arXiv:1701.00008

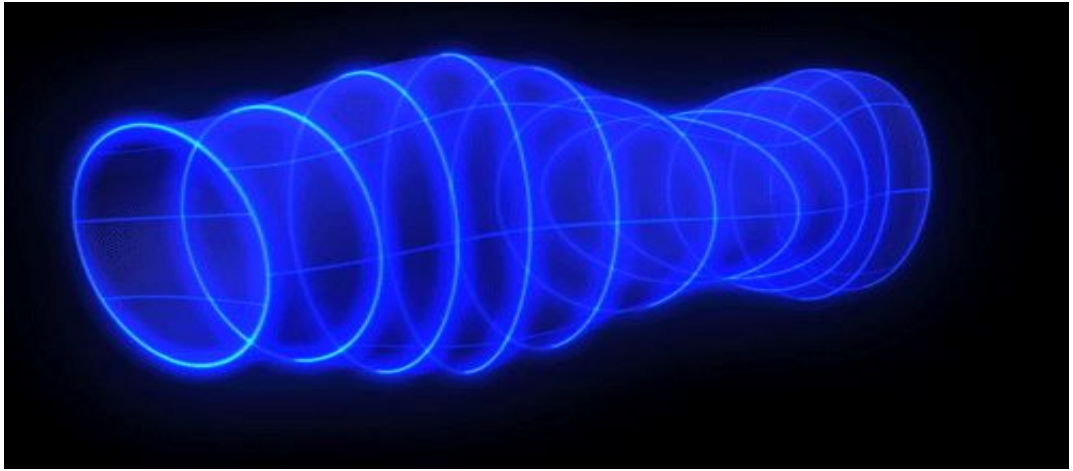
Deep Learning for Hidden Signals: Enabling Real-time Multimessenger Astrophysics

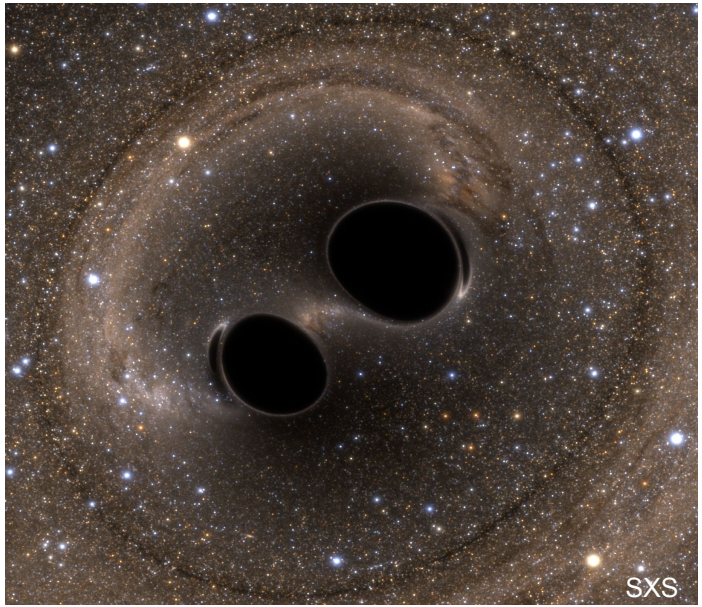
Daniel George, E. A. Huerta

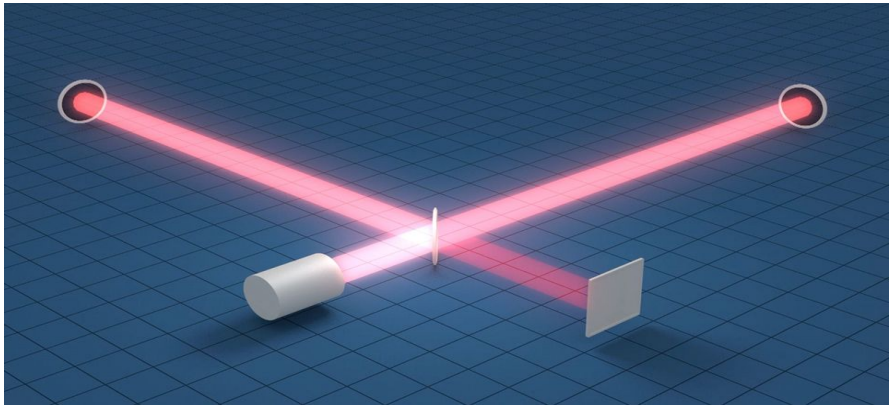
NCSA, University of Illinois at Urbana-Champaign

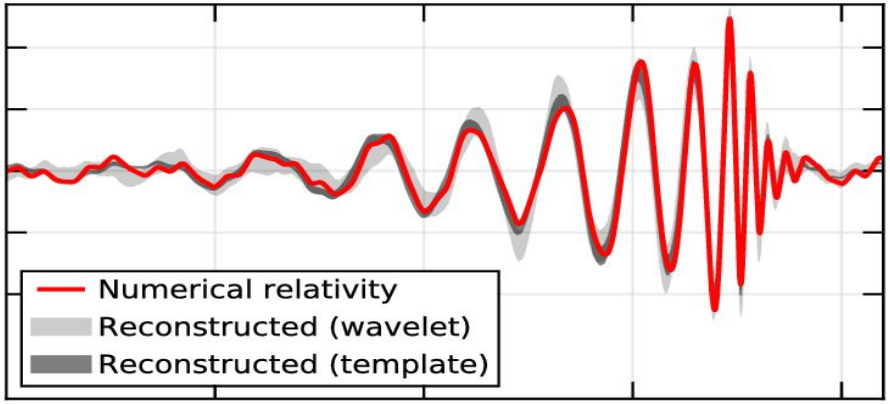
June 5, 2017

Gravitational Waves



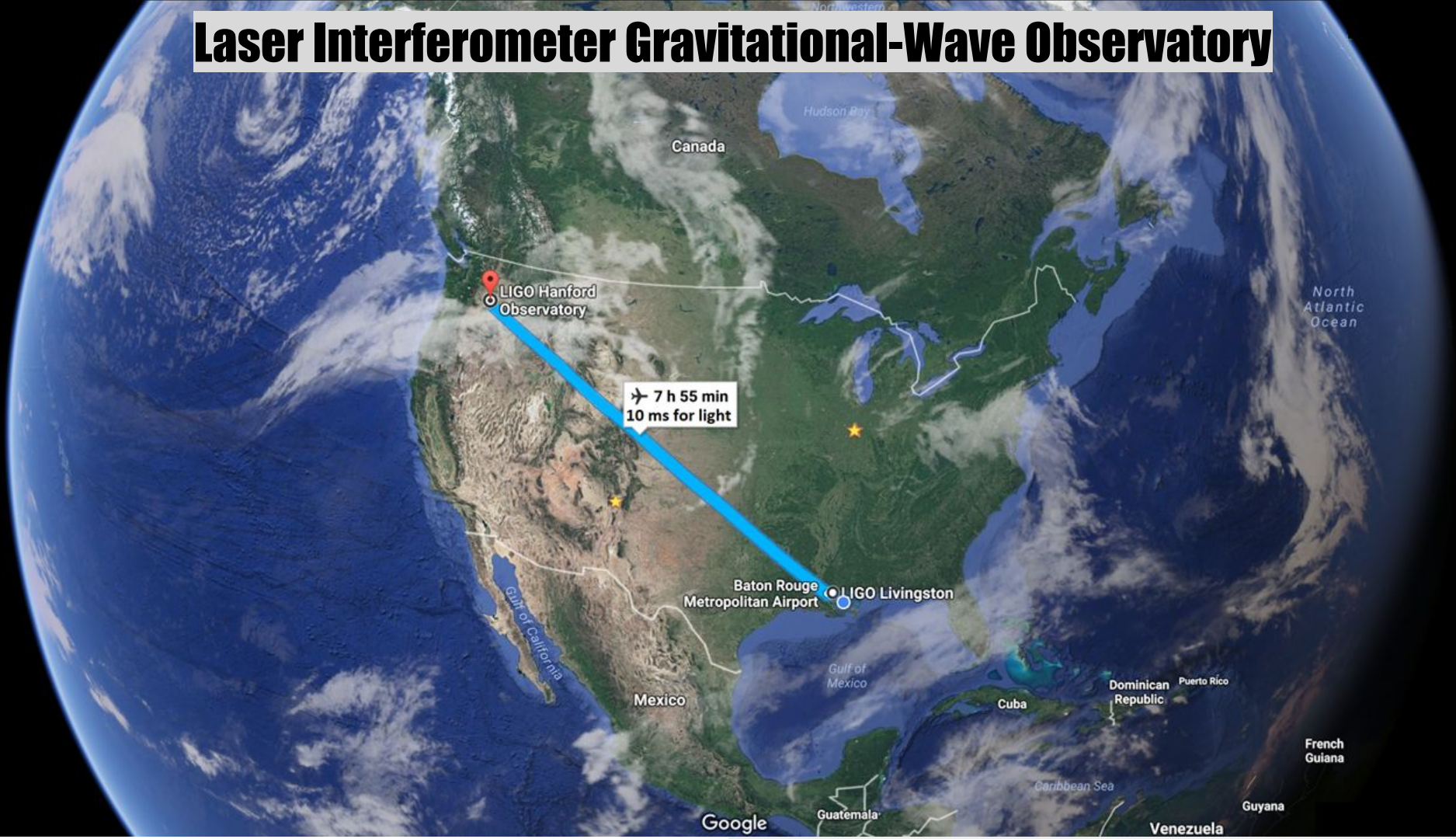


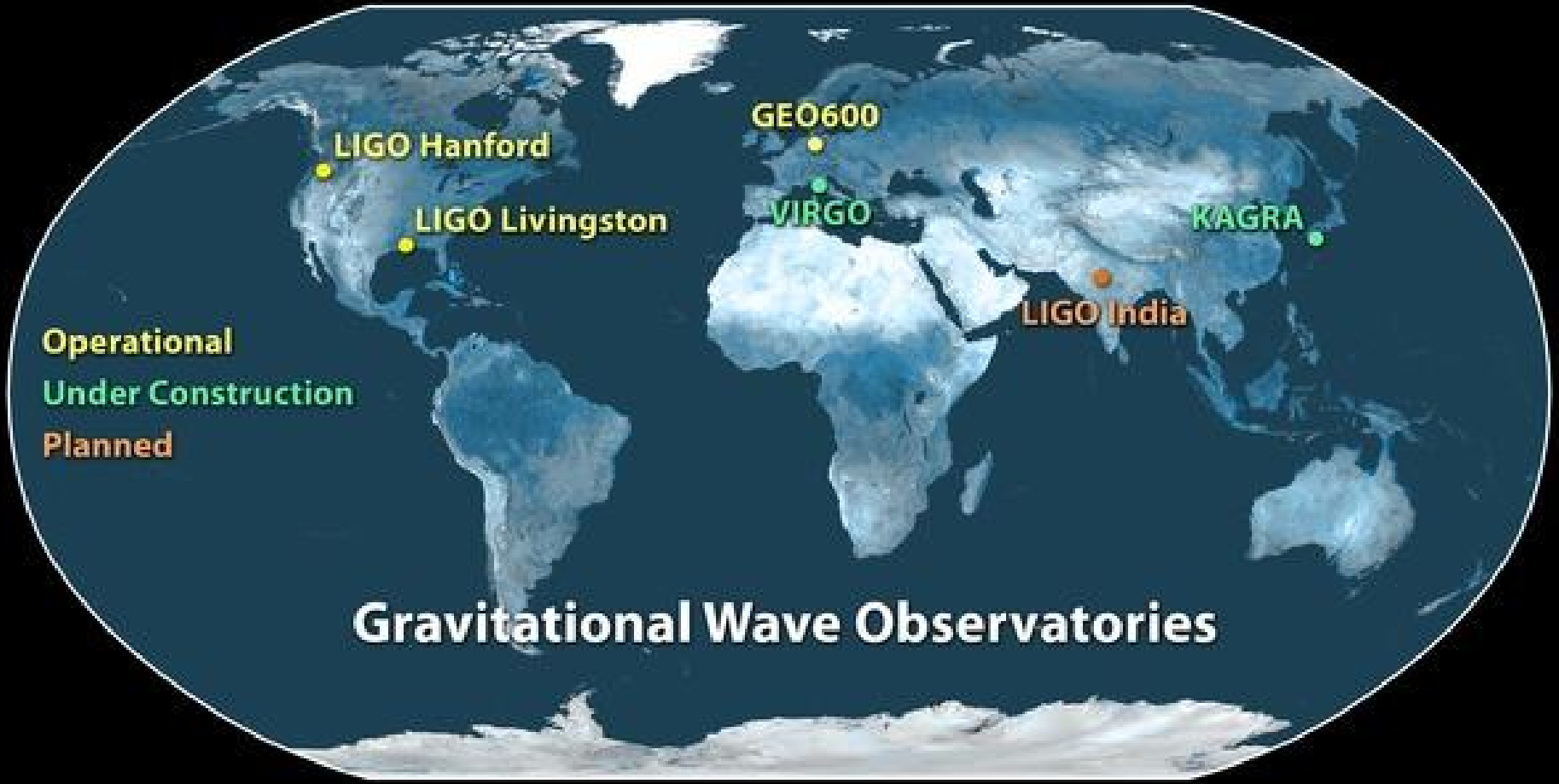




Source: ligo.org

Laser Interferometer Gravitational-Wave Observatory



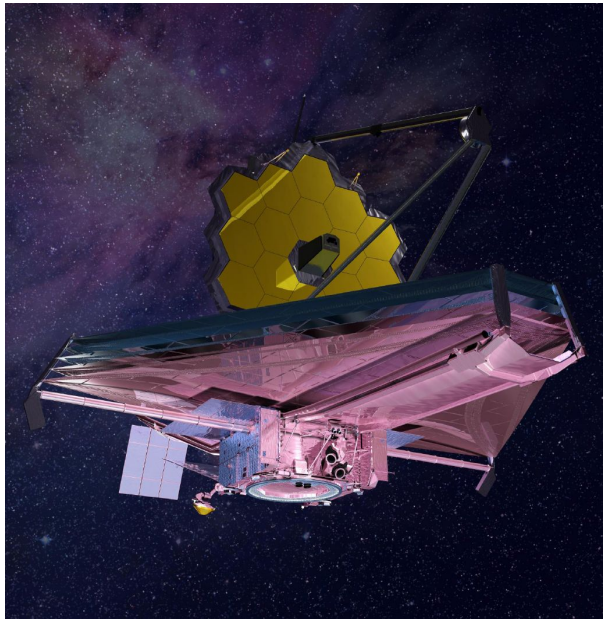

GEO600
LIGO Hanford
LIGO Livingston
VIRGO
KAGRA
LIGO India
Operational
Under Construction
Planned
Gravitational Wave Observatories

Aim: Enable Multimessenger Astrophysics

***Hear* gravitational waves**

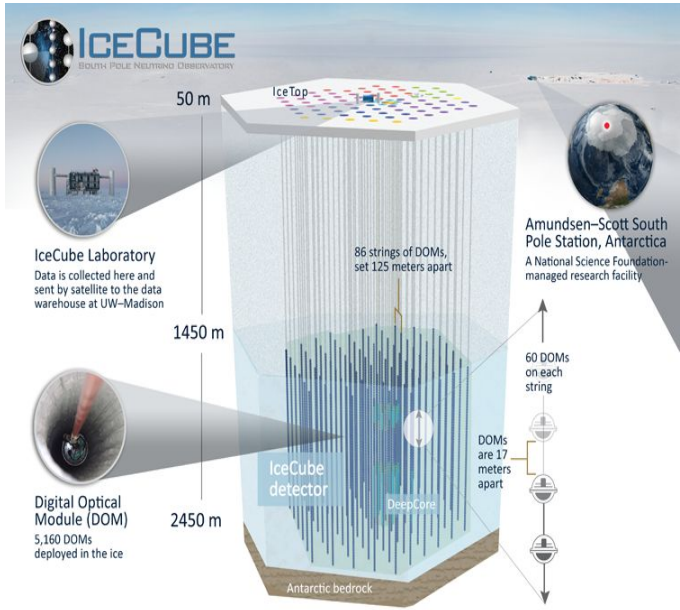
LIGO, VIRGO, KAGRA, eLISA

***See* electromagnetic waves**

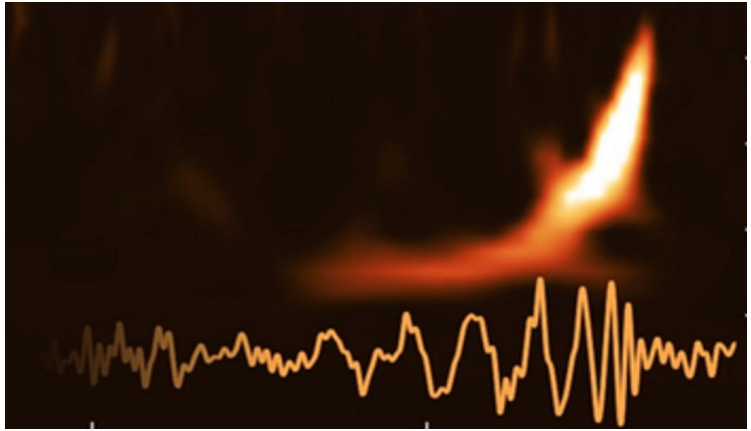


DES, LSST, JWST, WFIRST

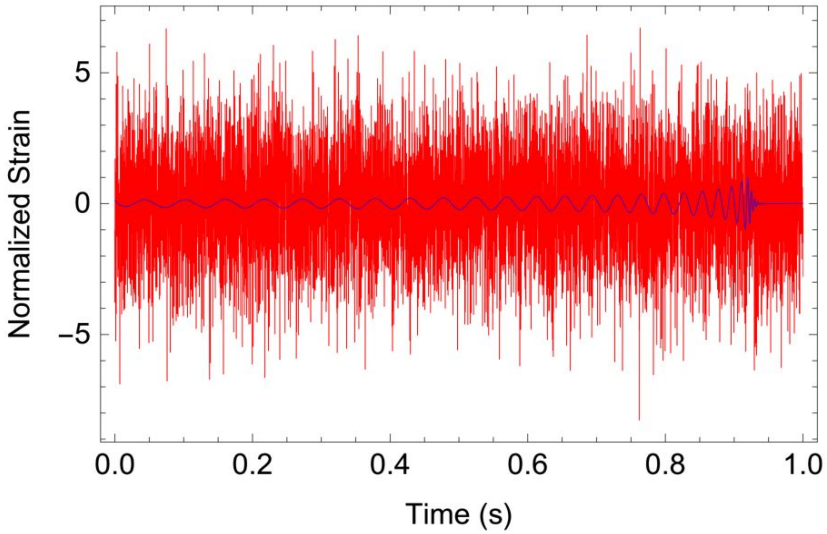
***Feel* astro-particles**

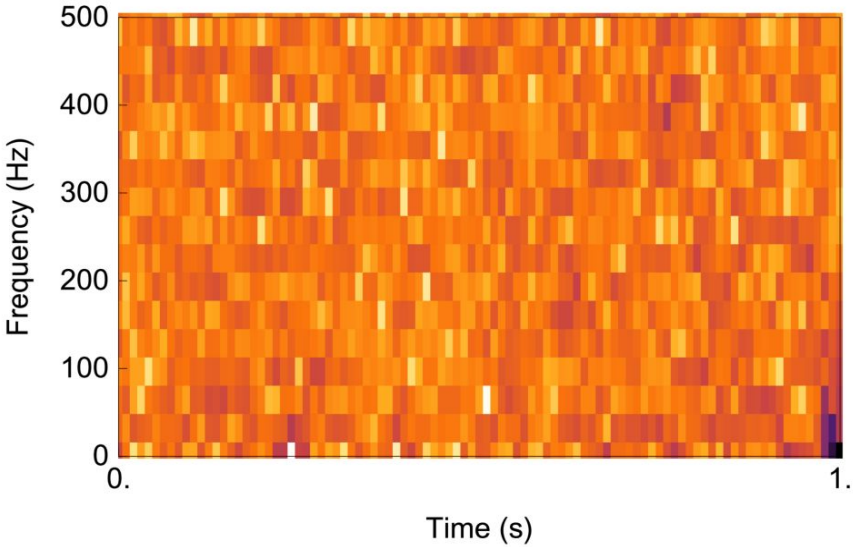


IceCube (neutrinos)



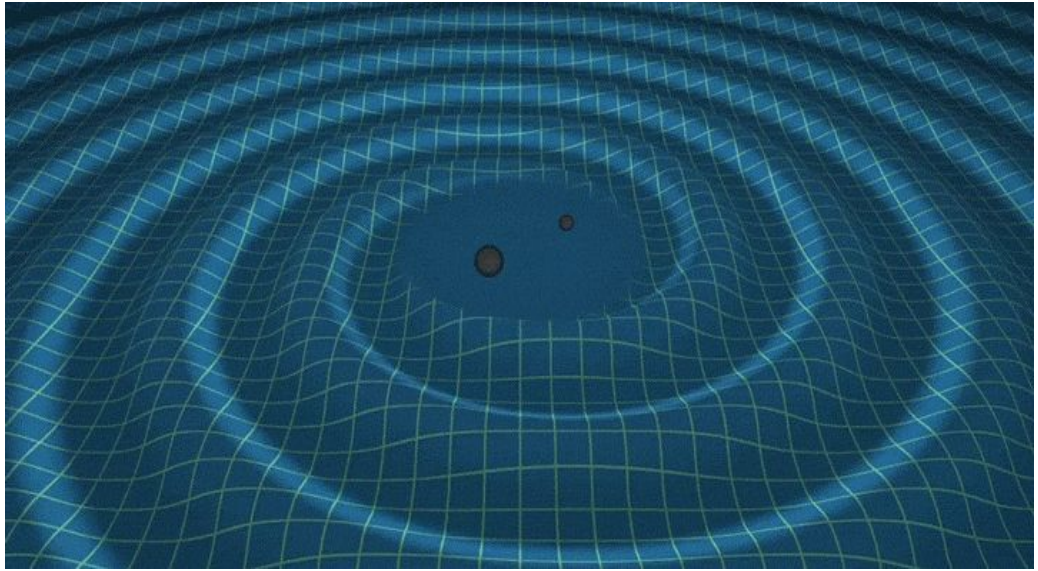
Example of LIGO Data





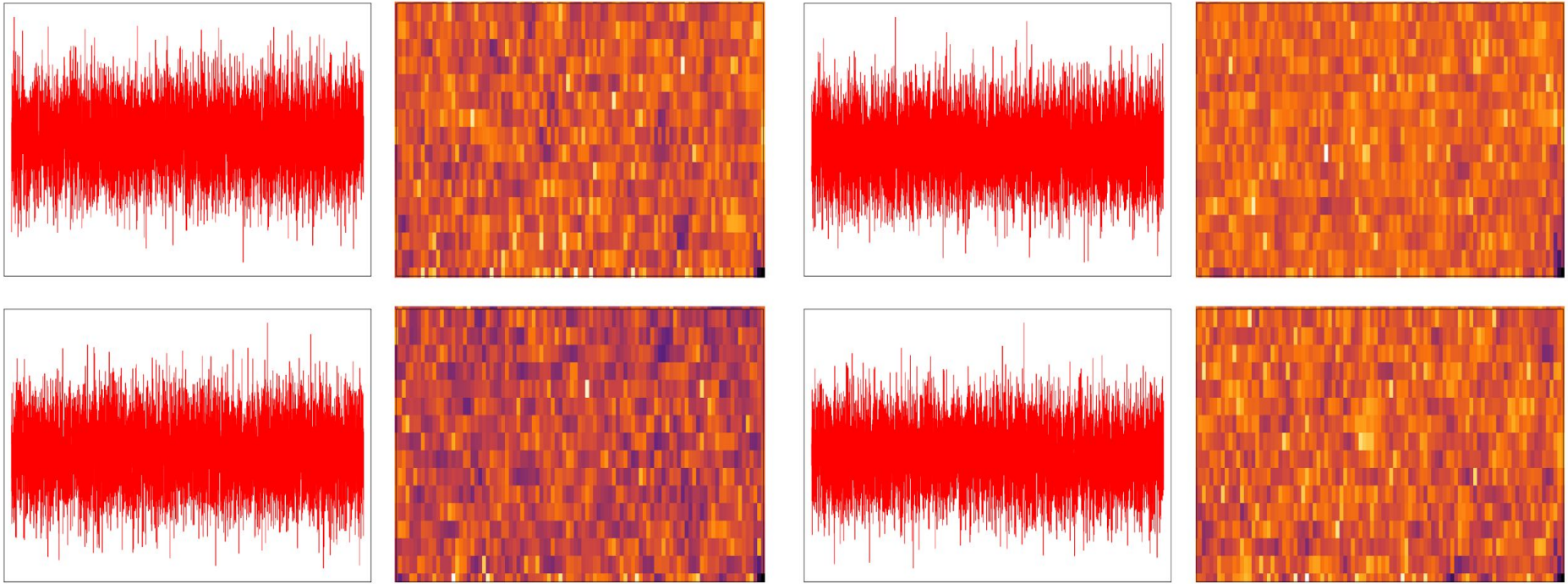
Numerical Relativity - Supercomputing

einstein toolkit

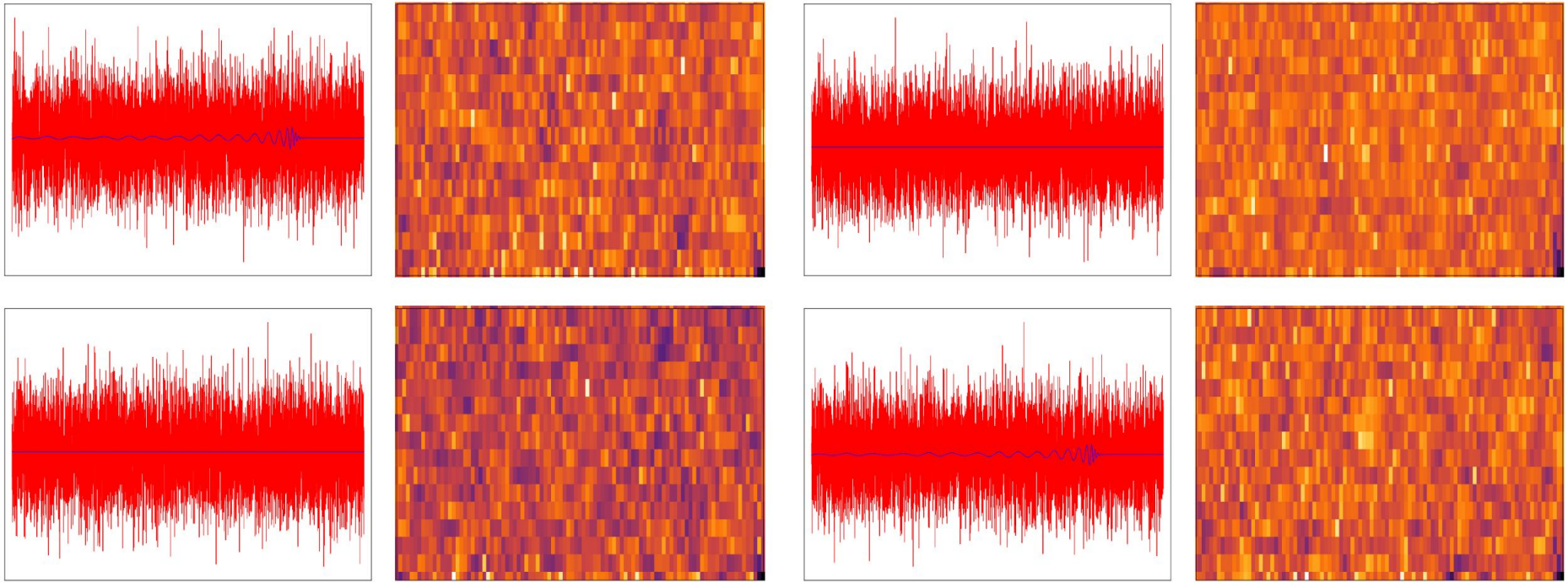


NCSA

Challenge



Solution



Bottleneck: Matched-filtering

DFJALDCATDSADFDR

Solution: Deep Learning with Neural Networks

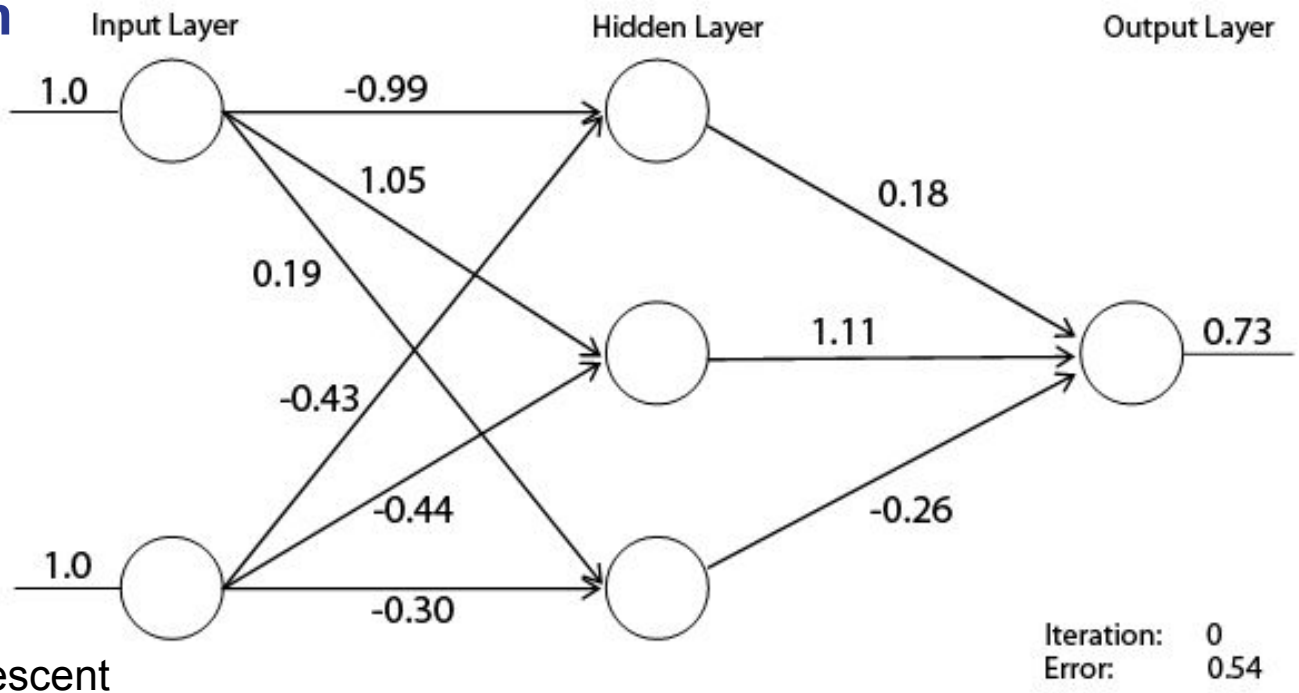
Artificial Neural Networks

Universality Theorem
Can model any function

Artificial Neurons
Weights (w) and bias (b)
Output = f(w . Input + b)

Activation
Nonlinear function (f)

Learning Algorithm
Backpropagation, steepest descent



Source: codeproject.com

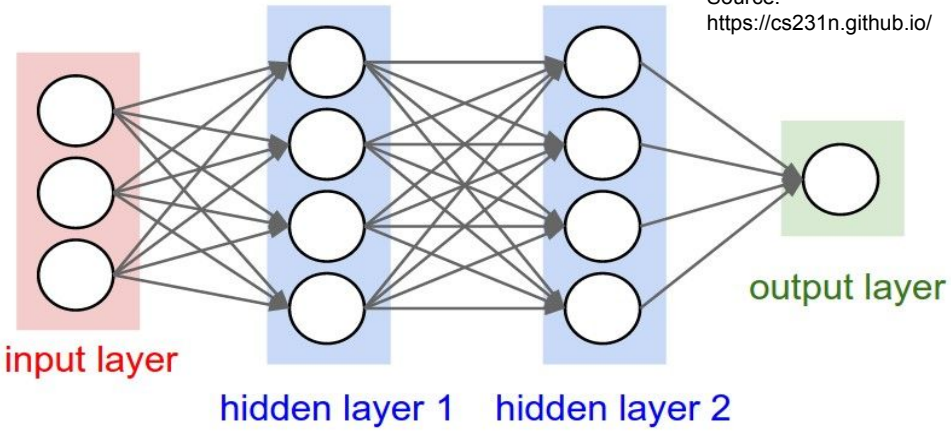
Deep Learning

Overview

- Very long networks of artificial neurons (dozens of layers)

- State-of-the-art algorithms for image processing, natural language understanding, speech recognition, web search engines, self-driving cars, etc.

Source: https://cs231n.github.io/



Representation learning

- Does not require hand-crafted features to be extracted first

- Automatic end-to-end learning

- Deeper layers can learn highly abstract functions

A person riding a motorcycle on a dirt road.

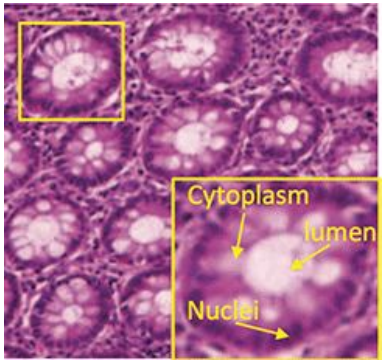
A herd of elephants walking across a dry grass field.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

amazon

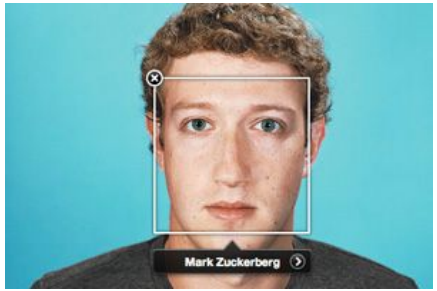
Google Translate



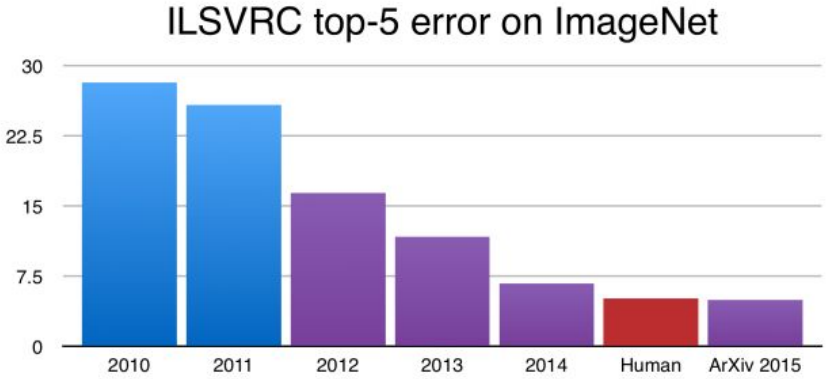
SwiftKey

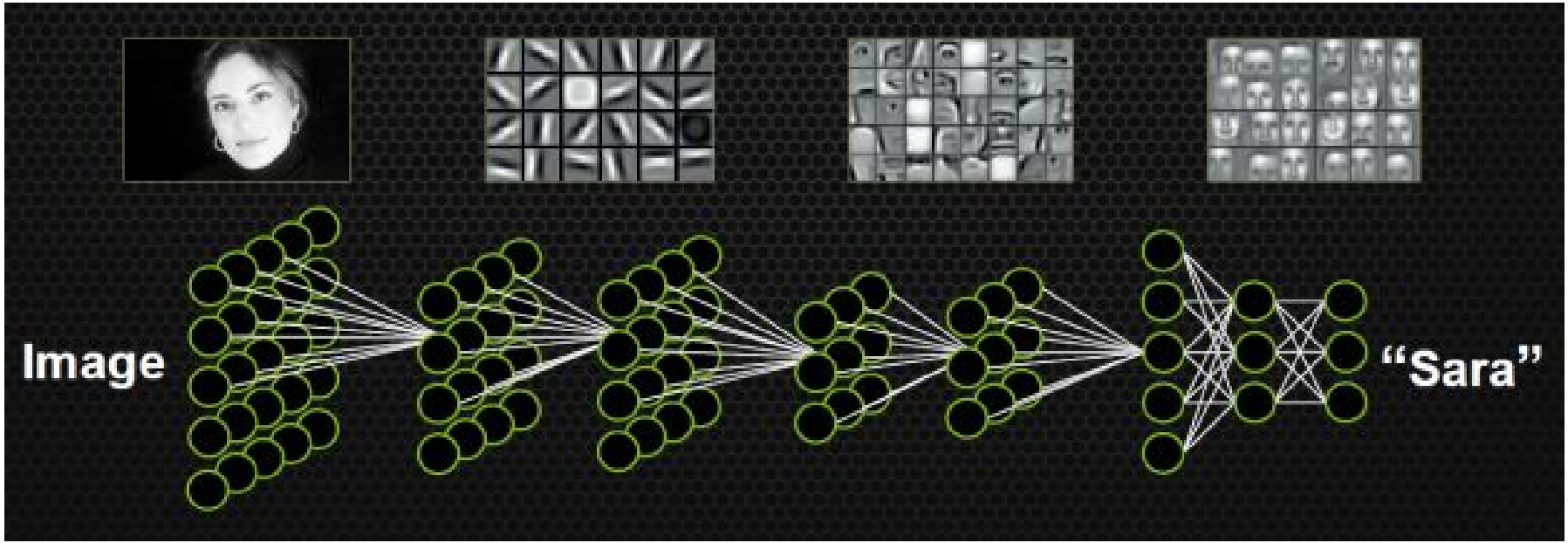
YouTube

AlphaGo

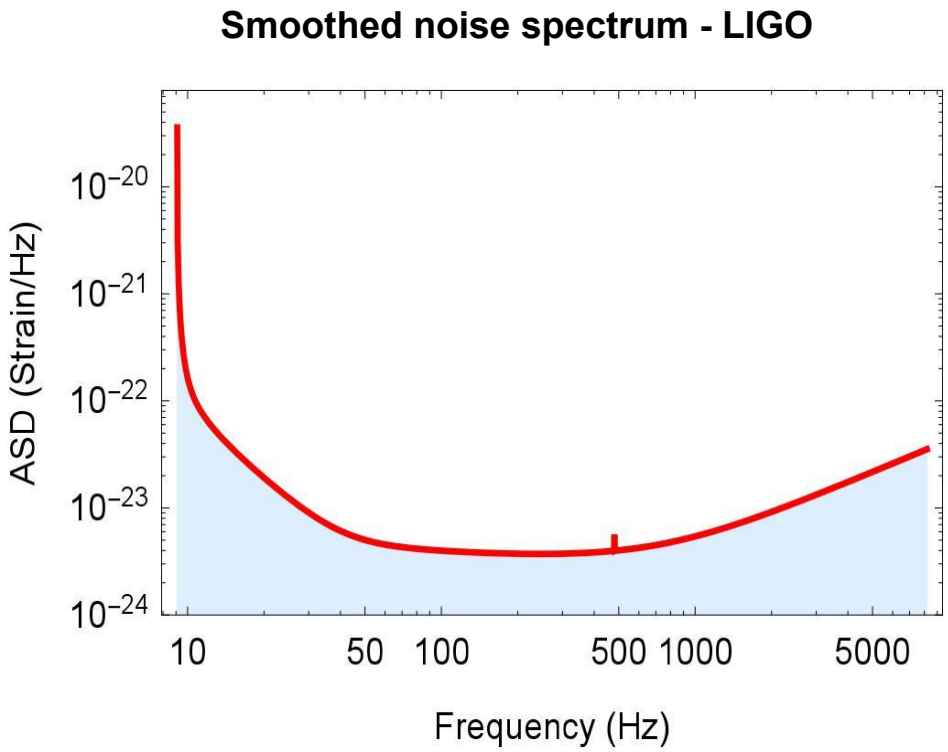


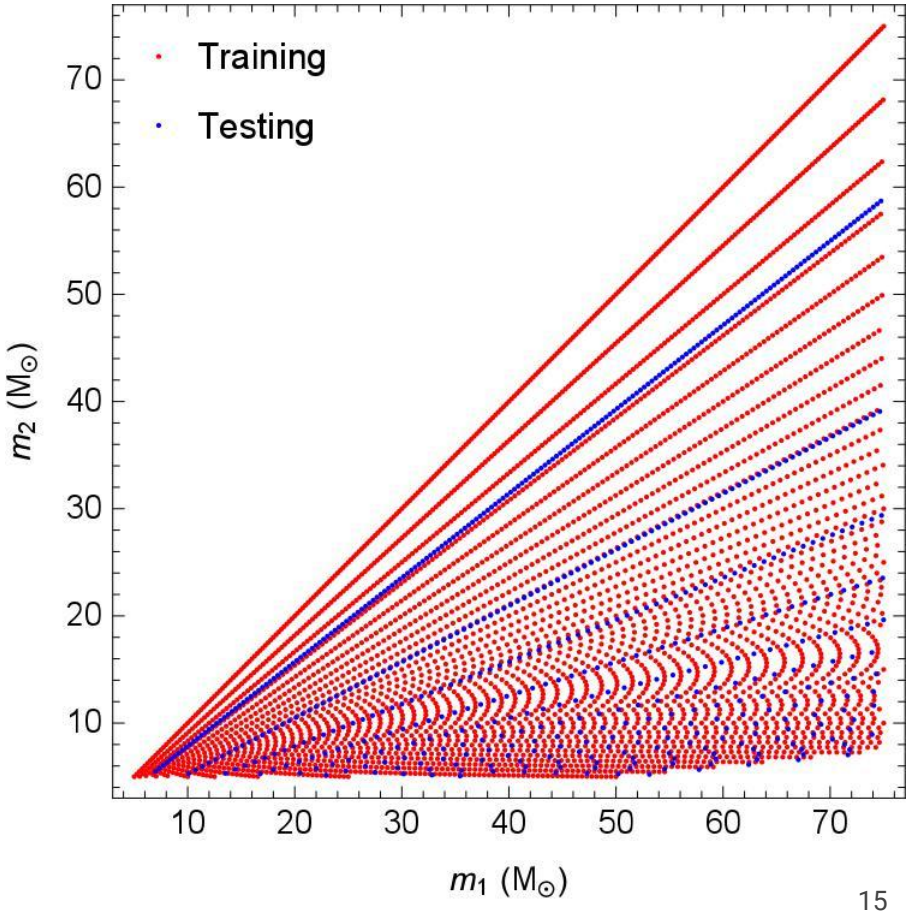
Convolutional Neural Networks





Data


Smoothed noise spectrum - LIGO
ASD (Strain/Hz)
10^{-20}
10^{-21}
10^{-22}
10^{-23}
10^{-24}
10
50
100
500
1000
5000
Frequency (Hz)


Training
Testing
m_2 ($M_\odot$)
m_1 ($M_\odot$)
10
20
30
40
50
60
70

Designing DNNs

Explored only simple designs.

Up to 4 dilated convolutional layers and 3 fully connected layers.

Separate Classifier and Predictor

Matched-filter: 1 convolutional layer

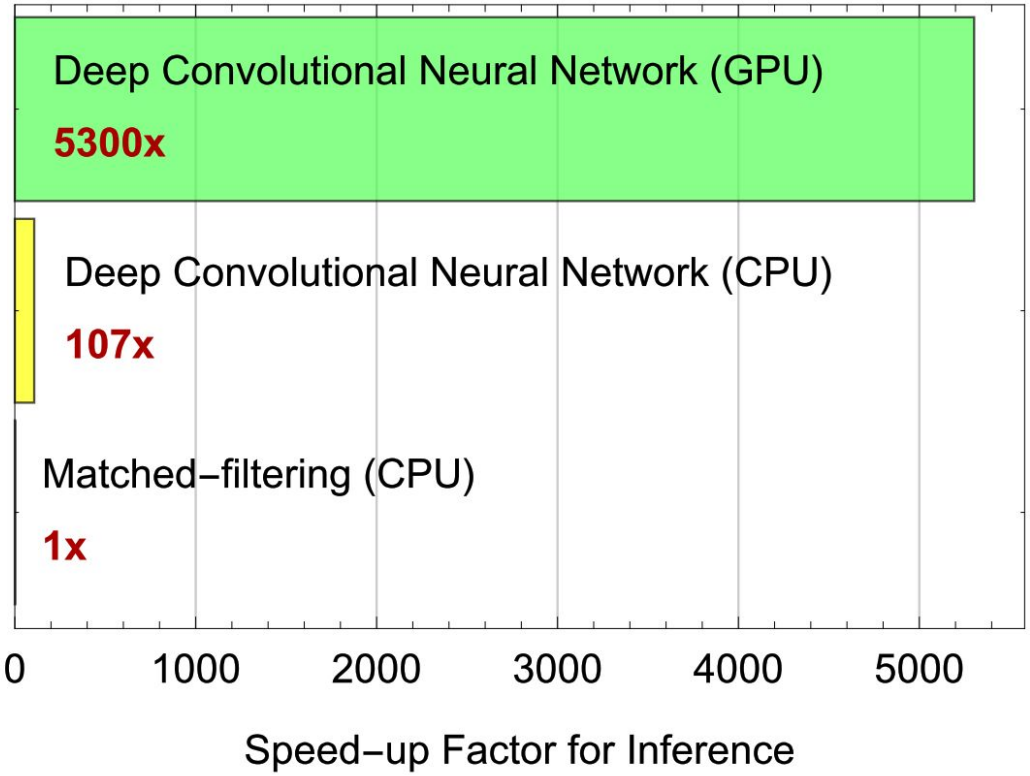
	Input	vector (size: 8192)
1	Reshape Layer	tensor (size: $1 \times 1 \times 8192$)
2	Convolution Layer	tensor (size: $16 \times 1 \times 8177$)
3	Pooling Layer	tensor (size: $16 \times 1 \times 2045$)
4	Ramp	tensor (size: $16 \times 1 \times 2045$)
5	Convolution Layer	tensor (size: $32 \times 1 \times 2017$)
6	Pooling Layer	tensor (size: $32 \times 1 \times 505$)
7	Ramp	tensor (size: $32 \times 1 \times 505$)
8	Convolution Layer	tensor (size: $64 \times 1 \times 477$)
9	Pooling Layer	tensor (size: $64 \times 1 \times 120$)
10	Ramp	tensor (size: $64 \times 1 \times 120$)
11	Flatten Layer	vector (size: 7680)
12	Linear Layer	vector (size: 64)
13	Ramp	vector (size: 64)
14	Linear Layer	vector (size: 2)
15	Softmax Layer	vector (size: 2)
	Output	vector (size: 2)

Speed-Up

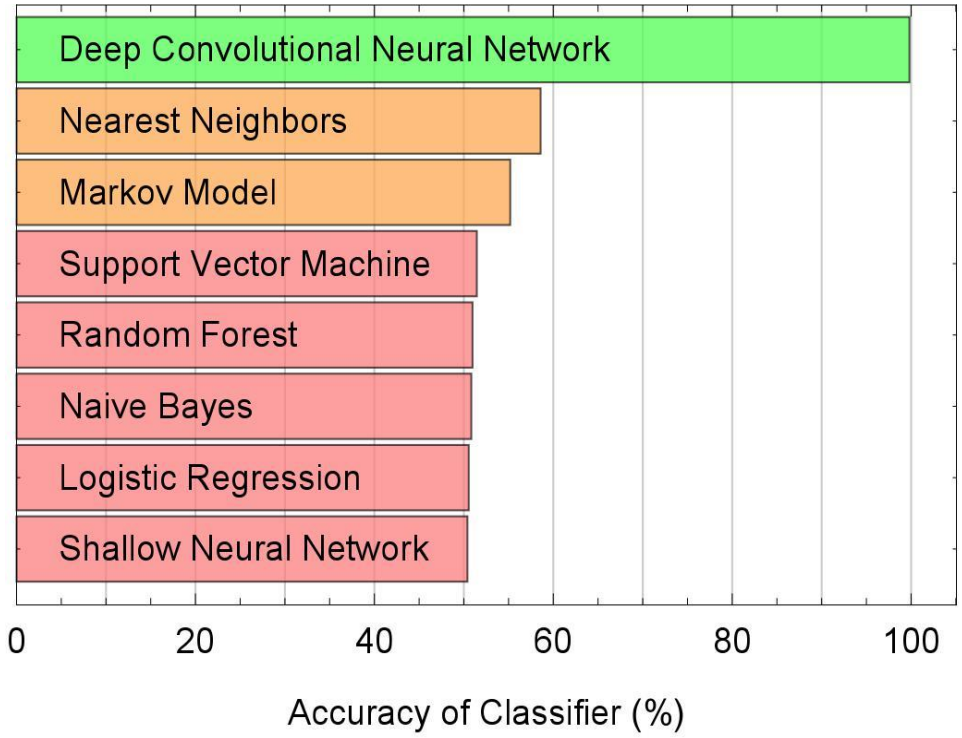
Real-time analysis (milliseconds).

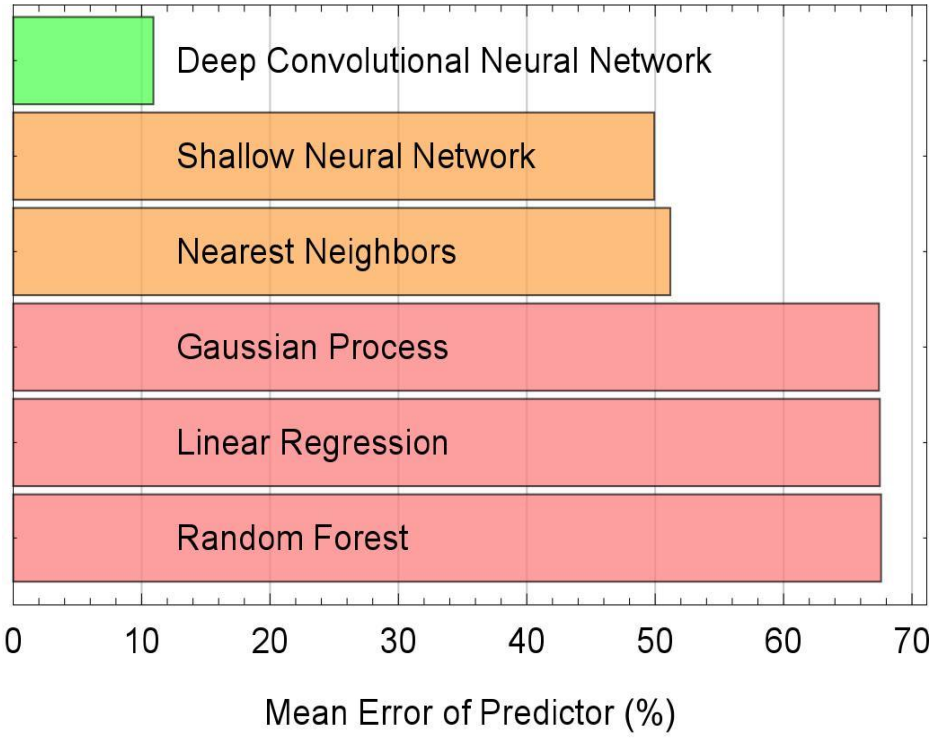
Constant time regardless of number of templates.

Thousands of inputs can be processed at once on a GPU.

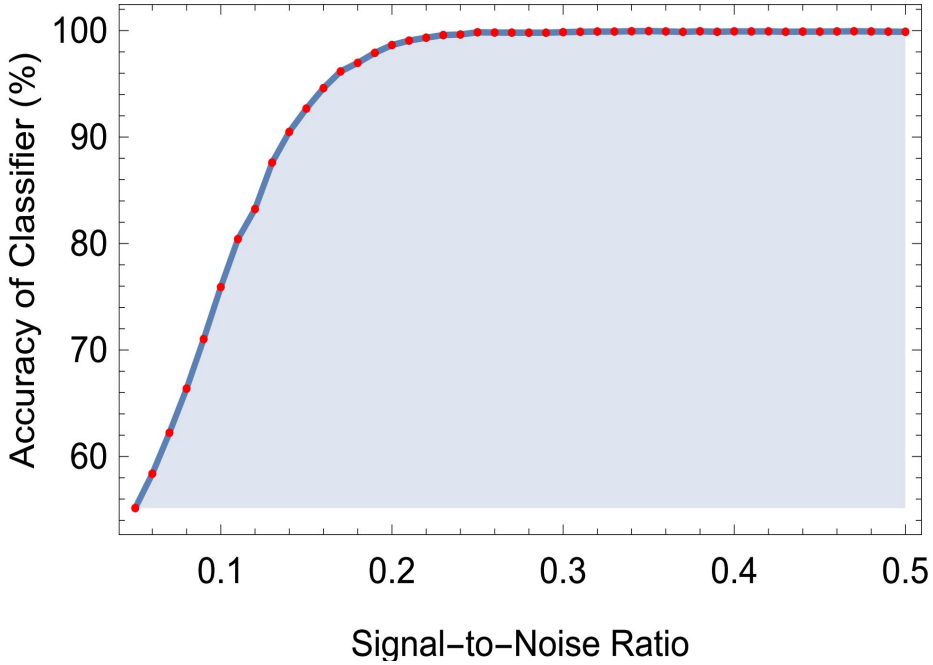


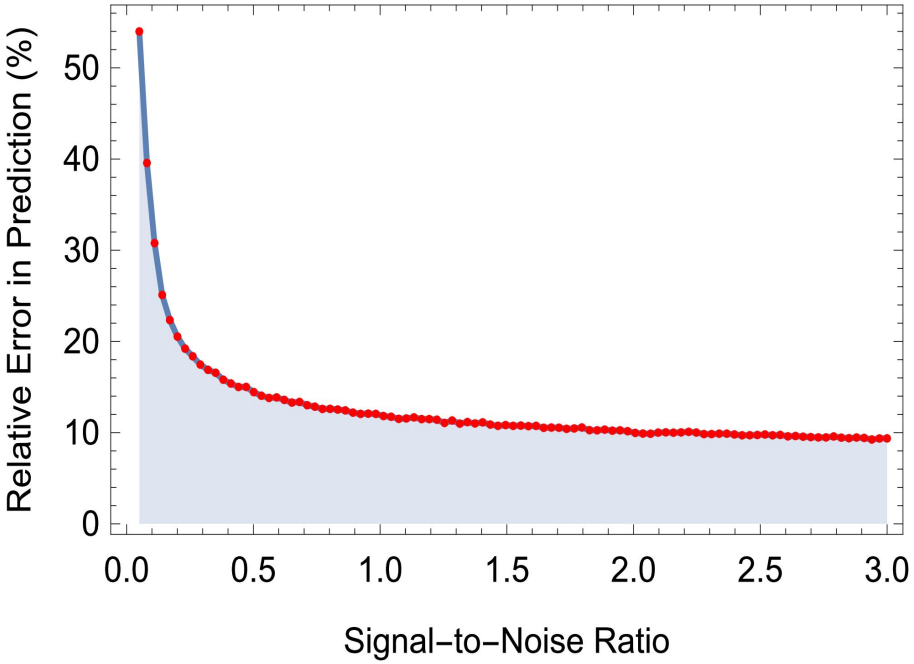
Detection and Parameter Estimation





Performance





New Types of GWs

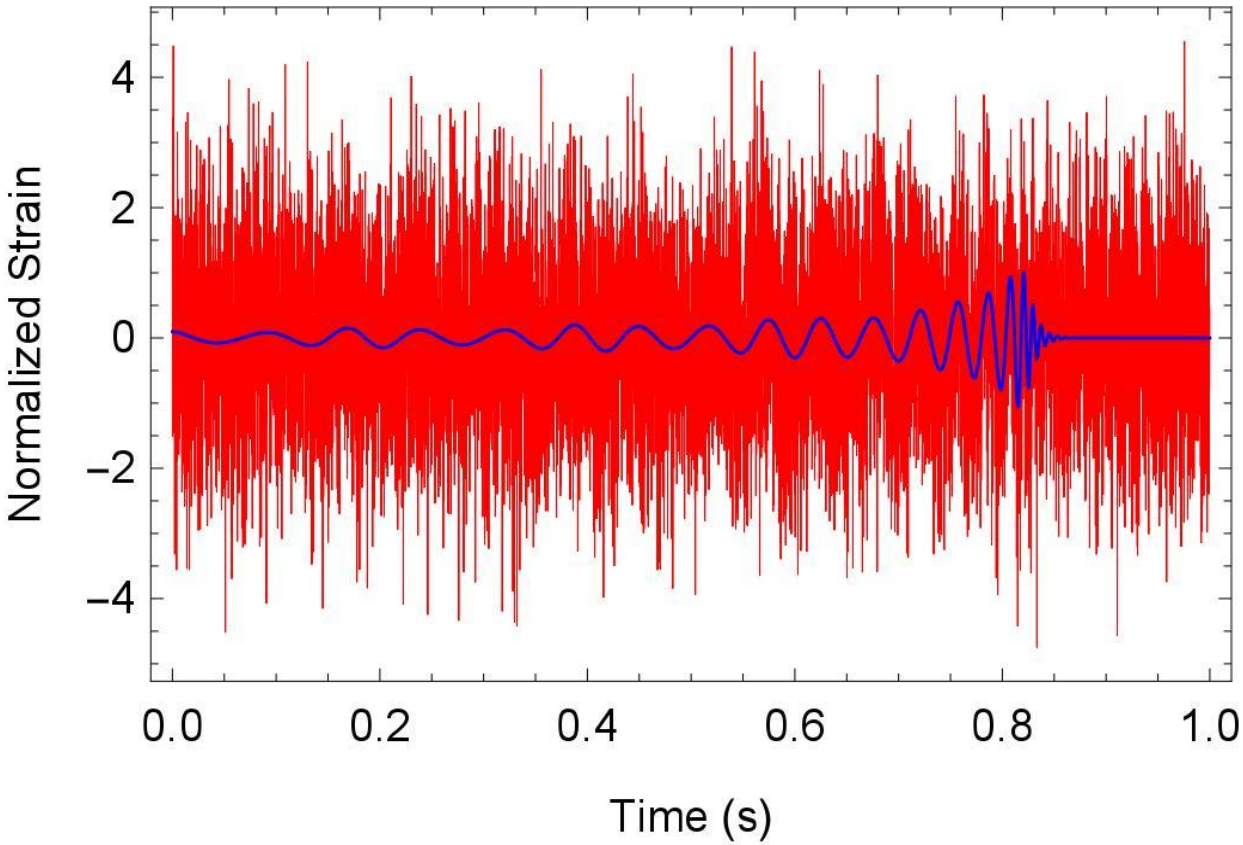
Eccentric, Spin-precessing

Not included in training.

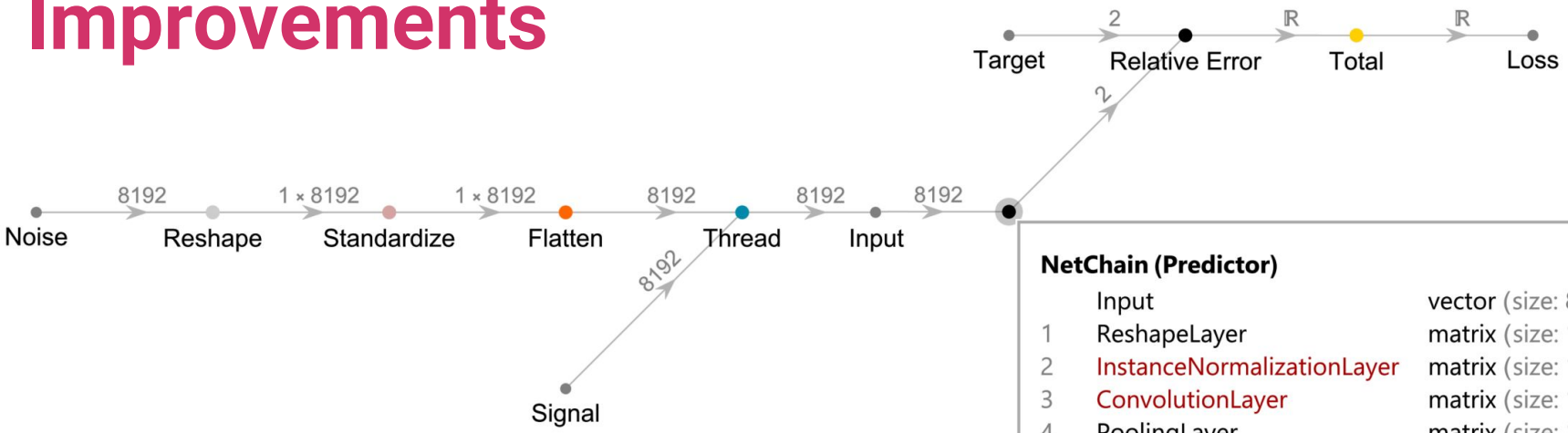
Same accuracy of detection.

DNNs learned to generalize.

Not covered by current methods.



Improvements



NetChain (Predictor)

	Input	vector (size: 8192)
1	ReshapeLayer	matrix (size: 1 × 8192)
2	InstanceNormalizationLayer	matrix (size: 1 × 8192)
3	ConvolutionLayer	matrix (size: 16 × 8177)
4	PoolingLayer	matrix (size: 16 × 2044)
5	Ramp	matrix (size: 16 × 2044)
6	ConvolutionLayer	matrix (size: 32 × 2016)
7	PoolingLayer	matrix (size: 32 × 504)
8	Ramp	matrix (size: 32 × 504)
9	ConvolutionLayer	matrix (size: 64 × 476)
10	PoolingLayer	matrix (size: 64 × 119)
11	Ramp	matrix (size: 64 × 119)
12	FlattenLayer	vector (size: 7616)
13	LinearLayer	vector (size: 64)
14	Ramp	vector (size: 64)
15	LinearLayer	vector (size: 2)
	Output	vector (size: 2)

Developed automated framework for incorporating real LIGO noise while training.

Obtained ~ 1% error for high SNR.

Used mean absolute relative error loss.

Conclusion

1. Introduced a new method for directly analyzing highly noisy time-series data
2. Potential alternative to existing methods for LIGO data analysis
3. Need catalogs of massively-parallel numerical simulations for training data
4. Extensive scope for research on deep learning and AI for science
5. Can exploit rapidly growing advances in AI algorithms and hardware