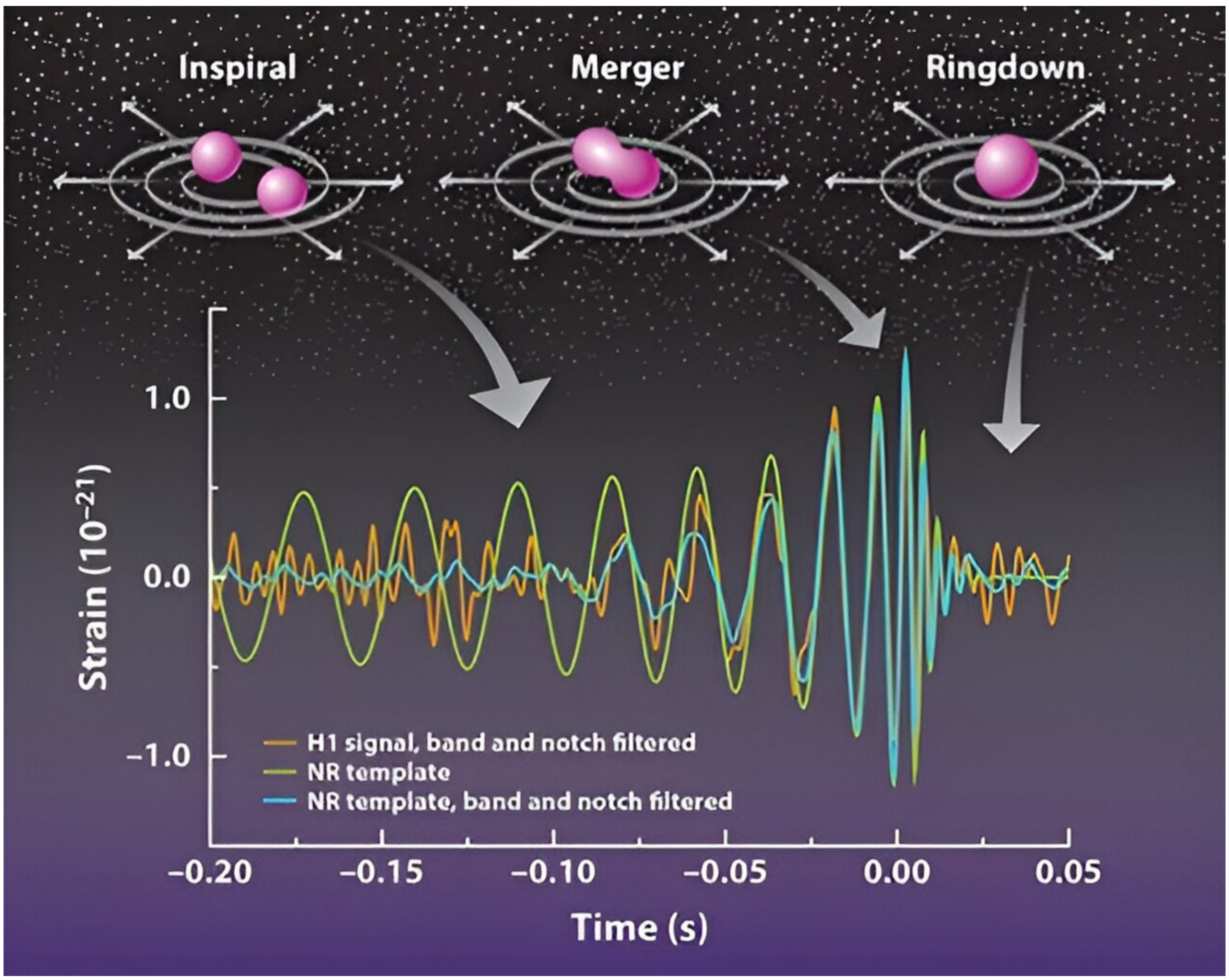
Creation of a deep learning algorithm to detect unexpected gravitational wave events

July 25 2024, by David Appell



The gravitational wave signal received at a LIGO detector (orange), overlain by a theoretical predictions from general relativity (green) and the appearance of the expected signal in the detector (blue). Credit: Physics magazine, APS [https://physics.aps.org/articles/v9/52]

Starting with the direct detection of gravitational waves in 2015, scientists have relied on a bit of a kludge: they can only detect those waves that match theoretical predictions, which is rather the opposite way that science is usually done.

Now a group of physicists have put forth a computational model that could capture all gravitational waves that pass by the Earth, instead of just the expected ones. The paper is published on the *arXiv* preprint server.

Decades after Einstein found that his general theory of relativity predicted gravitational waves—traveling ripples in the fabric of spacetime—physicists calculated their expected signatures for a few simple scenarios. One was the passing waveform for black hole-black hole mergers, which was the first such wave detected from interferometric data received on September 14, 2015. (The paper wasn't published until February of the following year.)

Assuming the event that produced the waves, gravitational scientists were able to predict the exact signal that would appear in the long-arm laser interferometric facilities such as LIGO (which has two locations in the US), VIRGO in Italy as well as several others around the world).

The observationalists needed to know what to expect in order to train their interferometers on what to look for, because a passing wave would only move the interferometer arms by a thousandth the width of a proton. Environmental noise, even passing trucks, could easily give rise to movement in the arms that had to be filtered out in order to distinguish a real gravitational wave.

Calculations were also performed for neutron star-black hole mergers and neutron star-neutron star mergers. Also, the signature of continuous gravitational waves produced by rapidly spinning symmetric neutron stars and stochastic gravitational waves from, for example, the Big Bang could be gleaned from the data. Using these models, over seven dozen gravitational wave events have been detected overall.

But this method misses gravitational waves that do not appear in the form of one of the known predictions, known as "transients" or "gravitational wave bursts," from unexpected events based on different physics. In addition, today's methods of detection are too slow.

After a gravitational wave passes, astronomers want to be able to quickly pinpoint its source in order to inform other observatories to look for any accompanying electromagnetic or particle events from the same source—known as multi-messenger astronomy.

Electromagnetic radiation, including visible light, and neutrinos are expected from certain large, violent astrophysical activity, including the usual binary pair mergers. Upon the reception of a possible gravitational wave train, processing and communication with other instruments can currently require hundreds of dedicated processing units and take tens of seconds or even minutes, too slow for a "heads-up" warning.

In recent years, physicists have been trying to improve on the waveform limitations by using convolutional neural networks (CNNs), a type of specialized deep learning algorithm, to avoid detectors trained to recognize only certain events.

However, to date, the CNNs that have been programmed still require a precise model of the target signal for training, and so won't notice unexpected sources such as those anticipated from core collapse of supernovae and long gamma-ray bursts. Both unknown physics and computational limits could ruin any chance of multi-messenger detection.

Here, researchers set a goal to use a single processor and report gravitational wave events in about a second. They developed a multi-component architecture where one CNN detects transients that are simultaneous in multiple detectors while a second CNN looks for correlation between the detectors to eliminate coincident background noise or glitches.

In this way, "our search utilizes machine learning and aims to help point the 'traditional' telescopes towards such a source in a matter of seconds," said Vasileos Skliris of the Gravity Exploration Institute at the School of Physics and Astronomy at Cardiff University in Wales, UK. "In this way, we will be able to extract the most information we can out of such unexpected events."

The group's deep-learning approach was different from previous methods in a crucial way: instead of teaching a CNN to identify specific signal shapes in the data, they created CNNs that could detect consistency in strength and timing between two or more streams of data.

The CNNs were then trained using simulated signals and random noise bursts that have similar characteristics. By using the same waveform patterns for both the signals and noise, the CNNs were prevented from relying on the pattern of the signal to make decisions; instead, the CNNs learn to evaluate how well the detectors agree with each other, allowing their models the possibility of true real-time detection of gravitational-wave transients.

As a test, they ran the observed data for the first two runs of LIGO and VIRGO and found good agreement.

"Back in the 1960s, gamma ray bursts were the novel astrophysical surprise when gamma ray astronomy took its first steps," said Skliris. "Gravitational wave astronomy is at that same early age, and we might have an exciting future ahead of us."

More information: Vasileios Skliris et al, Real-Time Detection of Unmodelled Gravitational-Wave Transients Using Convolutional Neural Networks, *arXiv* (2020). DOI: 10.48550/arxiv.2009.14611

Citation: Creation of a deep learning algorithm to detect unexpected gravitational wave events (2024, July 25) retrieved 26 July 2024 from https://phys.org/news/2024-07-creation-deep-algorithm-unexpected-gravitational.html