waves verified Einstein's General Theory of Relativity and is one of the most remarkable scientific achievements in recent history.

The detection of gravitational waves requires processing very noisy time series data using various hands-on techniques that require expert supervision. These include things like matched-filtering, spectrogram visualizations, fourier analysis, and template bank generation. These techniques are also computationally expensive and hard to perform in an online fashion. The detection pipeline could be streamlined by inserting an online classifier to detect if there is a signal in the raw data. This positive time series could then be transferred to an analysis pipeline to determine the parameters. The main obstacle in the classification task is the very high signal-to-noise ratio present at the LIGO experiment, as the detection scale for typical gravitational wave phenomeon is displacements of magnitude equal to fractions of the width of an atom. The question then arises, can deep learning models perform well under noisy conditions?

As a first step towards automating the detection pipeline I conducted several experiments to verify the feasability of using deep learning models to classify noisy time series data. A deep learning model was trained to classify synthetic gravitational waveforms with additive gaussian noise from pure gaussian noise. The experiments and the Gravitational-WaveNet architecture used to perform them are detailed in the following sections.

2. Gravitational-WaveNet

The Gravitational-WaveNet (GWN) architecture is a deep learning model designed to filter noisy time series data into a latent space which is amenable to binary classification. It consists of one-dimensional dilated-causal convolutional layers, followed by a recurrent layer with Gated Recurrence Unit cells, and finally a dense layer with a sigmoid activation function. The network takes in a fixed-length time-series and outputs the estimated probability that the input contains a gravitational wave signal.

The idea is that the convolutional layers learn to filter the noise from the signal and then pass a compressed representation to the recurrent layer for classification. The namesake of the architecture is inspired by Google's WaveNet where dialted-causal convolutions were first introduced for learning long-term structure in time-series data in the domain of synthetic voice generation.

2.1. Convolutional Layers

The dilated-causal convolutional layers work as regular convolutional layers but enforce a causal structure on the convolutions so as to preserve the time-ordering of the original signal. Concretely, the output of the model depends only on the previous timesteps. This makes the model better suited for time series data. In additional, dilation allows for the receptive field of the network to expand and "see" a wider view of the data.

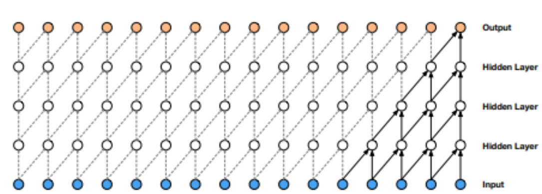

Figure 1: Visualization of dilated causal convolutions in

successive hidden layers.

The convolutional layers in GWN use a ReLU activation function given by

$$ReLU(x) = max(0, x)$$

Where x is the raw output of the convolutional layer.

2.2. Recurrent Layers

The recurrent layer in the network containes Gated Recurrence Unit (GRU) cells in place of the classical RNN cells or the more recent LSTM cells. GRU cells have been shown to perform just as well as LSTMs (which perform much better than classical RNNs) with a smaller number of parameters. The GRU maintains a hidden state to pass information from one cell to the next. Its output can be computed as

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \phi(W_h x_t + U_h(r_t \otimes h_{t-1}) + b_h)$$

Where $\otimes$ denotes element wise multiplication (Hadamard product operator), x_t is the input vector, r_t is the "reset" gate vector, h_t is the output vector, z_t is the "update" gate vector, σ is the sigmoid function, ϕ denotes the $\tanh$ function, and W, U, and b are learnable parameters. The individual cell essentially takes in an input and the output from the previous timestep and determine show much to update or reset the value depending on the learnable parameters.

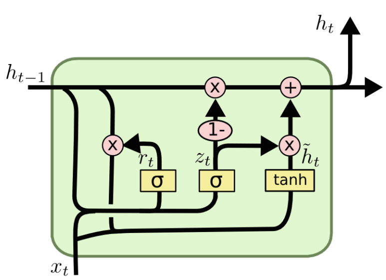

Figure 2: A schematic of the Gated Recurrence Unit cell

3. Experimental Design

3.1. Synthetic Data

The question we are trying to answer is if a deep neural network can distinguish well between pure noise, and a signal with additive noise (at a specified signal-to-noise ratio). To do this, we generate a spectrum of synthetic gravitational waveforms corresponding to binary mergers of like-mass objects in a specified range and add white (gaussian) noise so as for the resultant signal to satisfy a given average signal-to-noise ratio. For the purposes of this experiment we define the signal-to-noise ratio (SNR) as

$$SNR = \frac{P_{noise}}{P_{signal}}$$

Where P denotes the power of the signal. The power of a discrete signal is given by

$$P = \frac{1}{N} \sum_{i=1}^{N} |x_i|^2$$

The average power of a spectrum of K signals is then

$$\bar{P} = \frac{1}{K} \sum_{i=1} P_i$$

Generating white noise of a given power P_{noise} equates to generating gaussian noise with $\mu = 0$ and $\sigma^2 = P_{noise}$. Thus, to produce a desired SNR, R, given an average power $\bar{P}_{signal}$ we need to produce gaussian noise with

$$\sigma = \sqrt{R\bar{P}}$$

In addition to the synthetic waveforms with additive noise we produce an equal amount of pure noise spectra of the form

$$X \sim \mathcal{N}(0, 1)$$

Generating an equal amount of data for both the positive (waveform) and negative (noise) class avoids us having to implement measures to counteract issues associated with class imbalances.

3.2. Experiments

[1] Instances of the Gravitational-WaveNet architecture were trained and evaluated on sets of 100 waveforms generated for equally spaced masses in the range 10-50. Waveforms were generated using the PyCBC package with a sample rate of 60 and 'IMRPhenomD' approximant. Due to the nature of the physical process, waveforms generated for mergers of high mass have a shorter spatial extent than those of smaller masses. To counteract this, zero-padding was added to the beginning of waveforms shorter than 256 time steps and waveforms longer than 256 timesteps had the difference trimmed from the beginning (the low-amplitude regime). The generated noise was also 256 time steps long.

The GWN model used has 2 convolutional layers each containing 32 filters of kernel size 4, a dilution rate of 2, and a RNN layer with 32 GRU cells. Hyperparameter optimization was not performed for the sake of time but presents an additonal avenue for future experimentation.

[1] Code, data, and experiment logs available at: https://github.com/Salazar-99/GW-ML

The models were trained using the Adam optimizer with standard hyperparameters and binary cross-entropy loss given by

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))$$

Low sample rates and batch sizes were used to allow for low computation cost and rapid experimentation. All experiments and data generation took a matter of seconds on a single Intel i7 CPU. We can imagine that increasing the fidelity of the data by increasing the sample rate as well as producing a larger training set of waveforms would only increase the accuracy of any reasonable model. In this sense, this study forms a reasonable lower bound on the expected performance of deep learning models on similar problems.

4. Results

The GWN architecture performs very well across a large range of SNR values. Recall that the binary classification problem is between a stationary distribution (the standard normal gaussian) and waveforms with variable signal-to-noise ratio. Thus, due to the differences in amplitudes the classification task could be performed with a simple rules-based approach across the majority of the spectrum. For small SNR the waveforms are much “smaller” than the pure noise signals. For very large SNR, the waveforms are then much “larger” than the pure noise signals. The interesting regime is when the signal with additive noise is about the same visually as the pure noise. In this regime performance dropped from 98%+ to about 60-70%. Adding more layers seemed to help marginally. The fact that the network is able to perform at better than 50% accuracy indicates that it is able to extract some information from even the noisiest of signals.

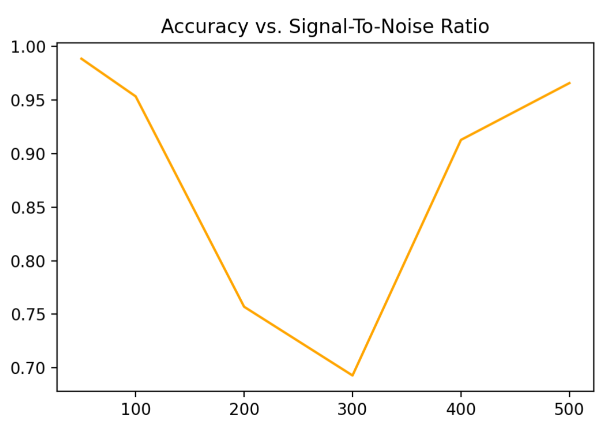

Figure 3: Largest dip in accuracy occurs in the regime of SNR: 200-400

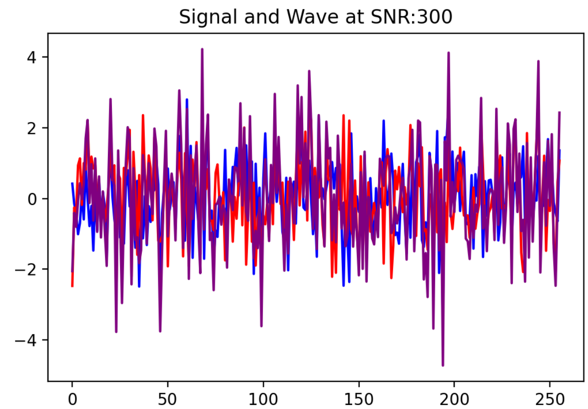

Figure 4: Signal and Wave at 300 SNR, when they are visually comparable. This marks the lowest accuracy point of the network.

5. Conclusion

Although the GWN architecture performed very well on a large range of SNRs, its accuracy fell to about 70% when the waveforms became visually similar to the noise. This is much to low to replace LIGOs current piepline but offers an interesting baseline survey for the applicability of deep learning in this problem domain. Future work could include extending the GWN architecture and experimenting on data with different kinds of noise spectra with variable power domains.

References

LIGO Collaboration. *Observation of Gravitational Waves from a Binary Black Hole Merger* (2016)

J.Chung, C. Gulcehre, K. Cho, Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling* (2014)

A. Geron. *Hands On Machine Learning with Scikit-Learn, Keras, and Tensorflow* (2019)