Leibniz Universität Hannover

and

Max Planck Institute for Gravitational Physics (Albert Einstein Institute)

Master thesis

Analysis of Gravitational-Wave Signals from Binary Neutron Star Mergers Using Machine Learning

Marlin Benedikt Schäfer

Supervisors: Dr. Frank Ohme and Dr. Alexander Harvey Nitz

May 20, 2019

This page is intentionally left blank. LÖSCHEN!!! Damit Eigenständigkeitserklärung nicht auf Rückseite gedruckt ist.

I hereby assure that the thesis at hand has been constituted independently and without the use of any other than the cited sources. I furthermore assure, that all passages taken textually or analogously from other sources are marked as such.
This thesis, in its current or a similar form, has not been submitted to any other examination office.

Hiermit versichere ich, dass die vorliegende Arbeit selbständig und ohne Verwendung anderer Quellen, als den angegebenen, verfasst wurde. Zudem versichere ich, dass alle Stellen, die wörtlich oder sinngemäß aus anderen Quellen entnommen wurden, als solche gekennzeichnet sind.
Diese Arbeit hat so oder in einer ähnlichen Form noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum	Marlin Benedikt Schäfer

This page is intentionally left blank. LÖSCHEN!!! Damit Inhaltsverzeichnis nicht auf Rückseite gedruckt ist.

Abstract

Put the abstract here

This page is intentionally left blank. LÖSCHEN!!! Damit Inhaltsverzeichnis nicht auf Rückseite gedruckt ist.

Contents

This page is intentionally left blank. LÖSCHEN!!! Damit erste Seite nicht auf Rückseite gedruckt ist.

3 Neural networks

Explain the use for this section.
Neural networks are machine learning algorithms inspired by research on the structure and inner workings of brains. [Insert quote (Rosenblatt?)] Though in the beginning NN were not used in computer sciences due to computational limitations [Citation] they are now a major source of innovation across multiple disciplines. Their capability of pattern recognition and classification has already been successfully applied to a wide range of problems not only in commercial applications but also many scientific fields. [Quote a few scientific usecases here. Of course using the one for gw but also other disciplines.]
Major use cases in the realm of gravitational wave analysis have been classification of glitches in the strain data of GW-detectors [Citation] and classification of strain data containing a GW versus pure noise [Citation]. A few more notable examples include [list of citations].
In this section the basic principles of NN will be introduced and notation will be set. The concept of backpropagation will be introduced and extended to a special and for this work important kind of NN. (maybe use the term ”convolution” here already?) It will be shown that learning in NN is simply a mathematical minimization of errors that can largely be understood analytically.

3.1 Neurons, layers and networks

What is the general concept of a neural network? How does it work? How does backpropagation work? How can one replicate logic gates? (cite online book)
The basic building block of a NN is - as the name suggests - a neuron. This neuron is a function mapping inputs to a single output. In the early days this output was always either 1 or 0 [Citation], whereas nowadays it is usually some real number.
In general there are two different kind of inputs. Those that are specific to the neuron itself and those that the neuron receives as an outside stimulus. We write the neuron as

$$f : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}; \quad (\vec{x}, \vec{w}, b) \mapsto f(\vec{x}, \vec{w}, b) := a(\vec{w} \cdot \vec{x} + b), \tag{3.1}$$

where $\vec{x}$ is the outside stimulus, $\vec{w}$ are called weights, b is a bias value and a is a function known as the *activation function*(change this to not be emphasized if it is not used for the first time here). The activation function is a scalar function

$$a : \ \mathbb{R} \to \mathbb{R} \tag{3.2}$$

determining the scale of the output of the neuron. To understand the role of part of the neuron, consider the following activation function:

$$a(y) = \begin{cases} 1, & y > 0 \\ 0, & y \leq 0 \end{cases}. \tag{3.3}$$

This tells us, that the neuron will only send out a signal if the input to the activation function is greater than 0. Therefore in order for the neuron to activate, the weighted

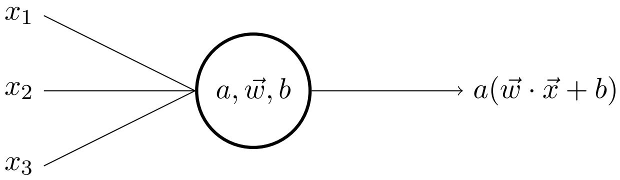

Figure 1: Depiction of a neuron with inputs $\vec{x} = (x_1, x_2, x_3)^T$, weights $\vec{w}$, bias b and activation function a.

sum of the inputs $\vec{w} \cdot \vec{x}$ has to be larger than the negative bias b. When a NN is training, i.e. learning, it tweaks these weights and biases to achieve its goal.
The effects of changing the weights make some input more or less important. The closer a weight w_i is to zero, the less impact the corresponding input value x_i will have. Choosing a negative weight w_i results in the corresponding input x_i being inverted, i.e. the smaller the value of x_i the more likely the neuron is to activate and vice versa.
Explain what changing the bias does.

3.2 Specific layers

Explain that there are not just dense layers.

3.2.1 Dense layer

What is a dense layer, how does it work (can maybe be omitted)

3.2.2 Convolution layer

What are the advantages of convolution layers and why do we use them? Disadvantages?

3.2.3 Inception layer

Explain what it is, how it works. (cite google paper) ONLY IF IT IS REALLY USED IN THE FINAL ARCHITECTURE!

3.2.4 Batch Normalization layer

Explain how batch normalization works and why it is useful. (cite according paper)

3.2.5 Dropout layer

Explain what a dropout layer is, what it does, why it is useful.

3.2.6 Max Pooling layer

Explain what max pooling does and why it is useful, even when it is counter intuitive.

Glossary

GW Gravitationswelle/n.

NN Neural network.