GA

MP

ML

CI

DEEP NEURAL NETWORKS TO ENABLE REAL-TIME MULTIMESSENGER ASTROPHYSICS

Allocation: Director Discretionary/1,000 Knh
PI: Eliu Huerta[1]
Collaborator: Daniel George[1]

[1]National Center for Supercomputing Applications

EXECUTIVE SUMMARY

The detection of gravitational waves has opened up a new spectrum of observation into the Universe. The LIGO flagship detection pipelines target a specific class of binary black holes that generate burst-like gravitational wave signals. In order to capitalize on the unique opportunities that gravitational wave astrophysics presents for new discoveries, it is necessary to extend the depth of gravitational wave searches to extract signals that currently go unnoticed with these pipelines. To address this issue, we introduce Deep Filtering: a new method that combines two deep convolutional neural networks for classification and regression to detect and characterize signals much weaker than background noise. We show that Deep Filtering significantly outperforms conventional machine learning techniques and enables the detection of new classes of gravitational wave signals that go unnoticed with existing detection algorithms.

RESEARCH CHALLENGE

Advanced LIGO (aLIGO) detection algorithms have confirmed the existence of a particular class of gravitational waves (GWs) using a 3D search: quasi-circular, spin-aligned binary black holes (BBHs). Extending these searches to target the full 8D parameter space of astrophysically motivated sources presents outstanding computational challenges [1, 2, 3].

Multimessenger searches of electromagnetic (EM) and astroparticle counterparts of GW transients rely on accurate and low-latency GW analyses, which at present take from days to months to finish. To overcome these limitations, we introduce Deep Filtering, a deep learning algorithm to directly process aLIGO data, which outperforms other machine-learning methods, and is many orders of magnitude more computationally efficient than matched filtering for both detection and parameter estimation.

METHODS & CODES

We consider a 2D parameter space that describes non-spinning BHs on quasi-circular orbits, with masses between 5 and 75 solar masses, and mass-ratios of 1 to 10. We generate our data sets using the surrogate waveform family introduced in [4]. The mass-ratio values of the BBH signals are between 1 and 10 in steps of 0.1 for training, and intermediate values for testing.

We superimposed different realizations of Gaussian white noise on top of the signals over multiple iterations, thus amplifying the size of the data sets. We then standardized the inputs to have zero mean and unit variance. The final training sets at each signal-to-noise ratio (SNR) contained about 100,000 time-series vectors produced from 4,000 templates of BBH signals by adding multiple batches of noise and shifting in time. The validation and testing sets at each SNR contained about 25,000 elements, produced from 586 clean templates and different noise realizations.

We designed simple deep neural networks (DNNs) from the ground up, since deep learning alternatives to matched filtering had not been proposed before. We tested around 80 configurations of DNNs, and found that a design with three convolutional layers followed by two fully connected layers yielded the best results. We utilized the neural network functionality in the Wolfram Language, based on the MXNet framework, which utilizes the CUDA deep learning library for acceleration with NVIDIA GPUs. We used the ADAM method as our learning algorithm.

We developed a new strategy to improve the performance and reduce the training times of the DNNs. By starting with training inputs having a high SNR of less than 16, and then gradually increasing the noise in each subsequent training session until a final SNR of 0.06, we observed that the performance of prediction can be quickly maximized for low SNRs. Our algorithm can be applied to a continuous data stream using a 1-second sliding window with offsets of 0.2 seconds.

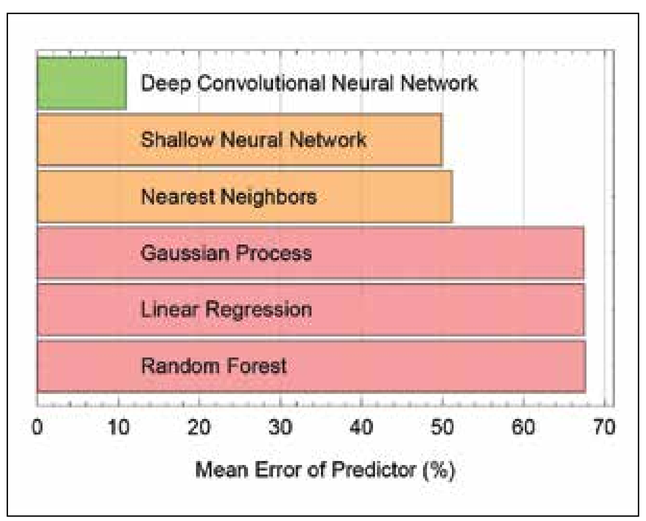

Figure 1: Mean relative error obtained by various machine learning algorithms for predicting a single parameter, i.e., mass-ratio, using a training set containing about 8,000 elements at SNR (Signal-to-Noise Ratio) = 0.36. Scaling these methods to predict multiple parameters is often difficult, whereas it simply involves adding more neurons to the final layer of neural networks.

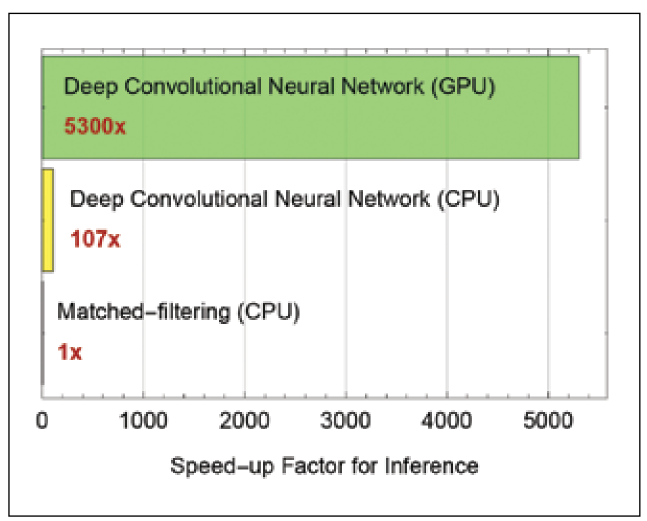

Figure 2: For a given template bank, Deep Filtering is many orders of magnitude faster than matched filtering. The evaluation time of a Deep Neural Network is constant regardless of the size of training data, whereas for matched filtering it is proportional to the size of the template bank.

RESULTS & IMPACT

We trained our classifier to achieve 100% accuracy with zero false positives for signals with a SNR > 0.36. For comparison, we trained standard implementations of all commonly used machine learning classifiers along with the DNNs on a training set of 8,000 elements. Unlike DNNs, none of these algorithms was able to directly handle raw noisy data, as shown in Fig. 1.

Our predictor successfully measured binary component masses given noisy GW signals that were not used for training, with an error of the same order as the spacing between templates for a SNR > 1. Although our initial goal was to create a pipeline for only non-spinning, quasi-circular BBH signals, we tested our DNNs using moderately eccentric simulations that we obtained using the Einstein Toolkit on the Blue Waters supercomputer. The classifier detected all these signals with nearly the same rate as the original test set (with over 99.7% mean accuracy for SNR > 0.36 and 100% accuracy at SNR > 0.5). The predictor was able to estimate the component masses of our eccentric simulations for SNR > 0.25 with a mean relative error less than 13%, 19%, 32%, and 34% for mass-ratio 1, 2, 3, and 4, respectively. This result is very encouraging, since these types of signals go unnoticed with aLIGO detection [5].

Our DNNs are only 2MB in size each. The average time taken for evaluating them per input of 1 second duration is approximately 6.7 milliseconds and 135 microseconds using a single CPU or GPU, respectively. For comparison, we estimated the evaluation time for time-domain matched filtering with the template bank of clean signals used for training; the results are shown in Fig. 2. Our extremely fast inference rate indicates that real-time analysis can be carried out with a single computer even with DNNs that are significantly larger, and can be trained over much bigger template banks of signals.

WHY BLUE WATERS

Blue Waters enabled us to create a large catalog of eccentric numerical relativity simulations, which required thousands of node hours that we ran in parallel to sample a deep region of parameter space. No other resource but Blue Waters can provide the required computational power to obtain a numerical relativity catalog of this nature in a timely manner.