UiO : **University of Oslo**

Thomas André Larsen Greiner

Deep learning in seismic wavefield reconstruction and resolution enhancement

Thesis submitted for the degree of Philosophiae Doctor

Faculty of Mathematics and Natural Sciences

2021

© **Thomas André Larsen Greiner, 2021**

*Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2476*

ISSN 1501-7710

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

Cover: Hanne Baadsgaard Utigard.

Print production: Reprosentralen, University of Oslo.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at Lundin-Norway AS, the University of Oslo, and at the Heriot-Watt University in Edinburgh where I had a 3-month research stay. The work was conducted under the supervision of Professor Leiv-J Gelius at the University of Oslo-Department of Geoscience, Dr. Jan Erik Lie and Dr. Andreas Kjelsrud Evensen at Lundin-Norway AS and Associate Professor Odd Kolbjørnsen at Lundin-Norway AS and the University of Oslo-Department of Mathematics. The work at Heriot-Watt University was conducted under the supervision of Professor Vasily Demyanov.

This research is part of Lundin-Norway's research project *Machine Learning in Geophysical Processing and Inversion* and is part of the industrial PhD-program that is financially supported by the Research Council of Norway, under project number 287664. The thesis is a collection of three papers, presented in chronological order of writing. The papers are preceded by an introduction chapter on the motivation for the work, and two chapters introducing the reader to the theoretical background and finally a chapter on discussion of future work and implications of the results.

Acknowledgements

The completion of this PhD project would not have been possible without the contribution from many people, to whom I am very grateful.

First and foremost I want to thank Dr. Jan Erik Lie for taking the initiative and suggesting the topic. I am truly grateful for your mentoring and the freedom granted to pursue my interests. I am also grateful to my academic supervisors Prof. Leiv Gelius and Associate Prof. Odd Kolbjørnsen for your mentoring and fruitful discussions on many theoretical matters of high importance to the project. A humble thanks should be directed to Dr. Andreas Kjelsrud Evensen and Espen Harris Nilsen for help and mentoring on seismic data processing, which has been crucial to this thesis.

I thank my fellow PhD student and co-author Volodya Hlebnikov and additional co-authors Dr. Vetle Vinje, Dr Hao Zhao and Prof. Vasily Demyanov for a successful collaboration and many fruitful discussions, and Prof. Demyanov for welcoming me to the group at Heriot-Watt university.

I thank Lundin-Energy Norway for providing excellent working facilities, computational resources, and together with the Research Council of Norway, for funding this PhD project.

In 2009, I climbed down from my high horse and gave up the sophisticated life as a plumber, to go back to school on a quest to become something “somewhat different”. I remember in particular some of the early conversations on the phone with my brother, Christopher, and my childhood friend, Henrik, where we discussed mathematical problems. One of these involved solving a system of linear equations, where a fox were racing a rabbit, and the rabbit have had a head start. I could feel the facepalms through the phone, as they struggled to spoon-feed this noob-plumber with simple physics and mathematics... The journey has been long and bumpy indeed, but I managed with a little help from my friends - and family.

Especially, I am grateful to my brother, Christopher, and sister, Mette Yvonne, who i consider important role models and a huge inspiration through my life. Also, I thank my close friend, Simen, who I teamed up with in the early days of academia. Of course, the greatest thanks should go to the two attention-seeking Tinkerbell’s, Heddi and Camilla, that would probably die if it weren’t for me constantly feeding them with my rather dark and juvenile sense of humor. Thank you for making the life at UiO so much more fun together with the rest of my friends and colleagues.

And finally, my deepest gratitude to my dearest Maria for your endless patience, putting up with long days and a long PhD-face. Lastly, a thank should also go to my beloved son Billy. Thank you for teaching me (and Maria) how to function in everyday life without sleep (what is sleep?).

Now, finally, the day has come when I can throw the thesis into the fires of Mount Doom and be free from the mental cave that is PhD-student life - Gollum Gollu....

• Thomas André Larsen Greiner
Oslo, October 2021

List of Papers

Greiner, T. A. L., V. Hlebnikov, J. E. Lie, O. Kolbjørnsen, A. K. Evensen, E. H. Nilsen, V. Vinje, and L.-J. Gelius, 2020, “Cross-streamer wavefield reconstruction through wavelet domain learning”. In: *Geophysics*. Vol. 85, V457-V471. DOI: <https://doi.org/10.1190/geo2019-0771.1>.

Greiner, T. A. L., J. E. Lie, O. Kolbjørnsen, A. K. Evensen, E. H. Nilsen, H. Zhao, V. Demyanov, and L.-J. Gelius, 2021, “Unsupervised deep learning with higher-order total-variation regularization for multidimensional seismic data reconstruction”. Accepted for publication in: *Geophysics*.

Hlebnikov, V., **T. A. L. Greiner**, V. Vinje, J. E. Lie, and G. Poole, 2021, “Modelling-based supervised learning for interpolation of 3D constant offset volumes”. Submitted to: *Geophysical Prospecting*.

List of additional contributions

Convolutional neural network approach for seismic data interpolation across the streamers

Greiner, T. A. L., O. Kolbjørnsen, J. E. Lie, E. H. Nilsen, A. K. Evensen, and L.-J. Gelius, 2019, “Cross-streamer wavefield interpolation using deep convolutional neural network”. In: *SEG Technical Program Expanded Abstracts*. 2207-2211. DOI: <https://doi.org/10.1190/segam2019-3214009.1>.

Convolutional neural network approach for seismic interference noise removal

Sun, J., S. Slang, T. Elboth, T. A. L. Greiner, S. McDonald, and L.-J. Gelius, 2020, “Attenuation of marine seismic interference noise employing a customized u-net”. In: *Geophysical Prospecting*. Vol. 68, pp. 845–871. DOI: <https://doi.org/10.1111/1365-2478.12893>.

Convolutional neural network approach for seismic data deblending

Sun, J., S. Slang, T. Elboth, T. A. L. Greiner, S. McDonald, and L.-J. Gelius, 2020, “A convolutional neural network approach to deblending seismic data”. In: *Geophysics*. Vol. 85, WA13-WA26. DOI: <https://doi.org/10.1190/geo2019-0173.1>.

Common reflection surface approach for multidimensional seismic data reconstruction

Zhao, H., V. Hlebnikov, E. H. Nilsen, A. K. Evensen, T. A. L. Greiner, J. E. Lie, and L. J. Gelius, 2020, “Kinematic wavefield attributes based multidimensional prestack data reconstruction”. In: *SEG Technical Program Expanded Abstracts*. 3304-3308. DOI: <https://doi.org/10.1190/segam2020-3428420.1>.

Contents

Preface	i
List of Papers	iii
List of additional contributions	v
Contents	vii
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.1.1 Problem description	3
1.1.2 Related work	4
1.1.3 Thesis objectives	6
1.1.4 Thesis outline	7
2 Inverse problems and regularization	9
2.0.1 Wavefield sampling and resolution	9
2.0.2 A quick look into ill-posed problems	12
2.0.3 Regularization methods	16
3 Deep learning for inverse problems	23
3.1 The deep learning model	23
3.1.1 Fully connected neural network	23
3.1.2 Convolutional neural network and multidimen- sional data	25
3.2 Training deep learning models	29
3.2.1 Supervised learning basics	29
3.2.2 The optimization problem	30
3.2.3 Model selection	33
3.2.4 Transform-domain learning	35
3.2.5 Guided unsupervised learning for inverse problems	39
4 Summary of papers, their contributions, implications and discussion of future work	43
4.1 Summary of papers	43
4.1.1 Summary Paper I: Cross-streamer wavefield re- construction through wavelet domain learning . .	43
	vii

4.1.2	Summary Paper II: Unsupervised deep learning with higher-order total-variation regularization for multidimensional wavefield reconstruction	44
4.1.3	Summary Paper III: De-migration-based supervised learning for interpolation and regularization of 3D constant offset volumes	45
4.2	Implications and discussion of future work	46
4.2.1	Alternative transform domains	46
4.2.2	2D-based vs. 3D/4D-based reconstruction	46
4.2.3	Implications of the results	53
Bibliography		57
Papers		66
I	Cross-streamer wavefield reconstruction through wavelet domain learning	67
II	Unsupervised deep learning with higher-order total-variation regularization for multidimensional seismic data reconstruction	83
III	De-migration-based supervised learning for interpolation and regularization of 3D constant offset volumes	103

List of Figures

1.1		1
1.2		2
2.1		10
2.2		10
2.3		11
2.4		12
2.5		13
2.6		14
2.7		14
2.8		16
2.9		18
2.10		18
2.11		19
2.12		20
2.13		21
3.1		24
3.2		26
3.3		30
3.4		30
3.5		32
3.6		34
3.7		35
3.8		38
3.9		40
3.10		41
3.11		42
4.1		47
4.2		48
4.3		49
4.4		50
4.5		52
4.6		53
4.7		54
4.8		55
4.9		56

Chapter 1

Introduction

1.1 Motivation

The seismic industry benefits from technological and conceptual advances within the entire seismic data pipeline, from acquisition, processing, imaging and interpretation. Exploration seismology deals with artificially generated elastic waves to provide an image of the subsurface, for further analysis of its properties as well as its geometrical appearance in order to locate oil and gas reservoirs. The seismic waves are recorded close to the sea-surface, traditionally by a 3D spread of ten or more streamers. The conventional 3D seismic spread, however, lacks near-offsets, which are important for imaging shallow parts of the subsurface [85]. In addition, imaging challenges occurs in the Barents Sea mainly due to the presence of high-velocity carbonate layer(s) that limits the range of offsets due to near-vertical critical angles. Thus, at shallow depths, reflections may go overcritical at relatively small angles, which limits the amount of recorded energy from conventional marine acquisition. This is illustrated in Figure 1.1. Therefore, a conventional 3D acquisition does not represent an optimal setup in these environments. To address such shallow target seismic imaging problems,

Figure 1.1: A regional profile over the Loppa High area in the Barents Sea including discovery wells and prospects. The shallow depths and complicated velocity-depth trends lead to reflections going overcritical at relatively small angles, which restricts the number of usable offsets considerably. Figure taken from Vinje et al [85].

1. Introduction

Lundin-Norway and CGG proposed an acquisition solution, known as TopSeis [85], yielding improved recording of the near-offsets and improved imaging of the subsurface compared to conventional 3D marine acquisition. Compared to conventional surveys, TopSeis utilize one streamer vessel and a separate source vessel positioned at the center of the streamer array as illustrated in Figure 1.2. TopSeis also introduce reduced streamer separation, wider source separation and an increase in number of sources. Even though this acquisition setup yields improved recording of the 3D wavefield, the near-offsets are still spatially undersampled, and the data suffer from severe spatial aliasing in the shallow parts of the seismic record. Thus, challenges within the seismic data processing pipeline still exists.

Figure 1.2: An illustration of a conventional survey (top) and a source-over-cable survey employing both a streamer vessel and a separate source vessel (bottom). The spatial sampling depends on receiver separation in the inline direction Δy and in the crossline direction Δx . The number of sources s and their separation Δs are also crucial factors considering different domains of sorting.

Seismic data processing involves various subtasks that are based on manual evaluation and testing of a complex interdependent multivariable parameter scheme, that are often biased towards our prior beliefs and approximate

descriptions of reality. Thus, it could substantially benefit from advances in data-driven technology and automatization algorithms. Recent advances in the field of artificial intelligence (AI), and particularly in classification for medical imaging—which is related to seismic imaging in many ways—, have demonstrated state-of-the-art performance on tasks that were once considered almost unique to intelligent beings. A sub-category of AI algorithms, known as machine learning (ML) give computers the ability to learn how to solve specific tasks from the use of data, and to improve with experience. Also, the increase in open source code and libraries for efficient computations, such as Tensorflow [1] and scikit-learn [61], has made AI research applied to challenging tasks within geophysical processing highly sought-after.

This thesis addresses the issue of undersampled wavefield in exploration seismology and investigates the use of ML, or more precisely a sub-category known as deep learning (DL), applied to wavefield reconstruction and resolution enhancement of seismic data. At the beginning of this thesis work, several promising results from other scientific fields, such computational inverse problems in natural image processing [51], were reported, which encouraged research of DL applications to similar problems within geoscience. Also, for classification problems, in 2012 the neural network AlexNet [44] more or less revolutionized image classification tasks. In geoscientific problems, we also started to see several interesting applications from AI on challenging problems such as seismic interpretation [86] and seismic data processing [38, 87]. The DL models are considered computational models that tries to mimic the way a brain process information. This is done through the use of artificial neural networks, or simply neural networks (NN).

In mathematical theory, NN's are considered universal function-approximators [36], which motivates the use of NN's in seismic data processing since the recorded seismic waves follows specific kind of functions and patterns. Conventional seismic data processing include series of assumptions and approximations in order to achieve a reasonable prediction of the subsurface model, which may be bypassed by using NN. However, we must also be aware of the weaknesses of AI applied to inverse problems in computational sciences [2, 3, 25], in order to build reliable tools for seismic data processing applications.

1.1.1 Problem description

Marine seismic data represent a multidimensional record of reflected and diffracted/scattered pressure waves. Contracting companies specialize on seismic data, as such data are carriers of information about the subsurface physical properties and its structural appearance. In marine seismic exploration, air guns generate acoustic pressure waves, that propagate through the earths interior. If the downward propagating waves encounters changes in material properties, the waves partition into reflected/diffracted- and transmitted parts, where the upward propagating pressure waves are recorded close to the sea-surface by the use of an array of receivers. The transmitted waves follow the same principle, where they partition further by reflection/diffraction and transmission. Seismic

data processing utilize mathematical and statistical principles from signal analysis in combination with known theory of the physics of wave-propagation in order to remove undesired signals, such as multiples, and to keep as much as possible of the recorded primary wavefield. In addition, seismic acquisition typically comprise a narrow source distribution and coarse receiver distribution, which leads to a sparse and irregular distribution of the recorded seismic traces. The wavefield therefore needs to be interpolated before migration, in which the reflected and diffracted events are focused back to its original subsurface points of origin. The process of migration therefore reveals the subsurface geological structures and discontinuities, which are “hidden” within the recorded seismic wavefield. It is therefore crucial that interpolation yields an accurate infill of missing data, since migration of misplaced waveforms can be highly detrimental for the final image [79, 80].

1.1.1.1 Reconstruction problem

In this thesis, reconstruction refers to the problem of improving the resolution of the seismic imaging system, by compensating for the coarse sampling from the 3D seismic spread. In this context, reconstruction of seismic data is cast as a compressed sensing problem [11] where we wish to interpolate missing frequency content, and/or as a super-resolution problem [10] where we wish to extrapolate the frequency content of the signal. Note, that the word “super-resolution” is used in different contexts, but commonly relates to the design of techniques for enhancing the resolution of a sensing system [10]. The seismic wavefield reconstruction problem, in this case, share many similarities with computational inverse problems in image processing. Such problems include image inpainting, image deblurring and image scaling in which the degraded and/or low-resolution data are transformed into its corresponding interpolated and/or high-resolution counterpart. In recent years, AI techniques have shown increased popularity in such problems due to improved efficiency and state-of-the-art performance compared to traditional approaches [91].

1.1.2 Related work

Knowledge-driven methods for seismic wavefield reconstruction typically incorporate domain-specific knowledge to obtain desired results. In contrast, pure data-driven AI techniques utilize large data sets and statistical principles to learn a generalized transformation from an input to a desired output. Both methodologies possess their individual strengths and weaknesses. For instance, our specific knowledge about the problem often reflects only in an approximate description of reality, and pure data-driven techniques may perform poorly if the data are sampled too scarce or unreliably.

The wavefield reconstruction problem has been studied for decades, where the well-known $f - x$ interpolation [73] from the early 90’s, and its variants, are still used as a benchmark method in multiple research papers. Traditional approaches and DL typically share many ingredients, and often consist of rigorous

mathematical treatment of statistical concepts and optimization. A backbone of general inverse problems, is to first define a regularized objective function, where one computes the error between the prediction and the observed data including an upper bound on some norm of the solution.

Historically, some of the most successful wavefield reconstruction techniques, include such treatments and utilize optimization algorithms to find the optimal solution amongst several feasible candidates. One of these techniques is the minimum weighted norm interpolation (MWNI) [47]. MWNI treats the problem as an underdetermined inverse problem, and tries to estimate an appropriate inverse of the subsampling operator including a regularization penalty in the wavenumber domain. Similarly, the shaping regularization [20] technique tries to generalize the regularized solution by introducing an user-defined shaping operator, instead of choosing an appropriate regularization operator as in classical *Tikhonov* regularization [62, 78]. Also, the concept of matrix decomposition and low-rank approximation has been applied to fill in missing data by completion of extended Hankel matrices [81]. In this case, one regards the reconstruction problem as a matrix rank reduction problem. Rather than relying on predefined transformations or matrix decomposition, some approximation methods represent the data as a sparse linear combination of an overcomplete and learned dictionary [82, 95]. DL-based methods can actually be viewed as special cases of the sparse approximation methods, but instead of optimizing each component separately, i.e., the dictionary including its sparse vectors, both components are instead jointly optimized [16]. Meaning, that the components are generalized to specific input-to-output patterns. Also, a class of NN's known as convolutional neural networks (CNN's) with encoder-decoder architecture, share many similarities with low-rank approximation of extended Hankel matrices [29, 92, 93].

The differences lie more within the methods capability to generalize to the specific input to output patterns, that are learned from large data sets. Some ML techniques [38, 39] and DL-based techniques [87] had already shown interesting and promising results in geophysical reconstruction problems. The problem in geophysical sciences, however, is the limited access to reliable targets or ground truth data. One of the working horses in DL is supervised learning, where one have access to a large set of input/target pairs from which the algorithm learns a generalized inverse mapping. However, the target is in many circumstances difficult or too costly to acquire. In some cases, one may construct targets by using traditional methods, and then learn a generalized inverse from the resulting input/target pairs, which may gain a speed up application-wise. However, in situations where traditional methods may fail to give adequate results, this is not optimal and may not lead to desired improvements of the applications. In these cases, integration of domain-knowledge into the AI techniques and/or utilize rules from physics by for example modelling the solution to the wave equation to acquire training data, are both possible paths to take. Such physics-driven AI may lead to a new era of geophysical research [35].

1.1.3 Thesis objectives

The objective of this thesis is to investigate and develop DL techniques for seismic processing applications, with main emphasis on: (1) *enhance the resolution and quality of seismic data image*, and with a potential synergy effect of (2) *faster and more efficient processing applications*. One of the core ideas of DL, is that the trained models have the capacity of being generalized to certain tasks, making the reconstruction both fast and effective, despite taking a long time to train. However, the main focus of this thesis is that the developed techniques compare to or supersede current industry standard state-of-the-art workflows, not to build a fast and effective application that copies traditional methodology.

In this case, problems that are of particular interest are cases where traditional methodology may fail, such as reconstruction of aliased data that contain both curved events and conflicting dips. However, as touched upon in the previous section is that one of the key challenges of DL when applied to geophysical inverse problems, is the limited access to labeled training data; where we have “full” knowledge about the target/ground truth. To this purpose, we investigate how we can exploit similarities between the acquisition domains, i.e., inline vs. crossline, for supervised learning without learning from a traditional technique. As an alternative direction, we investigate the use of synthetic data to perform supervised learning, and apply the trained model to representative field data. Finally, we also investigate the use of unsupervised DL, in which we do not need labeled training data. In the unsupervised approach, we combine data-driven theory and knowledge-based theory to obtain a desired solution.

To study the aforementioned problems, we investigate DL techniques in combination with traditional mathematical regularization and transformation-based methods, in both supervised and unsupervised learning frameworks. Although motivated by the challenging processing of seismic data from source-over-cable geometry, i.e., TopSeis, the approaches and methodology presented are generic and applicable for any types of data that can be structured in similar ways.

To meet these concerns, the following approaches have been studied/developed:

1. Exploitation of the similarities between inline and crossline direction to predict a denser crossline sampling, using supervised learning in a transform domain.
2. Using pre-stack depth migrated images from field data to de-migrate and generate coarse and dense synthetic data for supervised learning, in order to interpolate and regularize offset classes.
3. Using only the irregularly sampled observed data as “training” data for unsupervised learning combined with mathematical regularization, in order to reconstruct the sparse near-offset classes.

1.1.4 Thesis outline

The main body of the thesis is organized as follows: Chapter 2 starts by introducing the wavefield sampling problem that comprise band-limited and undersampled data, with a focus on the source-over-cable geometry. Then follows a quick look into inverse problems and a simple analysis using analytical techniques on how we can utilize regularization to stabilize and give preference to a desired behavior of the solution. In chapter 3, I introduce the basics of deep learning for relevant inverse problems and the most common network architectures used in such problems. Then, I introduce the basics of supervised learning for inverse problems, and how deep learning models are trained using optimization and how model selection is carried out in DL. Then I discuss the transform based learning approach, followed by a section introducing unsupervised deep learning. In chapter 4, I give a short summary of the contributions and implications learned from the papers. I finalize this chapter with a section where I discuss limitations and outlook.

Chapter 2

Inverse problems and regularization

2.0.1 Wavefield sampling and resolution

In marine seismic exploration the resolution and available bandwidth of the recorded seismic wavefield depends on the bandwidth of the seismic source, the earths filtering effect on the seismic waves, the seismic survey source array, i.e., number of air guns and their separation, and the seismic survey sensing array, i.e., number of streamers and their separation. For example, the different sampling interval along the streamers compared to across the streamers, shown in Figure 2.1, have impact on the resolution of the recorded wavefield. To discuss the resolution problem, let $d(x, y, t)$ denote a recorded 3D wavefield where x, y, t denotes the crossline-, inline- and temporal coordinate, respectively. The process of recovering frequency information above the highest frequency content in $d(x, y, t)$, may be considered a super-resolution problem [10], where from coarse information only we try to restore finer details. The available bandwidth of the recorded wavefield is a result of a filter $g(x, y, t)$ acting on the “ideal” wavefield $m(x, y, t)$, and the sampling from a function $s(x, y, t)$ which depends on the geophysical domain of sorting. From a mathematical perspective, the filter effect from $g(x, y, t)$ can be written as

$$d(x, y, t) = (g * m)(x, y, t) \quad (2.1)$$

where $*$ denotes the convolution. Since convolution in space/time corresponds to multiplication in the frequency domain, the Fourier transform of $d(x, y, t)$ becomes

$$D(k_x, k_y, f) = G(k_x, k_y, f)M(k_x, k_y, f) \quad (2.2)$$

where f denotes the temporal frequency, and k_x and k_y denote the wavenumbers in the crossline- and inline directions, respectively. The transfer function $G(k_x, k_y, f)$ is commonly represented by a band-pass filter that removes both the low- and high end of the frequency content from the true model M . Thus, $G(k_x, k_y, f) = 0$ for k_x, k_y, f above/below some cutoff-frequencies. Furthermore, the subsampling function $S(k_x, k_y, f)$ takes into account the actual source/receiver design and modifies equation 2.2 to

$$D_s(k_x, k_y, f) = S(k_x, k_y, f)D(k_x, k_y, f) \quad (2.3)$$

Figure 2.1: A 3D shot gather, displayed in the inline direction along one streamer (left) and across the streamers (right).

In this case, if $D_s(k_x, k_y, f)$ is not sampled according to the Nyquist criterion, i.e., two times the highest frequency in signal, aliasing will take place. This means, that we only observe Fourier samples in $D_s(k_x, k_y, f)$ up to one half of the temporal sampling frequency $f_s/2 = 1/2\Delta t$, and one half of the spatial sampling frequencies $k_{s,x}/2 = 1/2\Delta x$ and $k_{s,y}/2 = 1/2\Delta y$, where Δx , Δy and Δt denote the crossline, inline and temporal sampling, respectively. Although, in practice we need even denser sampling due to noise. If sampling follows the Nyquist criterion, then the Nyquist-Shannon sampling theorem states that we

can recover missing signals through sinc interpolation [70]. However, to acquire seismic data over a sufficiently large aperture, the streamers must often be separated by larger distances than desirable [65]. As a consequence, the wavefield in the crossline direction is therefore not sampled up to Nyquist. In the deeper sections of the seismic record, the crossline sampling is typically not a problem due to the lower frequency content will only exhibit a mild degree of aliasing. However, for the shallow section where we may utilize higher frequencies in processing, the high temporal frequencies make recovery of these signals a much more challenging task.

Figure 2.2: This figure shows the near-offset (25 m) extracted from a source-over-cable (TopSeis) survey. Here, the action of the operator S on the 25 m offset reflects in a sparse data volume with an irregularly distributed set of traces.

In the shot-domain (where traces reflect a single shot with many receivers), the 3D split-spread gathers from TopSeis (Figure 2.1) are sampled along the streamers at an interval of $\Delta y = 12.5$ m and at an interval of $\Delta x = 50$ m in the crossline direction. Thus, the sampling of the wavefield is dependent on the domain of sorting. Although, $S(k_x, k_y, f)$ acts merely as an all-pass filter in the temporal coordinate, and in most cases, the same applies along the streamers in the inline direction. In the crossline direction, however, $S(k_x, k_y, f)$ has a cutoff-frequency below $G(k_x, k_y, f)$, and the data $D(k_x, k_y, f)$ is downsampled further, determined by the streamer separation. In the offset-domain —where the seismic traces are distributed based on their distance from the source array—the data exhibit spatial irregular distribution of traces as shown in Figure 2.2, with large gaps between traces. In this case, the population of traces becomes denser at higher offsets, but with an alteration of the wavefields character. The increase in trace population from near- to higher offset is illustrated on time-slices in Figure 2.3, where the 25 m offset from TopSeis is much sparser compared to the 125 m offset.

Figure 2.3: Time-slices from 4D sorted data, where this figure shows the 25 m offset (left) and the 125 m offset (right). In this domain, higher offsets are more densely sampled.

In compressed sensing and super-resolution theory, it is often convenient to refer to the null-space of an operator. This means that the operator, projects the data onto a lower-dimensional subspace, and the missing data thus constitute the orthogonal complement to the null-space of the operator. Meaning that the missing data are orthogonal to the row-space of the operator. Hence, infinitely many solutions may explain the observations if we do not constrain the solution space. Considering real numbers, let $\mathbf{A}$ denote an arbitrary operator, then the null-space of $\mathbf{A}$ is defined as

$$\mathcal{N}(\mathbf{A}) = \{\mathbf{x} \in \mathcal{M} \mid \mathbf{A}(\mathbf{x}) = \mathbf{0}\}, \quad (2.4)$$

where $\mathbf{x}$ is an element in the vector space $\mathcal{M}$. In this thesis, we have studied problems where the null-space of an operator is highly relevant, since two quite different vectors can map into the same or similar low-dimensional subspace. Thus, we need to find a proper way to solve for the inverse projections. Figure 2.4

Figure 2.4: In the direct/forward problem, the masking operator $\mathbf{S}$ projects the complete wavefield $\mathbf{m}_d$ onto a lower-dimensional sub-space $\mathbf{d}_s$. The inverse problem then aims to find an approximation $\hat{\mathbf{m}}_d$. This Figure is taken and modified from paper II.

for instance, shows an example of a masking operator $\mathbf{S}$, which zeroes out large parts of the full wavefield $\mathbf{m}_d$ and gives the observed wavfield $\mathbf{d}_s$. The task is then to find an appropriate approximation to the inverse operator $\mathbf{S}^{-1}$ that projects the observed data back onto a higher-dimensional solution-space. These operations are commonly referred to as the direct/forward problem and the inverse problem.

2.0.2 A quick look into ill-posed problems

Inverse problems relates to the task of finding an approximation to an unknown true model that we cannot directly observe in our measurements. In the continuous 1D case, let this unknown model be denoted by $m(t)$, where t denotes the time coordinate. The observed measurements can be formulated as an integral equation, expressed as the Fredholm integral equation of the first kind [30]:

$$d(s) = \int_0^a g(s, t)m(t)dt, \quad 0 \leq s \leq a, \quad (2.5)$$

where $d(s)$ are the observed measurements, and $g(s, t)$ is a kernel that describes the underlying system that gave rise to the observed data. From equation 2.5 we see that if $g(s, t)$ and $m(t)$ are known, we can compute an evaluation of the measurements. This is known as the forward problem. Thus, the inverse

problem relates to computing an estimation of $m(t)$ given the measurements $d(s)$ and the kernel $g(s, t)$. The kernel $g(s, t)$ is in many cases assumed to be known—or known approximately—by our understanding of the relationship between $m(t)$ and $d(s)$. If we assume that the integral kernel $g(s, t)$ is invariant, then equation 2.5 reduces to the special case of time-invariant deconvolution

$$d(s) = \int_0^a g(s - t)m(t)dt, \quad 0 \leq s \leq a, \quad (2.6)$$

where $g(s - t)$ denotes a convolution kernel. In the discrete case, the integral equation 2.6 can be approximated by the following linear system, i.e., the matrix-vector product

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (2.7)$$

where $\mathbf{G} \in \mathbb{R}^{N \times N}$ denotes a matrix representation of the convolution operation and $\mathbf{m} \in \mathbb{R}^N$ and $\mathbf{d} \in \mathbb{R}^N$ are vectors denoting the observed data and true model, respectively. Finally, N denotes the number of measurements. As in equation 2.2, the operator $\mathbf{G}$ may denote a band-pass or low-pass filter, removing frequency content from the high- and/or low-end of the frequency spectrum of $\mathbf{m}$.

Figure 2.5: The observed data (left) were computed by convolving the Gaussian kernel (middle) with a series of four spikes (right).

Consider a simple example where the kernel $\mathbf{G}$ is given by a Gaussian function, and the true model consists of a series of spikes, as displayed in Figure 2.5. The objective is to recover these spikes from the observed data. Since $\mathbf{G}$ is square, we can try to solve for $\mathbf{m}$ by the direct inverse

$$\hat{\mathbf{m}} = \mathbf{G}^{-1}\mathbf{d}. \quad (2.8)$$

The solution is displayed in Figure 2.6, where we observe that the approximated model has a near perfect fit to the true model. However, in reality we need to take noise into consideration, in which equation 2.7 is modified to:

$$\mathbf{d} = \mathbf{G}\mathbf{m} + \boldsymbol{\varepsilon}, \quad (2.9)$$

2. Inverse problems and regularization

where ε denotes the noise. The direct inverse in presence of a small amount of white noise is displayed in Figure 2.7, where we observe a blow-up in the solution. Hence, computing the direct inverse of this forward operator is naïve in the sense that even a small amount of noise leads to large errors.

Figure 2.6: The approximated model (right) shows a near perfect fit when the direct inverse is computed on the observed data (left).

Figure 2.7: Introducing a small amount of noise to the data (left) renders the solution useless, where the approximated model (right) does not resemble the true model in any way.

The cause of this instability relates to the forward operator $\mathbf{G}$ being ill-conditioned, giving an unbounded inverse operator which reflects in non-physical blow-ups in the solution. Instead, we can approximate the solution through a linear combination of a finite set of basis functions $\{\phi_i\}_{i=1}^K$ and coefficients $\{c_i\}_{i=1}^K$, i.e.,

$$\hat{\mathbf{m}} = \sum_{i=1}^K c_i \phi_i = \mathbf{\Phi} \mathbf{c}, \quad (2.10)$$

where the basis $\mathbf{\Phi}$ can be pre-defined or learned. A popular data-driven approach for such problems is known as the singular value decomposition (SVD).

This decomposition can be interpreted as transform-based through unitary transformations. The instability issue can also be analysed through the SVD. The SVD of an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$, is given by the following matrix factorization

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (2.11)$$

where $\top$ denotes the transpose, $\mathbf{U} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{M \times M}$ are unitary matrices, where the column vectors of $\mathbf{U} \in \{\mathbf{u}_i\}_{i=1}^N$ and row vectors of $\mathbf{V}^\top \in \{\mathbf{v}_i^\top\}_{i=1}^M$ are known as the left- and right-singular vectors respectively. The matrix $\mathbf{\Sigma} \in \mathbb{R}^{N \times M}$ is a rectangular diagonal matrix with diagonal elements $\sigma_1 \geq \dots \geq \sigma_M$, where $\sigma_i \geq 0$ denotes the singular values. The inverse of a square matrix is then defined as

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top. \quad (2.12)$$

Similarly, in the case of non-square matrices ($M \neq N$), the inverse of $\mathbf{A}$ reads

$$\mathbf{A}^\dagger = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{V}(\mathbf{\Sigma}^\top \mathbf{\Sigma})^{-1} \mathbf{\Sigma}^\top \mathbf{U}^\top. \quad (2.13)$$

where $\mathbf{A}^\dagger$ is commonly referred to as the pseudo-inverse, or More-Penrose inverse. Decomposing matrix $\mathbf{G}$ in equation 2.9 gives

$$\hat{\mathbf{m}} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top \mathbf{d} = \sum_{i=1}^N \frac{1}{\sigma_i} \langle \mathbf{u}_i, \mathbf{d} \rangle \mathbf{v}_i, \quad (2.14)$$

Thus, the solution is given as a linear combination of the right singular vectors in $\mathbf{V}$ weighted by the coefficients $\langle \mathbf{u}_i, \mathbf{d} \rangle / \sigma_i$. In relation to equation 2.10, the right singular vectors in $\mathbf{V}$ corresponds to the basis matrix and the left singular vectors $\mathbf{u}_i$ perform a change of basis into its transform coefficients, i.e.,

$$\hat{\mathbf{m}} = \mathbf{V}\mathbf{c}(\mathbf{d}), \quad (2.15)$$

where $\mathbf{c}(\mathbf{d})$ are the coefficients from the unitary transform of $\mathbf{d}$ weighted by the inverse of the singular values. We see that problems arise when the singular values rapidly decreases towards zero, and faster than $|\mathbf{u}_i^\top \mathbf{d}|$. If a square integrable solution $\mathbf{m}$ to the integral equation 2.5 exist, $\mathbf{d}$ must satisfy [30]

$$\|\mathbf{m}\|_2 = \sum_{i=1}^{\infty} \left(\frac{1}{\sigma_i} \langle \mathbf{u}_i, \mathbf{d} \rangle \right)^2 < \infty, \quad (2.16)$$

which is also known as the Picard condition. From this definition $|\mathbf{u}_i^\top \mathbf{d}|$ must decay faster to zero than the singular values σ_i , which is not the case in the presence of noise, as shown in Figure 2.8.

Figure 2.8: The singular values decay rapidly towards zero, which gives an unstable inverse.

Another related inverse problem, is the approximation of $\mathbf{d}$ given only a subset of its measurements, i.e.,

$$\mathbf{d}_s = \mathbf{S}(\mathbf{G}\mathbf{m} + \boldsymbol{\epsilon}) = \mathbf{S}\mathbf{m}_d + \boldsymbol{\epsilon}_s, \quad (2.17)$$

where $\mathbf{m}_d \in \mathbb{R}^N$ defines a model of the complete data and the subsampling operator $\mathbf{S} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ projects the data $\mathbf{d}_s \in \mathbb{R}^M$ onto a lower-dimensional subspace. Note, that the subsampling operator $\mathbf{S}$ is well defined with singular values $\sigma_i = 1$ for $i = 1, \dots, M$ and $\sigma_i = 0$ for $i > M$, thus the operator is singular and its inverse $\mathbf{S}^{-1}$ does not exist, making the problem ill-posed. A solution to equation 2.17 can be provided by the use of a proper set of basis functions and corresponding coefficients, as in equation 2.10. However, the Picard condition reveals that the unbounded inverse operator needs to be replaced by a bounded operator, in order to stabilize the solution. This is the goal of regularization.

2.0.3 Regularization methods

The process of regularization adds information that constrains the solution space, in order to solve an ill-conditioned problem. These constraints commonly consists of enforcing an upper bound on the norm of the solution. These norms are defined for a constant $p \in [1, \infty]$ as

$$R(\mathbf{m})_{\ell_p} = \|\mathbf{m}\|_p = \begin{cases} \left(\sum_{i=1}^n |m_i|^p \right)^{\frac{1}{p}}, & \text{for } p \in [1, \infty) \\ \max_{i=1, \dots, n} |m_i|, & \text{for } p = \infty. \end{cases} \quad (2.18)$$

where $\|\cdot\|_{p=2}$ denotes the Euclidean norm, also referred to as ℓ_2 -norm. A mathematically convenient approach is to introduce the square of the ℓ_2 -norm, which is defined from 2.18 as

$$R(\mathbf{m})_{\ell_2} = \|\mathbf{m}\|_2^2 = \sum_{i=1}^n m_i^2 = m_1^2 + m_2^2 + \dots + m_n^2. \quad (2.19)$$

A common practice for solving ill-posed problems is to define an objective function, that consists of a misfit functional in addition to a suitable regularization penalty. A well-known and historical regularization technique is known as *Tikhonov* regularization [77, 78] —also sometimes referred to as *Tikhonov-Phillips* regularization [62]. In the presence of noise, a common practice is to use a least squares misfit function, which seeks to minimize the sum of squared differences between the forward model and the observed data. The idea of *Tikhonov* regularization is to introduce an operator Γ that gives preference to a specific behaviour of the linear combination between Γ and $\mathbf{m}$ constrained through an upper bound on its ℓ_2 -norm. In the *Tikhonov* regularized inverse, we therefore seek to minimize an objective function on the form

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d} - \mathbf{G}\mathbf{m}\|_2^2 + \frac{1}{2} \|\Gamma\mathbf{m}\|_2^2. \quad (2.20)$$

where the minimization objective has become a compromise between the minimization of the squared misfit and the penalty term. Equation 2.20 has a unique solution given by the following closed form

$$\hat{\mathbf{m}} = (\mathbf{G}^\top \mathbf{G} + \Gamma^\top \Gamma)^{-1} \mathbf{G}^\top \mathbf{d}, \quad (2.21)$$

which is also referred to as the regularized solution of the *normal equations* in statistical learning theory and ML. To see that this condition may stabilize the solution, let $\Gamma = \sqrt{\lambda} \mathbf{I}$, where $\lambda \geq 0$ is a regularization parameter that defines the trade-off between the penalty and misfit functional. Then, by introducing the SVD matrices from equation 2.11, equation 2.21 can be rewritten as

$$\hat{\mathbf{m}} = (\Sigma \Sigma^\top + \lambda \mathbf{I})^{-1} \mathbf{V} \Sigma^\top \mathbf{U}^\top \mathbf{d} = \sum_{i=1}^N \frac{\sigma_i}{\sigma_i^2 + \lambda} \langle \mathbf{u}_i, \mathbf{d} \rangle \mathbf{v}_i. \quad (2.22)$$

In this case, the inverse is stabilized through a shrinkage effect on the singular values in the numerator, and for a suitable choice of λ , the Picard condition is met. Figure 2.9 shows the effect ℓ_2 -norm regularization has on this problem. We see that the peaks of the solution are correctly positioned at the spikes, but the magnitudes are small and the behaviour is too smooth.

Other types of operators for the *Tikhonov* matrix exist. For instance, derivative operators are known for penalizing higher frequencies, such as the choice $\Gamma = \sqrt{\lambda} \nabla^2$, where ∇^2 denotes a second-order gradient. Gradients of different order are commonly approximated by finite-difference methods. Penalizing with the second-order derivative enforces a roughness penalty, and by setting λ to a large value constrains the solutions to being close to linear in $\mathbf{m}$. Thus, derivative operators act as high-pass filters and the identity operator is by definition an all-pass filter. In some cases, we may be interested in non-smooth solutions. In equation 2.18, when $p = 1$ we have the ℓ_1 -norm constrain, defined as

$$R(\mathbf{m})_{\ell_1} = \|\mathbf{m}\|_1 = \sum_{i=1}^n |m_i| = |m_1| + |m_2| + \dots + |m_n|. \quad (2.23)$$

Figure 2.9: *Tikhonov* inverse using the identity operator for the *Tikhonov* matrix with a regularization parameter $\lambda = 0.1$.

where this norm promotes sparse solutions. However, ℓ_1 type of regularizers do not have a closed form solution and one needs to rely on gradient-based optimization algorithms, or sparse coding such as basis pursuit [13], also referred to as LASSO in statistical learning theory [31]. The LASSO minimization problem has the following objective function

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d} - \mathbf{G}\mathbf{m}\|_2^2 + \lambda \|\mathbf{m}\|_1. \quad (2.24)$$

Penalizing $\mathbf{m}$ with the ℓ_1 -norm gives preference to a sparse behaviour. In Figure 2.10 we see that LASSO gives a near perfect fit, when comparing the approximated model to the true model.

In some cases, one may be interested in other types of solutions, such that contain sharp edges. The total-variation (TV) regularization for instance,

Figure 2.10: Inverse using LASSO with a regularization parameter $\lambda = 0.001$.

introduced by Rudin et al. [68], has proved to be quite efficient for preserving edges and boundaries of objects of interest. The idea is not to add a sparseness constrain directly on the model, but through a gradient sparse constrain using derivative operators. The TV-norm is defined for an arbitrary bounded real valued function $f(m)$, and if f is differentiable, the TV-norm generalizes to [67]

$$\text{TV}(\mathbf{m}) = \int_{\Omega} |\nabla f(m)| dm \quad (2.25)$$

where Ω is an open subset defined on $\mathbb{R}^n$ and ∇ denotes the gradient. For regularization purposes I will consider the discrete anisotropic TV-norm, which is defined as

$$R(\mathbf{m})_{\text{TV}} = \frac{1}{N} \sum_{i=1}^N \sum_j \alpha_j |[\mathbf{D}_j \mathbf{m}]_i|, \quad (2.26)$$

where $\mathbf{D}_j$ is a first-order finite-difference operator acting on the coordinate j , e.g. time coordinate, and the regularization parameter α_j defines the trade-off between the TV penalty and the misfit function. Introducing $\mathbf{m}$ to be piece-wise constant in the time coordinate $j = t$, the TV regularized objective function reads

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d} - \Psi \mathbf{m}\|_2^2 + \frac{1}{N} \sum_{i=1}^N \alpha_t |[\mathbf{D}_t \mathbf{m}]_i|, \quad (2.27)$$

where $\Psi = \mathbf{G} \mathbf{D}_t$ is the derivative of the Gaussian kernel. The logic behind equation 2.27 is that, if we consider $\mathbf{m}$ to be piece-wise constant, then $\mathbf{D}_t \mathbf{m}$ is sparse, and the penalty term in equation 2.27 therefore promotes gradient sparse solutions through the ℓ_1 -norm penalty.

Figure 2.11: Inverse using the TV-norm with a regularization parameter $\alpha = 0.08$.

However, the first-order TV-norm has its shortcomings, such as staircasing artifacts, which we can observe around 1 second on the piece-wise linear part of the model in Figure 2.11. Generalizations to the TV-norm have been proposed, such as the total generalized variation [7], which generalizes the TV-norm for higher-order derivatives. The second order total-variation norm, for instance, is

2. Inverse problems and regularization

commonly used to mitigate stair-casing artifacts, by combining the first- and second order [58, 59]. The second-order anisotropic TV-norm can be defined as

$$R(\mathbf{m})_{\text{TV}^2} = \frac{1}{N} \sum_{i=1}^N \sum_j \sum_k \alpha_{jk} |[\mathbf{D}_{jk}\mathbf{m}]_i| \quad (2.28)$$

where $\mathbf{D}_{jk}$ is the second-order finite-difference operator acting on the coordinates j and k , and the regularization parameter α_{jk} defines the trade-off. Introducing the first- and second-order TV regularization the objective function now reads (1D time-domain example)

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d} - \Psi\mathbf{m}\|_2^2 + \frac{1}{N} \sum_{i=1}^N \alpha_t |[\mathbf{D}_t\mathbf{m}]_i| + \frac{1}{N} \sum_{i=1}^N \alpha_{tt} |[\mathbf{D}_{tt}\mathbf{m}]_i|. \quad (2.29)$$

In Figure 2.12, we see that the stair-casing around 1.0 s is penalized towards piece-wise linear behaviour, while keeping piece-wise constant behaviour in the rest of the model.

Similar to the example above, we can try solving for $\mathbf{d}_s = \mathbf{S}\mathbf{m}_d + \boldsymbol{\varepsilon}_s$ using objective functions as given in equations 2.20, 2.24 or 2.27 or objective functions that combine different norms, such as the elastic net regularization [31]. Since $\mathbf{d}_s$ is sufficiently smooth, we could utilize equation 2.20 in which we would like to minimize the following objective function

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d}_s - \mathbf{S}\mathbf{m}_d\|_2^2 + \frac{1}{2} \|\Gamma\mathbf{m}_d\|_2^2. \quad (2.30)$$

If we employ the identity operator for the *Tikhonov* operator Γ , it gives the following closed form solution

$$\hat{\mathbf{m}}_d = (\mathbf{S}^\top \mathbf{S} + \lambda \mathbf{I})^{-1} \mathbf{S}^\top \mathbf{d}_s. \quad (2.31)$$

Figure 2.12: Regularization with the combined first- and second-order TV inverse with a regularization parameter for the first-order TV term $\alpha = 0.08$ and $\beta = 0.01$ for the second order TV term.

Figure 2.13 (middle) show an example using a *Tikhonov* inverse of the subsampling operator using an identity operator, where we see that the inverse is stable, but another important and desired property is not met; accuracy. In this example, the null-space of the identity operator and the subsampling operator intersect trivially (zero vector). Thus, a simple ℓ_2 -norm on the model gives no desired smoothing of the missing data. Instead, we can introduce the second-order finite difference operator (Figure 2.13(right)) as the *Tikhonov* operator, often referred to as roughness penalty [15, 31], i.e.,

$$\hat{\mathbf{m}}_d = (\mathbf{S}^\top \mathbf{S} + \lambda \mathbf{D}_{tt}^\top \mathbf{D}_{tt})^{-1} \mathbf{S}^\top \mathbf{d}_s. \quad (2.32)$$

The roughness penalty represents a more suitable regularizer, since the missing values contributes to high frequencies that does not exist within the signal, in which the roughness penalty enforce both stability and accuracy.

Figure 2.13: The Tikhonov inverse with identity operator (middle) and the second-order finite-difference operator (right). Both examples were inverted using a regularization penalty of $\lambda = 0.1$.

Similarly, we can search for approximate solutions through the use of basis functions, e.g., equation 2.10, where for some proper choice of basis, the corresponding weighting coefficients are sparse and can be solved using the LASSO algorithm, i.e.,

$$\mathcal{C} = \frac{1}{2} \|\mathbf{d}_s - \mathbf{S}\Phi\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (2.33)$$

where $\mathbf{m}_d$ is approximated through the sparsity promoting basis $\hat{\mathbf{m}}_d = \Phi\hat{\mathbf{c}}$. From a probabilistic point of view, the penalty function corresponds to a log-prior distribution, the objective function corresponds to a log-posterior distribution, and the misfit function as the likelihood [31]. From this perspective, the solution follows a maximum a posteriori estimation (MAP) from which one maximizes a point estimate of the posterior distribution. Thus, the penalty function includes a priori information on the likelihood of the solution.

These examples illustrates the importance to have insight and knowledge about the given problem, data and model behaviour in order to stabilize the inverse and to obtain a proper solution. In many circumstances, this also applies for the use of DL in inverse problems. For more thorough analysis on data-driven

2. Inverse problems and regularization

techniques applied to inverse problems, see Arridge et al [4] with examples therein.

Chapter 3

Deep learning for inverse problems

3.1 The deep learning model

3.1.1 Fully connected neural network

Where traditional approaches employ domain-knowledge and careful engineering of some penalty functional to acquire the desired solution, the deep learning (DL) framework make predictions by learning from data. The DL models, also known as neural networks (NNs), try to learn a generalized functional relationship between the input and the desired solution. NNs are loosely based on and inspired by biological systems and how neurons in the human brain process information. Since DL is meant to generalize the inverse map, training the DL model involves large scale computations of a considerable amount of labeled training examples.

There are mainly two problems considered in DL: (1) classification problems and (2) regression problems. In classification, the task can be; given a set of input images where some contain cats and some does not. The output (or goal) of the computational model is either provide an answer yes/no if there is a cat in the given input image, where the NN has (optimally) learned some sort of pattern that are unique to cats. Regression problems however, are not this categorical. In regression, we try to estimate continuous variables, such as the models we have seen in the former section. Inverse problems, in this case, is a regression problem. In this context, consider the observed data represented by the vector $\mathbf{d} \in \mathbb{R}^M$ and the object of interest by the vector $\mathbf{m} \in \mathbb{R}^N$, then we aim to solve for the inverse of the following forward problem

$$\mathbf{d} = \mathbf{S}\mathbf{m} + \boldsymbol{\varepsilon}, \quad (3.1)$$

where $\mathbf{S} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is the forward operator and $\boldsymbol{\varepsilon}$ denotes the noise. Now, in case we have access to the true model $\mathbf{m}$ —or optimally a large data set with different but related models— we can then try to find a functional relation between $\mathbf{d}$ and $\mathbf{m}$ that minimizes a residual ϵ between the true model $\mathbf{m}$ and a function $f(\cdot)$ of the input $\mathbf{d}$, i.e.,

$$\epsilon = \mathbf{m} - f(\mathbf{d}; \mathbf{W}, \mathbf{b}), \quad (3.2)$$

where $f(\cdot; \mathbf{W}, \mathbf{b})$ defines an arbitrary NN, and the parameters $\mathbf{W}$ and $\mathbf{b}$ are commonly referred to as the weight and bias parameters respectively. The main goal is then to find/learn the function $\mathbf{m}_f = f(\cdot; \mathbf{W}, \mathbf{b})$, which plays the role of a generalized inverse of $\mathbf{S}$ for new and representative data. However, both due to

3. Deep learning for inverse problems

observation noise and computational limitations, we will not be able to fit $\mathbf{m}_f$ exactly to $\mathbf{m}$, but rather seek an approximation $\hat{\mathbf{m}}_f : \mathbb{R}^M \rightarrow \mathbb{R}^N = \hat{f}(\cdot; \hat{\mathbf{W}}, \hat{\mathbf{b}})$ parameterized by the learned parameters $\hat{\mathbf{W}}$ and $\hat{\mathbf{b}}$. In the simplest form, $\mathbf{m}_f$ is denoted by a L -layered function, with $L - 1$ layers of linear affine maps and non-linear transformations, which can be defined as

$$\mathbf{m}_f = \varrho(\mathbf{W}_{L-1}\rho(\dots\rho(\mathbf{W}_2\rho(\mathbf{W}_1\mathbf{d} + \mathbf{b}_1) + \mathbf{b}_2)\dots) + \mathbf{b}_{L-1}), \quad (3.3)$$

where $\rho(\cdot)$ and $\varrho(\cdot)$ denote component-wise transformations —commonly referred to as activation functions—, and $\mathbf{W}_{l+1}$ and $\mathbf{b}_{l+1}$ denote the parameters in an arbitrary layer $l \in \{0, 1, \dots, L - 1\}$. In regression, the activation function $\varrho(\cdot)$ is typically chosen as a linear transform, i.e., the identity function. This implies that the output is just a linear combination of the $L - 1$ weights and the non-linear transformations from the previous layer

$$\mathbf{m}_f = \mathbf{W}_{L-1}\boldsymbol{\rho}(\mathbf{d}) + \mathbf{b}_{L-1}, \quad (3.4)$$

where $\boldsymbol{\rho}(\mathbf{d})$ denote the coefficients of the last layer —or features— defined from the hierarchy of linear and non-linear transformations

$$\boldsymbol{\rho}(\mathbf{d}) = \rho(\dots\rho(\mathbf{W}_2\rho(\mathbf{W}_1\mathbf{d} + \mathbf{b}_1) + \mathbf{b}_2)\dots). \quad (3.5)$$

Such classes of NN are often referred to as fully connected neural networks (FCNN), or multilayer feedforward perceptrons [63]. FCNNs has a minimum of three layers; input, hidden and output layer. Consider a single hidden layer DNN with four hidden units, that takes two data samples as input and output. Excluding the bias parameter, this FCNN will have the simple matrix multiplication structure

Figure 3.1: A single hidden layer DNN, composed of two matrix multiplications and an activation on the hidden units.

$$\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} w_{11}^{[2]} & w_{12}^{[2]} & w_{13}^{[2]} & w_{14}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} & w_{23}^{[2]} & w_{24}^{[2]} \end{pmatrix} \rho \left(\begin{pmatrix} w_{11}^{[1]} & w_{12}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} \\ w_{31}^{[1]} & w_{32}^{[1]} \\ w_{41}^{[1]} & w_{42}^{[1]} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \right), \quad (3.6)$$

or the alternative form

$$\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} w_{11}^{[2]} & w_{12}^{[2]} & w_{13}^{[2]} & w_{14}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} & w_{23}^{[2]} & w_{24}^{[2]} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}. \quad (3.7)$$

Thus, the hidden layer feature, or coefficients, are given first by a linear transform $\mathbf{z} = \mathbf{W}_1 \mathbf{d}$ followed by the activation $\mathbf{a} = \rho(\mathbf{z})$. Then the output is given by a linear combination between the non-linear transform-coefficients and the last weight matrix. A simple illustration of this single-hidden layer DNN is depicted in Figure 3.1.

Compared to equation 2.15, the transform-coefficients in the DNN are composed of a set of non-linear transformations, in which deep architectures can give rise to highly complex feature separation and correlations in non-linear space. Thus, the DL model may provide a richer building block of features that may not be separable in linear space. As for the non-linearities, some of the most frequently used activation functions are the rectified linear unit (ReLU), leaky ReLU (lReLU), sigmoid and tanh, which are defined as

$$\begin{aligned} \text{ReLU} &= \max(\mathbf{z}, 0), \quad \text{lReLU} = \max(\mathbf{z}, a\mathbf{z}), \\ \text{sigmoid} &= \frac{1}{1 + e^{-\mathbf{z}}}, \quad \tanh = \frac{e^{\mathbf{z}} - e^{-\mathbf{z}}}{e^{\mathbf{z}} + e^{-\mathbf{z}}}, \end{aligned} \quad (3.8)$$

where $0 < a < 1$ is a user-defined hyperparameter defining the slope of the negative arguments. Choosing the “right” type of activation function is more or less considered to be a trial and error task, where one settles for the activation function that makes the model learn and gives good prediction performance. However, in many applications the ReLU function is regarded as the most popular choice of activation function [64], due to some desired properties, such as the close connection to sparse coding [60], improved gradient propagation and scale invariance.

3.1.2 Convolutional neural network and multidimensional data

The convolutional neural network (CNN) is a special type of NN architecture. Compared to the DNN, which has full connection between the arbitrary weight matrices and the input vectors, the CNN architecture instead constrains the linear operations to involve convolutions. The computations within CNNs therefore incorporate knowledge of local correlations and invariance of shapes and feature

3. Deep learning for inverse problems

position, and impose a constrain on the learnable parameters [45]. The CNNs are arguably the most common architecture for multi-dimensional data with structural patterns. The image-based CNNs consist of 2D convolution operations performed on 3D objects. In an arbitrary layer l , the k^{th} convolution from one layer to the next is defined as

$$(\mathbf{w}_{k,l+1} * \mathbf{a}_l)_{y,t} = \sum_{i=1}^{c_l} \sum_{u=-f/2}^{f/2} \sum_{v=-f/2}^{f/2} w_{u,v,i,k}^{[l+1]} a_{y-u,t-v,i}^{[l]}, \quad (3.9)$$

where the indexes (y, t) and (u, v) define the sample position and kernel position, respectively. The convolution operation in equation 3.9 can be considered a special case of convolution, where the k^{th} output is computed as a sum of 2D convolutions over the input features $i = 1, \dots, c_l$, i.e., one convolution kernel for each input feature. This is illustrated in Figure 3.2 for a 2D CNN and a 3D CNN. In 3D CNNs, the convolution kernels are extended with one dimension and the feature space now becomes 4D objects, i.e.,

$$(\mathbf{w}_{k,l+1} * \mathbf{a}_l)_{x,y,t} = \sum_{i=1}^{c_l} \sum_{z=-f/2}^{f/2} \sum_{u=-f/2}^{f/2} \sum_{v=-f/2}^{f/2} w_{z,u,v,k,i}^{[l+1]} a_{x-z,y-u,t-v,i}^{[l]}. \quad (3.10)$$

The k^{th} output feature is then superimposed with the bias parameter in each pixel (2D) or voxel (3D) position, i.e.,

$$\mathbf{z}_{k,l+1} = (\mathbf{w}_{k,l+1} * \mathbf{a}_l)_{y,t} + b_{k,l+1}. \quad (3.11)$$

Thus, there are as many bias parameters as there are features in the next layer. The number of weights are dependent on the kernel size and the number of

Figure 3.2: Simplified illustration of the convolution operation on an image-based 2D input with 3D objects (left) and a volume-based 3D input with 4D objects (right). Here, the two features $\mathbf{a}_1$ and $\mathbf{a}_2$ are convolved with 2D/3D kernels to give the linear transform $\mathbf{z}_1$. A different set of 2D/3D kernels are then used to compute the linear transforms $k = 2, \dots, c_{l+1}$.

input and output features. To simplify notations and further discussion, I will use notations with matrices and vectors as before. In this case, I will write the non-linear output simply as the product

$$\mathbf{a}_{l+1} = \rho(\mathbf{W}_{l+1}\mathbf{a}_l + \mathbf{b}_{l+1}) = \rho(\mathbf{z}_{l+1}). \quad (3.12)$$

To train deep CNN's, a common practice is to introduce residual layers, where activations from a previous layer are skipped over one layer and then added to the next (single skip), i.e. ResNet [32]. Single skip can be explained by introducing the $l - 1$ activation

$$\mathbf{a}_{l-1} = \rho(\mathbf{W}_{l-1}\mathbf{a}_{l-2} + \mathbf{b}_{l-1}), \quad (3.13)$$

and the next activation

$$\mathbf{a}_l = \rho(\mathbf{W}_l\mathbf{a}_{l-1} + \mathbf{b}_l). \quad (3.14)$$

The single skip layer is then given by

$$\mathbf{a}_{l+1} = \rho(\mathbf{W}_{l+1}\mathbf{a}_l + \mathbf{b}_{l+1} + \mathbf{a}_{l-1}) = \rho(\mathbf{z}_{l+1} + \mathbf{a}_{l-1}). \quad (3.15)$$

This allows training deeper networks because optimizing the residual is considered easier than a plain stack of non-linear layers [32]. Networks with residual layers also have the property of truncating the least significant subspace [92].

Many different convolutional architectures have been proposed. One of the popular architectures for inverse problems is based on an encoder-decoder (ED) structure, such as the U-net architecture [66]. Commonly, EDs are explained through two functional transitions, where the encoder transition $\zeta = \phi(\mathbf{d}_s)$ maps the input space to a latent representation, and the decoder transition $\mathbf{m}_f = \psi(\zeta)$ maps the latent representation to the output. If the encoder maps the input onto a lower dimensional *Hilbert* space, then the ED is considered undercomplete. If the encoder maps the input onto a higher dimensional *Hilbert* space, it is considered overcomplete. In ED networks, the features are commonly downsampled, which can be done through fixed operators, such as average- or max-pooling, and defined in 1D as the following matrices

$$\Phi_{ave} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \quad (3.16)$$

and

$$\Phi_{max} = \begin{pmatrix} b_1 & 0 & \cdots & 0 \\ 1-b_1 & 0 & \cdots & 0 \\ 0 & b_2 & \cdots & 0 \\ 0 & 1-b_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_{N/2} \\ 0 & 0 & \cdots & 1-b_{N/2} \end{pmatrix}, \quad (3.17)$$

where the max-pooling parameters $\{b_i\}_{i=1}^{N/2}$ are binary numbers (0,1) determined by the signal statistics [92]. A potential drawback is that pooling operators are non-unitary transformations. Consider the simple case, by applying average pooling to an arbitrary feature $\mathbf{a}_l \in \mathbb{R}^4$, i.e.,

$$\mathbf{z}_l = \Phi_{ave}^\top \mathbf{a}_{l-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1^{[l-1]} \\ a_2^{[l-1]} \\ a_3^{[l-1]} \\ a_4^{[l-1]} \end{pmatrix} \quad (3.18)$$

where the pooling operator $\Phi_{ave} : \mathbb{R}^4 \rightarrow \mathbb{R}^2$ projects the input $\mathbf{z}_l$ onto a lower-dimensional subspace. The unpooling operator gives the following linear transform

$$\begin{aligned} \mathbf{z}_{l+1} = \Phi_{ave} \mathbf{z}_l &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1^{[l-1]} \\ a_2^{[l-1]} \\ a_3^{[l-1]} \\ a_4^{[l-1]} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1^{[l-1]} \\ a_2^{[l-1]} \\ a_3^{[l-1]} \\ a_4^{[l-1]} \end{pmatrix} \end{aligned} \quad (3.19)$$

Here, the features $\mathbf{z}_{l+1}$ are considered low-rank approximations of $\mathbf{a}_{l-1}$, since the matrix product $\Phi_{ave} \Phi_{ave}^\top$ spans $\mathbb{R}^2$, with $\Phi_{ave}^\top \Phi_{ave} = \mathbf{I}_2$. The same logic applies for max pooling. Thus, if pooling is employed in reconstruction applications, valuable information may be lost in the forward pass of the network. Alternative to pooling, are strided convolutions, in which the pixel-distance between the consecutive convolutional kernel position are greater than 1. Thus, strided convolutions represents learned downsampling operations.

The advantage of downsampling is that the kernel size in the consecutive layers increases relative to the feature size, which increases the networks receptive field. Introducing strided convolutions for downsampling, the receptive field increases exponentially compared to linearly when no downsampling is employed.

However, it also implies that the number of filters need to increase exponentially to preserve overcompleteness. To upsample the features, the decoder introduces either fixed unpooling operators, or learned operators through strided transposed convolution, or periodic resampling [71]. Other properties of CNNs such as input size, kernel size, zero padding etc., are also important for network performance. For a thorough guide on CNN arithmetic, see Dumoulin and Visin [18].

3.2 Training deep learning models

3.2.1 Supervised learning basics

The term *supervised* comes from the use of a “teacher” which is provided by the desired target model $\mathbf{m}$. As before, consider the inverse problem $\mathbf{d} = \mathbf{S}\mathbf{m} + \boldsymbol{\varepsilon}$, and that our model $\mathbf{m}_f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ defines an arbitrary NN where $\mathbf{S}$ is considered known. Now, assume we have access to a set of targets $\{\mathbf{m}^{(i)} \in \mathbb{R}^N\}_{i=1}^T$. Thus, we can construct a set of corresponding input data $\{\mathbf{d}^{(i)} \in \mathbb{R}^M\}_{i=1}^T$, and establish training data that consists of the set of target-input pairs $\mathcal{D} = \{(\mathbf{m}^{(i)}, \mathbf{d}^{(i)})\}_{i=1}^T$. To train the DL model, we minimize the misfit between $\mathbf{m}_f$ and the target data $\mathbf{m}$, commonly the mean squared error (MSE), defined as

$$\mathcal{C}(\mathbf{W}, \mathbf{b})_{\ell_2} = \frac{1}{2TN} \sum_{i=1}^T \|\mathbf{m}^{(i)} - f(\mathbf{d}^{(i)}; \mathbf{W}, \mathbf{b})\|_2^2. \quad (3.20)$$

One could also consider other types of misfits, such as the ℓ_1 -norm or the Huber misfit function [31]. For notational convenience, I will write the objective function on a more compact form. With a slight abuse of notation, the objective function becomes

$$\mathcal{C}(\mathbf{W}, \mathbf{b})_{\ell_2} = \frac{1}{2} \|\mathbf{m} - f(\mathbf{d}; \mathbf{W}, \mathbf{b})\|_2^2, \quad (3.21)$$

where the scale factors N and T are implicitly defined within the norm. Now, given an example of an unseen and representative data $\mathbf{d}_u$, we can predict an approximation to the true model $\mathbf{m}$ from the trained function $\hat{\mathbf{m}}_f = \hat{f}(\mathbf{d}_u, \hat{\mathbf{W}}, \hat{\mathbf{b}})$, where $\hat{\mathbf{W}}, \hat{\mathbf{b}}$ denotes the parameters that are learned from for example the minimization of equation 3.21.

When working with images or volumes, in some circumstances the sizes of input/output are different due to computational considerations. In the case of CNN’s, any arbitrary input/output size can be employed, although, when downsampling layers are included the input must be divisible by the stride of the convolutions. In CNN’s employed for super-resolution problems there are commonly two ways to treat the input; (1) same size as output, or (2) same size as subsampled. A common practice in natural image processing, is to upsample the input by either introducing zero values or employing traditional interpolation such as bicubic interpolation [16]. If the input is subsampled at regular intervals, we can instead skip the pre-interpolation and move the upsampling within the network [71].

Figure 3.3: Here, network **A** is designed to map an input to an output of the same size, compared to network **B** which maps directly from the subsampled input to the upsampled output.

If the number of samples varies from one input example to the next, we need to regularize the data. Thus, we need to introduce zero values or some pre-interpolation procedures, such that the ratio between the number of input samples and target samples are constant. For a regularly downsampled input, two examples are illustrated on shot-gathers in Figure 3.3. However, using the downsampled data as input to the NN reduces the computational complexity of the method. In case of irregular sampling, which is common when the data is sorted to the offset domain, zero values are introduced as illustrated in Figure 3.4.

Once we have gathered training data, determined the data input/output and the misfit function, and the structure of the NN architecture, we can try to learn the desired mapping function. This is performed by solving an optimization problem.

Figure 3.4: In the offset-domain, the number of observed samples varies from one example to another. The input therefore needs to be upsampled to the same size as the output. In this example, this was performed with binning by introducing zero values between the observed data.

3.2.2 The optimization problem

Minimizing equation 3.21 and learning the NN parameters $\{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{L-1}$ is achieved by solving an optimization problem. In DL, this is most commonly

performed by stochastic first-order gradient based optimization algorithms, where the basic idea is to reach a local minimum by taking relatively small steps in the gradient direction of a differentiable objective function. Let $\mathcal{C}(\mathbf{W}, \mathbf{b})$ denote the function we want to minimize, then the value of $\mathcal{C}(\mathbf{W}, \mathbf{b})$ decrease fastest in the direction of the negative gradient $-\nabla\mathcal{C}(\mathbf{W}, \mathbf{b})$ for the weights and bias parameters. Since optimization of the weights and bias parameters are more or less equal, I will focus on the weights in this section.

Let n denote the iteration steps, then updating the parameters $\mathbf{W}_{l,n+1}$ from the initialized values $\mathbf{W}_{l,n=0}$, is then performed by evaluating $\nabla\mathcal{C}(\mathbf{W}_{l,n})$ at each iteration step $n = 0, 1, \dots$, i.e.,

$$\mathbf{W}_{l,n+1} = \mathbf{W}_{l,n} - \eta \nabla\mathcal{C}(\mathbf{W}_{l,n}). \quad (3.22)$$

where the step size $\eta \geq 0$ is commonly referred to as the learning rate, or global learning rate. It follows, that for a suitable choice of η , then $\mathcal{C}(\mathbf{W}_n) \geq \mathcal{C}(\mathbf{W}_{n+1})$, which implies $\mathbf{W}_{n+1}$ moves closer to a local minima. The first-order gradient descent algorithm is considered an efficient optimization algorithm, since the partial derivatives with respect to all the parameters is actually of the same computational complexity as just evaluating the function [42]. Although, the gradient of $\mathcal{C}(\mathbf{W})$ in NN needs to be evaluated at all layers $l \in \{1, \dots, L-1\}$. The gradient flow through the network is performed efficiently by the algorithm known as backpropagation [69], where gradients in earlier layers are evaluated using the chain rule.

The simplicity of combining first-order gradient algorithms and backpropagation makes optimization of the parameters in NN fairly efficient, even for large scale training when parallelized on GPUs. However, convergence is slow compared to algorithms that incorporate knowledge about the second-order derivative, i.e., information about the curvature of the differentiable function, and thus needs fewer steps to converge. Although, first-order gradient-based algorithms can be significantly improved without sacrificing simplicity, by introducing additional terms that store the information about the past gradients. One of these historical algorithms is stochastic gradient descent with momentum, which keep track of the first moment of the gradient and therefore takes into account the weight change from the previous steps [69]. A similar algorithm, is the Nesterov's accelerated momentum [57], which has shown improved convergence rate in some situations [76]. One of the more recent optimization algorithms, and arguably one of the most popular, is the Adaptive Moment Estimation (Adam) algorithm [42], which combines the gradient descent with momentum with an algorithm known as RMSprop [34]. Adam is defined as

$$\mathbf{v}_{l,n} = \beta_1 \mathbf{v}_{l,n-1} + (1 - \beta_1) \nabla\mathcal{C}(\mathbf{W}_{l,n}), \quad (3.23)$$

$$\mathbf{s}_{l,n} = \beta_2 \mathbf{s}_{l,n-1} + (1 - \beta_2) (\nabla\mathcal{C}(\mathbf{W}_{l,n}))^2, \quad (3.24)$$

$$\hat{\mathbf{v}}_{l,n} = \frac{\mathbf{v}_{l,n}}{1 - \beta_1^{n+1}}, \quad (3.25)$$

3. Deep learning for inverse problems

$$\hat{\mathbf{s}}_{l,n} = \frac{\mathbf{s}_{l,n}}{1 - \beta_2^{n+1}}, \quad (3.26)$$

$$\mathbf{W}_{l,n+1} = \mathbf{W}_{l,n} - \frac{\eta}{\sqrt{\hat{\mathbf{s}}_{l,n} + \epsilon}} \hat{\mathbf{v}}_{l,n}, \quad (3.27)$$

where the second moment (squared gradient) is an element-wise squaring, $\hat{\mathbf{v}}_{l,n+1}$ and $\hat{\mathbf{s}}_{l,n+1}$ are the bias corrected estimation of the first- and second moment, respectively, and β_1 and β_2 are the decay rates for first- and second moment, respectively. Finally, ϵ is a user defined hyperparameter that avoids division by zero. Thus, the Adam optimization algorithm introduces smoothing of highly oscillating gradients and accelerate learning in direction of small gradients.

In some situations, one might need to introduce a regularization term, such as the ℓ_2 -norm penalized function, i.e., $\mathcal{C}(\mathbf{W}) = L(\mathbf{W}) + R(\mathbf{W})$, where $L(\mathbf{W})$ is the misfit and $R(\mathbf{W})$ is the regularization term. In this case, the gradient descent algorithm in equation 3.22 reads

$$\mathbf{W}_{l,n+1} = \mathbf{W}_{l,n} - \eta(\nabla L(\mathbf{W}_{l,n}) + \nabla R(\mathbf{W}_{l,n})). \quad (3.28)$$

where $\nabla R(\mathbf{W}_{l,n})$ are the partial derivatives of the regularization term with respect to the weights. The ℓ_2 regularizer for instance, constrains the solution space, as shown in Figure 3.5 (right). A potential problem arise when regularized

Figure 3.5: The stochastic gradient descent (SGD) algorithm have large oscillations, which makes convergence towards an optimal minimum slower compared to the momentum based (SGDM). In addition, with weight decay one constrain the solution space in order to avoid overfitting at the global minimum. This simple example however, show two parameters w_1 and w_2 , on a convex objective function, while in DL one often optimize for potentially millions of parameters within an objective functions surface that is nonconvex.

objective functions are optimized with adaptive gradient methods, such as Adam, since the gradient of the regularization penalty is scaled along with the gradient of the misfit term. This may lead to insufficient regularization of the parameters that have large gradients and may therefore also have negative effects on generalization performance [50]. To fix this, one can decouple the regularization term and the misfit term by using the adaptive methods on the misfit only. In this case, the decoupled weight decay [50] modifies equation 3.27 to

$$\mathbf{W}_{l,n+1} = \mathbf{W}_{l,n} - \eta \left(\frac{1}{\sqrt{\hat{\mathbf{s}}_{l,n}} + \epsilon} \hat{\mathbf{v}}_{l,n} + \lambda \mathbf{W}_{l,n} \right), \quad (3.29)$$

where $\lambda \mathbf{W}_{l,n}$ term is derived from the derivative of the ℓ_2 -norm regularization penalty, and λ denotes the regularization parameter.

3.2.3 Model selection

DL models tend to be over-parameterized, and may overfit the solution at the global minimum. This means that our trained model $\hat{f}(\cdot)$ explains well the data that it has been fitted to, but does not generalize to new and representative data. In statistical learning theory, a well known property of statistical models known as the bias-variance trade-off, is used to analyse the concept of overfitting. The bias-variance trade-off is defined as [31]

$$\begin{aligned} \mathbb{E}(\mathcal{C}(\mathbf{W}, \mathbf{b})) &= \sigma^2 + Bias^2(\hat{f}(\mathbf{d})) \\ &\quad + Var(\hat{f}(\mathbf{d})). \end{aligned} \quad (3.30)$$

where $\mathbb{E}$ denotes the expected error and σ^2 denotes the irreducible error, i.e., the variance in the data. The second term $Bias^2(\hat{f}(\mathbf{d}_s))$ denotes the model bias, defined as

$$Bias^2(\hat{f}(\mathbf{d}_s)) = \sum_i (f(\mathbf{d}_s) - \mathbb{E}(\hat{f}(\mathbf{d}_s)))^2, \quad (3.31)$$

which is a measurement of the discrepancy between the expectation of the trained model and the truth $f(\mathbf{d})$.

A large bias means that we do not have enough variation in the prediction in order to explain the data. The third term $Var(\hat{f}(\mathbf{d}))$ denotes the model variance, defined as

$$Var(\hat{f}(\mathbf{d})) = \mathbb{E}[(\hat{f}(\mathbf{d}) - \mathbb{E}(\hat{f}(\mathbf{d})))^2]. \quad (3.32)$$

which is a measure of the variance of the model. A large variance could imply that the function $\hat{f}(\cdot)$ are explaining the data it has been trained on too well. Intuitively, the bias-variance trade-off implies that we actually desire a model that yields simultaneously low variance and low bias. The key idea is that one can reduce variance by increasing the bias. In DL, a common practice is to

3. Deep learning for inverse problems

Figure 3.6: A schematic display of cross-validation, where in this example a test set (randomly extracted) of size 5% serves as a hold-out set for final performance. Here the remaining $k > 100$ observations were further split into 10% and 85% of the total number of examples. For a K-fold ($K > 1$) cross-validation one runs the training on all the different K-folds, and compute the average expectation error.

increase bias by regularization, such as introducing the ℓ_2 -norm penalty to the weights. In this case, the objective function reads

$$\mathcal{C}(\mathbf{W}, \mathbf{b})_{\ell_2} = \frac{1}{2} \|\mathbf{m} - f(\mathbf{d}; \mathbf{W}, \mathbf{b})\|_2^2 + \frac{\lambda}{2} \|\mathbf{W}\|_2^2, \quad (3.33)$$

where $\lambda \geq 0$ is the regularization parameter. Finding an appropriate value for the regularization parameter is commonly done through model selection methods, such as K-fold cross-validation or bootstrap [31]. The number of folds to use in cross-validation are dependent on available computational resources and/or size of the model. In DL the data sets are commonly large, so that 1-fold cross-validation is sufficient to include significant statistics. The 1-fold cross-validation is done by first splitting the training data $\mathcal{D}$ into three sets: training set $\mathcal{D}_{train}$, a validation set $\mathcal{D}_{val}$ and a test set $\mathcal{D}_{test}$, as depicted in Figure 3.6. Model-selection is then done by fitting the model to the $\mathcal{D}_{train}$ and computing the expectation error on the validation set, also referred to as out of sample error ($\mathbb{E}_{out}$ in Figure 3.6). The optimal regularization parameter λ then corresponds to the value that gives the lowest expectation error on the validation set, as depicted in Figure 3.7. The test set in this case, only serves as a hold out data set to compute the performance for the final model.

In DL, many different approaches have regularization effect on the model, such as data augmentation, early stopping, ensemble methods or even noise contamination [24]. In DNNs, a regularization technique known as dropout [74] is considered one of the most effective. However, dropout is not widely used in CNNs, most likely due to feature coefficients in close proximity are likely to be correlated and thus dropout of feature coefficients may be detrimental to the learning process.

Figure 3.7: The training data set is split into three different sets: $\mathcal{D}_{train}$, $\mathcal{D}_{val}$ and training set $\mathcal{D}_{test}$. The in-sample error E_{in} , computed on $\mathcal{D}_{train}$, decreases with increasing model complexity. One can decrease model complexity by increasing the λ value. This Figure is adapted from Mehta et al [55].

3.2.4 Transform-domain learning

We have seen that the linear transformations in DL can also be performed by predefined operators, e.g., average- and max pooling. Alternative to pooling operators, are predefined analytical basis functions. Signal representation theory using compact analytic functions, such as wavelets, shearlets, curvelets and framelets have been an active research area for many years [26, 27, 33, 37, 46, 52, 54, 72, 94]. The power of these compact basis functions, in comparison with Fourier basis, is their ability to represent non-stationary signals [22]. Some have also attempt to design such basis functions specifically for seismic data, i.e., seislets [21, 22]. Closely related to DL, Mallat [53] and Bruna and Mallat [8] proposed the wavelet scattering network which can be consider analogous to CNNs, but without the learning component.

The generalization of a basis for instance, also referred to as framelets — introduced by Duffin and Schaeffer [17]—, are related to filters and filter banks [12], and are therefore also related to the linear transformations in CNNs. A subgroup of frames, known as finite frames are in general redundant, hence, frames have more vectors than the dimension of the *Hilbert* space. Consider a simple single-layer ED network with an arbitrary input vector $\mathbf{x} \in \mathbb{R}^N$, which can be represented by the following transformation

$$\hat{\mathbf{x}} = \hat{\mathbf{W}}_2 \rho(\hat{\mathbf{W}}_1 \mathbf{x}), \quad (3.34)$$

where $\hat{\mathbf{W}}_1$ and $\hat{\mathbf{W}}_2$ defines the learned model weights, and ρ represents the non-linear activation function. Similarly, we can write a frame expansion, first as the redundant linear combination into its expansion coefficients

3. Deep learning for inverse problems

$$\mathbf{c} = \Phi^\top \mathbf{x}, \mathbf{c} \in \mathbb{R}^K, \quad (3.35)$$

where $\Phi^\top \in \{\phi_i^\top \in \mathbb{R}^N\}_{i=1}^K$ defines a redundant set of basis vectors, i.e., $K > N$. The reconstruction of $\mathbf{x}$ is then given by

$$\mathbf{x} = \tilde{\Phi} \mathbf{c} = \tilde{\Phi} \Phi^\top \mathbf{x} = \mathbf{F} \mathbf{x}, \quad (3.36)$$

where $\tilde{\Phi}$ satisfies the frame condition

$$\tilde{\Phi} \Phi^\top = \mathbf{I}. \quad (3.37)$$

and $\mathbf{F}$ is commonly referred to as the frame operator [12]. If for instance the basis $\Phi^\top$ defines a suitable sparsifying transform of $\mathbf{x}$, we can introduce a thresholding function Γ to remove expansion coefficients with small amplitudes, i.e.,

$$\hat{\mathbf{x}} = \tilde{\Phi} \Gamma(\Phi^\top \mathbf{x}). \quad (3.38)$$

We may also consider further expansion of the coefficients in order to make the frame expansion deeper, thus introducing more complex transformations and feature separations. In this case, thresholded framelet expansions and DL share similar properties. First of all, we want deep networks to properly handle the complexity of the signal, i.e., capture the non-linear relationships hidden within the signal structure. Also, completeness ($K = N$) only allows for a unique reconstruction operator $\tilde{\Phi}$, while redundancy ($K > N$) give infinite number of possibilities for $\tilde{\Phi}$. Thus, redundant transformations avoids the downside of corruption or loss of expansion coefficients that can result in severe reconstruction errors using complete basis. The redundancy of frames have other beneficial properties, such as being robust against noise and gives design flexibility compared to linearly independent basis [12].

In frame theory, in order to meet the formal definition of a frame, the family of vectors $\{\phi_i^\top\}_{i=1}^K$ in a *Hilbert* space $\mathcal{H}^N$, must satisfy the following inequality

$$\alpha \|\mathbf{x}\|^2 \leq \sum_{i=1}^K |\langle \phi_i, \mathbf{x} \rangle|^2 \leq \beta \|\mathbf{x}\|^2, \mathbf{x} \in \mathbb{R}^N, \quad (3.39)$$

where the constants $0 < \alpha \leq \beta < \infty$ are known as the frame bounds. In frame theory the different operators are commonly referred to as the frame operator, analysis operator and the synthesis operator, where the frame operator $\mathbf{F} : \mathcal{H}^N \rightarrow \mathcal{H}^N$ is defined in equation 3.36 where the $K \times N$ matrix

$$\Phi^\top = \begin{pmatrix} - & \phi_1^\top & - \\ - & \vdots & - \\ - & \phi_K^\top & - \end{pmatrix}, \quad (3.40)$$

defines the analysis operator. The $N \times K$ synthesis operator is given explicitly from the pseudo-inverse

$$\tilde{\Phi} = (\Phi\Phi^\top)^{-1}\Phi = \begin{pmatrix} | & | & | \\ \tilde{\phi}_1 & \cdots & \tilde{\phi}_K \\ | & | & | \end{pmatrix}, \quad (3.41)$$

The frame bounds are defined from the eigenvalues of $\Phi\Phi^\top = \mathbf{V}\Sigma^2\mathbf{V}^\top$, where $\alpha = \sigma_{\min}^2$ and $\beta = \sigma_{\max}^2$, i.e., the upper- and lower frame bounds define the square of the largest and smallest singular value, respectively. Note that, the pooling matrices Φ_{ave} (equation 3.16) and Φ_{max} (equation 3.17), does not meet the condition of a frame operator (equation 3.39). Thus, pooling operators are therefore not considered frames.

A particular desirable frame is the tight-frame, which has frame bounds $\alpha = \beta$. To see why, let $\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\varepsilon}$ represent a noisy measurement, which gives the following signal expansion

$$\hat{\mathbf{c}} = \Phi^\top \hat{\mathbf{x}} = \Phi^\top \mathbf{x} + \boldsymbol{\varepsilon} \quad (3.42)$$

where $\hat{\mathbf{c}}$ represents noisy expansion coefficients and $\boldsymbol{\varepsilon} = \Phi^\top \boldsymbol{\varepsilon}$ represents the transformation noise. The synthesis is given by

$$\hat{\mathbf{x}} = \tilde{\Phi}\hat{\mathbf{c}} = \tilde{\Phi}\Phi^\top \mathbf{x} + \tilde{\Phi}\boldsymbol{\varepsilon} = (\Phi\Phi^\top)^{-1}\Phi\Phi^\top \mathbf{x} + (\Phi\Phi^\top)^{-1}\Phi\boldsymbol{\varepsilon} \quad (3.43)$$

The sensitivity of the matrices to perturbations of the data, or noise amplification, may be analyzed through the condition number κ , where a large condition number implies that small perturbations will cause large errors in the solution. For a matrix that is square and invertible, the amplification of the transformation noise $\boldsymbol{\varepsilon}$ may therefore be given from the condition number of $\Phi\Phi^\top$ [19]:

$$\kappa_2(\Phi\Phi^\top) = \|(\Phi\Phi^\top)^{-1}\|_2 \times \|\Phi\Phi^\top\|_2 = \frac{\sigma_{\max}^2(\Phi\Phi^\top)}{\sigma_{\min}^2(\Phi\Phi^\top)} = \frac{\beta}{\alpha}. \quad (3.44)$$

where $\|\cdot\|_2$ is the 2-norm of the matrix. This implies that tight-frame has the minimum noise amplification of all frames. An example of such a frame is the Haar basis [37]. In $\mathbb{R}^4$ the Haar basis can be represented as a 4×4 analysis matrix

$$\Phi^\top = \begin{pmatrix} - & \phi_1^\top & - \\ - & \phi_2^\top & - \\ - & \phi_3^\top & - \\ - & \phi_4^\top & - \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}. \quad (3.45)$$

In the 2D wavelet transform, the Haar basis vectors ϕ_1 and $\phi_i, i = (2, 3, 4)$ are commonly referred to as the scaling function and the wavelet basis, respectively. These vectors, or filters/kernels, are used to convolve a 2D signal into sub-band information represented by their respective wavelet coefficients [54]. The Haar basis in this case, are represented by the following digital filters

Figure 3.8: Example of a 2D wavelet transform using the Haar filters on the fully sampled shot-gather shown in Figure 3.3. This figure is taken and modified from paper I.

$$\begin{aligned}\phi_1^\top &= \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \phi_2^\top = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \\ \phi_3^\top &= \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, \quad \phi_4^\top = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}.\end{aligned}\tag{3.46}$$

The Haar basis has the smallest possible support for 2D frames, which reduces the computational cost of the convolutions. Another desired property of frames is to have good frequency selectivity, which in a sense advocates larger filters. Thus, a good frame design includes small support and good feature separation where each filter emphasizes a particular region in the frequency domain, although, support of the frame and frequency separation do compete [12].

In CNNs, the Haar basis can be introduced as a set of fixed filters (non-learnable), where convolutions have strides equal to the filter length. Considering now the inverse problem $\mathbf{d} = \mathbf{S}\mathbf{m} + \varepsilon$, where $\mathbf{m}$ is assumed to be known. A 2D wavelet transform using the Haar basis gives four downsampled features represented by wavelet and scaling coefficients, i.e., a four-band representation $\mathbf{m}^A$, $\mathbf{m}^H$, $\mathbf{m}^V$ and $\mathbf{m}^D$, where each representation is commonly referred to as the approximation, horizontal detail, vertical detail and diagonal detail, respectively. A 2D wavelet transform is illustrated in Figure 3.8, where $\mathbf{m}$ is represented by the seismic shot-gather shown in Figure 3.3.

In transform based learning, we can utilize frames to first transform the data, then train the DL model in the transform domain. Hence, we first apply the the forward transform on the target and input variables, i.e., $\Phi^\top \mathbf{m}$ and $\Phi^\top \mathbf{d}$, where we learn the function. In this case, the predictions are given by the inverse transform of the learned function, i.e.,

$$\hat{\mathbf{m}}_f = \tilde{\Phi} \hat{f}(\Phi^\top \mathbf{d}; \hat{\mathbf{W}}, \hat{\mathbf{b}}).\tag{3.47}$$

In one sense, the motivation of DL is to automatically extract this type of mapping, and that an additional mapping from the forward $\Phi^\top$ and inverse Φ may seem redundant. A potential benefit using transform-based learning is that noise artifacts may become topological simpler in an appropriate feature space [5, 28, 41, 49]. Also, if the image contains complicated patterns, state-of-the-art CNNs, trained in the image domain, may show inferior performance compared to traditional methods [5].

We can also utilize framelet expansion as part of the internal structure of the network, thus replacing for instance the pooling operators with frame expansions within the network layers to avoid low-rank approximation of the features. For instance, Ye et al [92] propose the use of wavelet expansions to optimize performance of the U-net architecture, where the pooling operators are replaced with Haar filters.

3.2.5 Guided unsupervised learning for inverse problems

In many circumstances, the ground truth or target data simply does not exist, or are considered too costly or difficult to acquire. In these circumstances, we need to turn to unsupervised approaches, where we try to learn the underlying pattern given only the observed data. Again, consider the forward model $\mathbf{d} = \mathbf{S}\mathbf{m} + \boldsymbol{\varepsilon}$, where we assume $\mathbf{S}$ is known. In this case, we can try to minimize

$$\mathcal{C}(\mathbf{W}, \mathbf{b}) = \frac{1}{2} \|\mathbf{d} - \mathbf{S}\mathbf{m}_f\|_2^2, \quad (3.48)$$

where $\mathbf{m}_f$ represents an arbitrary NN of the parameters $\mathbf{W}$ and $\mathbf{b}$ taking $\mathbf{d}$ as input. In this case, the forward operator $\mathbf{S}$ projects the model back onto a lower dimensional subspace. Hence, this procedure tries to copy its input to its output, which is considered the basics of an autoencoder (AE) approach. AE's are typically based on ED designs and with all the basic properties of a NN, such as downsampling, residual layers, and the concept of under- and overcompleteness. An undercomplete AE tries to learn the principal subspace of the training data [24], which makes AE a non-linear analogue to principal component analysis, and therefore leads to a low-rank regularization effect on the solution. In overcomplete AEs, we need to introduce regularization to avoid overfitting, such as the ℓ_2 -norm penalty on the weights, i.e.,

$$\mathcal{C}(\mathbf{W}, \mathbf{b}) = \frac{1}{2} \|\mathbf{d} - \mathbf{S}\mathbf{m}_f\|_2^2 + \frac{\lambda}{2} \|\mathbf{W}\|_2^2. \quad (3.49)$$

Although the ℓ_2 -norm penalty on the weights is one of the traditional regularizers in DL, it may give preference to a different type of behaviour than what is desired. Alternatively, one may instead infer an explicit behaviour to the model, e.g.,

$$\mathcal{C}(\mathbf{W}, \mathbf{b}) = \frac{1}{2} \|\mathbf{d} - \mathbf{S}\mathbf{m}_f\|_2^2 + \frac{\lambda}{2} \|\mathbf{W}\|_2^2 + R(\mathbf{m}_f), \quad (3.50)$$

where the penalty functional $R(\mathbf{m}_f)$ is introduced to give preference to other properties, such as sparse or gradient sparse solutions.

3. Deep learning for inverse problems

To illustrate the implications of equation 3.50, consider a DL-based TV inverse example in 3D. In this case, let the observed data $\mathbf{d}$ represent a 3D seismic cube, and the model $\mathbf{m}_f$ represent a deep 3D convolutional AE (CAE). Finally, Ψ denote the vertical (time) derivative of a 3D convolution kernel, or point spread function (PSF).

Figure 3.9: Illustration of the observed data (left), which is computed by a spatio-temporal convolution between a PSF (middle left) and the true model (middle right) including additive white Gaussian noise (right).

In this case, we want to find an approximation to the true model $\mathbf{m}$ by removing the 3D blurring effect represented by the PSF. The model $\mathbf{m}$ represents now a blocky structured model, as illustrated in Figure 3.9. To include blockiness into the solution we solve for an objective function on the form

$$\begin{aligned} \mathcal{C} = & \frac{1}{2} \|\mathbf{d} - \Psi \mathbf{m}_f\|_2^2 + \frac{\lambda}{2} \|\mathbf{W}\|_2^2 \\ & + \frac{1}{N} \sum_{i=1}^N (\alpha_x |\mathbf{D}_x \mathbf{m}_f|_i| + \alpha_y |\mathbf{D}_y \mathbf{m}_f|_i| + \alpha_t |\mathbf{D}_t \mathbf{m}_f|_i|), \end{aligned} \quad (3.51)$$

where λ , α_x , α_y and α_t define the trade-off between the different regularization terms and the misfit function, and $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ denote the derivative operators acting in the crossline, inline and temporal direction, respectively. By penalizing the model $\mathbf{m}_f$ towards a gradient sparse solution the prediction improves and displays sharper transitions between layers, as shown in Figure 3.10.

The main observation is: a basic unsupervised approach (equation 3.48 and equation 3.49) is not working properly and therefore need constraints to limit the solution space that works explicitly on the model.

Similarly, we can solve for the inverse problem where the forward problem is governed by the subsampling operator $\mathbf{S}$, which in case we also need to include suitable regularization penalties. This may be achieved by minimizing equation 3.50 using for instance a CAE, and use the trained function $\mathbf{m}_f$ to predict both the “training” data and new and unseen data. This is illustrated in Figure 3.11.

Other classical procedures include denoising AE’s which are trained by introducing stochastic corruption of the input during training [84]. Training with noise has also been proven as an effective means to reduce overfitting and

Figure 3.10: An inverse example in 3D using DL, employing a regularized 3D CAE with the combined ℓ_2 and first-order TV-norm. Here, displaying the results from a set of regularization parameters $\{\lambda = 0, \alpha_x = 0, \alpha_y = 0, \alpha_t = 0\}$ (left), $\{\lambda = 0.01, \alpha_x = 0, \alpha_y = 0, \alpha_t = 0\}$ (middle) and $\{\lambda = 10^{-5}, \alpha_x = 0.06, \alpha_y = 0.06, \alpha_t = 0.06\}$ (right). All examples have similar misfit error.

to stabilize inverse problems with DL [23]. Training with noise also have similar regularization effects as of *Tikhonov* regularization [6]. Another more recently introduced unsupervised approach is the deep image prior (DIP) [83], which instead of trying to copy its input to its output, the input consist of a random (fixed) vector $\mathbf{x}$ and learn to map this vector to the observed data. Liu et al [48] further improved the DIP approach by introducing a TV penalty to the output from the network. DIP and TV-DIP types of unsupervised approaches exploit the networks ability to capture low-level statistics of the data. However, a challenge for the DIP approach compared to autoencoding, is that the model does not generalize to new and unseen data. This means that for every new prediction one needs to retrain the network parameters, which becomes computationally demanding with large data sets.

Figure 3.11: An illustration of unsupervised learning using autoencoding, e.g., a CAE to find an underlying pattern, hidden within the data. First, the forward problem $\mathbf{S}\mathbf{m}_f$ is computed during training, where we fit the model only to the observed data (top). After training, the approximated solution is then given by $\hat{\mathbf{m}}_f$ (bottom).

Chapter 4

Summary of papers, their contributions, implications and discussion of future work

In this chapter, I give a short summary of the contributions and implications learned from the papers. I finalize this chapter with sections where I discuss limitations of the methods and outlook for future research.

4.1 Summary of papers

4.1.1 Summary Paper I: Cross-streamer wavefield reconstruction through wavelet domain learning

In paper I, we investigate DL as a tool for wavefield reconstruction beyond spatial aliasing. Reconstruction beyond aliasing is a longstanding and important problem in seismic processing, since coarse sampling of the wavefield can lead to artifacts in consecutive processing steps, such as migration. Predicting the dense wavefield in the crossline direction involves approximation of a function that computes the inverse projection of the subsampling operator—an operator that inherits a large null-space due to coarse crossline sampling relative to the inline sampling.

In this case, infinite many solutions may fit the observed data. In the DL framework, we try to train a function which represents an approximation to the generalized inverse of the forward operator. Several DL techniques, especially involving CNNs, have shown to be effective for similar problems in related research areas. First and foremost, the problem in geophysical sciences is access to good training data. Since we did not have access to densely sampled crossline data, we hypothesized that the nonlinear relationships between the subsampled wavefield and the corresponding densely sampled wavefield can be learned in one direction—where the spacing between sensors is adequately dense—and then use the trained function to reconstruct the wavefield in the coarser direction. Thus, we use similarities between acquisition domains and the structure of the 3D wavefield to access training data.

The proposed approach is based on a single-image super-resolution technique, which employs a CNN to directly upsample low-resolution images into their corresponding high-resolution counterparts. In our approach, we transform the data into a feature space prior to learning, such that the CNN is trained in this domain. This may simplify the learning process in the presence of complicated patterns and low signal-to-noise ratio. The feature space prior to learning can

4. Summary of papers, their contributions, implications and discussion of future work

be represented by any transform domain using analytic functions. However, we investigate use of the wavelet transform, where we experienced that the Haar basis [37] was the most stable in our experiments using shot-gathers. The proposed method combines effectiveness of small input sizes from the SubPixel CNN [71] and the transform-based approach proposed in Deep Wavelet CNN [28], and modified them accordingly to more appropriately fit the seismic data problem. The wavelet-based CNN was tested on synthetic broad-band data and compared with an industry-standard processing workflow —employing the anti-leakage Fourier Transform (ALFT) method [89, 90]. In addition, tests on TopSeis field data from the Barents sea were carried out. The approach showed improved results in the root-mean-squared error close too a factor of 2 when compared with the state-of-the-art industry approach.

4.1.2 Summary Paper II: Unsupervised deep learning with higher-order total-variation regularization for multidimensional wavefield reconstruction

In paper II, we investigate the problem of reconstructing sparse near-offsets on constant-offset gathers, manifested by an irregular distribution of traces on a regularized 4D grid, i.e., inline, crossline, time and constant offset. In these circumstances, we can not use the sampling geometry or structure of the wavefield to define labeled training data for supervised learning. As an alternative, we tried to solve for the inverse problem directly, using DL methodology. Here we formulated the inverse problem as an unsupervised learning problem through autoencoding, where the functional inverse mapping is learned only from the observed data and guided by appropriate constraints.

The proposed method is based on a regularized convolutional autoencoder (RCAE). For this to work properly we extended the typical image-based (2D) into 4D employing a 3D CNN, in which the fourth dimension is represented by offset-classes. Meaning, that the offset dimension is treated as a feature space of the input, similar to what is performed on color images. As CNN design, we propose a basic overcomplete encoder-decoder architecture, and stabilize the inverse by constraining the solution space employing three different regularization penalties: the ℓ_2 -norm on the convolutional kernels, and a first- and second-order TV-norm on the model. The first-order total-variation (TV) norm was introduced to promote laterally sharp characteristics of the wavefield and the second-order TV-norm for promoting events with curvature where the wavefield display laterally large gaps.

The RCAE approach was tested on synthetic broad-band data, and a field data set from the Barents Sea represented by a 3D source-over-cable survey. In the synthetic example, the unsupervised approach was able to reconstruct the sparse near-offsets with curved events and conflicting dips. However, we experienced that the RCAE struggled with conflicting dips from the weaker events and where conflicting dips displayed different strengths in amplitude.

In the field data example, we compared the RCAE with an industry-standard

full production workflow, based on ALFT. The two approaches were compared after reconstruction and in the migrated domain. In the near-offsets, the unsupervised learning approach showed in general a less noisy character and improved structural definition compared to state-of-the-art industry workflow. The same observation was made after constant offset migration. After adding the migrated offsets, the two methods displayed similar structural characteristics.

4.1.3 Summary Paper III: De-migration-based supervised learning for interpolation and regularization of 3D constant offset volumes

In paper III, we investigate the problem of reconstructing sparse constant-offset gathers in a supervised learning framework, utilizing synthetic data suitable for this task. The supervised DL approach benefits from its simplicity, by incorporating fewer assumptions into the algorithm, compared to the unsupervised approach —where careful engineering of some appropriate penalty functional are typically incorporated. However, one of the key challenges related to supervised DL-based interpolation of seismic data, is access to good training data.

The training data in our approach is generated by de-migration of stacked PSDM seismic volumes. We use these de-migration volumes to train an overcomplete 3D convolutional encoder-decoder (CED), regularized with the ℓ_2 -norm on the convolutional kernels. In this case, for each offset class volume, we de-migrate the PSDM image into two configurations; (1) the original survey configuration in the real survey and (2) a target survey configuration with constant offset and azimuth for each 3D offset class. Thus, the NN is trained solely on synthetic data generated from reflectivity volumes represented by field data and the trained NN is then applied to field data in order to regularize and interpolate the seismic data.

We test this novel supervised learning approach on one synthetic and one field data example and show that it performs better than state-of-the-art industry standard processing based on ALFT, especially for the nearest offset classes. On the synthetic data we also demonstrate that the CED method preserves the amplitude versus offset (AVO) as reliably as the standard method.

4.2 Implications and discussion of future work

4.2.1 Alternative transform domains

In transform-based learning for seismic wavefield reconstruction, one may consider alternatives to the wavelet domain learning proposed in paper I and discussed in section 3.2.4. One may for instance introduce deeper transformations, similar to wavelet packet decomposition [14]. This may give better separation between the signal and noise. Utilizing the Haar basis in this case, would be straight forward to implement in existing DL libraries, such as Tensorflow. Maybe even more optimally, would be to employ other basis functions, i.e., more appropriate basis expansions tailored for seismic data. A couple of good candidates could be curvelets [9] or seislets [21, 22]. However, employing basis functions with large support in the shot-domain may cause problems, e.g., edge effects, due to the small number of samples in the crossline direction. This was also discussed in paper I, where we suggested different basis function in each direction as a potential solution to this problem.

In a DL setting, one could also consider using an ensemble of different basis functions, i.e., an ensemble of wavelets, curvelets, seislets, etc., which would make the approach more general and maybe less susceptible to reconstruction errors from the loss of expansion coefficients. One could also replace some of the internal (hidden) layers with such basis functions, which would ease the computational burden of the learning phase since the linear transformations become fixed instead of learned. In this case, the fixed kernels in a layer l are not optimized through gradient descent.

Furthermore, one may also consider to exploit appropriate basis expansions as regularization penalty, similar to the gradient sparse solution in equation 3.51. In this case, the differential operators are then replaced with appropriate basis expansions.

4.2.2 2D-based vs. 3D/4D-based reconstruction

One of the drawbacks of applying image-based (2D) approaches from natural image processing to seismic processing, are their limited ability to capture correlations and patterns from the full 3D seismic wavefield. In some circumstances, however, 2D-based DL techniques may show superior performance to state-of-the-art traditional processing workflows. This has been demonstrated in our experiments in paper I employing synthetic data, displayed in Figure 4.1 and Figure 4.2. Here, our wavelet-based CNN method is compared to a 3D ALFT.

However, the example shown here may be (too) ideal for learning algorithms, because the simplicity of such synthetic data does not necessarily reflect the complexities found in field data where irregularities of the sensing system, processing artifacts and various types of noise occurs. This makes training on data represented by field data a much more challenging task for the learning algorithm.

Figure 4.1: Reconstruction results in the crossline direction from the shallow section of the wave-field, that is, the water bottom. From **a)** the ground truth, we observe a complex wavefield setting, which follows that **b)** the subsampled cross-line direction yields a challenging wavefield to reconstruct. From **c)** the 3D Fourier and **e)** the wavelet CNN, we observe more residual energy within the **d)** 3D Fourier interpolation error than the **f)** wavelet CNN reconstruction error. Figure taken from Paper I.

As observed from our experiment in paper I, displayed in Figure 4.3, the crossline reconstruction shows striped artifacts and loss of frequencies above 40 Hz, caused by close to perfect reconstruction of the input traces, but a smoother characteristics of the predicted traces in between. Even though the reconstruction yields promising results, an extension to higher dimensions would be worth investigating further for possible improvement of the method.

A natural extension to the wavelet CNN in paper I would be to introduce the third dimension to make it a 3D approach. However, even though an extension to 3D would be straight forward to implement, we would not have access to a densely sampled 3D wavefield to perform supervised training in these circumstances. This is, of course, part of the main problem in DL applied to seismic data; access to good and reliable training data.

4. Summary of papers, their contributions, implications and discussion of future work

Figure 4.2: Migrated synthetic data displaying a crossline section, and time slices through 450 ms, where a), b) represent the migration of the ground truth, c), d) the 3D ALFT, and e), f) the wavelet CNN. Figure taken from Paper I.

In paper II we solved the inverse problem in an unsupervised fashion, by using autoencoding and regularization. This could also be a possible approach for the shot-domain reconstruction problem. Furthermore, in paper II we utilized knowledge about the reflected wavefield to simplify the reconstruction problem, by employing move-out correction, i.e., applying a static shift to the data that follows the arrival times of the seismic waves, in order to make them (more) flat.

Figure 4.3: Reconstruction result in the crossline direction from an arbitrary shot gather extracted from the training data, where **a)** represents a 3D shot gather with 14 input traces, **b)** represents the reconstruction, **c)**, **d)** correspond to their respective frequency-wavenumber spectra and **e)** corresponding amplitude spectra. Figure taken from Paper I.

In Figure 4.4 we see that unsupervised DL in higher dimensions (4D) manages to reconstruct highly undersampled near-offset on TopSeis field data.

4.2.2.1 Network architecture

The network architecture and its capacity (width and depth) are in some circumstances crucial for performance. If any given layer in the network is built in such a way that important information might be lost during forward propagation, we may not be able to reach the desired result, despite having good training data and/or a large training data set. Some of the most popular architectures, the U-net [66] and the ResNet [32] (and their adaptations), have been proposed in multiple research papers for seismic reconstruction problems [40, 43, 75, 87, 88].

Despite their success, the performance of a network is very case dependent; where a network performance might change drastically on different but related cases, such as super-resolution vs. image inpainting, or supervised vs. unsupervised. Therefore, I would argue that copying networks directly that have been designed for related tasks in image processing is not the best approach. For example, the original ResNet design is not good in capturing far distance correlations due to small convolutional kernels and no downsampling, i.e., it has a small receptive field. One may have to increase the size of the kernels and/or introduce downsampling layers to make it fit better to the given task. The U-net architecture on the other hand has multiple advantages, such as both increase of receptive field and use of multi-resolution layers. Thus, it captures

4. Summary of papers, their contributions, implications and discussion of future work

both local correlations at higher levels and distant correlations at lower levels of the network.

However, the U-net network has also shortcomings. One is that it puts too much emphasis on the low-frequency component, due to the combination of low-rank approximations and skip-connections [92]. In many seismic data processing tasks, this may not be a problem due to adequate sampling. However, the short distance from input to output in the higher levels, may cause problems in unsupervised learning for reconstruction problems [83]. We experienced similar problems when testing different U-net designs in the unsupervised approach, conducted in paper II. In our experiments, we either overfitted the observed traces, or the interpolated traces were too smooth. This is why we settled for a simpler overcomplete ED network. Although, introducing framelet expansions, to separate the low-frequency component from the high, to optimize the U-net structure as proposed by Ye et al [92], could be a possible topic for future studies on seismic data.

One could also argue that seismic data does not need highly complicated and sophisticated architectures, as proposed for various inverse problems in natural image processing. The reason being the fewer features—or patterns—present within seismic data compared to natural images. The complexities in wavefield reconstruction relate more to the issue of coarse spatial sampling, poor signal-to-noise-ratio, the dynamic range of amplitudes and varying frequency band from shallow to deeper parts of the seismic record. In this case, the original ResNet or U-net designs may be unnecessarily deep. Introducing suitable regularization penalties and/or fixed basis function to simplify learning, may be more relevant than introducing highly complicated networks. As we have seen in this thesis, simple overcomplete ED networks with regularization gives promising results on complicated field data.

Figure 4.4: A crossline section from field data where **a** is the observed data, **b** the regularized convolutional autoencoder (RCAE) reconstruction and **c** interpolation from an industry standard processing (ISP) workflow employing ALFT. Figure taken from paper II.

4.2.2.2 The instability issue in DL

Another relevant problem, is that DL for inverse problems may experience an internal conflict between stability and accuracy. Meaning that, even though NN's may outperform traditional state-of-the-art methods, this may come at the cost of stability, and even though stable NN's could be easily created, these networks may come at the cost of accuracy [2, 25]. The instability phenomenon is referred to as *the universal instability theorem* [3, 25] of DL. In addition, the instability phenomenon is most likely independent of the sensing domain, such as reconstruction from either Fourier samples, Radon samples or directly in the physical domain [2]. Also, the instability issue is closely related to concepts of kernel functions and resolution in geophysical inverse theory, described by the data- and the model resolution matrix [56]. I will not go deeply into this matter, but discuss some of the important consequences that the instability theorem may have.

The instability issue can be explained on the basis of *Lipschitz continuity* [25], which is a strong form of uniform continuity for real valued functions [67]. In this case, assume we have access to two ground truth vectors $\mathbf{x}^{[1]}$ and $\mathbf{x}^{[2]}$ that are well separated in the vector space $\mathcal{N}_1$ (Figure 4.5). Now, if their difference $\mathbf{x}^{[1]} - \mathbf{x}^{[2]}$ lies close to or in the null-space of an operator $\mathbf{S}$, i.e., $\mathbf{S}(\mathbf{x}^{[1]} - \mathbf{x}^{[2]}) = \tilde{\mathbf{x}}^{[1]} - \tilde{\mathbf{x}}^{[2]} \simeq \mathbf{0}$, and our trained NN $\hat{f}(\cdot)$ accurately predicts $\mathbf{x}^{[1]}$ and $\mathbf{x}^{[2]}$, then the NN simply lacks *kernel awareness* [25]. Formally, a real-valued function $\hat{f}(\cdot)$ is said to be *Lipschitz* if there exists a positive real constant $C \geq 0$ such that

$$\|\hat{f}(\tilde{\mathbf{x}}^{[1]}) - \hat{f}(\tilde{\mathbf{x}}^{[2]})\| \leq C \|\tilde{\mathbf{x}}^{[1]} - \tilde{\mathbf{x}}^{[2]}\|, \quad (4.1)$$

where $\|\cdot\|$ denotes an error metric. The inevitable effect is that this leads to a large *Lipschitz constant* C , implying that the NN is unstable. In the worst case scenario, the inputs are identical, as shown in Figure 4.5.

In sparse regularization [11, 68], one exploits sparsity of the data under certain sparsifying transforms, e.g., wavelets or gradient operators, where the behaviour of the solution space is chosen explicitly as a suitable a priori model for the class of data [25], i.e., the reconstruction method is designed specifically with the vector space $\mathcal{N}_1$ (Figure 4.5) of acceptable solutions in mind. This is different from typical DL approaches, where the information of the behaviour of the solution space is only accessed through the training data.

The instability phenomenon may have some consequences for the work presented in this thesis. In paper I the instability issue definitely applies, because of the coarse crossline sampling may give two or more inputs in which their difference lies close to the null space of the forward operator. In this case, we may observe over-performance and various instabilities as described in Gottschling et al [25]. In unsupervised training for seismic wavefield reconstruction —as proposed in paper II— the NN is trained only to fit the observed data. Meaning, that the subsampling operator $\mathbf{S}$ projects the output from the network back onto the lower dimensional subspace. But appropriate regularization penalties are needed to stabilize the solution and to prevent overfitting. Thus, one could

4. Summary of papers, their contributions, implications and discussion of future work

argue that it is not likely that unsupervised learning will exhibit similar over-performance as observed in supervised training — where the labeled inputs are “forced” to produce their corresponding labeled target data. Another issue is the consequence of false negatives: where the NN under-performs on features that it have not seen [25].

Figure 4.5: The subsampling operator $\mathbf{S}(\cdot)$ maps **a)** the targets $\mathbf{x}^{[1]}, \mathbf{x}^{[2]} \in \mathcal{N}_1$ to $\tilde{\mathbf{x}}^{[1]}, \tilde{\mathbf{x}}^{[2]} \in \mathcal{N}_2$, where $\mathcal{N}_2$ is a subspace of $\mathcal{N}_1$. The trained NN $\hat{f}(\tilde{\mathbf{x}})$ maps **b)** the subsampled counterparts $\tilde{\mathbf{x}}^{[1]}, \tilde{\mathbf{x}}^{[2]}$ back to the high-dimensional space, with high accuracy. Intuitively, one can imagine a worst case scenario where two targets are given by **c)** patterns with 45deg dip in two different directions, in which **d)** gives identical patterns in lower-dimensional subspace. Here the subsampling operator keeps every second column.

This applies both to supervised and unsupervised approaches. However, we should bear in mind that seismic waves arguably are associated with a more limited solution space than natural images, because seismic waves follow more strict patterns compared to natural images. In these circumstances, DL may actually be an optimal choice since generalization may be easier to achieve with less data. Another argument, is that we can utilize our knowledge about reflected waves to introduce transformations of the reflected events into a different model space. In this model space, e.g. normal move-out correction, the reconstruction problem becomes simpler since we can approximate the signal with fewer basis functions. Flattening by normal move-out correction is common practice in most interpolation workflows performed in 4D and above. This was also performed in paper II and paper III.

4.2.3 Implications of the results

In this thesis, we have addressed the issue of spatial undersampling, which is central in seismic imaging, and investigated DL approaches for reconstructing the seismic wavefield. It is well known that the processing steps employed before migration have direct consequence on the quality of the final seismic image. Reconstruction techniques that produce fewer migration artifacts and improved structural definition of the geological layers after imaging may as a consequence lead to improved seismic interpretation and analysis. As shown in Figure 4.6 and based on our experiments in paper II, DL may yield improved reconstruction of the near-offsets compared to industry standard processing workflows. This was performed with unsupervised learning through autoencoding, which is a possible approach when labeled training data are not available.

Figure 4.6: This figure represents time-slices after migration of the 25 m offset from a TopSeis survey, where **a)** show the reconstruction from unsupervised deep learning and **c)** show a industry standard approach, with **b)** and **d)** as their corresponding zoomed sections (red rectangles), respectively. This Figure is taken from paper II.

4. Summary of papers, their contributions, implications and discussion of future work

Figure 4.7: A crossline example from a source-over-cable survey for offset 37.5 m, where **a)** is the sparsely sampled observed data, **b)** is the industry standard approach, **c)** is the result from the DL approach trained on synthetic data **d)** are the amplitude spectra from **a)**, **b)**, **c)**, normalized to the largest peak. This Figure is taken from paper III.

One of the difficulties with unsupervised approaches though, is that one may need to include one or more regularization penalties in which the hyperparameter space becomes larger. This is different from supervised learning where you commonly only include the ℓ_2 -norm to reduce overfitting. In addition, in unsupervised learning, it is difficult to measure goodness of fit. In our case, it was done by finding an appropriate lower bound on the ℓ_2 -norm error of the misfit function such that the network did not overfit the observed data, which is also another type of hyperparameter. The hyperparameter search can be done by a trial and error approach on a smaller subset of the data.

Also, the problem by utilizing field data only for learning is that field data possess various processing artifacts and noise patterns that may camouflage the true pattern of the wavefield. Thus, as an alternative to training on field data, one can try to learn the generalized inverse map on representative synthetic data that, in an optimal setting, includes all or most of the features that field data

possess. We found that a suitable approximation to the inverse can be found by utilizing training data that show similar characteristics to the field data, by generating synthetic data from de-migrated PSDM images of field data. This approach lead to an large uplift in reconstruction of wave patterns compared to industry standard approaches, especially in the near offsets. This is shown in Figure 4.7 from our experiments in paper III, which display promising results on both diffractions, conflicting dips and curved events.

The structural patterns of the migrated sections, as seen in the crossline section in Figure 4.8 and the time-slices in Figure 4.9, also implies that the interpolated traces are more correctly positioned in time and space using a DL methodology compared to the traditional workflows performed by the industry. As a consequence, the DL approaches may lead to both improved imaging results and also faster and practically simpler applications for the industry in the future.

Figure 4.8: A crossline example from a source-over-cable survey for offset 37.5 m after Kirchhoff depth migration stretched back to the time domain, where **a)** represents the industry standard reconstruction (ALFT) and **b)** represents the CED trained on synthetic data. This Figure is taken from paper III.

4. Summary of papers, their contributions, implications and discussion of future work

Figure 4.9: A time-slice example from a source-over-cable survey for offset 37.5 m after Kirchhoff depth migration stretched back to the time domain, where **a**), **c**) represent the industry standard reconstruction and **b**), **d**) represent the DL approach trained on synthetic data. This Figure is taken from paper III.

Bibliography

- [1] Abadi, M. et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).
- [2] Antun, V. et al. “On instabilities of deep learning in image reconstruction and the potential costs of AI”. In: *Proceedings of the National Academy of Sciences* vol. 117, no. 48 (2020), pp. 30088–30095.
- [3] Antun, V. et al. *SIAM NEWS Deep Learning in Scientific Computing: Understanding the Instability Mystery*. <https://sinews.siam.org/Details-Page/deep-learning-in-scientific-computing-understanding-the-instability-mystery>. 2021. (Visited on 09/14/2021).
- [4] Arridge, S. et al. “Solving inverse problems using data-driven models”. In: *Acta Numerica* vol. 28 (2019), pp. 1–174.
- [5] Bae, W., Yoo, J., and Chul Ye, J. “Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 145–153.
- [6] Bishop, C. M. “Training with noise is equivalent to Tikhonov regularization”. In: *Neural computation* vol. 7, no. 1 (1995), pp. 108–116.
- [7] Bredies, K., Kunisch, K., and Pock, T. “Total generalized variation”. In: *SIAM Journal on Imaging Sciences* vol. 3, no. 3 (2010), pp. 492–526.
- [8] Bruna, J. and Mallat, S. “Invariant scattering convolution networks”. In: *IEEE transactions on pattern analysis and machine intelligence* vol. 35, no. 8 (2013), pp. 1872–1886.
- [9] Candès, E. J. and Demanet, L. “The curvelet representation of wave propagators is optimally sparse”. In: *Communications on Pure and Applied Mathematics* vol. 58, no. 11 (2005), pp. 1472–1528.
- [10] Candès, E. J. and Fernandez-Granda, C. “Towards a mathematical theory of super-resolution”. In: *Communications on pure and applied Mathematics* vol. 67, no. 6 (2014), pp. 906–956.
- [11] Candès, E. J., Romberg, J., and Tao, T. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on information theory* vol. 52, no. 2 (2006), pp. 489–509.
- [12] Casazza, P. G. and Kutyniok, G. *Finite frames: Theory and applications*. Springer, 2012.

- [13] Chen, S. and Donoho, D. “Basis pursuit”. In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Vol. 1. IEEE. 1994, pp. 41–44.
- [14] Coifman, R. R., Meyer, Y., and Wickerhauser, V. “Wavelet analysis and signal processing”. In: *In Wavelets and their applications*. Citeseer. 1992.
- [15] Constable, S. C., Parker, R. L., and Constable, C. G. “Occam’s inversion: A practical algorithm for generating smooth models from electromagnetic sounding data”. In: *Geophysics* vol. 52, no. 3 (1987), pp. 289–300.
- [16] Dong, C. et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* vol. 38, no. 2 (2015), pp. 295–307.
- [17] Duffin, R. J. and Schaeffer, A. C. “A class of nonharmonic Fourier series”. In: *Transactions of the American Mathematical Society* vol. 72, no. 2 (1952), pp. 341–366.
- [18] Dumoulin, V. and Visin, F. “A guide to convolution arithmetic for deep learning”. In: *arXiv preprint arXiv:1603.07285* (2016).
- [19] Faul, A. C. *A Concise introduction to numerical analysis*. Chapman and Hall/CRC, 2018.
- [20] Fomel, S. “Shaping regularization in geophysical-estimation problems”. In: *Geophysics* vol. 72, no. 2 (2007), R29–R36.
- [21] Fomel, S. “Towards the seislet transform”. In: *SEG Technical Program Expanded Abstracts 2006*. Society of Exploration Geophysicists, 2006, pp. 2847–2851.
- [22] Fomel, S. and Liu, Y. “Seislet transform and seislet frame”. In: *Geophysics* vol. 75, no. 3 (2010), pp. V25–V38.
- [23] Genzel, M., Macdonald, J., and März, M. “Solving Inverse Problems With Deep Neural Networks–Robustness Included?” In: *arXiv preprint arXiv:2011.04268* (2020).
- [24] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.
- [25] Gottschling, N. M. et al. “The troublesome kernel: why deep learning for inverse problems is typically unstable”. In: *arXiv preprint arXiv:2001.01258* (2020).
- [26] Goupillaud, P., Grossmann, A., and Morlet, J. “Cycle-octave and related transforms in seismic signal analysis”. In: *Geoexploration* vol. 23, no. 1 (1984), pp. 85–102.
- [27] Guo, K., Kutyniok, G., and Labate, D. *Sparse multidimensional representations using anisotropic dilation and shear operators*. 2006.
- [28] Guo, T. et al. “Deep wavelet prediction for image super-resolution”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 104–113.

- [29] Han, Y. and Ye, J. C. “Framing U-Net via deep convolutional framelets: Application to sparse-view CT”. In: *IEEE transactions on medical imaging* vol. 37, no. 6 (2018), pp. 1418–1429.
- [30] Hansen, P. C. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.
- [31] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.
- [32] He, K. et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [33] Hennenfent, G., Fenelon, L., and Herrmann, F. J. “Nonequispaced curvelet transform for seismic data reconstruction: A sparsity-promoting approach”. In: *Geophysics* vol. 75, no. 6 (2010), WB203–WB210.
- [34] Hinton, G., Srivastava, N., and Swersky, K. “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent”. In: *Cited on* vol. 14, no. 8 (2012), p. 2.
- [35] Hoop, M. V. de. “Deep neural-network architectures arising in seismic-inverse problems”. In: *SEG Technical Program Expanded Abstracts 2018*. Society of Exploration Geophysicists, 2018, pp. 4654–4658.
- [36] Hornik, K., Stinchcombe, M., and White, H. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* vol. 2, no. 5 (1989), pp. 359–366.
- [37] Haar, A. “Zur theorie der orthogonalen funktionensysteme”. In: *Mathematische Annalen* vol. 69, no. 3 (1910), pp. 331–371.
- [38] Jia, Y. and Ma, J. “What can machine learning do for seismic data processing? An interpolation application”. In: *Geophysics* vol. 82, no. 3 (2017), pp. V163–V177.
- [39] Jia, Y., Yu, S., and Ma, J. “Intelligent interpolation by Monte Carlo machine learning”. In: *Geophysics* vol. 83, no. 2 (2018), pp. V83–V97.
- [40] Jonge, T. de et al. “De-bubbling Seismic Data using a Generalized Neural Network”. In: *Geophysics* vol. 87, no. 1 (2021), pp. 1–73.
- [41] Kang, E., Ye, J. C., et al. “Wavelet domain residual network (WavResNet) for low-dose X-ray CT reconstruction”. In: *arXiv preprint arXiv:1703.01383* (2017).
- [42] Kingma, D. P. and Ba, J. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [43] Klochikhina, E. et al. “Leveraging deep learning for seismic image denoising”. In: *First Break* vol. 38, no. 7 (2020), pp. 41–48.
- [44] Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* vol. 25 (2012), pp. 1097–1105.

- [45] LeCun, Y. et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* vol. 86, no. 11 (1998), pp. 2278–2324.
- [46] Liang, J., Ma, J., and Zhang, X. “Seismic data restoration via data-driven tight frame”. In: *Geophysics* vol. 79, no. 3 (2014), pp. V65–V74.
- [47] Liu, B. and Sacchi, M. D. “Minimum weighted norm interpolation of seismic records”. In: *Geophysics* vol. 69, no. 6 (2004), pp. 1560–1568.
- [48] Liu, J. et al. “Image restoration using total variation regularized deep image prior”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 7715–7719.
- [49] Liu, P. et al. “Multi-level wavelet-CNN for image restoration”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 773–782.
- [50] Loshchilov, I. and Hutter, F. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).
- [51] Lucas, A. et al. “Using deep neural networks for inverse problems in imaging: beyond analytical methods”. In: *IEEE Signal Processing Magazine* vol. 35, no. 1 (2018), pp. 20–36.
- [52] Mallat, S. *A wavelet tour of signal processing*. Elsevier, 1999.
- [53] Mallat, S. “Group invariant scattering”. In: *Communications on Pure and Applied Mathematics* vol. 65, no. 10 (2012), pp. 1331–1398.
- [54] Mallat, S. G. “A theory for multiresolution signal decomposition: the wavelet representation”. In: (1987).
- [55] Mehta, P. et al. “A high-bias, low-variance introduction to machine learning for physicists”. In: *Physics reports* vol. 810 (2019), pp. 1–124.
- [56] Menke, W. *Geophysical data analysis: discrete inverse theory: MATLAB edition*. Vol. 45. Academic press, 2012.
- [57] Nesterov, Y. “A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$ ”. In: *Sov. Math. Dokl.* Vol. 27.
- [58] Papafitsoros, K., Schoenlieb, C. B., and Sengul, B. “Combined first and second order total variation inpainting using split Bregman”. In: (2013).
- [59] Papafitsoros, K. and Schönlieb, C.-B. “A combined first and second order variational approach for image reconstruction”. In: *Journal of mathematical imaging and vision* vol. 48, no. 2 (2014), pp. 308–338.
- [60] Pappayan, V., Romano, Y., and Elad, M. “Convolutional neural networks analyzed via convolutional sparse coding”. In: *The Journal of Machine Learning Research* vol. 18, no. 1 (2017), pp. 2887–2938.
- [61] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* vol. 12 (2011), pp. 2825–2830.

- [62] Phillips, D. L. “A technique for the numerical solution of certain integral equations of the first kind”. In: *Journal of the ACM (JACM)* vol. 9, no. 1 (1962), pp. 84–97.
- [63] Pinkus, A. “Approximation theory of the MLP model in neural networks”. In: *Acta numerica* vol. 8 (1999), pp. 143–195.
- [64] Ramachandran, P., Zoph, B., and Le, Q. V. “Searching for activation functions”. In: *arXiv preprint arXiv:1710.05941* (2017).
- [65] Robertsson, J. O. et al. “On the use of multicomponent streamer recordings for reconstruction of pressure wavefields in the crossline direction”. In: *Geophysics* vol. 73, no. 5 (2008), A45–A49.
- [66] Ronneberger, O., Fischer, P., and Brox, T. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [67] Royden, H. L. and Fitzpatrick, P. *Real analysis*. Vol. 32. Macmillan New York, 1988.
- [68] Rudin, L. I., Osher, S., and Fatemi, E. “Nonlinear total variation based noise removal algorithms”. In: *Physica D: nonlinear phenomena* vol. 60, no. 1-4 (1992), pp. 259–268.
- [69] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. “Learning representations by back-propagating errors”. In: *nature* vol. 323, no. 6088 (1986), pp. 533–536.
- [70] Shannon, C. E. “Communication in the presence of noise”. In: *Proceedings of the IRE* vol. 37, no. 1 (1949), pp. 10–21.
- [71] Shi, W. et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883.
- [72] Simons, F. J. et al. “Solving or resolving global tomographic models with spherical wavelets, and the scale and sparsity of seismic heterogeneity”. In: *Geophysical Journal International* vol. 187, no. 2 (2011), pp. 969–988.
- [73] Spitz, S. “Seismic trace interpolation in the FX domain”. In: *Geophysics* vol. 56, no. 6 (1991), pp. 785–794.
- [74] Srivastava, N. et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* vol. 15, no. 1 (2014), pp. 1929–1958.
- [75] Sun, J. et al. “Attenuation of marine seismic interference noise employing a customized U-Net”. In: *Geophysical Prospecting* vol. 68, no. 3 (2020), pp. 845–871.
- [76] Sutskever, I. et al. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

- [77] Tihonov, A. N. “Solution of incorrectly formulated problems and the regularization method”. In: *Soviet Math.* vol. 4 (1963), pp. 1035–1038.
- [78] Tikhonov, A. N. “Regularization of incorrectly posed problems”. In: *Soviet Mathematics Doklady*. 1963.
- [79] Trad, D. “Five-dimensional interpolation: New directions and challenges”. In: *CSEG Recorder* vol. 39, no. 3 (2014), pp. 40–46.
- [80] Trad, D. “Five-dimensional interpolation: Recovering from acquisition constraints”. In: *Geophysics* vol. 74, no. 6 (2009), pp. V123–V132.
- [81] Trickett, S. et al. “Rank-reduction-based trace interpolation”. In: *SEG Technical Program Expanded Abstracts 2010*. Society of Exploration Geophysicists, 2010, pp. 3829–3833.
- [82] Turquais, P. et al. “Parabolic dictionary learning for seismic wavefield reconstruction across the streamers”. In: *Geophysics* vol. 83, no. 4 (2018), pp. V263–V282.
- [83] Ulyanov, D., Vedaldi, A., and Lempitsky, V. “Deep image prior”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9446–9454.
- [84] Vincent, P. et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.
- [85] Vinje, V. et al. “Shooting over the seismic spread”. In: *First Break* vol. 35, no. 6 (2017).
- [86] Waldeland, A. U. et al. “Convolutional neural networks for automated seismic interpretation”. In: *The Leading Edge* vol. 37, no. 7 (2018), pp. 529–537.
- [87] Wang, B. et al. “Deep-learning-based seismic data interpolation: A preliminary result”. In: *Geophysics* vol. 84, no. 1 (2019), pp. V11–V20.
- [88] Wang, B. et al. “Intelligent missing shots’ reconstruction using the spatial reciprocity of Green’s function based on deep learning”. In: *IEEE Transactions on Geoscience and Remote Sensing* vol. 58, no. 3 (2019), pp. 1587–1597.
- [89] Xu, S., Zhang, Y., and Lambaré, G. “Antileakage Fourier transform for seismic data regularization in higher dimensions”. In: *Geophysics* vol. 75, no. 6 (2010), WB113–WB120.
- [90] Xu, S. et al. “Antileakage Fourier transform for seismic data regularization”. In: *Geophysics* vol. 70, no. 4 (2005), pp. V87–V95.
- [91] Yang, W. et al. “Deep learning for single image super-resolution: A brief review”. In: *IEEE Transactions on Multimedia* vol. 21, no. 12 (2019), pp. 3106–3121.
- [92] Ye, J. C., Han, Y., and Cha, E. “Deep convolutional framelets: A general deep learning framework for inverse problems”. In: *SIAM Journal on Imaging Sciences* vol. 11, no. 2 (2018), pp. 991–1048.

- [93] Yin, R. et al. “A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets”. In: *SIAM Journal on Imaging Sciences* vol. 10, no. 2 (2017), pp. 711–750.
- [94] Yu, S. et al. “Interpolation and denoising of high-dimensional seismic data by learning a tight frame”. In: *Geophysics* vol. 80, no. 5 (2015), pp. V119–V132.
- [95] Zhu, L., Liu, E., and McClellan, J. H. “Joint seismic data denoising and interpolation with double-sparsity dictionary learning”. In: *Journal of Geophysics and Engineering* vol. 14, no. 4 (2017), pp. 802–810.

Papers