



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

BLENDA ÚLIMA RODRIGUES CESAR GUEDES

COMPACT BINARY SYSTEM'S MASS INFERENCE
FROM GRAVITATIONAL WAVES VIA ARTIFICIAL
NEURAL NETWORKS

RECIFE – PE

2022

BLENDA ÚLIMA RODRIGUES CESAR GUEDES

**COMPACT BINARY SYSTEM'S MASS INFERENCE
FROM GRAVITATIONAL WAVES VIA ARTIFICIAL
NEURAL NETWORKS**

Dissertation submitted to Coordenação do
Programa de Pós-Graduação em Informática
Aplicada of Departamento de Estatística e
Informática - DEINFO - Universidade Federal
Rural de Pernambuco, as part of requirements
to obtain the Master Science Degree.

ADVISOR: Prof. Dr. Tiago Alessandro Espínola Ferreira

COADVISOR: Prof. Dr. Antônio de Pádua Santos

RECIFE – PE

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- G924i Guedes, Blenda Úlima Rodrigues Cesar
Inferência da Massa de Sistemas Binários Compactos a partir de Ondas Gravitacionais via Redes
Neurais Artificiais / Blenda Úlima Rodrigues Cesar Guedes. - 2022.
63 f. : il.
- Orientador: Tiago Alessandro Espinola Ferreira.
Coorientador: Antonio de Padua Santos.
Inclui referências e apêndice(s).
- Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em
Informática Aplicada, Recife, 2021.
1. Ondas Gravitacionais. 2. Redes Neurais. 3. Astroinformática. I. Ferreira, Tiago Alessandro Espinola,
orient. II. Santos, Antonio de Padua, coorient. III. Título

BLENDA ÚLIMA RODRIGUES CESAR GUEDES

COMPACT BINARY SYSTEM'S MASS INFERENCE
FROM GRAVITATIONAL WAVES VIA ARTIFICIAL
NEURAL NETWORKS

Dissertation submitted to Coordenação do
Programa de Pós-Graduação em Informática
Aplicada of Departamento de Estatística e
Informática - DEINFO - Universidade Federal
Rural de Pernambuco, as part of requirements
to obtain the Master Science Degree.

EXAMINATION BOARD

Prof. Dr. Tiago Alessandro Espínola Ferreira (Advisor)
Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática

Prof. Dr. Antônio de Pádua Santos
Universidade Federal Rural de Pernambuco - UFRPE
Departamento de Física

Prof. Dr. Antônio Carlos Miranda
Universidade Federal Rural de Pernambuco - UFRPE
Departamento de Física

Dedico este trabalho a minha mãe, Núbia
Carla, que fez a saudade ser uma companheira
diária.

Agradecimentos

Diante de tempos tão difíceis agradecer se tornou uma tarefa fácil.

Obrigada a meus pais, Núbia e Leonardo, por trazer-me a essa aventura que é a vida e dar-me asas para que eu voe além dos campos conhecidos.

Obrigada as minhas avós, Lúcia e Zenilda, pelo grande suporte e amor dados ao longo dos anos.

Obrigada meus queridos Gustavo, Rebeka, Ricardo e Sarah, pela amizade, compreensão, alegria, torcida e suporte em tempos difíceis e felizes. Dividir a vida com vocês é um grande prazer.

Obrigada a minha tia Sandra pela palavras de conforto e afeto de uma mãe, obrigada também ao meu tio Abel, e novamente a Sarah, com vocês o caminho se torna iluminado e a esperança renasce novamente.

Obrigada a minha querida amiga Cris, pelas várias horas de suporte e estudos, sem você esse mestrado não vingaria.

Finalmente agradeço ao meu querido professor Tiago, o senhor é o melhor professor do mundo, e traz consigo brilhantismo e uma inteligência encantadora. Sua forma de ensinar é inspiradora. Também sua paciência e resiliência diante da nossa ciência nos dar forças e esperança para continuar o tão digno trabalho de ensinar no nosso país.

Feminism Is for Everybody.

(Bell Hooks)

Abstract

Before the detection of gravitational waves by LIGO in 2015, a new era of possible knowledge about the cosmos has emerged. With this, the possibility of analyzing these detection data and refining techniques that can characterize it. Therefore, we suggest the development of an algorithm capable of characterizing gravitational wave signals, more precisely inferring the gravitational wave masses caused by binary systems, and we use the low computational cost approach. With that, we analyse the simulated signals of gravitational waves, using only the spiral phase of the wave, applying artificial neural networks to analyze these signals, as the output of the network we obtain the masses value of both objects. As a step of sophistication of our algorithm, we use signals from waves detected by LIGO to consolidate its suitability for real data. As a preliminary result of this experiment, we were able to infer the data with minimized error in up to 12% of difference with the data presented by LIGO.

Keywords: Gravitational waves; Neural networks; Astroinformatics.

Resumo

Diante da detecção de ondas gravitacionais pelo LIGO, em 2015, uma nova era de possível conhecimento sobre o cosmos surgiu. Com isso a possibilidade de análise desses dados de detecção e refinamento de técnicas que caracterizam essas detecções. Deste modo sugerimos o desenvolvimento de um algoritmo capaz de caracterizar sinais de ondas gravitacionais, mais precisamente inferir as massas de ondas gravitacionais provocadas por sistemas binários, e utilizamos a abordagem de baixo custo computacional. Com isso analisando os sinais simulados de ondas gravitacionais, fazendo uso apenas da fase espiral da onda, por seguinte aplicamos redes neurais artificiais para analisar estes sinais, como saída da rede obtemos o valor de ambas as massas. Como um passo de sofisticação do nosso algoritmo, utilizamos sinais de ondas detectadas pelo LIGO para consolidar sua aptidão para dados reais. Como resultado preliminar deste experimento conseguimos inferir os dados com erro minimizado em até 12% de diferença com os dados apresentados pelo LIGO.

Palavras-chave: Ondas gravitacionais; Redes neurais; Astroinformática.

List of Figures

| | |
|---|----|
| Figure 1 – The bend that the Sun causes in space-time and the smaller bend caused by the Earth. Credit: (T. Pyle/Caltech/MIT/LIGO Lab, 2016) | 20 |
| Figure 2 – The spiral gravitational waves are formed by two black holes. Credit: (LIGO/T. Pyle, 2016) | 21 |
| Figure 3 – Phases of a binary system. Blue continuous line - gravitational wave signal from the spiral phase Equation 2.1. Red dashed line - ring down stage Equation 2.8 | 22 |
| Figure 4 – Simplified diagram of an Advanced LIGO detector (not to scale). Credit: (ABBOTT; AL., 2016) | 25 |
| Figure 5 – Illustrating how the twin LIGO observatories work. Credit: (LIGO/T. Pyle, 2016) | 26 |
| Figure 6 – The Architecture of a Multilayer Perceptron Artificial Neural Network. | 29 |
| Figure 7 – Different activation functions used in the research. | 30 |
| Figure 8 – LIGO and Virgo have observed the merger of several compact object binaries. The black holes represent a new population with masses that are larger than what had been seen previously with X-ray studies alone. This plot shows the masses of the binary components before the merger, as well as the mass of the merger remnant. Credit: (LIGO-Virgo/Northwestern Univ./Frank Elavsky, 2020) | 35 |
| Figure 9 – The disparity of points between two wave signals. | 36 |
| Figure 10 – Histogram of the waves with a different mass chirp. Now with the same quantitative points. | 36 |
| Figure 11 – Distribution of Training data set. | 38 |
| Figure 12 – Loss Graph of the 9 st neural network on the ranking, in a logarithmic view. The network has 2 hidden layers with 50 nodes each, using ReLU as the activation function and $1e10^{-3}$ as the learning rate. On blue is the Training score, and on red is the validation score, showing the balance between both data and continuous algorithm generalization during the training process. | 40 |

| | |
|--|----|
| Figure 13 – Zoom on the Loss Graph of the 9 st neural network on the ranking. The last 200 points. | 40 |
| Figure 14 – Histogram of the difference between target and prediction using the best found neural network. On the left (blue color) we have the first mass (M1) difference and on the right (red color) we have the second mass (M2) | 41 |
| Figure 15 – LIGO waves used in the analysis from the <i>PyCBC</i> library. | 42 |
| Figure 16 – Histogram of the difference between target and prediction using the best found neural network. On the left (blue color) we have the first mass (M1) difference and on the right (red color) we have the second mass (M2) | 43 |
| Figure 17 – Zoom on a wave with M1 and M2 mass equal to $20 M_{\odot}$ | 44 |
| Figure 18 – Simulated waveform from PyCBC library. | 45 |
| Figure 19 – Waveform simulated from PyCBC library used as input to the Neural Network. It has exactly 409 points. Starts -0.1 second before the objects merge and end at the exact point of the merge. | 45 |
| Figure 20 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right. | 47 |
| Figure 21 – LIGO Data test results histograms. On the left (blue) is the M1, and on the right (red) is the M2. There is a normal wave fitted to the data, it is possible to see it does not lay on the data well. For the M1 the mean is 10.80 and the standard derivation 10.24. For M2 the mean is 17.39 and the standard derivation 13. | 48 |
| Figure 22 – Test data error histogram for Second Round. First mass on the left and the second mass on the right. | 50 |
| Figure 23 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right. | 51 |
| Figure 24 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right. | 52 |
| Figure 25 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right. | 52 |

Figure 26 – LIGO Data test results histograms. On the left (blue) is the M1, and on the right (red) is the M2. There is a normal wave fitted to the data, it is possible to see it does not lay on the data well. For the M1 the mean is -2.62 and the standard derivation 4.44 . For M2 the mean is -3.85 and the standard derivation 4.75 53

List of Tables

| | |
|--|----|
| Table 1 – Neural Networks Architecture Configuration | 37 |
| Table 2 – The top 10 Neural Network Performance with 1000 epochs | 39 |
| Table 3 – All 8 Neural Network Performance with 5000 epochs | 41 |
| Table 4 – Results with LIGO Data - Round 1 | 43 |
| Table 5 – Second Round Experiment - Preliminary Result | 46 |
| Table 6 – Second Round Experiment - Early Stop | 47 |
| Table 7 – Results with LIGO Data - Round 2 | 48 |
| Table 8 – Third Round Experiment - Preliminary | 51 |
| Table 9 – Third Round Experiment - Early Stop | 52 |
| Table 10 – Results with LIGO Data - Phase 3 | 53 |

List of symbols

| | |
|-------------|---|
| MSE | Mean Squared Error |
| LIGO | Laser Interferometer Gravitational-Wave Observatory |
| ANN | Neural Network |
| CBIC | Congresso Brasileiro de Inteligência Computacional |
| M1 | First Mass |
| M2 | Second Mass |
| M_{\odot} | Solar Mass |

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 15 |
| 1.1 | Justification | 16 |
| 1.2 | Related Works | 17 |
| 1.3 | Objective | 18 |
| 1.3.1 | General Objective | 18 |
| 1.3.2 | Specific Objective | 19 |
| 1.4 | Document Structure | 19 |
| 2 | Gravitational Waves | 20 |
| 2.1 | Gravitational Waves of a Binary System | 22 |
| 2.2 | Gravitational Wave Detection by LIGO | 24 |
| 3 | Neural Networks and the Input-Output Problem | 27 |
| 3.1 | Neural Networks | 27 |
| 3.2 | Neural Networks and Multi-layer Perception | 28 |
| 3.2.1 | Activation Function | 29 |
| 3.2.2 | Background Propagation | 31 |
| 3.2.3 | Network Architecture | 31 |
| 3.2.4 | Data Normalization | 32 |
| 4 | First Round of Experiments | 34 |
| 4.1 | Methodology | 34 |
| 4.1.1 | Building the database | 34 |
| 4.1.1.1 | The problem with the wave points | 35 |
| 4.1.2 | Search for the best Neural Network | 36 |
| 4.2 | Results | 39 |
| 4.2.1 | Test with LIGO Data | 42 |
| 5 | Second Round of Experiments | 44 |
| 5.1 | Methodology | 44 |
| 5.2 | Results | 46 |
| 5.2.1 | Test with LIGO Data | 47 |
| 6 | Third Round of Experiments | 49 |
| 6.1 | Methodology | 49 |

| | | |
|----------|--|-----------|
| 6.2 | Results | 50 |
| 6.2.1 | Test with LIGO Data | 53 |
| 7 | Result and Discussion | 54 |
| 7.1 | Future work | 55 |
| 7.2 | Published work | 55 |
| | Bibliography | 56 |
| | APPENDIX | 59 |
| | APPENDIX A Code | 60 |
| A.1 | Wave generator | 60 |
| A.2 | Generate <i>chirp mass</i> | 61 |
| A.3 | PyCBC wave generator | 62 |

1 Introduction

For the first time, in the year 2015, gravitational waves were detected directly (ABBOTT; AL., 2016). This event occurred almost 100 years after Einstein's publication of the Theory of General Relativity, where gravitational waves are described for the first time (EINSTEIN, 1916). A century ago there was no technology advanced enough to evidence Einstein's theory, but in 2015, after 30 years of being built the Laser Interferometer Gravitational-Wave Observatory (LIGO) detected gravitational waves for the first time directly.

The emphasis on the word *directly* occurs because gravitational waves had already been observed indirectly, by the 1993 Nobel Prize winners in Physics, Russell A. Hulse and Joseph H. Taylor, Jr (Nobel Prize in Physics, 2019). Physicists found a binary pulsar in 1974, and through observations, it was calculated that the binary pulsar would be releasing gravitational waves (Hulse; Taylor, 1975). Over the years, this pulsar matched the predicted behavior. Thus, through this observation, the existence of gravitational waves was confirmed. The entire observation was made through telescopes, so the source rescued was electromagnetic waves, what we can call the indirect observation of gravitational waves (LIGO Laboratory, 2015).

The gravitational waves released by the Hulse and Taylor binary pulsar or detected by LIGO are space-time vibrations, the gravity emitted by these waves is capable of deforming space-time and propagating through the Universe. With this, it can be said that gravitational waves are generated by every accelerated non-spherical object, so as we speak, we scratch a paper or when a rocket is launched these waves are created. Although gravitational waves are generated all the time, knowing that gravity is the weakest known force, these waves have a very small magnitude (WHEELER., 2008).

Thus, we still do not have enough technology to simulate gravitational waves physically on Earth, so gravitational wave detection occurs from astrophysical phenomena, with a large amount of energy, such as the merger of two black holes, the merger of neutron stars, or waves coming from the Big Bang itself (THORNE, 2014). Taking as an example the first observation made by LIGO, which was of a merger of two black holes (ABBOTT; AL., 2016), this phenomenon detected waves of magnitude 10^{-21} meters (SCHUTZ, 2009; HOBSON et al., 2006). Which is a million times smaller than the size of a proton, that

has a magnitude of 10^{-15} m (BEZGINOV et al., 2019).

The detection of these waves brought the possibility of more comprehensive astrophysical observations, where until that point only electromagnetic waves were the source of observation used by us. Unlike electromagnetic waves, which are vibrations of electromagnetic fields, as already said, gravitational waves are the propagation of the vibration of space-time itself. As these waves have less external interference than electromagnetic waves, they enable greater accuracy of your signal and greater integrity of information about your source (LIGO Laboratory, 2015).

Knowing it, this work aims to extract information about the bodies capable of generating gravitational waves. We know that to generate these waves we need very large objects, at a solar mass scale, so we try to infer the mass of these gravitational wave-generating objects. We use neural network algorithms, which is a universal function approximator (HAYKIN, 1999), to extract the mass information. Although gravitational waves have been detected and their signal is known, we have few samples of this phenomenon, and for good training of a neural network, it is necessary a significant sample of patterns. To overcome this lack of data, we used data that simulate the shape of the gravitational wave as training networks.

This project was developed in three different experiments, in which continuous improvement of the inference was sought throughout the experiments. The first experiment yielded an article published in CBIC XVI, the article published only analyzed simulated data, in this text, all experiments rely on experimentation with data from LIGO observations. The second and third experiments have the result of this thesis.

1.1 Justification

In the past, only electromagnetic waves were sources of information about the cosmos, today we can also count on gravitational waves. These waves, as already mentioned, carry with them more precision about the data from their source than electromagnetic waves, since they have less interaction with the medium. That way gravitational waves bring the possibility of understanding open questions about the universe (LIGO Laboratory, 2015). Most of the category observed from LIGO is about the merger of two large objects, it is a type of phenomenon that occurs in fractions of a second and is capable of causing

waves that travel thousands of light years. In this category, two objects merge into one. Related objects can be neutron stars and black holes. Singular objects may also be able to generate gravitational waves, such as pulsars. Also, there is the possibility of capturing the wave signal that originated the universe in the Big-Bang (WHEELER., 2008).

With that, we need to learn to process and extract information from these gravitational wave signals in an increasingly efficient way and with the wide collaboration of the scientific community. Considering the resource limitations, we tried in this work to present a study using only a personal computer to process gravitational wave detection data.

We will study here the merger events of two large astrophysical objects, a binary system. Within these events we will only analyze the spiral phase of this wave, better described in Chapter 2.

1.2 Related Works

HAEGEL; HUSA, in 2020 uses deep neural networks to infer the remnants mass and spin magnitude of the merge of two black holes. It is capable of producing an efficient method to predict the remnant black hole in a binary black hole merger. The performance of its deep neural network is compared with the accuracy of the LIGO algorithm. Although both experiments work with masses, the cited work considers gravitational waves given from a relativistic basis, while our data only considers the numerical and post-Newtonian parts. We also differ in the desired output, we want to understand the mass value of both black holes before the merge occurs, in the cited work, the interest is in understanding the characteristics of the black hole produced through the merge.

GEORGE, in 2017 presents the use of Deep Learning for signal analysis, indicating whether those waves are gravitational wave detections, and also makes an estimation of the wave parameters. The article uses LIGO data and shows that Convolutional Neural Networks can be used for both wave detection and parameter estimation. They also knowledge that the techniques we have developed can be useful in many other domains of science and technology. Again we have the use of the neural network to analyze gravitational waves signals, although they focus mainly on having a real time signal detection, differentiating on our experiments that have the detection in hand and later

process the signal to characterize the detection.

SANTOS et al., in 2020, show a new form of calculating the ring-down time for the detection of the gravitational waves, again with LIGO data. It also presents the use of neural networks and a small computational capacity, calling the approach simple. The results show that the methodology proposed was able to define the ring-down time for all the binary black hole events, including the events where LIGO cannot calculate its ring-down time. Yet we have an experiment that is able to characterize gravitational wave events using their signals, but still a different outcome than ours.

All the works presented here use artificial neural networks and gravitational wave signals to archive their results. However, although working with the same database, LIGO detection database, and using similar techniques, these researches are different in their objective. Thus, they can be considered complementary between them.

1.3 Objective

The general and specific objectives that will guide the conduct of this exploration will be presented below. The general idea defines the purpose of the study and the specifics characterize the stages or phases of the design.

1.3.1 General Objective

Although LIGO has made its data available and even a library in Python, making a characterization of these waves through their signal is still an arduous task that requires a lot of computational power. Therefore, several academic works try to invest in the use of machine learning, whether to detect if the signal is a gravitational wave or to characterize a wave signal.

We propose with this thesis to extract physical characterization of the objects capable of generating gravitational wave signals with significant amplitude using a personal computer. Showing that it is possible to generate simulations of gravitational waves and process these simulations to extract information from them using Machine Learning (ML), more specifically Artificial Neural Networks (ANN). Here, we are just looking at the generation of gravitational waves coming from an astronomical binary system and

determined to extract information from the spiral phase of the wave on the value of the masses of the two waveform objects before they merge into one.

1.3.2 Specific Objective

To achieve the general objective, the following specific objectives were defined:

- Study of computational physical modeling of gravitational waves using a personal computer;
- Computational simulation of the spiral phase of astrophysical binary systems;
- Artificial neural network training with the simulated data of these waves;
- Use the neural network model algorithm to infer the masses of the objects that characterized gravitational wave simulations;
- Use the modeled neural network algorithm to infer masses of the objects that generated gravitational wave signals captured by LIGO data.

1.4 Document Structure

In the next chapter, we will delve deeper into the issues of gravitational waves, emphasizing the description of waves generated from the merger of binary objects. In the Chapter 3, we will present a brief introduction to neural networks, considering only the networks used in this text. Soon after, in Chapter 4, 5 and 6, we will explain the stages of experimentation and the methodologies used, where the chronological order of the works developed here are presented, with their respective approaches to the pursuit of the most accurate astronomical bodies' mass information inference. Finally, we will present the conclusions generated from our experiments.

2 Gravitational Waves

Space, as we cognize, is seen as a three dimension entity, but after Einstein's theory of general relativity, time was grouped with it, forming what is called space-time a continuous of four dimensions. In that theory the physical behavior of gravity is explained, although Newton's gravity laws could still be applied, the instant an instantaneous force that comes from bodies is now described as space-time bending. A body with a significant density could cause the bend of the Universe (Figure 1). Objects such as black holes, planets, and other with significant density are able to perform it on a considerable scale (SCHUTZ, 2009).

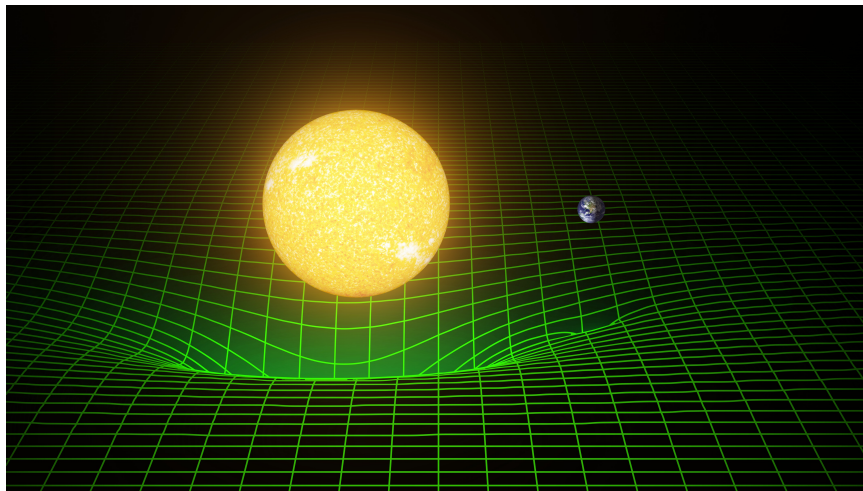


Figure 1 – The bend that the Sun causes in space-time and the smaller bend caused by the Earth. Credit: (T. Pyle/Caltech/MIT/LIGO Lab, 2016)

Still, in the theory of general relativity, gravitational waves are called the vibration of the space-time. As said before, gravitational waves are produced by every accelerated and not spherical object. To visualize it we can imagine a rock being thrown on a lake, the rock falls, and the water starts to propagate waves from the same spot where the rock landed. That is how we can visually imagine gravitational waves, however, if in that event the force transmitted from the rock propagates on the water, the gravitational waves case waves reverberate space-time itself (WHEELER., 2008).

Historically we have tremendous discoveries about the cosmos only by analyzing electromagnetic waves, and still much more to be found. On the other hand, gravitational waves are unrelated to electromagnetic waves, this creates an opportunity for an entirely

new domain of observation. When we compare those two waves, electromagnetic and gravitational, notwithstanding both travel at the same speed, the second is extremely weak, on a scale of 10^{-21} meter (SCHUTZ, 2009; HOBSON et al., 2006). Also, electromagnetic waves have high interactivity with the cosmos, they can be refracted, absorbed, bent, or reflected, contrary to it gravitational waves interact very weakly with the matter, bringing us a sign with a piece of clear information about its source.

As cited, we still do not have enough resources to reproduce gravitational waves as physical experiments, making us look at the sky, searching for astronomic events capable of producing them. Often those events are a collision of black holes, collision of neutron stars, supernova explosion, and gravitational waves from the Big Bang (SCHUTZ, 2009). Knowing these waves can come from this type of phenomenon, we are able to understand those phenomena better and hear what electromagnetic waves could not tell us until now (LIGO Laboratory, 2015).

Our work treats specifically gravitational waves at the spiral phase, they come from binary systems, where two objects spin around the same bodycenter¹. We can see in Figure 2 an illustration of what happens with space-time when a spiral wave is in action. As will be clear in the next section, the bodies of that system (Figure 2) are a tremendous amount of speed, and they also have masses equivalent to multiple solar masses. Since we seek to understand spiral gravitational waves we will present only the calculus of the waves of this sort of wave.

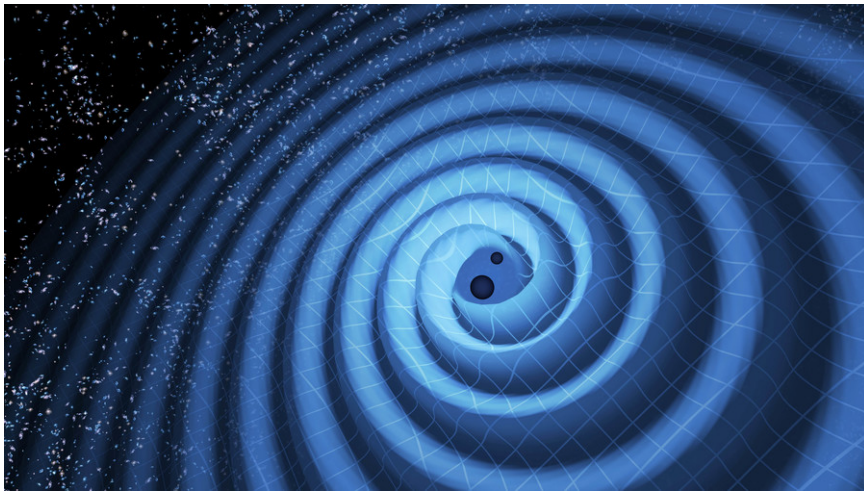


Figure 2 – The spiral gravitational waves are formed by two black holes. Credit: (LIGO/T. Pyle, 2016)

¹ Center of mass.

2.1 Gravitational Waves of a Binary System

As said before, phenomena like neutron stars and black holes collision are astrophysical events able to produce gravitational waves within a significant magnitude (amplitude bigger than 10^{-21} meters). Such events were the only source that the LIGO was able to detect (ABBOTT; AL., 2016).

A binary system is composed of two objects. They can be a combination of black holes and neutron stars. Those objects meet in the cosmos and stay orbiting a common barycenter, over time they become closer and closer until they collapse together and become only one object (RUBBO et al., 2007). It is important to say that these two objects orbit around each other for an undetermined amount of time. We only have data to evidence when they are close enough and on a very high velocity², to the point of producing gravitational waves with a significant scale.

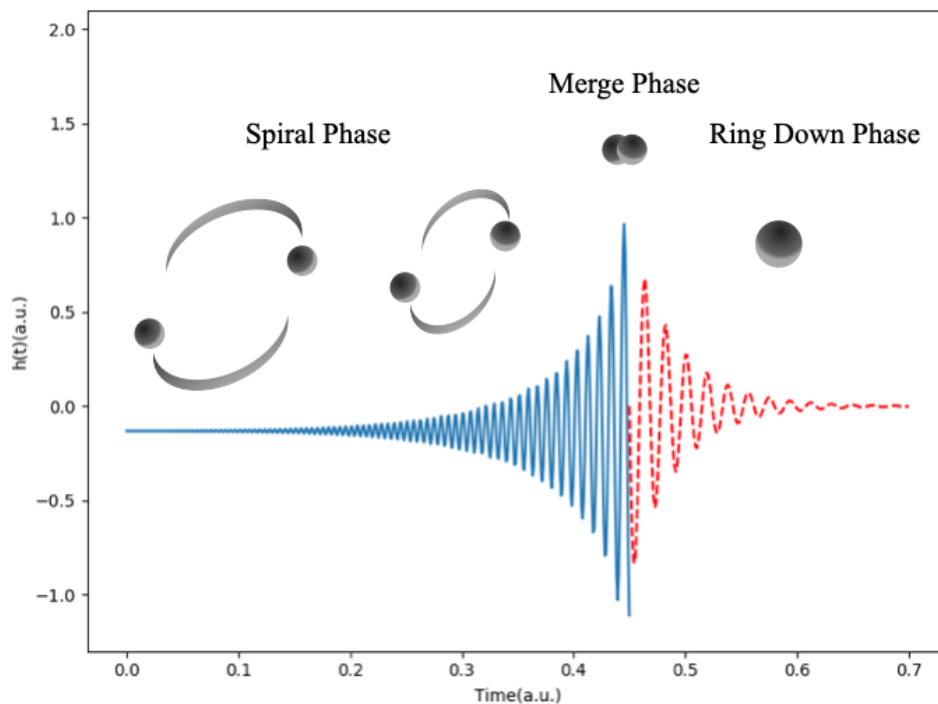


Figure 3 – Phases of a binary system. Blue continuous line - gravitational wave signal from the spiral phase Equation 2.1. Red dashed line - ring down stage Equation 2.8

This phenomenon can be described in three phases (Figure 3). The first one is the spiral phase, when both objects are orbiting each other at a very high speed, with fractions

² The GW150914 detection achieved a velocity higher than 0.3 speed of the light (ABBOTT; AL., 2016)

of the light speed (ABBOTT; AL., 2016). This phase can be described phenomenologically by Equation 2.1. Where it seeks to define the gravitational waveform of the simple and circular binary system (RUBBO et al., 2007).

$$h(t) = \mathcal{A}(t)\cos\Phi(t) \quad (2.1)$$

Where $h(t)$ indicates the waveform, $\mathcal{A}(t)$ is the amplitude and $\Phi(t)$ is the wave phase. $\mathcal{A}(t)$ and $\Phi(t)$ can be described as:

$$\mathcal{A}(t) = \frac{2(G\mathcal{M})^{5/3}}{c^4 r} \left(\frac{\pi}{P_{gw}(t)} \right)^{2/3} \quad (2.2)$$

G is Newton's gravitational constant; c is the speed of light; r is the initial observation distance, in light-year; Φ_0 is the initial phase and P_{gw} is the period in the instant t . The value of \mathcal{M} (Equation 2.3) is called "*mass chirp*", given this name, because it can be determined from the evolution of the signal received at the time *chirp* of the system's evolution ³

$$\mathcal{M} \equiv (M_1 M_2)^{3/5} / (M_1 + M_2)^{1/5} \quad (2.3)$$

Considering the Equation 2.1, it is possible to see that the fluctuation of the space-time metric depends only on time. Therefore $\Phi(t)$ can be written as the Equation 2.4.

$$\Phi(t) = \Phi_0 + 2\pi \int_0^t \frac{dt'}{P_{gw}(t')} \quad (2.4)$$

The period P_{gw} is easily related to the orbital period P_{orb} :

$$P_{orb}(t) = 2P_{gw}(t) \quad (2.5)$$

$$P_{orb}(t) = \left(P_0^{\frac{8}{3}} - \frac{8}{3}kt \right)^{\frac{3}{8}} \quad (2.6)$$

³ Exact moment of approximation between the two objects and gradual increase of the frequency of the received signal.

$$k \equiv \frac{96}{5} (2\pi)^{\frac{8}{3}} \left(\frac{GM}{c^3} \right)^{\frac{5}{3}} \quad (2.7)$$

Observing the equations above, it is noticed that some input variables are determinants of the waveform. These are the initial orbital period (P_0), mass *chirp* (\mathcal{M}), the observation distance (r) and the initial phase of the wave (Φ_0).

The second phase is the moment of collapse when objects unite and become one. The last one is the ring-down phase, gravitational waves begin to fatigue until they return to low amplitude. At this part the waves can be represented phenomenologically as a damped harmonic oscillator (MARION; THORNTON, 2003), see below:

$$h(t) = B e^{-\frac{t}{\tau}} \sin(\omega t + \delta_0) + D \quad (2.8)$$

where B and D are constants, δ_0 is an initial phase, and τ is the damping constant. It can be seen as the red dashed line in Figure 3.

2.2 Gravitational Wave Detection by LIGO

It took 25 years and 2 detectors for the first LIGO detection to occur. As cited before, LIGO on September 14 of 2015 the LIGO laboratory detected a gravitational wave produced by the merge of two black holes. The astrophysical objects had respectively the masses of $36_{-4}^{+5} M_{\odot}$ and $29_{-4}^{+4} M_{\odot}$ before the merge. Within the merge they radiate the energy of $3_{-0.5}^{+0.5} M_{\odot}$ in the gravitational waveform, ending with a new black hole of $62_{-4}^{+4} M_{\odot}$.

LIGO laboratories are twins, which means that there are two laboratories, one confirming the information of the other so that there is better reliability of the inferred information. As the name says, the detectors are interferometers, these interferometers have the precision capable of measuring 1/1000th part of the nucleus of an atom.

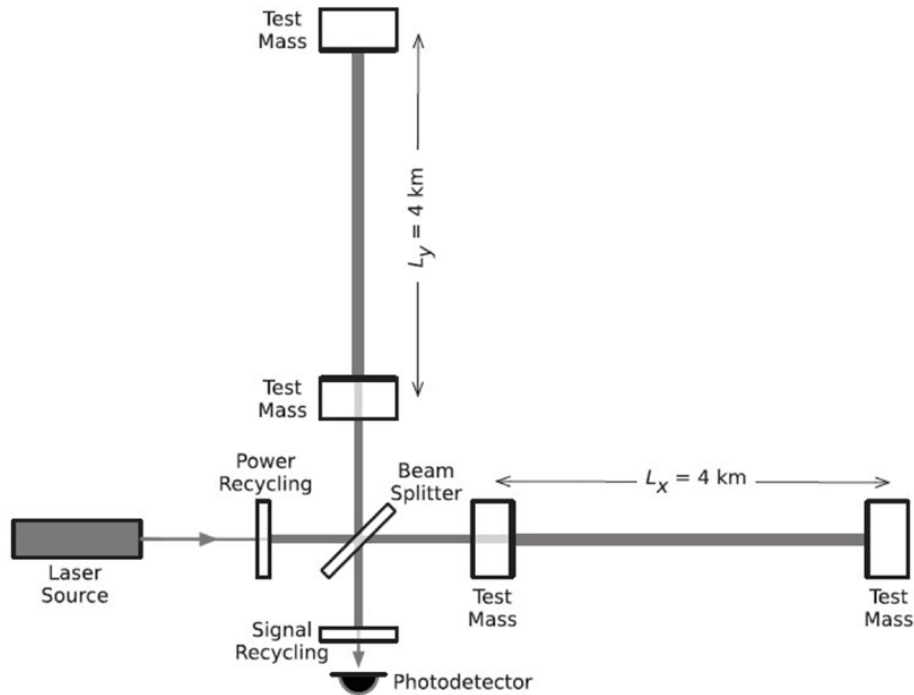


Figure 4 – Simplified diagram of an Advanced LIGO detector (not to scale). Credit: (ABBOTT; AL., 2016)

Both of the detectors are located in the United States of America, one is in Livingston, state of Louisiana, and the other in Hanford, Washington state. The gravitational waves are measured as the arms length happens to have the orthogonal difference, see Figure 5. Two arms each operate a modified Michelson interferometer, each arm is formed by two mirrors acting as test masses, separated by $L_x = L_y = L = 4km$. A passing gravitational wave effectively changes the lengths of the arms so that the measured difference is Equation 2.9, where δL_x and δL_y are respectively the difference the arms have in their length when a gravitational wave passes through it, and h is the deformation amplitude of the gravitational wave projected onto the detector.

$$\Delta L = \delta L_x - \delta L_y = h(t)L \quad (2.9)$$

This differential length variation changes the phase difference between the two light fields returning to the beam divisor, transmitting an optical signal proportional to the gravitational wave deformation to the output photodetector. In Figure 5 it is possible to see that on the image on top the waves cancel themselves signals, but on the image below, where the red arrow points, when a gravitational wave passes through it one of the arms

squeeze (δL_x) while the other stretch (δL_y), this movement causes the light beam that hits the detector to flicker ($h(t)L$).

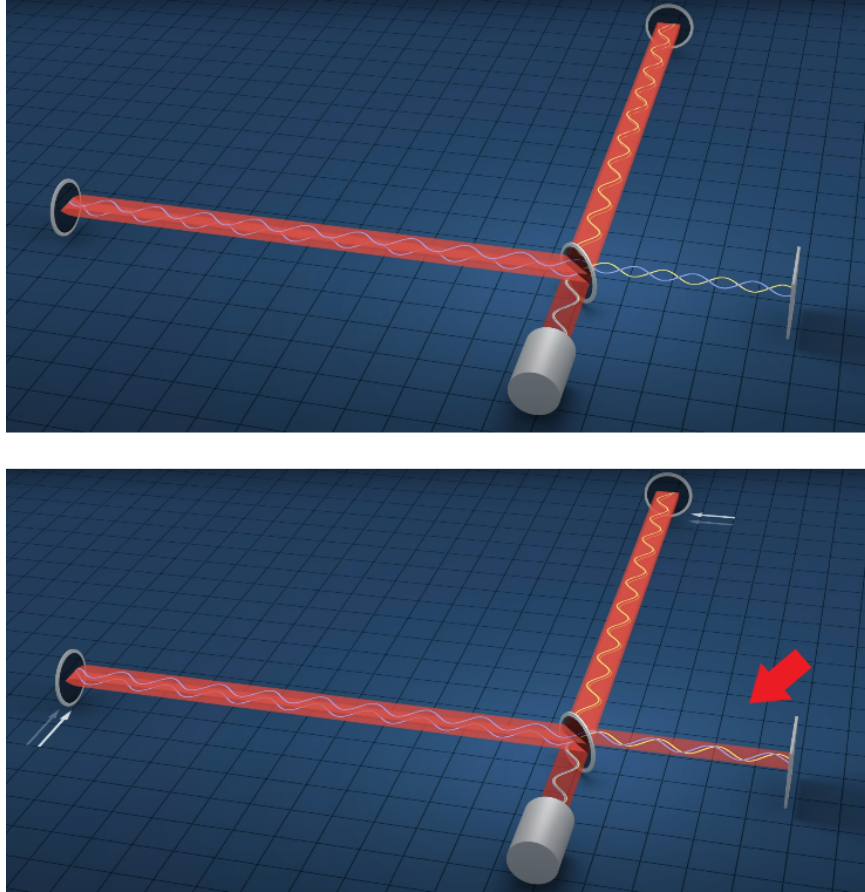


Figure 5 – Illustrating how the twin LIGO observatories work. Credit: (LIGO/T. Pyle, 2016)

Knowing how the detection of gravitational wave signals works, we used in this dissertation the data made available by LIGO from its research and detections. We also consider as parameters for approximation metrics of our algorithms the mass values of the gravitational wave generating objects reported by LIGO.

3 Neural Networks and the Input-Output Problem

To the better understating of machine learning to create our models, this Chapter contains a breath introduction to Artificial Neural Networks (ANNs), limited to the ones used in this dissertation.

3.1 Neural Networks

As said in (HORNIK et al., 1989), neural networks are universal function approximators, to ensure a satisfactory accuracy we would need the right training and number of nodes.

The core ground of ANNs are the biological neural networks, formulated in 1943 by McCulloch and Pitts. ANNs are distributed parallel systems composed of simple processing units, called neurons or nodes, which compute mathematical functions, normally non-linear (BISHOP, 1995).

Neurons or nodes are arranged in a spatial arrangement usually composed of one or more layers and interconnected by a large number of connections. In most models, these connections are associated with weights, which serve as that magnifier of the importance of the input signals of the neurons in the network. Each of the units of an ANN receives a signal. This signal, weighted by the respective unit's input connections, is treated by a mathematical function, called the activation function. Finally producing a new output signal that is propagated through the network (FERREIRA, 2006).

ANNs have the ability to learn through examples, and perform data interpolation. In the learning process of an ANN, the main task is to determine the strength of the connections between neurons, and the learning algorithms used to adapt these parameters. These algorithms have the purpose of performing the adjustment of the weights of the connections for the best possible generalization of the information contained in the exposed examples.

3.2 Neural Networks and Multi-layer Perception

When McCulloch and Pitts first created the concept of ANN, they created a mathematical model of a Perceptron, which attempts to copy the structure of a neuron (BISHOP, 1995). A perceptron can receive several *entries* (x_1, \dots, x_m) which the relevance of those entries are made by *weights* (w_1, \dots, w_m). In mathematical form, one can describe a neuron in a function of its index k (Equation 3.1).

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (3.1)$$

The Equation 3.2 includes *bias* (b) which has the purpose of increasing or decreasing the value of the download function. That is, the value of the function is negative or positive. The Equation 3.2 is a linear function, in machine learning, it is called linear regression when our model responds to a linear function. It can also be written as a one layer neural network.

$$h = u + b = w_mx_m + b \quad (3.2)$$

$$y_k = \varphi(h_k) = \varphi(u_k + b_k) \quad (3.3)$$

On Equation 3.3 u_k is the *linear combination* of the input values (x). Already φ is an activation function and y_k is output from the function.

For a better understating of layers, let us observe the Image 6. The input is on the very left, represented as a closed black circle, the lines connecting the input layer with forwarding layers have the representation of weights, and each line has a different weight for each connection between the input layer and the node ahead. The nodes are the bigger circle on white. Each node has a Equation 3.3 on it making a calculus as passing the result forward through another liner connector that another node receiving it. Until it arrives at the output layer.

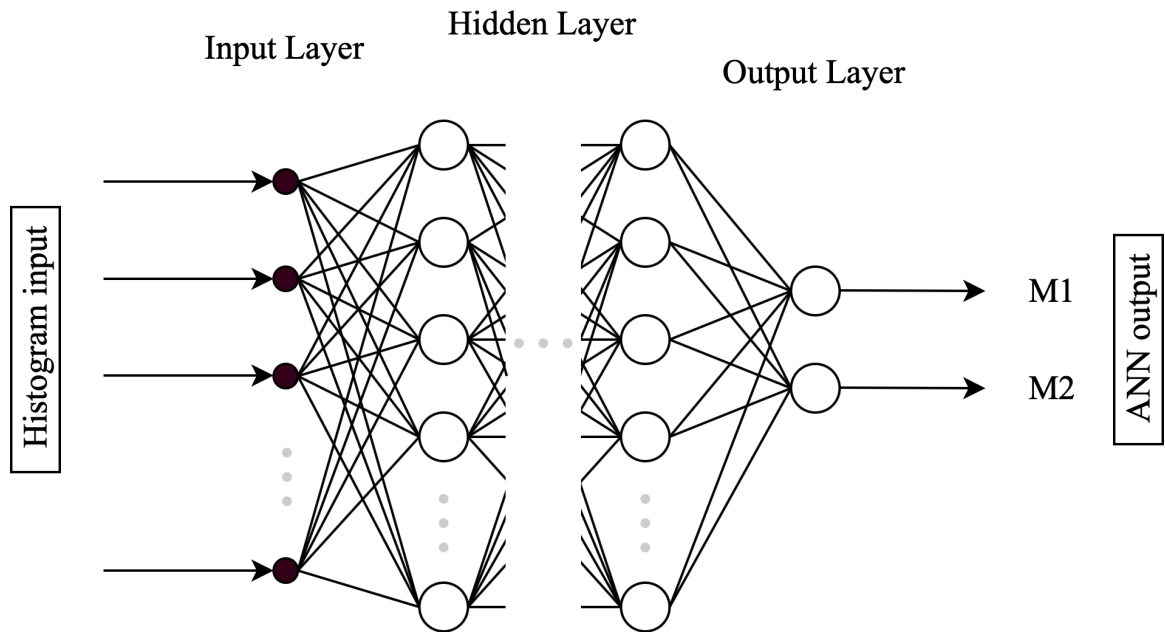


Figure 6 – The Architecture of a Multilayer Perceptron Artificial Neural Network.

The type of neural network we represent in Figure 6 is called a Multilayer Perceptron (MLP). Those networks are fully connected and are directed feed-forward. This means that all the nodes on the previous layer are connected to the layer forward, which can be notable in the figure above. And direct feed-forward says that the input layer values are projected only in their direct sense to the layer of exit, and never the other way around (HAYKIN, 1999).

The input layers receive the data for the training. The intermediate or hidden layers catch the entrance value passing from the connections and exit it to the neurons of the following layer. Finally, the output layer delivers the result of the network (BISHOP, 1995).

3.2.1 Activation Function

The activation function (φ on Equation 3.3) determines the relationship between the inputs and outputs of a neuron and, in general, introduces a degree of non-linearity in most networks. Any second-order differentiable function (Class C^2) can be an activation function, however in practice, only a small group of activation functions (FERREIRA, 2006).

In this research we used the Rectified Linear Unit function (ReLU), it is a Class

C^0 function, because its first deviation is not continuous. This function is used mostly because of its speed up and simplicity of gradient computing (NAIR; HINTON, 2010).

$$f(x) = \max(0, x) \quad (3.4)$$

Other functions were also used as Hyperbolic Tangent (Equation 3.6), Sine (Equations 3.5), and the Sigmoid (Equation 3.7) function (PEDREGOSA et al., 2011). See Figure 7 for the form of each of the functions used in this work.

$$f(x) = \sin(x) \quad (3.5)$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.6)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.7)$$

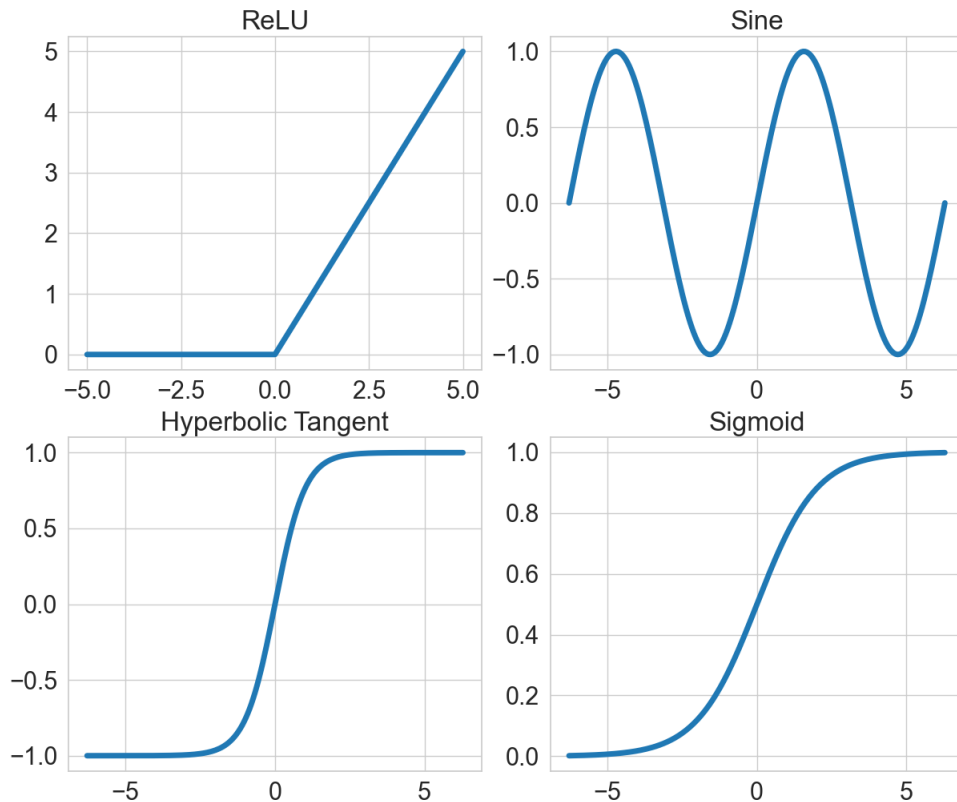


Figure 7 – Different activation functions used in the research.

An ANN can have different activation functions for neurons from the same layer or from different layers, however, the vast majority of ANNs have the same activation function for all neurons in the same layer.

As said before, with the right training and configuration neural networks can describe any function with decent accuracy (HORNIK et al., 1989). We then tried to persuade here the right training and configuration to find our correct algorithm.

3.2.2 Background Propagation

The training process of an ANN is a non-linear minimization problem in which the weights (or connections) of the network are interactively adjusted to provide a minimum error between the desired output and the output obtained by the network.

To adjust the weights of the network connections, we use the backpropagation technique (RUMELHART et al., 1986). This method is used as a parameter for the network prediction error propagating this error retroactively, so it receives this name (BISHOP, 1995). The function used to make this adjustment is called Optimizer. Among the existing ones, the Stochastic Gradient Descent (SGD) has evidenced itself as an efficient optimizer used in many machine learning problems. Furthermore, there's the Adam optimizer, which only requires first-order gradients and consumes less computing power (KINGMA; BA, 2017). In the article, we used the Adam optimizer.

3.2.3 Network Architecture

As mentioned earlier, Honik cites that neural networks need the right amount of nodes and training. With this, we can understand that the network architecture of an MLP needs definitions in its parameters, which are: input quantity, number of hidden layers, number of nodes in its hidden layers, and number of output nodes.

These parameters selection is a dependent part of the problem. There are different approaches to find the right parameters for the ANN, such as pruning algorithm, polynomial time algorithm, and others. However, in general, these methods are quite complex and difficult to implement, in addition to the fact that none of these can guarantee the optimal solution to the problem. For this reason, in most cases, the network architecture is defined

based on preliminary experiments.

The technique applied to choose the ANN parameters was the *Grid Search*. This technique consists of receiving a finite number of possible combinations of parameters and executing all combinations, the combination of parameters that obtains the best result is chosen (PEDREGOSA et al., 2011).

For the most successful combination of parameters to be pointed, a result estimator, and a loss function. This estimator is a metric that indicates how well the model performs. Our loss function is an adaptation of the Mean Squared Error (MSE) (Equation 3.8) as a loss function. Instead of using only the MSE for both of the masses, we used the MSE for both plus MSE for $M1$ and MSE for $M2$ (Equation 4.1).

$$MSE(y, \tilde{y}) = \frac{1}{m} \sum_{i=1}^m (y_i - \tilde{y}_i)^2 \quad (3.8)$$

$$loss = mse(y, \tilde{y}) + mse(y_{M1}, \tilde{y}_{M1}) + mse(y_{M2}, \tilde{y}_{M2}) \quad (3.9)$$

Where in both equations m is the number of network outputs, y is the expected result and \tilde{y} is the forecast value. Again $M1$ is the first mass and $M2$ is the second mass of the objects that generated the gravitational waves.

3.2.4 Data Normalization

Activation functions that are nonlinear have a different domain. As an example we have the Sigmoid and Hyperbolic Tangent functions, their interval go between $[0, 1]$ and $[-1, 1]$ respectively.

Therefore, all data reported to the network must be normalized over the domain of the activation function used. The data normalization process is performed before training, in pre-processing. The normalization technique used in this research was the Range of Linear Transformation, also called Min-max Scaling:

$$X_n = \frac{X_0 - X_{min}}{X_{max} - X_{min}} \quad (3.10)$$

where X_n is the normalized value, X_0 is the original value, X_{min} is the minimum value, and X_{max} is the maximum value.

The input and output data normalization are independent. Therefore, we can use normalization only in data entry, as was done in the experiments presented here.

4 First Round of Experiments

4.1 Methodology

For the development of this phase of the research, we used a MacBook Pro, with a Mac OS Catalina operating system, 500Gb SSD, and RAM 16Gb. We used Python 3.6 and the PyTorch library for the ML experiments (PASZKE et al., 2019).

4.1.1 Building the database

In order to infer what the masses of the objects generating the observed waves would be, we used simulated data to have a significant database for training the network, as mentioned in the introduction. Thus, the text described in 2.1 was the inspiration for building the code in Python that would generate the wave simulation. To correctly write the equation (Code A.1), we use the trapezoid rule (PRESS et al., 2007), where t (time) is the limit. The value chosen for dt (time interval) is 0.001, the value that decreases the global error value of the integral, increasing the wave precision degree, however, the wave generation is within feasible time. Thus, while the value of t does not reach the value of $(P_0^{\frac{8}{3}} - \frac{8}{3}k)^{\frac{3}{8}}$ the wave was being generated.

For the simulation, considering our objective, we only vary the value of the solar masses of the objects that would be the generators of the waves, as a consequence only the chirp mass. Therefore we left all other variables fixed, they would be: distance to the event is kpc , which was 410,000 (equal to the luminous distance of the first LIGO observation GW150914), also the initial phase was 0.1 and the initial period was 0 (ABBOTT et al., 2019; ABBOTT et al., 2021).

The chosen range value for the masses (M1 and M2) was between 20 and 40 solar masses. These values are justifiable considering the LIGO observation data so far, this mass range is the majority among the observations, see Figure 8, the red square outlier the masses of the black holes defections. It is important to point out that the range of masses limits our algorithm, so if we are observing a gravitational wave with a range that exceeds what we are training, we will not be able to masterfully infer the data. What is further in this work will limit the analysis of data collected by LIGO, so not all observed

by LIGO will be able to pass through the inference of our algorithm.

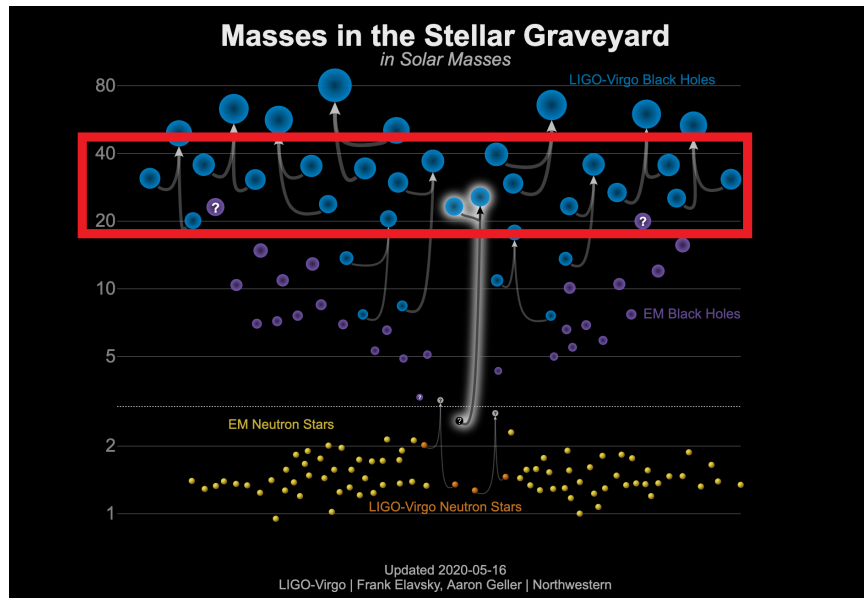


Figure 8 – LIGO and Virgo have observed the merger of several compact object binaries. The black holes represent a new population with masses that are larger than what had been seen previously with X-ray studies alone. This plot shows the masses of the binary components before the merger, as well as the mass of the merger remnant. Credit: (LIGO-Virgo/Northwestern Univ./Frank Elavsky, 2020)

As said before, we just vary the chirp mass at the time of wave generation. Code 1 describes how we calculate the chirp mass for all waves, looking at line 6 we check to see if there is no duplicate chirp mass. The reason for this is that different mass combinations can lead to the same chirp mass, and in this way we ensure that there are no two identical waves with different values of $M1$ and $M2$, which would confuse the network. Code A.2 also shows how the chirp mass data was generated, ensuring that the mass selection was a match, where the order doesn't matter, so if we have $M1 = 24$ and $M2 = 35$, we don't need to generate another chirp mass with $M1 = 35$ and $M2 = 24$, as the order does not matter.

4.1.1.1 The problem with the wave points

After all, waves have been generated, it was possible to see that the greater the mass was more points the wave had (Figure 9). Since the time step is fixed, the wave lasts longer when the mass has increased, which made the waves have more amplitude points.

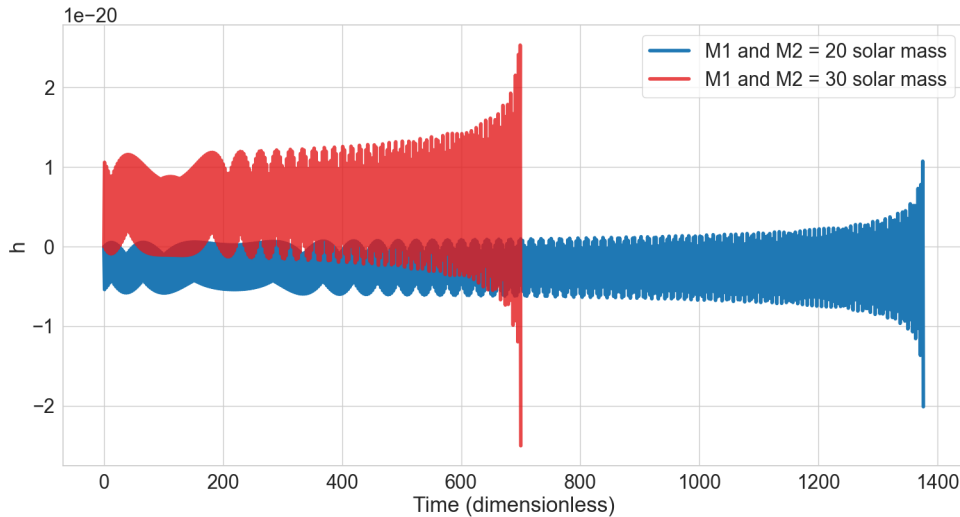


Figure 9 – The disparity of points between two wave signals.

To manage this situation, the wave's histograms (Figure 10) were used as the network input. That way, all waves are maintained with an equivalent characterization, and the network can be trained because all the input data have the same size of 300 points.

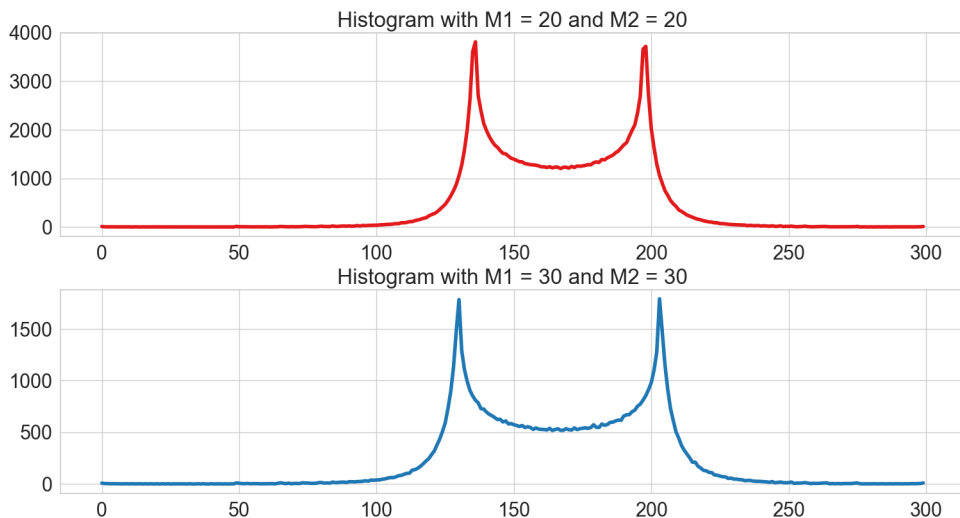


Figure 10 – Histogram of the waves with a different mass chirp. Now with the same quantitative points.

4.1.2 Search for the best Neural Network

To analyze the generated data we use Neural Networks, described in Chapter 3, first we tried to find the best combination of the network architecture (disposed in Table 1) and its learning rate, where we used: $1e10^{-1}$, $1e10^{-2}$ and $1e10^{-3}$. We applied the Grid Search technique, where a set of hyperparameters is disposed and combined, then we

try every possible combination to find the one that leads to the best performance of the algorithm (FEURER; HUTTER, 2019). All the networks used have 300 nodes as input layer and 2 nodes as output layer to bring us the approximated mass of M_1 and M_2 .

Table 1 – Neural Networks Architecture Configuration

| | |
|---|---------------------|
| Number of hidden layer | 2, 3, 5, 10 |
| Number of node on the hidden layer | 50, 100, 150 |
| Activation function | ReLU, Sigmoid, Tahn |

To train the networks, we also applied stationary configurations for all the networks searched. We used the *Adam* optimizer with the default values provided by the PyTorch library (PASZKE et al., 2019). Our loss function is an adaptation of the mean squared error (MSE) as a loss function. Instead of using only the MSE for both of the masses, we used the MSE for both plus MSE for 1 and MSE for M2 (Equation 4.1).

$$loss = mse(Y, O) + mse(Y_{m1}, O_{m1}) + mse(Y_{m2}, O_{m2}) \quad (4.1)$$

Where Y is the label (masses) and O is the network output.

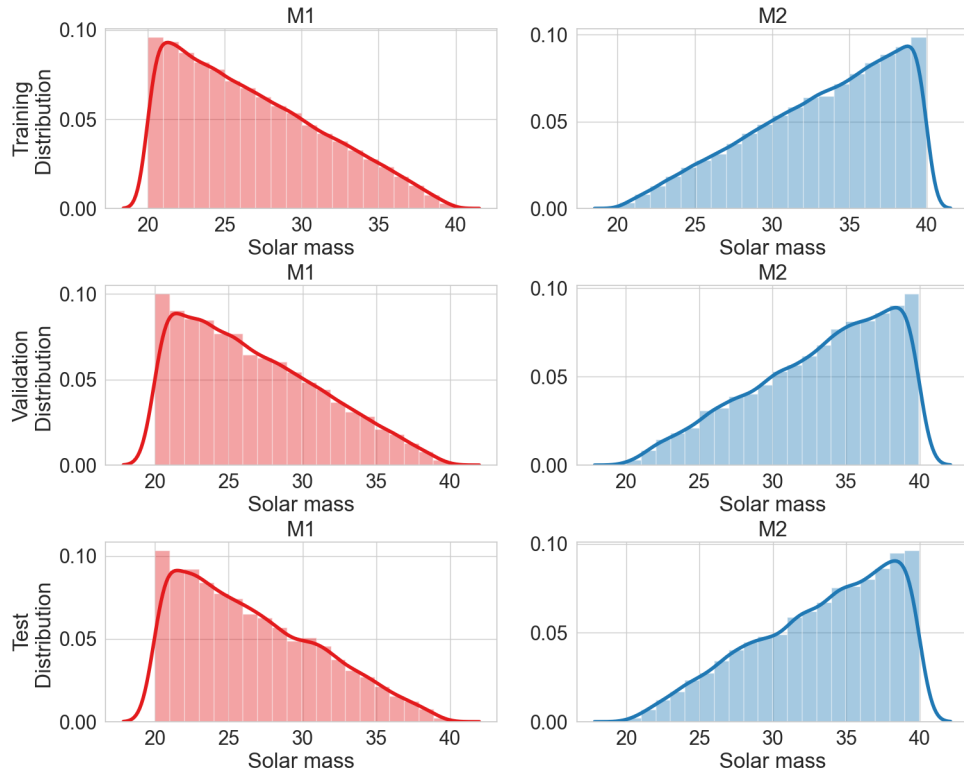


Figure 11 – Distribution of Training data set.

Proceeding with the experiment, we applied a fixed seed of value 2 to the database split. We separated the database into a training set with 70%, validation and test each one with 15% of the data. The balance of distribution between the data sets can be seen in (Figures 11). We first ran the Grid Search with 1,000 epochs for each combination and selected the best 10 networks according to the test set score (Table 2).

4.2 Results

Table 2 – The top 10 Neural Network Performance with 1000 epochs

| Hidden Layer | Hidden Layer Units | Activation Function | Learning Rate | MSE Test Score | σ^2 of Difference |
|--------------|--------------------|---------------------|---------------|----------------|--------------------------|
| 3 | 150 | ReLU | $1e10^{-2}$ | 4.437 | 4.438 |
| 2 | 150 | ReLU | $1e10^{-2}$ | 4.438 | 4.438 |
| 3 | 100 | ReLU | $1e10^{-2}$ | 4.438 | 4.438 |
| 2 | 100 | ReLU | $1e10^{-2}$ | 4.441 | 4.441 |
| 5 | 100 | ReLU | $1e10^{-2}$ | 4.443 | 4.443 |
| 2 | 50 | ReLU | $1e10^{-2}$ | 4.443 | 4.443 |
| 3 | 50 | ReLU | $1e10^{-2}$ | 4.443 | 4.448 |
| 5 | 100 | ReLU | $1e10^{-3}$ | 4.449 | 4.449 |
| 2 | 50 | ReLU | $1e10^{-3}$ | 5.458 | 5.458 |
| 3 | 150 | ReLU | $1e10^{-3}$ | 5.515 | 5.515 |

After this phase of the experiments, we could notice that some loss graphs were still decaying (Figure 12 and 13). We stored the networks with their exact states and weight values. We proceed to more 4000 epochs only applied on the top 10 networks. They were having a total of 5000 epochs of training for each network.

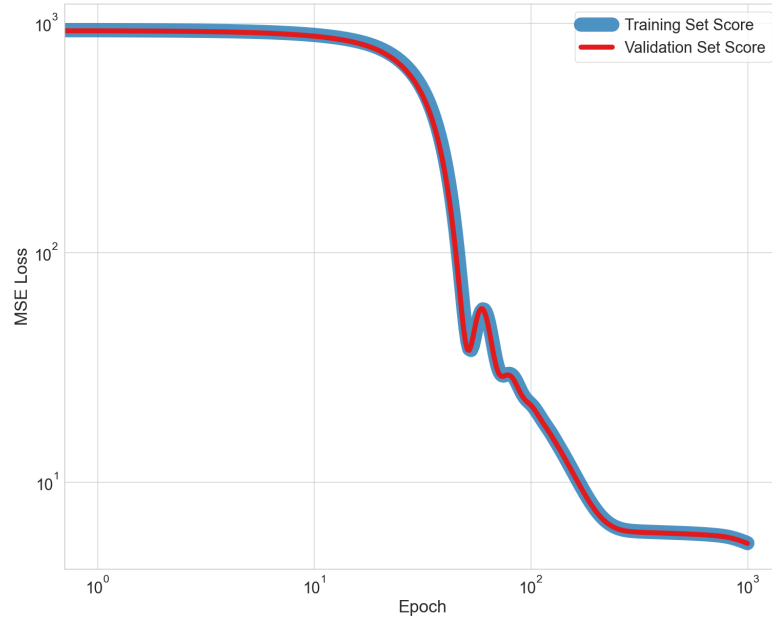


Figure 12 – Loss Graph of the 9st neural network on the ranking, in a logarithmic view. The network has 2 hidden layers with 50 nodes each, using ReLU as the activation function and $1e10^{-3}$ as the learning rate. On blue is the Training score, and on red is the validation score, showing the balance between both data and continuous algorithm generalization during the training process.

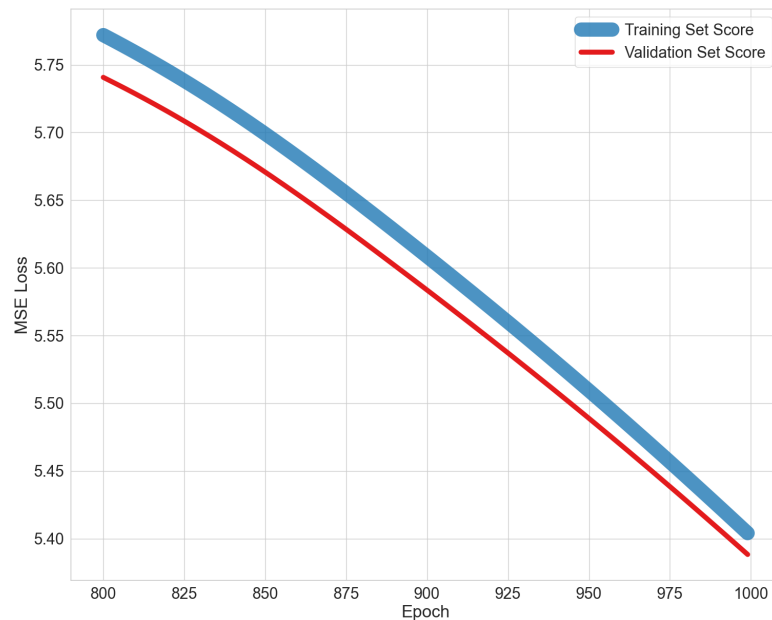


Figure 13 – Zoom on the Loss Graph of the 9st neural network on the ranking. The last 200 points.

With the new experiment, we also considered the training score to classify the best performance, which made the table change (Table 3).

Table 3 – All 8 Neural Network Performance with 5000 epochs

| Hidden Layer | Hidden Layer Units | Learning Rate | MSE Test Score |
|--------------|--------------------|---------------|----------------|
| 3 | 100 | $1e10^{-2}$ | 4.439 |
| 5 | 100 | $1e10^{-2}$ | 4.441 |
| 5 | 150 | $1e10^{-3}$ | 4.443 |
| 2 | 50 | $1e10^{-3}$ | 4.446 |
| 5 | 100 | $1e10^{-3}$ | 4.455 |
| 3 | 50 | $1e10^{-2}$ | 4.456 |
| 2 | 50 | $1e10^{-2}$ | 4.457 |
| 2 | 150 | $1e10^{-2}$ | 4.463 |
| 3 | 150 | $1e10^{-2}$ | 4.465 |
| 2 | 100 | $1e10^{-2}$ | 4.499 |

We also noticed that all the scores are under 5 now. The distribution of the real and predicted values can be seen in Figure 14, where one can visualize a flatness of the masses values through the prediction process. Also notable is that our mean floats around zero. Therefore, we had extra work to build this Table 3, the networks that were on top of their performance did not have an improvement.

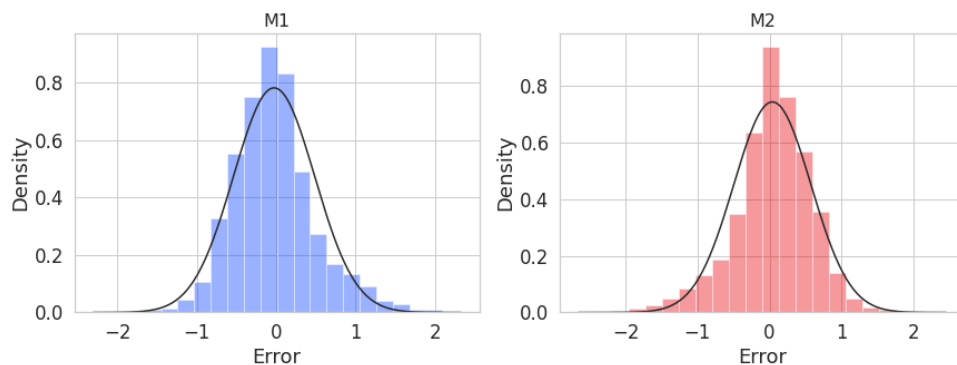


Figure 14 – Histogram of the difference between target and prediction using the best found neural network. On the left (blue color) we have the first mass (M1) difference and on the right (red color) we have the second mass (M2)

The best Neural Network we found has 3 hidden layer with 100 units on each layer, uses a learning rate of $10e^{-2}$ and got a Mean Squared Error on its test set of 4.439, which

leads to no improvement over the MSE metric score.

4.2.1 Test with LIGO Data

The final goal of this part of the experiment was to test the best neural network on LIGO's data. For that, we used the help of the *PyCBC* library (Biwer et al., 2019), where we could get easy access to LIGO data and make some transformations over it.

We used 6 different waves and the range was inside our range from $20M_{\odot}$ to $40M_{\odot}$, their plot can be seen in Figure 15. On those waves, we applied a low pass and high pass filter, with the value of $35 - 350Hz$ bandpass filter, for each of the waves. This filter is justified by the (ABBOTT; AL., 2016) where the same filter is applied to the data.

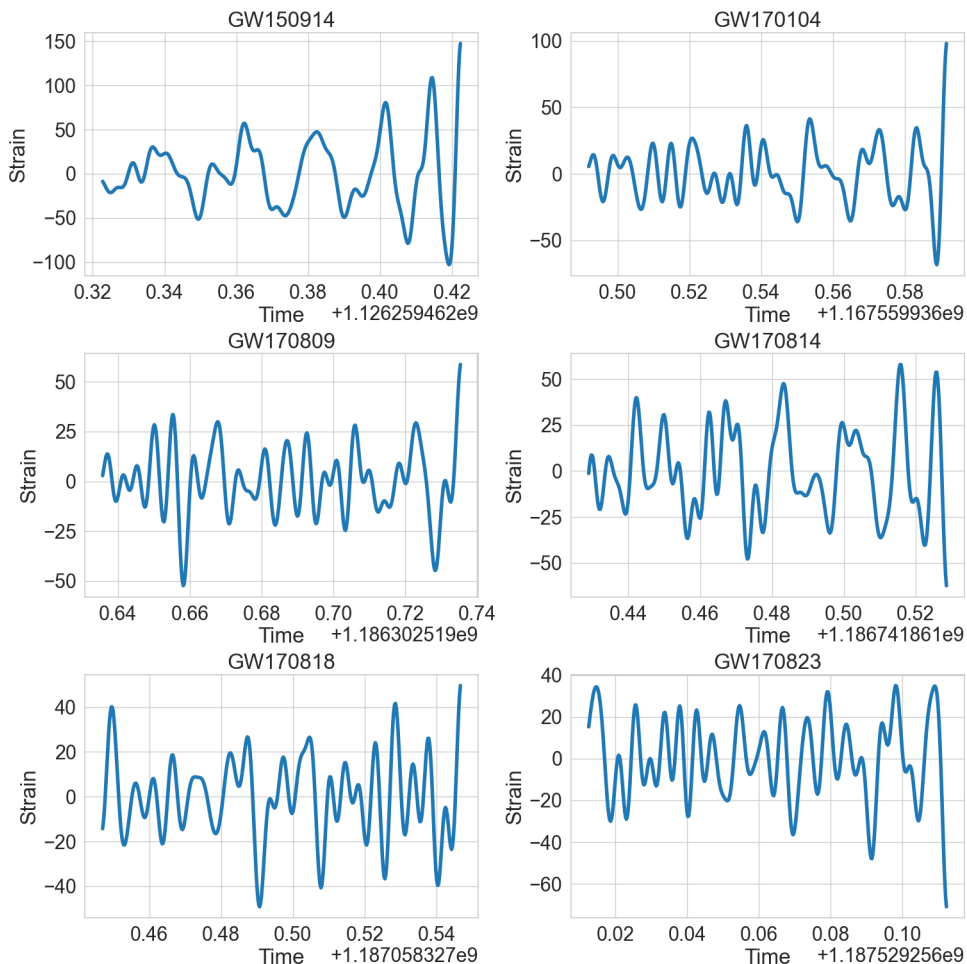


Figure 15 – LIGO waves used in the analysis from the *PyCBC* library.

As we did with our database we transform the waves into their histograms with 300 bins to make it inside our neural network. The results of it can be found in Table 4

and the distribution for the error can be found in Figure 16. A constant outlier is found on the tests with the LIGO data, this outlier goes above our standard deviation found on the first our test data. The mean of the error lies at -11.58 , the standard deviation is 13.99 , and the mean squared error is 329.55 . Considering that our range of data is 20 we could get an error with 67% of imprecision.

Table 4 – Results with LIGO Data - Round 1

| Waves | M1 | | | | M2 | | | |
|----------|-------|--------|------------|---------|-------|--------|------------|---------|
| | LIGO | Output | Difference | Diff. % | LIGO | Output | Difference | Diff. % |
| GW150914 | 30.60 | 27.60 | 3.00 | 10 % | 35.60 | 24.30 | 11.30 | 32 % |
| GW170104 | 20.00 | 49.20 | -29.20 | 146 % | 30.80 | 57.80 | -27.00 | 88 % |
| GW170809 | 23.80 | 50.80 | -27.00 | 113 % | 35.00 | 60.00 | -25.00 | 71 % |
| GW170814 | 25.20 | 47.00 | -21.80 | 87 % | 30.60 | 54.40 | -23.80 | 78 % |
| GW170818 | 26.70 | 44.60 | -17.90 | 67 % | 35.40 | 50.80 | -15.40 | 44 % |
| GW170823 | 29.00 | 35.50 | -6.50 | 22 % | 39.50 | 33.30 | 6.20 | 16 % |

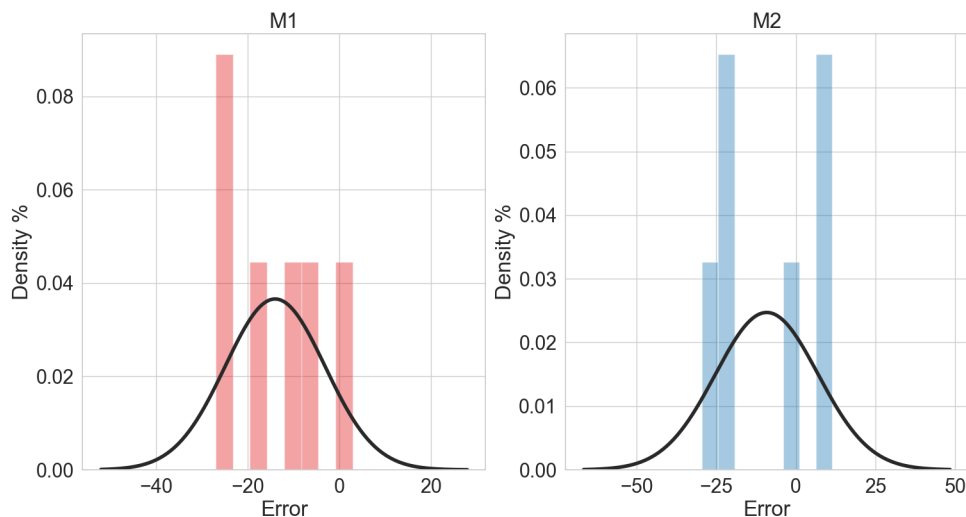


Figure 16 – Histogram of the difference between target and prediction using the best found neural network. On the left (blue color) we have the first mass (M1) difference and on the right (red color) we have the second mass (M2)

That was when we could not find a good fit for this experiment around the problem we wanted to solve. Furthermore, it is possible to say that our model is very good for our modeling data. But, it does not take what is needed for the real world problem.

5 Second Round of Experiments

This round was made using the Google Colaboratory platform (GOOGLE...). We still used the same Python libraries as the First Round 4.

5.1 Methodology

After the experiments did not have a good performance on the LIGO real data, we went to give a deeper look into the data we have modeled. We could notice a distortion (Figure 17) on our waves, which could lead to data that did not reflect precisely the event of gravitational waves that we were trying to represent. When these references were used, there was a distortion of the information that was significant when we wanted to obtain more accurate information from it.

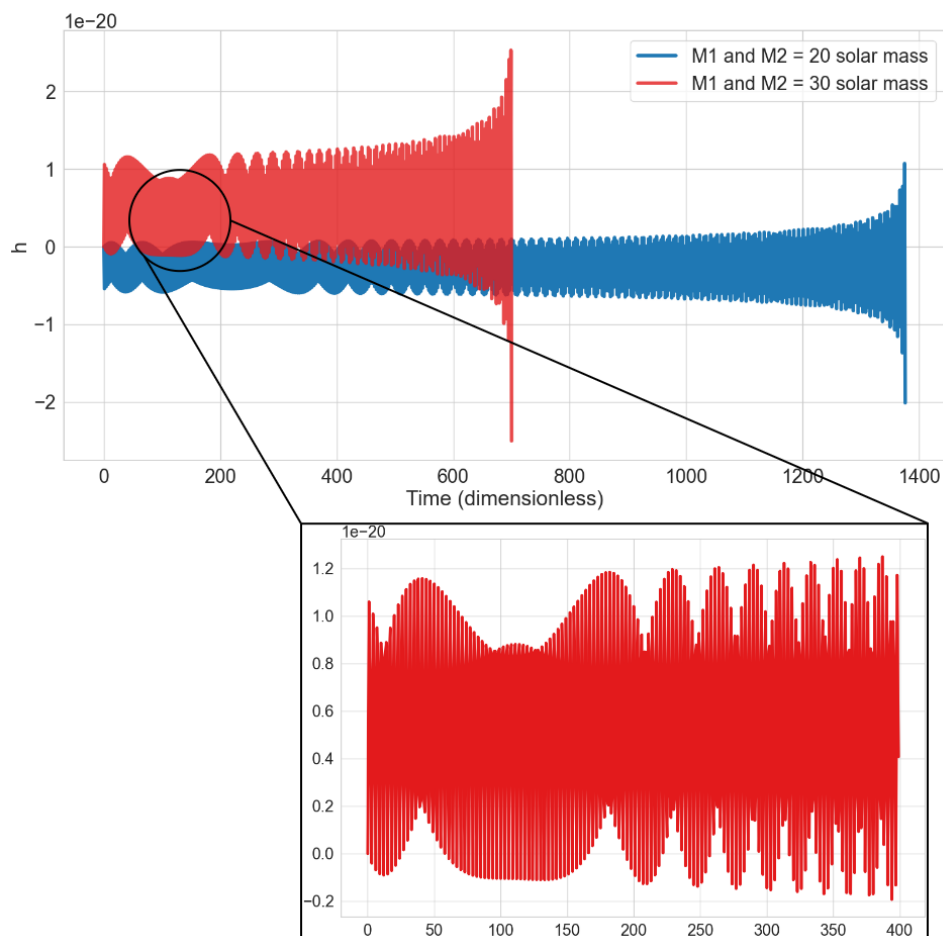


Figure 17 – Zoom on a wave with M1 and M2 mass equal to $20 M_{\odot}$

Back to our problem, we tried on curve fit techniques to find new pre-set values for the equations. Unfortunately, we had another round of unsuccessful experiments. At the time it was more important to keep the data as close as possible to the representation of the gravitational waves event, therefore instead of modeling our own data we used a data set already with its own equations using the PyCBC library (Biwer et al., 2019).

For this data set, we simulated another 80,200 waves, with the same mass range between $20M_{\odot}$ and $40M_{\odot}$. The simulated waveform representation can be seen in Figure 18, and the code is on A.3.

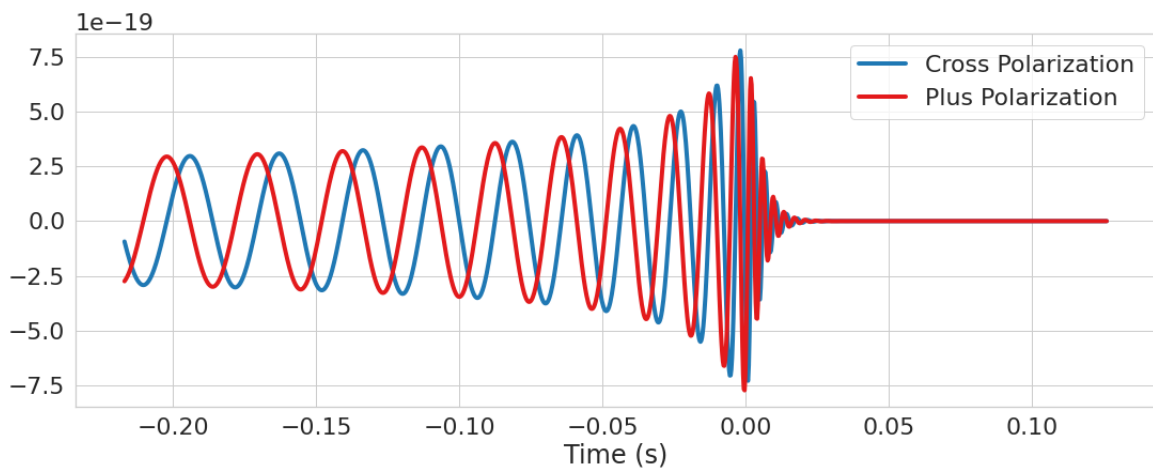


Figure 18 – Simulated waveform from PyCBC library.

It is possible to see in Figures 18 and 19 that there are two line waves (cross-polarization and plus polarization) for our experiments we decided to use cross-polarization, mathematically there would be no difference in which one to select.

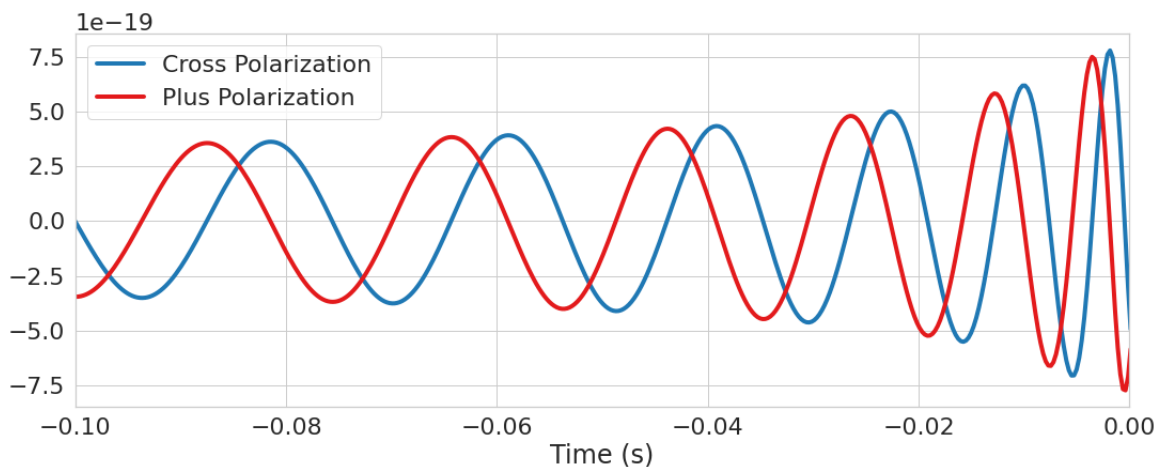


Figure 19 – Waveform simulated from PyCBC library used as input to the Neural Network. It has exactly 409 points. Starts -0.1 second before the objects merge and end at the exact point of the merge.

We ran the experiments using the same combinations of hyperparameters and Neural Network as in Chapter (4), the only difference is that rather than use 300 histogram point for our Neural Network input we used 409 wave amplitude points (Figure 19).

5.2 Results

We ran a preliminary experiment 1,000 epochs twice using the Grid Search technique to select the parameters. Considering the average value of the mean squared error for the validation data for each neural network, we could find that the best performance was the network with the Sin activation function, which had 2 hidden layers, 200 hidden layer unit on each hidden layer and a learning rate of $10e^{-2}$ (Table 5), with the average score of 4.337. In Table 5, there are two neural networks results, the second one was noticed because had the same configuration as the best one, except for the learning rate, which was 10 times smaller, this tells us that the neural network takes more time to archive a minimum, so we reinforce its training to know if it could achieve an even better result than the best performance neural network until now.

Table 5 – Second Round Experiment - Preliminary Result

| Hidden Layer | Hidden Layer Units | Learning Function | MSE Score | | |
|-----------------|-----------------------|----------------------|-----------|------------|-------|
| | | | Training | Validation | Test |
| 2 | 200 | $1e10^{-2}$ | 4.327 | 4.337 | 4.416 |
| 2 | 200 | $1e10^{-3}$ | 5.291 | 5.339 | 5.424 |

Thereafter using the early stop technique we ran another round of experiments with both neural networks. Resulting in the neural network with 10^{-3} learning rate having the best result, considering the validation data rate, which attained 17,798 epochs and 1.595 on the validation set, on the training set 1.525 and 1.589 with the test set. The results of both neural networks can be seen in Table 6

Table 6 – Second Round Experiment - Early Stop

| Hidden Layer | Hidden Layer Units | Learning Function | Epochs | MSE Score | | |
|--------------|--------------------|-------------------|--------|-----------|------------|-------|
| | | | | Training | Validation | Test |
| 2 | 200 | $1e10^{-2}$ | 2,815 | 3.818 | 3.826 | 3.902 |
| 2 | 200 | $1e10^{-3}$ | 17,798 | 1.525 | 1.595 | 1.589 |

To see the closeness of our model with the test data we plot the histogram of the error for it. The histogram shows the error distribution of (Test data - Foretasted Test Data) (Figure 20), for both masses.

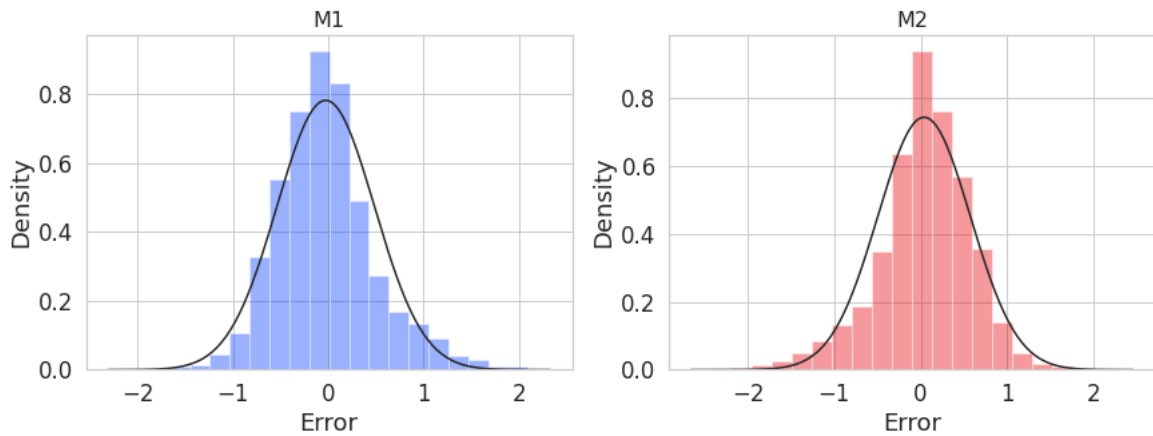


Figure 20 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right.

5.2.1 Test with LIGO Data

Once more for our final test of the network, we used the LIGO real data to analyze the neural network. The results can be found in Table 7.

The Figure 21 below shows the histogram of the difference between LIGO masses results and our model output masses. Together both masses have their error with a mean of 14.1, the variance was 147.71, the standard deviation on 12.15 and the MSE was 346.52. Considering our range of 20, between $20M_{\odot}$ and $40M_{\odot}$ the percentage difference is 53%.

Table 7 – Results with LIGO Data - Round 2

| Waves | M1 | | | | M2 | | | |
|----------|-------|--------|------------|---------|-------|--------|------------|---------|
| | LIGO | Output | Difference | Diff. % | LIGO | Output | Difference | Diff. % |
| GW150914 | 30.60 | 12.50 | 18.10 | 59% | 35.60 | 10.90 | 24.70 | 69% |
| GW170104 | 20.00 | 11.20 | 8.80 | 44% | 30.80 | 10.70 | 20.10 | 65% |
| GW170809 | 23.80 | 19.10 | 4.70 | 20% | 35.00 | 19.90 | 15.10 | 43% |
| GW170814 | 25.20 | 11.80 | 13.40 | 53% | 30.60 | 13.60 | 17.00 | 56% |
| GW170818 | 26.70 | 13.50 | 13.20 | 49% | 35.40 | 13.50 | 21.90 | 62% |
| GW170823 | 29.00 | 26.90 | 2.10 | 7% | 39.50 | 31.50 | 8.00 | 20% |

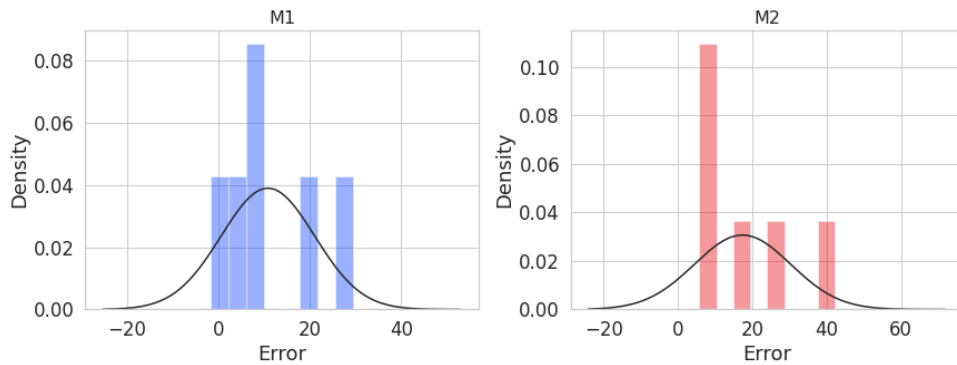


Figure 21 – LIGO Data test results histograms. On the left (blue) is the M1, and on the right (red) is the M2. There is a normal wave fitted to the data, it is possible to see it does not lay on the data well. For the M1 the mean is 10.80 and the standard derivation 10.24. For M2 the mean is 17.39 and the standard derivation 13.

6 Third Round of Experiments

6.1 Methodology

After our neural network does not be able to perform a good regression value for real data we went for a third round, now in insert noise on the simulated waves following the steps below:

1. Collect the LIGO that was used for comparison: GW150914, GW151012, GW170104, GW170729, GW170809, GW170814, GW170818, and GW170823.
2. Find the equivalent waves, with the same M1 and M2 masses on the modeled database.
3. Subtract the modeled wave data from the LIGO wave, for each of the ten waves. Resulting in noise for each of the waves.
4. Summarize the errors found and divide by ten, getting an average error for all the waves.
5. Apply the same average error found on each of the 80,200 modeled waves we had.

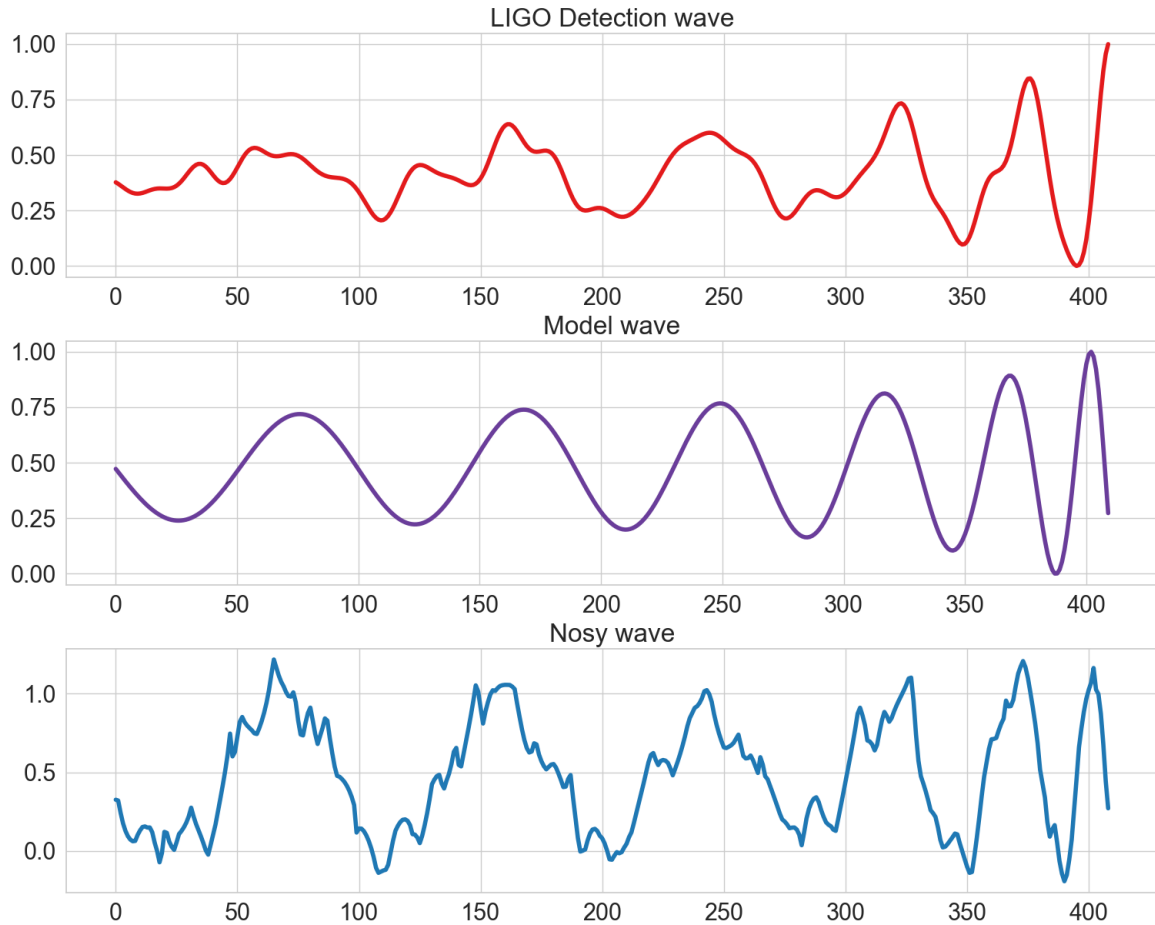


Figure 22 – Test data error histogram for Second Round. First mass on the left and the second mass on the right.

The Figure 22 illustrates our process. On the first wave, on red, we have the LIGO detection of GW150914, after the LIGO filter and signal process are applied (ABBOTT; AL., 2016). The second wave, in purple, is the simulated wave, generated using PyCBC with M_1 equaling 30.6 and M_2 equals 35.6. On the third graph is the subtraction of the detection wave per the simulated wave, leaving behind the noise.

6.2 Results

With this new noisy data, we ran the same experiments as Phase 2, a Grid Search within 6 times. For the preliminary, we had the neural network with 5 hidden layers, 300 hidden layer units, Sin as activation function and we used a learning rate of 1^{-2} . In Table 8, we can see the networks with the smallest rates for the preliminary experiments.

Table 8 – Third Round Experiment - Preliminary

| Hidden Layer | Hidden Layer Units | Learning Function | MSE Score | | |
|--------------|--------------------|-------------------|-----------|------------|-------|
| | | | Training | Validation | Test |
| 5 | 300 | $1e10^{-2}$ | 1.962 | 1.235 | 1.944 |
| 5 | 200 | $1e10^{-2}$ | 2.922 | 2.74 | 2.913 |

Afterward, we ran an early stop with the network of 300 hidden layers units. It took 1260 epochs (Table 9). The performance of the network over the epochs is placed in Figure 23, where we can notice a wide drop a little before the 100 epochs. The behavior on the last 260 epochs can be seen closely in Figure 24, a early stop technique prevented to the increase of error on the algorithm. The results can be found in Table 9.

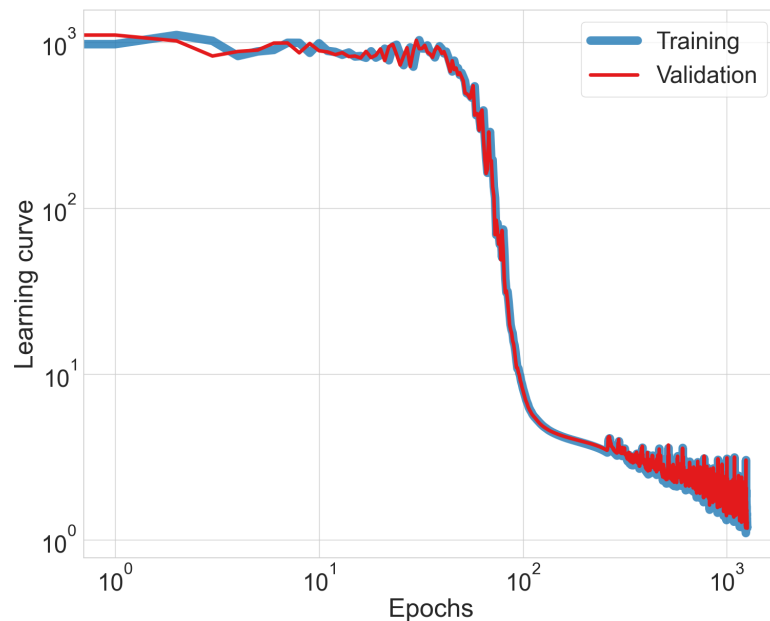


Figure 23 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right.

In Figure 24 the curve is oscillating in a way that the general error is meaningless for us. We also tried to variate the learning rate to 10^{-1} but the results were not newsworthy, even with a smaller learning rate and more time to archive a better result the learning rate of 10^{-2} had a better result of all networks.

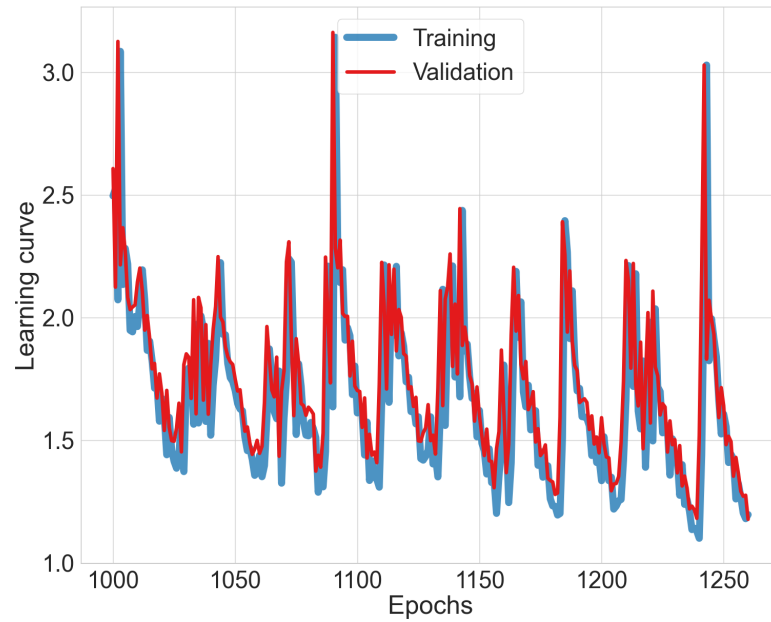


Figure 24 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right.

In the Table below we can see the results of the metrics. Also, the Figure 25 shows us the error distribution for the test data.

Table 9 – Third Round Experiment - Early Stop

| MSE Score | | |
|-----------|------------|-------|
| Training | Validation | Test |
| 1.096 | 1.177 | 1.190 |

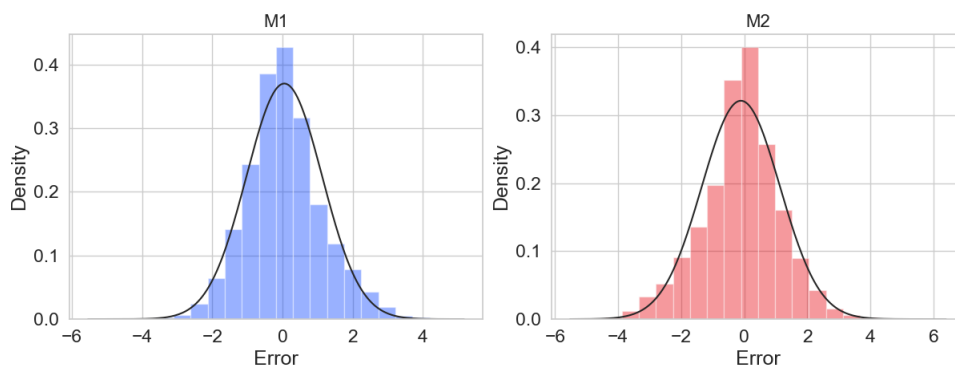


Figure 25 – Test data error histogram for Second Round. The first mass is on the left and the second mass is on the right.

6.2.1 Test with LIGO Data

Finally, we used the network on LIGO real data to analyze the neural network. The results can be found in Table 7.

Table 10 – Results with LIGO Data - Phase 3

| Waves | M1 | | | | M2 | | | |
|----------|-------|--------|------------|---------|-------|--------|------------|---------|
| | LIGO | Output | Difference | Diff. % | LIGO | Output | Difference | Diff. % |
| GW150914 | 30.60 | 23.10 | 7.50 | 25 % | 35.60 | 34.40 | 1.20 | 3 % |
| GW170104 | 20.00 | 25.00 | -5.00 | 25 % | 30.80 | 31.40 | -0.60 | 2 % |
| GW170809 | 23.80 | 21.80 | 2.00 | 8 % | 35.00 | 28.50 | 6.50 | 19 % |
| GW170814 | 25.20 | 24.20 | 1.00 | 4 % | 30.60 | 31.20 | -0.60 | 2 % |
| GW170818 | 26.70 | 24.80 | 1.90 | 7 % | 35.40 | 31.70 | 3.70 | 10 % |
| GW170823 | 29.00 | 20.60 | 8.40 | 29 % | 39.50 | 26.60 | 12.90 | 33 % |

The plot below (Figure 26) shows the error distribution. For the mean now we have -3.24 , a variance of 21.55 , a standard deviation of 4.64 , and an MSE of 32.03 . Considering the percentage it was 12% error.

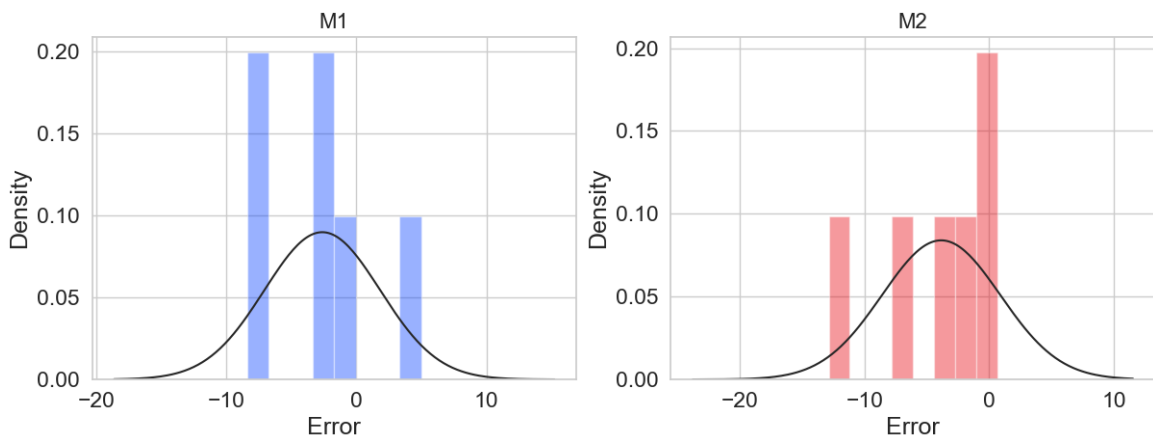


Figure 26 – LIGO Data test results histograms. On the left (blue) is the M1, and on the right (red) is the M2. There is a normal wave fitted to the data, it is possible to see it does not lay on the data well. For the M1 the mean is -2.62 and the standard derivation 4.44 . For M2 the mean is -3.85 and the standard derivation 4.75 .

With this round, we end our experiments for this work.

7 Result and Discussion

With the algorithm developed over this work, we were able to extract information about the fusion of astrophysical objects, through their gravitational wave signal. We can consider our results fast, cheap, and computationally low range.

We treat this work as cheap because it was made using personal computers. We also can consider it fast because of the training time of the neural networks used during the process and the inference time after we have the neural network weighted. The neural networks performed well and the training took some hours.

As a result, we developed it in three phases. In Round 1 (Chapter 4) we build our own database, generating the waves ourselves, using the technique described in (RUBBO et al., 2007). The network input was the wave's histograms. We archived an MSE of 4.44 for our simulated database, but for the LIGO data, we did not archive a satisfactory result, with an MSE of 419.40.

Since the need for improvement for LIGO data was clear, we insist on another round of experiments. Due to a numeric distortion on the waves of our generated database, we decide to persuade another database, for that we used the PyCBC library, where we could generate our database with only the input of the masses. We also insert Sin as the activation function, and we found it was the best performance at the moment, with an MSE for the test data on 3.902. Round 2 (Chapter 5) shows us that we still need to work on another database for the LIGO data, our mean error is 14.1 and we could get a 53% error still.

Since we saw some progress in Round 2, we went for our final round of experiments (Chapter 6). Now we insert some noise on our waves instant of input the simulated signal itself. With that, we found a more promising result for our test data and the LIGO data. For the test data, we found an MSE of 1.19, and for the LIGO data, we saw a decrease in the standard deviation dropping to 4.64. Which lead us to an error of 12%.

Therefore, the error reached in the third experimental round still is relatively high, where it is around 12%. However, the masses estimates are a very good first approximation, given the computational infrastructure needed and the avoid knowledge about sophisticated numerical relativity methods employed as the true procedure to execute inference of these masses. In this way, the simplicity of the proposed methodology is the advantage to obtain

an initial astrophysical body's mass estimation.

Finally, we can affirm, we apply a methodology that is well known on the computer science field, but not widespread on the physics, more specifically, gravitational waves field. Hence we have brought innovation in the field of astroinformatics.

7.1 Future work

This dissertation tries to find a valuable algorithm capable of characterizing a very complex natural phenomenon. Our preliminary results are satisfactory because they show the possibility of doing it with limited resources. Notwithstanding we went for continuous improvement of results in this work, there is space for more development. If continue from here it could be done with a superior reflection of the noises being inserted into the data. Also better processing of LIGO data and its filters. A study on the automation of searching for the best neural network architecture.

7.2 Published work

During Round 1 (Chapter 4) of our experiments, we published some results on XV Congresso Brasileiro de Inteligência Computacional (CBIC 2021), under the title Mass Determination of Black Holes from Gravitational Wave Data Using Neural Networks, in the section of Applications of Computational Intelligence in Physics.

The results reached in the second and third experimental rounds have great potential to be published. Nowadays, they are being formatted for a future paper publication.

Bibliography

ABBOTT, B.; ABBOTT, R.; ABBOTT, T.; ABRAHAM, S.; ACERNESE, F.; ACKLEY, K.; ADAMS, C.; ADHIKARI, R.; ADYA, V.; AFFELDT, C.; AL. et. Gwtc-1: A gravitational-wave transient catalog of compact binary mergers observed by ligo and virgo during the first and second observing runs. **Physical Review X**, American Physical Society (APS), v. 9, n. 3, Sep 2019. ISSN 2160-3308. Disponível em: <<http://dx.doi.org/10.1103/PhysRevX.9.031040>>.

ABBOTT, B. P.; AL. et. Observation of gravitational waves from a binary black hole merger. **Phys. Rev. Lett.**, American Physical Society, v. 116, p. 061102, Feb 2016. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.116.061102>>.

ABBOTT, R.; ABBOTT, T.; ABRAHAM, S.; ACERNESE, F.; ACKLEY, K.; ADAMS, A.; ADAMS, C.; ADHIKARI, R.; ADYA, V.; AFFELDT, C.; AL. et. Gwtc-2: Compact binary coalescences observed by ligo and virgo during the first half of the third observing run. **Physical Review X**, American Physical Society (APS), v. 11, n. 2, Jun 2021. ISSN 2160-3308. Disponível em: <<http://dx.doi.org/10.1103/PhysRevX.11.021053>>.

BEZGINOV, N.; VALDEZ, T.; HORBATSCH, M.; MARSMAN, A.; VUTHA, A. C.; HESSELS, E. A. A measurement of the atomic hydrogen lamb shift and the proton charge radius. **Science**, v. 365, n. 6457, p. 1007–1012, 2019. Disponível em: <<https://www.science.org/doi/abs/10.1126/science.aau7807>>.

BISHOP, C. M. **Neural Networks for Pattern Recognition**. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN 0198538642.

Biwer, C. M.; Capano, C. D.; De, S.; Cabero, M.; Brown, D. A.; Nitz, A. H.; Raymond, V. PyCBC Inference: A Python-based Parameter Estimation Toolkit for Compact Binary Coalescence Signal. , v. 131, n. 996, p. 024503, fev. 2019.

EINSTEIN, A. Näherungsweise integration der feldgleichungen der gravitation. **Königlich Preußische Akademie der Wissenschaften**, 1916.

FERREIRA, T. A. E. **Uma nova metodologia híbrida inteligente para a previsão de séries temporais**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2006. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/2640>>.

FEURER, M.; HUTTER, F. **Hyperparameter Optimization**. Cham: Springer International Publishing, 2019. 3–33 p. ISBN 978-3-030-05318-5. Disponível em: <https://doi.org/10.1007/978-3-030-05318-5_1>.

GEORGE, E. A. H. D. Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data. **Physics Letters B**, **778 (2018) 64-70**, 2017.

GOOGLE Colaboratory FAQ. <<https://research.google.com/colaboratory/faq.html>>. Accessed: 2022-08-02.

HAEGEL, L.; HUSA, S. Predicting the properties of black-hole merger remnants with deep neural networks. **Classical and Quantum Gravity**, IOP Publishing, v. 37, n. 13, p. 135005, jun 2020. Disponível em: <<https://doi.org/10.1088/1361-6382/ab905c>>.

HAYKIN, S. **Neural Networks and Learning Machines**. 3. ed. [S.l.]: Pearson Education, Inc., 1999. ISBN 978-0-13-147139-9.

HOBSON, M. P.; EFSTATHIOU, G. P.; LASENBY, A. N. **General Relativity An Introduction for Physicists**. 1. ed. [S.l.]: Cambridge University Press, 2006. ISBN 978-0-511-13795-2.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0893608089900208>>.

Hulse, R. A.; Taylor, J. H. Discovery of a pulsar in a binary system. **apjl**, v. 195, p. L51–L53, jan. 1975.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2017.

LIGO Laboratory. **Why Detect Them?** 2015. Disponível em: <<https://www.ligo.caltech.edu>>.

LIGO-Virgo/Northwestern Univ./Frank Elavsky. **Masses of LIGO/Virgo detections**. 2020. [Online; accessed August 7, 2022]. Disponível em: <https://www.ligo.org/detections/images/MassPlot_graveyard_190814.png>.

LIGO/T. Pyle. **Spiral Dance of Black Holes**. 2016. [Online; accessed July 08, 2022]. Disponível em: <<https://www.ligo.caltech.edu/image/ligo20160615f>>.

MARION, J. B.; THORNTON, S. T. **Classical dynamics of particles and systems**. 5. ed. [S.l.]: Cengage Learning, 2003. ISBN 978-0534408961.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **ICML 2010**. [S.l.: s.n.], 2010. p. 807–814.

Nobel Prize in Physics. **The Nobel Prize in Physics 1993**. 2019. Disponível em: <<https://www.nobelprize.org/prizes/physics/1993/press-release/>>.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; ALCHÉ-BUC, F. d'; FOX, E.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 32**. [S.l.]: Curran Associates, Inc., 2019. p. 8024–8035. [Http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learnin-g-library.pdf](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learnin-g-library.pdf).

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical Recipes 3rd Edition: The Art of Scientific Computing**. 3. ed. Cambridge University Press, 2007. ISBN 0521880688. Disponível em: <http://www.amazon.com/Numerical-Recipes-3rd-Scientific-Computing/dp/0521880688/ref=sr_1_1?ie=UTF8&s=books&qid=1280322496&sr=8-1>.

RUBBO, L. J.; LARSON, S. L.; LARSON, M. B.; INGRAM, D. R. Hands-on gravitational wave astronomy: Extracting astrophysical information from simulated signals. **Utah State University DigitalCommons@USUs**, All Physics Faculty Publications, ago. 2007.

RUMELHART, D. E.; MCCLELLAND, J. L.; PDP Research Group (Ed.). **Parallel Distributed Processing. Volume 1: Foundations**. Cambridge, MA: MIT Press, 1986.

SANTOS, G. R.; FIGUEIREDO, M. P.; SANTOS, A. d. P.; PROTOPAPAS, P.; FERREIRA, T. A. E. **Gravitational Wave Detection and Information Extraction via Neural Networks**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2003.09995>>.

SCHUTZ, B. F. **A First Course in General Relativity**. 2. ed. [S.l.]: Cambridge University Press, 2009. ISBN 978-0-511-53995-4.

T. Pyle/Caltech/MIT/LIGO Lab. **Massive Bodies Warp Space-Time**. 2016. [Online; accessed July 08, 2022]. Disponível em: <<https://www.ligo.caltech.edu/image/ligo20160211e>>.

THORNE, K. **The Science of Interstellar**. 1. ed. W. W. Norton Company, 2014. ISBN 9780393351378; 0393351378; 9780393351385; 0393351386. Disponível em: <libgen.li/file.php?md5=cf3f0b879bca735fcb3ba8533d854ffc>.

WHEELER., C. W. M. . K. S. T. . J. A. **Gravitation**. 1. ed. [S.l.]: New York, NY : Freeman, 2008. ISBN 0716703343; 9780716703341; 0716703440; 9780716703440.

APPENDIX

APPENDIX A – Code

A.1 Wave generator

```

1 def wave_generator(chirp_mass, luminosity=1.0):
2
3     c = 3.0e8 # light speed
4     G = 6.674e-11 # Newton's gravitational constant
5     k = (96/5)*(2*math.pi)**(8/3)*(G*chirp_mass/c**3)**(5/3)
6     r = luminosity
7     x = 2.0*(G*chirp_mass)**(5/3)/(r*c**4)
8     P0 = 0.5 # initial orbital period
9     fi = 0.0 # initial wave's phase
10    dt = 0.001 # time frame interval
11    t = 0.0
12    h = 0.0
13    strain = []
14
15    while t < (3*P0**(8/3)/(8*k)):
16        strain.append(h/1e23)
17        pgw = 0.5*(P0^(8/3)-(8/3)*k*t)^(3/8)
18        amp = x*(math.pi/pgw)^(2/3)
19        phi = fi - (4.0/5.0)*math.pi*(3.0*P0^(8/3)-8.0*k*t)/
20            (k*(P0^(3/8)-(8/3)*k*t)^(3/8))
21        h = h + amp * math.cos(phi)
22        t = t + dt
23    return strain

```

A.2 Generate *chirp mass*

```
1 sm = 1.989e30 # Sun mass
2
3 for m1 in range(20,40,0.1):
4     for m2 in range(m1,40,0.1):
5         chirp_mass = calculate_chirp_mass(m1*sm, m2*sm)
6         if chirp_mass not in chirp_mass_list:
7             chirp_mass_list.append(chirp_mass)
```

A.3 PyCBC wave generator

```
1 import numpy as np
2 import pandas as pd
3
4 min_mass, max_mass = 20.0, 40.0
5
6 values_list = []
7 times_list = []
8 mass1_list = []
9 mass2_list = []
10
11 for m1 in np.arange(min_mass, max_mass, 0.05):
12     for m2 in np.arange(m1, max_mass, 0.05):
13         m1 = round(m1, 2)
14         m2 = round(m2, 2)
15
16         hp, hc = get_td_waveform(approximant="SEOBNRv4_opt", mass1=m1,
17                                 mass2=m2, delta_t=1.0/4096, f_lower=30)
18
19         times = np.array(hp.sample_times)
20         wanted_index = np.where((times >=-0.1) & (times <=0))
21
22         times = times[wanted_index]
23         values = np.array(hp[wanted_index])
24
25         m1_arr, m2_arr = np.empty(times.size), np.empty(times.size)
26         m1_arr.fill(m1)
27         m2_arr.fill(m2)
```