Otto-von-Guericke-University Magdeburg
Faculty for Computer Science

A hybrid deep-learning approach for reliable real-time assessment of high magnitude earthquakes

Master Thesis

Author:
Viola Hauffe

Advisor:
Prof. Sebastian Stober

Other advisors:
Dr. Marius Kriegerowski
Jannes Münchmeyer

Magdeburg, 8th September 2021

Contents

Abstract

wie angekündigt hier die Paper, an die ich gedacht hatte, und nochmal eine Kurzbeschreibung davon, was ich bei Hybridsystem im Kopf hatte.

Das Gerät zeichnet kontinuierlich Wellenformen auf und auf denen müsste man dann Picking machen, also feststellen wann eine seismische Welle auftrifft. Dazu gibt es inzwischen einen ganzen Stapel Paper. Ein guter Einstieg ist Ross et al. (2018) "Generalized Seismic Phase Detection with Deep Learning". Entweder damit gekoppelt (indem man die Trainingsdaten geeignet zusammenstellt) oder als nächster Schritt, muss man unterscheiden, ob es tatsächlich ein Erdbeben oder nur impulsive noise ist. Hier gibt es zum Beispiel Li et al. (2018) "Machine learning seismic wave discrimination: Application to earthquake early warning".

Der übliche Deep Learning Ansatz wäre jetzt Magnitude und Lokalisation direkt zu schätzen, siehe z.B. Mousavi et al. (2019) "A Machine-Learning Approach for Earthquake Magnitude Estimation" und Mousavi et al. (2019) "Bayesian-Deep-Learning Estimation of Earthquake Location from Single-Station Observations". Mein Vorschlag war jetzt, Deep Learning nur für die Distanz zu nutzen und für die Magnitude eine parametrische Modellierung zu wählen. Das ist im early warning eine etablierte Methode, um Magnituden schnell zu schätzen (z.B. Zollo et al. (2006) "Earthquake magnitude estimation from peak amplitudes of very early seismic signals on strong motion records"). Für eine genauere Schätzung könnte man wahrscheinlich Teile der Methode auf Nutzung in Echtzeit adaptieren, die ich hier vorgeschlagen habe (Münchmeyer et al. (2020) "Low uncertainty multifeature magnitude estimation with 3-D corrections and boosting tree regression: application to North Chile"). Eine Echtzeit Magnitudenschätzung wäre schon ein exzellentes Ergebnis und könnte zum Beispiel ziemlich direkt für early warning verwendet werden.

Todo list

Index of Notation

In a lot of cases it makes sense to give an overview over your mathematical notation.

Mathematical

x	Point in 3D space
$\overrightarrow{\mathrm{xy}}$	Normalized direction vector from x to y
$\mathbf{v}$	Direction vector in 3D space
$\mathrm{p}_x, \mathbf{v}_x$	x component of point / vector
$\mathbf{v} \cdot \mathbf{w}$	Dot product of vectors $\mathbf{v}$ and $\mathbf{w}$
$(\mathbf{v} \cdot \mathbf{w})^+$	Dot product of vectors $\mathbf{v}$ and $\mathbf{w}$ with negative values clamped to zero
$\mathbf{v} \times \mathbf{w}$	Cross product of vectors $\mathbf{v}$ and $\mathbf{w}$
$\|\|\mathbf{v}\|\|$	Euclidean length of vector $\mathbf{v}$
$\hat{\mathbf{v}}$	Normalized vector $\mathbf{v}$

Quantities & Functions

A	**Area**
ω	**Solid Angle**
ϕ	**Radiant Flux**, light power
I	**Radiant Intensity**, flux density per solid angle
E	**Irradiance**, flux density per area
L	**Radiance**, flux density per area per solid angle
ρ	**Reflectance**, ratio between incoming and outgoing flux
f_r	**BRDF**, function on the relation between irradiance and outgoing radiance

1 Introduction

Earthquakes are still an ongoing thread for humans, infrastructure and buildings. There were some big earthquakes in the last years. (include photo) As technology improves early warning systems get more elaborate each year. Much work is put into researching earthquake types, earthquake physics simulation and the prediction of shaking or magnitude of an arriving earthquake.

1.1 Background

1.1.1 Seismometers

Since the 1900s seismographs were developed. Today's seismometers can measure ground motion very precisely. In earthquake rich countries there exit big networks of seismometers. We will also use such a network of seismometers in our dataset.

1.1.2 Strength of an earthquake

While we will have a look at the magnitude of an earthquake its important to keep in mind, that, while the magnitude is based on the physical properties of an earthquake, it is not the sole factor how strong the shaking is, or how much damage is going to occur.

1.2 Motivation

1.3 Goals

Algorithm goals: Safety before accuracy, detect big earthqaukes, even though they are underrepresented in data, estimate distance, detect, and estimate magnitude Algorithmusentwurf, Ziele Schnelligkeit, Embeddedgeeignet, läuft auf schlechten Sensordaten, sagt Magnitude, Epizentrum mit einer gewissen Sicherheit richtig voraus

1.4 Tasks

Algorithm implementation, testing

1.5 Data

what kind of data, which structure, which dataset

1.6 structure and contents of thesis

explain how the thesis is going to explain the work

2 Prerequisites

2.1 Earthquake Physics

2.1.1 The Earthquake Event

An earthquake are caused by movements within the earth crust. These might result from earth plates moving under or against each other or artificial causes like mining works. After building up a lot of stress against friction, rock fractures along a fault line while the other tectonic layers can slip past each other. The fault line can even be visible on the surface.

The point where the fracturing starts is called hypocenter, while the epicenter is this point's projection onto the surface as seen in figure 2.1. The breakage can extend to other points along the fault line as well as cause more ruptures.

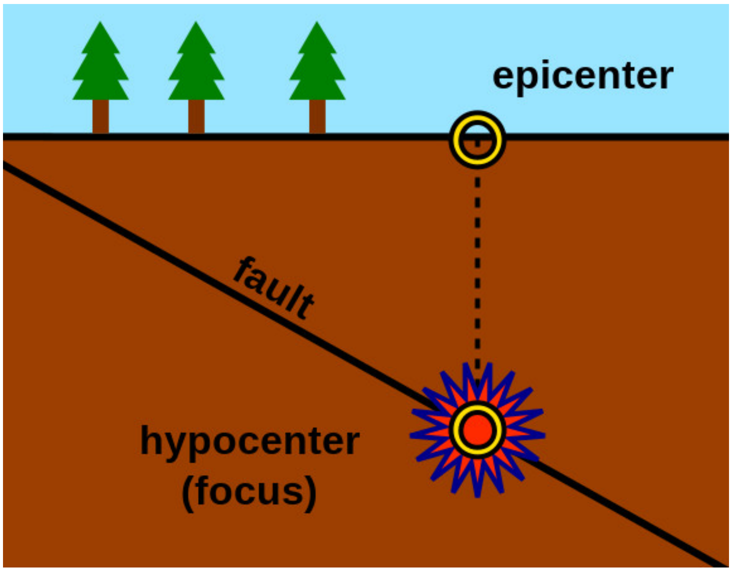


Figure 2.1: Hypocenter and Epicenter

2.1.2 Seismic Waves

The energy, which is emitted as the fracturing occurs, travels through the earth crust as seismic waves. We denote different types of waves, which travel at different velocities. Soil and material type (air, liquid or solid) also affect wave arrival times. Waves also reflect on certain materials, creating many sub-types of waves to arrive.

single, multi-station? distance computation?

magnitu computation and magnitude scales

azimuth and single/-multi-station

Station Codes

distance detection up to which point

epicentr and hypocen tral distance, depth (on an image)

magnitu klassisch

Machin learning intro

The first wave to arrive at a site is the p-wave or primary/pressure wave. The propagating waves moves the material back and forth and is therefore also known as a compressional wave.
S-waves, second or shear waves move the material in a right angle to the movement direction. Contrary to p-waves, s-waves cannot move through liquid and are slightly slower than p-waves. Figure 2.2 shows this relationship. S-Wave and P-Waves are often

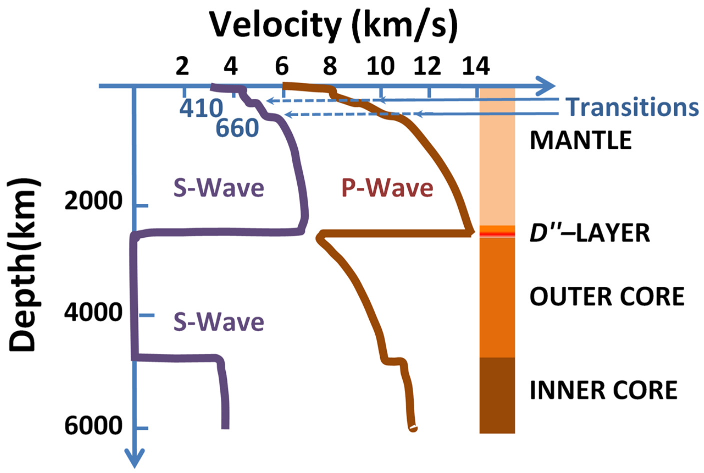


Figure 2.2: Velocity of S and p Wave

denoted as body waves. Surface waves arrive even later and are the most damaging ones. Two important waves are Rayleigh and Love waves. Rayleigh waves cause the surface to "ripple", they move the ground up and down while Love waves cause horizontal shearing. We can see all mentioned types of waves in figure 2.3.¸

2.2 Recording an Earthquake

2.2.1 Seismic Data

The seismic waves which arrive on a site can be recorded by a seismometer, which is the instrument recording ground motion as displacement, velocity or acceleration. A seismograph is the system build around the seismometer. Seismograms show the measured motions as a 1D signal for the three motion orthogonal axes, namely north-south, east-west and ground-up. The recordings depend on the sensitivity of the sensors, but also on earthquake and location. For further reading see Stein and Wysession [2003], page 398. An example for a recording of one axis at an earthquake arrival can be seen in figure 2.4. The change from noise to P-Wave to S-Wave arrival is clearly visible in the example.

2.2.2 Networks

Multiple seismometers might be organized in a network, where stations are linked together virtually or physically and their data can be processed nearly simultaneously. Network

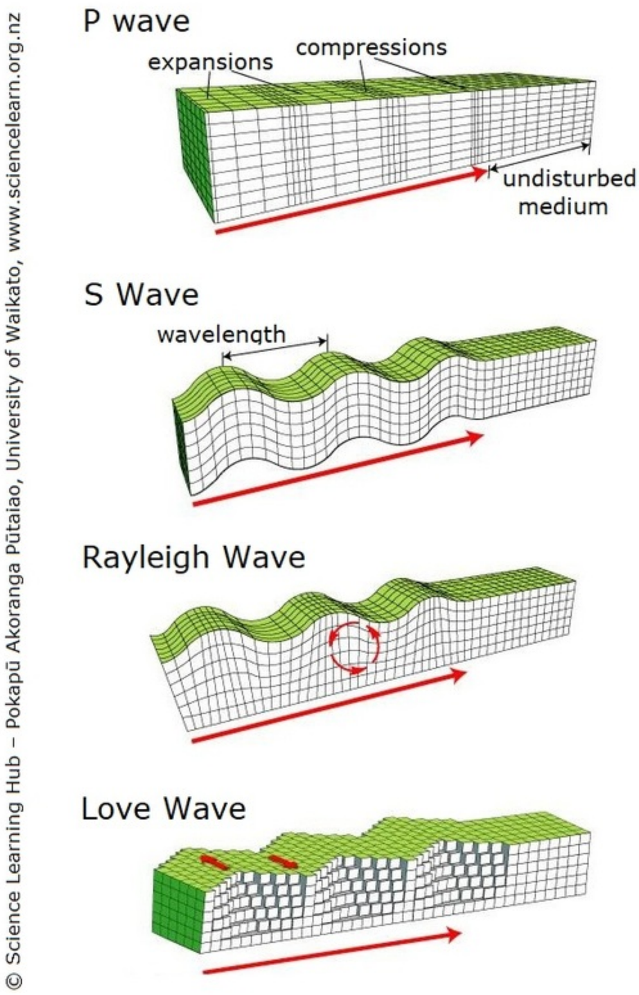


Figure 2.3: Overview of seismic waves

goals are often earthquake prediction with the according depth and distance, but network size, sensor type and network range can be drastically different depending on the type and magnitude of earthquake which must be detected. As the emitted waves get weaker depending on distance and soil type good coverage is crucial to get realistic predictions. This is especially important for location computation, as we need three or more sensors on the corners of a preferably equilateral triangle to get a precise location with the circle method. More information on the organisation of sensor networks can be found in Havskov and Alguacil [2016].

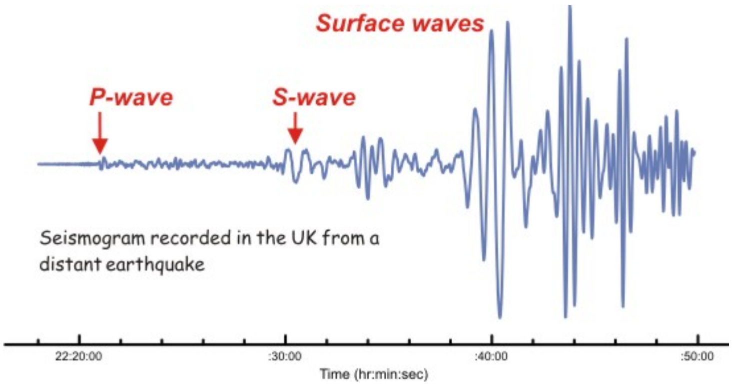


Figure 2.4: One axis recording of a seismograph

2.2.3 Early Warning

2.3 Differences between regression and classification

2.3.1 Typical networks and techniques

2.4 Estimation Representation

It is in important to keep in mind, that we do not take into account the uncertainty of our magnitude model and the representation of an earthquake with just ground motion sensors is itself a simplification.

2.4.1 Types of uncertainty

We divide between epistemic and aleatoric uncertainty.

- Epistemic uncertainty is our model uncertainty. The model can be unsure about certain parts of our dataset, because it has seen too few data points, but it can also be lacking the design to wholly learn and understand the dataset. The model can also be unsure, if we are unsure about the model's parameters for prediction or if we have multiple models to choose from when predicting.
  Epistemic uncertainty can be reduced by adding more training examples and choosing a good model architecture.
- Aleatoric uncertainty is uncertainty engrained in the data. This includes instrument noise and (over-)simplification of data information. We further divide between homoscedatic and heteroscedatic aleatoric noise. While homoscedatic uncertainty stays the same for all data point examples, as it would be with a constant sensor-offset noise, heteroscedatic uncertainty can be different for each input sample and is much harder to track and.

2.4.2 Uncertainties in our model

There are a few key points in the way our dataset is build which contribute to aleatoric uncertainty. While seismometers are extremely precise instruments, they are not completely free from sensor noise. We remove the sensor response in preprocessing, but they are still physical instruments, which cannot guarantee total precision, although it is more than enough for our model.
Even more, our learning goals, distance and magnitude are itself hard to determine. The distance to an earthquake has to be assumed from the emitted waves recorded by multiple seismometers and relies on certain simplifications of earthquake physics. As this is the same for the magnitude, the magnitude itself is a physical concept, that tries to capture the strength of an earthquake, but uses a simplified representation of the energy the earthquake emits, which again is only computed on the basis of our recorded data.
Epistemic uncertainty is far easier to detect. As also seen in /refdataset our data is not equally distributed and we are lacking examples for higher magnitudes and (naturally, as small earthquakes cannot be detected by remote sensors) higher distances.
As distance and magnitude are measurements of our sensor data, the model uncertainty also changes while the signal goes in and out, as earthquake waves arrive and depart. We do expect that, because both magnitude and distance are linked to the wavelength and amplitude of certain emitted waves.

2.4.3 Implementation

A very basic technique is taking the softmax output of a classification network as an uncertainty measure. While this is a good rule-of-thumb, most larger networks are overconfident in their predictions, meaning their accuracy is in general lower than the certainty that is suggested by the softmax output. Guo et al. [2017] propose a few methods for improvement.
For Bayesian neural networks instead of a single value a gaussian distribution is placed over the weights. Techniques similar to "Bayes by Backprop" Blundell et al. [2015] allow learning the weight distributions with an algorithm very similar to standard backpropagation. When predicting an example, the weights are sampled from the distribution every time, giving an uncertainty measure for this example after a reasonable number of runs.
In contrast to Bayesian neural networks, which require a change of loss functions and learning routines, "Monte Carlo Dropout" does not need this. Instead of just using dropout during training, Gal and Ghahramani [2016] apply dropout also when predicting examples and generate an uncertainty measure by aggregating the (all slightly different) outputs.
An even simpler method, that can be used for regression networks, is not learning a single valued output, but the mean and standard deviation of a gaussian distribution by changing the loss function and the last layer of the network. The mean can then be used as a output prediction, while the standard deviation gives an uncertainty.

3 Related Work

short introduction about early warning in literature?

3.1 Earthquake Detection

Perol et al. [2018] try to detect an earthquake by dividing the physical area into clusters with k-means. Their CNN then classifies an example as either noise or an earthquake event located in one of the six clusters.
Li et al. [2018] train a generative adversarial network so that the generator generates p-waves. The discriminator, which was trained to recognize true p-waves is then used partially as a feature extraction network. The features are used as an input to a random forest classifier, which can differentiate between p-wave and noise.
Ross et al. [2018] trained a CNN, which can classify a four-second waveform input as either P-wave, S-wave or noise. They conclude from their tests, that the algorithms detects the onset of a wave visually and is therefore not bound to events with a certain magnitude.
PhaseNet from Zhu and Beroza [2019] picks S-wave and P-wave arrivals on a 30sec-input. To deal with imprecise manually estimated picking times they place a gaussian distribution of the arrivals. PhaseNet is based on U-Net which uses down- and upsampling to process time-series data. cite?

3.2 Estimating magnitude and distance

Mousavi and Beroza [2020] propose a method for directly computing the earthquake magnitude from the waveforms in a single-station use. They use a CNN with LSTMs on a 30sec- three channel seismogram input. While their results look very promising, their dataset had just a few events with a magnitude above 5 and is therefore showing a poorer performance on higher magnitudes, or does not even take them into account.
Münchmeyer et al. [2021] build a multi-station real-time model for predicting magnitude and location. They use an attention based transformer network, which does feature extraction and feature combination while using multiple stations. The authors also denoted a saturation effect for magnitudes above 7, however the underestimation could be partially helped by transfer learning, as they had other datasets at hand.
Ristea and Radoi [2021] use Short-Time Fourier Transform on one minute long seismograms as an input to a complex CNN to compute magnitude, distance and depth on a single station model. Lomax et al. [2019] build on the work of Perol et al. [2018] and extend ConvQuakeNet to detect earthquakes in any location and compute magnitude and distance. This is an explorational work, where the authors use accuracy measurements for binned estimates of distance, magnitude, azimuth and depth.
Mousavi and Beroza [2019] use a network with dilated residual convolutions to estimate

epicentral distance an p-travel time on 1 minute long seismograms. For predicting the uncertainty they implement a loss function which learns a variance for each output parameter, as well as applying Monte Carlo Dropout.
Zhang et al. [2021] build two convolutional networks, one for a 3D location prediction and one for 1D magnitude predictions. They place a gaussian distribution over the labels and use down- and then upsampling in their networks.

3.3 Parametric Methods

Kuyuk and Allen [2013] compare different parameters for the best regressional fit when computing the magnitude. They use three datasets to determine, that the maximum peak displacement four seconds after the p-Wave arrival and the epicentral distance. They discover an underestimation of magnitude values above magnitude 7, which might either be determined by their small datasets or the short timeframe where they measure the peak displacement.
In a similar way Wu et al. [2006] estimated the magnitude with a regression, using the first three seconds of the p-wave for the peak displacement and hypocentral distance as a second parameter. They propose usage of their method for events with a magnitude up to 6.5.

4 Main

4.0.1 Hypothesis

The underlying question of our project comes from an early warning background. We want to be able to detect high magnitude earthquakes really well. Naturally, strong earthquakes rarely occur and those cannot found, or just scarsely in the training data. For a classic neural network, this makes it incredibly difficult to reliably detect them. The network likely underestimates the magnitude, which is dangerous in early warning. Some more classic parametric methods are only partially affected by this. Especially simple regression lines do not exclude outlier points, which we think, can prove helpful in this scenario. Our hypothesis is, that a simple parametric method will be able to detect high magnitude earthquakes much better, than a simple neural network designed for the same task.

Our algorithm

Our project is based on research already being done in seismology. For the parametric method we use work by Kuyuk and Allen [2013]. Their regression computation uses the maximum peak displacement of the P-wave and the epicentral distance. While the peak displacement can directly be extracted from the incoming signal, we also need an estimation for the distance. For this we will use a simple neural network, which should output a distance estimation when recording the signal. To gain better insight into the distance model and to make our predictions more reliable, we introduce an uncertainty for the distance model. The network is going to predict not just a distance value, but also a range of uncertainty, where the distance value might also be located. This uncertainty can directly be transferred into the magnitude evaluation and be used, to also give an uncertainty for the final prediction.

The baseline

Our baseline model is going to get the waveform as an input and output a magnitude estimation. We want to keep the neural network as simple as possible to gain insight about the general performance of neural networks. To make magnitude estimation easier for the model, we will also feed the maximum amplitude of the original waveform as a separate input.

4.0.2 Boundary Conditions

Our aim is to create a simple early warning algorithm. This generates a few demands:

- The earthquake must be detected at first: We want to know, when it is clever to start estimating the magnitude. Today's algorithms can detect earthquake arrivals very well and it would be foolish to not use one of them. After detecting an earthquake, the algorithm can be (nearly a 100%) sure, that its input data contains not only noise, but really an earthquake. This makes predictions more reliable and limits computation time.
- The algorithms should work on a single station. Multi-station models give better results, because we can exclude faulty stations and average over station predictions. However, they come with a lot of additional architecture and computation overhead, which makes it more difficult to assure a timely estimation. Plus even multi-station models might rely on a single-station estimation.
- We want to get good predictions as fast as possible. An early warning algorithm should give an alarm before the damaging waves arrive. Therefore we deliberately learn our models on a timeframe around the start of the earthquake. While later waves include more information about magnitude and distance, they also cause more damage than the first waves. In our evaluation we will have a clear distinction between results that include those later waves, and those which do not. Hopefully this will give us a realistic impression of what our algorithm is able to achieve.

4.0.3 Program Structure

An overview of the program design can be seen in figure 4.1.

Seismogram input

We get continous signal data from our station, which can be seen in the "original seismogram". In regular intervals, this data can be evaluated by an earthquake detection algorithm. We use an adaption of the algorithm by Ross et al. [2018], which needs a four second long input. This adapted algorithm can classify the input into P-wave and noise. When detecting multiple P-waves for a few seconds of a continuous signal stream, we can step into part two of the procedure.

Detecting an earthquake

After detecting the earthquake, we try to determine the arrival time of the earthquake. We can just take the time of the very first P-wave detection. With that information we use up to four seconds after the P-wave arrival to cut out a "first four seconds seismogram". This is used to compute our peak displacement, which is just the maximum displacement of this window. If we want a prediction earlier, maybe two seconds after the beginning, we take only the two seconds we got and compute their maximum displacement. However

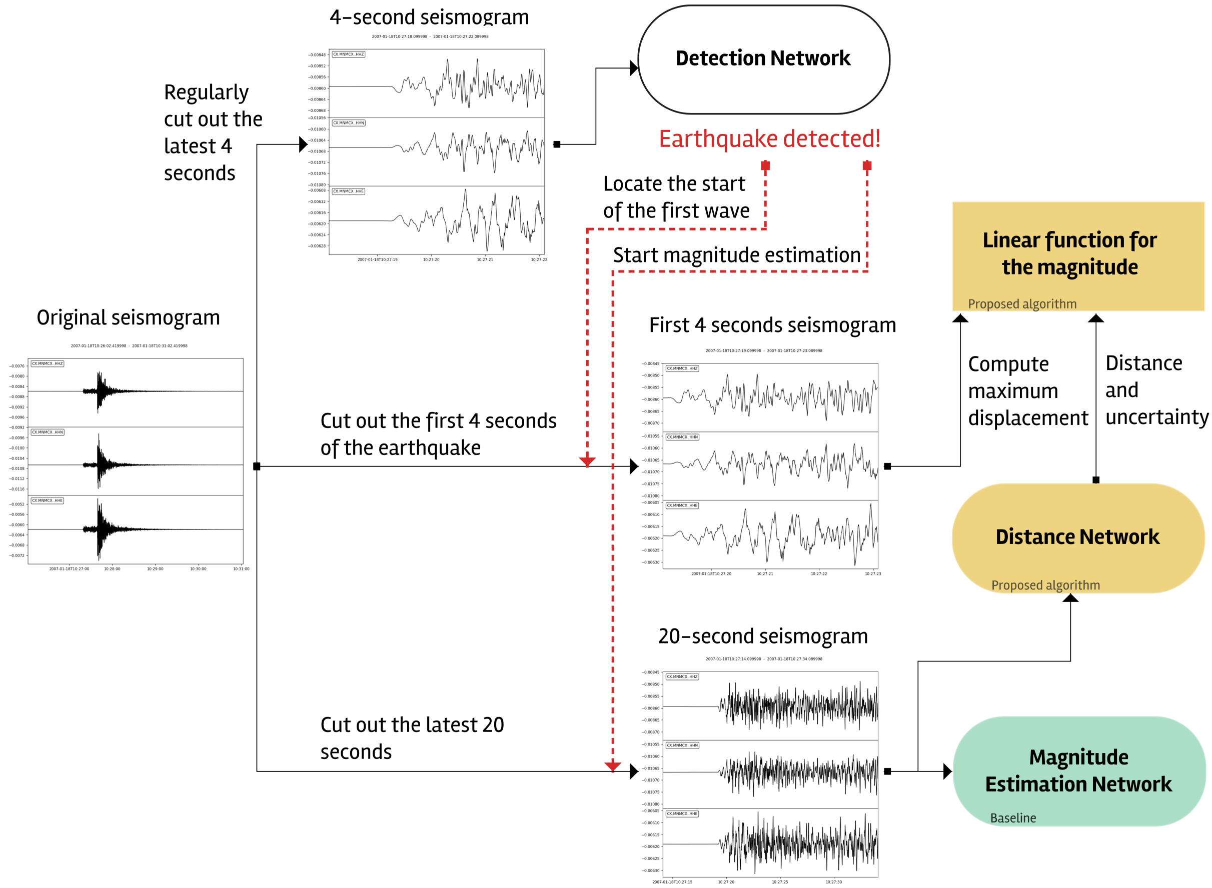


Figure 4.1: Overview of the program design. Preprocessing steps are not shown. Neural networks are shown in rounded rectangles, the proposed algorithm in orange, the baseline in green.

we will not use more than the first four seconds for peak displacement computation.
After detecting an earthquake we can directly start with estimating the magnitude. This is possible even if the 20-second window contains 19 seconds of noise and 1 second of earthquake information. In case of our baseline magnitude network, the data is directly fed into the network. It will then output a prediction.
For our algorithm we take an additional step. The 20-second window is used to generate a distance prediction, alongside an uncertainty for the generated value. Both distance and uncertainty are parameters in our "linear function for the magnitude", our adapted parametric method. Together with the value for the peak displacement our function gives us a magnitude prediction. We might use the uncertainty to get upper and lower bounds for the magnitude prediction.
While the detection is a crucial part of early warning, we will later focus on comparing the magnitude estimation routines. The question when to classify multiple detections of P-waves as an earthquake is mostly answered from a theoretical perspective and is not going to be evaluated.

4.1 Algorithm Design and requirements

Our aim is to design a framework, which is in general suitable for early warning. Therefore we decided to firstly implement an algorithm which detects an incoming earthquake, as this makes it easier to eventually use the whole project later and gives us good reason to continue computing the magnitude, if we are sure, that there is really an earthquake. As time is also an important factor, we just use data of a single station. While it is much more reliable to use a network of seismometers, we omitted this due to time and complexity overhead. Plus, the project was designed to work on small devices, which might not be able to process lots of data at once. As we wanted a reliable prediction, which would be able to include the uncertainty of the dataset, we want to at least capture the uncertainty of our deep neural network by not learning a distinctive value, but a Gauss function with an expected value and a variance, representing the uncertainty.

4.2 Data overview

The dataset consists of earthquakes detected by multiple stations in North Chile. A map, taken from Münchmeyer et al. [2020] can be seen in figure 4.2. The original catalogue consists of 101,601 earthquake events in the years 2007 to 2014. The stations are mostly part of the IPOC network (CX, GFZ German Research Centre For Geosciences & Institut Des Sciences De L'Univers-Centre National De La Recherche CNRS-INSU 2006, but a few station were added from other networks: GEOFON (GE, GEOFON Data Centre 1993, Minas (5E, Asch et al. 2011), WestFissure (8F, Wigger et al. 2016), CSN (C, C1,Universidad de Chile 2013) and Iquique (IQ, Cesca et al. 2009).

4.2.1 The Dataset

After sorting out all dataset entries, where either the waveform traces were too short or the waveform data was missing or faulty we end up with 993174 data points, each identifiable with a pair of station code and earthquake event name.´ For each data point we have got information about the epicentral distance to the concurrent earthquake event, the depth of the earthquake (measured perpendicular to the surface) and the magnitude of the earthquake. Additionally, we have a waveform file with the recorded seismograms for each earthquake, where we can find every data point's station's signal.

numbers

Train, Test and Validation Sets

We further split our dataset in train, test and validation sets. This is a (roundabout) 60:30:10 split, with a temporal ranking. Like in production training data is the oldest, with test and validation data following in time. Namely we have 595875 test, 297806 train and 99493 validation entries to work with.

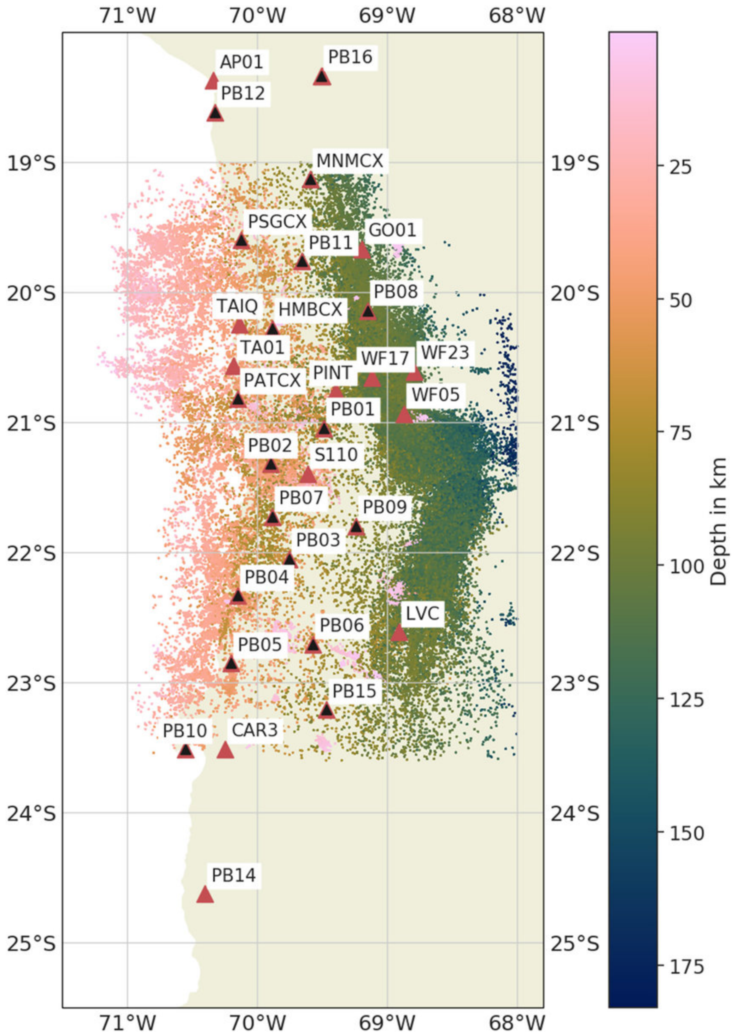
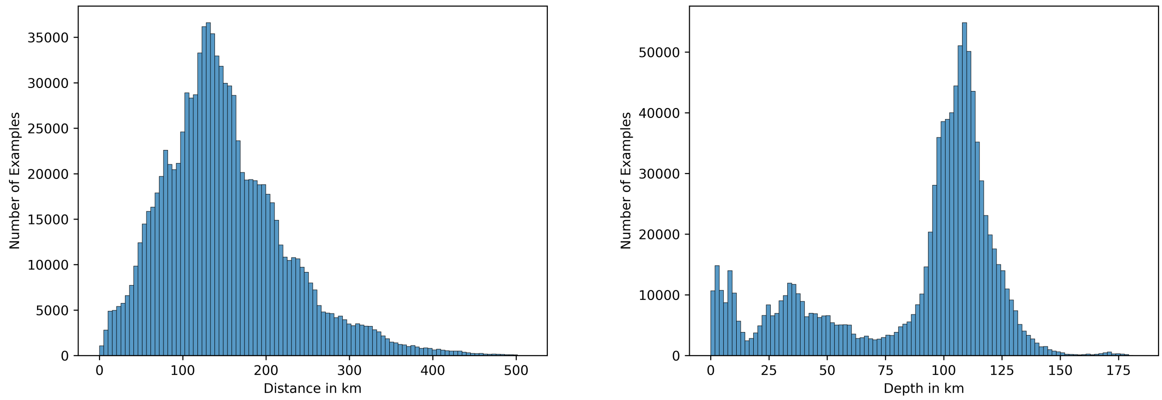


Figure 4.2: A map of all station of the Chile dataset with colour-coded depth for the earthquake events.

Parameter Distribution

Naturally the distribution of our learning parameters is not even. In figure 4.3a we see a clear peak around 150km for the epicentral distance with the most of the examples laying between 0 and 250 kms. Most of the smaller earthquakes cannot be detected far away, so the number of entries for high distances is low.
Because we measure epicentral distance, it is interesting to see, how much of this distance is accountable to the depth of an earthquake. In figure 4.3b we see that the chile dataset consists of very deep earthquakes with a lot of entries having a depth of 100 to 125 kilometers. Further description and explanation can be found with the authors of the dataset Sippl et al. [2018]. This distinctive appearance of many deep earthquakes is even more conspicous in the distance to depth scatterplot of figure 4.4.

magnitu distri- bution



(a) Distribution of epicentral distance

(b) Distribution of depth

Figure 4.3: Histogram plots of epicentral distance and depth for the Chile dataset, each with 100 bins

4.2.2 Data preparation

For the neural network

The neural networks gets two different input types: Firstly, the waveforms, as input for learning and later for predicting. Secondly the learning goal parameters, either distance or magnitude.
As a first preprocessing step the instrument sensitivity is removed from the waveform files. Depending on the station's instrument properties a constant factor is multiplied with the signal. While the shape of the signal does not change, the maximum amplitude does. We do this to make the waveform input station independent, and to later feed the maximum amplitude of the signal into the magnitude network as additional information during the learning process. Therefore we now extract the maximum amplitude of the input window.
Then we do simple detrending by removing a straight line through the first and last point of the signal from the data. Afterwards we apply a two factor high-pass filter at 2 Hz and a two-factor low-pass filter at 35 Hz. In the end the waveform data is normalized around zero into a [-1,1] range. Preprocessing is done separately for every input window, so the network cannot deduce any additional information from any preprocessing done for adjacent parts of the signal.

The learning parameters, distance and magnitude are both normalized between zero and one with a Min-Max-Scaler. As maximum and minimum values 1 and 600000 meters were chosen for the epicentral distance, with the current extrema in the whole dataset being approximately 118 meters and 512 kilometers.
Similarly the magnitude was scaled between 0 and 9, whereas the minumum and maximum values are 0.716 and 8.055 respectively.

4.2.3 For the displacement computation in the formula

water level

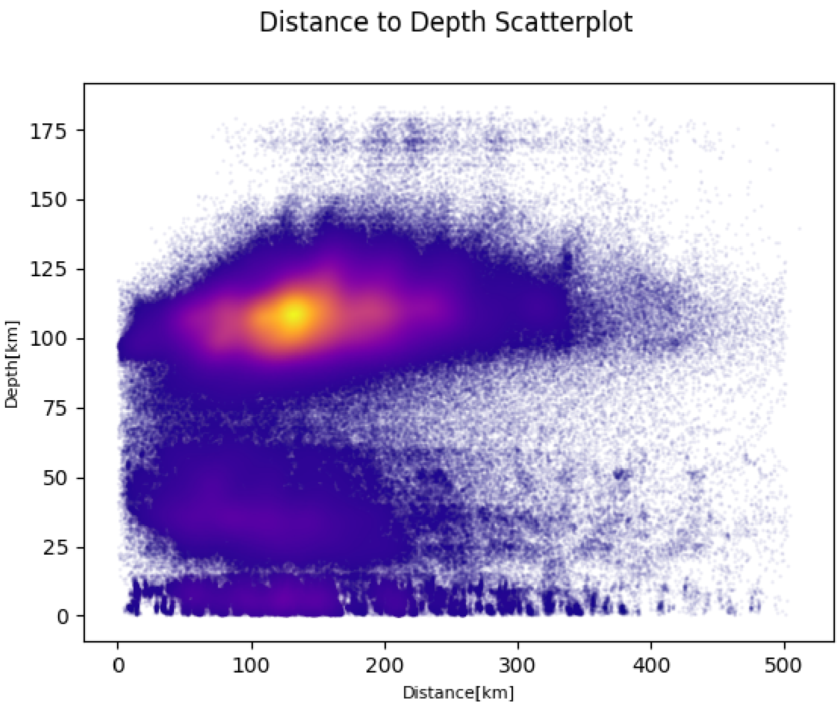


Figure 4.4: Distance and depth plotted against each other in a scatterplot. The dense areas are recognizable by a lighter and more radiant colour

We use the same waveforms to compute a value for peak displacement, which we use either for a linear regression on the train dataset or when applying the algorithm on an incoming signal. We don't remove the sensitivity, but start with detrending the data by removing a straight-line fit through the first and last point. Instead of using a generic frequency for the high-pass filter, we use an adaptive filtering technique:

Adaptive Filtering

We do not use the 2 Hz high-pass filter when computing the displacement for the parametric methods. Here we adapt the high pass filter frequency to a value from [0.001,0.1,0.3,0.5,0.75]. The method is taken from Münchmeyer et al. [2020]. As the noise level can occlude the signal for smaller earthquakes, we may want to take a high filter frequency, except for larger earthquakes, where the signal is so strong that we might loose important information by using a more influential filter.
For this they use the signal-to-noise ratio from 30 seconds before and after the P-pick. The lowest filter frequency while retaining a signal-to-noise ratio above 4 is chosen. Figure **??**hows the filter frequencies for the different magnitudes computed by Münchmeyer et al. [2020] and is also taken from their publication.
It is important to mention that we use this method not only for computing the regression line on the dataset but also when estimating the magnitude from an incoming signal with the parametric method. This is possible because we pre-computed the frequencies also on our test dataset, but some differences would have to be made when using our method for unknown data. The signal-to-noise ratio could only be computed up to the latest incoming signal data, which would result in a more inaccurate filter frequency choice, likely

underestimating the ratio at first. The frequency could be adapted every time we rerecord the arriving signal, making it more reliable over time.

Afterwards we apply a low-pass filter at 35 Hz. We then remove the sensitivity, but apply a water level to prevent enhancing noise. Station instruments tend to be less sensitive for very high or very low frequencies. Removing the sensitivity is equivalent to dividing the signal input through a sensitivity value. If this value is very small, this division can result in a very high output, just because the instrument was not able to detect well. This can be hindered by clipping the sensitivity value before it gets too small. This is called the water level. It essentially smoothes the signal and prevents enhancing noise. Additionally we have to remove the instrument response, which is another unique factor As an additional measure we apply a water level when removing the instrument response from the signal directly before extracting the displacement all frequencies above 30Hz are deleted from the signal. This removes jittering in the data and smoothes the input.

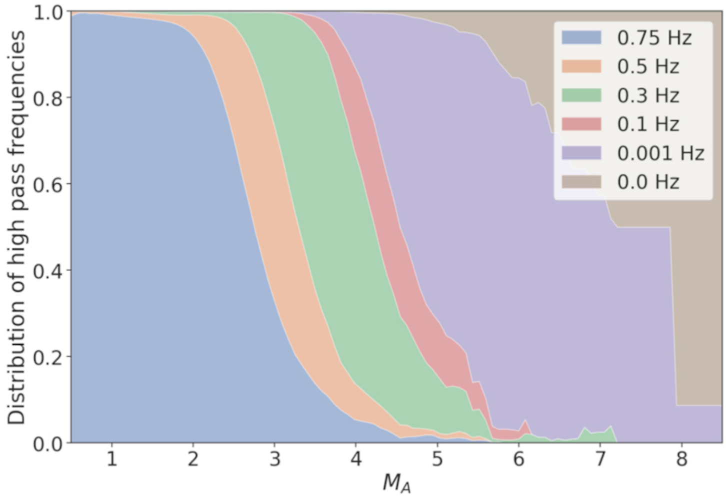


Figure 4.5: High-pass filter frequencies depending on magnitude, computed and plotted by Münchmeyer et al. [2020]

4.3 Detecting the earthquake

4.4 Ground-truth algorithm

The algorithm which we will use to compare our new technique to, is a simple CNN network, similar to the network we will use in our algorithm. It directly gives us a value for the magnitude from a 20 second seismometer input. Our proposed algorithm will be evaluated against a basic algorithm on the same dataset.

4.5 Proposed algorithm

The proposed algorithm consists of two parts: At first we take the whole 20 second input and compute a distance to the earthquake. Then we estimate the magnitude by using the formula proposed in cite

4.6 Bringing it all together

´

5 Evaluation

In evaluation it is important to not take into account any input, where the network already sees the s-wave. While we have included these examples into the learning, we now do not want to include those when evaluating the performance.

6 Conclusion

6.1 Summary

Sum up your work. Similar to the abstract but more technical.

6.2 Evaluation

Be hard with yourself, but not too hard. Stay scientific!

6.3 Future Work

What would be next? What did your thesis not touch? Münchmeyer et al. [2020]

Appendices

A. A section

Weird stuff that you didn't want to put into the main text but didn't want to leave it out either? You found the right place for it.

Bibliography

G. Asch, F. Tilmann, B. Schurr, and T. Ryberg. Seismic network 5e: Minas project (2011/2013), 2011.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

S. Cesca, M. Sobiesiak, A. Tassara, M. Olcay, E. Günther, S. Mikulla, and T. Dahm. The iquique local network and picarray, 2009.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

GEOFON Data Center. Geofon seismic network, 1993.

GFZ German Research Centre For Geosciences and Institut Des Sciences De L'Univers-Centre National De La Recherche CNRS-INSU. Ipoc seismic network, 2006.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

J. Havskov and G. Alguacil. *Seismic Networks*, pages 261–308. Springer International Publishing, Cham, 2016. ISBN 978-3-319-21314-9. doi: 10.1007/978-3-319-21314-9_8. URL https://doi.org/10.1007/978-3-319-21314-9_8.

H. S. Kuyuk and R. M. Allen. A global approach to provide magnitude estimates for earthquake early warning alerts. *Geophysical Research Letters*, 40(24):6329–6333, 2013.

Z. Li, M.-A. Meier, E. Hauksson, Z. Zhan, and J. Andrews. Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophysical Research Letters*, 45(10):4773–4779, 2018.

A. Lomax, A. Michelini, and D. Jozinović. An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. *Seismological Research Letters*, 90(2A):517–529, 2019.

S. M. Mousavi and G. C. Beroza. Bayesian-deep-learning estimation of earthquake location from single-station observations. *arXiv preprint arXiv:1912.01144*, 2019.

S. M. Mousavi and G. C. Beroza. A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1):e2019GL085976, 2020.

J. Münchmeyer, D. Bindi, C. Sippl, U. Leser, and F. Tilmann. Low uncertainty multifeature magnitude estimation with 3-D corrections and boosting tree regression: Application to North Chile. *Geophysical Journal International*, 220(1):142–159, 2020. ISSN 1365246X. doi: 10.1093/gji/ggz416.

J. Münchmeyer, D. Bindi, C. Sippl, U. Leser, and F. Tilmann. Low uncertainty multifeature magnitude estimation with 3-d corrections and boosting tree regression: application to north chile. *Geophysical Journal International*, 220(1):142–159, 2020.

J. Münchmeyer, D. Bindi, U. Leser, and F. Tilmann. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophysical Journal International*, 226(2):1086–1104, 2021.

T. Perol, M. Gharbi, and M. Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2):e1700578, 2018.

N.-C. Ristea and A. Radoi. Complex neural networks for estimating epicentral distance, depth, and magnitude of seismic waves. *IEEE Geoscience and Remote Sensing Letters*, 2021.

Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton. Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, 108 (5A):2894–2901, 2018.

C. Sippl, B. Schurr, G. Asch, and J. Kummerow. Seismicity structure of the northern chile forearc from> 100,000 double-difference relocated hypocenters. *Journal of Geophysical Research: Solid Earth*, 123(5):4063–4087, 2018.

S. Stein and M. Wysession. *Introduction to Seismology, Earthquakes, and Earth Structure*. Blackwell Publishing, 2003.

Universidad de Chile. Red sismologica nacional, 2013.

P. Wigger, P. Salazar, J. Kummerow, W. Bloch, G. Asch, and S. Shapiro. West–fissure- and atacama-fault seismic network (2005/2012), 2016.

Y.-M. Wu, H.-Y. Yen, L. Zhao, B.-S. Huang, and W.-T. Liang. Magnitude determination using initial p waves: A single-station approach. *Geophysical Research Letters*, 33(5), 2006.

X. Zhang, M. Zhang, and X. Tian. Real-time earthquake early warning with deep learning: Application to the 2016 m 6.0 central apennines, italy earthquake. *Geophysical Research Letters*, 48(5):2020GL089394, 2021.

W. Zhu and G. C. Beroza. Phasenet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019.

´

Statement of Authorship / Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbstständig und ausschließlich unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.
Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder einer anderen Prüfungsbehörde vorgelegt oder noch anderweitig veröffentlicht.

Unterschrift	Datum