

Searching for gravitational waves in LIGO noise using neural networks

Ondřej Zelenka

in collaboration with Bernd Brügmann, Frank Ohme, Marlin B. Schäfer and Alexander H. Nitz

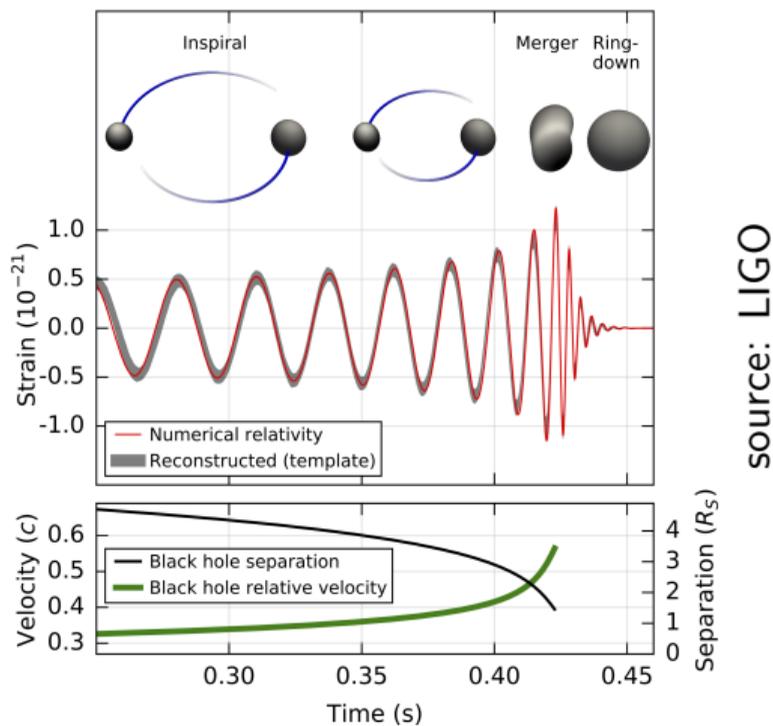
Astronomical Institute of the Czech Academy of Sciences
Theoretical Physics Institute, University of Jena, Germany

11 December 2024

5th EPS Conference on Gravitation, Prague

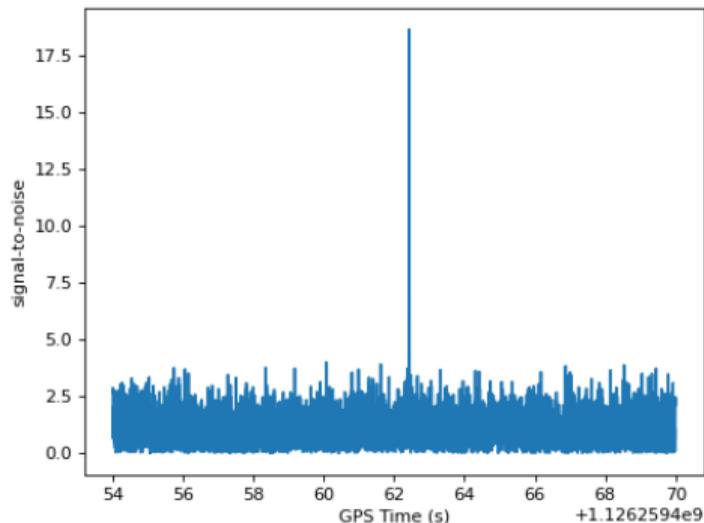
Gravitational waves

- ▶ disturbances in spacetime geometry
- ▶ most frequent source: BBH mergers
- ▶ cosmology, SMBH/AGN formation



Gravitational wave searches

- ▶ binary black holes
- ▶ coloured noise with transients
- ▶ standard: matched filtering
 - ▶ dense template bank
 - ▶ slide templates over signal
 - ▶ non-optimal
 - ▶ computationally demanding



source: PyCBC

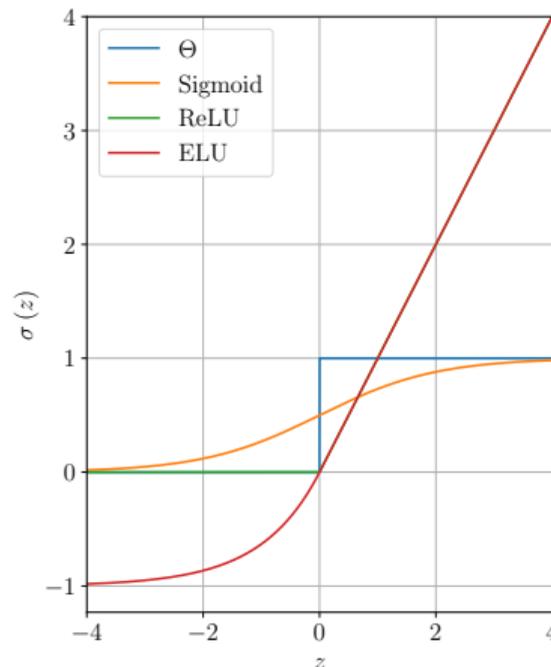
Neural Networks

artificial neurons

$$f(\mathbf{x}) = \sigma \left(\sum_j w_j x_j + b \right)$$

activation function

$$\text{ELU}(z) = \begin{cases} \exp(z) - 1 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$



Neural Networks

organized in layers:

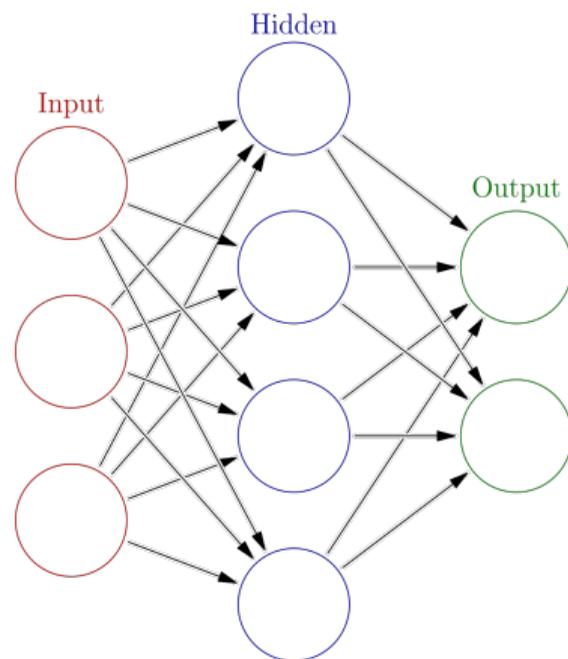
- ▶ matrix multiplication
- ▶ element-wise non-linear functions

training:

- ▶ dataset
- ▶ loss function (BCE)

$$\text{BCE}(\bar{\mathbf{Y}}, \mathbf{Y}) = -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^2 \log(\bar{Y}_{ij}) Y_{ij}$$

- ▶ GD-based optimizer



Feed forward and back-propagation

$$\mathbf{a}^{(0)} = \mathbf{x} ,$$

$$\forall l = 1, \dots, L :$$

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} ,$$

$$\mathbf{a}^{(l)} = \sigma^{(l)} \left(\mathbf{z}^{(l)} \right) ,$$

$$\mathbf{y} = \mathbf{a}^{(L)} .$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(L)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{y}} ,$$

$$\forall l = L, \dots, 1 :$$

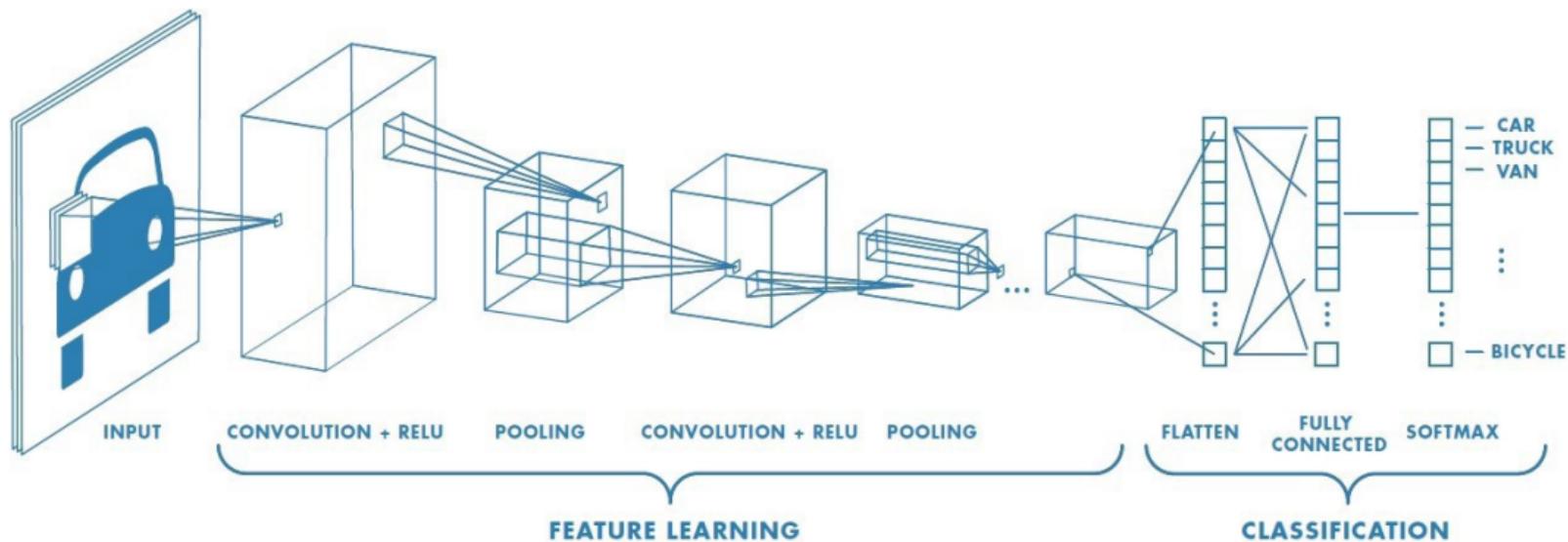
$$\frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(l)}} \odot \sigma^{(l)'} \left(\mathbf{z}^{(l)} \right) ,$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(l-1)}} = \left(\frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} \right)^T \mathbf{W}^{(l)} ,$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{W}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} \otimes \mathbf{a}^{(l-1)} ,$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{b}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} .$$

Convolutional Neural Networks



source: MathWorks

Neural Network frameworks

- ▶ TensorFlow: good for learning the basics
 - ▶ **PyTorch**
 - ▶ Theano, Caffe, Chainer, ...: not as common
- typically combination of Python, C/C++, CUDA, with Python interfaces

optimized for:

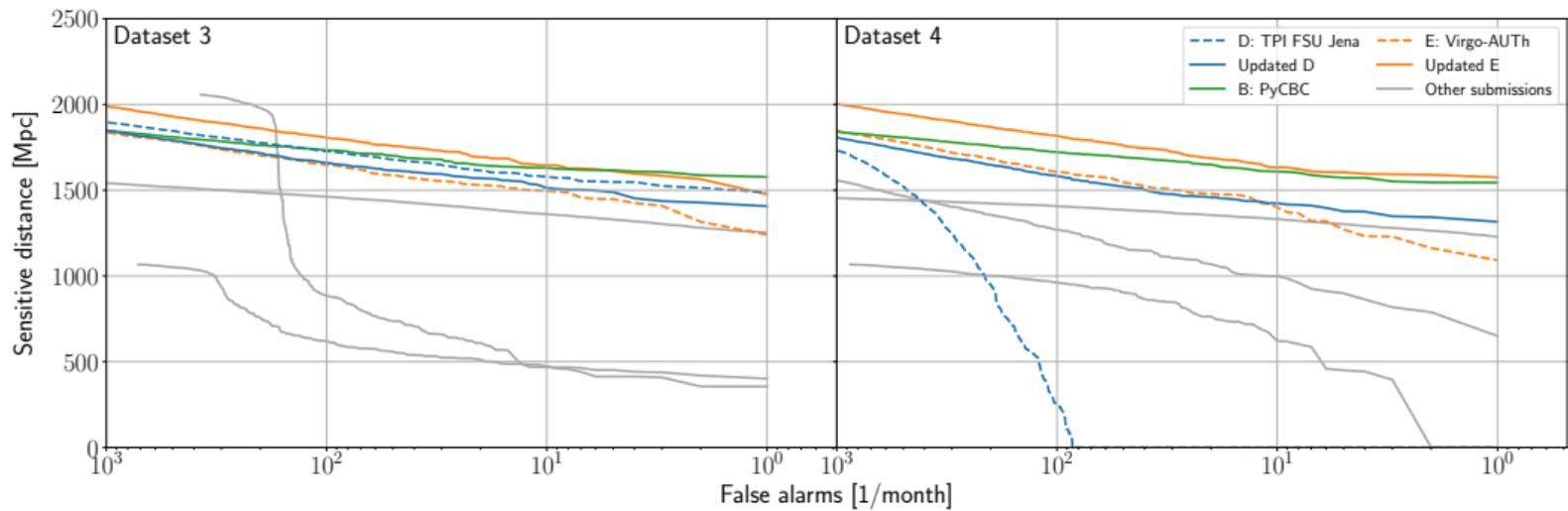
- ▶ large tensors
- ▶ GPU parallelization (CUDA etc.)



TensorFlow

 PyTorch

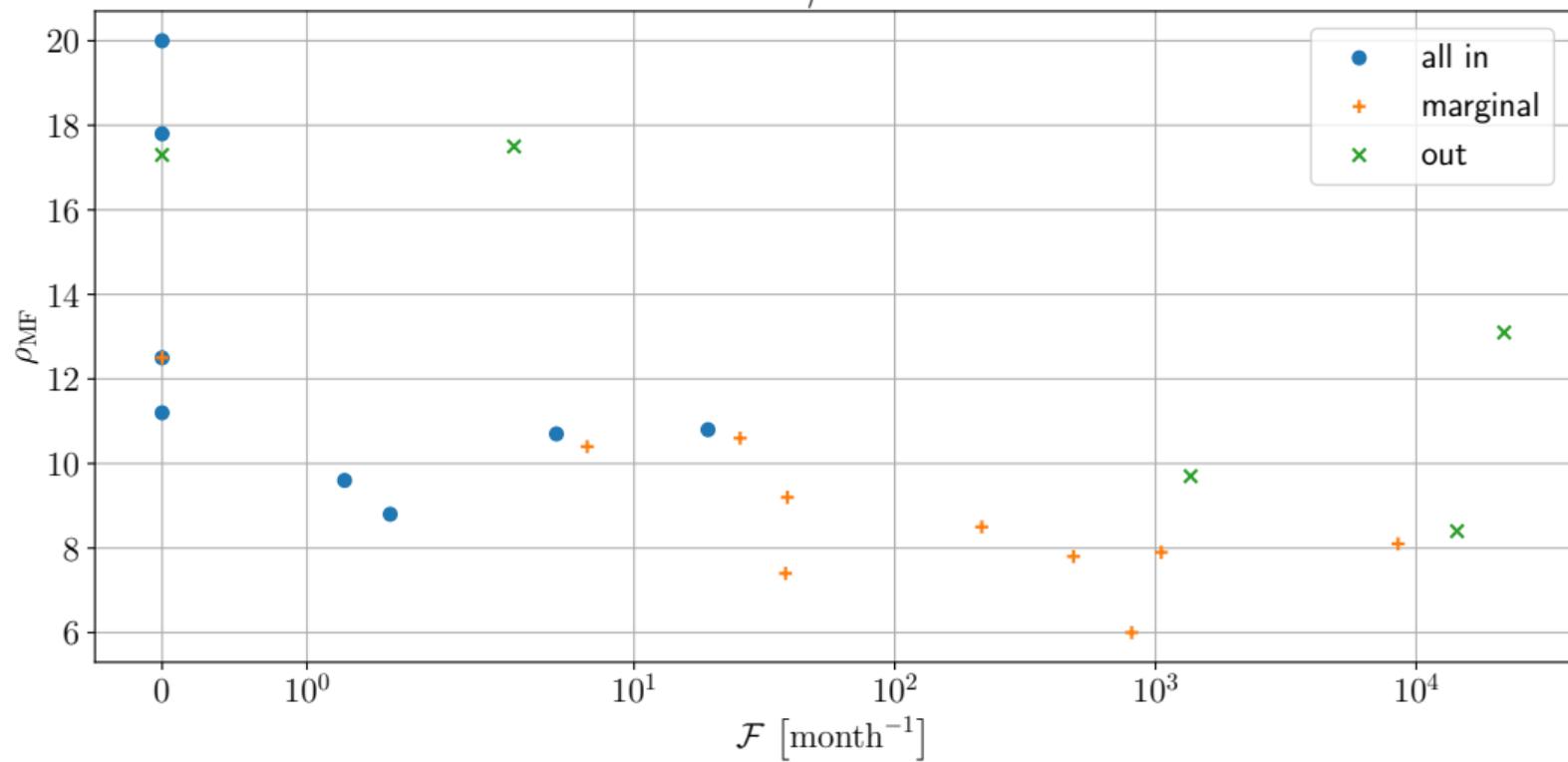
theano



Application to O3b data

- ▶ 1 November 2019 - 27 March 2020: ~ 147 days
- ▶ high data quality in both L1 and H1: ~ 95 days
- ▶ cropped GWTC-3: 31 confident events
- ▶ 90% intervals: $m_i \in [10M_{\odot}, 50M_{\odot}]$?
 - ▶ all in: found 8/9
 - ▶ marginal: found 10/11
 - ▶ out: found 5/11

R1/0021



Conclusions

Ground-based detectors:

- ▶ ML-based searches competitive on Gaussian noise
- ▶ handling real noise not fully understood
- ▶ speed

Extension to LISA data:

- ▶ longer signals
 - ▶ computational complexity
 - ▶ motion of detector → larger effective parameter space
- ▶ overlap
 - ▶ global fit