THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Deep neural networks for electromagnetics applied to optimization and inverse problems

Simon Stenmark

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Deep neural networks for electromagnetics applied to optimization and inverse problems

Simon Stenmark

This thesis has been prepared using LaTeX.

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, August 2023

Abstract

This thesis explores the possibilities to complement full-wave electromagnetic solvers with fully-connected neural networks. We emphasize problems that must be solved a very large number of times in a limited parameter domain.

We present and evaluate the normalization method ForwardNorm. ForwardNorm normalizes the outputs of the hidden layers of a neural network and enables training of very deep fully-connected neural networks. To minimize the number of samples needed train the very deep neural networks, we formulate a loss function that includes the misfit in (i) the output of the neural network and (ii) the derivatives of the output of the neural network with respect to its inputs. For certain combinations of input and output, we use continuum sensitivity analysis to compute these derivatives at a low computation cost. We also develop an auto-calibration method that simultaneously determines (i) a set of unknown amplification factors and (ii) the mean permittivity of an unknown medium under test. The method assumes that we have access to a set of measurements that are made a-priori for the purpose of characterization. The method is intended for on-line applications.

We test the methods on four different test-problems. For the first three test-problems, we consider a type of microwave measurement-device intended for an inhomogeneous dielectric medium transported through a metal pipe. In the first test-problem, we train a neural network to determine the point-wise mean and variance of the permittivity of the inhomogeneous dielectric. The trained neural network is very computationally cheap to evaluate, which makes the method appealing for real-time applications. For the second test-problem, we apply the auto-calibration method to simultaneously determine (i) the mean permittivity in the pipe and (ii) a set of unknown amplification factors. For the third test-problem, we use a deep neural network to model the microwave measurement-device with a stochastic dielectric medium and estimate high-dimensional histograms. For the fourth test-problem, we train a deep neural network to model the frequency response of an H-plane waveguide filter as a function of its geometrical parameters. We then use the neural network to optimize the geometry of the filter to achieve pass-band characteristics under geometrical uncertainty.

Keywords: Inverse problems, calibration, optimization, neural networks, deep learning, normalization.

List of Publications

This thesis is based on the following publications:

[A] **Simon Stenmark**, Thomas Rylander, Tomas McKelvey, "Neural Networks for the Estimation of Low-Order Statistical Moments of a Stochastic Dielectric". Published in 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC).

[B] **Simon Stenmark**, Thomas Rylander, Tomas McKelvey, Andrei Ludvig-Osipov, "Very Deep Fully-Connected Neural Networks Applied to Microwave Problems". Submitted for publication in IEEE Transactions on Microwave Theory and Techniques.

[C] Andrei Ludvig-Osipov, **Simon Stenmark**, Thomas Rylander, Tomas McKelvey, "Auto-calibration for near-field microwave measurements". Submitted for publication in IET Science, Measurement & Technology.

Acknowledgments

I would like to thank my supervisors Thomas and Tomas for their support and advice. In particular, I would like to thank my main supervisor Thomas for his endless patience. I realize that your time is extremely valuable, yet you still go above and beyond to support me in my bumpy journey. For this I am ever grateful.

I also want to thank my friends and family. I want to thank my beautiful wife, Moa, for your helpful advice and for always believing in me. I want to thank my lovely daughter, Elisabeth, for being a bundle of joy and always lighting up my day. I want to thank my baby boy, Hilbert, for the cozy snuggles and for being the cutest baby there is. I want to thank my in-laws, Kerstin and Börje, for always being there for us. Finally, I want to thank my parents, Mia and Bo, for your endless support and love. You are the best parents one could ever wish for.

Contents

Abstract i

List of Papers iii

Acknowledgements v

I Overview 1

1 Background 3
1.1 Problem formulation 6
Optimization problems 6
Inverse problems 7
Calibration problems 8

2 Data-driven methods in electromagnetics 11
2.1 Neural networks as surrogate models 12
Deeper neural networks with ForwardNorm 14
Gradient enhanced training of neural networks 15
2.2 Optimization methods based on surrogate models 16
Grid-search method 17
Iterative optimization methods with automatic differentiation . 18

2.3 Domain knowledge in data-driven methods for near-field problems 19
Finite element method 21
Sensitivity analysis 21
Padé approximants 22
2.4 Data generation by computations 23

3 Summary of included papers **25**
3.1 Paper A 25
3.2 Paper B 26
3.3 Paper C 26

4 Summary and Concluding Remarks **29**

References **33**

II Papers **37**

A NN for Stochastic Dielectric **A1**
1 Introduction A3
2 Method A4
2.1 Field problem A5
2.2 Collection and organization of data A6
2.3 Feature extraction A8
2.4 Regression algorithms A8
3 Results A11
3.1 Training, validation and test data set A11
3.2 Hyperparameter tuning for the neural network A11
3.3 Performance of the prediction and their comparison . . A13
4 Conclusion A17
References A19

B GINN **B1**
1 Introduction B3
2 Method B6
2.1 Neural-Network Architecture B6
2.2 Optimization Algorithm B8
2.3 Passive and Reciprocal Microwave Circuits B12

3 Results . . . B14
3.1 Computer Implementation . . . B14
3.2 Test 1 – Stochastic Medium in Circular Cavity . . . B14
3.3 Test 2 – Optimized H-Plane Filter Under Uncertainty . B22
4 Conclusion . . . B32
References . . . B34

C Auto-calibration **C1**
1 Introduction . . . C3
2 Measurement system model and auto-calibration problem . . . C5
2.1 Scattering-parameters model . . . C6
2.2 Noise model . . . C7
2.3 Measurement-domain parametrization . . . C8
2.4 Problem formulation . . . C8
3 Method . . . C10
3.1 Averaging operators . . . C11
3.2 Solving the auto-calibration problem . . . C11
3.3 Local approximation functions . . . C14
4 Numerical examples . . . C15
4.1 Geometry . . . C15
4.2 Signal-to-noise ratio . . . C17
4.3 Amplification matrices . . . C18
4.4 Error assessment . . . C18
4.5 Numerical tests . . . C18
5 Conclusion . . . C20
References . . . C26

Part I

Overview

CHAPTER 1

Background

The advancement of computer hardware makes it possible to solve increasingly complicated computational problems in engineering and science applications. However, new and more sophisticated algorithms are at least as important to be able to handle challenging computational problems. Moore's law predicts that computing power will double roughly ever two years, a prediction that has held up surprisingly well since its formulation. However, this exponential growth of computing power can not be taken for granted. As transistors start to reach atomic scales, it is anticipated that the exponential growth of computing power will start to flatten by 2025 [1]. In this post-Moore era, advancements in algorithmic efficiency become more important than ever to enable continued technological progress.

Maxwell's equations form the foundation of electromagnetic theory, and their efficient solution is of great importance for the development of novel solutions to electromagnetic problems. For most electromagnetic problems, we are unable to find an analytical solution to Maxwell's equations. In these situations, we have to resort to one of the many numerical methods that form the field of Computational Electromagnetics (CEM). In many electromagnetic problems, it is desirable to compute the quantities of interest for the appli-

cation at hand as a function of the frequency and a set of parameters that describe the geometry and materials of the problem, where we wish to learn as much as possible about the behaviour of the problem with respect to these parameters.

The Finite-Differences Time-Domain (FDTD) [2] scheme is one of the most common methods for microwave problems. In the FDTD scheme, we represent the electric field and magnetic field on grids that are staggered with respect to space and time. We then use centered finite-differences to approximate the derivatives in Ampère's law and Faraday's law and perform explicit time-stepping by means of the leap-frog time-stepping scheme. An advantage of the FDTD scheme is that the computer memory can be used almost exclusively for the storage of the electromagnetic field. Another advantage is that we can compute the response for many frequencies in one simulation if we excite the problem with a pulse of large frequency-bandwidth. However, we need to do a completely new simulation if a different waveguide port is excited or if the angle of incidence changes in a scattering problem. We also need to do a new simulation for any change in parameters that describe the material or the geometry of the problem.

The Finite Element Method (FEM) [3] is a popular method for solving Partial Differential Equations (PDEs). Here, we consider the frequency-domain FEM with a single excitation frequency, which is a common formulation for electromagnetic problems. In the FEM, the fields are expanded in basis functions that are defined on finite elements. Then, a weighted average of the residual of the PDE is set to zero. This results in a system of linear equations $\mathbf{Ax} = \mathbf{b}$ that are derived from the boundary value problem, where the sources of the problem are associated with the right-hand side vector $\mathbf{b}$. For sufficiently small problems, we can factorize the matrix $\mathbf{A}$ and reuse it as a different waveguide port is excited or the angle of incidence changes in a scattering problem. Thus, we can change the sources of the problem at the relatively low computational cost associated with a forward and backward substitution given a new vector $\mathbf{b}$. However, we need to do a completely new simulation for any changes to the frequency or the parameters that describe the materials or geometry of the problem.

A third popular CEM method is the Method of Moments (MoM) [4]. Here, we consider the frequency-domain MoM. Based on Maxwell's equations and appropriate information on boundary conditions, a so-called Green's function

is used to formulate an integral equation, where the sources are expanded in basis functions. The weighted average of the residual of the integral equation is set to zero, which results in a system of linear equations. The MoM is well suited for open-boundary problems and for situations where the sources are described by relatively few degrees of freedom compared to the degrees of freedom required to describe the field, such as scattering problems and antenna problems in free space. As a frequency-domain method, the MoM shares many properties with the frequency-domain FEM with regards to the dependencies for the system of linear equations $\mathbf{Ax} = \mathbf{b}$. If we factorize the matrix $\mathbf{A}$, we can reuse the factorization of the matrix for any changes to the right-hand side of the problem, where the sources are associated with the vector $\mathbf{b}$. However, we need to do a completely new simulation for any changes to the frequency or the parameters that describe the materials or geometry of the problem.

For all the considered CEM methods, it is computationally costly to change any parameters that describe the materials or geometry of the problem. In effect, we can only sample discrete points in the parameter space that describes the materials and geometry of the problem. In many applications, we wish to know the behavior of the quantities of interest as a function of material and/or geometry parameters for a specific (and limited) region in the parameter space that describes the material and/or geometry. An approach to achieve this is to fit a data-driven model to data that is computed by a conventional CEM method. The model can then be used to approximate the conventional CEM method for intermediate parameter values. A simple example of such a model is a linear model, where we use gradients of the quantities of interest with respect to the parameters that describe the problem to construct a local model around a linearization point. To extend the range of validity of the model, we can use more expressive models such as higher-order polynomials or rational functions. In recent years, the neural network has been of particular interest as a data-driven black-box model. Neural networks offer powerful generalization capabilities due to their ability to express complicated non-linear relationships between input and output.

In this thesis, we explore possibilities to complement or replace full-wave solvers with computationally inexpensive data-driven models. In particular, we exploit neural networks to model the complicated non-linear relationships present in microwave problems. We train the data-driven models to emu-

late the full-wave solvers in a limited domain of the parameter space, which provides a powerful complement to full-wave solvers.

1.1 Problem formulation

In this thesis, we consider frequency-domain electromagnetic problems with linear materials. We introduce an input vector $\mathbf{x}$ that contains parameters that describe the problem in terms of its geometry, materials, excitation frequency, and so on. We also introduce an output vector $\mathbf{y}$ that contains quantities of interest for the specific problem. The input vector $\mathbf{x}$ and output vector $\mathbf{y}$ are related through a model $\mathbf{f}$ as

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{P}, \tag{1.1}$$

where $\mathcal{P}$ is the *domain* of the model. We refer to the set of attainable $\mathbf{y}$ for $\mathbf{x} \in \mathcal{P}$ as the *range* of the model. While this thesis only considers electromagnetic problems, this formulation encapsulates any type of problem where one or more quantities $\mathbf{y}$ are deterministically dependent on the parameters $\mathbf{x}$ that define the problem.

As an example, we can consider a situation where we want to solve the Helmholtz equation on a domain that contains a homogeneous dielectric. Based on the field solution, we want to compute scattering parameters as a function of the permittivity of the dielectric. Then, the input $\mathbf{x}$ is the permittivity of the dielectric. The output $\mathbf{y}$ is the scattering parameters. Finally, the model $\mathbf{f}$ consists of the solution to the Helmholtz equation and the computations to determine the scattering parameters from the field solution.

Optimization problems

In an *optimization problem*, we wish to find an input $\mathbf{x}_{\text{opt}}$ such that the output $\mathbf{y} = \mathbf{f}(\mathbf{x}_{\text{opt}})$ minimizes a loss function G. We can formulate this as

$$\begin{aligned} \mathbf{x}_{\text{opt}} &= \underset{\mathbf{x}\in\mathcal{P}}{\operatorname{argmin}}\, G(\mathbf{f}(\mathbf{x})) \\ \text{s.t.} \quad &\begin{cases} g_1(\mathbf{x}) \leq 0,\ g_2(\mathbf{x}) \leq 0, \ldots,\ g_m(\mathbf{x}) \leq 0 \\ h_1(\mathbf{x}) = 0,\ h_2(\mathbf{x}) = 0, \ldots,\ h_l(\mathbf{x}) = 0 \end{cases} \end{aligned} \tag{1.2}$$

where we seek to minimize a scalar loss function G under a set of inequality constraints g_i for $i = 1, 2, \ldots, m$ and equality constraints h_j for $j = 1, 2, \ldots, l$. Some inverse and calibration problems can be formulated in terms of the optimization problem (1.2). An optimization problems is called *feasible* if there is a vector $\mathbf{x}$ such that all constraints are fulfilled simultaneously, otherwise the optimization problem is called *infeasible*. For feasible problems, the set of vectors $\mathbf{x}$ where all constraints are fulfilled is then called the *feasible region*. Furthermore, we distinguish between *linear* and *non-linear* optimization problems, as they typically require different solution methods. An optimization problem is only linear if the loss function and all the constraints are linear functions of $\mathbf{x}$, otherwise it is non-linear. In general, the loss function G may have many *local minima*. When we solve an optimization problem, we are interested in finding the *global minimum*. The global minimum of an optimization problem is the smallest value of $G(\mathbf{f}(\mathbf{x}))$ that is attainable for all $\mathbf{x}$ within the feasible region. For many conventional non-linear optimization methods, we provide an initial guess and the method finds a nearby minimum [5]. To avoid poor local minima, it is therefore of crucial importance to have a strategy to obtain a good initial guess.

In Paper B, we optimize the geometry of a waveguide filter to achieve the lowest possible reflection in a pass band. Here, the input $\mathbf{x}$ consists of a set of parameters that specify the geometry of the filter. The output $\mathbf{y}$ is the reflection coefficient of the filter at a set of frequency points. The model $\mathbf{f}$ is the solution to the electromagnetic problem as well as the computations to determine the reflection coefficient from the field solution. Inside the pass band, we define the non-linear loss function $G(\mathbf{y})$ to be the function which returns the largest reflection. Here, we use linear inequality constraints to define bounds for each input parameter. Inside the stop band, we use a non-linear inequality constraint to ensure that the lowest reflection is above a certain threshold. In this problem, there are no equality constraints.

Inverse problems

The *inverse problem* is to, given $\mathbf{y}$, find $\mathbf{x}$ such that $\mathbf{y} = \mathbf{f}(\mathbf{x})$. When we solve an inverse problem, there are complications that we must consider [6]. There might not be an input $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{y}$. For example, the output $\mathbf{y}$ may stem from a true model $\mathbf{f}_0$ and the model $\mathbf{f}$ is an approximation of $\mathbf{f}_0$, or the output $\mathbf{y}$ may contain noise. Conversely, there may be infinitely many inputs

$\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{y}$.

In situations where there is no input $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{y}$, we often attempt to instead find an estimate $\mathbf{x}_{\text{est}}$ that minimizes the misfit between $\mathbf{f}(\mathbf{x}_{\text{est}})$ and $\mathbf{y}$ with respect to some suitable metric. We can formulate this minimization problem in terms of an optimization problem by the loss function $G(\mathbf{u}) = ||\mathbf{u} - \mathbf{y}||$. The minimization problem then becomes

$$\mathbf{x}_{\text{est}} = \underset{\mathbf{x} \in \mathcal{P}}{\operatorname{argmin}} ||\mathbf{f}(\mathbf{x}) - \mathbf{y}||. \tag{1.3}$$

The inversion process might be unstable, or ill-conditioned, such that a small perturbation in the output $\mathbf{y}$ leads to a large change in the estimate $\mathbf{x}_{\text{est}}$. To obtain useful estimates for such problems, we often need to impose additional constraints on the problem to limit the set of valid solutions. This process is known as *regularization*.

We distinguish between *linear* and *non-linear* inverse problems. When $\mathbf{f}$ is a linear function of $\mathbf{x}$, Eq. (1.1) can be written as a system of linear equations $\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a matrix. For linear problems, there are a set of general solution procedures that can be employed. When $\mathbf{f}$ is non-linear, we often require strategies that are specific to the problem at hand. A common strategy to handle non-linear problems is therefore to use iterative methods with local linearization.

In Paper A, we wish to determine statistical properties of an inhomogeneous dielectric given statistical moments of measured scattering parameters. Here, the input $\mathbf{x}$ consists of the statistical properties of the inhomogeneous dielectric that we wish to determine. The output $\mathbf{y}$ contains the statistical moments of the scattering parameters. The model $\mathbf{f}$ then consists of the electromagnetic model as well as the statistical computations used to determine the quantities in $\mathbf{x}$ and $\mathbf{y}$. We then employ data-driven methods to approximate an inverse model $\mathbf{g}$ such that $\mathbf{x}_{\text{est}} = \mathbf{g}(\mathbf{y})$ where $\mathbf{x}_{\text{est}}$ is an estimate of $\mathbf{x}$. For this approach to be successful, we consider a limited domain $\mathcal{P}$ where $\mathbf{f}$ is locally invertible. Then, we obtain an inverse model that we can use to estimate a solution to the inverse problem (1.3) for a limited range of $\mathbf{y}$.

Calibration problems

In physical experiments, there are always errors associated with measurements. The errors can be divided into three main types [7]: (i) random errors;

(ii) systematic errors and (iii) drift errors. Random errors are stochastic in nature, and they are typically caused by electronic noise. We measure the noise level due to random errors in terms of a signal-to-noise ratio (SNR). To reduce the effect of random errors, we can use techniques such as signal averaging [8] to increase the SNR. Systematic errors come from imperfections in the measurement setup, for example from cables and connectors. Drift errors are systematic errors that vary over time, and they are often caused by temperature fluctuations.

We can compensate for systematic errors and drift errors through *calibration* of the measurement system. In the context of microwave measurements, the Vector Network Analyzer (VNA) is an important type of measurement system which is used to measure scattering parameters. There are several methods for calibration of VNAs. In these methods, we typically model the systematic errors and drift errors in terms of unknown parameters in an *error model*. In the calibration process, we then perform measurements on known calibration loads to obtain enough equations to solve for these parameters. It is then possible to use the error model to account for their effect on the measurements. An example of a common calibration method for VNAs is Thru-Reflect-Line (TRL) [9]. TRL is used to calibrate two-port systems with a seven-term error model. Here, measurements are performed on three loads: Thru - a direct connection between the two ports; Reflect - a load with high reflection connected to each of the two ports; and Line - some length of impedance matched transmission line that connect the two ports. Other calibration methods include SOLT [10] and QSOLT [11]. To account for drift errors, the calibration needs to be repeated in regular intervals. This can be problematic in settings with limited access to the measurement system, such as production settings.

In Paper C, we present an auto-calibration method that eliminates the need for a separate calibration phase. In this paper, we wish to use measurements of scattering parameters that are corrupted by systematic/drift errors to determine the average permittivity of an unknown medium under test. We assume that we have access to a series of a-priori characterisation measurements of media with a known permittivity. Once we collect measurements on an unknown medium under test, we then essentially solve the inverse problem and the calibration problem simultaneously. Here, we formulate an optimization problem that includes both the parameters of an error model and the average permittivity of the medium under test. We then exploit the set of charac-

terization measurements to solve the optimization problem. As we gather the characterization measurements in advance, the method is well suited for on-line applications with limited access to the measurement system.

CHAPTER 2

Data-driven methods in electromagnetics

A drawback of most conventional methods in CEM is that they are computationally costly, especially in three space-dimensions. In many solution procedures for inverse and optimization problems the model $\mathbf{f}$ must be evaluated a very large number of times. If the model $\mathbf{f}$ includes a conventional CEM method, this can therefore be a severely limiting factor for a given computational budget.

Another alternative is to replace the conventional CEM method with a computationally inexpensive data-driven model. In this approach, we use a conventional CEM method to create a data set that consists of input-output pairs, or *samples*. We then train a data-driven model on the input-output pairs in a *training set* to make the response of the model match the provided data. The data-driven model itself can incorporate a-priori knowledge of the system in its design, a so-called *grey-box* model, or it can be designed without any prior assumptions, a so-called *black-box* model. If the model is properly chosen, it is able to match the data points well and interpolate correctly between them. To verify this, we typically reserve a portion of the data in a *test set*, which we use to evaluate the trained model. If we choose a model that is inexpensive to evaluate, it can then be a very powerful and useful

complement to solve inverse and optimization problems.

It is imperative that the data used to train a data-driven model is of good quality. This means that we need to consider and appropriately handle any errors that might be present in the data, such as numerical errors that are associated with the numerical method. The data should also be representative of the problem we wish to solve. For instance, when we use a data-driven model to solve an optimization problem, we wish to have a strategy to ensure that there are enough samples in the vicinity of the minimum we wish to find. For problems with small parameter domains, we can simply populate the training set and test set with enough samples to cover the entire parameter domain. For problems with larger parameter domains, this might not be feasible and more sophisticated strategies may be required. In Section 2.4, we discuss an example of such a strategy where we iteratively populate the training set with additional samples in the vicinity of minima that are found by an optimization method. To decrease numerical issues in the training process, it is often beneficial to *normalize* the input and output features. We typically normalize such that each input and output feature has zero mean and unit variance.

2.1 Neural networks as surrogate models

A surrogate model is a computationally inexpensive model that is trained to emulate a detailed model, often in a limited parameter domain. We define the surrogate model $\hat{\mathbf{f}}$ as

$$\hat{\mathbf{y}} = \hat{\mathbf{f}}(\mathbf{x}; \theta) \tag{2.1}$$

where $\hat{\mathbf{y}}$ is an estimate of $\mathbf{y}$ and θ is a set of tunable parameters. When we train the model, we tune the parameters θ of the model such that

$$\hat{\mathbf{f}}(\mathbf{x}; \theta) \approx \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \hat{\mathcal{P}} \subseteq \mathcal{P} \tag{2.2}$$

where $\hat{\mathcal{P}}$ is a subset of the full parameter domain $\mathcal{P}$. Typically, the detailed model $\mathbf{f}(\mathbf{x})$ consists of a conventional CEM method, such as the FEM. Surrogate models can be used by themselves, or as a complement to a detailed model in a space-mapping scheme [12]. Examples of commonly used surrogate models include polynomial response surface models, radial basis function models and support vector regression models [13].

Neural networks have been of particular interest in recent years for their powerful performance as black-box models. Architectures such as convolutional neural networks, recurrent neural networks and auto-encoders [14] have emerged with impressive results for tasks in computer vision and natural language processing. Attention-based mechanisms have recently gained popularity, where transformer-based models [15] achieve state-of-the-art results in tasks such as image classification [16] and machine translation [17]. Neural networks allow for rapid and massively parallel execution on a Graphics Processing Unit (GPU). This makes neural networks very attractive to use as surrogate models in optimization and inverse problems.

In this work, we consider Fully-Connected Neural Networks (FCNN). A FCNN consists of an input layer, one or more hidden layers and an output layer. FCNNs are well suited for regression tasks with a fixed number of inputs. We can describe an L-layer FCNN surrogate model as the function composition

$$\hat{\mathbf{f}} = \mathbf{h}^{(L)} \circ \mathbf{h}^{(L-1)} \circ \ldots \circ \mathbf{h}^{(1)} \tag{2.3}$$

where $\mathbf{h}^{(l)}$ denotes the fully connected layer l. The number of layers L is referred to as the *depth* of the neural network. Here and in the following, we use the super-index (l) in parentheses to indicate that a quantity is associated with fully connected layer l. The operation $\mathbf{f} = \mathbf{g} \circ \mathbf{h}$ is the function composition operation that takes two functions $\mathbf{g}$ and $\mathbf{h}$ and returns a function $\mathbf{f}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{h}(\mathbf{x}))$. The fully connected layer l is given by

$$\mathbf{h}^{(l)} = g\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{2.4}$$

where g is an element-wise non-linear activation function, $\mathbf{W}^{(l)}$ a weight matrix and $\mathbf{b}^{(l)}$ a bias vector. In this description, the layer $l = 1$ is a special case for which we define $\mathbf{h}^{(0)} = \mathbf{x}$. Common choices for the activation function g include Rectified Linear Units (ReLU) [14] and hyperbolic tangent functions. When we train an FCNN, we tune the elements in the weight matrices and bias vectors to make the function (2.3) described by the FCNN match the input-output pairs in the training set. The powerful hardware of today allows training of very deep networks with the capability to express complicated non-linear relationships between input and output.

When we train deep networks there are potential complications that we must consider. For deep neural networks, we might run into numerical issues

such as the vanishing gradient problem [18]. For networks with very many degrees of freedom and the capability to express complicated input-output relationships, we also risk *overfitting* the neural network to the training data. Overfitting is when a neural network too closely memorizes the samples in the training set and fails to give accurate estimates $\hat{\mathbf{y}}$ when presented with unseen inputs $\mathbf{x}$. A clear indicator of overfitting is when the error on the training set is much smaller than the error on the test set. Overfitting can happen when the number of samples in the training set is too small in relation to the degrees of freedom and expressive capabilities of the neural network. Typically, the solution is to add more samples to the training set. However, these samples can be expensive to obtain. Therefore, the number of samples is often the primary limiting factor for how expressive models that we can successfully train.

Here, we present a normalization method that enables the training of very deep FCNNs. We also present a loss function which also incorporates the derivatives of the output $\mathbf{y}$ with respect to the input $\mathbf{x}$ in the training of a neural network to decrease the number of samples necessary for training. The two methods together makes it possible to train very deep neural networks with comparatively few samples.

Deeper neural networks with ForwardNorm

Normalization between hidden layers in neural networks is an important technique to train deep neural networks. Important examples of normalization techniques include BatchNorm [19] and LayerNorm [20]. In these techniques, we insert normalization layers between the fully-connected layers that transform each element of the layer output as

$$\tilde{h}_j^{(l)} = \frac{h_j^{(l)} - \mu_j^{(l)}}{\sigma_j^{(l)}}. \tag{2.5}$$

Here, $\mu_j^{(l)}$ and $\sigma_j^{(l)}$ are normalization parameters and $\tilde{h}_j^{(l)}$ is element j of the normalized output of fully connected layer l. The statistics of $h_j^{(l)}$ typically shift during training of the neural network which means that $\mu_j^{(l)}$ and $\sigma_j^{(l)}$ need to be repeatedly updated during training for the normalization to be effective. The specifics of how the normalization parameters are determined

differ between normalization techniques. However, FCNNs are difficult to normalize with this kind of normalization technique as the updates of the normalization parameters perturb the training process [21].

In Paper B, we present a novel normalization procedure that we refer to as *ForwardNorm*. Here, we insert normalization layers (2.5) between each fully connected layer. During the training of the neural network, we repeatedly update the parameters of the normalization layers. After each update of the normalization parameters, we also update the weight matrices and bias vectors of the neural network such that the function $\hat{\mathbf{f}}$ remains unchanged by the update. In this way, we reparametrize the weight space of the neural network with no effect on its output. The procedure lessens the perturbation of the training process by the update of the normalization parameters which allows us to efficiently train very deep FCNNs.

Gradient enhanced training of neural networks

If we also have access to the Jacobian matrix

$$J_{pq} = \frac{\partial y_p}{\partial x_q} \tag{2.6}$$

for the samples in our training and test sets, we can include it in the loss function that we use when we train the neural network [22]. This can massively reduce the number of samples needed to successfully train the network. In situations where the derivatives are computationally inexpensive to obtain, this can allow training much more expressive neural networks within a given computational budget.

In Paper B, we exploit automatic differentiation to differentiate (2.3) with respect to the input $\mathbf{x}$ as shown in Figure 2.1. As we evaluate the neural network, we then obtain both the function values $\hat{y}_p$ and the Jacobian matrix

$$\hat{J}_{pq} = \frac{\partial \hat{y}_p}{\partial x_q} \tag{2.7}$$

simultaneously. To include the derivatives in the training of the neural network, we formulate the total loss function

$$L = F(\mathbf{y}, \hat{\mathbf{y}}) + \alpha G(\mathbf{J}, \hat{\mathbf{J}}) \tag{2.8}$$

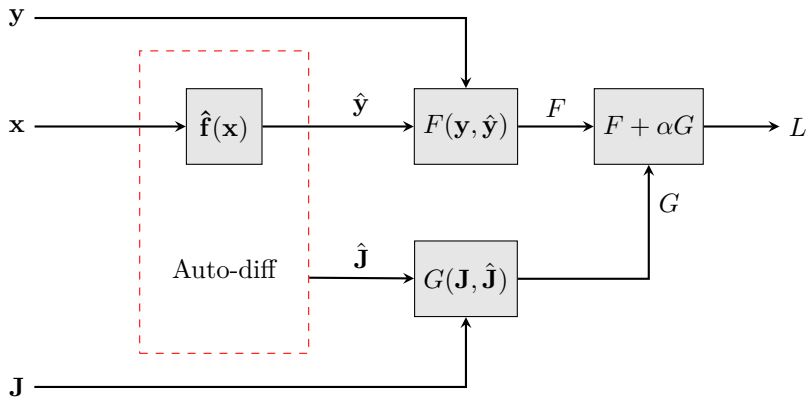

Figure 2.1: Flowchart illustrating how the loss L is computed from the input $\mathbf{x}$, the output $\mathbf{y}$ and the Jacobian matrix $\mathbf{J}$. Here, the neural network is denoted by $\hat{\mathbf{f}}(\mathbf{x})$. The Jacobian $\hat{\mathbf{y}}$ is computed using automatic differentiation.

which is a linear combination of two loss terms: $F(\mathbf{y}, \hat{\mathbf{y}})$ - the misfit of the function values; and $G(\mathbf{J}, \hat{\mathbf{J}})$ - the misfit of their derivatives. The weight α is considered a hyper-parameter that needs to be tuned. For the test-problems that we consider, the number of samples that are required to train the neural network is significantly reduced when we include the derivatives in the loss function. In Section 2.3, we describe how we use continuum sensitivity analysis to compute these derivatives at a very low computational cost for certain electromagnetic problems.

2.2 Optimization methods based on surrogate models

Surrogate models open exciting possibilities to solve optimization problems. In general, methods for non-linear optimization deal with local optimization, which locates a minimum in the vicinity of an initial guess [5]. To avoid poor local minima, it is therefore necessary to find a good initial guess before we apply any local optimization methods. Here, a computationally efficient surrogate model can be of great benefit. If the surrogate model is differentiable,

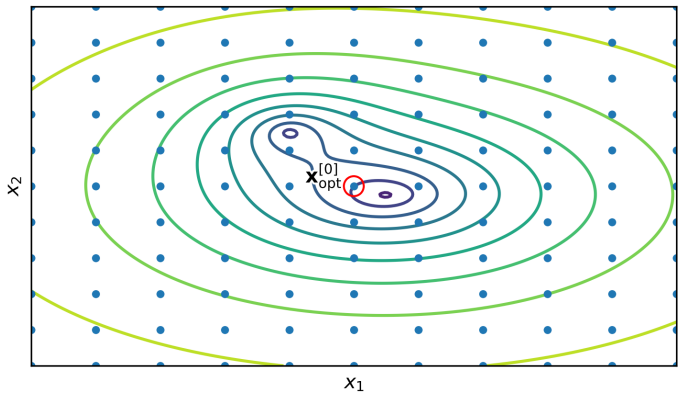

Figure 2.2: Example of a grid-search method for a two-dimensional parameter vector $\mathbf{x}$. The level curves indicate value of the goal function $G\left(\hat{\mathbf{f}}\left(\mathbf{x}\right)\right)$. The blue dots indicate the set of candidates $\mathbf{x}^{\{i\}}$ for $i = 1, 2, \ldots, N$ which are spaced on an equi-distant grid. The red circle shows the initial guess $\mathbf{x}_{\text{opt}}^{[0]}$, which is the candidate $\mathbf{x}^{\{i\}}$ that gives the lowest value of $G\left(\hat{\mathbf{f}}\left(\mathbf{x}^{\{i\}}\right)\right)$.

we can also compute the Jacobian matrix (2.7) for any input $\mathbf{x}$.

Here, we discuss how a computationally efficient surrogate model allows us to massively sample the parameter domain to find an attractive initial guess. We also discuss how the Jacobian matrix can be used to formulate iterative methods for local optimization.

Grid-search method

When we solve a non-linear optimization problem, we often have little to no prior knowledge of the behavior of the loss function that we wish to minimize. When this is the case, we must sufficiently explore the parameter domain to find a good initial guess. A simple approach is to find an initial guess $\mathbf{x}_{\text{opt}}^{[0]}$ as

$$\mathbf{x}_{\text{opt}}^{[0]} = \underset{\mathbf{x}^{\{i\}}}{\operatorname{argmin}}\, G\left(\hat{\mathbf{f}}\left(\mathbf{x}^{\{i\}}\right)\right), \quad i \in \{1 \ldots N\} \tag{2.9}$$

for some suitable set of candidates $\mathbf{x}^{\{i\}}$ for $i = 1, 2, \ldots, N$. Here, we should choose N large enough to sufficiently explore the parameter space. Figure 2.2 shows an example of this, where the candidates $\mathbf{x}^{\{i\}}$ for a two-input model are chosen on a Cartesian grid. With a sufficiently small grid-spacing, grid-search methods allow thorough exploration of the parameter domain. However, the number of function evaluations grows very rapidly when the grid-spacing is decreased, particularly for high-dimensional input. Here, a computationally efficient surrogate model is of great benefit.

In Paper B, we train a neural network to use as a surrogate model for the optimization of a waveguide filter. As the neural network is very inexpensive to evaluate, we can sufficiently sample the parameter domain to find an initial guess for the geometry parameters. We then iteratively refine the guess to obtain a better estimate.

Iterative optimization methods with automatic differentiation

Iterative optimization methods based on a search direction are an important kind of local optimization methods that exploit the updating scheme

$$\mathbf{x}_{\text{opt}}^{[n+1]} = \mathbf{x}_{\text{opt}}^{[n]} - \gamma^{[n]} \mathbf{d}^{[n]} \tag{2.10}$$

where we, for each iteration n, take a step of length $\gamma^{[n]}$ in direction $\mathbf{d}^{[n]}$ [23]. Examples of such methods are the method of steepest descent, Newton's method and line-search methods [5], where different choices of method dictate how γ_n and $\mathbf{d}_n$ are chosen. When we apply such methods, it is of great benefit to have access to the gradient $\nabla_{\mathbf{x}} G$. We note that

$$\nabla_{\mathbf{x}} G = \hat{\mathbf{J}}^T \nabla_{\hat{\mathbf{y}}} G \tag{2.11}$$

and thus, as long as we can compute $\nabla_{\hat{\mathbf{y}}} G$, we can use the Jacobian $\hat{\mathbf{J}}$ to compute $\nabla_{\mathbf{x}} G$.

Here, it is advantageous to use neural networks as surrogate models, since then $\hat{\mathbf{J}}$ can cheaply be evaluated through automatic differentiation. For conventional CEM methods, the derivatives in $\mathbf{J}$ can be costly to obtain. Some derivatives can be obtained through continuum sensitivity analysis as discussed in Section 2.3, but this is not always possible. While it is possible to estimate $\hat{\mathbf{J}}$ through finite-difference approximations of the derivatives, this

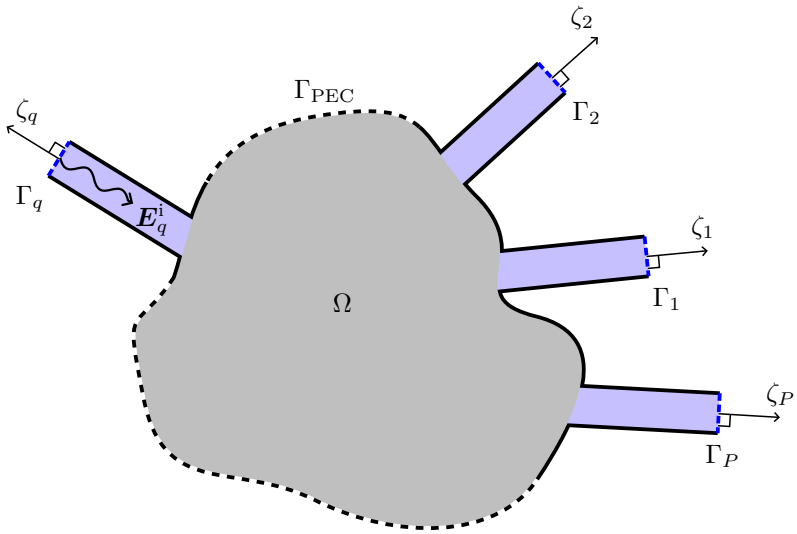

Figure 2.3: The boundary value problem under consideration. The enclosed volume Ω is surrounded by the PEC surface Γ_{PEC}. The volume is connected to P waveguides that are truncated by waveguide ports that coincide with the surfaces $\Gamma_1, \Gamma_2, \ldots, \Gamma_P$. Waveguide port q is excited by the incident wave $\boldsymbol{E}_q^{\text{i}}$. The local longitudinal coordinates $\zeta_1, \zeta_2, \ldots, \zeta_P$ are directed normally outwards from $\Gamma_1, \Gamma_2, \ldots, \Gamma_P$.

approach is computationally costly and prone to numerical errors. With automatic differentiation, we can compute $\hat{\mathbf{J}}$ for any combination of quantities in the input $\mathbf{x}$ and the output $\hat{\mathbf{y}}$ at an inexpensive computational cost. It is also possible to compute higher-order derivatives through automatic differentiation. This is useful in optimization methods such as Newton's method [5], where the update scheme involves the inverse of the Hessian matrix.

2.3 Domain knowledge in data-driven methods for near-field problems

In this thesis, we consider frequency-domain problems within an enclosed volume Ω that is enclosed by PEC surfaces Γ_{PEC} as shown in Figure 2.3. The volume is connected to P waveguides that are truncated by waveguide ports

that coincide with the boundary surfaces $\Gamma_1, \Gamma_2, \dots \Gamma_P$. The waveguides have identical, constant cross-sections and are filled with a homogeneous dielectric. Here, we consider situations where the frequency ω is within the frequency range where only the dominant mode of the waveguides is propagating. We also assume that the length of the waveguide is sufficiently large to make all evanescent higher-order modes have a negligible amplitude at the waveguide port. In this setting, we consider the boundary value problem

$$\nabla \times \nabla \times \boldsymbol{E} - \mu_0(\omega^2\epsilon - j\omega\sigma)\boldsymbol{E} = \boldsymbol{0} \quad \text{in } \Omega, \tag{2.12}$$

$$n \times \boldsymbol{E} = \boldsymbol{0} \quad \text{on } \Gamma_{\text{PEC}}, \tag{2.13}$$

$$n \times \nabla \times \boldsymbol{E} + \gamma_m n \times n \times \boldsymbol{E} = 2\gamma_m n \times n \times \boldsymbol{E}_q^{\text{i}} \quad \text{on } \Gamma_q, \tag{2.14}$$

$$n \times \nabla \times \boldsymbol{E} + \gamma_m n \times n \times \boldsymbol{E} = \boldsymbol{0} \quad \text{on } \Gamma_p \forall p \neq q. \tag{2.15}$$

The electric field $\boldsymbol{E}$ is governed by a vector Helmholtz' equation. The materials in Ω are modeled by the permeability in vacuum μ_0, permittivity ϵ and conductivity σ. Here, the permittivity ϵ and conductivity σ may vary in space. The unit normal n is directed outwards from the volume Ω. We use a Dirichlet boundary condition on the PEC surfaces Γ_{PEC}. On the surfaces $\Gamma_1, \Gamma_2, \dots \Gamma_P$ that coincide with the waveguide ports, we use Robin boundary conditions. Here, γ_m is the complex propagation constant of the dominant waveguide mode $\boldsymbol{m}$ [24]. Port q is excited by the incident electric field

$$\boldsymbol{E}_q^{\text{i}} = E_{0,q}^{\text{i}} e^{+\gamma_m \zeta_q} \boldsymbol{m} \tag{2.16}$$

where ζ_q is the local longitudinal coordinate associated with port q as shown in Figure 2.3. We define the *scattering parameter* S_{pq} as the ratio

$$S_{pq} = \frac{E_{0,p}^{\text{r}}}{E_{0,q}^{\text{i}}} \tag{2.17}$$

where $\boldsymbol{E}_p^{\text{r}} = E_{0,p}^{\text{r}} e^{-\gamma_m \zeta_p} \boldsymbol{m}$ is the reflected electric field at port S_p. To compute all scattering parameters, we need to excite each port once, meaning that we need to solve P boundary value problems. The matrix $\mathbf{S}$ that contains all scattering parameters is referred to as the *scattering matrix*.

Finite element method

We solve the boundary value problem (2.12)-(2.15) with the FEM. Find the electric field $\boldsymbol{E} \in H_0(\mathrm{curl}; \Omega)$ such that

$$a(\boldsymbol{v}, \boldsymbol{E}) = b(\boldsymbol{v}), \quad \forall\ \boldsymbol{v} \in H_0(\mathrm{curl}; \Omega) \tag{2.18}$$

where

$$\boldsymbol{H}(\mathrm{curl}, \Omega) = \left\{ \boldsymbol{E} : \left(\int_\Omega |\boldsymbol{E}|^2 \, d\Omega < \infty \right) \text{and} \left(\int_\Omega |\nabla \times \boldsymbol{E}|^2 \, d\Omega < \infty \right) \right\} \tag{2.19}$$

$$\boldsymbol{H}_0(\mathrm{curl}, \Omega) = \{ \boldsymbol{E} \in \boldsymbol{H}(\mathrm{curl}, \Omega) : \hat{n} \times \boldsymbol{E} = \boldsymbol{0} \text{ on } \Gamma_{\mathrm{PEC}} \} \tag{2.20}$$

For the boundary value problem (2.12)-(2.15), the bilinear form a and linear form b become

$$\begin{aligned} a(\boldsymbol{v}, \boldsymbol{E}) = &\int_\Omega \left[(\nabla \times \boldsymbol{v}) \cdot (\nabla \times \boldsymbol{E}) - \mu_0(\omega^2 \epsilon - j\omega\sigma) \boldsymbol{v} \cdot \boldsymbol{E} \right] d\Omega \\ &+ \gamma_m \sum_{p=1}^{P} \int_{\Gamma_p} (n \times \boldsymbol{v}) \cdot (n \times \boldsymbol{E}) \, d\Gamma \end{aligned} \tag{2.21}$$

$$b(\boldsymbol{v}) = 2\gamma_m \int_{\Gamma_q} (n \times \boldsymbol{v}) \cdot (n \times \boldsymbol{E}_q^{\mathrm{i}}) \, d\Gamma. \tag{2.22}$$

Next, we expand $\boldsymbol{E}$ in the curl-conforming basis functions $\boldsymbol{N}_j \in \boldsymbol{H}_0(\mathrm{curl}, \Omega)$ [3] according to

$$\boldsymbol{E}(\boldsymbol{r}) = \sum_j E_j \boldsymbol{N}_j(\boldsymbol{r}). \tag{2.23}$$

We use Galerkin's method and choose the weighting functions from the set of basis functions $\boldsymbol{N}_j$. Finally, Equation (2.18) yields a system of linear equations with the unknown and sought E_j.

Sensitivity analysis

We can expand the scattering parameter (2.17) around a linearization point $\mathbf{x}_{\mathrm{lin}}$ as

$$S_{pq}(\mathbf{x}_{\mathrm{lin}} + \delta\mathbf{x}) = S_{pq}(\mathbf{x}_{\mathrm{lin}}) + \delta S_{pq}(\mathbf{x}_{\mathrm{lin}}, \delta\mathbf{x}) + \ldots \tag{2.24}$$

where $\delta S_{pq}(\mathbf{x}_{\text{lin}}, \delta\mathbf{x})$ is the *sensitivity* of scattering parameter S_{pq} with respect to the linearization point $\mathbf{x}_{\text{lin}}$ and $\delta\mathbf{x}$ is an arbitrary infinitesimal perturbation. If we neglect the higher order terms, we obtain the linear model

$$S_{pq}(\mathbf{x}_{\text{lin}} + \delta\mathbf{x}) \approx S_{pq}(\mathbf{x}_{\text{lin}}) + \delta S_{pq}(\mathbf{x}_{\text{lin}}, \delta\mathbf{x}). \tag{2.25}$$

If we have access to the sensitivity $\delta S_{pq}(\mathbf{x}_{\text{lin}}, \delta\mathbf{x})$, we can use (2.25) as a locally applicable model of S_{pq}.

We can apply sensitivity analysis to compute the sensitivity $\delta S_{pq}(\mathbf{x}_{\text{lin}}, \delta\mathbf{x})$ from the field solutions of the *original* problem and an *adjoint* problem. For reciprocal microwave problems, the field solutions to all adjoint problems are available once the full scattering matrix is computed. Then, we can compute all the corresponding sensitivities without the need to solve any additional field problems. In comparison to the computational cost associated with the solution of the field problem, the computational cost associated with the evaluation of the sensitivity is negligible since it only involves a weighted inner product between two (already available) field solutions.

In Paper B, we present two examples of sensitivity expressions. In the first example, the input $\mathbf{x}$ describes the permittivity of an inhomogeneous material. Here, the elements of $\mathbf{x}$ constitute the coefficients of a weighted sum of basis functions that describes the permittivity. In the second example, the input $\mathbf{x}$ contains parameters that describe the geometry of a waveguide filter in terms of lengths and widths. In both examples, we use sensitivity analysis to compute the Jacobian matrix $\mathbf{J}$ of $\mathbf{y}$ with respect to $\mathbf{x}$. We then use $\mathbf{J}$ to aid the training of a neural network as described in Section 2.1.

Padé approximants

Another class of model for a scattering parameter S_{pq} is the *Padé approximant*. In general, the Padé approximant of order $[m/n]$ for a function $f(x)$ is a rational function that approximates $f(x)$ as

$$f(x) \approx \frac{\sum_{i=0}^{m} b_i x^i}{1 + \sum_{j=1}^{n} a_j x^j}, \tag{2.26}$$

where $m \geq 0$ and $n \geq 1$.

In frequency response applications, it is natural to set $m = n - 1$ [25]. In

this case, we can write the Padé approximant as the pole-residue expansion

$$S_{pq}(\omega) \approx \sum_{k=1}^{n} \frac{r_{pq,k}}{\omega - \omega_k}. \tag{2.27}$$

where the scattering parameter $S_{pq}(\omega)$ is characterized by the n poles ω_k and the residues $r_{pq,k}$. We assume that all scattering parameters share a common set of poles [26] and thus omit the indices pq in the denominator of Eq. (2.27). With the help of algorithms such as FSID [27] or Padé-via-Lanczos [25], we can identify the residues and poles from samples in a training set. Here, the number of samples that are needed for identification increases with the model order n. For noisy data, we can increase the number of samples that we use for identification for a more robust approximation.

The Padé approximant is useful as an interpolant for scattering parameters as a function of frequency. Here, it is often sufficient to identify the dominant pole-residue pairs in the frequency range of interest. Then, the Padé approximant can be used to approximate the scattering parameters at additional frequencies at a low computational cost. As an additional benefit, it is simple to differentiate the Padé approximant with respect to frequency to compute the derivative of the approximated scattering parameters with respect to frequency. We can then use these derivatives to train a gradient-informed neural network.

2.4 Data generation by computations

It is advantageous in several ways to generate data through computations. For supervised learning, all parameters in $\mathbf{x}$ that define the output $\mathbf{y}$ must be known, which is also necessary when the electromagnetic field problem is solved. For all samples, we thus have ready access to the parameters $\mathbf{x}$ that define the output $\mathbf{y}$. Additionally, it is simple to generate more data as needed. This is useful to tailor the size of the data set to the application at hand, and to only generate as much data as needed to minimize the necessary computations.

There are several important considerations when we wish to use data generated through computations to train a data-driven model. It is essential that we are in control of the errors present in the simulations. For most compu-

tational methods, we can reduce the error at the cost of additional computational resources in terms of computer memory and floating-point operations. For example, in the FEM we can make the mesh finer or increase the order of the basis functions to reduce the error. In this way, we can achieve an error on acceptable levels for the application at hand. When we generate data to train a surrogate model for use in an optimization problem, it is important that we choose a parameter domain $\mathcal{P}$ that contains input vectors that are relevant for the optimization problem at hand. Here, it is very useful to have a-priori knowledge of a reference input vector $\mathbf{x}_{\text{ref}}$ that is known to be close to a minimum of the optimization problem at hand. Then, we can choose a parameter domain in the vicinity of $\mathbf{x}_{\text{ref}}$ under the assumption that this vicinity contains other minima of interest.

In Paper B, we present a procedure to iteratively re-train a surrogate model which is used to optimize the geometry of a waveguide filter to find a set of filter geometries that correspond to good pass-band filters. We choose a reference input $\mathbf{x}_{\text{ref}}$ that corresponds to a filter geometry that is known to give reasonable pass-band characteristics. We then generate an initial training set in the vicinity of $\mathbf{x}_{\text{ref}}$. Then, we use the initial training set to train the surrogate model. Once we apply the optimization algorithm and find a set of minima, we populate the training set with additional samples within the vicinity of these minima. We then re-train the surrogate model and optimize the geometry once again with the previously found minima as a starting point. This procedure significantly reduces the error of the surrogate model in the vicinity of the minima.

CHAPTER 3

Summary of included papers

This chapter provides a summary of the included papers.

3.1 Paper A

Simon Stenmark, Thomas Rylander, Tomas McKelvey
Neural Networks for the Estimation of Low-Order Statistical Moments of a Stochastic Dielectric
Published in 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC),
pp. 1–6, May. 2021.
 DOI: 10.1109/I2MTC50364.2021.9459996 .

In this paper, we train a neural network to solve an inverse problem. We consider a problem with an inhomogeneous stochastic medium that is defined by the point-wise mean and standard deviation of its permittivity. Here, we assume that we have access to the scattering matrix computed for each realization of the stochastic permittivity. We train a fully-connected neural network to estimate point-wise mean and standard deviation of the permit-

tivity given statistical moments of the scattering parameters. We demonstrate the method on a numerical example with a cylindrical geometry surrounded by four parallel-plate waveguide.

3.2 Paper B

Simon Stenmark, Thomas Rylander, Tomas McKelvey, Andrei Ludvig-Osipov
Very Deep Fully-Connected Neural Networks Applied to Microwave Problems
Submitted for publication in IEEE Transactions on Microwave Theory and Techniques.

In this paper, we present a method to train neural networks for microwave problems. We introduce the normalization technique ForwardNorm which enables training of very deep fully-connected neural networks. We use a loss function that includes the Jacobian matrix of the output with respect to the input to reduce the number of samples that are needed to train the neural networks. Here, we use continuum sensitivity analysis to compute the derivatives in the Jacobian matrix. The method is demonstrated on two numerical examples. We show that ForwardNorm enables training of networks 30 layers deep and that we considerably reduce the number of samples needed to train a network by including the Jacobian matrix in the training. Finally, we use a surrogate model to optimize the geometry of an H-plane waveguide filter.

3.3 Paper C

Andrei Ludvig-Osipov, **Simon Stenmark**, Thomas Rylander, Tomas McKelvey
Auto-calibration for near-field microwave measurements
Submitted for publication in IET Science, Measurement & Technology.

In this paper, we present an auto-calibration method for scattering-parameter measurements. Here, we simultaneously estimate the average permittivity in the measurement domain and a set of unknown amplification factors. To do this, the method utilises a set of a priori calibration measurements. We

demonstrate the method on two numerical examples with cylindrical geometries surrounded by four and six parallel-plate waveguides respectively.

CHAPTER 4

Summary and Concluding Remarks

Machine learning and has revolutionized many technological fields, where today's powerful hardware and recent developments in data-driven algorithms open up exciting new possibilities. However, in the field of computational electromagnetics, machine-learning methods are still relatively unexplored. This thesis examines the possibilities to complement conventional computational electromagnetic methods with data-driven models. In particular, we focus our work on neural networks due to their powerful generalization capabilities. We emphasize problems that must be solved a very large number of times in a limited parameter domain, where the data-driven models can be of particular benefit.

We present and evaluate a normalization procedure that we refer to as ForwardNorm. ForwardNorm consists of (i) normalization layers inserted between the hidden layers of a fully-connected neural network and (ii) a custom update procedure. In the first part of the update procedure, we optimize the weight matrices and bias terms of the fully-connected layers using a conventional stochastic steepest-descent optimizer. In the second part, we update the parameters of the normalization layers to ensure that the outputs of each hidden layers have zero mean and unit variance. Then, we update the weight

matrices and bias vectors of the neural network such that its output remains unchanged by update of the normalization layers. ForwardNorm reduces numerical issues that are associated with training deep neural networks and enables us to efficiently train very deep fully-connected neural networks. To minimize the number of samples that are required to train the neural networks, we formulate a loss function that is a linear combination of the misfit in (i) the output of the network and (ii) the derivatives of the outputs of the network with respect to its inputs. By including the misfit in the derivatives in the training of the neural networks, the number of samples required for training is substantially reduced. Here, we apply continuum sensitivity analysis to compute derivatives of scattering parameters with respect to parameters that describe either (i) the permittivity within the computational domain or (ii) the geometry of the conducting walls that surround the computational domain. For reciprocal problems, the sensitivity analysis can be performed at a very low computational cost. We also develop an auto-calibration method that is suitable for on-line applications. Here, we assume that we have access to a set of a-priori characterization measurements that we exploit to simultaneously determine (i) a set of unknown amplification factors and (ii) the mean permittivity of an unknown medium under test.

We test the methods on four different test-problems. For the first three test-problems, we consider a type of microwave measurement device intended for an inhomogeneous dielectric medium transported through a metal pipe [28], [29]. The problem consists of a circular cavity connected to a number of parallel-plate waveguides, where the circular cavity contains an inhomogeneous dielectric. In the first test-problem, we train a neural network to solve an inverse problem. Here, we determine the point-wise mean and variance of the permittivity in the measurement domain given statistical moments of measured scattering parameters. The trained neural network is very computationally inexpensive to evaluate, which makes the method appealing for real-time measurements. For the second test-problem, we apply the auto-calibration method to determine the mean permittivity in the pipe while we simultaneously determine a set of unknown amplification factors related to a measurement system. We evaluate the method for a range of noise levels. As all characterization measurements are performed in advance, the method is well-suited for online measurement applications with limited access to the measurement system. For the third test-problem, we use a deep neural net-

work to model the device and estimate high-dimensional probability distribution functions and their histograms. Here, we benefit from the very deep neural networks that ForwardNorm allows us to train. Histograms such as these can for example be used in detection problems [28]. For the fourth test-problem, we train a deep neural network to model the frequency response of an H-plane waveguide filter. The filter features five cavities and has two symmetry planes. We use the neural network to optimise the geometry of the filter to achieve the lowest possible reflection in the pass band. Inside the stop band, we put a constraint on the stop-band frequencies to ensure that the smallest reflection is above a certain threshold. Here, we benefit from being able to massively parallelize the evaluation of the neural network. Through brute force, we can explore the design space to find a good starting guess for the optimization algorithm. We augment the data set with geometries in the vicinity of the optimized geometries to ensure good performance of the neural network.

References

[1] J. Shalf, "The future of computing beyond Moore's Law," *Philos. trans., Math. phys. eng. sci.*, vol. 378, no. 2166, p. 20 190 061, Jan. 2020.

[2] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed. Boston: Artech House, 2005.

[3] J. M. Jin, *The Finite Element Method in Electromagnetics*, 3rd ed. Hoboken, NJ: John Wiley & Sons, 2014.

[4] W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston: Artech House, 2001.

[5] L. E. Scales, *Introduction to Non-Linear Optimization*. Heidelberg, Germany: Springer-Verlag, 1985.

[6] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*. Amsterdam: Amsterdam: Elsevier, 2018.

[7] A. Technologies, "AN 1287-3 applying error correction to network analyzer measurements," Tech. Rep.

[8] A. V. Oppenheim, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[9] G. F. Engen and C. A. Hoer, "Thru-reflect-line: An improved technique for calibrating the dual six-port automatic network analyzer," *IEEE Trans. Microw. Theory Tech.*, vol. 27, no. 12, pp. 987–993, 1979.

[10] M. Zeier, D. Allal, and R. Judaschke, "Guidelines on the evaluation of vector network analyzers (VNA)," *EURAMET Calibration Guide*, vol. 3, no. 12, pp. 507–521, 2018.

[11] A. Ferrero and U. Pisani, "QSOLT: A new fast calibration algorithm for two port S parameter measurements," in *38th ARFTG Conference Digest*, vol. 20, IEEE, 1991, pp. 15–24.

[12] J. Bandler, R. Biernacki, S. H. Chen, P. Grobelny, and R. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 12, pp. 2536–2544, Dec. 1994.

[13] P. Jiang, Q. Zhou, and X. Shao, *Surrogate Model-Based Engineering Design and Optimization*. New York: Springer, 2020.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[16] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. on Computer Vision*, 2021, pp. 357–366.

[17] Q. Wang, B. Li, T. Xiao, *et al.*, "Learning deep transformer models for machine translation," 2019, arXiv preprint arXiv:1906.01787.

[18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, May 2010, pp. 249–256.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Mar. 2015, arXiv:1502.03167 [cs].

[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016, arXiv:1607.06450 [cs, stat].

[21] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," Sep. 2017, arXiv:1706.02515 [cs.LG].

[22] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Świrszcz, and R. Pascanu, "Sobolev training for neural networks," Jul. 2017, arXiv:1706.04859 [cs.LG].

[23] A. Ruszczynski, *Nonlinear Optimization*. Princeton, NJ: Princeton University Press, 2011.

[24] D. M. Pozar, *Microwave Engineering*. Hoboken, NJ: John Wiley & Sons, 2011.

[25] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, 1995.

[26] C. C. Chou and J. E. Schutt-Aine, "Equivalent Circuit Synthesis of Multiport S Parameters in Pole-Residue Form," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 11, no. 11, pp. 1971–1979, Nov. 2021.

[27] T. McKelvey and M. Gibanica, "FSID - A Frequency Weighted MIMO Frequency Domain Identification and Rational Matrix Approximation Method for Python, Julia and Matlab," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 403–408, Jan. 2021.

[28] J. Nohlert, T. Rylander, and T. McKelvey, "Microwave measurement system for detection of dielectric objects in powders," *IEEE Trans. Microw. Theory Tech.*, vol. 64, no. 11, pp. 3851–3863, 2016.

[29] J. Winges, L. Cerullo, T. Rylander, T. McKelvey, and M. Viberg, "Compressed sensing for the detection and positioning of dielectric objects inside metal enclosures by means of microwave measurements," *IEEE Trans. Microw. Theory Tech.*, vol. 66, no. 1, pp. 462–476, 2017.