Deep Learning For Gravitational wave detection

Ching-Yao Tang

Department of Physics, NDHU

I. Introduction

Laser Interferometer Gravitational Wave Observatory(LIGO) had first detected the gravitational wave(GW)[1] in 2015. Currently, low-latency detection algorithm is based on matched-filtering(MF) method[2]. However, the larger database slows down the execution speed of MF.

In this poster, the method of Deep Learning(DL)[3], which is based on 2D Convolutional Neural Network(CNN), is proposed to speed up the process of identification and parameter estimation(PE) of the GW detection.

II. Method

This poster considers the GW signal from BBH mergers only. There are two networks with a similar structure to implement two different tasks, which are identification and PE.

The noisy time-series data(1D) had converted into the spectrogram(2D) by Short Time Fourier Transform(STFT), before applying CNN.

1. Assumption

Assume the GW signals are optimally oriented respect to the detector. Therefore, the parameter space reduces to two dimensions, i.e., respective BH masses.

Assume there are no glitches existed so that after whitening with the LIGO power spectrum density(PSD), shown in Fig. 1, the noise in the data should remain white noise. There are three examples shown in Fig. 2.

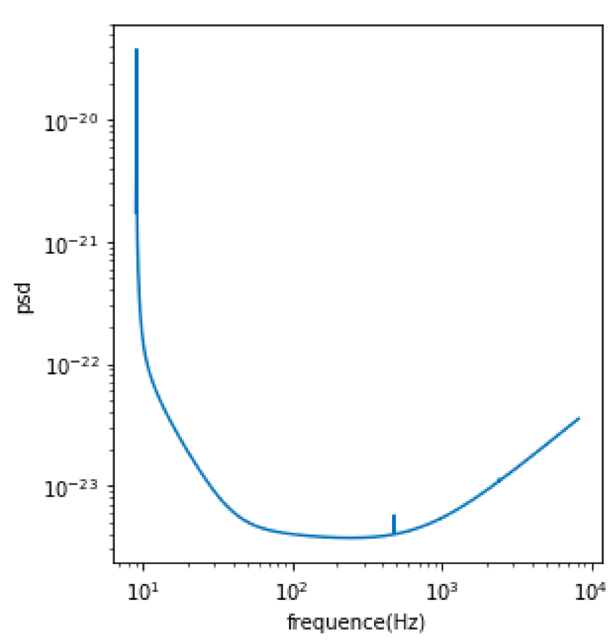
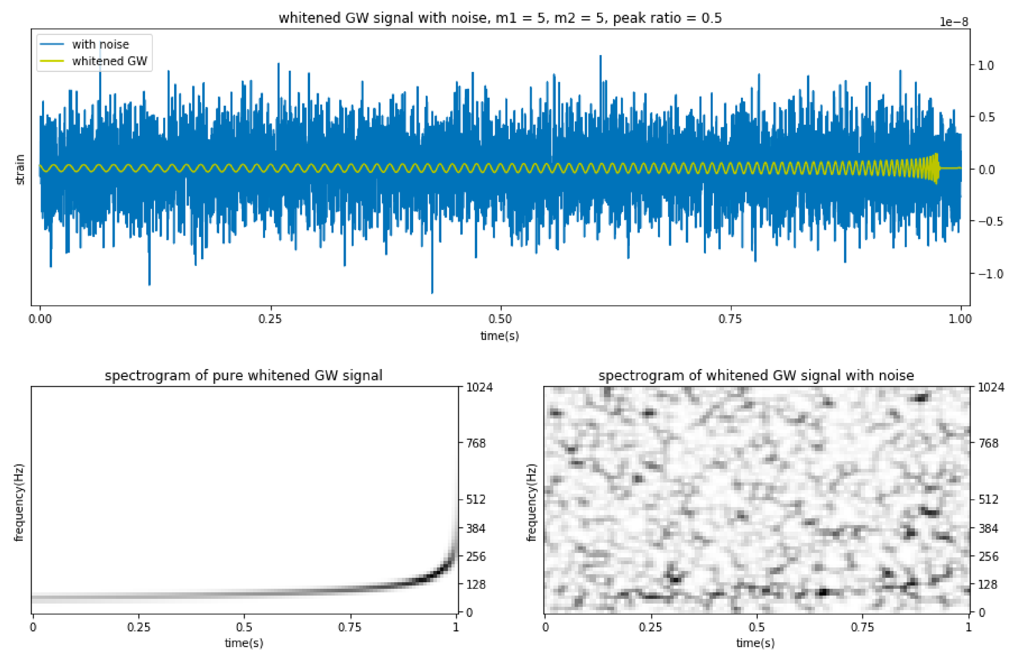


Fig. 1: *aLIGOs PSD at the zero-detuned high-power design sensitivity. The PSD is used for whitening data.*

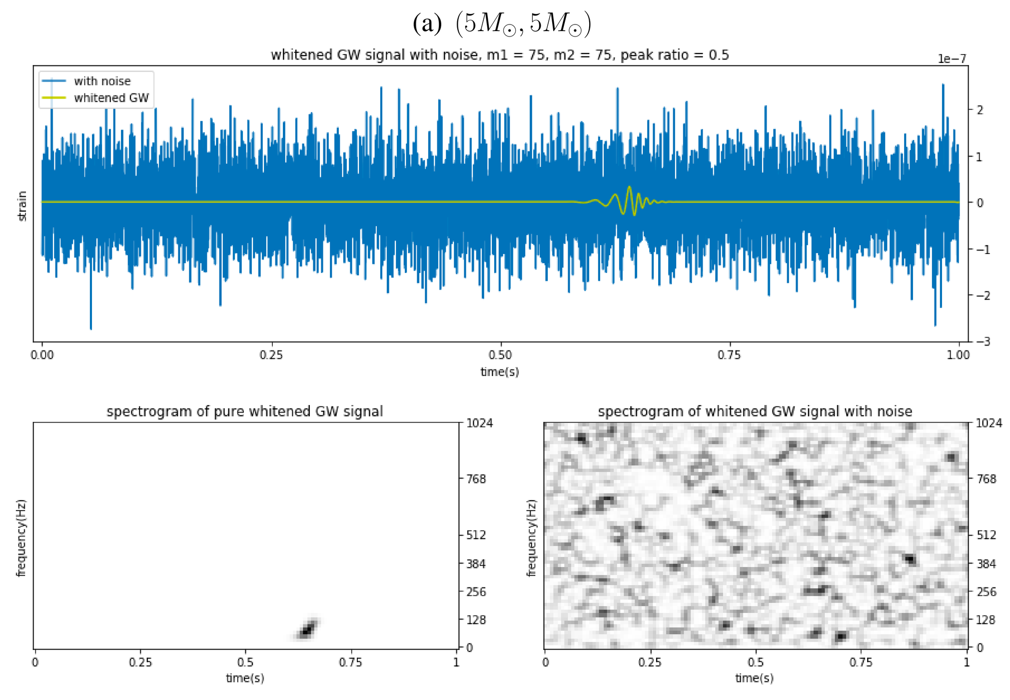
2. Architecture of CNN

The best-performance CNN model is based on residual learning[4, 5], called "resnet4", shown in Fig. 3.

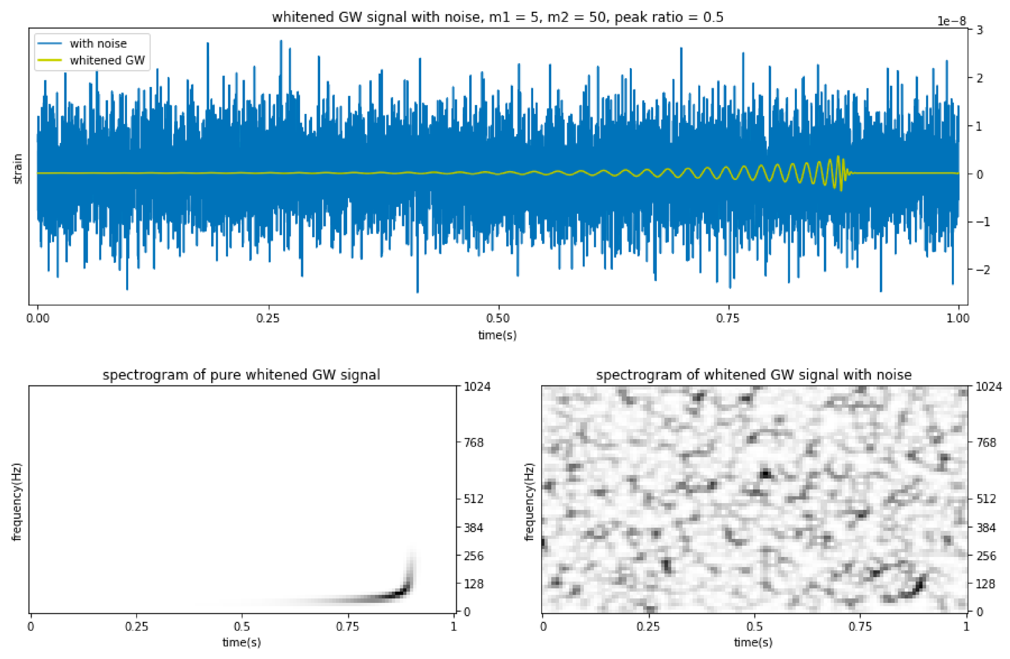
The identification model outputs two probability for the existence of the GW signal. And the PE model outputs the predictive mass for two BH, respectively.



(a) $(5M_{\odot}, 5M_{\odot})$



(b) $(75M_{\odot}, 75M_{\odot})$



(c) $(5M_{\odot}, 50M_{\odot})$

Fig. 2: *Three different mass combination examples of simulated GW data. Each subfigure contain three different view of the same data, described as following. Top: Whitened GW signal with Gaussian white noise in Time-series data. Lower left: Spectrogram of pure GW signal. Lower right: Spectrogram with Gaussian white noise.*

Number	Layer Name	Output Shape
1	Input	Matrix (size:[33,121])
2	Reshape	Matrix (size:[33,121,1])
3	Convolution Layer (filter=14*14)	Matrix (size:[33,121,3])
4	ReLU	Matrix (size:[33,121,3])
5	Convolution Layer (filter=11*11)	Matrix (size:[33,121,3])
6	Shortcut Connection (from output 2)	Matrix (size:[33,121,3])
7	ReLU	Matrix (size:[33,121,3])
8	Maxpooling (kernel=2*2, stride=2)	Matrix (size:[17,61,3])
9	Convolution Layer (filter=8*8)	Matrix (size:[17,61,5])
10	ReLU	Matrix (size:[17,61,5])
11	Convolution Layer (filter=5*5)	Matrix (size:[17,61,7])
12	Shortcut Connection (form output 8)	Matrix (size:[17,61,7])
13	ReLU	Matrix (size:[17,61,7])
14	Flatten	Vector (size:[5185,])
15	Fully Connected Layer	Vector (size:[128,])
16	ReLU	Vector (size:[128,])
17	Fully Connected Layer	Vector (size:[32,])
18	ReLU	Vector (size:[32,])
19	Output	Vector (size:[2,])

Fig. 3: *Architecture of resnet4. It's used for both identification and PE*

III. Results

The identification of the resnet4 on parameter space has shown in Fig. 4. The testing results in different SNRs have shown in Fig. 5 (a) to (c). For SNR $\geq$ 8.7, resnet4 has perfect sensitivity. The overall testing results for the data that contain SNR lie between 5 and 15 are accuracy = 98.75%, sensitivity = 98.3% and false alarm rate = 0.8%. The average prediction error of the PE is around 15%, shown in Fig.5 (d).

For both of training and testing were completed via Nvidia Tesla P100 GPU. It can identify a single data in 133 microseconds.

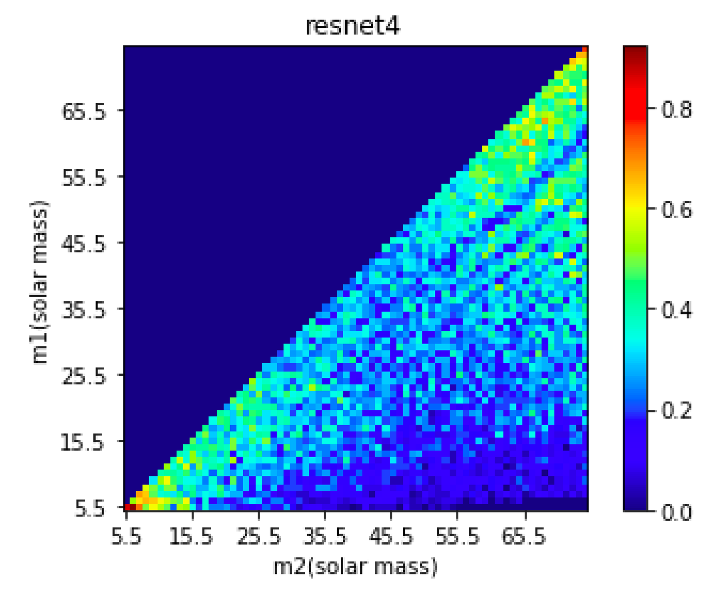
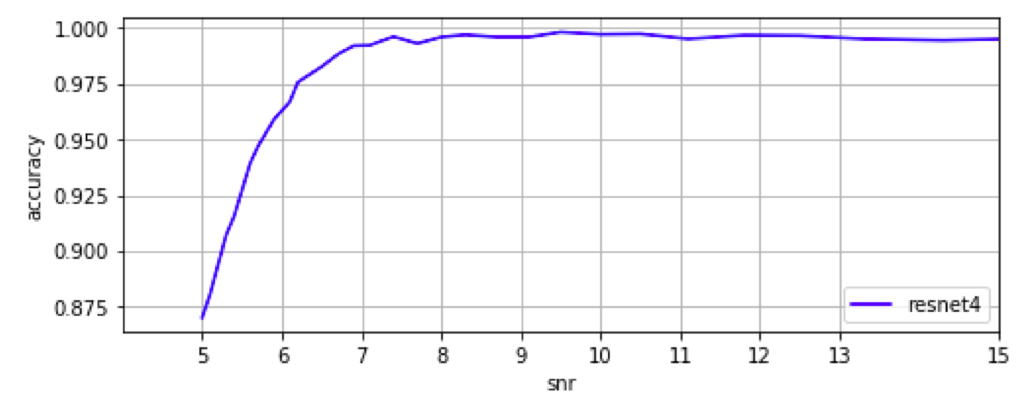
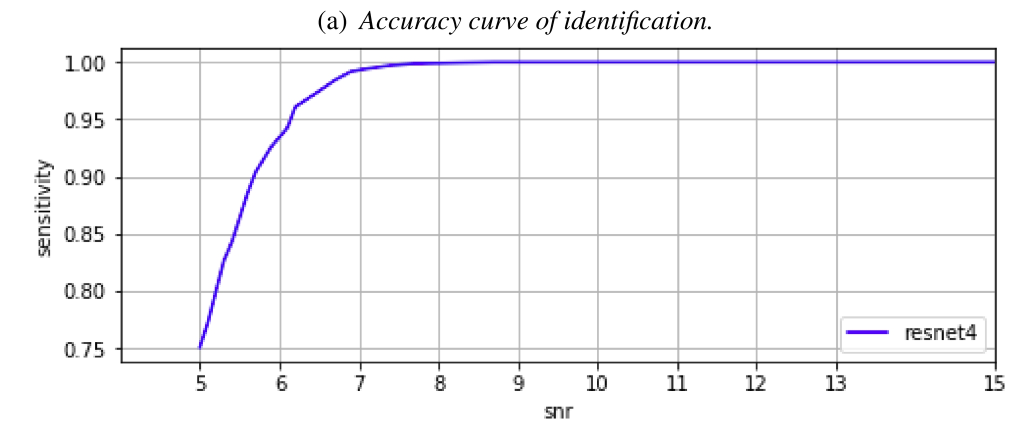


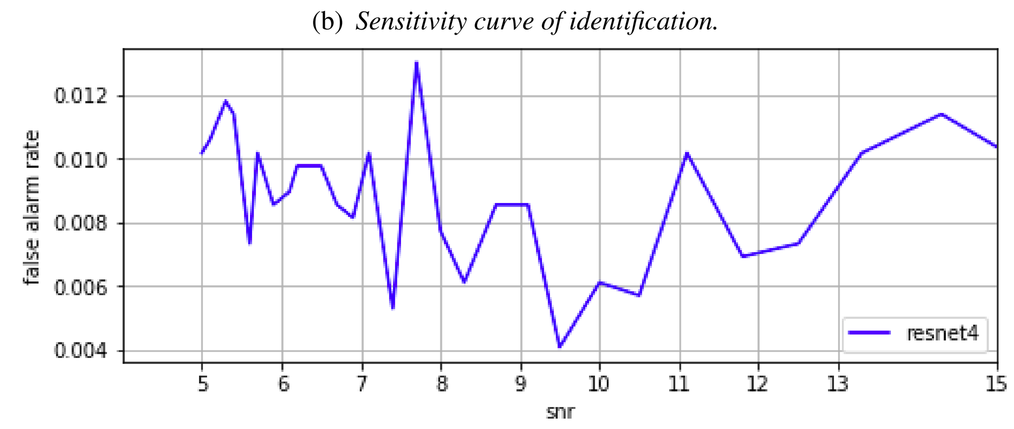
Fig. 4: *Testing error rate for identification in parameter space. The SNR here is fixed at 5.*



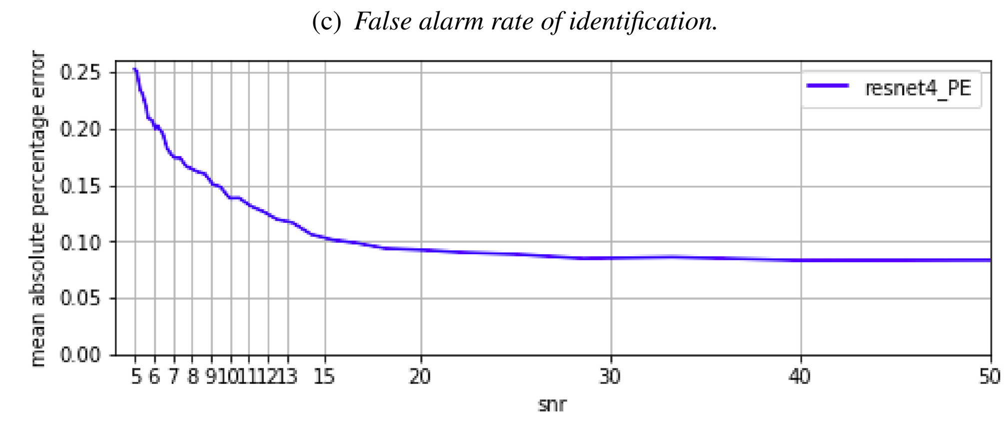
(a) *Accuracy curve of identification.*



(b) *Sensitivity curve of identification.*



(c) *False alarm rate of identification.*



(d) *mean absolute error of parameter estimation.*

Fig. 5: *Several testing result related with SNR for resnet4.*

IV. Conclusions

For the identification part, compared with the 1D CNN method in [6, 7], my 2D CNN method performs slightly better. Furthermore, both CNN methods have similar sensitivity with MF. However, the 2D method takes extra time to transform the time-series data to the spectrogram. For the PE part, the 1D method performs better than the 2D method.

References

[1] B.P.Abbott et al. *Phys. Rev. Lett. 116, 061102*, 2016.

[2] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton. *Phys. Rev. D 85, 122006*, 2012.

[3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun. *arXiv:1512.03385*, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun. *arXiv:1603.05027*, 2016.

[6] D. George and E. A. Huerta. *Phys. Rev. D 97, 044039*, 2018.

[7] D. George and E. A. Huerta. *Physics Letters B, 778*, 2018.