

Zero-Shot Denoising of Distributed Acoustic Sensing Data using Deep Priors

Jannik Mänzer

Bachelor of Science
March 2025

Supervisors:

Prof. Dr. Stefan Harmeling
Sebastian Konietzny

Artificial Intelligence (VIII)

Department of Computer Science
TU Dortmund University

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Contents

1	Introduction	1
2	Background	3
2.1	Denoising	3
2.2	Distributed Acoustic Sensing	3
2.3	Deep Learning	5
2.3.1	Deep Neural Networks	5
2.3.2	Convolutional Neural Networks	5
2.3.3	Normalization	6
2.3.4	Attention Mechanisms	7
3	Related Work	9
3.1	Supervised Methods	9
3.2	Self-Supervised Methods	9
3.2.1	Target-Based Methods	9
3.2.2	Blind-Spot-Based Methods	9
4	Methods	11
5	Experimental Setup	13
6	Results	15
7	Discussion	17
8	Conclusion	19
	Bibliography	20
A	Appendix	23

Chapter 1

Introduction

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Chapter 2

Background

This chapter provides the necessary background for the denoising methods explored in this work. We begin by defining the general denoising problem and discussing its inherent challenges. We then introduce distributed acoustic sensing (DAS) as a real-world application and highlight the unique difficulties it presents. Finally, we present key deep learning concepts and techniques that are used in the context of denoising.

2.1 Denoising

In general, denoising refers to the process of recovering a clean signal from a noisy observation. Formally, it can be described by the inverse problem

$$y = x + n \tag{2.1}$$

where y is the noisy observation, x is the underlying clean signal and n represents some form of noise, for example Gaussian noise $n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. Denoising is an inherently ill-posed problem, as the noise and, in many cases, the noise distribution are unknown. Therefore, additional assumptions about the solution are essential for solving the denoising problem. This process is known as regularization and typically involves imposing certain constraints on the solution space to favor more natural solutions [15]. The choice of regularizer depends on the specific problem setting and the type of data involved.

2.2 Distributed Acoustic Sensing

Distributed acoustic sensing (DAS, or distributed *vibration* sensing, DVS) [5] is an innovative technology for high-resolution vibration measurements over large distances by utilizing fiber optic cables as sensor arrays. It is based on a phenomenon called Rayleigh scattering, which occurs in optical fibers. When a short laser pulse is sent through the fiber by a DAS interrogator unit, a fraction of the light is scattered back due to small variations or imperfections in the fiber. Vibrations along the cable caused by external influences, e.g. seismic events, strain the fiber, which in turn causes phase shifts in the backscattered light.

These shifts are detected by the interrogator and, since the travel time of the light is known, can be used to accurately locate the vibrations along the cable [8]. In order to extract meaningful measurements, strain is analyzed over sections of the fiber, rather than at individual points. The length of these sections is called the *gauge length*, while another parameter, the *channel spacing*, determines how much this section is moved for each measurement, or channel, along the cable [3]. In practice, each channel corresponds to a virtual sensor capturing the average strain within its gauge length. Typically, the gauge length is selected to be bigger than the channel spacing, meaning that the measurement sections of neighboring channels overlap, as visualized in Figure 2.1. This concept of virtual sensors leads to high cost-effectiveness and, paired with the high sample rates enabled by the optical approach, allows measurements with significantly higher spatial and temporal resolution compared to conventional seismographs.

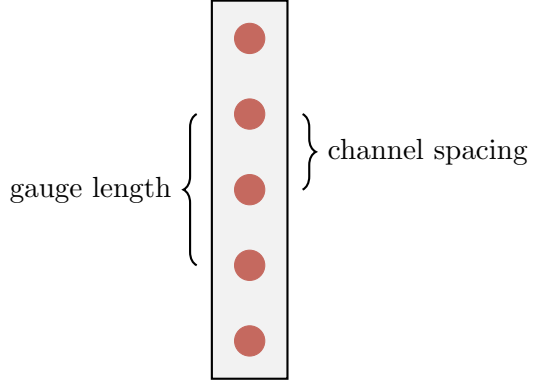


Figure 2.1: Gauge length and channel spacing. Red dots represent the individual channels along the fiber. Figure adapted from [3].

Despite these advantages, DAS systems often suffer from much lower signal quality than conventional seismographs, as they are more affected by various sources of noise. These can be divided into environmental noise and optical noise. Environmental noise includes natural phenomena such as winds or ocean waves, but also vibrations caused by vehicular and pedestrian traffic. Optical noise originates from various interactions between the light and the fiber. It includes high-amplitude erratic noise and common mode noise [2].

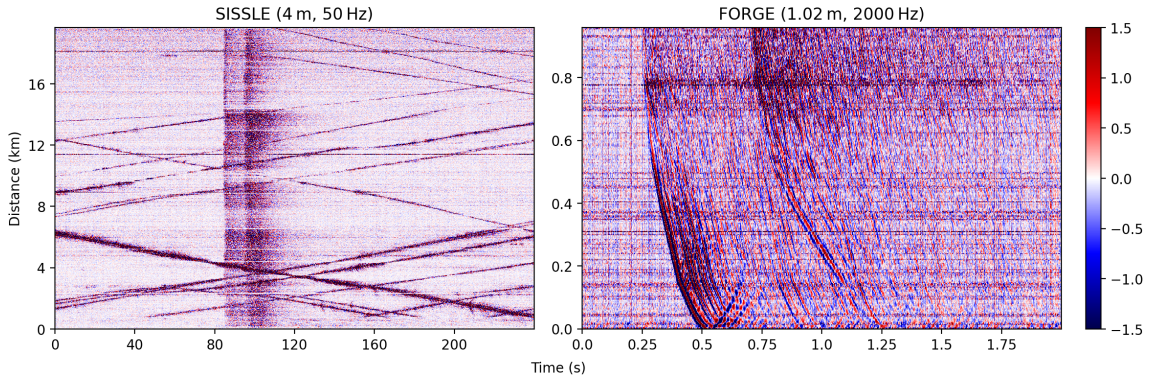


Figure 2.2: Types of noise in different DAS setups. Both measurements capture seismic events; however, the SISSLE data mainly suffers from traffic noise (the diagonal lines). In contrast, erratic and common mode noise (the horizontal and vertical lines, respectively) are most prominent in the FORGE data.

The actual noise characteristics of DAS data not only depend on the environment, but also the measurement parameters such as channel spacing and sample rate, as visualized in Figure 2.2 for data from the Sissle experiment near Haast, New Zealand [11] and the FORGE site in Utah [10].

2.3 Deep Learning

Deep learning is a subfield of machine learning that utilizes deep neural networks to learn complex patterns from data. Over the past decade, deep learning has established itself as the state-of-the-art approach for a wide range of problems across various different fields.

2.3.1 Deep Neural Networks

In its most basic form, a (fully-connected) neural network consists of neurons organized in layers, where each neuron applies a linear transformation followed by a non-linear activation function. The output of a single neuron is given by

$$y = \varphi(\mathbf{w}^T \mathbf{x} + b) \quad (2.2)$$

for an input \mathbf{x} , a weight vector \mathbf{w} , a bias value b and an activation function φ . The activation function is needed in order to avoid the network collapsing into a single linear transformation. The outputs of each layer are then passed as inputs to the next layer. Therefore, a neural network can be described as a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by θ , where θ represents the weights and biases across all layers [4].

In order to optimize these parameters, a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is defined, which measures the difference between the predicted output and the target value. Now, the gradient of the loss function with respect to the parameters, $\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta}$, represents the direction of steepest ascent. Therefore, by moving the parameters in the opposite direction of the gradient, the loss function can be minimized. Typically, the gradient is not calculated for a single data point or for the whole dataset, but instead for a small subset of the dataset, which is why this approach is referred to as (mini-batch) gradient descent. Backpropagation [13] is used to efficiently compute the gradient by making use of the chain rule, enabling fast optimization.

While traditionally neural networks only consisted of a few layers and required hand-crafted features to work effectively, advances in computing power allow modern architectures to automate feature extraction by using additional layers, hence the term *deep* neural network.

2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) [9] are a specific type of neural network that learns features using kernels. Prior to the rise of deep learning, such kernels were designed

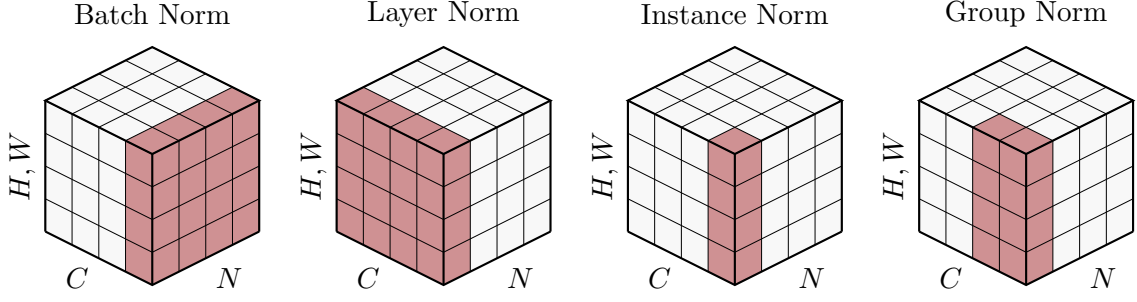


Figure 2.3: Different normalization techniques. N is the batch dimension, C is the channel dimension and H and W are the spatial dimensions of a 4D tensor. The input is normalized across the dimensions highlighted in red. Figure adapted from [18].

manually for various computer vision tasks, for example the Sobel kernel [14] used for edge detection. In CNNs, these kernels are automatically learned from data. In contrast to fully-connected layers, the output of a convolutional layer is obtained by convolution with one or multiple kernels, replacing the matrix multiplication. For a kernel $\mathbf{K} \in \mathbb{R}^{n \times m}$, the convolution is defined as

$$(\mathbf{X} * \mathbf{K})_{i,j} = \sum_n \sum_m X_{i+n,j+m} \cdot K_{n,m}. \quad (2.3)$$

The output of the convolution is then passed through a non-linear activation function, just like in fully-connected layers. CNNs provide two main advantages: First, since the weights are shared across the spatial dimensions, convolutional layers drastically reduce the number of parameters compared to fully-connected layers. Second, convolutions are translationally equivariant, meaning that local patterns in the input can be recognized regardless of their position, which makes CNNs very suitable for image data [4].

2.3.3 Normalization

During the training process, the inputs of each layer change with each iteration as the parameters are optimized. That slows down the training because now each layer has to adapt to the new distribution of its inputs. This process is often referred to as internal covariate shift. To counteract this issue, Ioffe et al. proposed Batch Normalization (BN) [7]. The idea behind BN is to normalize the inputs across the whole mini-batch and their spatial dimensions. The normalized input for a channel i is given by

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu^{(i)}}{\sigma^{(i)}}, \quad (2.4)$$

where $\mu^{(i)}$ and $\sigma^{(i)}$ are the per-channel mean and standard deviation of the mini-batch, respectively. In order to allow the model to learn the identity if that were the optimal transformation, two additional learnable parameters, γ and β , are introduced. The output of the BN layer is then defined as

$$y^{(i)} = \gamma^{(i)} \hat{x}^{(i)} + \beta^{(i)}. \quad (2.5)$$

While BN is widely used, there exist various similar normalization techniques [1, 16, 18] mainly differing in the dimensions across which they are applied. A selection of them is visualized in Figure 2.3.

2.3.4 Attention Mechanisms

In neural networks, some input features are typically more important than others. An attention mechanism helps the network to focus on (*attend to*) the most relevant parts of the input, rather than processing all inputs equally. This works by dynamically reweighting the features based on their importance [12]. While attention is often associated with natural language processing (NLP), especially since the introduction of the Transformer architecture [17] where it is the underlying key principle, it also has applications beyond NLP. In computer vision, for example, it can help CNNs by reweighting feature channels [6] or highlighting important spatial regions.

Chapter 3

Related Work

3.1 Supervised Methods

3.2 Self-Supervised Methods

3.2.1 Target-Based Methods

3.2.2 Blind-Spot-Based Methods

Chapter 4

Methods

Chapter 5

Experimental Setup

Chapter 6

Results

Chapter 7

Discussion

Chapter 8

Conclusion

Bibliography

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML]. URL: <https://arxiv.org/abs/1607.06450>.
- [2] Yangkang Chen et al. “Denoising of Distributed Acoustic Sensing Seismic Data Using an Integrated Framework”. In: *Seismological Research Letters* 94.1 (Nov. 2022), pp. 457–472.
- [3] Tim Dean, T. Cuny, and Arthur Hartog. “Determination of the Optimum Gauge Length for Borehole Seismic Surveys Acquired Using Distributed Vibration Sensing”. In: June 2015. DOI: 10.3997/2214-4609.201412740.
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [5] Arthur H. Hartog. *An Introduction to Distributed Optical Fibre Sensors*. CRC Press, 2017. URL: <http://dx.doi.org/10.1201/9781315119014>.
- [6] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV]. URL: <https://arxiv.org/abs/1709.01507>.
- [7] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: <http://arxiv.org/abs/1502.03167>.
- [8] S Lapins et al. “DAS-N2N: machine learning distributed acoustic sensing (DAS) signal denoising without clean data”. In: *Geophysical Journal International* 236.2 (Nov. 2023), pp. 1026–1041. ISSN: 1365-246X. DOI: 10.1093/gji/ggad460. URL: <http://dx.doi.org/10.1093/gji/ggad460>.
- [9] Y. LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.
- [10] A. Lellouch et al. “Low-Magnitude Seismicity With a Downhole Distributed Acoustic Sensing Array—Examples From the FORGE Geothermal Experiment”. In: *Journal of Geophysical Research: Solid Earth* 126.1 (Dec. 2020). ISSN: 2169-9356. DOI: 10.1029/2020jb020462. URL: <http://dx.doi.org/10.1029/2020JB020462>.

- [11] Meghan S. Miller, John Townend, and Voon Hui Lai. “The South Island Seismology at the Speed of Light Experiment (SISSLE): Distributed Acoustic Sensing Across and Along the Alpine Fault, South Westland, New Zealand”. In: *Seismological Research Letters* (Dec. 2024). ISSN: 0895-0695. DOI: 10.1785/0220240322. eprint: <https://pubs.geoscienceworld.org/ssa/srl/article-pdf/doi/10.1785/0220240322/7068598/srl-2024322.1.pdf>. URL: <https://doi.org/10.1785/0220240322>.
- [12] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. “A review on the attention mechanism of deep learning”. In: *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2021.03.091>. URL: <https://www.sciencedirect.com/science/article/pii/S092523122100477X>.
- [13] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (Oct. 1986), pp. 533–. URL: <http://dx.doi.org/10.1038/323533a0>.
- [14] Irwin Sobel. “An Isotropic 3x3 Image Gradient Operator”. In: *Presentation at Stanford AI Lab (SAIL)* (1968).
- [15] “TODO”. In: ().
- [16] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization”. In: *CoRR* abs/1607.08022 (2016). arXiv: 1607.08022. URL: <http://arxiv.org/abs/1607.08022>.
- [17] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.
- [18] Yuxin Wu and Kaiming He. “Group Normalization”. In: *CoRR* abs/1803.08494 (2018). arXiv: 1803.08494. URL: <http://arxiv.org/abs/1803.08494>.

Appendix A

Appendix