Real-Time Detection of Gravitational Waves from Binary Neutron Stars using Artificial Neural Networks

Preprint · August 2019

CITATIONS
0

READS
113

1 author:

Plamen Krastev
Harvard University
60 PUBLICATIONS **750** CITATIONS

SEE PROFILE

Real-Time Detection of Gravitational Waves from Binary Neutron Stars using Artificial Neural Networks

Plamen G. Krastev[1*]

[1]Harvard University, Faculty of Arts and Sciences, Research Computing, 38 Oxford Street, Cambridge, MA 02138, U.S.A

The groundbreaking discoveries of gravitational waves from binary black-hole mergers [1–3] and, most recently, coalescing neutron stars [4] started a new era of Multi-Messenger Astrophysics and revolutionize our understanding of the Cosmos. Machine learning techniques such as artificial neural networks are already transforming many technological fields and have also proven successful in gravitational-wave astrophysics for detection and characterization of gravitational-wave signals from binary black holes [5–7]. Here we use a deep-learning approach to rapidly identify transient gravitational-wave signals from binary neutron star mergers in noisy time series representative of typical gravitational-wave detector data. Specifically, we show that a deep convolution neural network trained on 100,000 data samples can rapidly identify binary neutron star gravitational-wave signals and distinguish them from noise and signals from merging black hole binaries. These results demonstrate the potential of artificial neural networks for real-time detection of gravitational-wave signals from binary neutron star mergers, which is critical for a prompt follow-up and detailed observation of the electromagnetic and astro-particle counterparts accompanying these important transients.

The detections of gravitational waves from binary black hole (BBH) mergers have verified Einstein's theory of General Relativity in extraordinary detail in the most violent astrophysical environments [1–4, 8]. In addition, the first observation of coalescing neutron stars in both gravitational and electromagnetic spectra has initiated the era of Multi-Messenger Astrophysics, which uses observations in electromagnetic radiation, gravitational waves, cosmic rays, and neutrinos to provide deeper insights about properties of astrophysical objects and phenomena [4, 9]. These discoveries were made possible by the Advanced Laser Interferometer Gravitational Wave Observatory (LIGO) and Virgo collaborations. As gravitational-wave detectors increase their sensitivity many more observations, including BBH, binary neutron star (BNS) and black hole - neutron star (BHNS) signals are likely to be detected more frequently. Conventional gravitational-wave detection techniques are based mainly on a method known as template matched filtering [7, 10], which typically uses large banks of template waveforms each with different compact binary parameters, such as component masses and/or spins. Since parameters are not known in advance, a template bank spans a large astronomical parameter space, which makes these approaches very computationally expensive and challenging, especially for real-time detection of important gravitational wave transients where a rapid follow-up is critical for successful observation of their

electromagnetic counterparts. In particular, the optical counterparts of gravitational waves from the merger of BNS and BHNS systems, know as kilonovae [11], encode key information required to constrain the physical properties of the transient, but due to their fast decay rate they need to be identified and localized within several hours after the compact binary merger and promptly observed in the entire electromagnetic spectrum. Therefore, the need arises for new methods to detect in real time binary neutron star (and black hole - neutron star) gravitational-wave signals.

In this work, we explore a deep-learning approach to rapidly detect gravitational-wave signals from binary neutron star mergers. Deep learning algorithms [12], a subset of machine learning, have been very successful in tasks, such as image recognition [12, 13], natural language processing [14], and recently also emerged as a new tool in gravitational-wave astrophysics for detection and characterization of gravitational wave signals from binary black holes [5–7]. Deep-learning methods are able to perform analysis rapidly since the computationally intensive part of the algorithm is done during the training stage before the actual data analysis [15], which could make them orders of magnitude faster than conventional match-filtering techniques [5]. Here, we demonstrate the power of the deep-learning approach on the specific example of rapidly classifying gravitational waves from binary neutron star mergers from detector noise and signals from binary black holes. This example shows clearly that machine learning can help in the real-time detection of BNS signals and thus trigger a prompt follow-up of the electromagnetic counterparts of the gravitational-wave transient.

Deep learning algorithms consist of processing units, neurons, which are arranged in arrays forming one to several layers. A neuron acts as a filter performing a linear operation between the input array and the weights associated with the neuron. A deep neural network has an input layer, typically followed by one or more hidden layers, and a final layer with one or more output neurons. In classification problems, the output neurons give the probabilities that an input sample belongs to a specific class. In this case, we distinguish between three classes of time series, BNS and BBH merger signals in additive Gaussian nose (signals plus noise), and Gaussian noise only. Accordingly, the data sets consist of simulated gravitational-wave time series where the compact binary merger signals (BNS and BBH) are generated using the LIGO Algorithm Library–LALSuite [16]. For the BNS signals, we use the PhenomPNRT waveform model [17] and simulate systems with component masses in the range from 1 to $2M_{\odot}$, including also tidal deformation contributions (see Methods for details). The BBH signals are simulated using the SEOBNRv2 waveform model [18], which models the inspiral, merger and ringdown components of the signal. We simulate systems with component masses in the range from 5 to $50M_{\odot}$, with zero spin. The simulated signals are chosen to be 2 seconds in duration sampled at 8192 Hz. This choice was made because BNS signals are considerably longer and contain typically much higher frequencies than BBH gravitational-wave signals. The simulated signals are "whitened" with Advanced LIGO's power spectral density (PSD) at the "zero-detuned high-power" [19] to rescale the contribution of each frequency to have equal power. Subsequently, the waveforms are shifted randomly such that the peak amplitude of each waveform is randomly positioned in the range from 1.75 to 1.95 seconds of the time series to reassure robustness of the network against

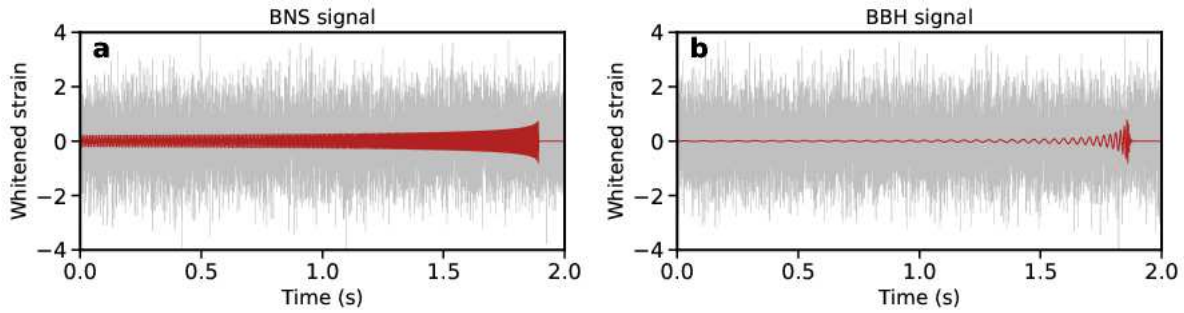

FIG. 1: **Sample BNS and BBH signals injected in simulated Gaussian noise. (a)** A whitened noise-free time series of a binary neutron star gravitational-wave signal with component masses $m_1 = 1.4M_\odot$ and $m_2 = 1.6M_\odot$ and dimensionless tidal deformability $\Lambda_1 = 261.9$ and $\Lambda_2 = 105.5$ (computed with the APR equation of state (EOS) [24]) with pSNR = 0.8 (dark red). The grey time series shows the same gravitational-wave signal with additive white Gaussian noise of unit variance. This time series is an example of the data sets used to train, validate and test the convolution neural network. **(b)** Same as (a) but for a binary black-hole gravitational-wave signal with component masses $m_1 = 27M_\odot$ and $m_2 = 49M_\odot$. ($\Lambda = 0$ for black holes.)

spatial translations. Different realizations of white Gaussian noise are superimposed on top of the signals, while the waveform amplitude is scaled to achieve a predefined peak signal-to-noise ratio (pSNR), defined as the ratio of the amplitude of the gravitational-wave signal to the standard deviation of the noise. The optimal matched-filter signal-to-noise ratio (SNR_{MF}) is on average 13 times the pSNR [5]. Example time series are shown in Fig. 1. See Methods for more details on data preparation.

Supervised learning requires that data sets are divided into training, validation and testing data. Training data is used by the network to learn from, validation data allows for verification of whether the networks is learning correctly, and the testing data is used to assess the performance of the trained model. The training sets used here consist of 100,000 independent time series with 1/3 containing BNS signal + noise, 1/3 BBH signal + noise, and 1/3 noise only. The validation and testing data sets each consist of 5,000 independent samples containing (approximately) equal fractions of each time-series class. To ensure that the neural network can identify BNS gravitational-wave signals over a broad range of astrophysically motivated pSNR values, we start the network training with large pSNR and then gradually reduce the pSNR to lower levels. This approach is adapted from "curriculum learning" [20] and enables the network to learn to distinguish signals with lower pSNR more accurately. (See Methods for details.)

The neural network used here is a convolutional neural network (CNN) [21] and have 4 convolutional and 4 pulling layers, followed by 2 dense fully connected layers. The first layer corresponds to the inputs to the neural network which in this case is a one-dimensional time-series vector (of dimension 16,384). Each neuron in the convolutional layers computes the convolution between the neuron's weight vector and the outputs from the layer below. Neuron weights are updated through an optimization back propagation algorithm [22]. Pooling layers perform a down-sampling operation along the spatial dimensions of their input. At the end, there is a hidden

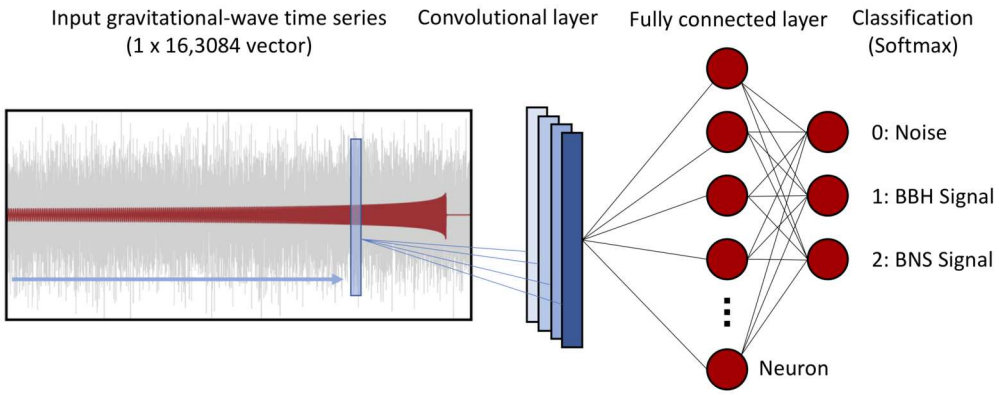

FIG. 2: **Using an artificial neural network to detect gravitational-wave signals from binary neutron-star mergers.** Gravitational-wave time series serve as input for a deep convolutional network with convolutional and fully connected layers. The blue arrow represents the sliding of the convolutional filters along the input time-series vector. The last softmax layer outputs the probability that the input time series belongs to a certain class (0: Noise, 1: BBH signal, or 2: BNS signal). The weights of the artificial neural network are tuned by trained on many labeled data samples and the network can then classify an unknown sample time series with high confidence.

dense layer connected to an output layer computing the inferred class probabilities. The network design is optimized by fine-tuning multiple hyper-parameters, which include here the number and type of network layers, the number of neurons in each layer, max-pooling parameters, and type of activation functions. The optimal network architecture was determined through multiple experiments and tuning of the hyper-parameters. More details are given in Methods. The process of using an artificial neural network to detect gravitational-wave signals from BNS mergers is illustrated in Fig. 2.

We assess the performance of the neural network by extracting the probability values produced by the neurons in the output layer. These values are between 0 and 1 with their sum being unity. Each neuron gives the inferred probability that the input time series belong to the noise, BBH signal, or BNS signal class, respectively. Specifically, for a given pSNR value we calculate the normalized confusion matrix, which evaluates the quality of the classifier's output. The diagonal elements represent the number of occasions where the predicted label is equal to the true label (true positive rate), while the off-diagonal elements are those mislabeled by the classifier. The normalization is done by class support size (number of samples in each class) and helps the visual interpretation of which class is being mislabeled. The confusion matrix is shown in Fig. 3 for two representative pSNR values, 0.5 and 0.75 ($\mathrm{SNR}_{MF} = 6.5$ and 9.75). These results show that all BNS and BBH signals are correctly classified for pSNR ≥ 0.75 ($\mathrm{SNR}_{MF} \geq 9.75$). For pSNR $= 0.5$ ($\mathrm{SNR}_{MF} = 9.75$) about 26% of BBH and 3% of BNS signals are misclassified as noise. Furthermore, all BNS signals are correctly distinguished from BBH signals for pSNR ≥ 0.5 ($\mathrm{SNR}_{MF} \geq 6.5$). We analyse further the performance of the classifier by looking at the sensitivity of detection of BNS and BBH signals for different pSNR values. These sensitivity curves are shown in Fig. 3 (c), where the sensitivity is plotted as a function of pSNR. We also show the overall accuracy of the deep neural network. All curves saturate at 100% for pSNR ≥ 0.75 (SNR_{MF}

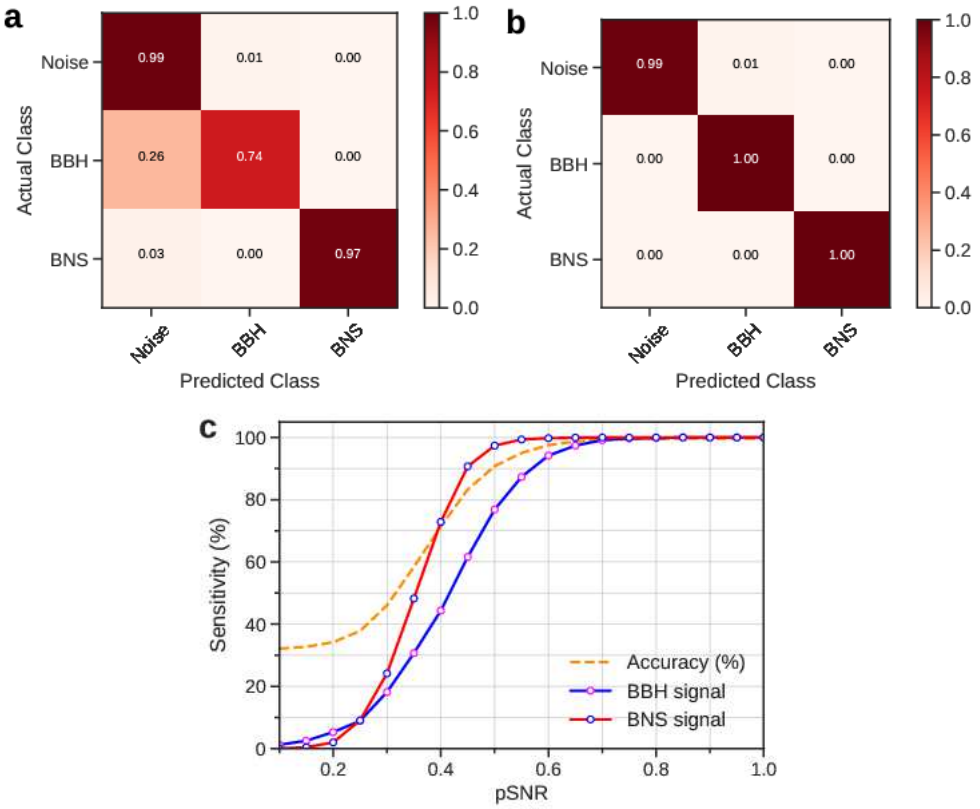

FIG. 3: **Performance of the artificial neural network.** **(a)** Confusion matrix for a classifier with 3 classes on a test data set with pSNR = 0.5 ($\mathrm{SNR}_{MF} \approx 6.5$). In this case the sensitivity of detection is 97% for a BNS signal and 74% for a BBH signal respectively. **(b)** Same as (a) but for pSNR = 0.75 ($\mathrm{SNR}_{MF} \approx 9.75$). In this case the sensitivity of detection is 100% for both BNS and BBH signals. **(c)** Sensitivity of detection as a function of pSNR. The false-alarm rate of this classifier was tuned to be about 0.6% for BBH and 0.03% for BNS signal detection. The broken orange line shows the overall accuracy of the classifier. All curves saturate at 100% for pSNR ≥ 0.75 ($\mathrm{SNR}_{MF} \geq 9.75$), i.e., all signals (both BNS and BBH) with $\mathrm{SNR}_{MF} \geq 9.75$ are always detected and correctly classified.

≥ 9.75), i.e., all signals, both BNS and BBH, with $\mathrm{SNR}_{MF} \geq 9.75$ are always detected and correctly classified. Furthermore, the deep neural network automatically extracts and compresses information by finding patterns in the training data, dramatically reducing data dimensionality and thus creating a very computationally efficient and portable model. For instance, the size of the trained model is only about 7 MB, including the network weights and architecture information, therefore compressing approximately 6.6×10^4 gravitational BBH and BNS waveforms (excluding noise samples), each of duration 2 seconds sampled at 8192 Hz, with a total size of about 4.2 GB. Once trained, processing 2 seconds of gravitational-wave data takes only milliseconds on both CPUs and GPUs. These results clearly demonstrate the ability of deep learning algorithms to promptly detect gravitational-wave signals from binary neutron star mergers and distinguish them from BBH signals and noise over a wide SNR range, with moderate computing resources (e.g., a standard laptop computer).

In conclusion, we have demonstrated the detection of gravitational waves from binary neutron star mergers via deep learning techniques on simulated gravitational-wave detector data using

the specific example of data containing BNS and BBH signals in Gaussian noise. These results point the way to real-time detection of gravitational waves from multi-messenger astrophysical sources, where a rapid follow-up is critical. Future directions include using machine learning algorithms for real-time parameter estimation of gravitational-wave signals from BNS and BHNS systems. In particular, machine learning approaches could help to extract challenging source parameters, e.g., neutron-star tidal deformability [23], which is extremely important for understanding properties of dense matter and fundamental interactions, but is theoretically controversial and observationally challenging to deduce. As in all technological and scientific areas of "big data", we already witness the rapid adoption of deep learning techniques as a basic research tool in gravitational and multi-messenger astrophysics.

I thank Brendan Meade for stimulating my interest in machine learning and for useful discussions. The computational resources were provided by the Faculty of Arts and Sciences Research Computing at Harvard University.

*Correspondence to: plamenkrastev@fas.harvard.edu

Methods

Data preparation. The gravitational BNS waveforms were generated with the PhenomP-NRT model [17] implemented in the LALSuite library under the name IMRPhenomPv2_NRTidal. This model includes a black-hole baseline waveform and added tidal effects. We compute the dimensionless tidal deformability, Λ, with the APR equations of state (EOS) [24]. (See, e.g., Ref. [25] for details on the formalism for calculating Λ.) The component masses in all data sets were randomly sampled in the range from $1M_\odot$ to $2M_\odot$ such that $m_1 > m_2$. The BNS waveforms were generated from the initial gravitational-wave frequency of 60 Hz using a sampling rate 8192 Hz. (Since we are interested in the last 2 seconds of the signal, the higher starting frequency helps speeding up the waveform generation.) The gravitational-wave BBH templates were generated with the SEOBNRv2 waveform model [18] from LALSuite. The BBH component masses in the data sets were randomly chosen in the range from $5M_\odot$ to $50M_\odot$ with $m_1 > m_2$, and the compact binary systems were modeled with zero spin. The last 2 seconds of data from these waveforms were selected and resampled at 8192 Hz. Subsequently, all waveforms were whitened by dividing, in Fourier space, with Advanced LIGO's design sensitivity amplitude spectral density (ASD) of noise. For this, the "zero-detuned high-power" sensitivity was used. The waveforms were also shifted randomly such that the peak of the gravitational-wave amplitude was randomly positioned within the interval [1.75, 1.95] seconds of the time series. Before each training session, different realizations of white Gaussian noise of unit variance were superimposed on top of the compact binary waveforms, while the waveforms were scaled according to a desired predefined pSNR. The resulting time-series were then rescaled to have zero mean and unit standard deviation. The final data sets were supplemented with samples containing noise only, and reshuffled to randomize the sample order. Each template is a vector of 16,384 real numbers. Random data samples from the final data set are shown in Fig. 4.

Neural-network architecture and implementation. Motivated by the recent success of convolutional neural networks in detection and characterization of gravitational waves from BBH mergers, here we employed a deep convolutional network to identify gravitational-wave signals from binary neutron stars. A variety of network designs were studied with different number of convolutional and fully connected layers, and various filter sizes, and it was determined that the network performance was rather robust to the choice of these parameters. For the results presented in the main text, we used the architecture defined in Fig. 5. To build and train the neural network, the Python toolkit Keras (https://www.tensorflow.org/guide/keras) was used, which provides a high-level programming interface to access the TensorFlow [26] (https://www.tensorflow.org) deep-learning library. We use the technique of stochastic gradient descent with an adaptive leaning rate with the ADAM method [27]. To train the neural network, we use an initial learning rate of 0.001 and choose a batch size of 100.

For all training sessions, the pSNR of each BBH and BNS waveform was randomly sampled in the range [pSNR_S, 1.5]. Initially, pSNR_S was set to 1.5 and then gradually decreased to 0.5 in steps of 0.1 in each subsequent training session. Thus, the final pSNR range was uniformly

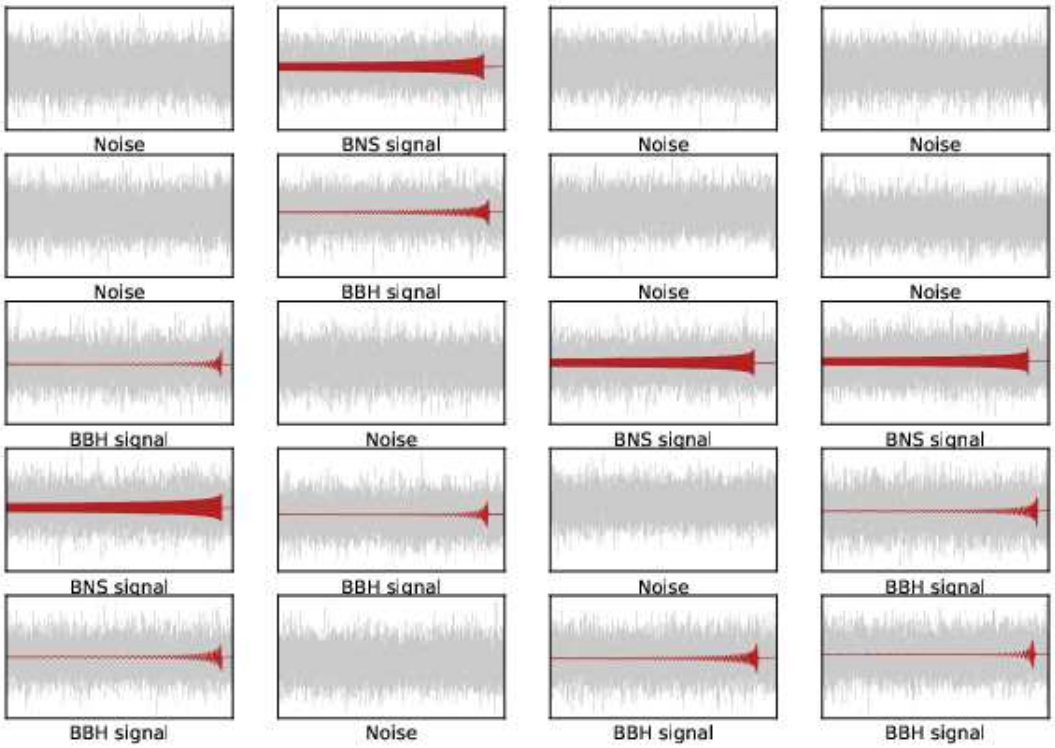

FIG. 4: **Random data samples.** The network is trained with many such labeled templates. The training data set has 100,000 samples with approximately equal number of templates representing each class (Noise, BBH signal and BNS signal). The dark red time-series have represent the whitened noise-free BBH and BNS waveforms and have pSNR = 1.

sampled in the range between 0.5 and 1.5 ($6.5 \leq \mathrm{SNR}_{MF} \leq 19.5$). This training strategy is adopted from "curriculum learning" [20] and ensures that the performance is rapidly maximized for low pSNR while retaining accuracy for high pSNR. For each pSNR range the number of training epoch is limited to 100, or until the error on the validation data set stops decreasing. The size of the mini-batches was chosen automatically depending on the specifics of the GPU and data sets. The cost function was selected to be the cross-entropy loss. The network training was performed on NVIDIA Tesla K80 GPU.

Neural-network performance. To evaluate the performance of the neural network at each pSNR, we compute the normalized confusion matrix, sensitivity (true positive rate) of each class, and overall accuracy. These quantities are conveniently calculated with the Python PyCM library (https://www.pycm.ir). The pSNR was varied from 0.1 to 1.5 in steps of 0.05 and the classifier was applied to time-series inputs containing approximately equal fractions of each class (0: Noise, 1: BBH signal, 2: BNS signal).

```
1   Input            vector (size: 16384)
2   Reshape          matrix (size: 1 x 16384)
3   Convolution (1D) matrix (size: 16 x 16369)
4   Pooling          matrix (size: 16 x 4092)
5   ReLU             matrix (size: 16 x 4092)
6   Convolution (1D) matrix (size: 32 x 4085)
7   Pooling          matrix (size: 32 x 1021)
8   ReLU             matrix (size: 32 x 1021)
9   Convolution (1D) matrix (size: 64 x 1014)
10  Pooling          matrix (size: 64 x 253)
11  ReLU             matrix (size: 64 x 253)
12  Convolution (1D) matrix (size: 128 x 246)
13  Pooling          matrix (size: 128 x 61)
14  ReLU             matrix (size: 128 x 61)
15  Flatten          vector (size: 7808)
16  Dense Layer      vector (size: 64)
17  ReLU             vector (size: 64)
18  Dense Layer      vector (size: 3)
19  Output           vector (size: 3)
```

FIG. 5: **Architecture of the deep neural network.** The architecture of the deep one-dimensional convolutional network consist of input layer, followed by 17 hidden layers, and output *softmax* layer. The size of the network is about 7 MB.

[1] B. P. Abbott *et al.* (Virgo, LIGO Scientific), Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.* **116**, 061102 (2016).

[2] B. P. Abbott *et al.* (Virgo, LIGO Scientific and Virgo Collaborations), GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence. *Phys. Rev. Lett.* **116**, no. 24, 241103 (2016).

[3] B. P. Abbott *et al.* (Virgo, LIGO Scientific), 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**, 129901 (2018)].

[4] B. P. Abbott *et al.* (Virgo, LIGO Scientific), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral. *Phys. Rev. Lett.* **119**, 161101 (2017).

[5] D. George and E. A. Huerta, Deep Neural Networks to Enable Real-time Multimessenger Astrophysics. *Phys. Rev. D* **97**, 044039 (2018).

[6] 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).

[7] 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).

[8] B. P. .Abbott *et al.* (Virgo, LIGO Scientific), GW170608: Observation of a 19-solar-mass Binary Black Hole Coalescence. *Astrophys. J.* **851**, L35 (2017).

[9] B. P. Abbott *et al.* (Virgo and LIGO Scientific), Estimating the Contribution of Dynamical Ejecta in the Kilonova Associated with GW170817. *Astrophys. J.* **850**, L39 (2017).

[10] T. Dal Canton *et al.*, Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors. *Phys. Rev. D* **90**, 082004 (2014).

[11] B. D. Metzger and E. Berger, What is the Most Promising Electromagnetic Counterpart of a Neutron Star Binary Merger? *Astrophys. J.* **746**, 48 (2012).

[12] Y. LeCun, Y. Bengio and G. Hinton, Deep Learning. *Nature*, **521**, 436 (2015).

[13] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **770** (2016).

[14] T. Young, D. Hazarika, S. Poria and E. Cambria, Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Computational Intelligence Magazine* **13**, 55-75 (2018).
[15] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, Cambridge, MA, 2016).
[16] LIGO Scientific Collaboration, LIGO Algorithm Library – LALSuite (GPL 2018).
[17] B. P. Abbott *et al.* (Virgo and LIGO Scientific), Properties of the binary neutron star merger GW170817. *Phys. Rev. X* **9**, 011001 (2019).
[18] M. Prrer, Frequency domain reduced order model of aligned-spin effective-one-body waveforms with generic mass-ratios and spins. *Phys. Rev. D* **93**, no. 6, 064041 (2016).
[19] B. P. Abbott *et al.* (Virgo and LIGO Scientific), Prospects for Observing and Localizing Gravitational-Wave Transients with Advanced LIGO and Advanced Virgo. *Living Rev. Relativ.*, **19**, 1 (2016).
[20] H. Shen, E. A. Huerta and Z. Zhao, Deep Learning at Scale for Gravitational Wave Parameter Estimation of Binary Black Hole Mergers. arXiv:1903.01998.
[21] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278-2324 (1998).
[22] Y. Hirose, K. Yamashita and S. Hijiya, Back-propagation algorithm which varies the number of hidden units. *Neural Networks* **4**, 61-66 (1991).
[23] T. Hinderer, B. D. Lackey, R. N. Lang and J. S. Read, Tidal deformability of neutron stars with realistic equations of state and their gravitational wave signatures in binary inspiral. *Phys. Rev. D* **81**, 123016 (2010).
[24] A. Akmal, V. R. Pandharipande, D. G. Ravenhall, Equation of state of nucleon matter and neutron star structure. *Phys. Rev. C* **58**, 1804 (1998).
[25] P. G. Krastev and B. A. Li, Imprints of the nuclear symmetry energy on the tidal deformability of neutron stars. *J. Phys. G* **46**, 074001 (2019).
[26] Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[27] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization. arXiv:1412.6980