IDEA League

Master of Science in Applied Geophysics

Research Thesis

Extracting geomechanical information from borehole microseismic data with machine learning

Maria Eugenia Carrizo Mascarell

August 7, 2020

Extracting geomechanical information from borehole microseismic data with machine learning

Master of Science Thesis

for the degree of Master of Science in Applied Geophysics
by

Maria Eugenia Carrizo Mascarell

August 7, 2020

Department of Geoscience & Engineering — Delft University of Technology
Department of Earth Sciences — ETH Zurich
Faculty of Georesources and Material Engineering — RWTH Aachen University

Copyright © 2020 by IDEA League Joint Master's in Applied Geophysics:

Delft University of Technology, Swiss Federal Institute of Technology in Zurich, RWTH Aachen University.

All rights reserved.

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or by any information storage and retrieval system, without permission from this publisher.

Printed in The Netherlands, Switzerland, Germany

August 7, 2020

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 7, 2020*

Supervisor(s):

Dr. Ir. G. G. Drijkoningen
Dr. Ir. Diego Rovetta

Committee Members:

Dr. Ir. G. G. Drijkoningen

Dr. Ir. Diego Rovetta

Dr. Ir. Florian Wellman

Dr. Ir. Deyan Draganov

Abstract

A key practice in the development of unconventional hydrocarbon resources is to monitor hydraulic fracturing operations and analyze in detail the induced microseismicity, for understanding the extent of the volume affected by the fractures. Even though microseismic data are one of the few geophysical measurements that can be used for this purpose they are not always used at their full potential: the common analyses are limited to the interpretation of the locations of the induced seismic events in terms of fractured volume, while the propagation and the dynamic evolution of the microseismic events that are directly related to the source process are not studied. The source moment tensor can be used to describe the deformation mechanism occurring at the seismic source and the corresponding geomechanical parameters, which are relevant aspects for the monitoring of hydraulic fracturing operations. Its value can be estimated through an inversion process using the P-wave and S-wave waveforms from the observations at the seismic receivers. However, the limited aperture of the receiver arrays in borehole acquisition is inadequate to solve the inverse problem.

The goal of this thesis project is study the feasibility of estimating the source moment tensor with borehole microseismic data making use of machine learning algorithms. The task is done by implementing a forward model to compute waveforms generated by known values of the source moment tensor in a borehole acquisition geometry (forward problem). Forward modeling was carried out using a modified version of the discrete wavenumber method. Afterwards, this data is used to train a neural network to estimate the moment tensor at the source given the microseismic data (inverse problem). The neural network was designed, adapting an existing architecture used in previous studies to solve similar inverse problems. The implementation was done entirely in the Python programming language, using the Tensorflow library for the machine learning parts. The network was trained with the dataset generated in the forward model until reaching an optimal mean squared error minimum. Finally, the capability of the network to invert microseismic measurements and retrieve acceptable values of the source moment-tensor components was tested using synthetic data, resulting in an average prediction accuracy of 0.997. Our results indicate that for this case, a neural network is able to satisfactory invert for the full moment-tensor components. Additionally, the neural network was also trained to handle the presence of Gaussian noise, achieving an average prediction accuracy of 0.990 for tested synthetic data with 10% Gaussian noise.

Acknowledgements

Para mi familia

I would like to express my deepest gratitude to all the people who have contributed to my thesis.

Foremost, I would like to express my sincere gratitude to my advisor Diego Rovetta for his continuous support, patience, encouragement and time devoted to the success of this project and to Guy Drijkoningen for his insights, suggestions and dedication.

Moreover, I take this opportunity to gratefully acknowledge the Society of Exploration Geophysicist for supporting me in pursuing this degree.

I am also very grateful for being a part of the Aramco Overseas Company during this thesis project. A special thanks to the Delft R&T team: Christian Reinicke, Marcin Dukalski, Paul Zwartjes, Yimin Sun, Hannes Kutscha, Rolf Baardman, Rob Hegge, Apostolos Kontakis, Yoo Jewoo, Mikhail Belonosov and Roald van Borselen, for their continuous feedback, inspiring advise and great conversations. Finally, I would like to express my heartfelt thanks to Roald van Borselen and Diego Rovetta for allowing me the opportunity to conduct this research study together with AOC.

Delft University of Technology
Swiss Federal Institute of Technology in Zurich
RWTH Aachen University

Maria E. Carrizo Mascarell
August 7th, 2020

Table of Contents

1 Introduction **1**

1-1 Microseismic monitoring of unconventional reservoirs . . . 1

1-2 Seismic moment-tensor inversion . . . 2

1-2-1 Deterministic inversion . . . 2

1-2-2 Stochastic inversion . . . 3

1-2-3 Prediction through Artificial Neural Networks . . . 4

1-3 Thesis objectives . . . 4

2 Seismic moment tensor **7**

2-1 Derivation of the moment tensor . . . 7

2-2 Decomposition of the moment tensor . . . 10

2-3 Deterministic inversion . . . 13

Frequency-domain moment-tensor inversion . . . 14

Time-domain moment-tensor inversion . . . 14

3 Forward modeling problem **17**

3-1 Discrete WaveNumber method . . . 17

3-2 Modified Discrete WaveNumber method for full-moment sources . . . 19

3-3 Forward modeling algorithm . . . 21

3-4 Benchmarking . . . 23

3-5 Generation of a realistic synthetic dataset . . . 28

4 Inverse problem **31**

4-1 ANN architecture design . . . 31

4-2 ANN training and validation . . . 33

4-2-1 Standardization of the data . . . 34

4-2-2 Loss function . . . 34

4-2-3 Optimizer . . . 35

4-2-4 Batch size . . . 37

4-2-5 Training data size . . . 39

4-2-6 Training summary . . . 39

4-3 ANN Testing . . . 41

4-3-1 ANN trained without noise . . . 41

4-3-2 ANN trained with noisy data . . . 42

4-3-3 Additional tests on training data size . . . 51

4-4 Receiver geometry effects on the ANN performance . . . 51

5 Conclusions and Discussion **59**

5-1 Future developments . . . 60

A Artificial Neural Networks **61**

B Calculation of seismograms using a modified DWN method **63**

B-1 Calculation of potentials at the source layer . . . 63

B-2 Propagation of potentials . . . 65

B-3 Reflectivity method . . . 65

B-4 Calculation of the spatial derivatives of the Green's functions . . . 66

C Forward modeling code **67**

Bibliography **69**

Chapter 1

Introduction

1-1 Microseismic monitoring of unconventional reservoirs

Unconventional reservoirs in the present time contribute to an important amount of the hydrocarbon production worldwide. The U.S. Energy Information Administration (EIA) estimates that in 2019, about 2.81 billion barrels (or 7.7 million barrels per day) of crude oil were produced directly from tight oil resources in the United States. This represents about 63% of total U.S. crude oil production in 2019. Hydrocarbon production in the U.S. has increased significantly during the past 10 years, driven mainly by production from tight rock formations developed using horizontal drilling and hydraulic fracturing to extract hydrocarbons (E.I.A. (2020)).

Unconventional resources reside in tight formations, where the low reservoir quality obstructs hydrocarbon extraction (Ma and Holditch (2016)). Therefore, most unconventional reservoirs must be stimulated by hydraulic injection, creating fractures that serve as permeable channels to move the fluids out of the reservoir. Hydraulic fracturing operations control the production of the reservoir and a clear understanding of the fractures generated and their effect on the reservoir is necessary (Warpinski (2009)).

Microseismic monitoring is an exceptional technique used to evaluate hydrofracturing operations in a reservoir (Ma and Holditch (2016); Baig and Urbancic (2010); Warpinski (2009)). This tool is mostly used in hydrofracture mapping, but it has been applied recently in monitoring CO2 injections (Verdon (2011); Verdon et al. (2011)), geothermal reservoir stimulation (Häring et al. (2008)), reservoir surveillance (McGillivray (2005); Maxwell et al. (2010); Clarkson and Beierle (2011)) and mining (Urbancic and Trifu (2000)).

Microseismic events are very small-scale earthquakes (magnitude less than zero in a logarithmic scale (Young and Baker (2001))), which can be generated as a result of human activities or industrial processes such as mining, hydraulic fracturing, enhanced oil recovery, geothermal operations or underground gas storage (van der Baan et al. (2013)). The magnitudes of these events are usually not properly detected by surface geophones, so it is common to place the

geophones in adjacent boreholes (Verdon (2011)). The goal of microseismic monitoring is to detect, locate and characterize microseismic events that illuminate the fracture networks.

Rapid turnaround in microseismic monitoring of hydraulic fracturing operations is a high priority in the industry. Reducing microseismic processing and delivering analysis results in near realtime is very important for decision making during completion operations. Therefore, fast and robust techniques are necessary for rapid microseismic data evaluation (Verdon (2011); van der Baan et al. (2013)).

The characterization of microseismic events provides information about the source deformation mechanisms. The concept of the seismic moment tensor can be used to describe microseismic point sources in general terms as a set of force couples (van der Baan et al. (2013)). Thus, the moment tensor can be used as a mathematical representation of a certain microseismic event.

1-2 Seismic moment-tensor inversion

The goal of seismic moment-tensor inversion is to identify the source mechanisms of the seismic event from measurements. Inversion techniques to estimate moment tensors from seismic recordings are well developed in earthquake seismology, and have been used in microseismic monitoring applications since the last decade (Baig and Urbancic (2010)).

The moment-tensor inversion is used to retrieve the estimation of the moment-tensor components and the fault plane solution. Understanding the source mechanisms of microseismic observations can lead to a reduction in the uncertainty of fracture type, propagation, and connectivity of a hydraulic fracture operation (Eyre and Baan (2017)).

There are numerous methods to perform the moment-tensor inversion. The inversion can be done in the time or frequency domain (Jost and Herrmann (1989)) and different types of measurements can be used as data (Pugh et al. (2016)). General methods to perform moment-tensor inversion are described in the following sections.

1-2-1 Deterministic inversion

A conventional approach in geophysical data analysis is to construct different geological models and compare synthetic geophysical data computed for these models with the observed data (Zhdanov (2015)). Therefore, we define a theoretical relationship between the geophysical data and a geological physical system (Tarantola and Valette (1982)). Modelling geophysical data for a certain model is called the forward problem.

The purpose of acquiring geophysical measurements is to determine geological information from the data. This is not an easy task considering the complexity of the Earth's interior. Therefore, it is common to approximate real geology to a more simple model and try to determine the model parameters from the geophysical data. This is called the inverse problem.

For a forward problem the values of the model parameters are assumed known and the values of some observable quantities (data) are unknowns. For an inverse problem the data resulted of some measurements are known, and the unknowns are the values of the model parameters.

To solve an inverse problem it is important to determine if the mathematical setting of a problem is well-posed or ill-posed (Denisov (1999)). Well-posedness of a mathematical problem was first formulated by Hadamard (1902), where he stated that a problem was well-posed if:

- The solution of the problem exists.
- The solution of the problem is unique.
- The solution is stable.

If at least of one the conditions fails the problem is considered ill-posed. Most of the problems in geophysics are ill-posed but they are mathematically and physically meaningful, and can be solved (Zhdanov (2015)).

The inverse problem in geophysical cases are generally non-linear (Denisov (1999)), meaning that the function linking the model parameters to the data is non-linear. To solve this, in practice, the inverse problem is often linearised and the final non-linear solution is obtained through the iterative application of linearised solvers.

The linearisation requires an initial model to be applied. The variation of the data with respect to the synthetic measurements calculated from the initial model, can be expressed by the multiplication of the variation of the estimated model from the initial model, with a Jacobian matrix, containing the first-order partial derivatives of the function linking the model to the data. A linearised inversion can be therefore obtained by inverting the Jacobian matrix and applying it to the variation of the data to obtain the variation of the model.

In our specific problem the model parameters are the moment tensor components and the data are the seismic observations, while the Jacobian matrix corresponds to the derivatives of the Green's functions, which is the operator linking the model to the data (Mustać and Tkalčić (2015), Vackář et al. (2017)).

More details about the deterministic inversion approach will be given in the next chapter.

1-2-2 Stochastic inversion

The solution of an inverse problem aims to find the best fitting model parameters explaining the observed data. In the case of the stochastic inversion, the probability of obtaining a certain set of measurements can be evaluated for possible values of the unknown model parameters. The result of this analysis is a probability density distribution function (PDF) that describes the likelihood of observing the data for a particular value of the model (Tarantola (2004)).

In a Bayesian approach the model parameters are treated as random variables, meaning that we can sample an ensemble of models instead of looking for the one that provides the best-fitting solution (Mustać and Tkalčić (2015)). In this framework the inversion method aims to obtain the probability of the model given a set of observations, called posterior PDF, which can be evaluated using the Bayes' Theorem (Bayes and Price (1763)).

In the moment tensor inversion case, the posterior PDF of an ensemble of moment tensors, given a set observed seismograms can be calculated as a multiplication of the likelihood of the seismograms for a particular moment tensor with a prior moment tensor PDF.

Many previous studies successfully use stochastic inversion to infer the moment tensor from seismic data, such as: Pugh et al. (2016), Mustać and Tkalčić (2015), Vackář et al. (2017) and Käufl et al. (2015).

1-2-3 Prediction through Artificial Neural Networks

A brief introduction on artificial neural networks (ANNs) is given in appendix A.

ANNs can be trained using a set of synthetic microseismic events with known moment tensors in a supervised learning manner (Binder (2018)). The primary advantage of the neural network approach for moment tensor inversion is that expensive forward modelling operations are only done at the beginning to generate the training dataset. Subsequently, the trained neural network can predict moment tensors on new events, making this approach attractive for evaluating large datasets of microseismic events or in real-time applications. It is also common to include prior information on model uncertainties, for example in the velocity model or source mechanisms, by sampling from the prior distribution when generating the training dataset.

Previous evaluations using artificial neural networks in moment tensor inversion indicate that main factors that can affect the network performance is the level of the noise in the data and the accuracy of the velocity model (Binder (2018), Maity et al. (2014), Ovcharenko et al. (2018)). However, many authors claim that ANNs are adaptive and noise tolerant when properly trained with noisy data and therefore can generate effective models which can provide robust results with respect to outliers and observational noise (Maity et al. (2014)).

1-3 Thesis objectives

The methods discussed previously can all been used to perform moment tensor inversion. These approaches have advantages and disadvantages that must be considered when choosing the most appropriate for a specific solution.

In the case of estimating the moment tensor from borehole microseismic measurements there are several factors to consider:

1. Borehole microseismic measurements are usually acquired using small-aperture geometries, making the inversion problem ill-posed (Grechka et al. (2016), Song and Toksoez (2011)). Vavrycuk (2007) states that for single-well measurements, only up to 5 of all 6 moment tensor components could be resolved.

2. In the deterministic and stochastic inversion approach, the Green's functions for each moment tensor component must be evaluated (Binder (2018)). Evaluation of the Green's functions can be computationally expensive for inhomogeneous velocity models and may require a 3D finite difference solution of the wave equation. This results in a challenge for real-time moment tensor inversion.

3. Uncertainty regarding the velocity model affects largely the results of deterministic moment tensor inversion. In the case of the stochastic moment tensor inversion the

uncertainty can be included in the prior model information (Pugh et al. (2016), Maity et al. (2014), Song and Toksoez (2011)).

4. According to several authors (Binder (2018), Pugh et al. (2016) and Ovcharenko et al. (2018)), noise can have a major negative influence on the accuracy of the moment tensor inversion on both deterministic and stochastic inversion.

Using ANNs in moment-tensor inversion provides advantages when considering these four factors. Data-driven machine learning techniques, such as ANNs, are able to generate effective models for solving complex and ill-posed problems (Maity et al. (2014), Binder (2018)). Additionally, ANNs do not require the estimation of the Green's functions and/or the velocity model during the prediction phase, making them practically inexpensive and ideal for real-time applications, even with large datasets. The bulk of the computational efforts is only carried out once, during the training phase of the network. Finally, it has been stated by numerous authors that ANNs can yield acceptable results even with data contaminated with high levels of noise when properly trained (Mira and Sanchez-Andres (1999)).

Ovcharenko et al. (2018) studied the use of ANNs while performing moment-tensor inversion for a vertical borehole array of receivers and using peak amplitudes and polarization of P and S wave measurements in a homogeneous velocity model. In this study the authors found a neural network architecture successfully predicting the moment tensor components from noisy data.

In this thesis we extend Ovcharenko's study by evaluating a case of combining vertical and horizontal arrays of borehole receivers and by considering full wavefield measurements under the assumption of 1D layered velocity model. The main objective is to use an ANN to predict the moment tensor associated to a microseismic event from the seismic data acquired through an array of receivers in the borehole (inverse problem). The ANN is trained with synthetic data generated by modeling (forward problem). The inversion technique is tested on synthetic data corrupted by Gaussian random noise.

Chapter 2 will introduce the concept of the moment tensor, its decomposition as well as commonly used inversion methods. The forward problem and the generation of the synthetic dataset are described in Chapter 3, while the training and prediction of the ANN, and the analysis of the inversion results is reported in Chapter 4. In Chapter 5 some conclusions and possible future developments are discussed.

Chapter 2

Seismic moment tensor

The application of physical forces at the location of a seismic event, due to the movement of a fault, volume collapse or increase, produces rotational effects on the ground that can be mathematically described by the seismic moment tensor. This tensor can be seen as a first order approximation of the rotational effects induced by the equivalent forces, representing the physical seismic forces (Jost and Herrmann (1989)). The concept of moment tensor was introduced by Gilbert (1971) for calculating displacements at the surface as the sum of the products of the moment-tensor elements with the corresponding Green's functions (Aki and Richards (2002)).

2-1 Derivation of the moment tensor

Let's consider a local geographic coordinate system, centered at a seismic event location, with positive X-axis being north, positive Y-axis eastward, and positive Z-axis downward (NED). The NED coordinate system can be used to derive relations between the angles of a ruptured fault (strike angle, ϕ, dip angle, φ, and rake angle, ψ) and the dislocation directions on the rupture plane (normal $\mathbf{n}$ and slip $\mathbf{s}$), as shown in Figure 2-1.

Using Aki and Richards (2002) representation theorem for the seismic source we can define the ith component ($i = 1, 2, 3$, corresponding to P, SH and SV propagation modes, respectively) of the displacement $u_i = u_i(\mathbf{x}, t)$ at the position $\mathbf{x}$ (of coordinates x, y and z) and at the time t as the superposition of the equivalent body force densities f_k multiplied by the Green's functions G_{ik}, in the k direction ($k = 1, 2, 3$, corresponding to x, y and z, respectively):

$$u_i(\mathbf{x}, t) = \int_{-\infty}^{\infty} \int_V f_k(\xi, \tau) G_{ik}(\mathbf{x}, t; \xi, \tau) d\tau dV(\xi) \tag{2-1}$$

where the Green's tensor $G_{ik} = G_{ik}(\mathbf{x}, t; \xi, \tau)$ describes the propagation effects of the waves along a path from the source to a receiver, between the source position $\xi = (\xi_x, \xi_y, \xi_z)$ at the time τ and the position $\mathbf{x}$ at the time t . The vector $f_k = f_k(\xi, \tau)$ represents the density of

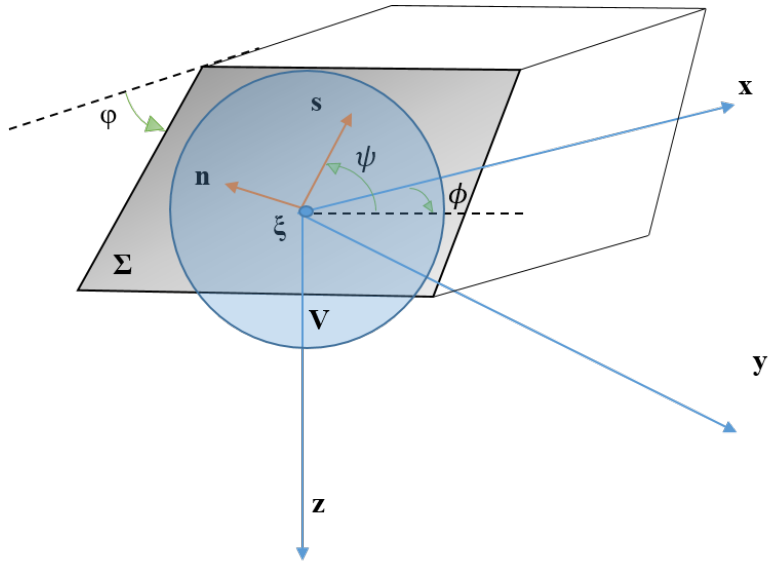

Figure 2-1: Coordinate system centered at the microseismic event location ξ (modified after Aki and Richards (2002)). Strike ϕ is measured clockwise from north, dip φ from horizontal down, and rake ψ counterclockwise from horizontal. Vectors $\mathbf{s}$ and $\mathbf{n}$ are the slip and fault normal directions, respectively

the body forces acting at the source. The integration is performed over focal volume V and time t. Equation (2-1) defines the solution of the motion for a single point force.

Actually, the body forces related to a seismic source are not distributed in a volume but along a fault (Vavryčuk (2015)). Moreover, the seismic source is not described by single forces but by dipole forces. The dipole forces cause the two blocks at opposite sides of the fault plane to mutually move.

The moment tensor density components $m_{kl} = m_{kl}(\xi, \tau)$ represent the dipole forces that define the motion at opposite sides of a fault along its plane Σ; the subscript $l = 1, 2, 3$, corresponds to the coordinate axis x, y and z. From Aki and Richards (2002) we can define the moment tensor density components m_{kl} from the fault vector components as expressed by:

$$m_{kl} = c_{klpq} s_p n_q \tag{2-2}$$

where the vector $\mathbf{s} = (s_x, s_y, s_z)$ represents the unit vector in the slip direction and $\mathbf{n} = (n_x, n_y, n_z)$ the unit vector normal to the fault direction, the tensor c_{klpq} corresponds to the stiffness tensor containing the elastic parameters of the medium surrounding the fault, and the subscripts $p = 1, 2, 3$ and $q = 1, 2, 3$ also correspond to the coordinate axis x, y and z.

Assuming an isotropic body, Jeffreys and Jeffreys (1972) define the stiffness tensor as:

$$c_{klpq} = \lambda \delta_{kl} \delta_{pq} + \mu(\delta_{kp}\delta_{lq} + \delta_{lp}\delta_{kq}) \tag{2-3}$$

where δ_{ij} represents the Kronecker delta:

$$\delta_{ij} = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases} \tag{2-4}$$

Combining (2-2) and (2-3) we can express the moment density component for a force couple in an isotropic body, according to Aki and Richards (2002) as:

$$m_{kl} = \lambda n_p s_p \delta_{kl} + \mu(n_k s_l + s_k n_l) \tag{2-5}$$

where λ and μ are the Lame's constants of the medium. Substituting in (2-1) the single force f_k with the moment tensor density components m_{kl} of a fault displacement (defined in 2-5) we obtain (Aki and Richards (2002)):

$$u_i(\mathbf{x}, t) = \int_{-\infty}^{\infty} \int_{\Sigma} m_{kl}(\xi, \tau) \frac{\partial}{\partial \xi_l} G_{ik}(\mathbf{x}, t; \xi, \tau) d\tau d\Sigma(\xi) \tag{2-6}$$

where ξ_l represents the displacement in the l direction. Furthermore, if the size of the fault plane is small compared to the distance between source and receiver, (2-6) can be simplified into:

$$u_i(\mathbf{x}, t) = \int_{-\infty}^{\infty} M_{kl}(\tau) \frac{\partial}{\partial \xi_l} G_{ik}(\mathbf{x}, t; \xi, \tau) d\tau = M_{kl}(t) * G_{ikl}(\mathbf{x}, t) \tag{2-7}$$

where the symbol "$*$" represents the convolution operation, G_{ikl} is the tensor of spatial derivatives of the Green's functions along the lth component for the ith displacement produced by a force couple M_{kl} defined by:

$$M_{kl} = \int_{\Sigma} m_{kl}(\xi, \tau) d\Sigma(\xi) \tag{2-8}$$

The moment tensor $\mathbf{M}$ can be represented by a 3×3 matrix whose elements describe the nine couples of equivalent dipole forces required to define a point source (Aki and Richards (2002)):

$$\mathbf{M} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{xy} & M_{yy} & M_{yz} \\ M_{xz} & M_{yz} & M_{zz} \end{bmatrix} \tag{2-9}$$

These nine generalized couples refer to pairs of vectors that act on points in space (see Figure 2-2). The diagonal elements represent linear vector dipoles, the off diagonal elements represent the force couples with moment (Jost and Herrmann (1989)). The moment tensor is symmetric due to the conservation of angular momentum: with no external force to act upon the system at a point in time, the original angular momentum of the system is conserved (Vavryčuk (2015)). Therefore, $\mathbf{M}$ is losing rank (rank = 2) and has only six independent components.

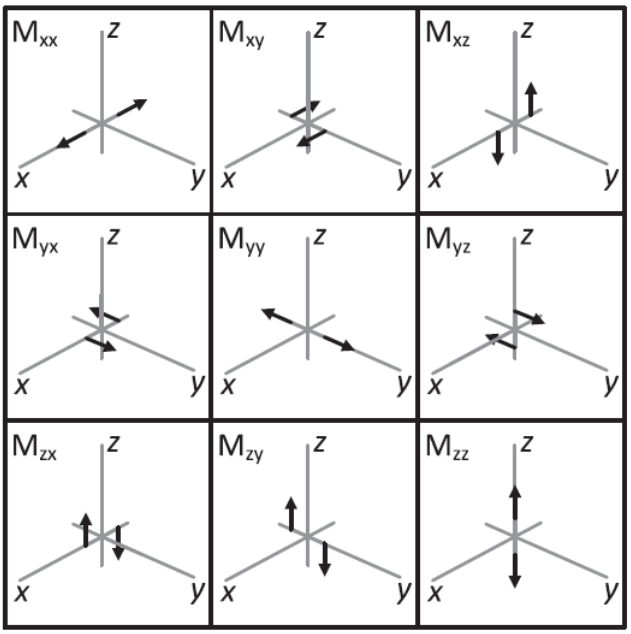

Figure 2-2: Force couples description of a moment tensor following Aki and Richards (2002).

2-2 Decomposition of the moment tensor

The equivalent forces that describe the source mechanism can be determined from an analysis of the eigenvalues and eigenvectors of the moment tensor. The method to decompose the moment tensor is based on the one proposed by Jost and Herrmann (1989).

The moment tensor can be represented as the sum of an isotropic part (the identity matrix multiplied for a scalar value) and a deviatoric part. In order to derive a general formulation of the moment-tensor decomposition, let's consider its eigenvalues and orthonormal eigenvectors.

From Jost and Herrmann (1989), the moment tensor is first decomposed in the principal axes system as:

$$\mathbf{M} = \begin{bmatrix} a_{1x} & a_{2x} & a_{3x} \\ a_{1y} & a_{2y} & a_{3y} \\ a_{1z} & a_{2z} & a_{3z} \end{bmatrix} \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} a_{1x} & a_{1y} & a_{1z} \\ a_{2x} & a_{2y} & a_{2z} \\ a_{3x} & a_{3y} & a_{3z} \end{bmatrix} \tag{2-10}$$

where $\mathbf{a_i}$ represent the eigenvector associated with the eigenvalue m_i. The eigenvectors $\mathbf{a_1} = (a_{1x}, a_{1y}, a_{1z})$, $\mathbf{a_2} = (a_{2x}, a_{2y}, a_{2z})$ and $\mathbf{a_3} = (a_{3x}, a_{3y}, a_{3z})$ define the tension ($\mathbf{T}$), intermediate or neutral ($\mathbf{N}$), and pressure ($\mathbf{P}$) axes, respectively.

The result of the decomposition in (2-10) is that a new orthogonal coordinate system, given by the eigenvectors, has been defined. In this new coordinate system, the source is described by a linear combination of the orthogonal dipole sources, where $\mathbf{M_d}$, the diagonalized moment tensor is:

$$\mathbf{M_d} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \tag{2-11}$$

The diagonalized moment tensor $\mathbf{M_d}$ can also be expressed as:

$$\mathbf{M_d} = \frac{1}{3}\begin{bmatrix} tr(\mathbf{M}) & 0 & 0 \\ 0 & tr(\mathbf{M}) & 0 \\ 0 & 0 & tr(\mathbf{M}) \end{bmatrix} + \begin{bmatrix} m_1^* & 0 & 0 \\ 0 & m_2^* & 0 \\ 0 & 0 & m_3^* \end{bmatrix} \tag{2-12}$$

where $tr(\mathbf{M})$ is the trace of the moment tensor; the first term on the right hand side describes the isotropic part, the second term contains the purely deviatoric eigenvalues m_i^*, defined as:

$$m_i^* = m_i - \frac{1}{3}tr(\mathbf{M}) \tag{2-13}$$

Using (2-12) the moment tensor $\mathbf{M}$ can now be expressed using the eigenvectors and purely deviatoric eigenvalues as:

$$\mathbf{M} = \frac{1}{3}tr(\mathbf{M})\mathbf{I} + (m_1^*\mathbf{a_1}\mathbf{a_1}^\mathsf{T} + m_2^*\mathbf{a_2}\mathbf{a_2}^\mathsf{T} + m_3^*\mathbf{a_3}\mathbf{a_3}^\mathsf{T}) = \mathbf{M}_{ISO} + \mathbf{M}_{DEV} \tag{2-14}$$

where $\mathbf{M}_{ISO} = \frac{1}{3}tr(\mathbf{M})\mathbf{I}$ represents the isotropic part and $\mathbf{M}_{DEV} = m_1^*\mathbf{a_1}\mathbf{a_1}^\mathsf{T} + m_2^*\mathbf{a_2}\mathbf{a_2}^\mathsf{T} + m_3^*\mathbf{a_3}\mathbf{a_3}^\mathsf{T}$ the deviatoric part. This decomposition is unique. $\mathbf{M_{DEV}}$ often has no direct geological meaning and $\mathbf{M}$ is therefore further decomposed into different geologically meaningful sources (Vavryčuk (2015)) as:

$$\mathbf{M} = \mathbf{M}_{ISO} + \mathbf{M}_{DC} + \mathbf{M}_{CLVD} \tag{2-15}$$

where $\mathbf{M}_{DC}$ represents the double-couple portion and $\mathbf{M}_{CLVD}$ represents the compensated linear vector dipole portion.

The isotropic portion ($\mathbf{M}_{ISO}$) of the moment tensor represents a uniform volume change. Only P-waves radiate from a purely isotropic source. A positive isotropic component corresponds to an expansion/explosion. A negative component corresponds to a contraction/implosion (Vavryčuk (2015)).

The double couple portion ($\mathbf{M}_{DC}$) represents the force equivalent of shear faulting on a planar fault. It is referred as a double couple because there are two force couples and two conjugate fault plane orientations that equally explain the displacement (Jost and Herrmann (1989)).

The compensated linear vector dipole portion ($\mathbf{M}_{CLVD}$) represents a normal dislocation on a plane. The normal displacement from one linear vector dipole is "compensated" by opposing displacement from the other two linear vector dipoles so there is no net volume change (Jost and Herrmann (1989)).

Non-double couple mechanisms (isotropic and compensated linear vector dipole) have been associated with nuclear explosions, volcanic events, geothermal regions and induced seismicity, including hydraulic fracturing (Vavryčuk (2015)). These non-double couple mechanisms explain processes such as fracture collapse or fracture opening, possibly associated with fluid movement.

(2-15) can be further decomposed into:

$$\mathbf{M} = M_{ISO}\mathbf{E}_{ISO} + M_{DC}\mathbf{E}_{DC} + M_{CLVD}\mathbf{E}_{CLVD} \tag{2-16}$$

where $\mathbf{E}_{ISO}$, $\mathbf{E}_{DC}$ and $\mathbf{E}_{CLVD}$ are elementary base tensors, and M_{ISO}, M_{DC} and M_{CLVD} are moment magnitudes. The values M_{ISO}, M_{DC} and M_{CLVD} can be further normalized and expressed using scalar seismic moment M and relative scale factors C_{ISO}, C_{DC} and C_{CLVD} as:

$$\begin{bmatrix} C_{ISO} \\ C_{DC} \\ C_{CLVD} \end{bmatrix} = \frac{1}{M} \begin{bmatrix} M_{ISO} \\ M_{DC} \\ M_{CLVD} \end{bmatrix} \tag{2-17}$$

where $M = |M_{ISO}| + |M_{DC}| + |M_{CLVD}|$. The scale factors C_{ISO}, C_{DC} and C_{CLVD} satisfy:

$$|C_{ISO}| + C_{DC} + |C_{CLVD}| = 1 \tag{2-18}$$

where C_{DC} is always positive and ranges from 0 to 1. Values for C_{ISO} and C_{CLVD} range from -1 to 1. This method for the decomposition of the moment tensor allows us to physically interpret the set of nine dipole forces representing a general point seismic source and to identify basic behaviours of the source in isotropic media (Jost and Herrmann (1989)).

In the following section are summarized basic rules to identify the source mechanisms in isotropic media according to Vavryčuk (2015):

- An explosion/implosion is an isotropic source, it is characterized by $C_{ISO} = \pm 1$ and by zero C_{CLVD} and C_{DC}. If C_{ISO} is positive, the isotropic component is due to an explosion. The source is implosive if the value is negative.
- Shear faulting (displacement of blocks along the fault plane) is represented by the double-couple force and characterized by $C_{DC} = 1$ and by zero C_{ISO} and C_{CLVD}.
- Pure tensile or compressive faulting (displacement of blocks normal to the fault plane) is free of shearing and thus characterized by zero C_{DC}. However, the non-DC components contain both ISO and CLVD. The ISO and CLVD components are of the same sign: they are positive for tensile faulting and negative for compressive faulting.
- Shear-tensile source, which combines both shear and tensile faulting is characterized by nonzero ISO, CLVD, and DC components. The positive values of C_{ISO} and C_{CLVD} correspond to fault opening tensile mechanisms. The negative values of C_{ISO} and C_{CLVD} correspond to fault closing compressive mechanisms.

- Shear faulting on a non planar fault is characterized generally by a nonzero C_{DC} and C_{CLVD}. The C_{ISO} is zero, because no volumetric changes are associated with this type of source.

2-3 Deterministic inversion

As mentioned in section 1-2-1 deterministic inversion is a widely used approach to derive the moment tensor values from microseismic measurements.

A general formulation for a forward modeling problem can be given by:

$$\mathbf{d} = g(\mathbf{m}) \tag{2-19}$$

where $\mathbf{d}$ represents the data, $\mathbf{m}$ the model parameters (physical quantities we want to retrieve) and the operator g represents the physical relations between model and data. If the problem is linear, then the right hand side of (2-19) can be expressed as the matrix-vector multiplication of a Jacobian matrix $\mathbf{G}$ with the vector $\mathbf{m}$.

In the case of moment tensor inversion, the data values $\mathbf{d}$ correspond to the seismic measurements $\mathbf{u}$, the moment tensor components are expressed as $\mathbf{m}$ and the matrix operator $\mathbf{G}$ contains the spatial derivatives of the earth's impulse response (Green's functions) (Vavryčuk (2015)). This system of equations can be derived through the approach explained in the following paragraphs.

The forward problem general formulation can be stated for our specific case starting from the definition of the displacements according to a seismic moment tensor explained in section 2-1. Let's consider (2-7), that can be further simplified into (2-20) (from Koch (1991)):

$$u_i(\mathbf{x}, t) = G_{ikl}(\mathbf{x}, t) * M_{kl}(t) = G_{in}(\mathbf{x}, t) * m_n(t) \tag{2-20}$$

where the six independent components M_{xx}, M_{xy}, M_{xz}, M_{yy}, M_{yz}, M_{zz} can be rearranged into the vector $m_n(t)$ of values m_1, m_2, m_3, m_4, m_5, m_6. According to Sipkin (1982) and Aki and Richards (2002), for a far-field approximation the six spatial derivatives of the Green's functions $G_{in}(\mathbf{x}, t)$ can be calculated as linear combinations of functions called "elementary seismograms" (elementary seismograms are further explained in Chapter 3).

Then, (2-20) can be described by a linear system of equations in matrix form as (Vavrycuk (2015)):

$$\mathbf{u} = \mathbf{G}\mathbf{m} \tag{2-21}$$

in which $\mathbf{G}$ is the $K \times 6$ matrix of the spatial derivatives of Green's functions for K receivers:

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ G_{K1} & G_{K2} & G_{K3} & G_{K4} & G_{K5} & G_{K6} \end{bmatrix}, \tag{2-22}$$

$\mathbf{m}$ is the vector composed of the six independent components of the moment tensor $\mathbf{M}$:

$$\mathbf{m} = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \end{bmatrix}^T, \tag{2-23}$$

and $\mathbf{u}$ is the vector of the full waveform measurements (seismograms) observed at the K receivers:

$$\mathbf{u} = \begin{bmatrix} u_x^1 & u_y^1 & u_z^1 & u_x^2 & u_y^2 & u_z^2 & \cdots & u_x^K & u_y^K & u_z^K \end{bmatrix}^T, \tag{2-24}$$

where the subscript indicates the field component and the superscript indicates the receiver. The deterministic inversion of the moment tensor can be performed by inverting (2-21) in the time domain or in the frequency domain.

Frequency-domain moment-tensor inversion

From (Vavrycuk (2015)), we can write (2-20) in the frequency domain ω and obtain:

$$u'_i(\mathbf{x}, \omega) = G'_{in}(\mathbf{x}, \omega)\ m'_n(\omega) \tag{2-25}$$

where $u'_i(\mathbf{x}, \omega)$, $G'_{in}(\mathbf{x}, \omega)$ and $m'_n(\omega)$ are the Fourier transformations of measurements, spatial derivatives of Green's functions and moment tensor components, respectively. The previous expression can be written in matrix form as:

$$\mathbf{u}' = \mathbf{G}'\mathbf{m}' \tag{2-26}$$

where the matrix of spatial derivatives of Green's functions, $\mathbf{G}'$, the vector of the moment tensor, $\mathbf{m}'$, and the measurements, $\mathbf{u}'$, are all expressed in the frequency domain. The equation corresponds to a linear system that can be solved for the estimated moment tensor $\hat{\mathbf{m}}'$, by calculating $\mathbf{G}'_\mathbf{p}$, the pseudoinverse of $\mathbf{G}'$, and by applying:

$$\hat{\mathbf{m}}' = \mathbf{G}'_\mathbf{p}\mathbf{u}' \tag{2-27}$$

The time-domain components of the moment tensor are then computed by applying an inverse Fourier transform of the solution $\hat{\mathbf{m}}'$ to obtain the solution $\hat{\mathbf{m}}$.

Time-domain moment-tensor inversion

Time-domain moment-tensor inversion can be performed using the approximation of the spatial derivatives of the Green's functions as elementary seismograms $G_{in}(\mathbf{x}, t)$ (Vavrycuk (2015)). For a point source, the time dependent moment tensor can be factorized as:

$$m_n(t) = m_n\ S(t) \tag{2-28}$$

where $S(t)$ is the source-time function, with the assumption that the focal mechanism is independent of time. Given a known source function $S(t)$, and factorizing as in (2-28), we define now a function $S_{in}(\mathbf{x}, t)$ as:

$$S_{in}(\mathbf{x}, t) = S(t) * G_{in}(\mathbf{x}, t) \tag{2-29}$$

We can rewrite (2-20) as:

$$u_i(\mathbf{x}, t) = [S(t) * G_{in}(\mathbf{x}, t)]m_n = S_{in}(\mathbf{x}, t)\ m_n \tag{2-30}$$

(2-30) can be further expressed in matrix form as:

$$\mathbf{u} = \mathbf{Sm} \tag{2-31}$$

where $\mathbf{S}$ contains the matrix of the convolution of the spatial derivatives of the Green's functions with the source function, $\mathbf{m}$ contains the moment tensor components and $\mathbf{u}$ contains the measurements, all expressed in the time domain. Finally, the estimated seismic moment tensor $\hat{\mathbf{m}}$ is obtained by inverting the previous equation as:

$$\hat{\mathbf{m}} = \mathbf{S_p u} \tag{2-32}$$

where $\mathbf{S_p}$ is the pseudoinverse of the matrix $\mathbf{S}$.

Chapter 3

Forward modeling problem

In this chapter the problem of calculating seismograms from a given moment tensor (forward problem) is analyzed and solved by means of the Discrete WaveNumber (DWN) method as described by (Bouchon and Aki (1977)). The forward modeling operation corresponds to the solution of the equation:

$$\mathbf{u} = \mathbf{G}\mathbf{m} \tag{2-21}$$

where the different components are explained in (2-21-2-24). The fault rupture geometry (strike, dip, rake), the moment magnitude, the source-receiver geometry and the geophysical properties of the propagation model (velocity and density) are used to define the moment tensor $\mathbf{m}$ and the spatial derivatives of the Green's functions $\mathbf{G}$, necessary for the calculation of the seismograms $\mathbf{u}$.

The matrix $\mathbf{G}$ of the spatial derivatives of the Green's functions can be calculated using the DWN method. These calculations are performed in the frequency-wavenumber domain (Bouchon and Aki (1977)). This approach assumes that the radiation from a source can be represented as a cylindrical wave and it introduces a spatial periodicity of the sources to discretize the radiated wavefield (Bouchon (2003)).

3-1 Discrete WaveNumber method

According to Bouchon and Aki (1977) the seismic wavefield radiated by an isotropic point source in a homogeneous space can be expressed in the cylindrical coordinates system (r, z), r being the radius and z the depth, by the potential Φ, using the Sommerfeld integral as:

$$\Phi(r, z; \omega) = \frac{iV_s(\omega)}{4\pi} \int_0^\infty \frac{k}{\nu} J_0(kr) e^{-i\nu|z|} dk \tag{3-1}$$

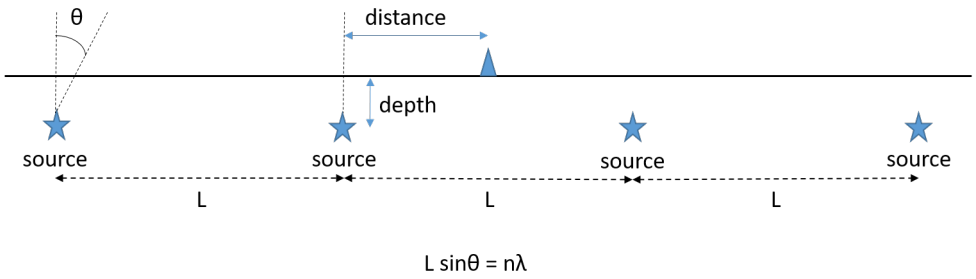

Figure 3-1: Physical interpretation of the DWN method. The original source is replaced by a finite number of sources distributed horizontally with spatial interval L (Bouchon (2003)).

where $V_s(\omega)$ is the volume change at the source, ω is the angular frequency, t is time, $\nu = \sqrt{\frac{\omega^2}{\alpha^2} - k^2}$ is the vertical wavenumber, $\mathrm{Im}(\nu) < 0$, α is the compressional wave velocity, k is the horizontal wavenumber and J_0 denotes the zeroth order Bessel function. The origin of the coordinate system is located at the source position.

(3-1) can be discretized (DWN method) as:

$$\Phi(r,z;\omega) = \frac{iV_s(\omega)}{2}\sum_{n=0}^{N}\frac{k_n}{\nu_n}J_0(k_n r)e^{-i\nu_n|z|} \tag{3-2}$$

with $k_n = \frac{2\pi}{L}n$. This can be physically interpreted as a substitution of the original source radiating in all the directions with a set of N sources radiating planar waves and distributed horizontally with spatial interval L (see Figure 3-1) (Bouchon (2003)). N is properly chosen to minimize the discretization error to a negligible value.

The point source can be expressed as a force of Cartesian components $\mathbf{F} = (F_x, F_y, F_z)$. The corresponding discretized P, SV and SH wave potentials (Φ, Ψ, χ) for this force can be calculated in the cylindrical coordinate system (r, θ, z) as (Bouchon (2003)):

$$\Phi(r,\theta,z;\omega) = \frac{1}{2L\rho\omega^2}\left[\mathrm{sgn}(z)F_z\sum_{n=0}^{N}k_n J_0(k_n r)e^{-i\nu_n|z|} - i(F_x\cos\theta + F_y\sin\theta)\sum_{n=0}^{N}\frac{k_n^2}{\nu_n}J_1(k_n r)e^{-i\nu_n|z|}\right] \tag{3-3}$$

$$\Psi(r,\theta,z;\omega) = \frac{1}{2L\rho\omega^2}\left[-iF_z\sum_{n=0}^{N}\frac{k_n}{\gamma_n}J_0(k_n r)e^{-i\gamma_n|z|} + \mathrm{sgn}(z)(F_z\cos\theta - F_y\sin\theta)\sum_{n=0}^{N}J_1(k_n r)e^{-i\gamma_n|z|}\right] \tag{3-4}$$

$$\chi(r,\theta,z;\omega) = i\frac{F_y\cos\theta - F_x\sin\theta}{2L\rho\beta^2}\sum_{n=0}^{N}\frac{1}{\gamma_n}J_1(k_n r)e^{-i\gamma_n|z|} \tag{3-5}$$

with

$$\gamma_n = \sqrt{\frac{\omega^2}{\beta^2} - k_n^2}, \quad \mathrm{Im}(\gamma_n) < 0 \tag{3-6}$$

and

$$\mathrm{sgn}(z) = 1 \text{ for } z > 0, \quad \mathrm{sgn}(z) = -1 \text{ for } z < 0 \tag{3-7}$$

where ρ and β are the model density and shear-wave velocity, and J_1 is the Bessel function of the first order.

Finally, the spatial derivatives of the Green's functions $\mathbf{G}$ are obtained using the Helmholtz potentials decomposition theorem (Aki and Richards (2002)) expressed in cylindrical coordinates (Cotton and Coutant (1996)):

$$\mathbf{G}(r,\theta,z) = \nabla\Phi + \nabla\times(\mathbf{e_z}\Psi) + \nabla\times\nabla\times(\mathbf{e_z}\chi) \tag{3-8}$$

where $\mathbf{e_z}$ is the vertical unit vector.

3-2 Modified Discrete WaveNumber method for full-moment sources

In practice, the source is not a point and the previous equations must be modified to take into account the effects of a more complex general source (full-moment source). The elastic radiation of such a general source can be obtained by the superposition of elementary point sources (Aki and Richards (2002)). This approach leads to a slightly modified version of the DWN method as described in Cotton and Coutant (1996).

This method calculates the spatial derivatives of the Green's functions ($\mathbf{G}$) for six independent moment tensor sources (five double couples (DC) and one isotropic (ISO)) in a homogeneous source layer, expressed as:

$$\mathbf{M}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{M}^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{M}^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \mathbf{M}^4 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-9}$$

$$\mathbf{M}^5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{M}^6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The full moment tensor is simply a linear combination of the six elementary moment tensors multiplied by trigonometric coefficients a_n $(n = 1, 2, ..., 6)$:

$$\mathbf{M} = \sum_{n=1}^{6} a_n \mathbf{M}^n = \begin{bmatrix} -a_4 + a_6 & a_1 & a_2 \\ a_1 & -a_5 + a_6 & -a_3 \\ a_2 & -a_3 & a_4 + a_5 + a_6 \end{bmatrix} \tag{3-10}$$

Since we assume that all elementary moment tensors $\mathbf{M^n}$ have no time dependence, these can be convolved with the corresponding Green's functions spatial derivatives, now expressed in a Cartesian coordinates system, to generate six elementary seismograms $E_i^n(\mathbf{x}, t)$:

$$E_i^n(\mathbf{x}, t) = \sum_{k,l} G_{ikl}(\mathbf{x}, t) * M_{kl}^n \tag{3-11}$$

Finally, we can compute the full seismograms as a linear combination of the six elementary seismograms with the same coefficients a_n, convolved with the known source function $S(t)$, as:

$$u_i(\mathbf{x}, t) = \left(\sum_{n=1}^{6} E_i^n(\mathbf{x}, t) a_n \right) * S(t) \tag{3-12}$$

The previous expression in frequency domain takes the form:

$$u_i(\mathbf{x}, \omega) = \left(\sum_{n=0}^{6} E_i^n(\mathbf{x}, \omega) a_n \right) S(\omega) \tag{3-13}$$

The coefficients a_n are derived by the general moment tensor M_{kl} defined in (2-8) as (Cotton and Coutant (1996)):

$$\begin{aligned} a_1 &= M_{xy} \\ a_2 &= M_{xz} \\ a_3 &= -M_{yz} \\ a_4 &= (-2M_{xx} + M_{yy} + M_{zz})/3 \\ a_5 &= (M_{xx} - 2M_{yy} + M_{zz})/3 \\ a_6 &= (M_{xx} + M_{yy} + M_{zz})/3 \end{aligned} \tag{3-14}$$

The procedure to compute the full seismograms (from Cotton and Coutant (1996)) follows these steps: first, we define the potentials of the sources at the earth model layer where the source is located. We then propagate the potentials through the earth model calculating their value at the different layers taking into account the transmission and reflection effects at the interfaces through the reflectivity method of Kennett and Kerry (1979) (described in Appendix B-3). The spatial derivatives of the Green's functions can be computed at every model layer by differentiating the corresponding potentials. This procedure is detailed in Appendix B. Finally, the elementary and the full seismograms can be calculated at the receiver position as in (3-11)-(3-13).

3-3 Forward modeling algorithm

The calculations described in the previous paragraphs can be represented by the diagram of Figure 3-2 and summarized by the following algorithm:

1. Initialize the source-receiver geometry: depths, azimuths and radial distances.
2. For each horizontal wavenumber k_r:
   (a) Calculate the Bessel functions J_0 and the vertical wavenumbers ν and γ.
   (b) Calculate the reflection/transmission coefficients (Appendix B-3).
   (c) Derive upward and downward potentials radiated from the source layer. The expressions for the potentials (Φ, Ψ and χ) at the source layer are calculated for each elementary moment tensor component (Appendix B-1).
   (d) The potential fields for the six sources S_1, S_2, ..., S_6 are propagated through the model (Appendix B-2). At the receivers each of them results in upgoing and downgoing potentials, including possible conversions. Expressions for the potentials are calculated for each elementary moment-tensor component.
   (e) Previously calculated potentials (Φ, Ψ and χ) at the receivers, for each elementary moment-tensor component, are differentiated to obtain the spatial derivatives of the Green's functions (Appendix B-4).
3. The spatial derivatives of the Green's functions calculated for every horizontal wavenumber are summed to obtain G_{ikl}. The corresponding expressions are transfomed from cylindrical coordinate system to a Cartesian coordinate system.
4. The elementary seismograms E_i^n are calculated from the elementary moment tensors M_{kl}^n and spatial derivatives of the Green's functions G_{ikl}.
5. Calculate the trigonometric coefficients a_n of the elementary sources.
6. Multiply the source function $S(\omega)$ with the sum of the elementary seismograms E_i^n, weighed for the coefficients a_n, to obtain the final displacements $u_i(\mathbf{x}, \omega)$ at each receiver as in (3-13).
7. Apply inverse Fourier transform to obtain the displacements for each component and receiver in the space/time domain.

The algorithm has been implemented in Python language, and performs the calculations for each frequency and horizontal wavenumber in parallel, using a meshgrid in the frequency/horizontal wavenumber domain. A meshgrid is a rectangular grid, containing all the possible combination of couples of values extracted by two given one-dimensional arrays representing the frequency axis and horizontal wavenumber axis. The advantage of using a meshgrid relies on the fact that the operations for all horizontal wavenumber and frequency values can be performed simultaneously, reducing the computation time.

The main core of the code is listed in Appendix C.

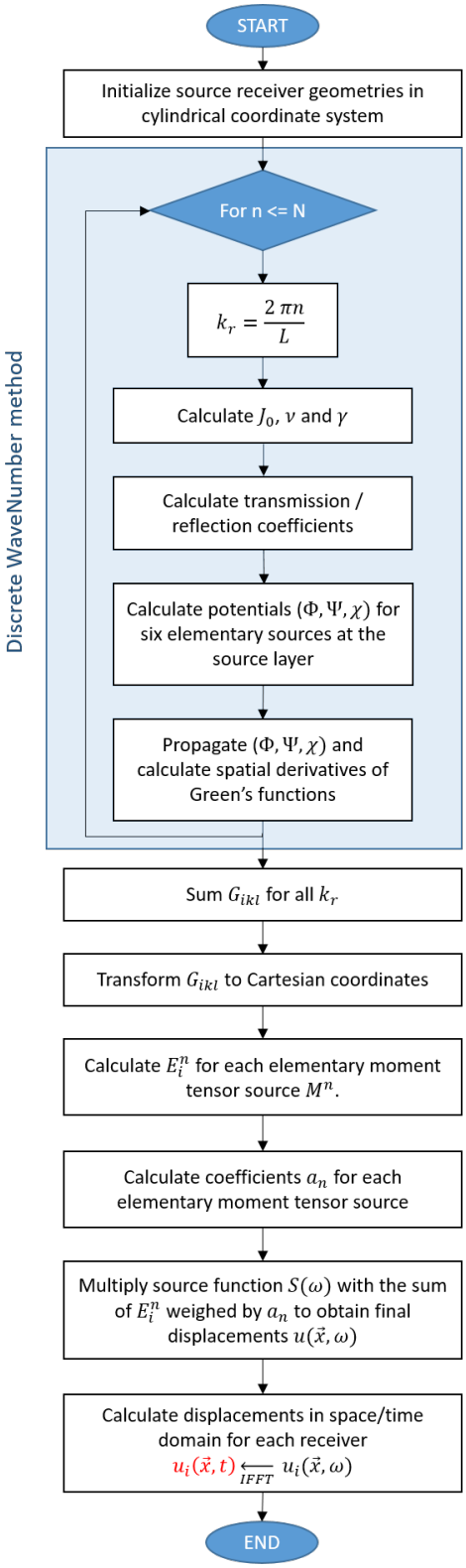

Figure 3-2: Forward modeling algorithm flowchart.

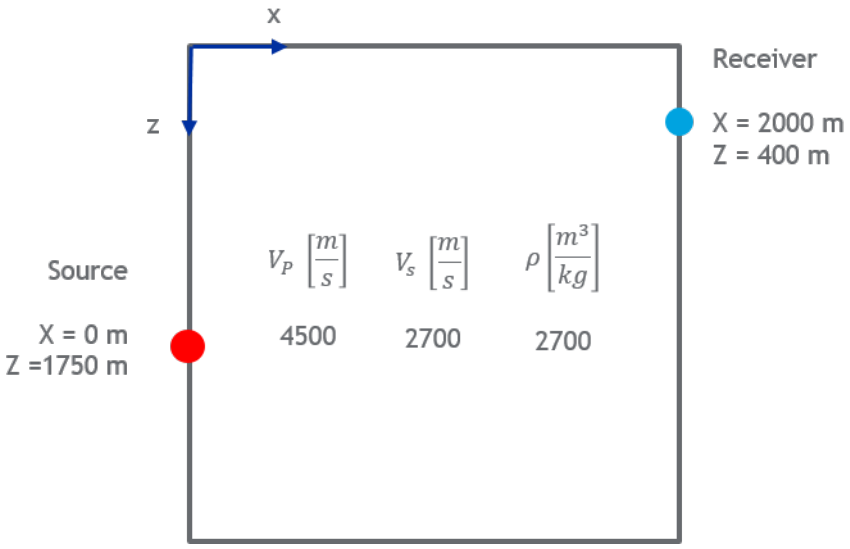

Figure 3-3: Source-Receiver geometry in the homogeneous velocity model.

3-4 Benchmarking

The code implementing the 3D forward modeling algorithm previously described and used in this thesis to generate synthetic datasets was benchmarked against a 2D reflectivity code based on (Kennett et al. (1990), Frasier (1970), Hubral et al. (1980) and Wapenaar and Berkhout (1989)). The proposed code is working with 3D models, while the pre-existing one works with 2D models. For the comparison and benchmarking we consider a homogeneous velocity model and a layered velocity model described in the following paragraphs. Moreover, sources and receivers are placed in layers with identical medium parameters to ensure that possible normalization differences between the two implementations do not interfere with the comparison results.

In the first comparison example we consider one source and one receiver located in a 2D space, characterised by a homogeneous velocity model as shown in Figure 3-3. We calculate the displacements at the receiver for the P-wave component and the S-wave component by using both methods. In Figure 3-4 the P-wave response for a P-wave source is displayed. In the same manner, the S-wave response for an S-wave source is shown in Figure 3-5. The resulting waveforms generated using the two methods are identical. The comparison of the methods on the amplitude spectra shows slight differences at the higher frequencies. This can be explained due a variation of the resolution in the wavenumber domain as a consequence of the difference in implementation methods: the DWN method uses a variable horizontal wavenumber sampling that can be set to a very fine resolution, while the pre-existing code uses a fix horizontal wavenumber sampling, that in this example was set coarser in comparison to the DWN.

In the second example we repeated the previous experiment but using the layered model presented in Figure 3-6, leading to similar results (see Figures 3-7 and 3-8). This example further validates that the code is suitable for modelling the synthetic data to train the ANN as described in the next steps.

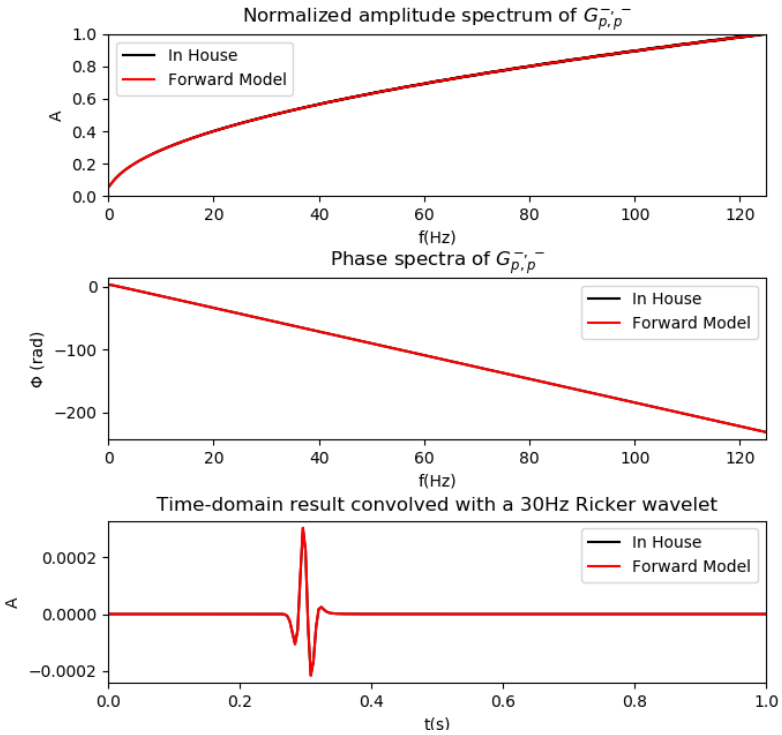

Figure 3-4: Comparison of the new and the pre-existing codes for the upgoing PP wave components in the homogeneous velocity model. Upper panel: Amplitude spectrum. Middle panel: Phase spectrum. Lower panel: Displacement convolved with a 30 Hz Ricker wavelet.

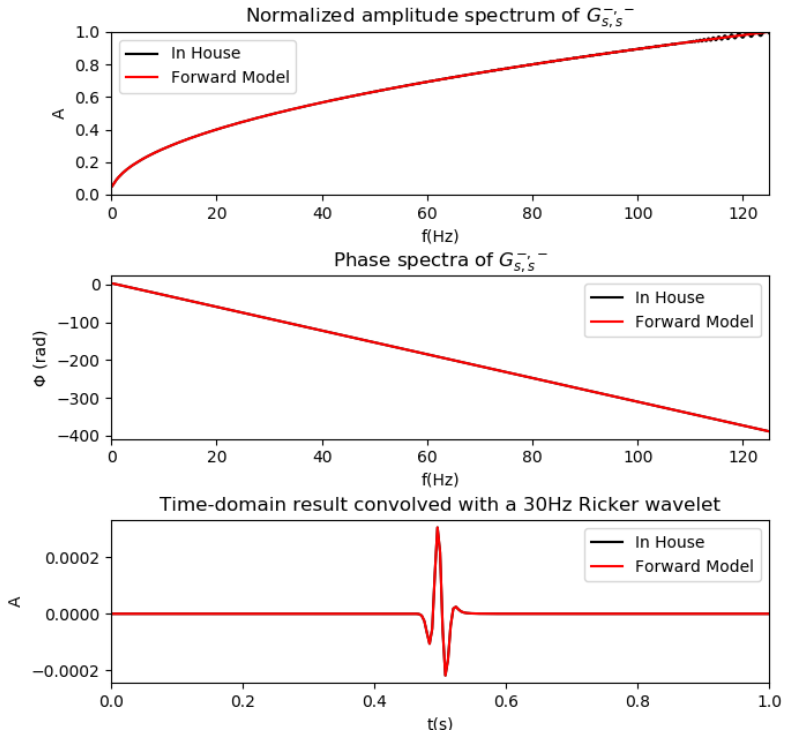

Figure 3-5: Comparison of the new and the pre-existing codes for the upgoing SS wave components in the homogeneous velocity model. Upper panel: Amplitude spectrum. Middle panel: Phase spectrum. Lower panel: Displacement convolved with a 30 Hz Ricker wavelet.

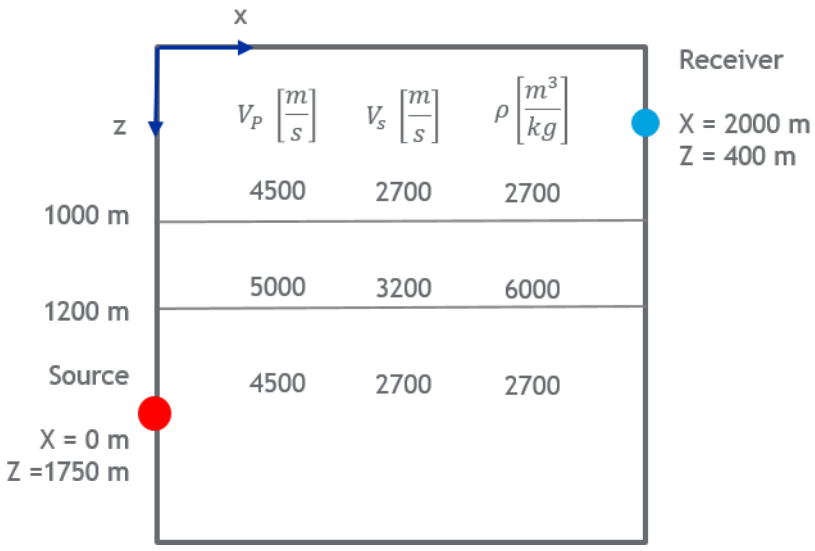

Figure 3-6: Source-Receiver geometry in a 1D layered velocity model.

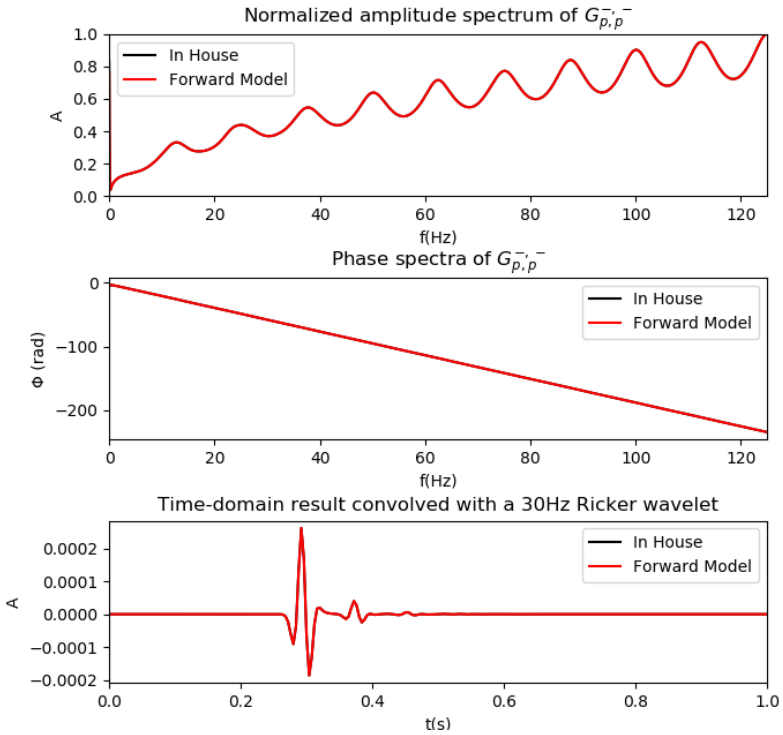

Figure 3-7: Comparison of the new and the pre-existing codes for the upgoing PP wave components in the 1D layered velocity model. Upper panel: Amplitude spectrum. Middle panel: Phase spectrum. Lower panel: Displacement convolved with a 30 Hz Ricker wavelet.

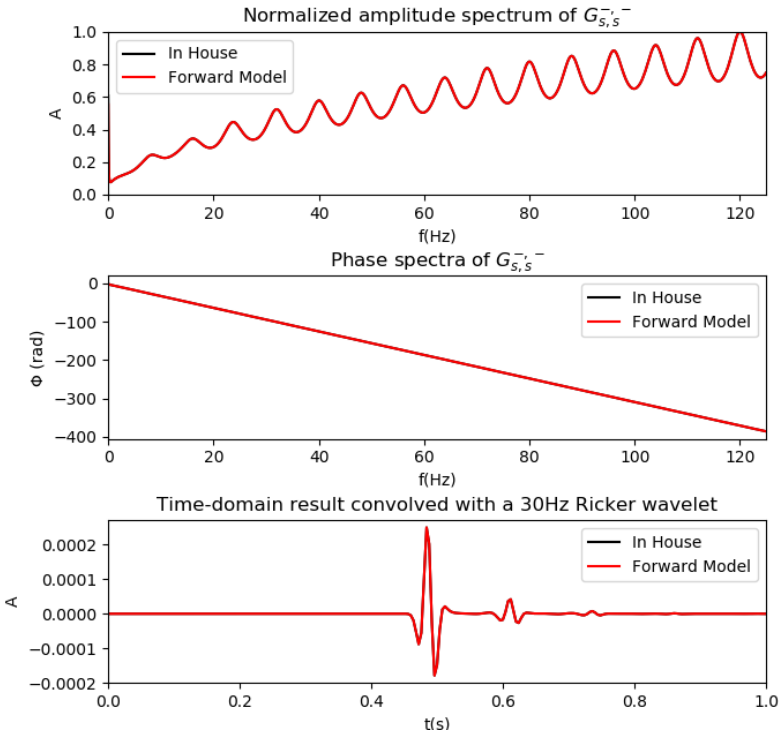

Figure 3-8: Comparison of the new and the pre-existing codes for the upgoing SS wave components in the 1D layered velocity model. Upper panel: Amplitude spectrum. Middle panel: Phase spectrum. Lower panel: Displacement convolved with a 30 Hz Ricker wavelet.

Layer	V_p $[m/s]$	V_s $[m/s]$	ρ $[kg/m^3]$	Q_p	Q_s
1	5000	2500	3000	1000	500

Table 3-1: Velocity model.

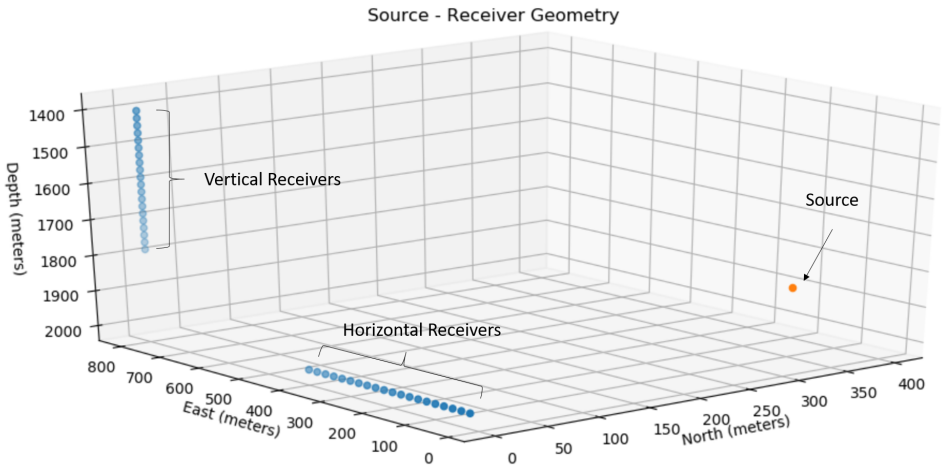

Figure 3-9: Source (orange marker) and receivers (blue markers) geometry.

3-5 Generation of a realistic synthetic dataset

A microseismic borehole synthetic dataset is generated using the forward modeling method described, to be used in the training and testing stages of an ANN (see Chapter 4).

A homogeneous velocity model is considered as described in Table 3-1.

To simplify, we consider a single source characterized by a pure strike-slip event (100% DC) with moment magnitude $M_w = -2$. We use 40 receivers geometrically distributed as shown in in Figure 3-9: 20 receivers are located in a horizontal array and 20 in a vertical array.

The fault angles (strike ϕ, dip φ and rake ψ) are sampled uniformly every $5°$, generating a set of 101,250 different possible fault geometries. This set is used to create synthetic waveforms capturing the possible effects of the variation of the moment tensor components on the acquired microseismic data.

Considering the 40 receivers with their three components, a total of 12,150,000 seismograms have been generated. Two examples of these seismograms modeled for some specific fault geometries are shown in Figures 3-10 and 3-11, respectively.

In these examples the P-wave and S-wave arrivals are clearly distinguishable. It is also clear that different fault geometries are producing different radiation patterns, causing significant variations in the amplitudes of the X, Y and Z components calculated at the receivers. For example, the horizontal components of the P-wave arrivals show very low amplitude in most of the receivers in 3-10 when compared with the ones observed in 3-11.

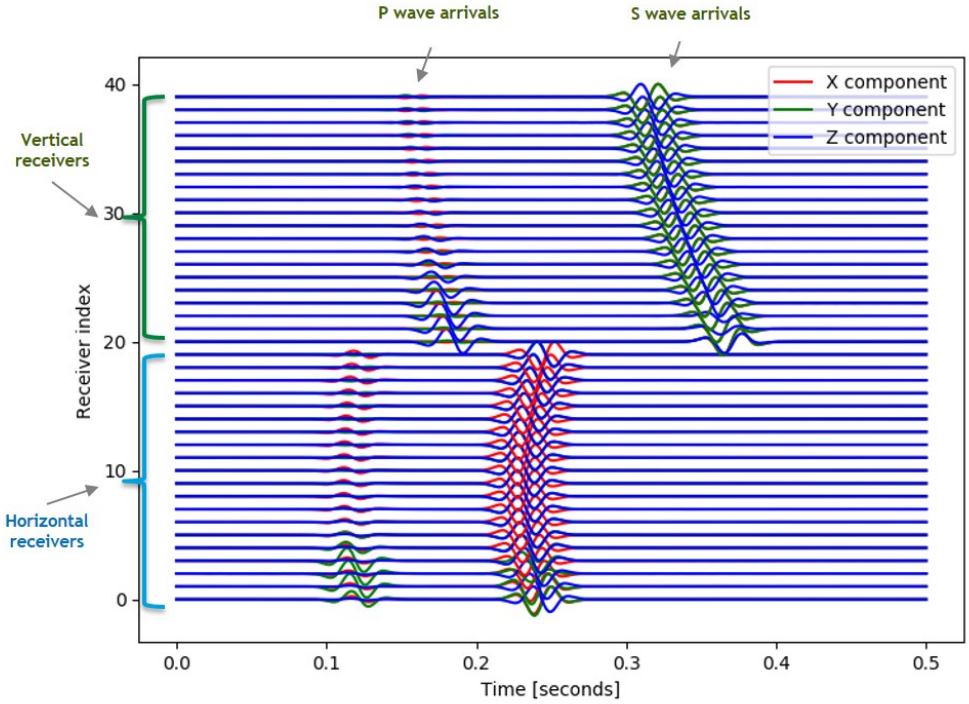

Figure 3-10: Seismograms generated for the fault geometry described by: strike $\phi = 70$, dip $\varphi = 10$, rake $\psi = -125$.

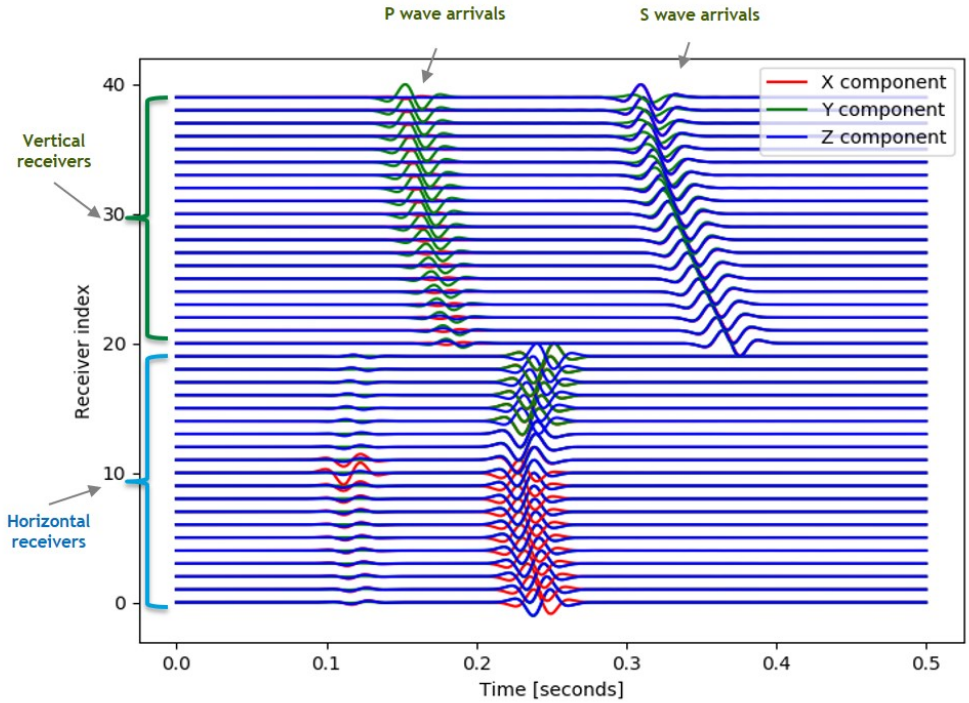

Figure 3-11: Seismograms generated for the fault geometry described by: strike $\phi = 235$, dip $\varphi = 0$, rake $\psi = 155$.

Chapter 4

Inverse problem

The solution to the problem of estimating the moment-tensor components of a microseismic event from the seismograms acquired by arrays of receivers (inverse problem) can be obtained by inverting (2-21) as:

$$\hat{\mathbf{m}} = \mathbf{G}_p \mathbf{u} \tag{4-1}$$

where the moment-tensor components, $\hat{\mathbf{m}}$, can be estimated by multiplying the seismic data, $\mathbf{u}$, with the pseudo-inverse of the Green's functions spatial derivatives matrix, $\mathbf{G}_p$.

In this thesis, instead of solving (4-1) explicitly, we propose to train an ANN to recognize the moment tensor components associated to specific seismograms. To achieve this goal, in this chapter we design a specific ANN architecture, we train it and test it on the borehole microseismic synthetic dataset created in the previous chapter. The ANN is created using the deep learning programming framework Keras, written in Python language, running on top of the machine learning platform TensorFlow (see more in Chollet et al. (2015)).

4-1 ANN architecture design

As explained in Appendix A an ANN architecture is defined by the number of inputs, outputs, hidden layers, and a certain number of neurons in each hidden layer. The neurons connections are defined by coefficients (weight and bias) and activation functions $f(\cdot)$, that introduce non-linearity in the relation linking inputs and outputs.

The architecture of the neural network in this thesis project is based on the results of Ovcharenko et al. (2018), where the authors used an ANN to derive the moment tensor components from attributes (first-arrivals peak value and polarity for P and S waves) extracted from synthetic microseismic data in a vertical borehole array. Here we extend this previous work by using synthetic full waveforms as input data, as well as using an additional horizontal array of receivers. Considering the similarity of our problem with the one discussed in the literature, we decided to adapt the existing ANN architecture for our specific needs.

Layer	Number of Neurons	Matrix shape	Number of Parameters (weights + bias)	Description
Input	30720	1×30720	0	X, Y, $Z \times 40$ receivers $\times$ 256 samples
Hidden 1	150	30720×150	4608150	
Hidden 2	100	150×100	15100	
Hidden 3	50	100×50	5050	
Output	6	50×6	306	six independent moment tensor components

Table 4-1: ANN architecture summary.

Layer	Number of Neurons	Matrix shape	Number of Parameters (weights + bias)	Description
Input	30720	1×30720	0	X, Y, $Z \times 40$ receivers $\times$ 256 samples
Hidden 1	30720	30720×1	0	Gaussian noise layer
Hidden 2	150	30720×150	4608150	
Hidden 3	100	150×100	15100	
Hidden 4	50	100×50	5050	
Output	6	50×6	306	six independent moment tensor components

Table 4-2: ANN architecture summary including noise layer.

The ANN structure proposed for this project (modified from Ovcharenko et al. (2018)) is a fully connected feed-forward network, containing 3 hidden layers with 150, 100 and 50 neurons respectively, where the neurons are connected with hyperbolic tangent activation functions in the hidden layers and linear functions in the output layer. The input layer comprises 30720 neurons (3 components seismograms $\times$ 40 receivers $\times$ 256 time samples), determining the size of the input array. The output layer contains 6 neurons, one for each moment tensor component.

The ANN architecture is summarized in Table 4-1, where in the first column is described the type of the layers, in the second column the number of neurons in these layers, in the third column we find the shape of the matrix described by the neurons connections in the upper and current layer, in the fourth column are expressed the number of parameters defined by the connections between the previous and the current layer (number of weights and bias) and the last column contains a brief description.

To further evaluate the ANN with noisy input data we included in the architecture an additional layer, which adds zero-mean Gaussian noise to the inputs (this structure is summarized in Table 4-2). This layer is only active during the training phase where the ANN learns how to recognize and to properly address the noise.

After designing the ANN architecture we proceed with the training phase: the network learns

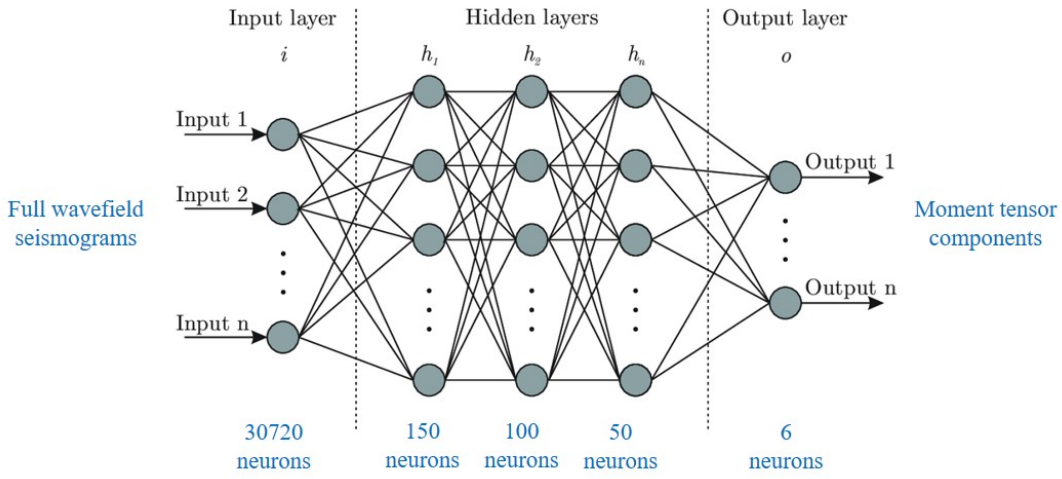

Figure 4-1: ANN architecture diagram (modified from Tano et al. (2020))

how to predict the outputs from the inputs for a specific dataset where the outputs are known by the network (training dataset). A validation phase is also necessary where the ANN performances are checked against a dataset with outputs not seen by the network during the training phase (validation dataset). Finally, we can use the network to predict the output from the input for an unknown dataset (test dataset).

In the examples described in the next sections we randomly divided the dataset generated in the previous chapter so that 40% of it is used for training, 10% for validation and 50% for testing. Testing the ANN with another dataset generated with a different velocity model and/or acquisition geometry could lead to network issues in generalization. A proper analysis of the ability of the ANN to generalize might be considered in future developments of this work.

4-2 ANN training and validation

The training phase in machine learning is a process where the neural networks learns the predict outputs for specific inputs. The network training is an iterative procedure and begins by setting random values for the weights and biases. Input arrays are presented to the network one at a time. The network processes the inputs, and compares the resulting outputs against the desired outputs, and calculates the differences (defined as losses). Losses are then backpropagated, so that weights and biases can be updated and reused in the next iteration of training. After all the arrays are presented, the process might be repeated until satisfactory results in terms of prediction accuracy are reached (Bishop (2006)). Each iteration corresponding to a full evaluation of the training dataset is called an "epoch" (Mitchell (1997)).

The training process should be validated while updating the network parameters (Bishop (2006)). For this purpose we use a validation dataset. During the validation step the network parameters are not updated but it is checked that an increasing accuracy in the prediction capabilities is observed not only in the training dataset but also in the validation one, never

seen by the network before. If the prediction accuracy increases during the processing of the training dataset but decreases or stay constant during the evaluation of the validation dataset, then the prediction model implemented by the network is overfitting the data and it is necessary to stop the training.

There are several techniques that can be applied to avoid overfitting (Bishop (2006), Mitchell (1997)). The simplest one is to decrease the complexity of the model either reducing the number of layers or the number of neurons per layer. Another easy technique is to use "early stopping", which means to stop the training process when the validation error starts to increase. Overfitting can also be avoided by increasing the dataset size or including noise which can be seen as a type of data augmentation. Additionally, more complex techniques exist such as regularization (adding certain penalty terms in the loss function) and dropout (randomly dropping neurons from the model in each training iteration).

4-2-1 Standardization of the data

An important step before training the ANN is to standardize the data to be used in the training so that the value of each input set has a zero mean and unit variance. The final purpose of this standardization is to have the input data spanning over the same range of values, therefore simplifying the work of the optimization algorithm during the training phase.

We used a simple standardization scheme based on the calculation of the data min, max and mean (Mean Standardization):

$$x' = \frac{x - mean(x)}{max(x) - min(x)} \tag{4-2}$$

where the standardized data x' is computed by the not standardized data x.

4-2-2 Loss function

To improve the ANN, the prediction error at each training iteration must be estimated. This requires the choice of an error calculation function, called loss function. The purpose of this function is to estimate the loss of the algorithm so that the weights can be updated with the purpose of reducing the loss on the next training iteration.

Since the neural network is trying to solve a regression, we tested two loss functions typically used for these type of problems (Binder (2018), Mitchell (1997), Chollet et al. (2015)): the mean squared error function and the cosine similarity function.

The mean squared-error function (MSE) calculates the average of the squared differences between the true values y_i and the predicted values $\hat{y}_i$:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4-3}$$

where N is the number of the elements of the predicted and true vectors, $\mathbf{y}$ and $\hat{\mathbf{y}}$ respectively.

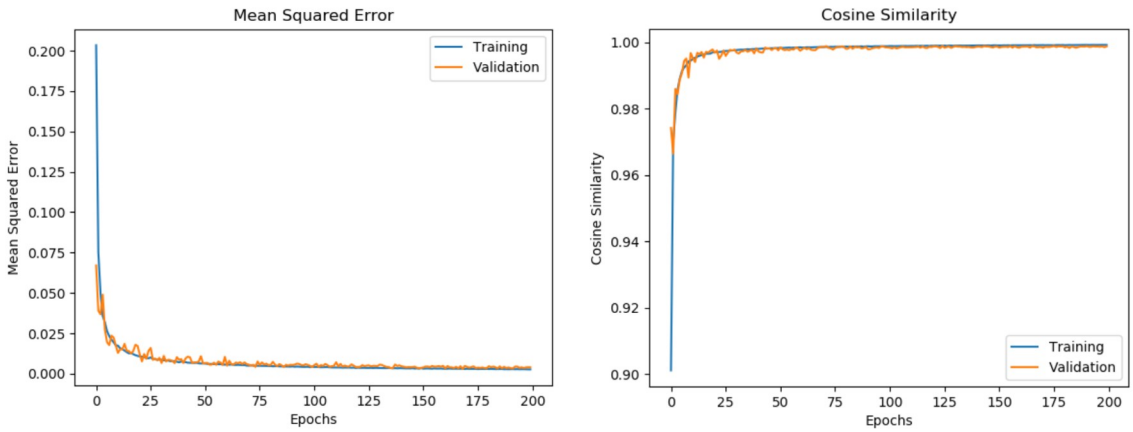

Figure 4-2: Loss-function history plot, while training the ANN with 8000 microseismic events. Left panel: mean squared-error function. Right panel: cosine similarity function. Blue line: training dataset error. Orange line: validation dataset error.

The result is always positive and a perfectly predicted value will produce a result equal to zero. Moreover, the squaring operation penalizes more large errors than small errors (Chollet et al. (2015)).

The cosine similarity function measures the similarity between two non-zero vectors of an inner product space and can be defined as (Hazewinkel (1994), Bishop (2006)):

$$CS = cos\theta = \frac{\mathbf{y} \cdot \hat{\mathbf{y}}}{\|\mathbf{y}\|\|\hat{\mathbf{y}}\|} = \frac{\sum_{i=1}^{N} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{N} y_i^2}\sqrt{\sum_{i=1}^{N} \hat{y}_i^2}} \tag{4-4}$$

where θ is the angle between the predicted and true vectors and "$\|\cdot\|$" represents their L2 norm. CS ranges from -1 for opposite vectors, to 1 for equal vectors. A value of 0 indicates that the vectors are orthogonal.

Figures 4-2 and 4-3 display the training results of the neural network, for 8000 and 16000 microseismic events, respectively, and in terms of losses vs. epochs for a mean squared error function and for a cosine similarity function. In both cases convergence is obtained after about 75 epochs and the network accuracy obtained with the two loss functions is practically the same. Also, the training time is increasing almost linearly with the number of training data and it does not change by using one loss function or the other (see Table 4-3).

In the following examples we use the mean squared error as the loss function for training the ANN, which is usually recommended when trying to estimate real valued outputs (Goodfellow et al. (2016)). However, it is possible to use the cosine similarity function as a metric to evaluate the prediction results.

4-2-3 Optimizer

During the training phase an optimization algorithm (optimizer) has to be chosen to estimate the network parameters that minimize the loss function (Mitchell (1997)).

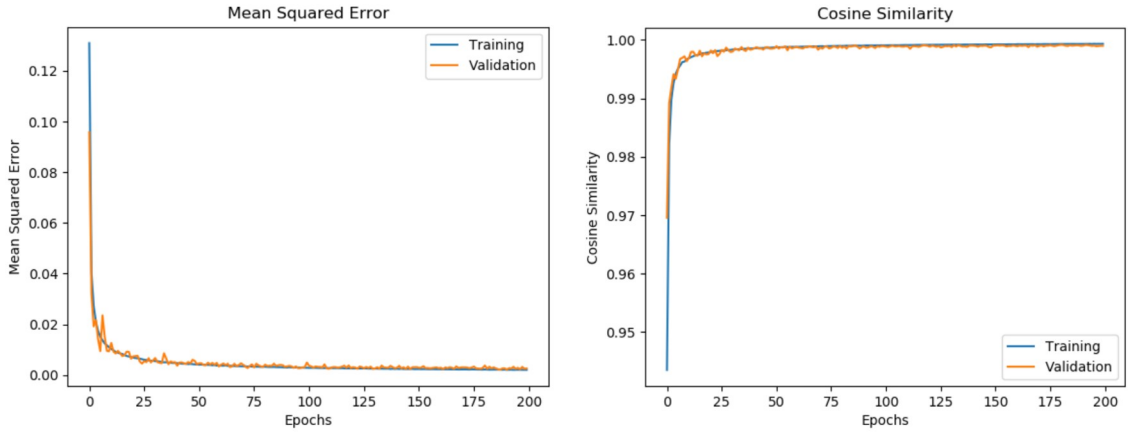

Figure 4-3: Loss-function history plot, while training the ANN with 16000 microseismic events. Left panel: mean squared-error function. Right panel: cosine similarity function. Blue line: training dataset error. Orange line: validation dataset error.

Events	Mean Squared Error	Cosine Similarity
8000	3.04 minutes	3.10 minutes
16000	5.93 minutes	6.06 minutes

Table 4-3: ANN Training Computation Time.

For complex non-linear problems iterative optimizers are normally chosen, typically based on the gradient descent algorithm.

If we consider the neural network as a non-linear function $\mathbf{y}$, applied to an input vector $\mathbf{x}$ and we assume to use a mean square error loss function, the task of the training phase is to find the weights $\mathbf{w}$ that minimize the function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} ||\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n||^2 \tag{4-5}$$

where N represents the number of samples used in the training, $n = 1, ..., N$, and $\mathbf{t}$ is the target vector corresponding to the desired output.

Considering this error function as a surface, making a small step in the weight space from $\mathbf{w}$ to $\mathbf{w} + \delta\mathbf{w}$ will produce a change in the error function $\delta E = \delta\mathbf{w}\nabla E(\mathbf{w})$, where the vector $\nabla E(\mathbf{w})$ points to the direction with the greatest increase in the error function. A global or local minimum value of the error function $E(\mathbf{w})$ will correspond to a point in the weight space where $\nabla E(\mathbf{w}) = 0$. Moreover, making a step in the direction $-\nabla E(\mathbf{w})$ will decrease the error value.

The error minimization problem can be solved iteratively, choosing an initial value $\mathbf{w}^0$ for the weights and moving in small steps as:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \delta\mathbf{w}^{(\tau)} \tag{4-6}$$

where τ represents a time step. The gradient descent method uses the gradient information to choose the weight update in the direction of the negative gradient as follows:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \tag{4-7}$$

where the parameter $\eta > 0$ corresponds to the "learning rate", which represents the distance of the step in the negative gradient direction. After every iteration the gradient is computed for the new weight vector and the process is repeated.

Following the previous work of Ovcharenko et al. (2018) in this thesis we use a modification of the gradient descent method called RMSprop optimizer which implements an adaptive learning rate (Tieleman and Hinton (2012)) as:

$$\mathbf{v}^{(\tau)} = \beta \mathbf{v}^{(\tau-1)} + (1-\beta) \nabla E(\mathbf{w}^{(\tau)})^2 \tag{4-8}$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \frac{\eta}{\sqrt{\mathbf{v}^{(\tau)} + \epsilon}} \nabla E(\mathbf{w}^{(\tau)}) \tag{4-9}$$

where the momentum $\mathbf{v}$ contains the "history" of the values of the weights used in the previous time steps, β determines the decay of $\mathbf{v}$, and ϵ is a term to ensure stability. This approach improves the learning process (Tieleman and Hinton (2012)).

In our ANN we set the values $\eta = 0.001$, $\beta = 0.1$ and $\epsilon = 1 \times 10^{-7}$, following Ovcharenko et al. (2018).

4-2-4 Batch size

The batch size defines the number of samples of the training dataset used in the estimate of the error gradient. For each update the training dataset will be divided into subsets called "batches", in which the model evaluate and modify the weights (Bishop (2006)).

The batch size is a parameter that can affect the computation of the error gradient. A large batch size typically increases the ANN prediction accuracy, but at the same time can produce poor generalization behaviour of the network (not performing well on not seen data samples). On the other side, a small batch size results in faster learning for the network, but also increases the possibility of being trapped in local minima due to the limited amount of data seen by the network at once (Goodfellow et al. (2016)).

Figure 4-4 shows the mean squared error as a function of the epochs, while training the ANN using different batch sizes and 16000 microseismic events. With a red line we point at the number of epochs needed to reach an acceptable mean squared error. We can observe that for larger batch sizes the number of epochs necessary to obtain such an error is larger. Moreover, we can observe that for all the batch sizes evaluated we have reached an equal mean squared error asymptotic behaviour, which is an optimal minimum.

Additionally, we observed that the computation time of the training phase increases as we decrease the batch size as shown in table 4-4. In 4-5 we show a summary of the analysis of the batch size. In the following examples we use a batch size of 200 samples to train the ANN.

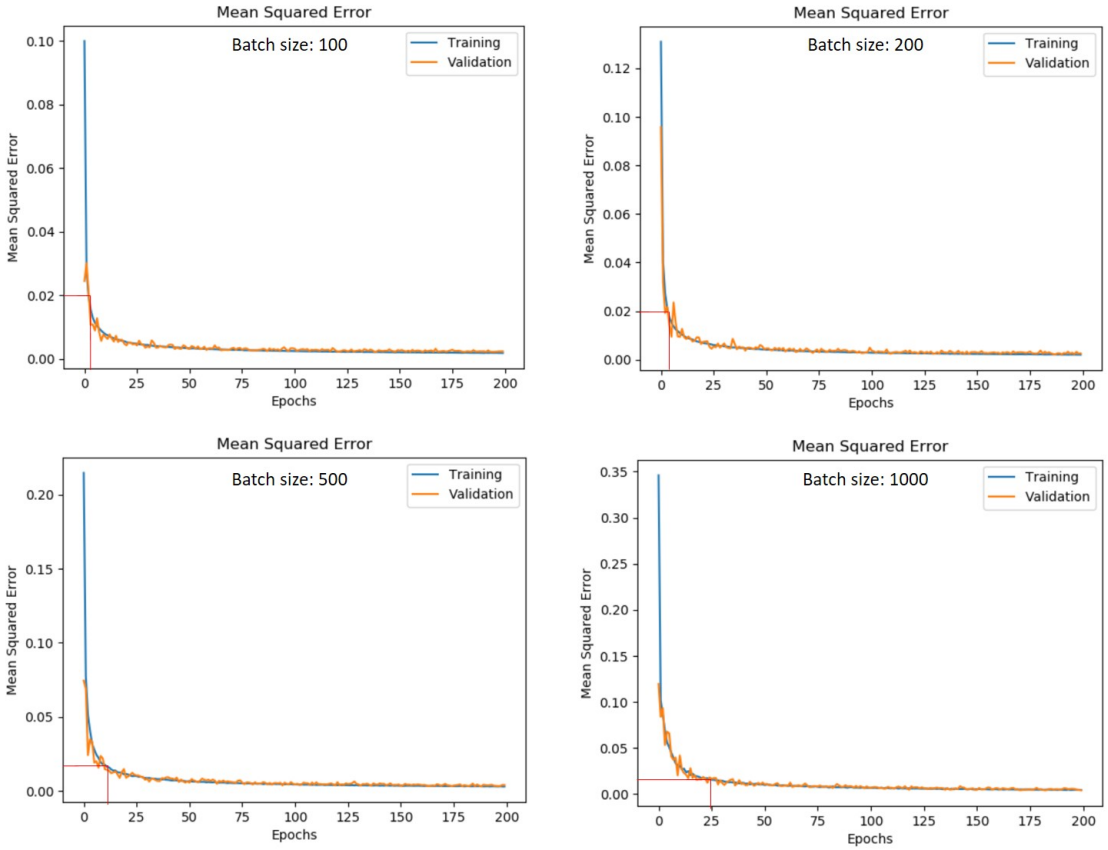

Figure 4-4: Mean squared error as a function of the epochs, while training the ANN using different batch sizes and 16000 microseismic events. The red lines indicate the number of epochs necessary to reach an acceptable error.

Batch Size	Computation Time of the training phase
100	9.50 minutes
200	5.83 minutes
500	3.71 minutes
1000	1.86 minutes

Table 4-4: ANN training computation time for different batch sizes, using 16000 events and 200 epochs.

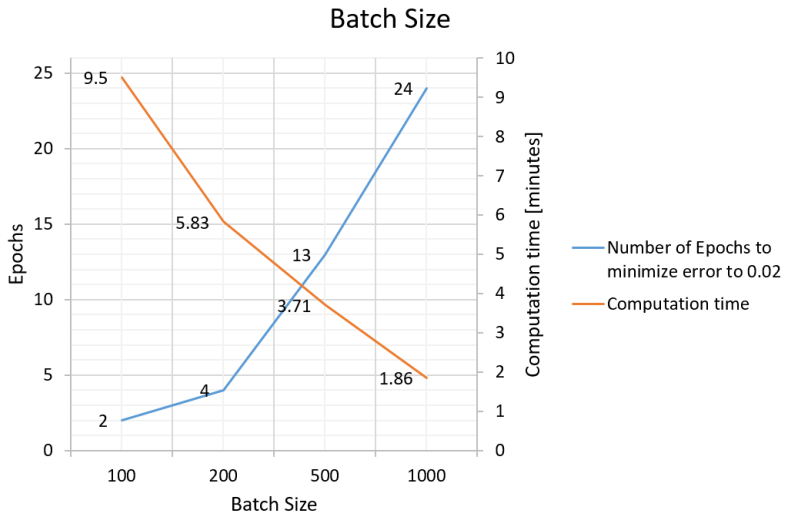

Figure 4-5: Summary of the analysis of the batch size.

Training dataset size	Computation time of the training phase
1000 events	1.52 minutes
8000 events	2.80 minutes
16000 events	6.02 minutes
32000 events	13.06 minutes

Table 4-5: Computation time for the training dataset size experiments.

4-2-5 Training data size

The amount of data necessary to train a neural network is hard to determine. It depends on the complexity of the problem to perform (number of variables and linearity of the problem) and the complexity of the ANN algorithm (Mitchell (1997)).

To study this factor we performed several experiments testing the accuracy of the ANN with different training dataset sizes, keeping the same batch size and number of epochs. The results are presented in Figure 4-6. Analyzing the history of error values per epoch we see that for larger training datasets the number of epochs necessary to minimize the error decreases.

However, an increase in the training dataset also increases the computation time. In Table 4-5 are displayed the time taken to finalize the training in the different experiments for comparison.

In Figure 4-7 a plot summarizing the results for the training dataset size experiments is presented. In the following examples we are using a training dataset size of 16000 microseismic events.

4-2-6 Training summary

After the analysis of the ANN training parameters we adopted the following configuration:

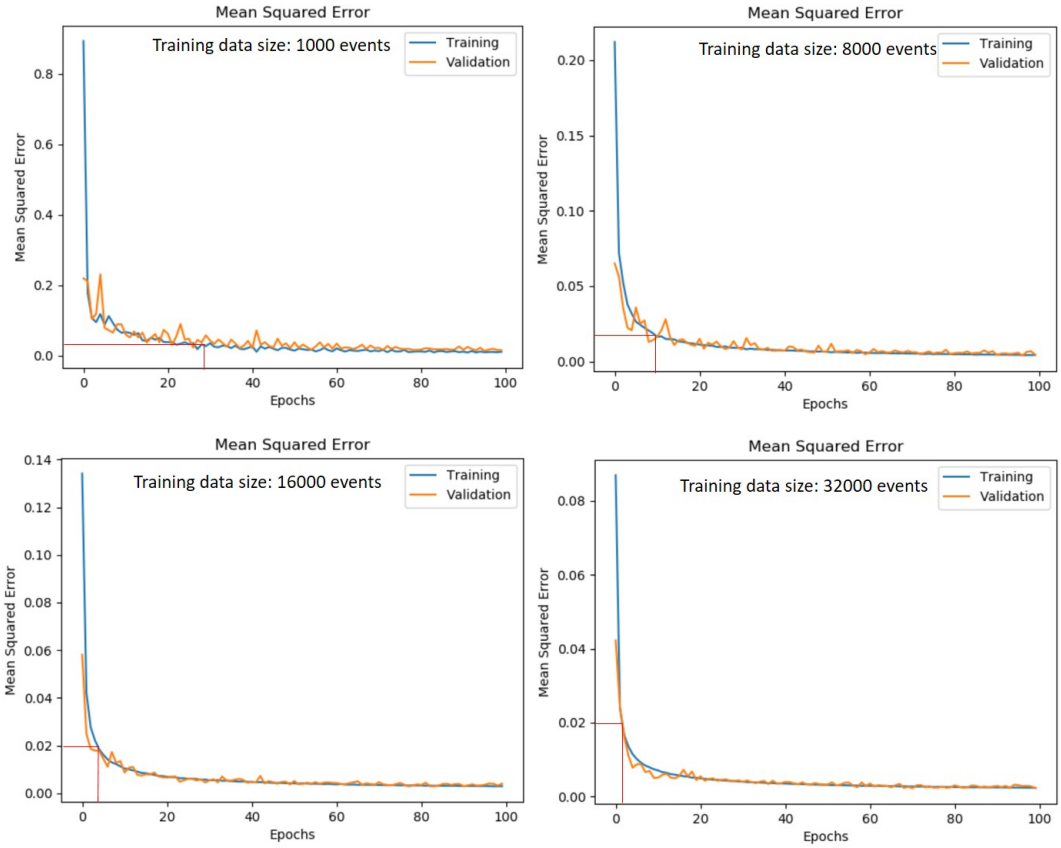

Figure 4-6: Mean squared error values per epoch plots, while testing the training datasize. Red lines indicate number of epochs to reach an acceptable error value.

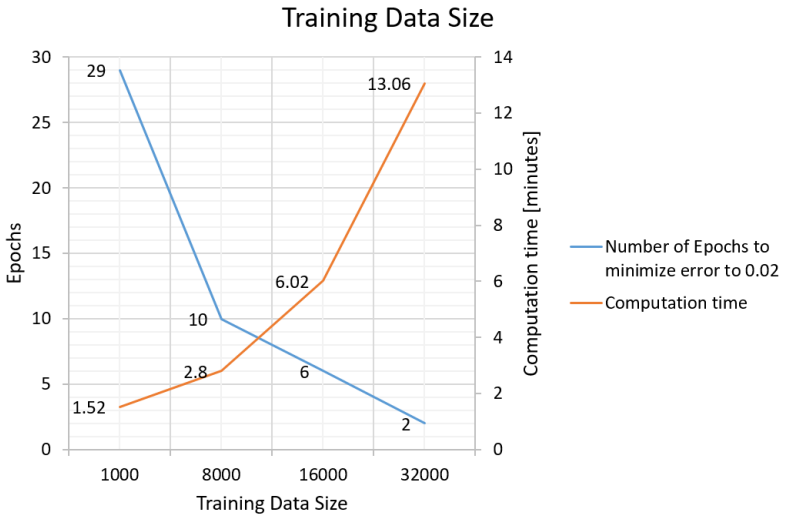

Figure 4-7: Summary of the training dataset size experiments.

- Loss function: mean squared error
- Optimization function: RMSprop
- Batch size: 200
- Training data size: 16000 events

We then trained with these parameters the network described in section 4-1 using data without noise first, and then using data with added Gaussian noise. Finally, we tested the prediction capabilities of the two trained networks as described in the next section.

4-3 ANN Testing

After training, the ANN has been used to predict the moment tensor components from microseismic data not previously used in the training or in the validation phases. The goal is to confirm the actual predictive power of the network.

In order to evaluate the prediction capabilities of the ANN it is important to analyse its robustness. In practice, this means to ensure that the ANN ignores perturbations, noise or erroneous inputs and provides larger weights to the proper input features.

We performed several tests including noise in the inputs. The tests carried out evaluated the accuracy of model while increasing signal-to-noise ratio in the data. These tests are described in the following sections.

4-3-1 ANN trained without noise

A network trained using data without noise was created using the parameters discussed in the previous sections. This network was tested using data with different levels of Gaussian noise: 0%, 1%, 5% and 10%. The results are shown in Figures 4-8, 4-9, 4-11, 4-10 and 4-12.

In Figure 4-8 it is possible to visually compare an example of 10 randomly chosen events, where the moment tensor components are represented as colored squares. The difference between predicted and true moment tensor components are also displayed in colored squares. From top to bottom are presented the test cases:

- Case A: Test using data with 0% added Gaussian noise
- Case B: Test using data with 1% added Gaussian noise
- Case C: Test using data with 5% added Gaussian noise
- Case D: Test using data with 10% added Gaussian noise

In Figure 4-9 the comparison of predicted and true moment tensor components are plotted for all the cases tested. It is possible to observe that increasing the Gaussian noise added to

the data results in a deviation of the predictions of the moment tensor from the true values. Also, at the top of every plot the full test accuracy value is indicated.

The accuracy of the tests is measured using the following root mean squared-error function. If $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted value of the i-th sample, and $\mathbf{y}_i$ is the corresponding true value, then the root mean squared error (E_{RMS}) estimated over is defined as:

$$E_{RMS}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \sqrt{||\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}||^2} \tag{4-10}$$

Furthermore, we computed from the predicted moment-tensors values the fault geometry angles. These predicted fault geometry angles are compared to the true fault angles used in the forward modeling stage. Comparisons of true vs. predicted fault angles are displayed for dip (Figure 4-10), strike (Figure 4-11) and rake (Figure 4-12) for all cases predicted by the network trained using data without noise. We can visually perceive, as was observed in the MT components prediction plot (Figure 4-9), that the predicted values spread out from the perfect prediction line (red line) as we increase the noise in the data. From these results we can claim that the network performance is negatively affected by increasing noise levels in the data.

Moreover, for any moment tensor there are two possible planes that can generate the same focal mechanism. One plane will be the fault surface and the other is called an auxiliary plane. This means that depending on rake and dip, a rotation of the strike by 180° can result in an unchanged moment tensor (Käufl et al. (2015)), meaning an intrinsic non-uniqueness of the fault angles given a single source.

Since there are auxiliary fault planes that can also correspond to the moment tensor solution obtained through the neural network, we can observe secondary bands in the dip, strike and rake plots in Figures 4-10, 4-11, and 4-12. It is not possible to determine the fault plane solution solely from the moment tensor. We would need to add an additional constraint to remove the ambiguity (Cronin (2010)).

4-3-2 ANN trained with noisy data

To increase the robustness of the ANN (ability of the network to ignore erroneous inputs), we trained it including an amount of Gaussian noise in the input layer, as discussed in section 4-1. Three cases were created:

- ANN trained including a layer with 1% added Gaussian noise
- ANN trained including a layer with 5% added Gaussian noise
- ANN trained including a layer with 10% added Gaussian noise

The results in accuracy for the predicted moment-tensor components and predicted fault angles are shown in Figure 4-13. The blue lines indicate the performance for the ANN that was trained using synthetic data without noise. The figure shows the performance of this

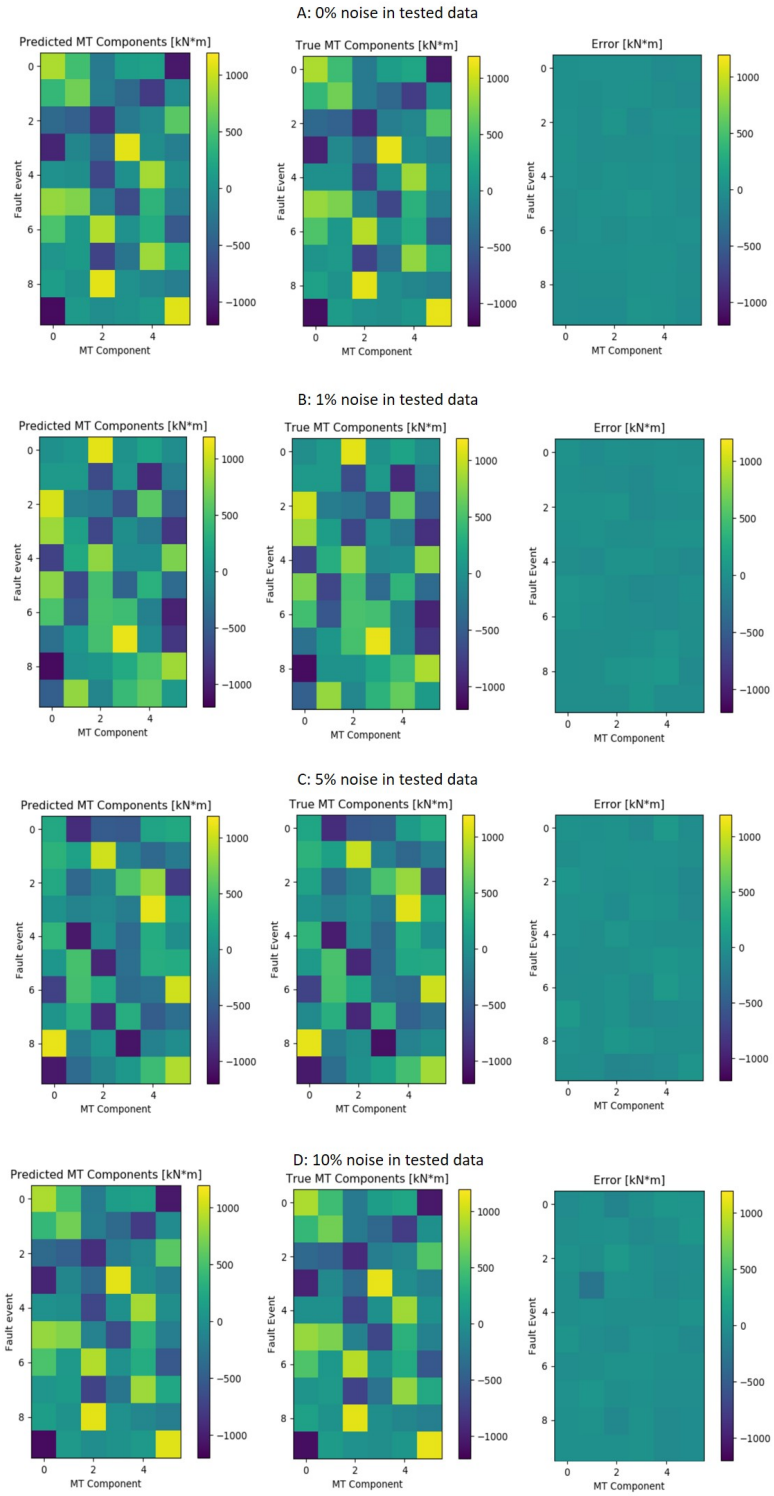

Figure 4-8: Comparison of True and Predicted Values for 10 randomly chosen faulting events. Left column: Predicted MT components. Middle column: True MT components. Right column: Prediction error for each MT component.

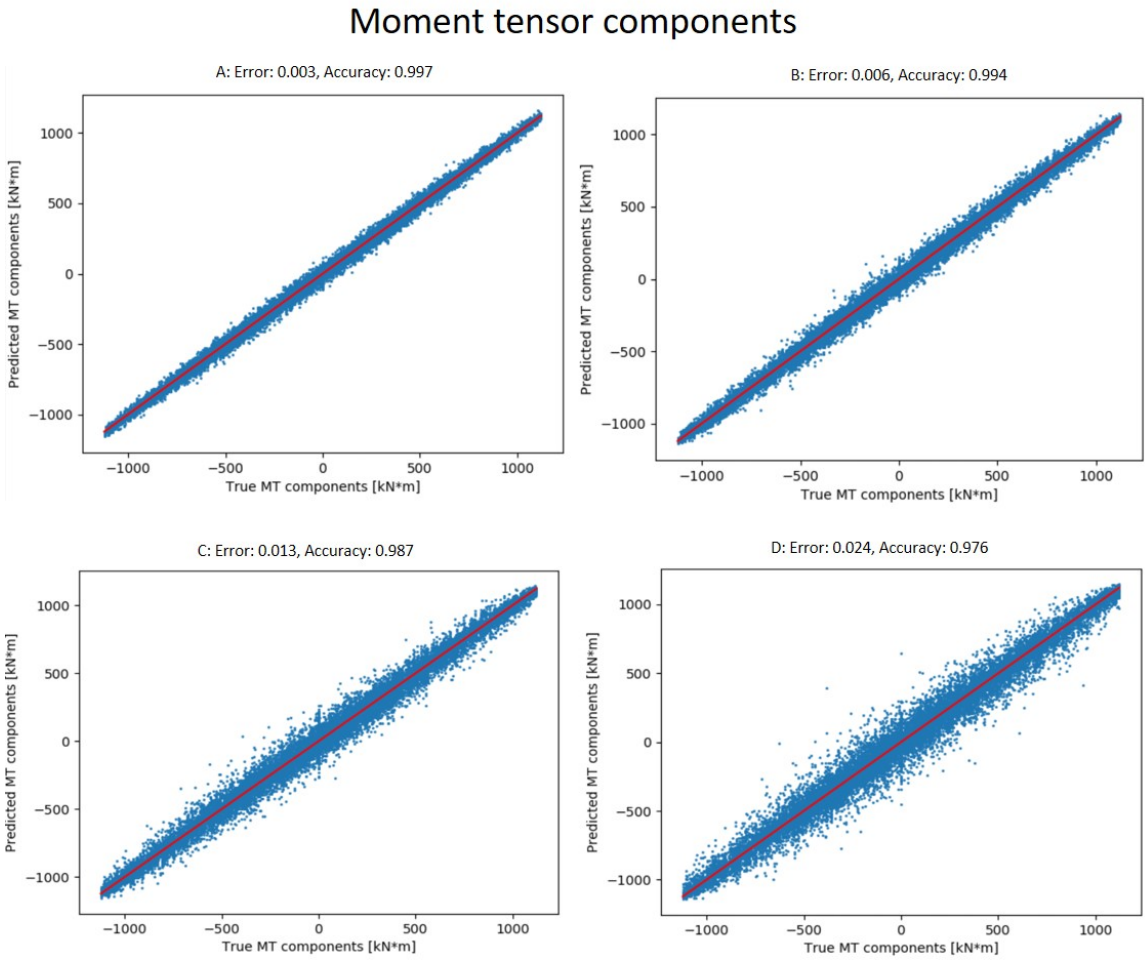

Figure 4-9: True vs Predicted moment-tensor components using a network trained without noise. Red line indicates perfect fit. Case A: 0 % noise in data, Case B: 1 % noise in data, Case C: 5 % noise in data, Case D: 10 % noise in data.

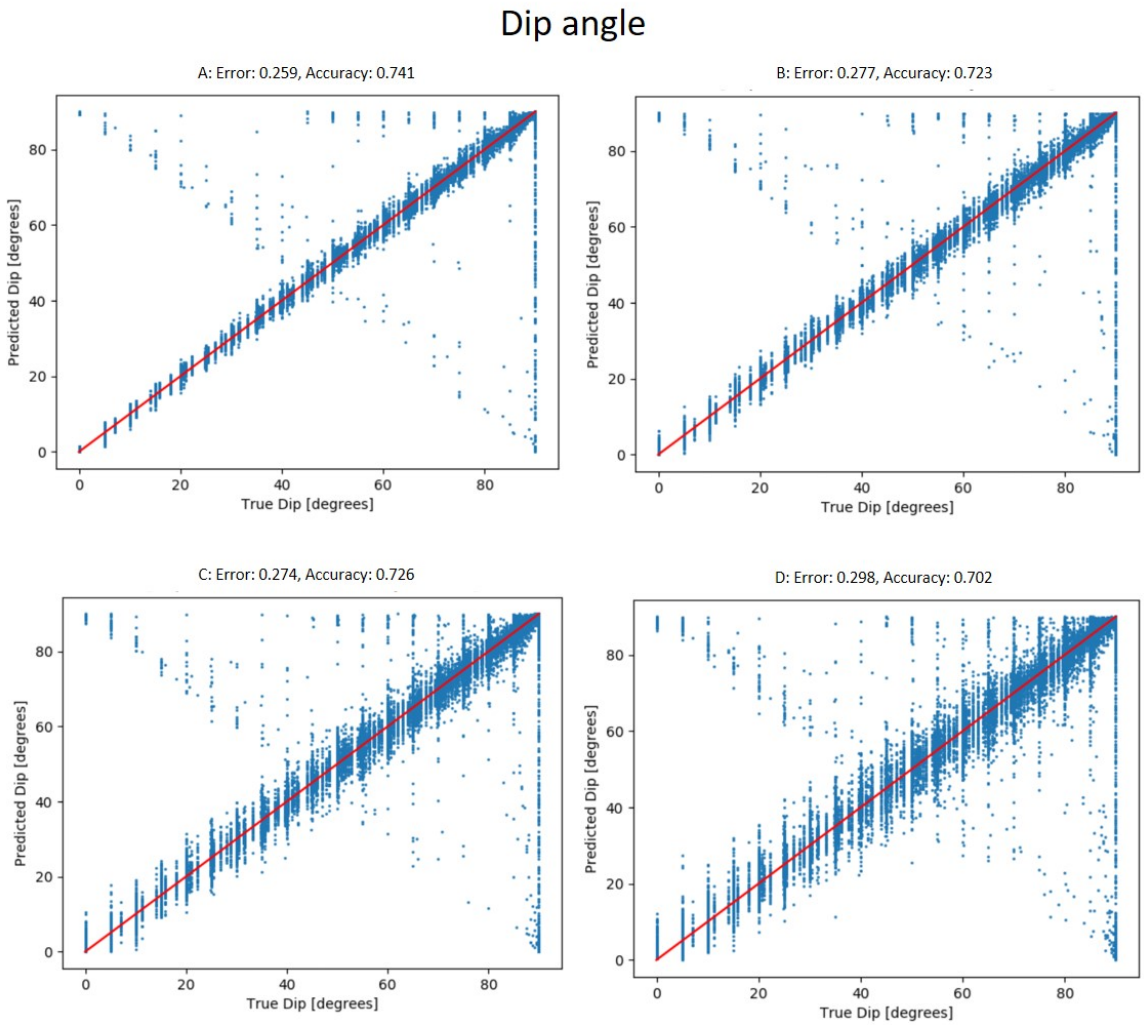

Figure 4-10: True vs Predicted dip-angle values using a network trained without noise. Red line indicates perfect fit. Case A: 0 % noise in data, Case B: 1 % noise in data, Case C: 5 % noise in data, Case D: 10 % noise in data.

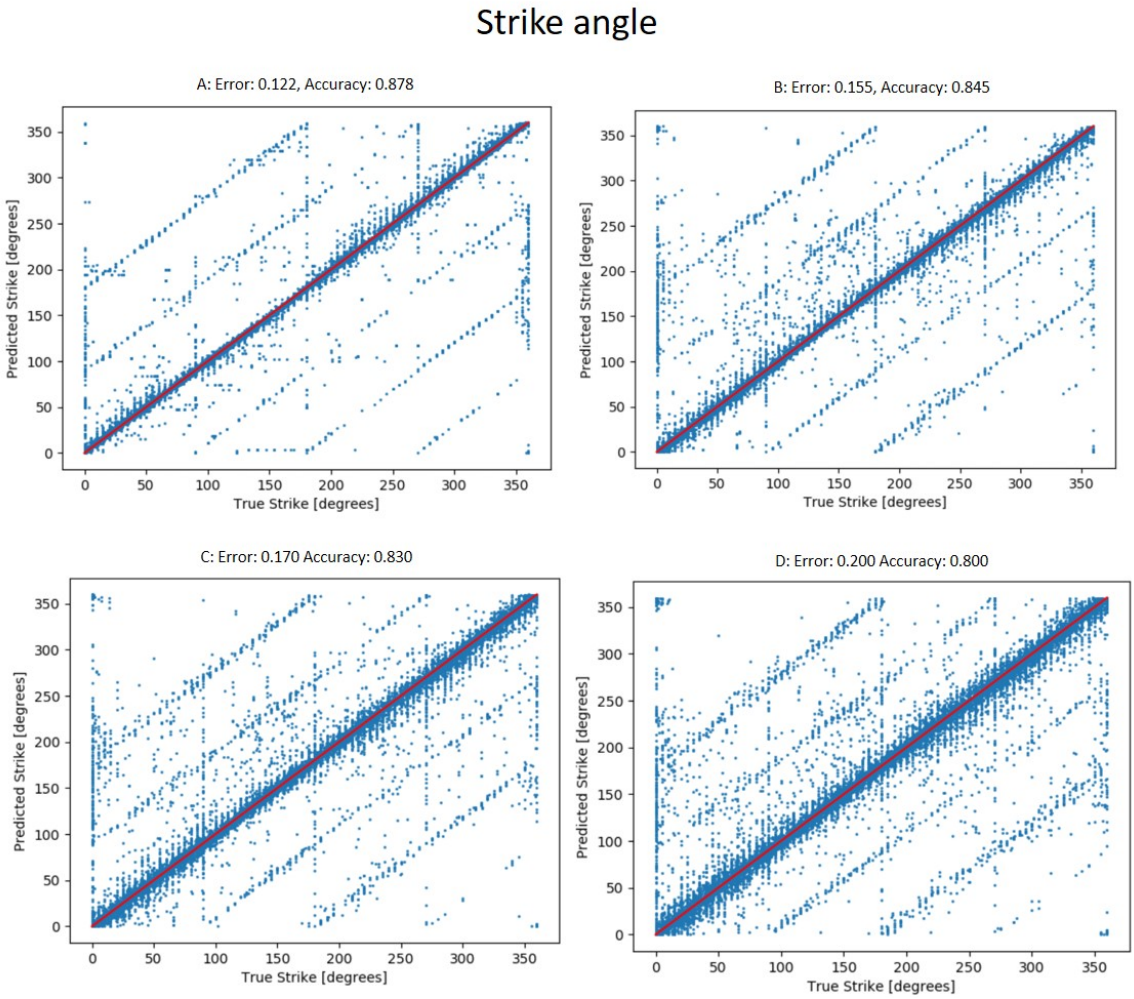

Figure 4-11: True vs Predicted strike-angle values using a network trained without noise. Red line indicates perfect fit. Case A: 0 % noise in data, Case B: 1 % noise in data, Case C: 5 % noise in data, Case D: 10 % noise in data.

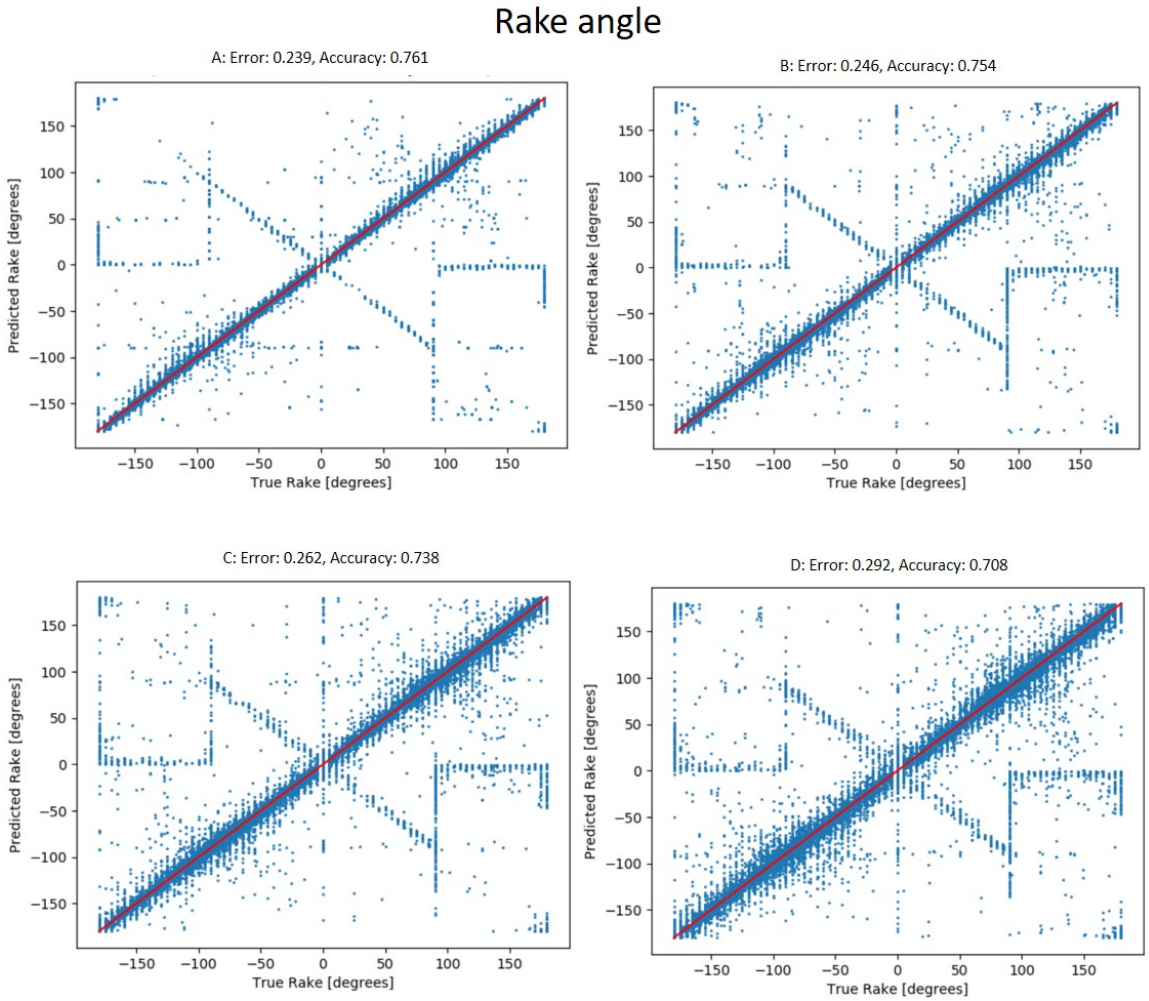

Figure 4-12: True vs Predicted rake-angle values using a network trained without noise. Red line indicates perfect fit. Case A: 0 % noise in data, Case B: 1 % noise in data, Case C: 5 % noise in data, Case D: 10 % noise in data.

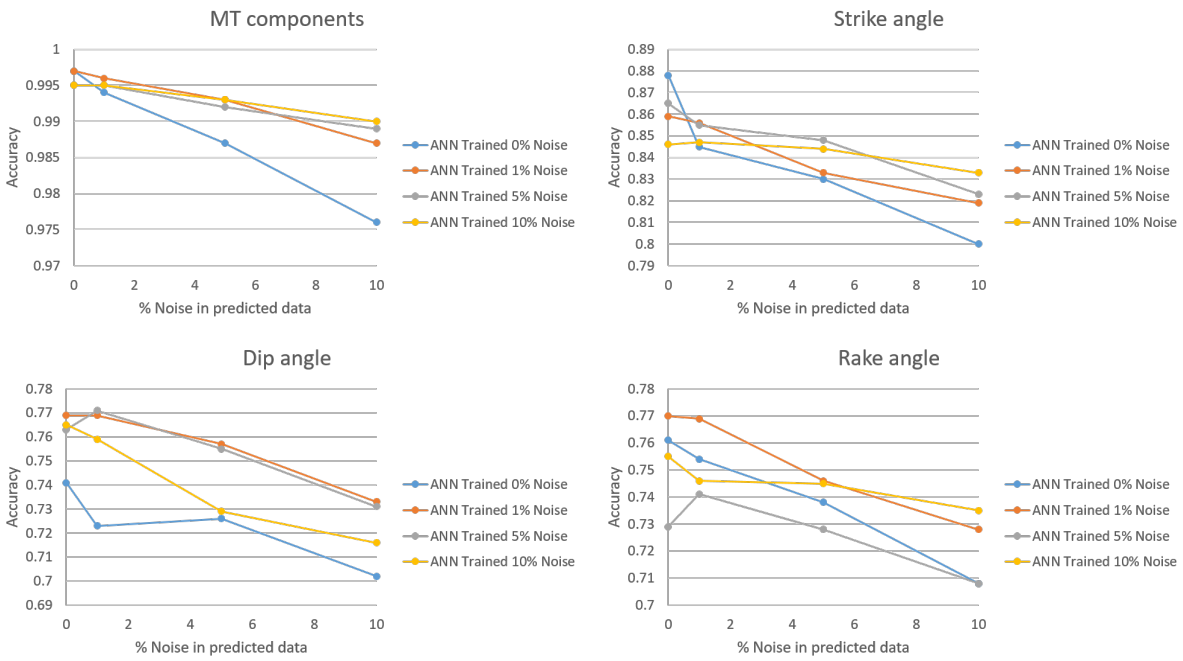

Figure 4-13: Accuracy of predicted moment-tensor components and calculated fault geometry angles while increasing noise in the test dataset. Blue line: results for ANN trained without noise, Orange line: ANN trained with 1% Gaussian noise, Gray line: ANN trained with 5% Gaussian noise, Yellow line: ANN trained with 10% Gaussian noise.

ANN (trained without noise) decreasing for all the predicted variables with increasing noise in the data, meaning the network model is not robust enough.

The results for the network trained including Gaussian noise are also plotted in Figure 4-13. From the plots we can see that for an ANN trained with an certain percentage of noise in the input data the performance is less affected as we increase noise levels in the test dataset, with respect to the performance of the network trained without noise.

The network that shows the best performance is the ANN trained with a 10% Gaussian noise, corresponding to the yellow line in the plots.

Similar conclusions can be obtained by analyzing the previous results with a different visualization tool, commonly used in earthquake seismology, and referred in the literature as "beachball" plots (Cronin (2010)). In our case we can prove that the predicted moment tensor components are accurate enough by visually comparing them with the corresponding true values as shown in Figure 4-14.

The effect of the noise in the predictions can also be visually evaluated as depicted in Figures 4-15, 4-16, 4-17, 4-18. The visual plots for the networks trained without noise and 1 % Gaussian noise show predictions that slightly deviate from the corresponding true values while increasing the noise in the data. On the other side, the networks trained with 5 % and 10 % Gaussian noise show better accuracy even for high level of noise in the data.

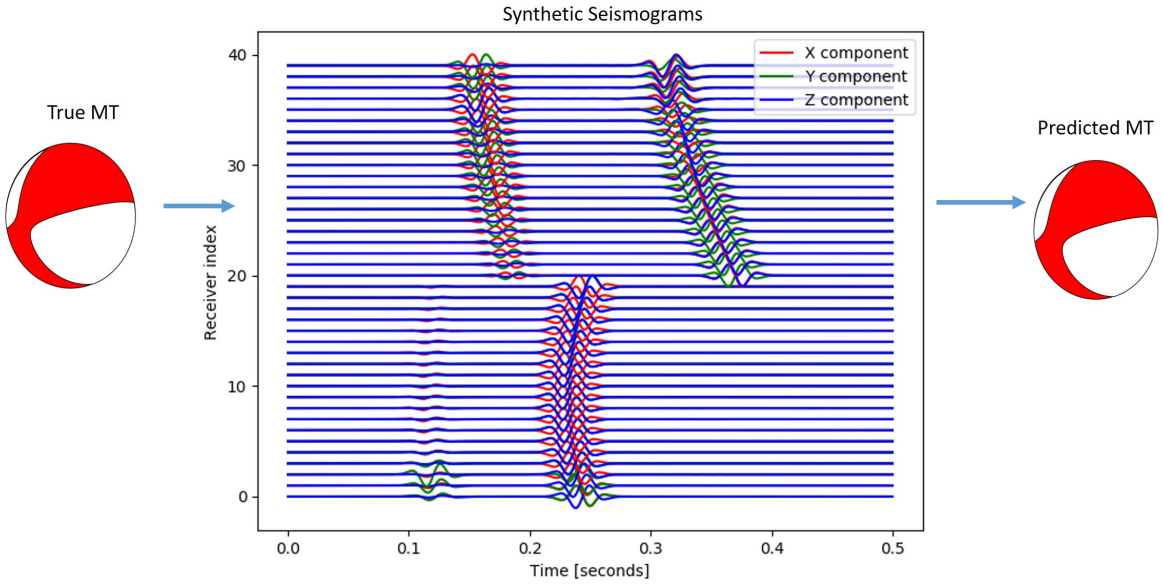

Figure 4-14: Example of visual comparison between true and predicted moment tensor components for a single event.

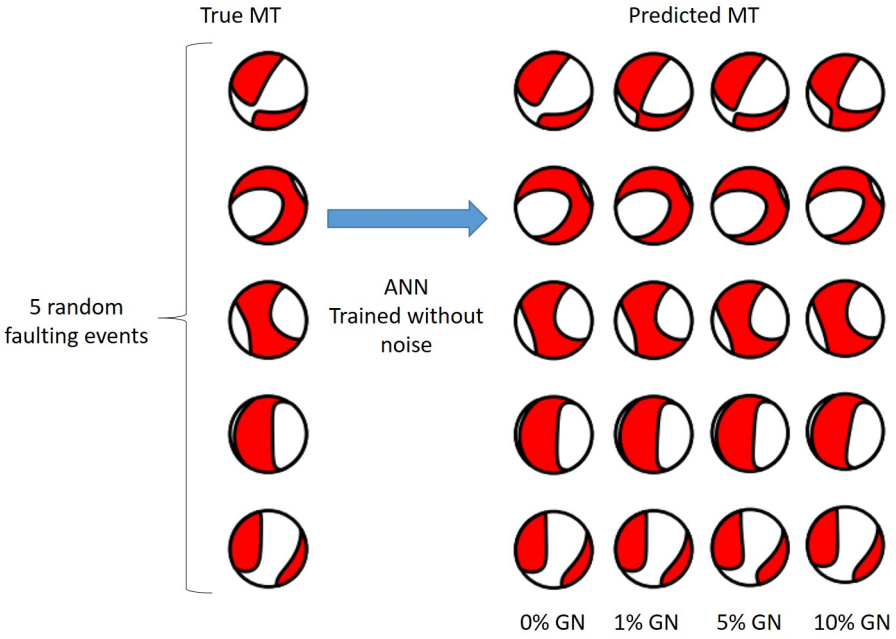

Figure 4-15: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained without noise.

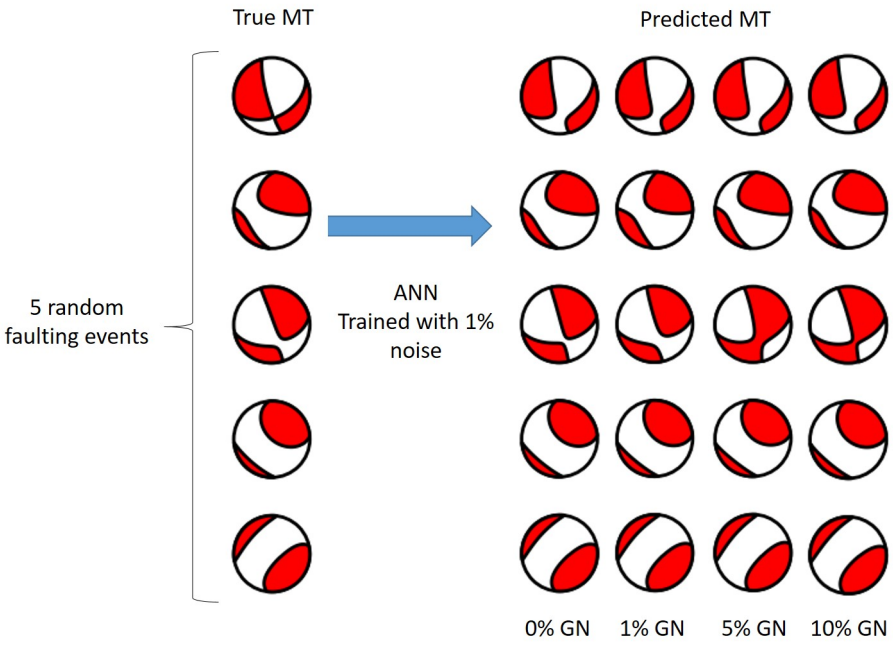

Figure 4-16: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained with 1 % Gaussian noise.

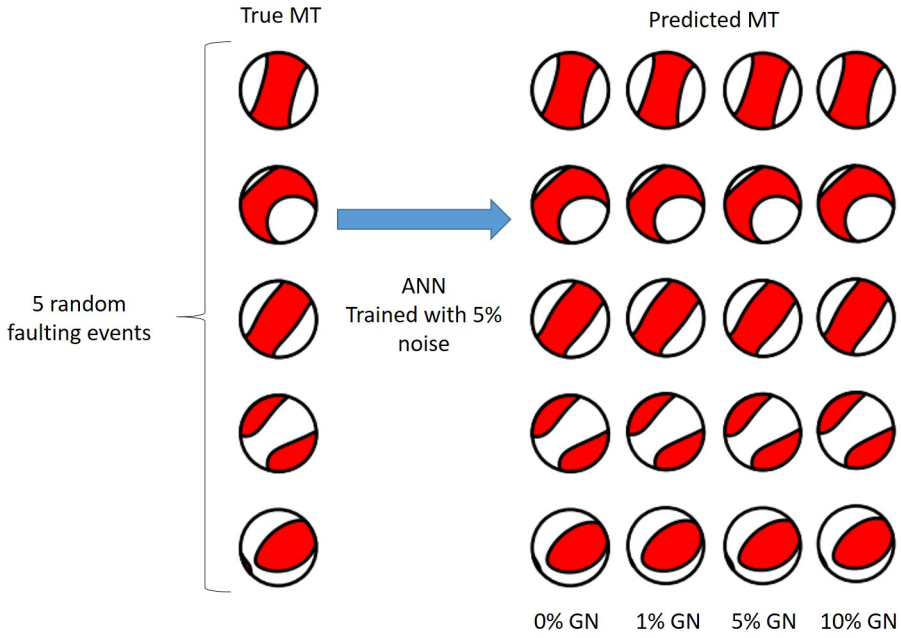

Figure 4-17: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained with 5 % Gaussian noise.

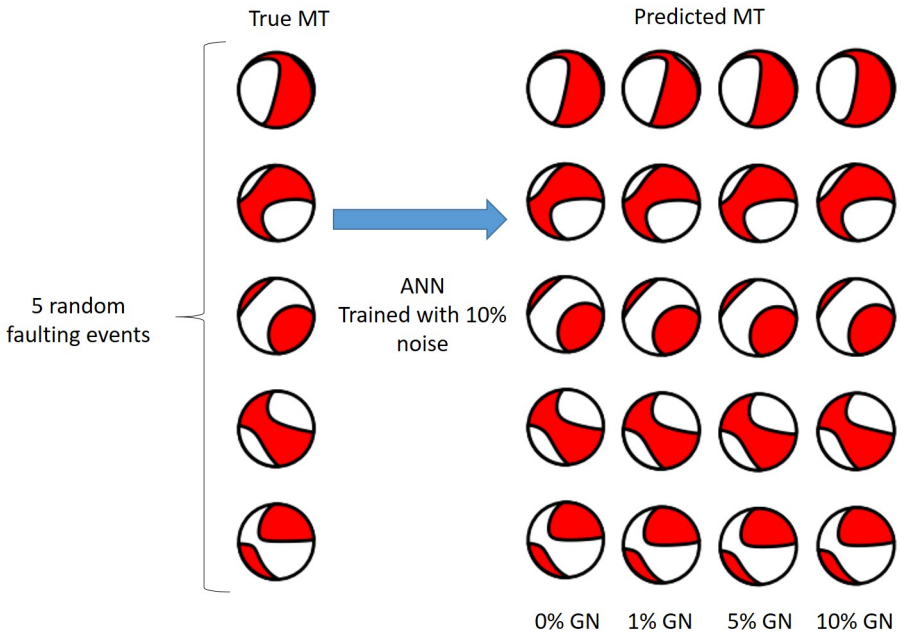

Figure 4-18: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained with 10 % Gaussian noise.

Training Dataset size	Predicted Events size	MT accuracy	Strike accuracy	Dip accuracy	Rake accuracy
8000	8000	0.995	0.864	0.758	0.734
16000	16000	0.997	0.878	0.741	0.761
32000	32000	0.998	0.867	0.722	0.751

Table 4-6: Accuracy values for predicted MT components and fault angles.

4-3-3 Additional tests on training data size

As supplementary experiments, we tested the ANN performance for network models trained with 8000, 16000 and 32000 events (without added Gaussian noise). The measured accuracy values for the different tests are presented in Table 4-6. The results show that for a larger training dataset size the accuracy of the predicted moment-tensor components increases. The better accuracy results for the predicted fault angles correspond to a network trained with a dataset size of 16000 events.

4-4 Receiver geometry effects on the ANN performance

Using the same neural network architecture and training parameters defined in Subsection 4-2-6 and an 10% gaussian noise layer, we trained the network using only the seismograms from the vertical array of receivers (3 components) shown in Figure 3-9. Then, we evaluated the prediction power of this network as in the previous section. A predicted moment tensor

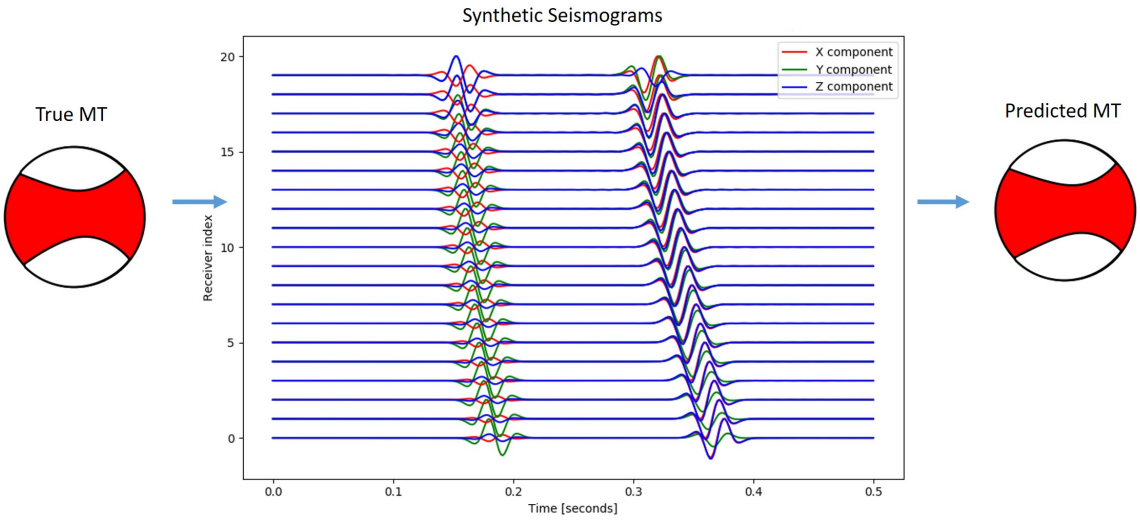

Figure 4-19: Example of visual comparison between true and predicted moment tensor components for a single event using only the vertical array

example is shown in Figure 4-19. The prediction results are presented for the moment-tensor components and computed fault angles: dip, strike and rake in Figure 4-20. We observe that for all the variables evaluated the values start to spread further from the perfect fit line with increasing noise percentages in the tested data. Comparing these figures with the ones corresponding to the combined array (4-9, 4-10, 4-11, 4-11) we can visually recognize that the vertical array prediction performance is inferior compared to the combined array, since it is clear that the predicted values are reaching further away from the perfect fit line.

We also repeated this procedure using only the seismograms of the horizontal array of 20 receivers (3 components) shown in Figure 3-9. A predicted moment tensor example is shown in Figure 4-21. The prediction results are presented for the moment-tensor components, and computed fault angles as well in Figure 4-22. The plots also depicts results with an inferior prediction performance to those of the combined array. An analysis of the overall accuracy of the networks is explained in the following paragraphs.

We analyzed the prediction power of the network using the combined array (using both horizontal and vertical arrays), the vertical array and the horizontal array for increasing noise percentages. The results are presented in Figure 4-23. From this graphical analysis it is clear that the network trained using the combined array shows higher accuracy values for all the evaluated variables, than the networks trained using only the vertical or the horizontal receivers.

These differences in accuracy values for the three receiver geometries tested could be due to several reasons. The most direct reason might be the fact that the combined array contains 40 receivers, instead of 20 as for the vertical and horizontal arrays, which means there are more seismograms evaluated per event in the combined array case.

However, the horizontal and vertical arrays contain equal number of receivers (meaning an equal number of seismograms evaluated per event), yet their accuracy values are quite different

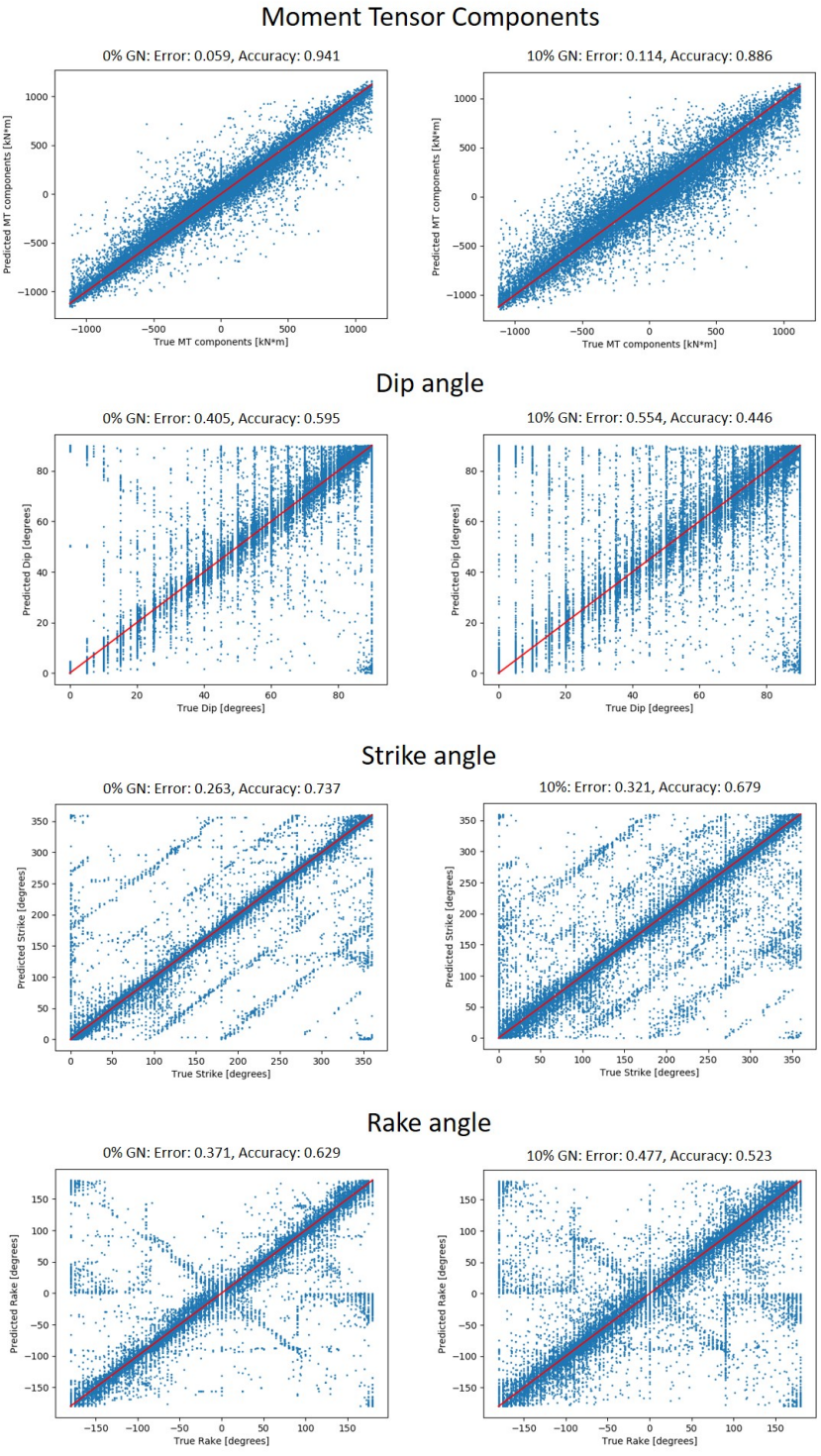

Figure 4-20: True vs Predicted scatter plots for moment tensor components and fault angles using a network trained with a vertical array of receivers. Red line indicates perfect fit. Left: 0% Gaussian noise in tested data, Right: 10% Gaussian noise in tested data

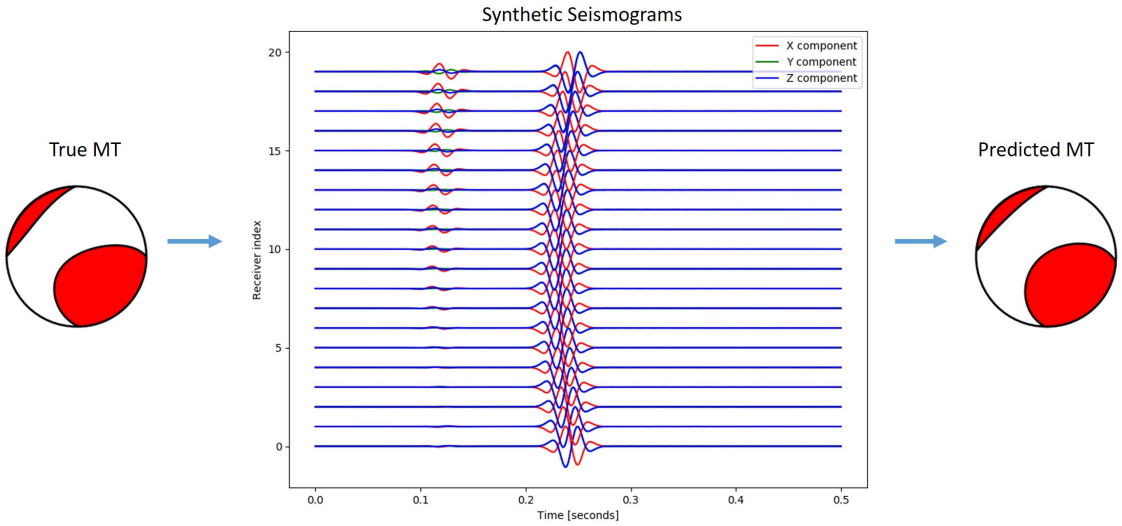

Figure 4-21: Example of visual comparison between true and predicted moment tensor components for a single event using only the horizontal array

for all the variables. The accuracy values are higher for the horizontal array in all the variables. This could be due to the differences in the geometry coverage from both arrays with respect to the source radiation. In Figure 3-9 the horizontal receivers are all below the source, with 10 receivers located eastward of the source and 10 receivers westward of the source. The vertical receivers are all located upwards and westward of the source, perhaps resulting in an lower coverage of the wave radiation pattern of the microseismic event.

Furthermore, the combined array accuracy values could also be the consequence of a better wave radiation pattern coverage, including the receivers in both the vertical and horizontal configurations presented. These results arise new questions on how the position of the receivers with respect to the source affect the prediction performance, pointing to more evaluations with respect to the geometry of the arrays and the number of receivers.

Finally, from Figure 4-23 we can also assume that using the combined array for training provides more robust predictions, since the accuracy values slightly decrease with increasing noise percentages in the tested data. Contrarily, using only the vertical array or the horizontal array, the results show a larger decrease, almost in a parallel manner with increasing noise percentages.

In the same manner of the previous section, we also present the effect of the noise in the predictions visually using the beachball plots as depicted in Figures 4-24 and 4-25. In these plots we can identify how the beachballs are increasingly describing imprecise sources while increasing noise in the tested data. The resulting plots also confirm the fact that the ANN trained using the combined array has a superior predicting capability when comparing with Figure 4-18.

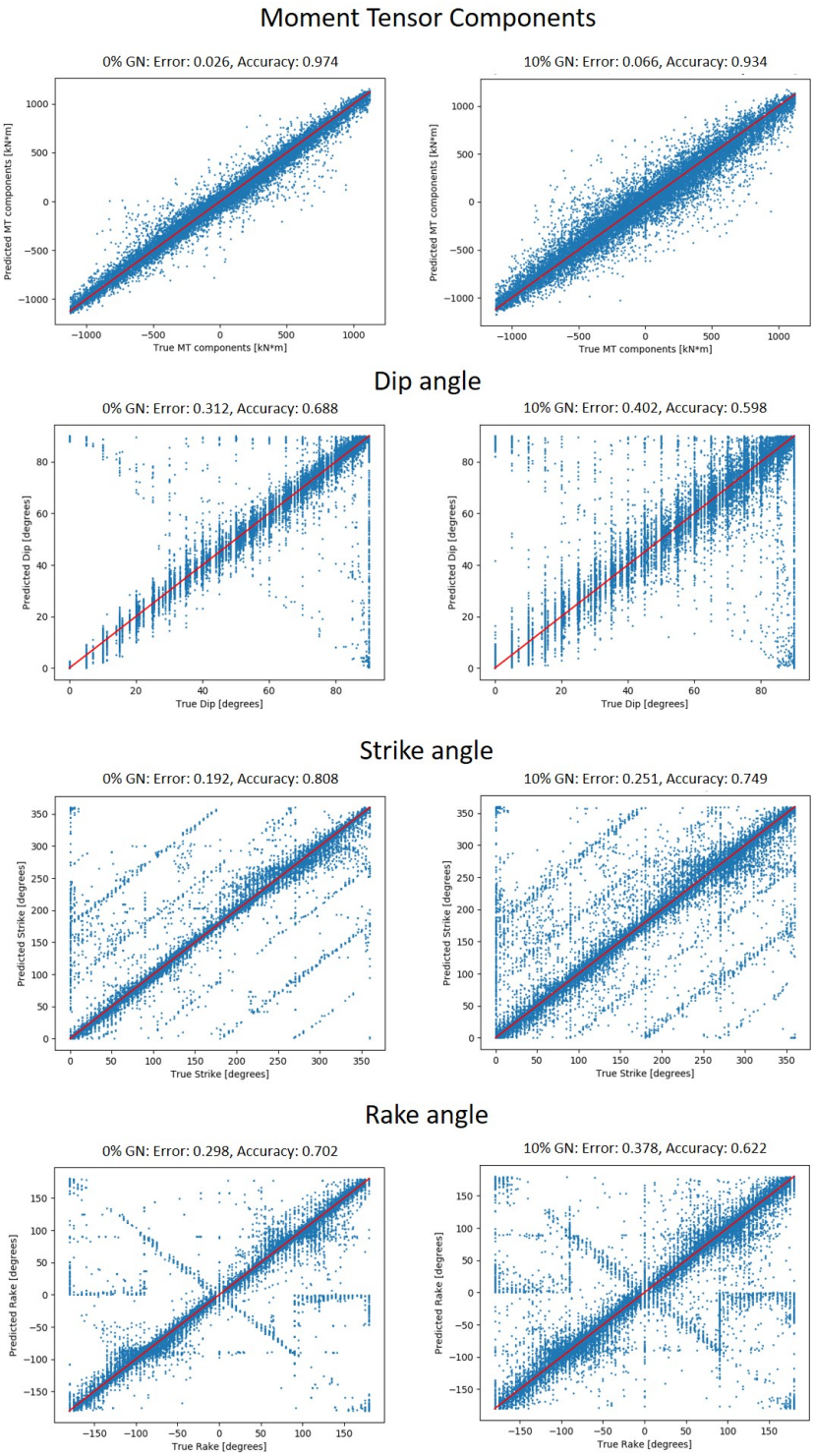

Figure 4-22: True vs Predicted scatter plots for moment tensor components and fault angles using a network trained with a horizontal array of receivers. Red line indicates perfect fit. Left: 0% noise in tested data, Right: 10% Gaussian noise in tested data.

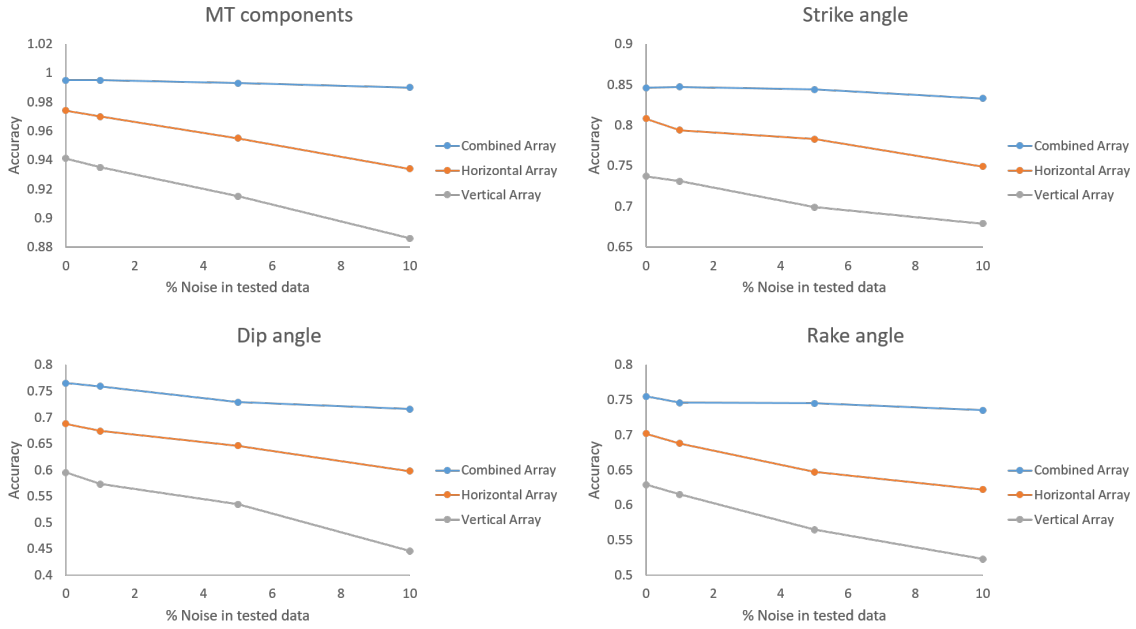

Figure 4-23: Accuracy of predicted moment-tensor components and calculated fault geometry angles while increasing noise in the test dataset. Blue line: results for ANN trained using combined array, Orange line: ANN trained using horizontal array, Gray line: ANN trained using vertical array

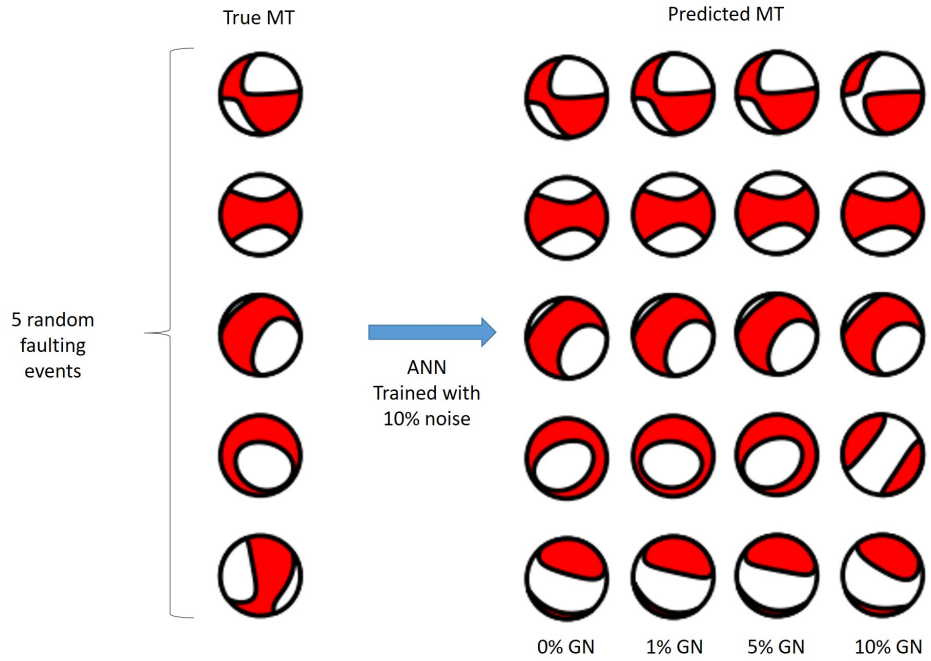

Figure 4-24: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained using a vertical array of 20 receivers

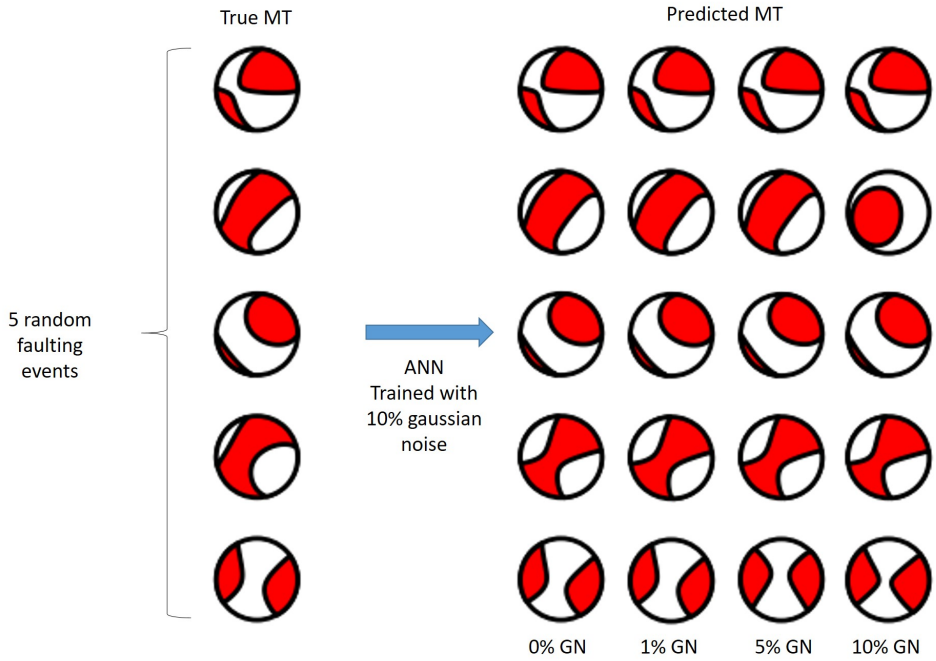

Figure 4-25: Moment-tensor beachball plots for five random faulting events: true values (left) vs. predicted values different noise levels in the data (right). The network was trained using a horizontal array of 20 receivers

Chapter 5

Conclusions and Discussion

Microseismic monitoring is an essential procedure in the development of unconventional resources, compaction/subsidence, reservoir surveillance, CO_2 injection, among other applications. Currently, this task is done using microseismic borehole data to evaluate the distribution of seismic source locations and in some cases infer the fracture extension. However, microseismic measurements can be analysed to invert for the seismic moment tensor, which describes the fracture mechanisms in greater detail.

In this thesis we propose to carry out the inversion of microseismic borehole measurements using neural networks. This approach has several clear advantages: we can solve ill-posed problems so that small aperture acquisition data can be used, neural networks are quite robust to noise when trained with proper noisy datasets, and results can be obtained in a fast turnaround time.

To solve the inverse problem, we need a high amount of data to train the neural network properly. For this purpose we used an algorithm based on the discrete wavenumber method to generate full waveforms from known moment tensor sources. We created a dataset sampling 101,250 possible moment tensors, generating thus 12,150,000 waveforms corresponding to 3 components $\times$ 40 receivers in a deviated borehole array (vertical and horizontal receivers).

Subsequently, a neural network was designed with the purpose of using the synthetic seismograms previously generated to obtain the six independent components of the moment tensor. The network was trained using the previously generated synthetic dataset. Several training options and network design parameters, such as the loss function and batch size were tested manually to boost the learning process. Additionally, the network was also trained using a Gaussian noise input layer to improve its performance in the presence of noisy data.

Finally, when the network was properly trained we proceeded to test its predictive power using synthetic seismograms with different levels of Gaussian noise. The prediction outcomes for the moment-tensor components is outstanding for data without noise. From the results of noisy data it is clear that using Gaussian noise in the training phase distinctly improves the robustness of the network and its predictions for noisy synthetic seismograms.

We succesfully trained a fully connected feed-forward neural network to predict moment-tensor components from synthetic seismograms in simulated microseismic monitoring borehole arrays. This neural network is able to produce an approximation of the seismic source using a set of synthetic seismograms, even in the presence of Gaussian noise.

Additional evaluations of the neural network using just a horizontal or a vertical array resulted in inferior predicting performance with respect to the use of combining horizontal and vertical arrays. The analysis also shows a faster decline in accuracy values with increasing noise levels in the tested data when compared to the network of the combined array. The outcome of this geometry analysis arises new considerations to be investigated in extensions of this research study.

5-1 Future developments

The encouraging results presented in the previous chapters of this thesis make a number of possible extensions of the method worthy of research in the future.

Further developments primarily include the application to datasets generated with different (inhomogeneous) velocity models and varied acquisition geometries. In particular, we suggest to extend the evaluation regarding the position of the arrays with respect to the source, their aperture and the number of receivers. The effect of the source-receiver offset on the predictions should be evaluated as well. In general, a reduction of the accuracy is expected when reducing the arrays aperture or when increasing the offset.

The experiments performed in this thesis project used a 100% double-couple seismic events. In future experiments the network architecture designed in this project could be trained and tested for the prediction of moment tensor components related to non double-couple sources.

Regarding the neural network design choices, changes in the architecture parameters such as the number of neurons and hidden layers, the activation functions and the connections types, could be also considered. This represents an exhaustive task since it has to be done by trial-and-error and there is not an exact procedure to follow.

Another possible option is to evaluate different optimization functions. There are numerous methods for optimization available in machine learning which could lead to different outcomes in terms of the prediction power of the network.

Moreover, an interesting future extension to this project could be done by additionally training the network to predict other variables important to microseismic monitoring operations, such as the magnitude of the microseismic event, source location and fault orientations.

Finally, the proposed approach has to be applied to field datasets. If proven successful, this could have a significant impact in borehole microseismic monitoring, reducing the processing workload and delivering results in near real-time, allowing the field operators to take quick decisions during the completion operations.

Appendix A

Artificial Neural Networks

The term "neural network" has its origins in attempts to find mathematical representations of information processing in biological systems (McCulloch and Pitts (1943)). One of the key mechanisms for information processing in a human brain is the way that information is processed by means of the connections (called synapses), of a large number of structurally simple elements (called neurons). Artificial neural networks (ANNs) are a family of models that mimic the structure of the neural system and learn patterns in observations (Suk (2017)).

The simplest artificial neural model is known as perceptron (Rosenblatt (1958)). It is structured with input visible units $\{v_i\}_{i=1}^{D}$, trainable connection weights $\{w_i\}_{i=1}^{D}$, a bias w_0 and an output unit y (See Equation A-1) (Suk (2017)). The observations are contained in vector $\mathbf{v} \in \mathbb{R}^D$, the value of the output unit y is obtained from an activation function $f(\cdot)$ by taking the weighted sum of the inputs as in Equation A-1 (Bishop (2006)).

$$y(\mathbf{v};\theta) = f\left(\sum_{i=1}^{D} v_i w_i + w_0\right) = f(\mathbf{w}^\intercal \mathbf{v} + w_0) \tag{A-1}$$

where $\theta = \{\mathbf{w}, w_0\}$ denotes the parameter set, $\mathbf{w} = \{w_i\}_{i=1}^{D} \in \mathbb{R}^D$ is the connection weight vector, and w_0 is a bias.

Some popular activation functions are the ReLu activation function and the sigmoid function. A typical example of sigmoid activation function is the hyperbolic tangent function defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{A-2}$$

where the input x is transformed into an output limited in the range $[-1, 1]$.

A basic architecture consists of three types of neuron layers: input, hidden, and output layers (See Figure A-1). In feed-forward networks, the signal flow is from input to output units,

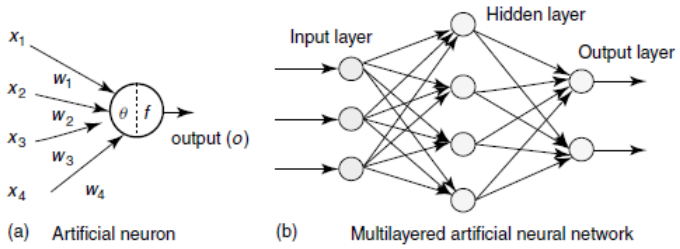

Figure A-1: Architecture of an artificial neuron and a multilayered neural network (Abraham (2005))

strictly in a feed-forward direction. The data processing can extend over multiple layers of units, but no feedback connections are present. A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods exist to set the strengths of the connections. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule (Abraham (2005)).

For an ANN the learning process starts with random values of the weights to break the symmetry of back-propagation. The goal of training is to minimize the error between true output and predicted output by the network. Given multiple samples, the ANN adjusts weights between layers in successive iterations. At the end, the trained network is just a set of matrices whose elements were tuned to be able to solve a specific problem (Ovcharenko et al. (2018)).

The neural network architecture plays a crucial role in the reconstruction of desired output parameters. The number of hidden layers, amount of neurons in each of them, weight initialization and type of activation function all have an important influence on the performance of a neural network. An optimal layout of hidden layers is usually found by trial-and-error techniques (Maity et al. (2014), Binder (2018), Ovcharenko et al. (2018)).

ANNs have proven to be a powerful tool in pattern recognition applications in geosciences. Some examples are shown in: Röth and Tarantola (1994), Moya and Irikura (2010), Binder (2018), Maity et al. (2014), Ovcharenko et al. (2018), Akram et al. (2017).

Appendix B

Calculation of seismograms using a modified DWN method

To compute the seismic waveforms corresponding to a known moment tensor source we follow the steps explained in the next sections, as mentioned in Cotton and Coutant (1996).

B-1 Calculation of potentials at the source layer

From the expressions giving the potentials radiated by single forces (Bouchon and Aki (1977)) the upward and downward potentials radiated at the source layer for the wave modes P, SV and SH are derived. Since we have three wave modes and upward and downward radiation, we will have six different source potentials $S_1(z)$, $S_2(z)$, $S_3(z)$, $S_4(z)$, $S_5(z)$, and $S_6(z)$ whose expressions are reported in the equations of Table B-1.

P-wave and S-wave vertical wavenumbers of the source potentials are respectively given by:

$$\gamma = \sqrt{k_\alpha^2 - k_r^2} \tag{B-1}$$

Wave mode	Vertical wavenumber	Direction	Source potential
P	γ	Upward	$S_1(z) = e^{-i\gamma\|z-z_0\|}$
P	γ	Downward	$S_2(z) = sign(z-z_0)e^{i\gamma\|z-z_0\|}$
SV	ν	Upward	$S_3(z) = e^{-i\nu\|z-z_0\|}$
SV	ν	Downward	$S_4(z) = sign(z-z_0)e^{i\nu\|z-z_0\|}$
SH	ν	Upward	$S_5(z) = e^{-i\nu\|z-z_0\|}$
SH	ν	Downward	$S_6(z) = sign(z-z_0)e^{i\nu\|z-z_0\|}$

Table B-1: Source potentials associated with the three wave modes and upward/downward radiation.

and

$$\nu = \sqrt{k_\beta^2 - k_r^2} \tag{B-2}$$

with $\mathrm{Im}(\gamma) < 0$ and $\mathrm{Im}(\nu) < 0$, $k_\alpha = \frac{\omega}{\alpha}$ and $k_\beta = \frac{\omega}{\beta}$ the P and S wavenumbers (corresponding to the P-wave and S-wave velocities, α and β), $k_r = \frac{2\pi n}{L}$ the horizontal wavenumber, L is the spatial period, and z_0 the source depth.

We can then decompose the potentials Φ, Ψ and χ for the six moment tensor components corresponding to the elementary moment tensors in 3-9, as derived by (Cotton and Coutant (1996)), using the source definitions in Table B-1:

$$\begin{aligned}
M_{xy}:&\Phi = C_f F \sin 2\theta cs2 K_1 S_1 & M_{yz}:&\Phi = C_f F\ 2\sin\theta(-k_r^2) J_1 S_2\\
&\Psi = C_f F \sin 2\theta K_1 S_4 & &\Psi = C_f F sin\theta cs9 J_1 S_3\\
&\chi = -C_f F \cos 2\theta cs3 K_1 S_5 & &\chi = C_f F \cos\theta k_\beta^2 J_1 S_6\\
M_{yy}:&\Phi = C_f F cs2(\sin^2\theta K_1 + K_2) S_1 & M_{xz}:&\Phi = C_f F\ 2\cos\theta(-k_r^2) J_1 S_2\\
&\Psi = C_f F(\sin^2\theta K_1 + K_2) S_4 & &\Psi = C_f F \cos\theta cs9 J_1 S_3\\
&\chi = -C_f \cos\theta \sin\theta cs3 K_1 S_5 & &\chi = C_f F \sin\theta(-k_\beta^2) J_1 S_6\\
M_{xx}:&\Phi = C_f F cs2(\cos^2\theta K_1 + K_2) S_1 & M_{zz}:&\Phi = C_f F(-i\nu k_r) J_0 S_1\\
&\Psi = C_f F(\cos^2\theta K_1 + K_2) S_4 & &\Psi = -C_f F J_0 S_4\\
&\chi = C_f F \cos\theta \sin\theta cs3 K_1 S_5 & &\chi = 0
\end{aligned} \tag{B-3}$$

where $J_i = J_i(k_r, r)$ is the Bessel function of order i and the other variables are defined as:

$$\begin{aligned}
C_f &= \frac{1}{2\rho\omega^2 L} & K_0 &= k_r J_0\\
K_1 &= k_r J_0 - 2J_1/r & K_2 &= J_1/r\\
K_3 &= \frac{\partial K_1}{\partial r} = -\frac{2}{r}K_1 - {k_r}^2 J_1 & K_4 &= \frac{\partial K_2}{\partial r} = \frac{K_1}{r}\\
K_5 &= \frac{\partial J_1}{\partial r} = k_r J_0 - J_1/r & &\\
K_6 &= \frac{\partial K_3}{\partial r} = -{k_r}^2 K_5 + \frac{2}{r^2}K_1 - \frac{2}{r}K_3 & &\\
K_7 &= \frac{\partial K_4}{\partial r} = -\frac{3K_1}{r^2} - {k_r}^2 K_2 = \frac{K_3 - K_4}{r} & K_8 &= \frac{\partial K_5}{\partial r} = K_3 + \frac{K_1}{r}\\
K_9 &= \frac{K_3 - 4K_4}{r} & K_{10} &= \frac{4K_4 - K_3}{r} + K_6 = K_6 - K_9\\
K_{11} &= \frac{K_3}{r} & K_{12} &= \frac{K_4}{r}\\
K_{13} &= \frac{K_8}{r} & cs2 &= \frac{{k_r}^2}{i\nu}\\
cs3 &= \frac{{k_\beta}^2}{i\gamma} & cs9 &= \frac{{k_\beta}^2 - 2{k_r}^2}{i\gamma}
\end{aligned} \tag{B-4}$$

B-2 Propagation of potentials

Afterwards, we propagate each of the six source potentials from the source layer through the model: three modes Φ, Ψ and χ with upgoing and downgoing components (including possible conversions of the P and SV waves) are reflected and transmitted at the interfaces, following the scheme (from Cotton and Coutant (1996)):

$$
\begin{aligned}
S_1 &\rightarrow S_1^{\Phi up}, S_1^{\Phi down}, S_1^{\Psi up}, S_1^{\Psi down} \\
S_2 &\rightarrow S_2^{\Phi up}, S_2^{\Phi down}, S_2^{\Psi up}, S_2^{\Psi down} \\
S_3 &\rightarrow S_3^{\Phi up}, S_3^{\Phi down}, S_3^{\Psi up}, S_3^{\Psi down} \\
S_4 &\rightarrow S_4^{\Phi up}, S_4^{\Phi down}, S_4^{\Psi up}, S_4^{\Psi down} \\
S_5 &\rightarrow S_5^{\chi up}, S_5^{\chi down} \\
S_6 &\rightarrow S_6^{\chi up}, S_6^{\chi down}
\end{aligned}
$$

with "$\rightarrow$" representing the split of the incident potential wavefield at the source interface into reflected/transmitted potential wavefields for the different components, computed as:

$$
\begin{aligned}
S_n^{\Phi up} &= P_n^{\Phi up} e^{i\nu(z-z_i)} \\
S_n^{\Phi down} &= P_n^{\Phi down} e^{-i\nu(z-z_i)} \\
S_n^{\Psi up} &= P_n^{\Psi u} e^{i\gamma(z-z_i)} \\
S_n^{\Psi down} &= P_n^{\Psi down} e^{-i\gamma(z-z_i)} \\
S_n^{\chi up} &= P_n^{\chi u} e^{i\gamma(z-z_i)} \\
S_n^{\chi down} &= P_n^{\chi down} e^{-i\gamma(z-z_i)}
\end{aligned}
$$

where P_n^{η} are the reflectivity coefficients, with $n = 1, 2, ..., 6$ and η pointing at the six modes and upgoing/downgoing components. The reflectivity and transmissivity matrices, representing all the reflectivity coefficients at the different interfaces of the model are calculated using the method of Kennett and Kerry (1979) (See Section B-3).

B-3 Reflectivity method

The reflectivity method is an algebraic scheme defined by Kennett and Kerry (1979). Suppose there exists a linear translation rule $\mathbf{Q}$ that relates the components of 2-dimensional vectors $\mathbf{v}$ between two points a and b as:

$$
\mathbf{v}(b) = \mathbf{Q}(b, a)\mathbf{v}(a) \tag{B-5}
$$

The arguments of the transfer matrix $\mathbf{Q}$ are read from right to left, meaning that $\mathbf{Q}(b, a)$ propagates $\mathbf{v}$ from point a to point b.

In the particular case of wave propagation, $\mathbf{Q}$ is mapping the wavefield $\mathbf{v}$ from one point to another and can be partitioned into four (nXn) blocks, factorized into the product of two matrices as follows:

$$\mathbf{Q}(b,a) = \mathbf{L}^{-1}(a,b)\mathbf{U}(b,a) \tag{B-6}$$

where

$$\mathbf{L}(a,b) = \begin{bmatrix} \mathbf{T}(a,b) & 0 \\ -\mathbf{R}(a,b) & \mathbf{I} \end{bmatrix}, \ \mathbf{U}(b,a) = \begin{bmatrix} \mathbf{I} & -\mathbf{R}(b,a) \\ 0 & \mathbf{T}(b,a) \end{bmatrix} \tag{B-7}$$

and I is the unit matrix. The matrices $\mathbf{T}$ and $\mathbf{R}$ represent the transmission and reflections matrices, respectively. Moreover, we can decompose the wavefield vector $\mathbf{v}$ in upper and lower terms, $\mathbf{v} = (\mathbf{v}_u, \mathbf{v}_d)^T$, representing "upwards" and "downwards" propagation. Following Kennett and Kerry (1979) it is also possible to define the transmission/reflection matrices for a path $a \to b \to c$ as:

$$\mathbf{T}(a,c) = \mathbf{T}(a,b)[\mathbf{I} - \mathbf{R}(c,b)\mathbf{R}(a,b)]^{-1}\mathbf{T}(b,c) \tag{B-8}$$

$$\mathbf{R}(a,c) = \mathbf{R}(b,c) + \mathbf{T}(c,b)\mathbf{R}(a,b)[\mathbf{I} - \mathbf{R}(c,b)\mathbf{R}(a,b)]^{-1}\mathbf{T}(b,c) \tag{B-9}$$

To obtain the transmission/reflection matrices $\mathbf{T}(c,a)$ and $\mathbf{R}(c,a)$ for the path $c \to b \to a$ is necessary only to interchange the arguments a and c.

B-4 Calculation of the spatial derivatives of the Green's functions

In this step, we calculate the potentials in every layer (following Cotton and Coutant (1996)) and deduce the corresponding spatial derivatives of the Green's functions (3-8) by differentiating equations B-3, after substituting the source potentials with the propagated potentials (calculated in Section B-2) at that specific layer. The differentiation of the potentials is performed following the formulas:

$$\begin{aligned} G_r &= \frac{\partial \Phi}{\partial r} + \frac{\partial^2 \Psi}{\partial r \partial z} + \frac{1}{r}\frac{\partial \chi}{\partial \theta} \\ G_\theta &= \frac{1}{r}\frac{\partial \Phi}{\partial \theta} + \frac{1}{r}\frac{\partial^2 \Psi}{\partial \theta \partial z} - \frac{\partial \chi}{\partial r} \\ G_z &= \frac{\partial \Phi}{\partial z} - \frac{\partial^2 \Psi}{\partial r^2} - \frac{1}{r}\frac{\partial \Psi}{\partial r} - \frac{1}{r^2}\frac{\partial^2 \Psi}{\partial \theta^2} = \frac{\partial \Phi}{\partial z} + k_r^2 \Psi \end{aligned} \tag{B-10}$$

Appendix C

Forward modeling code

In this appendix the forward modeling code using the DWN method is provided, as mentioned in Chapter 3.

```
def MTforward(InputValues):

    ## Initialize geometry
    r, nzs, nzr, cosr, sinr = initialize(nc, nr, ns, zr, zs)

    ## Discrete WaveNumber Method
    kaxis = np.arange(0, kmax, dk)
    faxis = np.arange(0, fmax, df)

    # Meshgrid
    F, K = np.meshgrid(faxis, kaxis)

    # Calculate Bessel functions and vertical wavenumbers
    fj0, fj1, cgam, cnu, c2, cka2, ckb2 = reflect0(nc, nr, ns, vp, vs, kr2,
    omega, qp, qs)

    # Calculate reflection and transmission coefficients
    ru, rd, rush, rdsh, tu, td, tush, tdsh = reflect1(nc, kr2, cka2, ckb2, cnu
    , cgam, rho)

    # Calculate reflectivity matrices for every layer
    mt, mb, mtsh, nt, nb, ntsh = reflect2(ru, rd, rush, rdsh, tu, td, tush,
    tdsh)

    # Calculate 6 sources of potentials at source layer
    su1, sd1, su2, sd2, su3, sd3, su4, sd4, su1sh, sd1sh, su2sh, sd2sh =
    reflect3(mt, mb, mtsh, nt, nb, ntsh)

    # Calculate potentials at each receiver and spatial derivatives of Green's
     functions
    pu, pd, push, pdsh, G = reflect4(su1, sd1, su2, sd2, su3, sd3, su4, sd4,
    su1sh, sd1sh, su2sh, sd2sh)

    # Sum potentials for all horizontal wavenumbers
```

```
    G_sum = np.sum(G, axis = 4)

    # Calculation of elementary seismograms in cartesian coordinate system
    E = reflect5(nr, ns, G_sum, cosr, sinr, a1)

    # Calculate moment tensor coefficients and  full moment tensor
    a, M = cmomentfull(strike, dip, rake, mu, lambd, mmoment, width, length,
    Ds, Dn)

    # Calculate displacements for each receiver
    ux, uy, uz = convm(E, a, fs)

    return ux, uy, uz
```

Listing C.1: Main core of the forward modeling code.

Bibliography

Abraham, A. (2005). Artificial neural networks. In *Handbook of Measuring System Design*, chapter 129. American Cancer Society. doi : 10.1002/0471497398.mm421.

Aki, K. and Richards, P. (2002). *Quantitative Seismology Theory*, pages 200–252. University Science Books, Sausalito, California.

Akram, J., Ovcharenko, O., and Peter, D. (2017). A robust neural network-based approach for microseismic event detection. *SEG Technical Program Expanded Abstracts 2017*, pages 2929–2933. doi : 10.1190/segam2017-17761195.1.

Baig, A. and Urbancic, T. (2010). Microseismic moment tensors: A path to understanding frac growth. *The Leading Edge*, 29:320–324. doi : 10.1190/1.3353729.

Bayes, T. and Price, R. (1763). LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418. doi : 10.1098/rstl.1763.0053.

Binder, G. (2018). Neural networks for moment-tensor inversion of surface microseismic data. *SEG Technical Program Expanded Abstracts 2018*, pages 2917–2921. doi : 10.1190/segam2018-2998626.1.

Bishop, C. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bouchon, M. (2003). A review of the discrete wavenumber method. *Pure and Applied Geophysics*, 160:445–465. doi : 10.1007/PL00012545.

Bouchon, M. and Aki, K. (1977). Discrete wave-number representation of seismic-source wave fields. *Bulletin of the Seismological Society of America*, 67(2):259–277.

Chollet, F. et al. (2015). Keras. *GitHub Repository*. https://github.com/fchollet/keras.

Clarkson, C. and Beierle, J. (2011). Integration of microseismic and other post-fracture surveillance with production analysis: A tight gas study. *Journal of Natural Gas Science and Engineering*, 3(2):382 – 401. doi : 10.1016/j.jngse.2011.03.003.

Cotton, F. and Coutant, O. (1996). Dynamic stress variations due to shear faults in a plane-layered medium. *Geophysics Journal International*, 128:676–688.

Cronin, V. (2010). A primer on focal mechanism solutions for geologists. *Science Education Resource Center, Carleton College*.

Denisov, A. M. (1999). *Elements of the theory of inverse problems - Inverse and ill-posed problems series*. De Gruyter, Berlin, Boston.

E.I.A. (2020). U. S. Energy Information Administration: Short-term energy outlook. May 2020. https://www.eia.gov/outlooks/steo.

Eyre, T. and Baan, M. (2017). The reliability of microseismic moment-tensor solutions: Surface versus borehole monitoring. *Geophysics*, 82:1–46. doi : 10.1190/geo2017-0056.1.

Frasier, C. W. (1970). Discrete time solution of plane p-sv waves in a plane layered medium. *Geophysics*, 35(2):197–219.

Gilbert, F. (1971). Excitation of the Normal Modes of the Earth by Earthquake Sources. *Geophysical Journal International*, 22(2):223–226. doi : 10.1111/j.1365-246X.1971.tb03593.x.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Grechka, V., Li, Z., Howell, R., and Vavrycuk, V. (2016). Single-well moment tensor inversion of tensile microseismic events. *Geophysics*, 81:KS219–KS229. doi : 10.1190/geo2016-0186.1.

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, pages 49–52.

Hazewinkel, M. (1994). *Encyclopaedia of Mathematics*. Springer.

Hubral, P., Treitel, S., and Gutowski, P. R. (1980). A sum autoregressive formula for the reflection response. *Geophysics*, 45(11):1697–1705.

Häring, M., Schanz, U., Ladner, F., and Dyer, B. (2008). Characterisation of the basel 1 enhanced geothermal system. *Geothermics*, 37:469–495. doi : 10.1016/j.geothermics.2008.06.002.

Jeffreys, H. and Jeffreys, B. S. (1972.). *Methods of mathematical physics /*. The University press,, Cambridge [Eng.] :, 3rd ed. edition. Includes most of Harold Jeffreys' "Operational methods in mathematical physics" and "Cartesian tensors." cf. Pref.

Jost, M. L. and Herrmann, R. B. (1989). A Student's Guide to and Review of Moment Tensors. *Seismological Research Letters*, 60(2):37–57. doi : 10.1785/gssrl.60.2.37.

Kennett, B. L. N. and Kerry, N. J. (1979). Seismic waves in a stratified half space. *Geophysical Journal of the Royal Astronomical Society*, 57(3):557–583. doi : 10.1111/j.1365-246X.1979.tb06779.x.

Kennett, B. L. N., Koketsu, K., and Haines, A. J. (1990). Propagation invariants, reflection and transmission in anisotropic, laterally heterogeneous media. *Geophysical Journal International*, 103(1):95–101.

Koch, K. (1991). Moment tensor inversion of local earthquake data—I. Investigation of the method and its numerical stability with model calculations. *Geophysical Journal International*, 106(2):305–319. doi : 10.1111/j.1365-246X.1991.tb03894.x.

Käufl, P., Valentine, A., Wit, R., and Trampert, J. (2015). Robust and fast probabilistic source parameter estimation from near-field displacement waveforms using pattern recognition. *Bulletin of the Seismological Society of America*, 105:2299–2312. doi : 10.1785/0120150010.

Ma, Y. Z. and Holditch, S. A. (2016). *Unconventional Oil and Gas Resources Handbook - Evaluation and Development*. Elsevier.

Maity, D., Aminzadeh, F., and Karrenbach, M. (2014). Novel hybrid artificial neural network based autopicking workflow for passive seismic data. *Geophysical Prospecting*, 62. doi : 10.1111/1365-2478.12125.

Maxwell, S. C., Rutledge, J., Jones, R., and Fehler, M. (2010). Petroleum reservoir characterization using downhole microseismic monitoring. *Geophysics*, 75(5):75A129–75A137. doi : 10.1190/1.3477966.

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. doi: 10.1007/BF02478259.

McGillivray, P. (2005). Microseismic and time-lapse seismic monitoring of a heavy oil extraction process at Peace River, Canada. *SEG Technical Program Expanded Abstracts 2004*, pages 572–575. doi : 10.1190/1.1842410.

Mira, J. and Sanchez-Andres, J. V. (1999). Engineering applications of bio-inspired artificial neural networks. *International Work-Conference on Artificial and Natural Neural Networks*, II. doi : 10.1007/BFb0100499.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., USA, 1st edition.

Moya, A. and Irikura, K. (2010). Inversion of a velocity model using artificial neural networks. *Computers & Geosciences*, 36:1474–1483. doi : 10.1016/j.cageo.2009.08.010.

Mustać, M. and Tkalčić, H. (2015). Point source moment tensor inversion through a Bayesian hierarchical model. *Geophysical Journal International*, 204(1):311–323. doi : 10.1093/gji/ggv458.

Ovcharenko, O., Akram, J., and Peter, D. (2018). Feasibility of moment tensor inversion from a single borehole data using artificial neural networks. *GEO Bahrain 2018*, 42212.

Pugh, D. J., White, R. S., and Christie, P. A. F. (2016). A Bayesian method for microseismic source inversion. *Geophysical Journal International*, 206(2):1009–1038. doi : 10.1093/gji/ggw186.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.

Röth, G. and Tarantola, A. (1994). Neural networks and inversion of seismic data. *Journal of Geophysical Research: Solid Earth*, 99(B4):6753–6768. doi : 10.1029/93JB01563.

Sipkin, S. A. (1982). Estimation of earthquake source parameters by the inversion of waveform data: synthetic waveforms. *Physics of the Earth and Planetary Interiors*, 30(2):242 – 259. Special Issue Earthquake Algorithms.

Song, F. and Toksoez, M. N. (2011). Full-waveform based complete moment tensor inversion and stress estimation from downhole microseismic data for hydrofracture monitoring. *Earth Resources Laboratory Industry Consortia Annual Report*, 14.

Suk, H. (2017). Chapter 1 - An introduction to neural networks and deep learning. In *Deep Learning for Medical Image Analysis*, pages 3 – 24. Academic Press. doi : 10.1016/B978-0-12-810408-8.00002-X.

Tano, M., Portwood, G., Ragusa, J., and Tano Retamales, M. (2020). Accelerating training in artificial neural networks with dynamic mode decomposition. *ArXiv*, abs/2006.14371.

Tarantola, A. (2004). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, USA.

Tarantola, A. and Valette, B. (1982). Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics*, 20:219–232.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural networks for machine learning*, 4(2):26–31.

Urbancic, T. and Trifu, C.-I. (2000). Recent advances in seismic monitoring technology at canadian mines. *Journal of Applied Geophysics*, 45:225–237. doi : 10.1016/S0926-9851(00)00030-6.

Vackář, J., Burjánek, J., Gallovič, F., Zahradník, J., and Clinton, J. (2017). Bayesian ISOLA: new tool for automated centroid moment tensor inversion. *Geophysical Journal International*, 210(2):693–705. doi : 10.1093/gji/ggx158.

van der Baan, M., Eaton, D., and Dusseault, M. (2013). Microseismic monitoring developments in hydraulic fracture stimulation. In Bunger, A. P., McLennan, J., and Jeffrey, R., editors, *Effective and Sustainable Hydraulic Fracturing*, chapter 21. IntechOpen, Rijeka. doi : 10.5772/56444.

Vavrycuk, V. (2007). On the retrieval of moment tensors from borehole data. *Geophysical Prospecting*, 55(03):381 – 391. doi : 10.1111/j.1365-2478.2007.00624.x.

Vavrycuk, V. (2015). Inversion for the composite moment tensor. *Bulletin of the Seismological Society of America*, 105(06):3024–3035.

Vavryčuk, V. (2015). *Moment Tensors: Decomposition and Visualization*, pages 1546–1559. Springer Berlin Heidelberg, Berlin, Heidelberg. doi : 10.1007/978-3-642-35344-4_288.

Verdon, J. (2011). Microseismic monitoring and geomechanical modeling of CO_2 storage in subsurface reservoirs. *Geophysics*, 76(5):Z102–Z103. doi : 10.1190/2011-0926-GEODIS.6.

Verdon, J. P., Kendall, J. M., and White, D. J. (2011). Linking microseismic event observations with geomechanical models to minimise the risks of storing CO_2 in geological formations. *Earth and Planetary Science Letters*, 305(1-2):143 – 152.

Wapenaar, C. and Berkhout, A. (1989). *Elastic wave field extrapolation: Redatuming of single-and multi-component seismic data*. Elsevier.

Warpinski, N. (2009). Microseismic monitoring: Inside and out. *Journal of Petroleum Technology*, 61(11):80–85. doi : 10.2118/118537-MS.

Young, R. and Baker, C. (2001). Microseismic investigation of rock fracture and its application in rock and petroleum engineering. *International Society for Rock Mechanics News Journal*, 7:19–27.

Zhdanov, M. S. (2015). *Inverse Theory and Applications in Geophysics*. Elsevier Science Technology, Oxford.