Machine learning classification of continuous gravitational-wave signal candidates

Filip Morawski[1], Michał Bejger[1] and Paweł Ciecieląg[1]

1. Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00–716 Warsaw, Poland

Continuous gravitational waves (GWs) emitted by non-axisymmetric, rotating neutron stars are expected to be detected in the near future by LIGO, Virgo and Kagra interferometers. Although the GW waveform is well known, its small amplitude makes it difficult to register. Searches for continuous GWs, among them the $\mathcal{F}$-statistic method used here, are based on matched-filtering i.e., evaluation of signal templates on a grid of parameters, resulting in distributions of candidate GW signals.

In our work we present the application of machine learning in the analysis of the distributions of $\mathcal{F}$-statistic signal candidates belonging to one of three distinct classes: pure Gaussian noise, astrophysical continuous GW, and stationary-frequency detector artifacts (lines). We study the robustness of candidates' classification with the use of one-dimensional convolutional neural networks, demonstrating their benefits in the context of the analysis of continuous GWs. We also show limits to the signal-to-noise ratio of the signal our method is able to correctly identify, and the ability of generalization from the training data.

1 Introduction

To date the LIGO and Virgo collaborations have registered several GWs emitted by the binary black hole and binary neutron star mergers (Abbott et al., 2019). Among other promising sources of GWs expected to be observed in the future are non-axisymmetric rotating neutron stars. They produce almost monochromatic, long-lasting signals - continuous GWs. We will focus on them in this article.

Studies of the continuous GWs allow to probe the interior of neutron stars and understand the dense-matter equation of state (EOS), which is still lagerly unexplored. The EOS determines the amplitude and the frequency of the GW signal. Thus, by detecting a continuous GW corresponding to the particular model, we will verify the nature of physical processes ongoing in the interior of the star (for a recent review see, e.g., Sieniawska & Bejger 2019).

The LIGO and Virgo collaborations performed several searches for continuous GWs, among them targeted searches for GWs from pulsars with known spin parameters and sky localization (Abbott et al. 2017c,b and references therein). On the other side of the computational dificulty spectrum are the all-sky searches for *a priori* unknown sources with unknown parameters (Abbott et al. 2017a, 2018 and references therein). In this case the parameter space consists of the GW frequency parameters in a sufficiently wide, pre-defined range, as well as possible sky positions of the source. Then, the space of parameters is being swept to find the best-matching template by looking for the highest signal-to-noise ratio (SNR) ρ.

In the present work, we study the output (candidate signals) of the all-sky time-domain $\mathcal{F}$-statistic search (`TD-Fstat search`, Astone et al. 2010; Aasi et al. 2014; Abbott et al. 2017a, 2018), which implements the $\mathcal{F}$-statistic (Jaranowski et al., 1998). In the all-sky search, the best-matched filter (a collection of best-matching parameters of the signal) are found by evaluating a specific maximum likelihood function - the $\mathcal{F}$-statistic - for a range of the GW frequency f, its time derivatives (spindown $\dot{f}$) and sky positions (right ascension α and declination δ in the equatorial coordinates), by performing Fourier transforms on the time-series data, constituting the GW detectors' output. By design, the number of sky coordinates' grid points as well as $\dot{f}$ grid points increases with frequency. Consequently the volume of the parameter space (number of evaluation of the $\mathcal{F}$-statistic) increases, see e.g., Fig. 4 in Poghosyan et al. (2015), as well as the total number of resulting *candidate GW signals* (crossings of the pre-defined SNR ρ threshold increases). For high frequencies, this type of a search is particularly computationally demanding.

The aim of this work is to assess distributions of the $\mathcal{F}$-statistic signal candidates via supervised machine learning algorithm: one-dimensional convolutional neural networks (CNNs). The data used in our studies was generated for three different types of candidates: pure Gaussian noise, Gaussian noise with injected purely monochromatic signals (simulating monochromatic spectral artifacts local to the detector) and Gaussian noise with injected realistic astrophysical signal. These classes are henceforth denoted as the **noise**, **line** and **cgw**.

This article is organized as follows. In Sect. 2 we describe the machine learning (ML) algorithm used in the studies. Section 3 covers the generation of data and the implementation of ML algorithms. Section 4 summarizes our results which are further discussed. The conclusions are gathered in Sect. 5.

2 Machine Learning

ML is a field of computer science based on a premise that algorithms can learn from examples in order to solve problems and make predictions, without the need of being explicitly programmed (Samuel, 1959). Among many ML algorithms, the artificial neural networks (ANN) belong to the most popular. Complex ANN consisting of many neurons combined with learning algorithms based on back-propagation and stochastic gradient descent (Goodfellow et al. 2016 and references therein) are able to capture complicated non-linear relationships in the data, by composing hierarchical internal representations. The complex (also called deeper) the algorithm is, the more abstract features it can learn from the data.

ML algorithms can be divided into two categories depending on the type of the training: supervised and unsupervised learning. In the first category, algorithms learn from labeled data. Thus, they can be used to solve classification or regression tasks. In unsupervised learning, algorithms infer from unlabeled and unstructured data. In this case models learn about internal relationships in data that is used in the data clustering.

In the present work we focus on the supervised approach. As the supervised learning algorithm we chose the CNN. CNN is a deep, feed-forward ANN (network that process the information only from the input to the output) whose structure is inspired by the structure of visual cortex in mammals, the part of the brain which specializes in processing visual information. The crucial element of CNNs is called a

convolution layer. It detects local conjunctions of features from the input data and maps their appearances to a feature map. As a result the input data is split into parts, creating local receptive fields and compressed into feature maps. The size of receptive field corresponds to the scale of details present in the data.

3 Methodology

Signal candidates were generated by the `TD-Fstat search`[1]. This program uses narrow-banded time series data as an input which simulates the raw data taken from the GW detector, downsampled from the original frequency (16384 Hz for LIGO and 20000 Hz for Virgo) to $dt = 0.5$ Hz, and divided into narrow frequency bands for convenience. We used the narrowband width of $B = 1/(2dt) = 0.25$ Hz, as the almost-monochromatic GW is not expected to vary substantially in frequency (only the minor change of $\dot{f}$). The `TD-Fstat search` requires also as input the ephemeris of the detector and pre-defined grid parameter space of values (f, $\dot{f}$, δ, α) on which the search is performed. Details of the input data used are presented in Table 1.

To obtain the candidate signal distributions for the CNN classification, the `TD-Fstat search` performs the search over above-mentioned injection parameters in a time segment data (2 days in our case) in the range of a few grid points. Each candidate is described by the value of the SNR ρ, related to the $\mathcal{F}$-statistic. Only those candidates that exceeded predefined threshold of $\rho = 4$ are recorded, as weaker signals are not separable from the noise. Each search generated the distribution of signal candidates for particular set of input parameters (see Table 1). Distributions corresponding to a given injection parameter set are considered a single datum instance in the CNN training data.

Detector	LIGO Hanford
Reference band frequency	50, 100, 200, 300, 500, 1000 Hz (250, 400, 700, 900 Hz for tests)
Segment length T_0	2 days
Bandwidth	0.25 Hz
Sampling time dt	2 s
Grid range	± 5 points
$\mathcal{F}$-statistic threshold	14.5 (corresponding to $\rho = 5$)
Signal-to-noise of injections ρ	from 8 to 20 (from 4 to 20 for tests)

Tab. 1: Parameters of the input to the `TD-Fstat search` code. Time series consist initially of random instances of white Gaussian noise, to which the **cgw**s or **line**s were added. T_0 equals to 2 sidereal days with 2 s sampling time results in 86164 data points.

To summarize, each candidate GW candidate signal instance consist of the values of ρ, and values of the frequency f (in fiducial units of the narrow-band, from 0 to π), spindown $\dot{f}$ (in $\mathrm{Hz\,s^{-1}}$), and two angles describing its sky position in equatorial coordinates: right ascension α (values from 0 to 2π) and declination δ (values from $-\pi/2$ to $\pi/2$). Fig. 1 presents exemplary output distribution of the signal

[1]Time-domain $\mathcal{F}$-statistic pipeline repository at: https://github.com/mbejger/polgraw-allsky

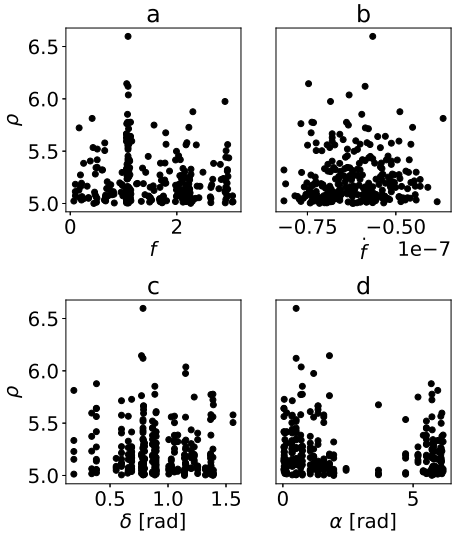

Fig. 1: Exemplary `TD-Fstat search` outputs for continuous GW signal generated for the $\rho = 10$ and frequency $f = 100\,\mathrm{Hz}$. The distributions represent the relation between ρ ($\mathcal{F}$-statistic ρ) with respect to: the frequency (panel a), the derivative of frequency (b), the declination (c) and the right ascension (panel d).

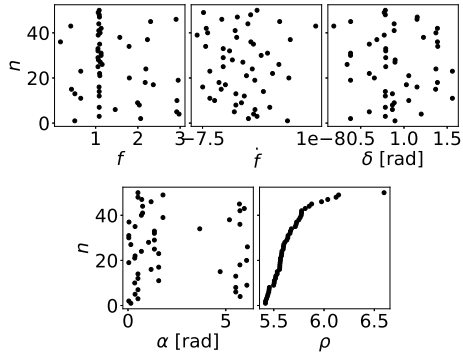

Fig. 2: The 1D representation of `TD-Fstat search` outputs used as an input to the CNN. The outputs are limited to the 50 maximum values of ρ (plots presented here correspond to the distributions shown on the Fig. 1): frequency (panel a), spin-down (b), declination (c), right ascension (d) and SNR ρ (panel e). The vector of ρ was sorted since this transformation allowed to reach a higher accuracy during training.

candidates.

In total we generated 292500 distributions of signal candidates corresponding to one of the three classes **noise**, **line**, **cgw** (2500 of distributions per ρ and per reference frequency). As specified in the Tab. 1, we limited range of ρ to 8 since below this value, candidates belonging to the studied classes seemed to be indistinguishable. In particular, in the domain of frequency (Fig. 1 a) peaks started to disappear in the noise. Nevertheless we increased the range of ρ down to 4 in the test phase of our models to analyse the response of ML models toward unknown during training, very weak signals.

CNN required the input data of fixed size. However distributions of signal candidates varied for each simulation. In particular the number of candidates increased significantly for higher frequencies. To address this issue, we transformed point-based distributions into set of five 1D vectors (five output `TD-Fstat search` parameters) shown on Fig. 2. The length of each vector was set to 50 points. The criterion for this length was the 50 greatest values of the ρ distribution. Later on we used these values to choose points from corresponding distributions of the other parameters (f, $\dot{f}$, δ, α). The length of the vector was chosen empirically and depends on the density of signal candidates distributions. The minimum number of candidates was around 50 (for the reference frequency 50 Hz in case of the **noise** class). Furthermore, the vectors in the 1D representation were sorted with respect to ρ values. This step allows to reach higher values of classification accuracy.

The datasets were then split into three separate subsets: training set (60% of signals from the total dataset), validation set (20% of signals from the total dataset) and the testing set (20% of signals from the total dataset). Validation subset was used during the training stage to monitor the CNN performance to avoid overfitting. Testing data was used after training to check how well the model performs with unknown samples.

The architecture of our ML model were as follows. The CNN contained three convolutional layers (40, 40, 20 with kernels of length 3) and two fully-connected layers (200, 100). These layers incorporated the ReLU activation functions. The final layer was a fully-connected layer with 3 neurons and softmax activation function. We added dropout (Goodfellow et al., 2016) to convolutional layers to avoid overfitting.

The ML architectures were built using the Python library Keras (Chollet et al., 2015) on top of the Tensorflow library (Abadi et al., 2015), with support for the hardware acceleration (GPU). The models were developed on the Quadro P6000 GPU card[2], and the production runs were performed on the Cyfronet Prometheus cluster[3] equipped with Tesla K40 GPUs, running CUDA 9.0 (Nickolls et al., 2008) and the cuDNN 7.3.0 (Chetlur et al., 2014).

4 Results

CNN described in previous section was trained on the generated datasets. During the training, CNN was able to correctly classify 93% of all signal candidates. To confirm this result, CNN was tested against unkown samples after the training (on the test set). The result is shown on Fig. 3 in the form of a confusion matrix. The model was able to correctly classify more 93% of **cgw**, 89% of **noise** and 97% of **line** test instances.

CNN was further tested against signal candidates generated for particular frequency and ρ. We included in this dataset signal candidates generated for different frequencies from those used for training and increased the range of ρ down to 4. This step allowed us to test the behaviour of CNN toward weak and unknown during training signals. The results are shown on Fig. 4. The model maintained nearly stable accuracy for the $\rho \geq 10$ (reaching the value of more than 90% for all of them). Interestingly, candidates with $\rho < 8$ were correctly classified in $60 - 70\%$ of samples for frequency ≥ 200 Hz. This was relatively high value, taking into consideration their noise-like pattern (for **cgw** and **line** instances). This pattern had the biggest influence on the classification of the signal candidates generated for frequencies: 50, 100 Hz and the $\rho < 8$. The small number of points contributing to the peak (see Fig. 1 *a* for comparison) with respect to the background noise, made these candidates hardly distinguishable from the **noise** class.

5 Summary

This research shows that ML, specifically CNNs, can be successfully applied in the classification of the `TD-Fstat search` results, multidimensional distributions of $\mathcal{F}$-statistic signal candidates which correspond to the three signal types: the continuous GW, the stationary line and the noise.

[2]Donation via the NVidia GPU seeding grant.
[3]Prometheus, Academic Computer Centre CYFRONET AGH, Kraków, Poland

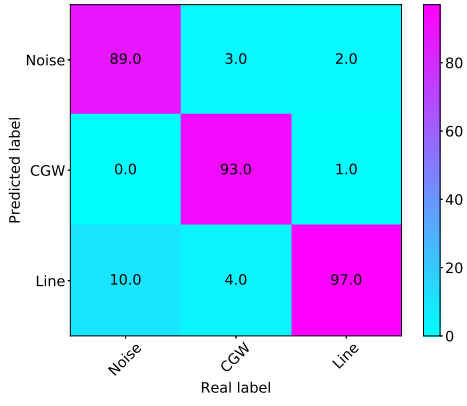

Fig. 3: Confusion matrix for the three-label classification evaluated on the test set for the CNN after the training.

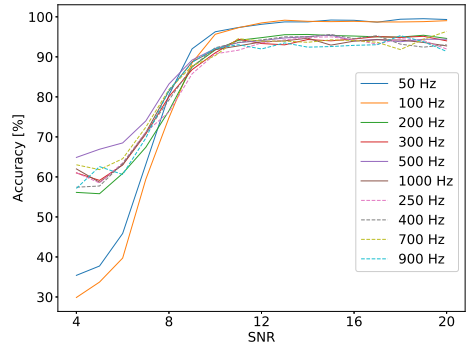

Fig. 4: The evolution of accuracy as the function of ρ for CNN over frequencies known (solid lines) and unknown (dashed lines) during training. The model achieved maximum level of accuracy for $\rho \simeq 10$ and maintained this performance for a whole range of frequencies.

CNN allowed to reach 93% of classification accuracy for the considered range of ρ and frequency (specified in Table 1). This level of accuracy was preserved at the classification of signal candidates generated for unknown during training frequencies. This result proved the generalization ability of our CNN - model's ability to adapt properly to new, previously unseen data. The limitation of the model was the minimum number of signal candidates it required as the input data. Since this number was associated with the frequency of the signal, the CNN was not able to classify signals below 50 Hz.

This project, as one of the few, concerns the application of ML in the studies of continuous GW. The advantage of our approach is the analysis of signal candidates instead of raw data. This allowed to avoid problems that other researchers encountered. As Gebhard et al. (2019) note, application of ML on raw data provides signal candidates of unknown or hard to define significance. Before ML could be used as a safe alternative to matched filtering for the detection of GW, it has to be studied further. However, our results can be considered in terms of supporting role to matched filtering. For example, CNN can be used in the classification of spectral artifacts, e. g., as an additional tool for flagging and possibly also removing spurious features from the data. Among many possibilities for further development within continuous waves searches we are considering is also the application of ML in the follow-up of signal candidates in multiple data segments, studying more complicated morphologies of the signals, as well as analysis of data from the network of detectors.

Acknowledgements. The work was partially supported by the Polish National Science Centre grants no. 2016/22/E/ST9/00037 and 2017/26/M/ST9/00978, and the European Cooperation in Science and Technology COST action no. CA17137 (G2Net). This research was supported in part by PL-Grid Infrastructure.

References

Aasi, J., et al., *Classical and Quantum Gravity* **31**, 16, 165014 (2014)

Abadi, M., et al. (2015), URL `https://www.tensorflow.org/`, software available from tensorflow.org

Abbott, B. P., et al., *Phys. Rev. D* **96**, 6, 062002 (2017a)

Abbott, B. P., et al., *Phys. Rev. D* **96**, 12, 122006 (2017b)

Abbott, B. P., et al., *ApJ* **839**, 12 (2017c)

Abbott, B. P., et al., *Phys. Rev. D* **97**, 10, 102003 (2018)

Abbott, B. P., et al., *Phys. Rev. X* **9**, 031040 (2019), URL `https://link.aps.org/doi/10.1103/PhysRevX.9.031040`

Astone, P., et al., *Phys. Rev. D* **82**, 2, 022005 (2010)

Chetlur, S., et al., *ArXiv e-prints* (2014)

Chollet, F., et al., `https://keras.io` (2015)

Gebhard, T. D., Kilbertus, N., Harry, I., Schölkopf, B., *arXiv e-prints* arXiv:1904.08693 (2019)

Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, The MIT Press (2016)

Jaranowski, P., Królak, A., Schutz, B. F., *Phys. Rev. D* **58**, 063001 (1998)

Nickolls, J., Buck, I., Garland, M., Skadron, K., *Queue* **6**, 2, 40 (2008), URL `http://doi.acm.org/10.1145/1365490.1365500`

Poghosyan, G., et al., *Computer Physics Communications* **188**, 167–176 (2015), URL `http://dx.doi.org/10.1016/j.cpc.2014.10.025`

Samuel, A. L., *IBM J. Res. Dev.* **3**, 3, 210 (1959), URL `http://dx.doi.org/10.1147/rd.33.0210`

Sieniawska, M., Bejger, M., *Universe* **5**, 11 (2019), URL `https://www.mdpi.com/2218-1997/5/11/217`