Deep Learning Continuous Gravitational Waves

Rahul Sharma[1,2]

In Collaboration with:
R Prix[1], C Messenger[3] and C Dreissigacker[1]

[1]Albert Einstein Institute, Hannover, Germany

[2]BITS Pilani, India

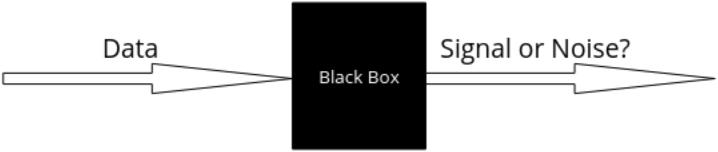
[3]University of Glasgow, UK

Continuous Gravitational Waves

- Generated by spinning non-axisymmetric compact objects.
  - Eg. Neutron Stars
- Weak signals. Buried under noise.
  - $h_0 \lesssim 10^{-24}$
  - $\sqrt{S_n} \gtrsim 10^{-23} Hz^{-\frac{1}{2}}$
  - depth $\equiv \frac{\sqrt{S_n}}{h_0} \gtrsim 10 Hz^{-\frac{1}{2}}$
- Matched filtering used for the search.
  - Fully Coherent: Computationally not feasible.
  - Empirical semi-coherent methods used.

Aim

- Find if the data contains a signal or just noise.

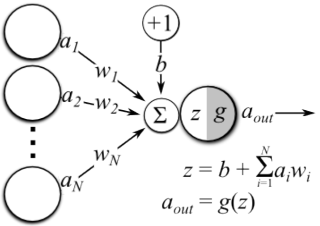


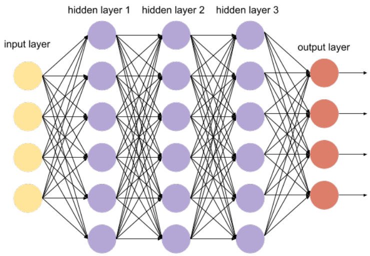
Matched Filtering Approach

- Generate templates of signals based on the search parameters.
- Match the templates with the signal and calculate a statistic.
  - for eg $\mathcal{F}$-Statistic
- Fix a threshold based on false alarm rate
  - for eg: only 1% of noise have the statistic $>$ threshold
- Data with statistic $>$ threshold considered a signal candidate.

Deep Neural Networks

- Algorithms which learn from examples
- Work on raw input data
- Already successful in CBC searches
  - George & Huerta, PRD(2018), Phys.Lett.B(2018)
  - Gabbard et al, PRL(2018)
- We use convolutional neural networks.
  - Using Keras + Tensorflow as our framework.





Our approach

- Generate training examples containing noise and signals.
  - In the frequency domain.
- Train the neural network to generate a statistic.
  - Higher value for signals, lower value for noise.
- Fix the threshold based on a fixed false alarm rate.
- Compare the performance with (coherent) matched filtering.

Organizing the data

- Dataset organized into "cases" depending on the parameters
  - frequency ranges, spin-down, duration of observation etc.
- Fix the strength (h_0) of the signal such that.
  - Matched filtering gets detection probability of 90% at 1% false alarm rate.
  - depth $= \frac{\sqrt{S_n}}{h_0}$

f (Hz)	$\dot{f}(Hz/s)$	$T_{obs}(s)$	depth $(Hz^{\frac{-1}{2}})$
[20,20.005]	$[-3 \times 10^{-9}, 0]$	1×10^5	10.53
[200,20.0025]	$[-1 \times 10^{-10}, 0]$	1×10^5	10.3
[20,20.001]	$[-1 \times 10^{-10}, 0]$	1×10^6	30.4
..	..	..	..

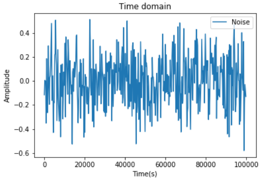
Training

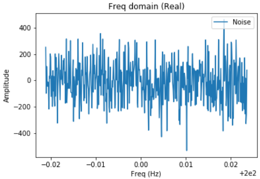
- Generate and store signals in a file using LALSuite[1]
- Generate and add Gaussian noise on the fly.
- Use noise and noise+signal examples to train the network.
- Check the performance using an independent validation set.
  - not seen by the network before.

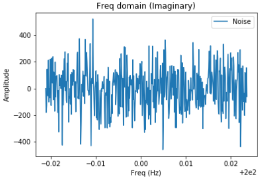
[1] https://wiki.ligo.org/DASWG/LALSuite

What does the input data look like?

f = [200,200.0025] Hz $\quad \dot{f} = [-1 \times 10^{-10}, 0]$ Hz/s $\quad$ Tspan $= 1 \times 10^{5}$s $\quad$ depth $= 10.3\ Hz^{\frac{-1}{2}}$

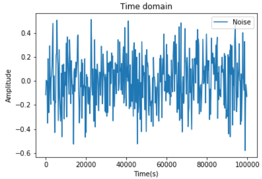


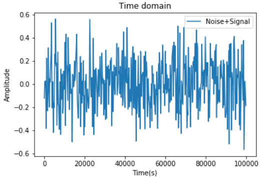


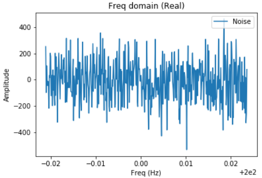


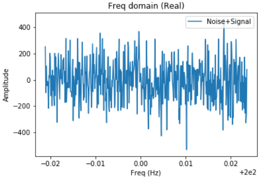
What does the input data look like?

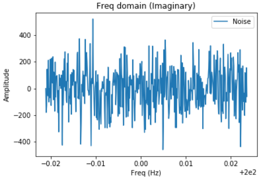
$f = [200, 200.0025]$ Hz $\qquad \dot{f} = [-1 \times 10^{-10}, 0]$ Hz/s $\qquad \text{Tspan} = 1 \times 10^{5}$s $\qquad \text{depth} = 10.3\ Hz^{\frac{-1}{2}}$

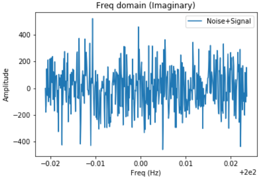






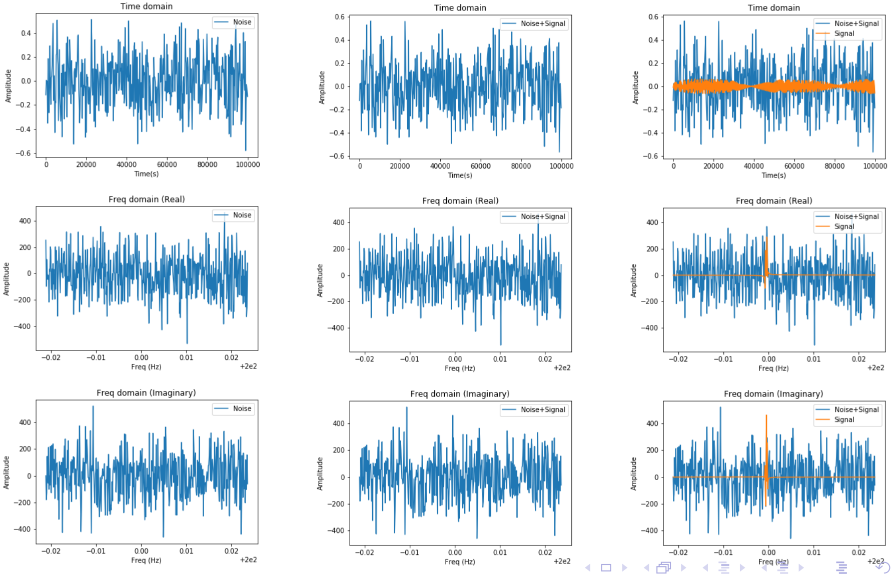






What does the input data look like?

$f = [200, 200.0025]$ Hz $\qquad \dot{f} = [-1 \times 10^{-10}, 0]$ Hz/s $\qquad$ Tspan $= 1 \times 10^{5}$s $\qquad$ depth $= 10.3\ Hz^{\frac{-1}{2}}$

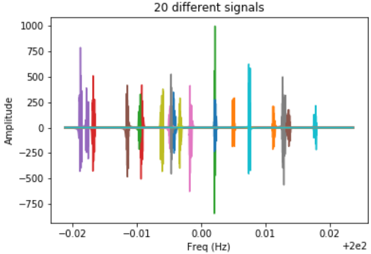


Frequency Band

- Sky position affects the doppler shifts.
- Signals cover wide frequency range.

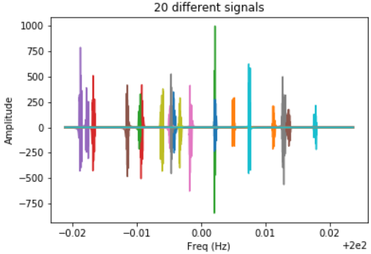
Frequency Band

- Sky position affects the doppler shifts.
- Signals cover wide frequency range.



Frequency Band

- Sky position affects the doppler shifts.
- Signals cover wide frequency range.
- Actual signal width quite small.
- Computationally more expensive.



Solution

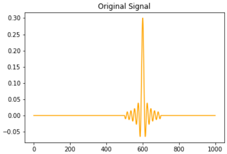
- freq range kept as twice the max width of a signal.
- Smaller input. Faster training.
- Slide over the whole range in the actual search.
  - Maximum value over the slides taken as the statistic.

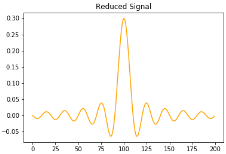
Solution

- freq range kept as twice the max width of a signal.
- Smaller input. Faster training.
- Slide over the whole range in the actual search.
  - Maximum value over the slides taken as the statistic.
- Example
  - max width = 200, total width = 1000

Solution

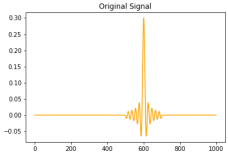
- freq range kept as twice the max width of a signal.
- Smaller input. Faster training.
- Slide over the whole range in the actual search.
  - Maximum value over the slides taken as the statistic.
- Example
  - max width = 200, total width = 1000

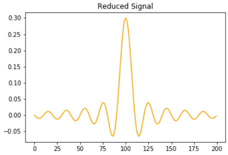


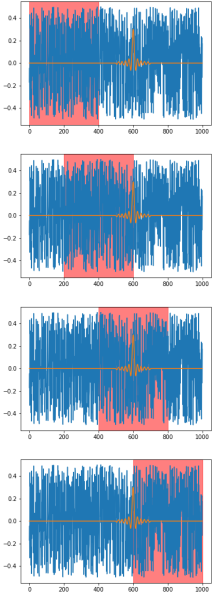


Solution

- freq range kept as twice the max width of a signal.
- Smaller input. Faster training.
- Slide over the whole range in the actual search.
  - Maximum value over the slides taken as the statistic.
- Example
  - max width = 200, total width = 1000

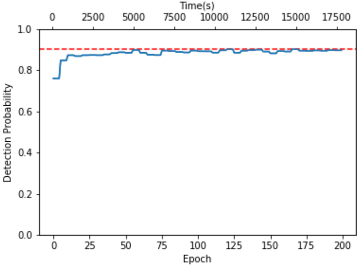


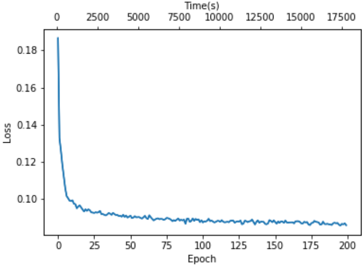


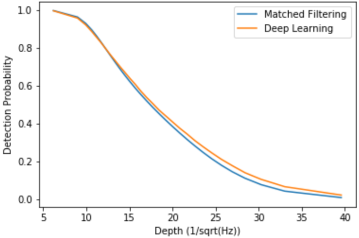


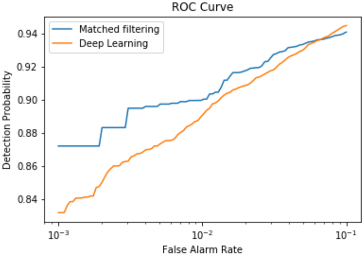
Results

- f=[20,20.005]Hz
- $\dot{f}$=[-3×10^{-9},0] Hz/s
- TSpan=1×10^5s
- depth=10.57 $\text{Hz}^{-\frac{1}{2}}$
- Matched Filtering Templates = 2.4×10^6
- Matched Filtering Time = 0.52s



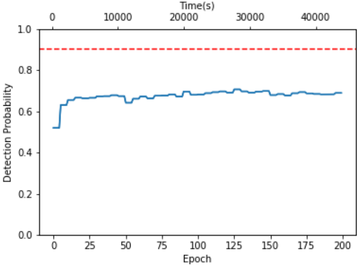


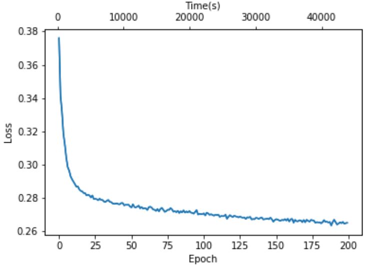


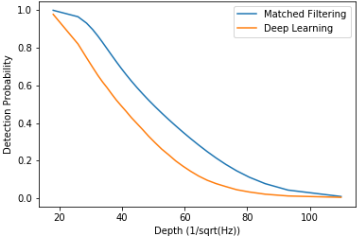


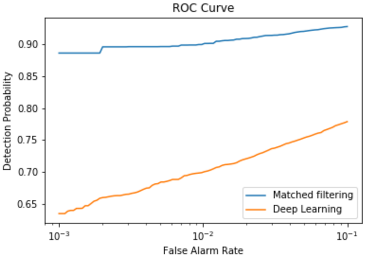
Results

- f=[20,20.001]Hz
- $\dot{f}$=[-1×10^{-10},0] Hz/s
- TSpan=1×10^6s
- depth=30.4 $\text{Hz}^{-\frac{1}{2}}$
- Matched Filtering Templates = 6.3×10^7
- Matched Filtering Time = 71s



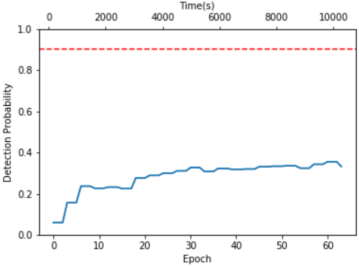


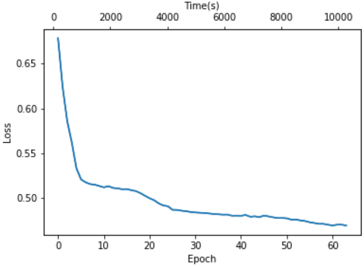


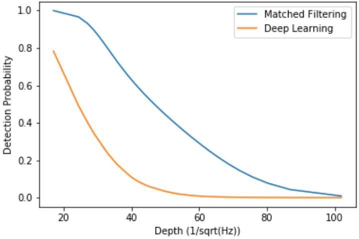


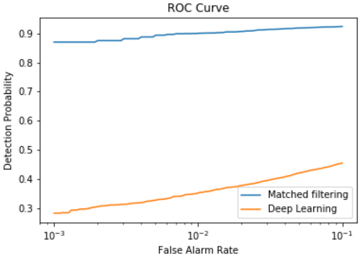
Results

- f=[100,100.001]Hz
- $\dot{f}$=[-1×10^{-10},0] Hz/s
- TSpan=1×10^6s
- depth=28.7 $\mathrm{Hz}^{-\frac{1}{2}}$
- Matched Filtering Templates = 1×10^9
- Matched Filtering Time = 1.2×10^3s







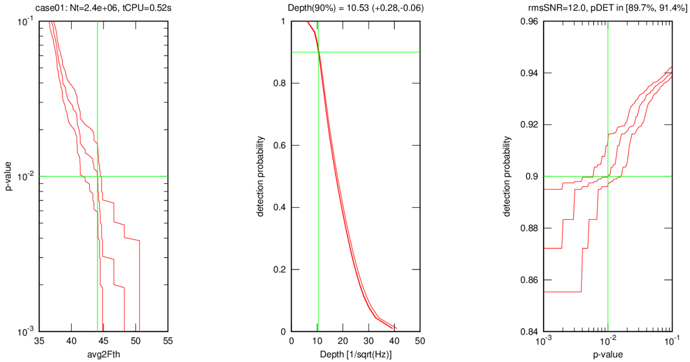


Future Work

- Find optimal network architecture for the cases.
- Better generalization over the frequencies.
    - Currently fixed narrow frequency range.
    - Generalizing over wider frequency bands.
- Parameter Estimation.

Thank You

Appendix


case01: Nt=2.4e+06, tCPU=0.52s
p-value
avg2Fth
Depth(90%) = 10.53 (+0.28,-0.06)
detection probability
Depth [1/sqrt(Hz)]
rmsSNR=12.0, pDET in [89.7%, 91.4%]
detection probability
p-value