Frontiers of Physics
https://doi.org/10.1007/s11467-019-0936-x
Front. Phys.
15(1), 14601 (2020)

RESEARCH ARTICLE

Extraction of gravitational wave signals with optimized convolutional neural network

Hua-Mei Luo[1,†], **Wenbin Lin**[2,3,‡], **Zu-Cheng Chen**[4,5,§], **Qing-Guo Huang**[4,5,¶]

[1] *School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China*
[2] *School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China*
[3] *School of Mathematics and Physics, University of South China, Hengyang 421001, China*
[4] *CAS Key Laboratory of Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[5] *School of Physical Sciences, University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing 100049, China*
E-mail: [†] *972660976@qq.com,* [‡] *lwb@usc.edu.cn,* [§] *chenzucheng@itp.ac.cn,* [¶] *huangqg@itp.ac.cn*
Received June 20, 2019; accepted October 14, 2019

Gabbard *et al.* have demonstrated that convolutional neural networks can achieve the sensitivity of matched filtering in the recognization of the gravitational-wave signals with high efficiency [*Phys. Rev. Lett.* 120, 141103 (2018)]. In this work we show that their model can be optimized for better accuracy. The convolutional neural networks typically have alternating convolutional layers and max pooling layers, followed by a small number of fully connected layers. We increase the stride in the max pooling layer by 1, followed by a dropout layer to alleviate overfitting in the original model. We find that these optimizations can effectively increase the area under the receiver operating characteristic curve for various tests on the same dataset.

Keywords gravitational wave, convolutional neural networks, deep learning

1 Introduction

Ten binary black hole (BBH) coalescences were detected during the first (O1) and second (O2) observing runs of LIGO/Virgo [1–8], which has led us to the era of gravitational-wave (GW) astronomy [9], as well as the era of multi-messenger astronomy [10]. LIGO/Virgo third (O3) observing run has already begun, and more GW events are expected to be observed. Developing fast and robust analysis methods to extract GW signals from data will be urgent as the improvement of detectors' sensitivities in the upcoming years. Currently, the standard method for detecting transient GW signals is the matched filtering technique. Although it works very well in the extraction of weak signals, a huge amount of computational cost is required to process the data which might contain the GW signals. This makes the matched filtering technique unsuitable for the future multi-messenger observations [11].

Deep learning is a new direction of machine learning. Its motivation is to establish and simulate the neural network of human brain for analytical learning. In recent years, deep learning has developed rapidly in the fields of image processing [12–14], medical diagnosis [15], and signal recognition [16]. Deep learning has also been applied to process the data containing the GW signals [17–27]. For example, Fan *et al.* showed that deep neural network (DNN) can effectively recognize the presence of GW signals and estimate the corresponding source space parameters, including the luminosity distance, right ascension, and declination of the compact binary star mergers [25]. Specifically, Gabbard *et al.* demonstrated that convolutional neural networks (CNN) can achieve the sensitivity of matched filtering [23]. They are generous to provide their codes on the github [28] so that we can follow and reproduce their results. We find that the fitting effect on the training set is much higher than the generalization effect on the test set in their model, which implies that there may exist an overfitting issue. In this work, we show that their model can be optimized to achieve a better accuracy.

The rest of this paper is organized as follows. Section 2 introduces the data preparation. Section 3 and Section 4 present the optimized model and the corresponding results. Summary is given in Section 5.

2 Data preparation

The data we use are the same as those given by [23]. Here we only consider Gaussian noise, and we use the same random seeds as those adopted by Gabbard *et al.* The application of deep learning on GW data with non-Gaussian noise can be found in Refs. [18, 29–33]. We only distin-

guish the two cases: BBH merger signals plus Gaussian noise, and Gaussian noise only. As was done in Ref. [23], we whiten the simulated GW time series with the detector noise's power spectral density (PSD) corresponding to the Advanced LIGO design sensitivity [34] to rescale the noise contribution at each frequency for equal power. Besides, a Turkey window function is chosen to truncate signal data.

In the simulations of the GW signals, we use the IMRPhenomD-type waveform [35, 36] which has been implemented in the `LALSuite` library [37]. The masses of black hole (BH) vary from 5 to $95M_{\odot}$, which are drawn from an astrophysically-motivated distribution where $m_1 > m_2$ and $m_{1,2} \sim \log m_{1,2}$ [6]. Here, m_1 and m_2 represent the masses of the primary and secondary BHs in a binary system. Similar to Ref. [23], we ignore the spins of BHs, and assume that the right ascension and declination follow an isotropic distribution on the sky. The polarization angle and phase are drawn from a uniform prior distribution on the range $[0,\ 2\pi]$. The cosine of the inclination angle is drawn from a uniform prior distribution on the range $[-1, 1]$. The waveforms are then randomly placed in the time series such that the peak amplitude of each waveform is randomly located within the fractional range $[0.75, 0.95]$ of the time series. All parameters are drawn from the same distributions as in Ref. [23].

For the training set, the number of noise implementations for each signal is chosen as 25. For the validation set and the test set, this number is chosen as 1. We also set different random seeds when generating these three sets. The waveform amplitude is scaled to achieve a predefined optimal signal-to-noise ratio (SNR), ρ_{opt}, which is defined as

$$\rho_{opt}^2 = 4\int \frac{|h(f)|^2}{S_n(f)}\mathrm{d}f, \tag{1}$$

where $h(f)$ is the GW strain in the frequency domain, $S_n(f)$ is the PSD of detector noise. The amplitude of a Gaussian noise being added into the time series is given by

$$A(f) = \sqrt{a \cdot T_{obs} \cdot S_n(f)}, \tag{2}$$

where T_{obs} is the observation duration, a is a free parameter which will be tuned to realize different noise configurations. The simulated time series are sampled at 8192 Hz and last for 1 s. In Fig. 1, we show a representative time series of the data sets used to train, validate and test the optimized CNN. Class 0 contains Gaussian noise only, while Class 1 represents the same Gaussian noise plus a GW signal.

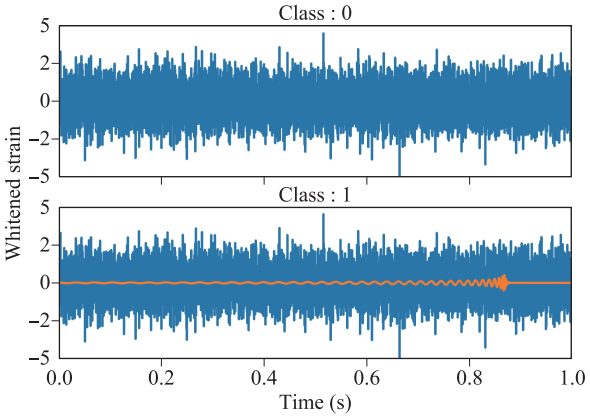

Fig. 1 A representative time series of the data sets used to train, validate and test the optimized CNN. Class 0 represents a whitened time series of Gaussian noise (blue), and Class 1 represents the same Gaussian noise plus a GW signal (orange) for the BBH with $m_1 = 17.10M_{\odot}$, $m_2 = 6.98M_{\odot}$, and $\rho_{opt} = 6$.

3 CNN model and optimization

Deep learning can be categorized into supervised learning and unsupervised learning. A major difference between supervised learning and unsupervised learning is that only the former needs label information in the training samples. Gabbard *et al.* [23] proposed a CNN for extracting the GW signals and achieved a similar performance on the Receiver–Operator Characteristic (ROC) as the matched filtering technique did. The CNN belongs to the supervised learning. Three data sets including a training set, a validation set and a test set are needed to accomplish the supervised learning. The parameters (such as the weight of the connections between neurons) of the CNN model are initially fitted on the training set, and then further tuned on the validation set [38]. Finally, the test set is used to assess the performance of a specified model [39]. The CNN consists of an input and an output layer, as well as one or multiple hidden layers. The structures of the CNN are built with the procedures – alternating the convolution layers and the max-pooling layers followed by a small number of the fully-connected layers. Convolution layers carry on the convolution operations – making the inner products of the input value and the weight of the filter. More details on the convolution operation can be found in Ref. [40].

In Ref. [23], the Nadam adaptive learning rate optimization is employed, which can be written as [41]

$$\begin{aligned}
g_t &= \bigtriangledown_{\theta_{t-1}} J(\theta_{t-1}),\\
\hat{g}_t &= \frac{g_t}{1-\prod_{i=1}^{t}\mu_i},\\
m_t &= \mu_t m_{t-1} + (1-\mu_t) g_t,\\
\hat{m}_t &= \frac{m_t}{1-\prod_{i=1}^{t+1}\mu_i},\\
n_t &= \nu n_{t-1} + (1-\nu) g_t^2,\\
\hat{n}_t &= \frac{n_t}{1-\nu^t},\\
\bar{m}_t &= (1-\mu_t)\hat{g}_t + \mu_{t+1}\hat{m}_t,\\
\theta_t &= \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{n}_t}+\varepsilon},
\end{aligned} \tag{3}$$

where $J(\theta)$ is the target loss function. g_t is the gradient of $J(\theta_{t-1})$ with respect to θ_{t-1}. m_t and n_t are the first and the second moment estimates of the gradient, respectively, which can be regarded as the estimates for the expectations of $|g_t|$ and $|g_t^2|$. $\hat{m}_t$ and $\hat{n}_t$ are the corrected m_t and n_t respectively, which can be approximated as the unbiased estimates of the expectations. μ and ν are the attenuation factors. η is learning rate, and ε is a small number used to ensure the denominator to be non-zero.

The nonlinearity of the CNN network is mainly due to the use of activation functions. Two types of activation functions are used in Ref. [23]. One is the exponential linear unit (Elu) function used in the convolutional layer and the hidden layer, and the other is the Softmax (SMax) function used in the output layer. The Elu function is expressed as

$$f(x) = \begin{cases} x, & x \geq 0, \\ \alpha(\mathrm{e}^x - 1), & x < 0, \end{cases} \tag{4}$$

where $\alpha > 0$. Compared to Sigmoid and the rectified linear unit (Rlu), the Elu activation function has soft saturation on the left side, which makes the Elu more robust to input changes or noise, and its linearity on the right side ensures that the Elu can alleviate the gradient disappearance. In addition, the output average of the Elu is close to zero, so the convergence speed is very fast.

The output layer uses the SMax function to perform the binary classification process. The output of the two neurons is mapped to the vector in the $[0, 1]$ interval, and the component with the value greater than 0.5 is taken as the input sample category. The `Keras` package [42] is employed to realize the CNN model.

The CNN model built in Ref. [23] achieves a good performance, but there still exist an overfitting issue. Based on their model, we make some modifications as follows. First, we add a dropout layer after the max pooling layer to optimize their model. Dropout drops units (along with their connections) randomly from the neural network during training. This prevents the units from co-adapting too much [43]. Second, it has been demonstrated that the max pooling can be replaced simply by a convolution layer with an increased step size (the amount of movement between the applications of the filter to the input) or the step of the max pooling can be increased to 2 without loss of precision on the data sets (CIFAR-10, CIFAR-100, ImageNet) [44]. The max pooling extracts several feature values from a filter, and only takes the largest one of them as a reserved value. All other feature values are discarded. Therefore, we also increase the stride of the max pooling and the corresponding convolution layer to 2 in the optimized model. All modifications can alleviate the overfitting issue.

In this work, we generate 5×10^5 independent time series samples, of which 80% are used for training, 10% for validation, and 10% for testing. Throughout this paper, we choose $\mu = 0.9$, $\nu = 0.999$, $\varepsilon = 10^{-8}$ and $\eta = 0.002$.

Table 1 The structure of the optimized CNN model. The parts being different from those of the original CNN model [23] are shown in red color, where "C", "H" and "n/a" represent the convolutional layer, the hidden layer and the not-applicable case, respectively.

Parameter	Layer								
(Option)	1	2	3	4	5	6	7	8	9
Type	C	C	C	C	C	C	H	H	H
No. Neurons	8	8	16	16	32	32	64	64	2
Filter size	64	32	32	16	16	16	n/a	n/a	n/a
Max pool size	n/a	8	n/a	6	n/a	4	n/a	n/a	n/a
Dropout	0	0	0	0.4	0	0.3	0.5	0.5	0
Max pool stride size	n/a	2	n/a	2	n/a	2	n/a	n/a	n/a
Activation function	Elu	Elu	Elu	Elu	Elu	Elu	Elu	Elu	SMax

Table 2 The sizes of the input data at different layers, where "Flatten" stands for transforming a matrix into a one-dimensional vector.

	Input	Vector (size: 8192)
1	Reshape	matrix (size: 1×8192)
2	Convolution	matrix (size: 8×8129)
3	Convolution	matrix (size: 8×4049)
4	Max pool size	matrix (size: 8×2021)
5	Convolution	matrix (size: 16×1990)
6	Convolution	matrix (size: 16×988)
7	Max pool size	matrix (size: 16×492)
8	Dropout	matrix (size: 16×492)
9	Convolution	matrix (size: 32×477)
10	Convolution	matrix (size: 32×231)
11	Max pool size	matrix (size: 32×114)
12	Dropout	matrix (size: 32×114)
13	Flatten	vector (size: 3648)
14	Hidden	vector (size: 64)
15	Dropout	vector (size: 64)
16	Hidden	vector (size: 64)
17	Dropout	vector (size: 64)
18	Hidden	vector (size: 2)
	Output	vector (size: 2)

The structure of the optimized CNN model used in this work is shown in Table 1. It is worth noting that we only increase the convolution-layer stride corresponding to the max pooling layer to 2. The sizes of the input data at each layer are given in Table 2.

4 Results

In order to evaluate the performance of the optimized CNN model, we compare the ROC of the optimized CNN model with that of the original CNN model [23] on the

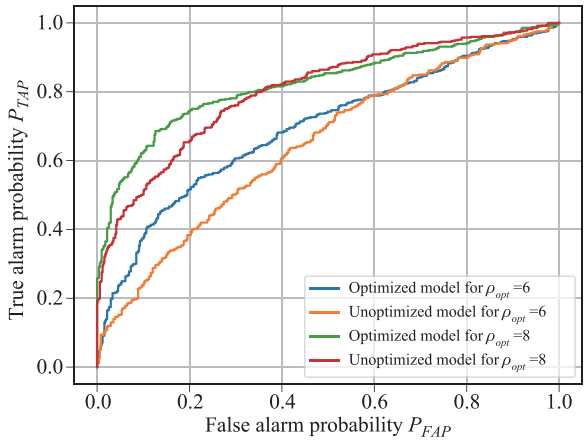

Fig. 2 The ROC curves, where the Gaussian noise amplitude parameter is set as $a = 0.25$.

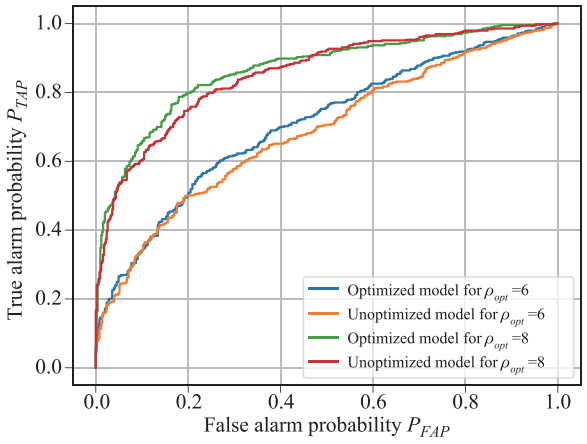

Fig. 4 The ROC curves, where the Gaussian noise amplitude parameter is set as $a = 0.2$.

simulated data. In the ROC plot, the abscissa is the false alarm probability (P_{FAP}) and the ordinate is the true alarm probability (P_{TAP}), which are defined as

$$P_{FAP} = \frac{N_{FP}}{N_{FP} + N_{TN}}, \tag{5}$$

$$P_{TAP} = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{6}$$

where N_{TP}, N_{FP}, N_{TN}, and N_{FN} represent the sample numbers of true positive, false positive, true negative and false negative, respectively.

The area enclosed by the ROC curve and the coordinate axis is defined as Area Under Curve (AUC), and the AUC value of the classifier is equivalent to the probability that the randomly selected positive samples are sorted before the randomly selected negative samples [45]. The larger the AUC, the better the classifier classification effect is.

Figure 2 shows the comparisons of the ROC curves for different ρ_{opt} when $a = 0.25$. The AUC values are 0.69465 for $\rho_{opt} = 6$ and 0.82346 for $\rho_{opt} = 8$ in the optimized CNN model, in contrast to 0.64665 and 0.80422 in the original CNN model. They are increased by 0.04800 for $\rho_{opt} = 6$ and 0.01924 for $\rho_{opt} = 8$, respectively.

In order to check the robustness of the optimized network to noise, we changed the amplitudes of the Gaussian noise, and the corresponding ROC curves are shown in Fig. 3 and Fig. 4, respectively. It can be seen that the AUC values have also been improved.

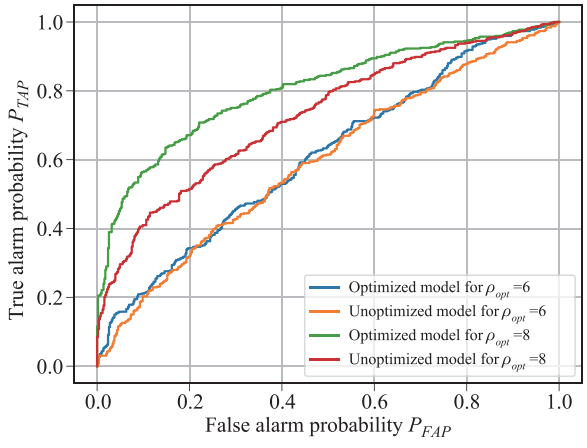

Fig. 3 The ROC curves, where the Gaussian noise amplitude parameter is set as $a = 0.3$.

5 Summary

The successful detections of GWs not only confirm the classical prediction of general relativity, but also open a new window of GW astronomy for exploring the universe. At present, the matched-filtering method is used in the identification of real GW signals. However, matched filtering is time-consuming. This shortcoming limits its ability and potential in the future detections of GWs. In recent years, deep learning has been greatly developed in data processing, and the classical CNN has been demonstrated to achieve the sensitivity of matched filtering with a high efficiency in the recognization of the GW signals [23]. In this work we make optimizations for the CNN model proposed by Gabbard *et al.* [23]. The simulation results show that the optimized CNN improves the AUC values by 0.04800 for $\rho_{opt} = 6$ and 0.01924 for $\rho_{opt} = 8$ when $a = 0.25$, without increasing the running time. Our work further illustrates the advantages of deep learning in the retrieval of the GW signals.

Acknowledgements We thank the reviewers for providing constructive comments and suggestions to improve the quality of this paper. W. L. was supported by grants from NSFC (Grant Nos. 11647314 and 11847307). Q. G. H. was supported by grants from NSFC (Grant Nos. 11690021, 11575271, and 11747601), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant Nos. XDB23000000 and XDA15020701), as well as Top-

Notch Young Talents Program of China. This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (https://www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration.

References and notes

1. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), Observation of gravitational waves from a binary black hole merger, *Phys. Rev. Lett.* 116, 061102 (2016), arXiv: 1602.03837 [gr-qc]
2. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GW151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence, *Phys. Rev. Lett.* 116, 241103 (2016), arXiv: 1606.04855 [gr-qc]
3. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GW170104: Observation of a 50-solar mass binary black hole coalescence at redshift 0.2, *Phys. Rev. Lett.* 118, 221101 (2017), [Erratum: *Phys. Rev. Lett.* 121 (12), 129901 (2018)], arXiv: 1706.01812 [gr-qc]
4. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GW170608: Observation of a 19-solar-mass Binary Black Hole Coalescence, *Astrophys. J.* 851, L35 (2017), arXiv: 1711.05578 [astroph.HE]
5. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GW170814: A three-detector observation of gravitational waves from a binary black hole coalescence, *Phys. Rev. Lett.* 119, 141101 (2017), arXiv: 1709.09660 [gr-qc]
6. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), Binary black hole mergers in the first advanced LIGO observing run, *Phys. Rev. X* 6, 041015 (2016) [erratum: *Phys. Rev. X* 8, 039903 (2018)], arXiv: 1606.04856 [gr-qc]
7. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs, arXiv: 1811.12907 [astro-ph.HE] (2018)
8. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, *Phys. Rev. Lett.* 119, 161101 (2017), arXiv: 1710.05832 [gr-qc]
9. G. González, A. Viceré, and L. Wen, Gravitational wave astronomy, *Front. Phys.* 8(6), 771 (2013)
10. B. Zhang, The delay time of gravitational wave – gamma-ray burst associations, *Front. Phys.* 14, 64402 (2019), arXiv: 1905.00781 [astro-ph.HE]
11. B. P. Abbott (LIGO Scientific Collaboration, Virgo Collaboration, KAGRA Collaboration), Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA, *Living Rev. Relativ.* 21(1), 3 (2018), arXiv :1304.0670 [gr-qc]
12. R. Zhang, P. Isola, and A. A. Efros, Colorful image colorization, arXiv: 1603.08511 (2016)
13. A. Karpathy, A. Joulin, and F. F. Li, Deep fragment embeddings for bidirectional image sentence mapping, arXiv: 1406.5679 (2014)
14. A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Commun. ACM* 60(6), 84 (2017)
15. I. Kononenko, Machine learning for medical diagnosis: History, state of the art and perspective, *Artif. Intell. Med.* 23(1), 89 (2001)
16. T. Gebhard, N. Kilbertus, G. Parascandolo, I. Harry, and B. Schölkopf, Convwave: Searching for gravitational waves with fully convolutional neural nets, in: Workshop on Deep Learning for Physical Sciences (DLPS) at the 31st Conference on Neural Information Processing Systems (2017)
17. D. George and E. A. Huerta, Deep neural networks to enable real-time multimessenger astrophysics, *Phys. Rev. D* 97, 044039 (2018), arXiv: 1701.00008 [astro-ph.IM]
18. D. George, H. Y. Shen, and E. A. Huerta, Deep transfer learning: A new deep learning glitch classification method for advanced LIGO, *Phys. Rev. D* 97, 101501(R) (2018), arXiv: 1706.07446 [gr-qc]
19. D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data, *Phys. Lett. B* 778, 64 (2018), arXiv: 1711.03121 [gr-qc]
20. D. George, H. Y. Shen, and E. A. Huerta, Glitch classification and clustering for LIGO with deep transfer learning, *Phys. Rev. D* 97, 101501 (2018), arXiv: 1711.07468 [astroph.IM]
21. D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation with LIGO data, in: NiPS Summer School 2017 Gubbio, Perugia, Italy, June 30–July 3, 2017 (2017), arXiv: 1711.07966 [gr-qc]
22. H. Y. Shen, D. George, E. A. Huerta, and Z. Z. Zhao, Denoising gravitational waves using deep learning with recurrent denoising autoencoders, arXiv: 1711.09919 [gr-qc] (2017)
23. H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Matching matched filtering with deep networks for gravitational-wave astronomy, *Phys. Rev. Lett.* 120, 141103 (2018), arXiv: 1712.06041 [astro-ph.IM]
24. X. R. Li, W. L. Yu, and X. L. Fan, A method of detecting gravitational wave based on time-frequency analysis and convolutional neural networks, arXiv: 1712.00356 [astro-ph.IM] (2017)
25. X. L. Fan, J. Li, X. Li, Y. H. Zhong, and J. W. Cao, Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors, *Sci. China Phys. Mech. Astron.* 62, 969512 (2019), arXiv: 1811.01380 [astro-ph.IM]
26. Z. J. Cao, H. Wang, and J. Y. Zhu, Initial study on the application of deep learning to the gravitational wave data analysis, *J. Henan Norm. Univ. (Nat. Sci. Ed.)* 46, 26 (2018)

27. T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Convolutional neural networks: A magic bullet for gravitational-wave detection? arXiv: 1904.08693 [astro-ph.IM] (2019)
28. https://github.com/mj-will/intro2ml/blob/master/bbh-example.ipynb
29. R. Biswas, L. Blackburn, J. Cao, R. Essick, K. A. Hodge, E. Katsavounidis, K. Kim, Y. M. Kim, E. O. Le Bigot, C. H. Lee, J. J. Oh, S. H. Oh, E. J. Son, Y. Tao, R. Vaulin, and X. Wang, Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data, *Phys. Rev. D* 88(6), 062003 (2013), arXiv: 1303.6984 [astro-ph.IM]
30. D. George, H. Y. Shen, and E. A. Huerta, Glitch classification and clustering for LIGO with deep transfer learning, in: NiPS Summer School 2017 Gubbio, Perugia, Italy, June 30-July 3, 2017 (2017), arXiv: 1711.07468 [astro-ph.IM]
31. M. Cavaglia, K. Staats, and T. Gill, Finding the origin of noise transients in LIGO data with machine learning, *Commun. Comput. Phys.* 25, 963 (2019), arXiv: 1812.05225 [physics.data-an]
32. S. B. Coughlin, et al., Classifying the unknown: discovering novel gravitational-wave detector glitches using similarity learning, *Phys. Rev. D* 99, 082002 (2019), arXiv: 1903.04058 [astroph.IM]
33. M. Llorens-Monteagudo, T. F. Alejandro, J. Font, and A. Marquina, Classification of gravitational-wave glitches via dictionary learning, *Class. Quant. Grav.* 36, 075005 (2019), arXiv: 1811.03867 [astro-ph.IM]
34. B. P. Abbott, et al. (LIGO Scientific Collaboration, Virgo Collaboration, KAGRA Collaboration), Prospects for observing and localizing gravitational-wave transients with advanced LIGO, Advanced Virgo and KAGRA, *Living Rev. Rel.* 21, 3 (2018), arXiv: 1304.0670 [gr-qc]
35. S. Husa, S. Khan, M. Hannam, M. Prrer, F. Ohme, X. J. Forteza, and A. Boh, Frequency-domain gravitational waves from nonprecessing black-hole binaries (I): New numerical waveforms and anatomy of the signal, *Phys. Rev. D* 93, 044006 (2016), arXiv: 1508.07250 [gr-qc]
36. S. Khan, S. Husa, M. Hannam, F. Ohme, M. Prrer, X. J. Forteza, and A. Boh, Frequency-domain gravitational waves from nonprecessing black-hole binaries (II): A phenomenological model for the advanced detector era, *Phys. Rev. D* 93, 044007 (2016), arXiv: 1508.07253 [gr-qc]
37. LIGO Scientific Collaboration, LIGO Algorithm Library - LALSuite, free software (GPL) (2018)
38. G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: With Applications in *R*, Springer Texts in Statistics, Springer New York, 2013
39. B. D. Ripley and N. L. Hjort, Pattern Recognition and Neural Networks, 1st Ed., Cambridge University Press, New York, NY, USA, 1995
40. M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, arXiv: 1311.2901 (2013)
41. T. Dozat, Incorporating nesterov momentum into Adam (2015)
42. F. Chollet, et al., Keras, https://keras.io (2015)
43. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15, 1929 (2014)
44. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, Striving for simplicity: The all convolutional net, arXiv: 1412.6806 (2014)
45. T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.* 27(8), 861 (2006)