Deep Learning for Inverse Problems: Performance Characterizations, Learning Algorithms, and Applications

Jaweria Amjad

A dissertation submitted in partial fulfillment
of the requirements for
Doctor of Philosophy
at the

Department of Electronics & Electrical Engineering
University College London

March 15, 2022

I, Jaweria Amjad, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Deep learning models have witnessed immense empirical success over the last decade. However, in spite of their widespread adoption, a profound understanding of the generalization behaviour of these over-parameterized architectures is still missing. In this thesis, we provide one such way via a data-dependent characterizations of the generalization capability of deep neural networks based data representations. In particular, by building on the algorithmic robustness framework, we offer a generalisation error bound that encapsulates key ingredients associated with the learning problem such as the complexity of the data space, the cardinality of the training set, and the Lipschitz properties of a deep neural network.

We then specialize our analysis to a specific class of model based regression problems, namely the inverse problems. These problems often come with well defined forward operators that map variables of interest to the observations. It is therefore natural to ask whether such knowledge of the forward operator can be exploited in deep learning approaches increasingly used to solve inverse problems. We offer a generalisation error bound that – apart from the other factors – depends on the Jacobian of the composition of the forward operator with the neural network.

Motivated by our analysis, we then propose a 'plug-and-play' regulariser that leverages the knowledge of the forward map to improve the generalization of the network. We likewise also provide a method allowing us to tightly upper bound the norms of the Jacobians of the relevant operators that is much more computationally efficient than existing ones.

We demonstrate the efficacy of our model-aware regularised deep learning algorithms against other state-of-the-art approaches on inverse problems involving various sub-sampling operators such as those used in classical compressed sensing setup and inverse problems that are of interest in the biomedical imaging setup.

Impact Statement

The outstanding performance offered by deep neural networks to longstanding problems has encouraged its use in a myriad of applications. However, deep neural networks – often called *black boxes* – are poorly understood, leading to predictions that are often not interpretable or explainable. While in certain application fields this issue may play a secondary role, in high-risk domains, e.g., healthcare, it is crucial to use machine learning models that are trustworthy.

In this work, we take a step in this direction by providing a framework for explaining the various factors that affect the performance of a deep neural network on a well-known class of problems: inverse problems. This is an important class of problems that arises in various scientific and engineering applications including imaging techniques widely used in healthcare. We offer a principled methodology offering the means to train more robust deep neural network models.

Our results will have implications both in theory and practice. In particular, our theoretical results offer insights as to how the forward operator of a given inverse problem can be exploited to guide the training procedure. Our proposed regularizers, which ensure generalization in the presence of small datasets, are critical in scenarios where access to large supervised training sets is hard or monetarily infeasible, as for example in medical imaging. Our proposed techniques augment the current model aware data-driven techniques for inverse problems and will lead to wider adoption of deep neural networks for such problems.

Acknowledgements

I am indebted to many. I would like to start by extending my gratitude to the Commonwealth Scholarship Commission in UK for sponsoring my education at the University College London.

My supervisor, Professor Miguel Rodrigues mentored me and at the same time provided me complete freedom to choose the research directions I wanted to pursue – and for that I am extremely grateful. I appreciate his patience and continued encouragement – specially during the stressful periods of my PhD. I could not have asked for a better advisor than him.

I would also like to thank my colleague Zhaoyan for introducing me to countless tricks in programming and endless discussions on a variety of subjects. Parts of Chapter 5 are the result of joint work with him. I would also like to thank my other peers Pingfan, Zahra, Chao, Afroditi, Wei and Martin for their friendship and kindness.

My deepest gratitude goes to my family in Lahore – my parents, and my sisters – for their unconditional love and support every step of the way. I would like to specially mention my sister Zoha for her support; for countless calls; for her concern and for an unshakeable belief in me. I would like to thank my husband, Nauman, for being my partner in this journey and for being a gem of a father to our son – this thesis could never have been done without him. Finally yet most importantly, I would like to thank my son, Hussain for being extremely patient when he had to spend evenings and weekends away from me while I was working. Your love and light has kept me going even in the most trying of times.

Contents

Nomenclature

Symbols

$\mathbb{R}$	Real Numbers
$\mathbf{A}$	Forward Operator
$\mathbf{J}$	Jacobian Matrix
$\mathbf{W}$	Weight Matrix
$\mathcal{D}$	Sample Space $\mathcal{Y} \times \mathcal{X}$
$\mathcal{F}$	Hypothesis Space
$n_{\mathcal{X}}$	Covering number of $\mathcal{X}$
$\mathcal{S}$	Training Set
$\mathcal{X}$	Output Space
$\mathcal{Y}$	Input Space
D_{KL}	Kullback–Leibler divergence
$f_{\mathcal{S}}$	Learning algorithm training on $\mathcal{S}$
p	Dimension of the vectors in $\mathcal{X}$
q	Dimension of the vectors in $\mathcal{Y}$
m	Training set size

Greek Symbols

δ	Covering ball diameter of $\mathcal{X}$
ϵ	Maximum absolute difference between loss in a partition
η	Noise Level
λ	Regularization coefficient
Λ_a	Lipschitz constant of the operator $\mathbf{A}$
Λ_f	Lipschitz constant of the neural network f
$\Lambda_{f\circ a}$	Lipschitz constant of the composite mappinf $f \circ \mathbf{A}$
ψ	Covering ball diameter of $\mathcal{D}$
ρ	Product metric

Acronyms

BPDN	Basis Persuit Denoising
CNN	Convolutional Neural Network
CT	Computed Tomography
DL	Deep Learning
DNN	Deep Neural Network
GE	Generalization Error
IID	Independently and Identically Distributed
MRI	Magnetic Resonance Imaging
PSNR	Peak Signal to Noise Ratio
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity Index Measure

Notation

We use the following notation: matrices, column vectors, scalars and sets are denoted by boldface upper-case letters ($\mathbf{X}$), boldface lower-case letters ($\mathbf{x}$), italic letters (x) and calligraphic upper-case letters ($\mathcal{X}$), respectively. The i-th element of the vector $\mathbf{x}$ is denoted by x_i, and the element of the i-th row and j-th column of the matrix $\mathbf{X}$ is denoted by $(\mathbf{X})_{ij}$. The ℓ_p norm of vector $\mathbf{x}$ is represented by $||\mathbf{x}||_p$ and is given by $\left(\sum_i x_i^p\right)^{1/p}$ for $p \geq 1$. The Frobenious norm of the matrix $\mathbf{X}$ is denoted by $||\mathbf{X}||_F$. The covering number of $\mathcal{X}$ with ℓ_2-metric balls of radius $\delta/2$ is denoted by $n_{\mathcal{X}}(\delta/2, \ell_2)$.

List of Figures

List of Tables

Chapter 1

Introduction

Mathematical modeling of the physics, underlying the naturally occurring phenomenon is often motivated by our desire to understand and predict the behaviour of these systems. An accurate mathematical model allows us to reliably determine the measurements resulting from the given causal parameters. These problems – known as the forward or direct problem – are usually well defined and obtaining a solution is straightforward. In contrast, a number of problems involve inferring – from a set of indirect observations – the unknown physical quantities. For instance, constructing an image from CT measurements or removing noise from a corrupted waveform [7]. These problems are typically known as *inverse problems* since they start from the effect and work towards the unknown cause. Inverse problems form an important branch of mathematical inference tasks and arise in a number of important applications. As opposed to the direct problem, an inverse problem is often ill-posed and therefore may not have a unique solution in the absence of any suitable priors [8]. A comparison of these phenomenons for an image processing task is shown in Fig. 1.1. The left side of the figure describes a forward problem where a degradation model is applied to a clean image resulting in a noisy output. The right side of the image aims to recover the ground truth image using a reconstruction algorithm. This is an ill-posed problem since a solution – if it exists – may not be unique.

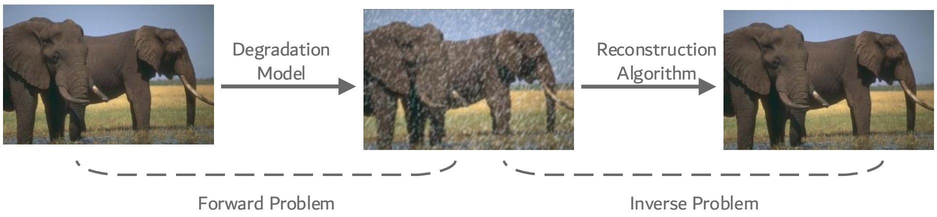


Figure 1.1: Forward and inverse problem in image restoration problem.

Classical signal processing algorithms used to solve such problems – such as approximate inversion [9], iterative algorithms [10] and other variational methods [11] – rely on simple, hand-designed algorithms that incorporate some form of domain knowledge. These knowledge-based algorithms, carry out inference based on prior information of the data manifold or the underlying model relating the signal of interest to the observed data. The success of these methods relies on simplifying assumptions and hand crafted priors that make them tractable and comprehensible. On the flip side, these approaches are not robust to the inaccuracies in knowledge of underlying physics and completely fail in the absence of a suitable prior [12].

Over the last decade, deep learning (DL) has been responsible for the tremendous empirical advances in many applications – ranging from computer vision [13] to natural language processing [14] to game play [15] – positioning itself as a major engineering discipline and prompting a general data-driven mindset. In view of their unprecedented success, these learned models have been used to solve family of problems that fall under the umbrella of inverse problems. However, in most of the preliminary works deep learning models were used as black boxes where simple principled techniques were replaced with purely data-led pipelines, trained end-to-end over huge annotated datasets [16]. These model-agnostic approaches are often prohibitive in scenarios where we lack massive labelled datasets, or where the underlying model is subject to variations requiring the deep learning model to be periodically re-trained. Moreover, these models are often considered black-boxes and therefore do not offer the interpretability afforded by the model based methods.

The limitations associated with the use of either knowledge based techniques or data-driven methods in isolation have given rise to hybrid knowledge-based data-driven approaches for inverse problems [12]. These techniques combine traditional signal processing with deep learning to give task specific solutions. These domain-aware deep learning methods have demonstrated clear advantage over model-agnostic approaches in terms of reconstruction quality and the need of large supervised datasets [4, 17]. Moreover, these methods are usually backed by theory that justifies the various design choices responsible for the impressive results. However, none of these techniques provides a concrete theoretical basis that explains the ability of such highly parameterized deep learning architectures to adapt to unseen inputs. Thus, the generalization behaviour of a learning algorithm on inverse problems is still not clearly understood.

This work – which aims to fill-in this gap – attempts to resolve two overarching questions:

- *How can we characterize the generalization performance of deep learning approaches for solving model based inverse problems?*

- *How can we incorporate domain knowledge in a data-driven approach such that generalization is guaranteed?*

Now that we have described the central themes of this work, it is worthwhile to take some time and shed light on the factors that stimulated our thought process.

1.1 Yet Another Generalization Bound?

The scientific community has witnessed a paradigm shift over the past decade, driven in large by the massive progress in the field of deep learning and artificial intelligence. Theoretical researchers have been trying to catch up to the empirical success of these models by providing mathematical characterizations of the various factors that enable this tremendous performance. One aspect of these parameteric functions that has attracted significant attention from the researchers is their prodigious ability to generalize to unseen data points. Since the knowledge of underlying data distribution is missing, estimating the exact generalization error is not feasible. This has led machine learning theoreticians to derive upper bounds on the generalization error of deep learning algorithms [18, 19, 20, 21]. Although, these works have started to add important layers to the quest of a unified generalization theory, a profound understanding of the generalization properties of these models is still missing. In our view, one of the challenges in understanding the generalization properties of a deep learning classifier is the absence of a postulated mathematical formulation or an underlying mapping of the data generation model. But, what if we consider a problem for which a data model is present? Would we be able to achieve more meaningful bounds then we have in the past? To this aim, we study the generalization error bounds for deep learning models in the context of inverse problems – a class of problems for which the underlying theoretical framework is already well-developed. Building upon our analysis, we propose a training strategy that penalizes key quantities associated with the forward map and the neural network. To interpret the efficacy of our technique, we further compare it to the the state of art methods present in the classical inverse problem theory literature.

1.2 Contributions

We have, so far stated in very broad terms, our intentions of studying the generalization behaviour of DNNs for inverse problems. In particular, in pursuit of this goal, we build upon the robustness framework introduced by Xu and Mannor in [22] to achieve the following objectives:

1. We derive generalization error guarantees for feedforward deep neural

networks, applicable to various tasks under the most popular loss functions: (1) classification tasks under the log-loss and (2) regression tasks under the ℓ_p-loss (Chapter 3).

2. Building upon our proposed generalization error guarantees, we design new regularizers allowing to learn deep neural network based data representations applicable to regression and classification problem. Such regularizers explicitly constrain the spectral norm of the input-output Jacobian matrix of the network. We also empirically demonstrate that our bounds – which, in contrast with existing ones, alleviate the exponential dependence of generalization error on network depth – lead to a regularization strategy offering superior generalization results in comparison with existing regularization strategies enforcing Lipschitz continuity (Chapter 5).

3. We present generalization error bounds for DNN based inverse problem solvers. Notably, such bounds depend on various quantities including the Jacobian matrix of the neural network along with the Jacobian matrix of the composition of the neural network with the inverse problem forward map. We also show how our bounds compare with the ones present in the classical literature, notably BPDN [23], for solving such sparse approximation problems (Chapter 4).

4. We also propose new regularization strategies that stem from our bounds and are capable of using knowledge about the inverse problem model during the neural network learning process via the control of the spectral and Frobenius norms of such Jacobian matrices. We also showcase computationally efficient methods to estimate the spectral and Frobenius norms of the aforementioned Jacobian matrices in order to accelerate the neural network learning process. We demonstrate the empirical performance of our algorithms on various inverse problems. These include the reconstruction of high-dimensional data from

low-dimensional noisy measurements where the forward model is a compressive random Gaussian matrix (Chapter 5).

5. We present a case study on the use of our model aware regularizers in the data-driven reconstruction of the popular imaging problems – the Computed Tomography (CT) and the accelerated Magnetic Resonance Imaging (MRI) which rely heavily on the theory of inverse problems (Chapter 6).

6. Finally we provide a brief preliminary generalization error analysis for inverse problems where we encounter uncertainties in the knowledge of the underlying model or the data distribution. Following our theoretical analysis, we discuss a roadmap and various directions that one can proceed in for further investigation (Chapter 7).

These contributions have led to the following manuscripts during the course of my PhD.

1. J. Amjad, Z. Lyu, and M. R.D. Rodrigues, "Deep Learning Model-Aware Regularization With Applications to Inverse Problems." IEEE Transactions on Signal Processing 69 (2021): 6371-6385.

2. J. Amjad, Z, Lyu, M. R.D. Rodrigues, "Regression with Deep Neural Networks: Generalization Error Guarantees, Learning Algorithms, and Regularizers." In 2021 29th European Signal Processing Conference (EUSIPCO) (pp. 1481-1485). IEEE.

3. J. Amjad, J. Sokolić, and M. R.D. Rodrigues. "On deep learning for inverse problems." In 2018 26th European Signal Processing Conference (EUSIPCO), pp. 1895-1899. IEEE, 2018.

1.3 Organization

This thesis is organized as follows:

In Chapter 2, we provide preliminary background that facilitates our theoretical analysis in the later part of the thesis. In particular, after introducing the supervised learning problem and the key ingredients of a deep learning algorithm, we briefly survey various theoretical results in the literature that quantify the performance of deep neural networks.

In Chapter 3, we derive new generalization error bounds for classification and regression settings. Our theoretical analyses show that for a neural network to be robust to the perturbations in the input and generalize well, the norm of its input-output gradient – the network Jacobian matrix – should be small.

We extend our generalization error analysis to robust deep learning architectures applicable to typical linear inverse problems in Chapter 4. Apart from the sample space and the Lipschitz constant of the neural network, our bounds depend on the Lipschitz constant of the product of the forward map and the neural network which can be tightly bounded by the product of Jacobian matrix of the forward map and the neural network.

The bounds in Chapter 3 and 4 naturally lead to a training strategy where norms of the Jacobians of the relevant mappings should be explicitly penalized. To this end, in Chapter 5, we propose to optimize a joint loss function where the empirical risk is augmented with a regularization term containing the norms of the Jacobians. We empirically demonstrate the effectiveness of our regularization technique by conducting extensive experimental studies on various classification, regression and inverse problems.

In Chapter 6, we test the performance of the proposed learning techniques on inverse problems that are of practical importance in clinical settings such as Magnetic Resonance Imaging and Computed Tomography. For these setting, we sample our ground truth data from publicly available databases and compare the performance of our novel model aware regularization strategies to both data-driven and knowledge based reconstruction algorithms present in literature.

Finally, in Chapter 7, we discuss some exciting future research direction stemming from our work and present concluding remarks.

Chapter 2

Deep Learning & Generalization

In this chapter, we first formally define the supervised learning problem and the various components of a DL algorithm. We then discuss generalization error (*GE*) and briefly review various theoretical bounds in the literature.

2.1 The Supervised Learning Problem

In the supervised learning problem, a set of labelled examples, $\mathcal{S} = \{(\mathbf{y}_i, \mathbf{x}_i) : i = 1, \ldots, m\}$, is observed by an agent and the task is to produce a rule (also known as hypothesis) that models the relationship from the input set $\mathcal{Y}$ to the outputs set $\mathcal{X}$. The quality of the chosen hypothesis can be assessed by a cost function. A successfully trained network is expected to perform well not only on the samples from the training set $\mathcal{S}$ but also on the separate test set that was not used to obtain the mapping. This is known as *generalization*. For the network to make accurate predictions on the test inputs, it is imperative that the training set be representative of the future unseen inputs. Therefore, it is assumed that all the training and test samples are drawn IID from some underlying probability distribution μ on the sample space $\mathcal{D} = \mathcal{Y} \times \mathcal{X}$. The distribution, however is unknown to the network and can only be approximated through the training examples.

We will now discuss the learning framework of a DL algorithm and various design choices that impact its performance.

2.2 Deep Learning

The three key components of a DL algorithm are the (i) architecture – specification of the number of neurons/layers in the network and their arrangements; (ii) objective function – quantification of how well the goal of the learning system is being achieved; and (iii) optimization algorithm – description of the set of rules for parameter update.

2.2.1 Architecture

The architecture of a DNN depends upon the particular needs of the task under consideration. A multitude of DNN designs have been proposed in literature for different application areas. For example, Convolution Neural Networks (CNN) are most suitable for image processing applications [24]. On the other hand, problems dealing with sequences of data can benefit more from Recurrent Neural Networks [25]. Nevertheless, regardless of its particular topology, a DNN can be represented by a directed graph whose nodes represent neurons and edges denote the weighted links between these neurons. A neuron can be understood as a processor that performs a simple function on its inputs. A set of neurons performing similar functions at a given time can be grouped together to form a layer. Each layer sequentially processes the output of the previous layer and forwards it to the next as depicted in Figure 2.1.

Formally, a d-layer DNN $f_{\mathcal{S}}(\mathbf{y};\Theta)$ – a nonlinear function that exploits the training set $\mathcal{S}$ to produce an estimate of its input $\mathbf{y} \in \mathbb{R}^q$ – can mathematically be expressed as the following sequential function[1]:

$$f_{\mathcal{S}}(\mathbf{y};\Theta) = \left(f_{\boldsymbol{\theta}_d} \circ \ldots\ldots f_{\boldsymbol{\theta}_1}\right)(\mathbf{y};\Theta) \tag{2.1}$$

where $f_{\boldsymbol{\theta}_i}, i \in \{1,2\ldots,d\}$ represents the non-linear function implemented by the i-th layer of the DNN and $\Theta = \{\boldsymbol{\theta}_i : i = 1,\ldots,d\}$ symbolizes the set of

[1] In the rest of the report, we omit Θ and denote a deep learning algorithm as $f_{\mathcal{S}}(\mathbf{y})$ for the sake of brevity.

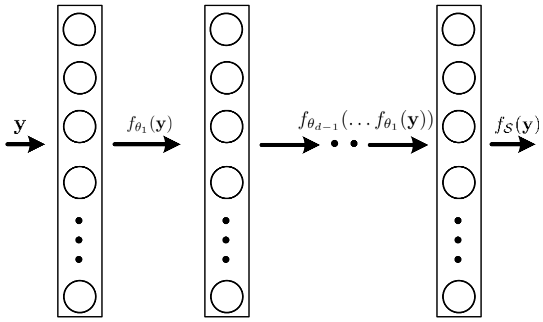

$\mathbf{y}$
$f_{\theta_1}(\mathbf{y})$
$f_{\theta_{d-1}}(\cdots f_{\theta_1}(\mathbf{y}))$
$f_{\mathcal{S}}(\mathbf{y})$

Figure 2.1: A schematic of a d-layer Deep Neural Network.

tunable parameters $\boldsymbol{\theta}_i$, learned by the DNN during the training mode. $f_{\boldsymbol{\theta}_i}$ is usually composed of the sum of the input to the i-th layer weighted by the layer wise weights $\mathbf{W}_i$ and offset by the bias vector $\mathbf{b}_i$. This is followed by a non-linear activation function. Common activation function include rectified linear unit (ReLU), sigmoid, hyperbolic tangent (tanh) and softmax function [26].

2.2.2 Objective Function

Training a DNN using a set $\mathcal{S}$ of examples, involves searching the parameter space for $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d\}$ that results in the closeness of the DNN output $f_{\mathcal{S}}(\mathbf{y}, \Theta)$ and the ultimate target value $\mathbf{x}$ that it wants to mimic.

The degree of 'closeness' between the network's output and the ground truth is mathematically quantified by an objective function $l(f_{\mathcal{S}}(\mathbf{y}), \mathbf{x})$. The objective function – also termed as cost function or loss function – measures how well the network $f_{\mathcal{S}}(\mathbf{y}, \Theta)$ is achieving its goals. Therefore training a DNN can be thought of as an exercise in function minimization where the network adjusts the values of its parameters to obtain a minima of the loss function. The speed with which a DNN learns a function is deeply influenced by the shape of the surface formed by the error measure in parameter space. Therefore, defining a loss function is an important part of the learning problem and different applications require appropriate loss functions to quantify the discrepancy in the DNN output. However, these cost functions can be specified without making reference to a specific problem or dataset. For example cross-entropy is the most commonly used objective function for gauging the performance of a classifier. Similarly, the pixel wise mean squared error is the preferred cost function of choice in imaging tasks where the output takes continuous values. Some of the most commonly used loss functions are ℓ_r-loss, logistic loss, hinge loss, cross-entropy loss and $0-1$ loss.

2.2.2.1 Regularization

Regularization, simply put, is the process of biasing the learning model to prevent overfitting. This is done by introducing additional information. It can be done by adding a regularization term to the loss function; or by imposing different training constraints. However, there is no one-fits-all technique. A regularization strategy that minimizes the generalization error for one task may not prove fruitful for the other. Below we list some of the most effective regularization techniques in use by the scientific community:

- *Parameter Penalties*: This regularization strategy is used when it is preferred to limit the capacity of a DNN [27]. It is achieved by minimizing the following regularized loss function instead of the simple empirical loss (2.6)

$$\mathcal{L}(f_{\mathcal{S}},\Theta)=\frac{1}{m}\sum_{i} l(f_{\mathcal{S}}(\mathbf{y}_i),\mathbf{x}_i)+\lambda R(\Theta) \tag{2.2}$$

  here $\frac{1}{m}\sum_i l(f_{\mathcal{S}}(\mathbf{y}_i),\mathbf{x}_i)$ is the unregularized empirical loss computed over the training set of cardinality m and $\lambda=[0,\infty)$ is the hyperparameter that controls the effect of the regularizer $R(\Theta)$. The greater the value of λ, the higher the penalty to the objective function. Now, the regularizer term $R(\Theta)$ represents the quantity that we wish to control. In some applications, it is preferable to have small ℓ_r norm of the weight matrices $\mathbf{W}_i$, resulting in the following regularization function

$$R(\Theta)=\sum_{i} ||\mathbf{W}_i||_r \tag{2.3}$$

where r can take integer values in $[1, \infty)$. Regularization by inducing an ℓ_2 norm penalty is also known as spectral norm regularization [28, 29, 30]. Frobenious norm punishment of the weight matrices is commonly referred to as weight decay since it drives the weight magnitudes towards zero [31]. On the other hand , ℓ_1 norm penalty encourages weight vectors that are sparse [32].

As opposed to reducing the norms of the linear affine mapping $\mathbf{W}$ for each layer towards zero, it is sometimes desirable to orthogonalize their rows (or columns) [33, 34, 35]. This kind of regularization enforces the weight matrices to lie on the Stiefel manifold and can be achieved by formulating a generalized Lagrange function of the following form [26]:

$$R(\Theta) = \sum_i ||\mathbf{W}_i^T \mathbf{W}_i - \mathbf{I}||_r \tag{2.4}$$

here $\mathbf{I}$ denotes the identity matrix of the appropriate dimension.

- *Gradient Penalties*: Weight based regularization techniques are often incorporated in the training procedure to enforce Lipschitz continuity in the prediction function. However, DNNs are nonlinear functions and regularizing only the linear layers while ignoring the non-linear ones results in an under utilization of the Lipschitz capacity [36, 35]. Moreover, in many neural network architectures it is not possible to easily regularize the affine mappings [37].

These problems can be avoided by penalizing the norm of the input-output Jacobian matrix of the network. The Jacobian matrix $\mathbf{J} \in \mathbb{R}^{p \times q}$, of a vector valued function $f_{\mathcal{S}}(.)$, is the matrix containing first-order partial derivatives of the output $f_{\mathcal{S}}(\mathbf{y}) \in \mathbb{R}^p$ with respect to the input

$\mathbf{y} \in \mathbb{R}^q$ and is given by:

$$\mathbf{J}(\mathbf{y}) = \begin{bmatrix} \frac{\partial f_{\mathcal{S}}(\mathbf{y})_1}{\partial \mathbf{y}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{y})_1}{\partial \mathbf{y}_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{\mathcal{S}}(\mathbf{y})_p}{\partial \mathbf{y}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{y})_p}{\partial \mathbf{y}_q} \end{bmatrix} \tag{2.5}$$

Jacobian regularization offers a model-agnostic technique to impose smoothness in the neural network and has been shown to result in increased classification margins [38] and improved robustness to white and adversarial noise in deep classifiers [39].

- *Dropout*: Dropout refers to removing certain nodes in a network during training phase to prevent overfitting. In particular, regularization by this method removes individual nodes from the DNN with a probability $(1-\tau), 0 \leq \tau \leq 1$ [40]. Intuitively, this technique can be interpreted as randomly producing a sparse network. It has been observed that DNNs with dropout are able to learn robust features [24].

- *Data Augmentation*: DNNs are tasked to learn complex functions from a limited number of examples therefore it is sometimes considered good practice to augment the training set by including different transformations of the examples already present in the set [41].

- *Early Stopping*: DNNs often exhibit the phenomenon of over-fitting i.e., although the training error is low, similar behaviour is not replicated on the unseen test examples. Early stopping provides a mechanism to prevents over-fitting by splitting the training examples into training and validation set. The network is trained on the training set and the validation set is used to anticipate the behaviour of learned parameters on the test set. Training is stopped as soon as the performance starts getting worse on the validation set [42].

- *Batch Normalization*: Neural networks are usually trained using mini-batch gradient descent. This can sometimes result in the change in distribution of the input to each layer at each iteration – a phenomenon known as *internal covariate shift* – causing the neural network to chase a moving target. Batch normalization can circumvent this issue by standardizing the inputs to a layer at each time step [43]. This leads to a stable and dramatically accelerated learning process of the neural network.

- *Weight Normalization*: Inspired by the batch normalization techniques, weight normalization reparameterizes the weight of each layer by decouplig their magnitude and direction [44]. This has been shown to condition the learning problem and reduce the number epochs required train the network.

2.2.3 Optimization Algorithm

The problem of training a DNN can be described as a credit assignment problem. Specifically, it searches for optimum values of network parameters that result in a minimum empirical loss. The non-linear nature of a DNN results in a loss function surface which is non-convex. Therefore instead of using the algorithms that provide global convergence guarantees, deep networks are trained using iterative gradient based methods that drive the empirical loss to a minimum in a gradual manner.

We will now briefly describe the basic Gradient Descent (GD) method for loss minimization. This method operates in two stages – forward propagation and backward propagation.

During the forward propagation phase of GD, DNN takes a sample from the training set, propagates it through its various layers and computes $f_{\mathcal{S}}(.,\Theta)$ for all members of the training set $\mathcal{S}$. The output of the network is the function of computation rule used at different layers and the initial values of parameters Θ. Several parameter initialization techniques have been proposed in literature for favourable training performance [26, 45, 46].

Once the outputs for all the samples in the training set have been computed, the average training error is calculated:

$$l_{\mathrm{emp}}(f_{\mathcal{S}}) = \frac{1}{m}\sum_{i} l(f_{\mathcal{S}}(\mathbf{y}_i),\mathbf{x}_i) \tag{2.6}$$

The information conveyed by this loss function is then propagated back (via backpropagation algorithm) to adjust the parameters of the network. Backpropagation is a method for computing, for every weight, the gradient of the error at the current setting of all the weights and is the most successful learning procedure and is at the heart of recent successes of machine learning, including state-of-the-art computer vision, sequence models and natural language processing.

Specifically, the backpropagation procedure computes the gradient of loss with respect to individual parameters in the multilayer DNN architecture

$$\frac{\partial l_{\mathrm{emp}}}{\partial \boldsymbol{\theta}_i}, \qquad \forall i \in \{1,\ldots,d\}$$

Once all the gradients are computed, the parameters are adjusted in proportion to the negative of these values:

$$\boldsymbol{\theta}^{(i)[l]} = \boldsymbol{\theta}^{(i)[l-1]} - \alpha\frac{\partial l_{\mathrm{emp}}}{\partial \boldsymbol{\theta}^{(i)}}, \qquad l = 1,\ldots,L$$

here the superscript $[l]$ indexes the GD iteration and the learning rate α represents the fixed step size that the GD algorithm takes towards the minima in each iteration. The recalibrated values of parameters are passed to next iteration of GD for the forward propagation. This process continues until l_{emp} plateaus or maximum number of iterations L is reached.

2.2.3.1 Improvements in GD

The standard batch GD, though the most widely used algorithm for training a DNN, has several pitfalls [47] – convergence to a local minima being the most important. Additionally, even if it learns parameters that result in desirable loss minimization over the training set, there is a possibility of poor generalization and over fitting. Several GD variants and optimization techniques have been proposed to circumvent these issues.

Mini-batch Stochastic Gradient Descent: It is similar to the standard GD algorithm in principle. The difference lies in the number of samples used for training [48]. Specifically, in Stochastic Gradient Descent (SGD), a set of $m' \ll m$ members is constructed by randomly selecting samples from the training set $\mathcal{S}$. This mini-batch of samples is then iteratively used to minimize the loss function. Repeating this process m/m' times by selecting m' samples randomly at each turn is referred to as running a single epoch of training.

Training a DNN in this manner, in practice, results in faster convergence and also helps prevent over fitting.

Other Variants: Though SGD results in improved training performance, there are still instances when the DNN gets trapped in the local minima or saddle points. This results in a painfully slow convergence rate. This problem can be avoided by adapting the learning rate at each iteration of the SGD. There are several ways of achieving this. For example in SGD with momentum, a fraction of the gradient step in the previous iteration is added to the update vector of the current iteration [49]. This prevents the gradients from diminishing. Other methods for adapting the learning rate include Adagard [50], Adam [51], Nadam [52], Adadelta [53] and AMSGrad [13].

2.3 Generalization Error

Most prominent DNN architectures in use today have number of neurons that far exceed the samples in the training set. Despite the formidable capacities represented by these hypothesis spaces, efficient algorithms such as SGD are able to achieve arbitrarily small amount of empirical loss on a myriad of tasks. However, the goal of training a DNN is not only to minimize empirical loss but for it to learn a mapping that generalizes well on the previously unobserved inputs. This ability of a network to perform well on the new, unseen inputs is quantified by the generalization error:

$$GE(f_{\mathcal{S}}) = |l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}})| \tag{2.7}$$

corresponding to the difference between the expected and empirical losses given by:

$$l_{\text{exp}}(f_{\mathcal{S}}) = \mathbb{E}_{(\mathbf{y},\mathbf{x})\sim\mu}[l(f_{\mathcal{S}}(\mathbf{y}),\mathbf{x})], \qquad l_{\text{emp}}(f_{\mathcal{S}}) = \frac{1}{m}\sum_{i} l(f_{\mathcal{S}}(\mathbf{y}_i),\mathbf{x}_i) \tag{2.8}$$

where $l(\cdot,\cdot)$ is an appropriate pre-specified loss function that depends on the specific task, $l_{\text{emp}}(f_{\mathcal{S}})$ is the empirical loss computed over the training set $\mathcal{S}$ and l_{exp} is the expected value of loss incurred by the network which is calculated by taking expectation over all the possible samples $(\mathbf{y},\mathbf{x})$, drawn according to the probability distribution μ, that the network might encounter. Since μ is not known, it is not possible to estimate GE for a particular DNN. A number of empirical works in literature have attempted to find a complexity measure of the network or loss landscape that can predict generalization [54, 55, 56]. However, a more principled approach to study generalization is to compute an upper bound on the expected error or the generalization gap in (2.7). Various theoretically motivated complexity measures measures have been proposed in literature to bound the GE [57, 58, 59, 21]. Most of these results, however, consider DNNs only in the context of classification.

Next, we briefly discuss some of the notable bounds present in literature.

2.3.1 Vapnik-Chervonenkis (VC) - Dimension

VC-Dimension provides a capacity based performance characterization of a hypothesis class that can be learned via a binary classification algorithm.

Definition 2.1. ([60]) Consider a hypothesis class $\mathcal{F}$ from the set $\mathcal{Y}$ to $\{\pm 1\}$. Now let, $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2 \ldots \mathbf{u}_m\} \subset \mathcal{Y}$. Then the shattering coefficient, $sh(\mathcal{F}, m)$ of class $\mathcal{F}$ is the maximum number of functions that can be derived from $\mathcal{U}$ to $\{\pm 1\}$. The Vapnik-Chervonenkis (VC) -dimension of the hypothesis class $\mathcal{F}$ is then defined as

$$VC(\mathcal{F}) = \sup\{m : sh(\mathcal{F}, m) = 2^m\} \tag{2.9}$$

A class $\mathcal{F}$ is said to have infinite VC-dimension if $sh(\mathcal{F}, m) = 2^m$ for an arbitrarily large value of m.

The VC-Dimension of a hypothesis class $\mathcal{F}$ can be understood as the maximum cardinality m of a training set $\mathcal{S}$ for which, some hypothesis from a class of binary classifiers $\mathcal{F}$ achieves zero training error. The following bound on the GE of $\mathcal{F}$ then holds with probability $1 - \zeta$

$$l_{exp} - l_{emp} \leq o\sqrt{\frac{VC(\mathcal{F})}{m}} \tag{2.10}$$

An upper bound on the VC-dimension of FC feed-forward DNNs was recently derived in [61].

Theorem 2.1. Consider a d-layer fully connected feed forward DNN f, that has t parameters in each layer and has ReLU non-linearities. Let $\mathcal{F}$ denote the class of all such DNNs. Then the VC-dimension of $\mathcal{F}$ is

$$VC(\mathcal{F}) = o\left(d^2 t^2\right) \tag{2.11}$$

As a consequence of Theorem 2.1 and (2.10), the GE of f is upper bounded by:

$$GE(f) \leq o\sqrt{\frac{d^2 f^2}{m}} \tag{2.12}$$

The GE bound in (2.12) is not particularly useful for dealing with over-parametrized learning algorithms such as DNNs on two accounts; 1) The bound becomes very loose for large number of parameters, 2) This bound does not explain the empirically observed behaviour of DNNs that tend to generalize well with the increase in the number of parameters [62].

2.3.2 Rademacher Complexity

Unlike *VC*-dimension, which only takes into account the binary hypotheses classes, Rademacher Complexity is a data-dependant statistic that measures the richness of real-valued functions [58].

Definition 2.2. Consider a space $\mathcal{Y}$. Let $\mathcal{F}$ denote a class of hypotheses $f : \mathcal{Y} \to \mathbb{R}$. Now, consider a set $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ drawn IID from the distribution μ_y over $\mathcal{Y}$. Then the empirical Rademacher complexity of $\mathcal{F}$ is,

$$\widehat{R}_m(\mathcal{F}, \mathcal{U}) = \mathbb{E}_{\mathbf{z}}\left[\sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^{m} z_i f(\mathbf{u}_i)\right)\right] \tag{2.13}$$

where $\mathbf{z} = \{z_i\}_{i \le m}$ is a set of independent random variables chosen uniformly from $\{-1, 1\}$. The Rademacher complexity of $\mathcal{F}$ is then defined as

$$R_m(\mathcal{F}) = \mathbb{E}_{\mathcal{U} \sim \mu_y}\left[\widehat{R}_m(\mathcal{F}, \mathcal{U})\right] \tag{2.14}$$

A high value of $R_m(\mathcal{F})$ for a binary classifier is reflective of its richness, namely, it has the ability to fit random noise.

The following *GE* bound holds with a probability $1 - \zeta$ for all $h \in \mathcal{F}$

$$l_{exp}(f) - l_{emp}(f) \le R_m(\mathcal{L}) + \sqrt{\frac{8 \log(2/\zeta)}{m}} \tag{2.15}$$

here $\mathcal{L} = \{(\mathbf{y}, x) \to l(f(\mathbf{y}), x)\}$ is a loss class defined for a fixed bounded loss function $l(f(\mathbf{y}), x) \in [0, 1]$.

This bound can be specialized for feed-forward deep neural networks. The authors of [59] compute a bound for the *GE* of a d-layer DNN which is independent of the number of parameters in the network.

Theorem 2.2. Consider a d-layer feed forward DNN $f(\mathbf{y}) = \sigma(\mathbf{W}_d \sigma(\ldots \sigma(\mathbf{W}_1 \mathbf{y})))$ that has ReLU as activations. Now suppose that the network parameters satisfy the following constraints

$$\prod_{i=1}^{d} ||\mathbf{W}_i||_F \leq \omega$$

Let $\mathcal{F}$, denote the class of all such DNNs. Then the $R_m(\mathcal{F})$ of a f in $\mathcal{F}$ is bounded by

$$R_m(\mathcal{F}) \leq o\left(\frac{2^d \omega}{\sqrt{m}}\right) \tag{2.16}$$

Though the bound in Theorem 2.2 is independent of the number of parameters of a DNN classifier, it still scales exponentially with the number of layers and thus doesn't explain the remarkable generalization properties of substantially deep neural nets.

2.3.3 Margin

The *margin* of a classifier can be understood as the confidence with which the predictions are made.

Definition 2.3. Consider a p-class predictor $f : \mathbb{R}^q \to \mathbb{R}^p$ that assigns real weights to its p labels. The label with the maximum magnitude is picked as the output of f. Then the margin is defined as the difference between the weight of the correct label and the maximum weight of any incorrect label

$$\gamma = f(\mathbf{y})_x - \max_{i \neq x} f(\mathbf{y})_i \qquad\qquad i,x \in \{1,\ldots,p\} \tag{2.17}$$

The notion of margin is considered important for studying the generalization behaviour of classifiers since it has been shown that an improvement in classification margin results in an improvement in the upper bound on the *GE* [63]. Efforts have been made to bound the *GE* of DNN classifiers using the margin. Neyshabur et. al recently showed the *GE* of a DNN can be controlled using the classification margin and the norm of DNN parameters [64]. The authors of [20] derived similar bounds independently using a covering number argument. Owing to the similarity of the two results, we will only include the *GE* bound in [64].

Theorem 2.3. Consider the input space $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^q : ||\mathbf{y}||_2 \leq \beta\}$. Now, let a d-layer classifier $f : \mathcal{Y} \rightarrow \mathbb{R}^p$ be defined as

$$f(\mathbf{y}) = \sigma(\mathbf{W}_d\sigma(\ldots(\sigma(\mathbf{W}_1\mathbf{y})))) \tag{2.18}$$

where σ denotes the ReLU function. Also, let γ be the maximum margin of f and t be the upper bound on the maximum number of parameters in any layer. Then for any $\gamma, \zeta > 0$ and a training set of size m, the following holds with probability $1 - \zeta$,

$$l_{exp}(f) \leq l_{emp,\gamma} + \mathcal{O}\sqrt{\frac{\beta^2 d^2 t \ln(dt) \prod_{i=1}^{d} ||\mathbf{W}_i||_2^2 \sum_{i=1}^{d} \frac{||\mathbf{W}_i||_F^2}{||\mathbf{W}_i||_2^2} + \ln\frac{dm}{\zeta}}{\gamma^2 m}} \tag{2.19}$$

here $l_{emp,\gamma} \leq \frac{1}{m}\sum_i \mathbb{1}\left[f(\mathbf{y})_x \leq \gamma + \max_{x \neq i} f(\mathbf{y})_i\right]$.

2.3.4 Algorithmic Stability

The notion of Algorithmic stability gauges the sensitivity of the learning algorithm with respect to the changes in the training set [65].

Definition 2.4. Given a sample space $\mathcal{D}$ and a training set $\mathcal{S} \subseteq \mathcal{D}$

$$\mathcal{S} = \{(\mathbf{y}_1, \mathbf{x}_1), \ldots, (\mathbf{y}_{i-1}, \mathbf{x}_{i-1}), (\mathbf{y}_i, \mathbf{x}_i), (\mathbf{y}_{i+1}, \mathbf{x}_{i+1}), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$$

let the set $\mathcal{S}^{\backslash i}$ be defined as:

$$\mathcal{S}^{\backslash i} = \{(\mathbf{y}_1, \mathbf{x}_1), \ldots, (\mathbf{y}_{i-1}, \mathbf{x}_{i-1}), (\mathbf{y}_{i+1}, \mathbf{x}_{i+1}), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$$

Then a learning algorithm f is said to hold a uniform stability β if

$$\forall \mathcal{S} \in \mathcal{D}, \forall i \in \{1, \ldots, m\}, ||l(f_{\mathcal{S}}(\mathbf{y}), \mathbf{x}) - l(f_{\mathcal{S}^{\backslash i}}(\mathbf{y}), \mathbf{x})||_\infty \leq \beta \tag{2.20}$$

The algorithmic stability has been used to derive generalization error bounds of the following form:

Theorem 2.4. For a learning algorithm f with uniform stability β, the following holds with probability at least $1-\zeta, 0 \leq \zeta \leq 1$ over the random draw of the training set $\mathcal{S}$

$$l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}}) \leq 2\beta + (4m\beta + M)\sqrt{\frac{1/\zeta}{2m}}$$

for $0 \leq l(f(\mathbf{y}), \mathbf{x}) \leq M$ for all $(\mathbf{y}, \mathbf{x}) \in \mathcal{D}$ and all training sets $\mathcal{S} \in \mathcal{D}$.

2.3.5 Compression Based Approaches

Compression based generalization error bounds have been proposed in literature to provide a tighter evaluation of the generalization error of the neural networks. These bounds are based on the premise that the SGD implicitly learns simple models and therefore a trained model is effectively compressed. Several notable works in literature have attempted to evaluate the dimensionality of these implicitly compressed models via different measures such as the layer cushion [66], pruning [67] or spectrum based mechanisms [68] and have subsequently derived generalization error bounds.

For the sake of brevity, we include the generalization bound derived in [66].

Definition 2.5. Let $\mathcal{G}_{\mathbf{W},\mathbf{s}} = \{g_{\mathbf{W},s} | \mathbf{W} \in \mathcal{W}\}$ denote a class of classifiers parameterized by the weight matrices $\mathbf{W}$ and fixed helper strings s. Then a classifier f is said to be $(\gamma, \mathcal{S})$-compressible via $\mathcal{G}_{\mathbf{W},s}$ if there exists $\mathbf{W} \in \mathcal{W}$ such that for any $\mathbf{y} \in \mathcal{S}$, the following holds for all x,

$$|f(\mathbf{y})_x - g_{\mathbf{W},s}(\mathbf{y})_x| \leq \gamma$$

Theorem 2.5. Consider a class of classifiers $\mathcal{G}_{\mathbf{W},\mathbf{s}} = \{g_{\mathbf{W},s} | \mathbf{W} \in \mathcal{W}\}$ where $\mathbf{W}$, a set of u parameters each, can at the most have v discrete values. If a trained classifier f is $(\gamma, \mathcal{S})$-compressible with respect to the class of classifier $\mathcal{G}_{\mathbf{A},s}$, then with high probability, there exists $\mathbf{W} \in \mathcal{W}$ over the training set,

$$l_{\exp,\gamma} \leq l_{\text{emp},\gamma} + \mathcal{O}\sqrt{\frac{u \log v}{m}}$$

here $l_{\exp,\gamma} \leq \mathbb{E}_{(\mathbf{y},x)\sim\mu}\left[f(\mathbf{y})_x \leq \gamma + \max_{x\neq i} f(\mathbf{y})_i\right]$ is the expected margin loss.

Unfortunately, these guarantees are applicable to the compressed network $g_{\mathbf{W}}$ only and not the original classifier f. Therefore, it is unable to explain the generalization properties of over parameterized deep networks.

2.3.6 Algorithmic Robustness

The notion of robustness was introduced by Xu and Mannor as a property of a fixed learning algorithm rather than a hypothesis class [22]. It is a data-dependant statistic that measures the performance of a learning algorithm by exploring sceneries where losses associated with a training sample and a test sample are close.

Definition 2.6. Let $\mathcal{S}$ and $\mathcal{D}$ denote the training set and sample space. A learning algorithm is said to be $(K,\epsilon(\mathcal{S}))$-robust if the sample space $\mathcal{D}$ can be partitioned into K disjoint sets $\mathcal{K}_k$, $k = 1,\ldots,K$, such that for all $(\mathbf{y}_i,\mathbf{x}_i) \in \mathcal{S}$ and all $(\mathbf{y},\mathbf{x}) \in \mathcal{D}$

$$(\mathbf{y}_i,\mathbf{x}_i),(\mathbf{y},\mathbf{x}) \in \mathcal{K}_k \implies \left|l(f_{\mathcal{S}}(\mathbf{x}_i),\mathbf{y}_i) - l(f_{\mathcal{S}}(\mathbf{x}),\mathbf{y})\right| \leq \epsilon(\mathcal{S}) \tag{2.21}$$

An upper-bound on K can be obtained via the covering number of the sample space $\mathcal{D}$.

Definition 2.7. (*Covering Number* [69]) For every $\psi > 0$, the $\psi/2$-covering number $\mathcal{N}(\psi/2;\mathcal{D},\rho)$ for a metric space $(\mathcal{D},\rho)$ is defined as the minimum cardinality of set $\widehat{\mathcal{D}}$, such that every point in $\mathcal{D}$ is at a distance not more than ψ from some point in $\widehat{\mathcal{D}}$.

The algorithmic robustness framework has been used to derive generalization bounds for deep learning classifiers [38, 33, 30]. Sokolić et al.

utilized the robustness framework proposed in [22] to derive *GE* bounds for large-margin DNN classifiers.

Theorem 2.6. Consider a metric space $\mathcal{Y}$, compact with respect to the metric ρ_y. Now consider a large margin DNN classifier $f : \mathcal{Y} \to \{1,2,\dots,p\}$ that trains on a training set S. Then f is $(p.\mathcal{N}(\gamma/2;\mathcal{N},\rho_y),0)$ robust if, $\forall \mathbf{s} \in \mathcal{S}$ there exists a γ such that,

$$0 < \gamma < \gamma^{\rho_y} \tag{2.22}$$

The *GE* of the large-margin classifier f in Theorem 2.6 is then upper bounded by:

$$GE(f) \leq \frac{1}{\sqrt{m}} \sqrt{2\log(2).p.\mathcal{N}(\gamma/2;\mathcal{Y},\rho_y)} \tag{2.23}$$

Unlike Theorem 2.3, the upper bound in Theorem 2.6 is independent of network size and depth. However, their *GE*-bound is a function of the classification margin γ^{ρ_y} that can be bounded by means of the Frobinious norm of the weights [38].

2.4 Summary

DNNs are learning algorithms that have the ability to learn useful information from few examples and extrapolate their knowledge to unseen data. This chapter provided an overview of various factors that have an impact on the generalization properties of DNNs. We also reviewed some of the prominent theoretical guarantees from the literature. As noted in the chapter; most of the bounds present in literature are parameter dependent and may deteriorate exponentially with depth. In the next chapter, we tackle this issue and present *GE* bounds for neural networks that depend upon the input-output network Jacobian. Our bounds alleviate the issue faced by parameter dependence in the bounds.

Chapter 3

Generalization Error Bounds for Robust DL Algorithms

In this chapter, building upon the robustness and generalization framework developed in [22], we offer new characterizations of the generalization ability of deep neural networks, when used to learn functions applicable to either classification or regression tasks. In particular, we derive new generalization error guarantees applicable to such tasks under the most popular loss functions.

3.1 Prior Work

Our work connects to various other works in the literature. In particular, a number of papers have in recent years offered characterizations of the generalization ability of deep neural networks that have in turn inspired new regularization strategies [70]. For example, both [18] and [20] independently provided *GE* bounds for deep neural networks, expressed in terms of different norms associated with the collection of network parameters – such as group norm, max norm and spectral norm – thus inspiring new regularizers aiming explicitly at constraining the network complexity by limiting the value of such norms. In [66], the authors provide a classification framework to characterize the generalization properties of neural networks, leading to linear-algebraic algorithms to effectively limit the number of parameters in individual layers. In [33], the authors proposed an upper bound to the *GE* of a DNN based classifier in the presence of the adversarial perturbations. The authors have also proposed to optimize the network in a manner that forces the weight matrices to remain on the Stiefel manifold [71], by forcing the Gram matrix of the weight matrices to be closer to an identity matrix. However, these works have typically led to somewhat vacuous bounds depending on quantities like the product of network weight norms that cannot completely capture the fact that deeper networks generalize better than shallower ones [72]. In [38], the authors derive a *GE* bound for large margin DNN classifiers under a uniformly bounded $0-1$ loss that leads to a new Jacobian regularizer – involving penalizing the Frobenius norm of the network Jacobian – that also further boosts a deep neural network performance. Our work departs from these works, mainly because we consider unbounded loss functions in our analysis. Most learning tasks in realistic settings are optimized over unbounded loss function such as cross entropy loss function for classification and ℓ_2 loss function in regression problems.. It should however be noted that some of the regularizers originating from our analysis – that propose to penalize an upper bound on the network Lipschitz constant – also connect to some regularizers already proposed in the literature [38, 73].

The fact that enforcing Lipschitz regularity in deep neural networks endows them with several desirable properties is well recognized [28, 74, 75, 76, 33, 77, 73, 38, 35, 78, 30, 79, 39]. For example, a small Lipschitz constant has also been shown to result in better generalization error guarantees [18, 20]. There are various other papers that have in turn suggested a number of regularization strategies based on empirical considerations, showcasing that such regularization approaches can lead to better performance than conventional ones. In particular [74] propose to explicitly enforce an upper bound on the Lipschitz constant of neural networks – via the operator norm of the weight matrices – in order to improve its performance. Various other works [46, 80, 34, 33] have advocated limiting the spectral norm of the weight matrices. Motivated by the norm preservation offered by orthogonal weight matrices, [46] propose orthogonal weight initializations to accelerate the training speed of the neural networks. However, initialization alone does not guarantee orthogonality throughout the training process, hence the orthogonality properties of the fine tuned weight matrices may differ substantially from the original ones. Taking this approach a step further, [29] propose to initialize the weight matrices by the technique proposed by [46] but simultaneously manually clip the singular values of weight matrices in a narrow window around 1 during the training process in order to maintain orthogonality properties. However, in addition to being computationally expensive owing to the cost associated with the calculation of singular value decompositions (SVD), this method seems counter intuitive since the *new* weight matrix with the clipped singular values may not be close to the original updated weight matrix, possibly resulting in performance deterioration. In another work, [81] empirically shows that regularizing weight matrices to make them orthogonal results in improved performance for generative networks. Motivated by the benefits offered by orthogonal regularization, in [34] the authors present a regularization technique for convolution neural networks that forces the weight matrices of a convolutional layer to have a small restricted isometry constant [82]. Other works such as [83] and [84] present efficient algorithms to calculate the spectral norm of the linear transform associated with the convolutional layers. Weight orthogonalization approaches have also received ample attention in recurrent neural networks as well [85, 86].

However, many of the existing techniques constrain only the Lipschitz constants of the layer-wise affine transformations in the network [28, 74, 75, 76, 33]. These approaches do not take into account the non-linearities in the network and thus under-utilize the Lipschitz capacity of the network by biasing it to learn simplistic functions [35]. In this work, motivated by our analysis, we propose to constrain the spectral norm of the input-output network Jacobian matrix which serves as a tight upper bound on the network Lipschitz constant. We also offer an algorithm to efficiently estimate it without significantly increasing the computational overhead in Chapter 5. The estimation of true penalty of the Lipschitz constant has been shown to be computationally infeasible [76]. However, to the best of our knowledge, the method presented in this work is the tightest and most efficient manner to bound a deep neural network Lipschitz constant.

3.2 The Learning Problem

We are interested in problems involving the estimation of a functional relationship between data points $\mathbf{y} \in \mathcal{Y}$ and target points $\mathbf{x} \in \mathcal{X}$ based on a set of examples $\mathcal{S} = \{(\mathbf{y}_i, \mathbf{x}_i)\}_{i \leq m}$ drawn i.i.d. from the space $\mathcal{D} = \mathcal{Y} \times \mathcal{X}$ according to an unknown probability measure μ. These can cover both classification problems, where the target points are drawn from a finite set, and regression problems, where the target points are drawn from some non-finite set. We assume that $\mathcal{Y}$ and $\mathcal{X}$ are compact metric spaces with respect to some pre-specified metrics. We also assume that the space $\mathcal{D} = \mathcal{Y} \times \mathcal{X}$ is compact with respect to a product space metric ρ [87].

In this supervised learning setting, we restrict our attention to the use of deep neural networks $f_{\mathcal{S}} : \mathbb{R}^q \to \mathbb{R}^p$ that are trained on the training set $\mathcal{S}$ to learn the underlying map between $\mathcal{Y}$ and $\mathcal{X}$. Such a feed forward deep neural network can be represented as a composition of d layer-wise mappings delivering an output $f_{\mathcal{S}}(\mathbf{y}) \in \mathbb{R}^p$, given the input $\mathbf{y} \in \mathbb{R}^q$ via (2.1). Our goal is to characterize the quality of such learnt deep neural network using a well-known measure quantifying the generalization capability of a machine learning models. In particular, we will use the generalization error (GE) associated with the learnt deep neural network:

$$GE(f_{\mathcal{S}}) = |l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}})| \tag{3.1}$$

Our ensuing analysis offers bounds to the *GE* of deep neural networks used for classification or regression settings under different losses, as a function of a number of quantities associated with the learning problem. These quantities include the covering number of the sample space $\mathcal{D}$, the size of the training set $\mathcal{S}$, and properties of the network encapsulated in its Jacobian matrix given (2.5).

3.3 Generalization Error Analysis

We now develop bounds to the generalization error associated with deep neural networks by leveraging the *algorithmic robustness* framework in [22].

The algorithmic robustness framework has already been used to derive generalization bounds for deep learning architectures [38, 33, 77, 30]. We instead build upon this framework in order to understand how deep neural networks can underlie the construction of input-output functional relations for a wide range of tasks including: (1) classification tasks under categorical loss functions, (2) regression tasks under per pixel (ℓ_r) loss functions and (3) regression tasks under perceptual loss functions [88].

Our analysis builds upon a simple characterization of the Lipschitz continuity of a deep neural network, based on the use of ℓ_r and ℓ_t norms to measure distances on the network input and output respectively. It represents a generalization of a key result in [38].

Lemma 3.1. ([38]) Let $f_{\mathcal{S}}(\cdot)$ be a deep neural network trained on the training set $\mathcal{S}$. It follows for any $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}$ that

$$||f_{\mathcal{S}}(\mathbf{y}') - f_{\mathcal{S}}(\mathbf{y}'')||_r \leq \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} ||\mathbf{J}(\mathbf{y})||_{r,t} ||\mathbf{y}' - \mathbf{y}'||_t$$

where ℓ_r and ℓ_t are the metrics used to measure distances on the network input and output respectively, $\mathbf{J}$ is the input-output network Jacobian matrix and *conv*$(\cdot)$ is the convex hull of a set.

3.3.1 Classification problems

We first consider learning problems associated with multi-class classification tasks where the input data points $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^q$ represent multi-dimensional objects such as images and the output data points $x \in \mathcal{X} \subseteq \{1, 2, \ldots, p\}$ represent the classes associated with the different objects. We assume $\mathcal{Y}$ is compact and we use ℓ_t to measure distances in $\mathcal{Y}$.

We will characterize the performance of a deep neural network based classifier with respect to the standard cross-entropy with softmax loss (a.k.a. log-loss). In particular, given a deep neural network $f_{\mathcal{S}}(\cdot)$ trained on a training set $\mathcal{S}$, a data point $\mathbf{y} \in \mathcal{Y}$, and the data point class $x \in \mathcal{X}$, the log-loss quantifies the loss as follows:

$$l(f_{\mathcal{S}}(\cdot), (\mathbf{y}, x)) = -f_{\mathcal{S}}(\mathbf{y})_x + \mathsf{lse}(f_{\mathcal{S}}(\mathbf{y})) \tag{3.2}$$

where

$$\mathsf{lse}(f_{\mathcal{S}}(\mathbf{y})) = \log \sum_{j=1}^{q} \exp(f_{\mathcal{S}}(\mathbf{y})_j) \tag{3.3}$$

The following theorem – building upon Lemma 3.1 – characterizes the robustness of such a deep neural network classifier trained under log-loss.

Theorem 3.1. (Robustness) A DNN $f_{\mathcal{S}}(\cdot)$ trained on the training set $\mathcal{S}$ under the cross-entropy with softmax loss function in (3.2) is

$$\left(q\mathcal{N}_{\mathcal{Y}}(\gamma/2, \ell_t), 2 \sup_{\mathbf{y} \in \mathrm{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_{r,t}\gamma \right) - \text{robust}$$

for any $\gamma > 0$ and $\mathcal{N}_{\mathcal{Y}}(\gamma/2, \ell_t) < \infty$.

Proof. We first establish a simple result underlying our theorems that asserts that the log-sum exponent $\mathsf{lse}(.)$ in equation (3.3) is 1-Lipschitz continuous.

To prove this simple result, let us first note that $\mathsf{lse}(.)$ is a continuous function with the gradient i-th element $\nabla_i \mathsf{lse}(f(\mathbf{y}))$ corresponding to the softmax function. Then, we can use the intermediate value theorem to show that for some $f(\mathbf{y}'), f(\mathbf{y})$ and $f(\mathbf{y}'')$ such that $f(\mathbf{y}') \leq f(\mathbf{y}) \leq f(\mathbf{y}'')$ it holds

$$\begin{aligned}
|\mathsf{lse}(f(\mathbf{y}')) - \mathsf{lse}(f(\mathbf{y}''))| &= |\nabla \mathsf{lse}(f(\mathbf{y})) \cdot (f(\mathbf{y}') - f(\mathbf{y}''))| \\
&\overset{(a)}{\leq} \sum_{i=1}^{q} \nabla_i \mathsf{lse}(f(\mathbf{y})) \| f(\mathbf{y}') - f(\mathbf{y}'')\|_t \\
&= \| f(\mathbf{y}') - f(\mathbf{y}'')\|_t
\end{aligned} \tag{3.4}$$

(a) is due to the Hölder's inequality.

Let $\mathcal{N}_{\mathcal{Y}}(\gamma/2, \ell_t)$ be a $\gamma/2$ cover of $\mathcal{Y}$. We can then partition the space $\mathcal{D}$ onto $K \leq q\mathcal{N}_{\mathcal{Y}}(\gamma/2, \ell_t)$ partitions, such that if $(\mathbf{y}', x')$ and $(\mathbf{y}'', x'')$ belong to the same partition, then $x' = x''$ and $\|\mathbf{y}' - \mathbf{y}''\|_t \leq \gamma$.

Let us now consider two data points $(\mathbf{y}', x') \in \mathcal{S}$ and $(\mathbf{y}'', x') \in \mathcal{D}$ associated with the same partition, implying that $x' = x'' = x \in \mathcal{X}$. Then,

$$\begin{aligned}
|l(f_{\mathcal{S}}, (\mathbf{y}', x')) - l(f_{\mathcal{S}}, (\mathbf{y}'', x''))| &= \left| -f(\mathbf{y}')_x + \mathsf{lse}(f(\mathbf{y}')) + f(\mathbf{y}'')_x - \mathsf{lse}(f(\mathbf{y}''))\right| \\
&\overset{(a)}{\leq} \left| f(\mathbf{y}')_x - f(\mathbf{y}'')_x \right| + \left| \mathsf{lse}(f(\mathbf{y}')) - \mathsf{lse}(f(\mathbf{y}''))\right| \\
&\overset{(b)}{\leq} \left\| f(\mathbf{y}') - f(\mathbf{y}'')\right\|_t + \left\| f(\mathbf{y}') - f(\mathbf{y}'')\right\|_t \\
&\overset{(c)}{\leq} 2 \sup_{\mathbf{y} \in \mathrm{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_{r,t} \|\mathbf{y}' - \mathbf{y}''\|_r
\end{aligned}$$

where (a) follows from Minkowski's inequality, (b) follows from norm equivalence together with (3.4) and (c) is due to. Lemma 3.1.

It follows immediately that:

$$|l(f_{\mathcal{S}}, (\mathbf{y}', x)) - l(f_{\mathcal{S}}, (\mathbf{y}'', x))| \leq 2 \sup_{\mathbf{y} \in \mathrm{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{x})\|_{r,t} \gamma$$

□

The following theorem – building upon the previous robustness result – now offers a *GE* bound for a deep neural network classifier trained under the cross entropy with softmax loss.

Theorem 3.2. A DNN $f_{\mathcal{S}}(\cdot)$ trained on the training set $\mathcal{S}$ under the cross-entropy with softmax loss function in (3.2) obeys with probability $1-\zeta$, for any $\zeta > 0$, the GE bound given by:

$$GE(f_{\mathcal{S}}) \leq 2 \sup_{\mathbf{y}\in \mathrm{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_{r,t}\gamma + M\sqrt{\frac{2q\mathcal{N}_{\mathcal{Y}}(\gamma/2,\ell_t)\log(2)+2\log(1/\zeta)}{m}}$$

for any $\gamma > 0$ and $M < \infty$.

Proof. The GE of a robust deep neural network based classifier follows immediately from the robustness result. In particular, it has been shown in [22], that with probability greater than $1-\zeta$

$$GE \leq \epsilon(\mathcal{S}) + M\sqrt{\frac{2K\log(2)+2\log(1/\zeta)}{m}} \tag{3.5}$$

where M represents the maximum value of loss over all the samples in the sample space $\mathcal{D}$ that can be shown to be finite for a Lipschitz continuous deep neural network [89]. Now, Theorem 3.1 shows that K can be upper bounded by $q\mathcal{N}_{\mathcal{Y}}(\gamma/2,\ell_t)$, it also shows that $\epsilon(\mathcal{S}) \leq 2\sup_{\mathbf{y}\in \mathrm{conv}(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\|_{r,t}\gamma$, leading immediately to the result. □

3.3.2 Regression problems

We now consider learning problems associated with regression tasks where the input data points $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^p$ and the output data points $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^q$. These problems are applicable to various practical scenarios ranging from image denoising to image inpainting and super-resolution [90], whereby the output data corresponds to some corrupted version of the input data. Owing to the nature of the problems and the ultimate goal, loss functions which minimizes the per pixel difference between the output of the network and the ground-truth – such as the ℓ_r-loss – are usually used as training objective. Therefore, next we characterize the performance of a deep neural network regressor $f_{\mathcal{S}}(\cdot)$ trained on a training set $\mathcal{S}$ with respect to the standard ℓ_r loss.

We assume $\mathcal{Y}$ and $\mathcal{X}$ are compact and we use ℓ_t and ℓ_r to measure distances in $\mathcal{Y}$ and $\mathcal{X}$ respectively.

3.3.2.1 ℓ_r loss

The standard ℓ_r-loss function given by:

$$l(f_{\mathcal{S}}(\cdot),(\mathbf{y},\mathbf{x})) = ||\mathbf{x} - f_{\mathcal{S}}(\mathbf{y})||_r \tag{3.6}$$

where $\mathbf{y} \in \mathcal{Y}$ and $\mathbf{x} \in \mathcal{X}$ represent the input and ground-truth output data, respectively.

Leveraging Lemma 3.1 we can now describe the robustness of a deep neural network regressor trained under a ℓ_r-loss.

Theorem 3.3. (Robustness under ℓ_r loss) A DNN $f_{\mathcal{S}}(\cdot)$ trained on a training set $\mathcal{S}$ under the ℓ_r-loss in (3.6) is

$$\left(\mathcal{N}_{\mathcal{D}}(\psi/2,\rho), \left(1 + \sup_{\mathbf{y} \in \mathrm{conv}(\mathcal{Y})} ||\mathbf{J}(\mathbf{y})||_{r,t}\right)\psi\right) - \text{robust}$$

for any $\psi > 0$ and $\mathcal{N}_{\mathcal{D}}(\psi/2,\rho) < \infty$.

Proof. We can establish a $\psi/2$ cover of $\mathcal{D}$ such that $K \leq \mathcal{N}_{\mathcal{D}}(\psi/2,\rho)$ such that $\forall(\mathbf{y}',\mathbf{x}') \in \mathcal{S}$ and $(\mathbf{y}'',\mathbf{x}'') \in \mathcal{D}$, if $(\mathbf{y}',\mathbf{x}')$ and $(\mathbf{y}'',\mathbf{x}'')$ correspond to the same partition, then $\rho((\mathbf{y}',\mathbf{x}'),(\mathbf{y}'',\mathbf{x}'')) \leq \psi$.

Let us now consider two data points $(\mathbf{y}',\mathbf{x}') \in \mathcal{S}$ and $(\mathbf{y}'',\mathbf{x}'') \in \mathcal{D}$ associated with one of the partitions. Then

$$\begin{aligned}
|l(f_{\mathcal{S}},(\mathbf{y}',\mathbf{x}')) - l(f_{\mathcal{S}},(\mathbf{y}'',\mathbf{x}''))| &= \left|||\mathbf{x}' - f_{\mathcal{S}}(\mathbf{y}')||_r - ||\mathbf{x}'' - f_{\mathcal{S}}(\mathbf{y}'')||_r\right| \\
&\overset{(a)}{\leq} ||\mathbf{x}' - f_{\mathcal{S}}(\mathbf{y}') - \mathbf{x}'' + f_{\mathcal{S}}(\mathbf{y}'')||_r \\
&\overset{(b)}{\leq} ||\mathbf{x}' - \mathbf{x}'||_r + ||f_{\mathcal{S}}(\mathbf{y}') - f_{\mathcal{S}}(\mathbf{y}'')||_r \\
&\overset{(c)}{\leq} ||\mathbf{x}' - \mathbf{x}''||_r + \max_{\mathrm{conv}(\mathcal{Y})} ||\mathbf{J}(\mathbf{y})||_{r,t} ||\mathbf{y}' - \mathbf{y}''||_t \\
&\overset{(d)}{\leq} \left(1 + \sup_{\mathbf{y} \in \mathrm{conv}(\mathcal{Y})} ||\mathbf{J}(\mathbf{y})||_{r,t}\right) \rho((\mathbf{y}',\mathbf{x}'),(\mathbf{y}'',\mathbf{x}''))
\end{aligned}$$

The inequalities (a),(b) and (c) hold due to reverse triangle inequality, Minkowski-inequality and Lemma 3.1, respectively, where (d) holds trivially because sup metric ρ upper bounds the distance metric on data space.

It now follows immediately that

$$|l(f_{\mathcal{S}}(\mathbf{y}'),\mathbf{x}')) - l(f_{\mathcal{S}},(\mathbf{x}'',\mathbf{y}''))| \leq \left(1 + \sup_{\mathbf{y}\in\text{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_{r,t}\right)\psi$$

and the theorem follows. □

Finally, leveraging Theorem 3.3, we can also describe a *GE* bound for a deep neural network regressor trained under a ℓ_r-loss.

Theorem 3.4. A DNN $f_{\mathcal{S}}(\cdot)$ trained on the training set $\mathcal{S}$ under the ℓ_r-loss in (3.6) obeys with probability $1-\zeta$, for any $\zeta > 0$, the *GE* bound given by:

$$GE(f_{\mathcal{S}}) \leq \left(1 + \sup_{\mathbf{y}\in\text{conv}(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_{r,t}\right)\psi + M\sqrt{\frac{2n_{\mathcal{D}}\left(\psi/2,\rho\right)\log(2) + 2\log\left(1/\zeta\right)}{m}}$$

for any $\psi > 0$ and $M < \infty$.

Theorem 3.4 derives immediately from Theorem 3.3, by adapting the proof technique of Theorem 3.2.

3.3.3 Discussion

Theorems 3.2 and 3.4 provide various insights that are also aligned with previous results in the literature. In particular, the bounds consist of two terms:

- The second term captures the interplay between the cardinality of the training set $\mathcal{S}$ and the complexity of the data space measured via its covering number. Intuitively, the generalization error decreases with the increase in the cardinality of the training set, and it also decreases with the decrease in the covering number of the data space. It is generally recognized that real-world data is associated with spaces exhibiting small intrinsic dimension – hence bounded covering numbers

[38] – where the optimal covering ball radius can be evaluated using network characteristics [38, 30].

- The first term is associated with the Lipschitz constant of the loss function that, in turn, is proportional to the Lipschitz constant of the deep neural network both for cross-entropy with softmax loss function and the ℓ_r loss function. This – in sharp contrast with existing parameter dependent bounds [59, 18, 20] – then suggests that the generalization capability of a deep neural network does not deteriorate exponentially with the network size/depth, in view of the fact that the Lipschitz constant of a deep neural network depends on a network input-output Jacobian operator norm (hence not direcly on the network depth). In fact, in agreement with empirical results reported in [54], we also show experimentally (in Chapter 5) that the generalization error tends to be directly proportional to the operator norm of the network input-output Jacobian matrix.

It should be noted that the GE bounds derived in our work augment the current literature on generalization error in that the existing bounds typically utilize capacity measures of the hypothesis class – such as the Rademacher complexity or the VC dimension – to characterize the generalization error. These bounds, in contrast to ours, ignore the interplay between the generalization and the properties of the learning algorithm. The algorithmic robustness framework allow us to capture the dependence of the generalization error on the key ingredients of the deep learning algorithm namely the network architecture, algorithm and the sample space. Moreover, the current algorithm dependent studies of the generalization error typically only cater to deep learning classifiers. Our analysis develops GE bounds for both cross-entropy and ℓ_2 loss and therefore is applicable to both regression and prediction.

3.4 Summary

In this chapter, we have studied the generalization behaviour of deep neural networks by building upon the robustness framework. In particular, we have offered new generalization bounds – applicable both to classification and regression problems – that encapsulate key quantities associated with the learning problem, including the complexity of the data space, the cardinality of the training set, and the network Jacobian matrix. Notably, our bounds lead to an entirely new regularization strategy – based on the penalization of the spectral norm of the network Jacobian – that, as we will show in Chapter 5, clearly outperforms existing regularizers both in classification and regression problems (see Chapter 5 for details). In the next chapter, we extend our analysis to include inverse problems – a very important class of problems for which we have well postulated mathematical forward models.

Chapter 4

Model Aware GE Bounds for Inverse Problems

4.1 Inverse Problems

In various signal and image processing challenges arising in practice – including medical imaging, remote sensing, and many more – one often desires to recover a number of latent variables from physical measurements. This class of problems – generally known as *inverse problems* – can often be modelled as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{4.1}$$

where $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^q$ represents a q-dimensional vector containing the physical measurements, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ represents a p-dimensional vector containing the variables of interest, and $\mathbf{n}$ is a bounded perturbation modelling measurement noise (i.e. $||\mathbf{n}|| \leq \eta$). The forward operator modelling the relationship between physical measurements and variables of interests is in turn modelled (in the absence of noise) using a matrix $\mathbf{A} \in \mathbb{R}^{q \times p}$. This forward operator is usually known and satisfies certain regularity conditions such as

Λ_a-Lipschitz continuity whereby [1]

$$||\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2||_2 \leq \Lambda_a ||\mathbf{x}_1 - \mathbf{x}_2||_2, \qquad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X} \tag{4.2}$$

The main challenge in solving (ill-posed) inverse problems via traditional methods relates to the fact that – without any prior assumption – it is not possible to recover the variables of interest from the observations (even when the forward model is perfectly known). Two broad classes of approaches have been adopted to solve inverse problems: (i) *model-based* methods and (ii) *data-driven* methods. Model-based methods exploit knowledge of the forward operator and/or the signal/noise model in order to recover the variables of interest from the measurements [10]. For example, well-known inverse problem recovery algorithms often leverage knowledge of data priors capturing stochastic [91] or geometric structure [92]. On the other hand, data-driven methods do not leverage explicitly the knowledge of the underlying physical and data models; instead, such methods rely on the availability of various data pairs $(\mathbf{y}, \mathbf{x})$ in order to learn how to invert the forward operator associated with the inverse problem [90]. The challenge relates to the fact that these approaches – specially deep learning ones – typically require the availability of various training examples that are not always available in a number of applications such as medical image analysis. This inevitably hinders the applicability of data-driven approaches to inverse problems arising in various scientific and engineering use-cases. Therefore, this has motivated researchers to put forth techniques that leverage both the model and data driven paradigms to solve inverse problems. Some of the noteworthy works in this vein include [4, 17]. In this work,we approach this challenge by offering new generalization guarantees that capture how the generalization ability is affected by various key quantities associated with the learning problem. Such interplay then immediately leads to an entirely new model-

[1] Note that such forward operators encountered in various applications of interest including Magnetic resonance Imaging (MRI), Computed Tomography (CT) etc obey some form of regularity constraint such as given in (4.2).

aware regularization strategy acting as a proxy to import knowledge about the underlying physical model onto the deep learning process [2].

Next, we present an overview of the related work before introducing our system setup generalization bounds applicable to neural network based inverse problem solvers.

4.2 Prior Work

4.2.1 Model-based techniques for inverse problems

The main challenge in solving (ill-posed) inverse problems relates to the fact that – without any prior assumption – it is not possible to recover the variables of interest from the observations (even when the forward model is perfectly known). A naive inversion of the forward map without any structural constraints will not result in meaningful reconstruction. Classical model-based approaches address this challenge via the formulation of optimization problems that include two terms in the objective: (1) a data fidelity term and (2) a data regularization one:

$$\arg\min_{\mathbf{x}} \quad l(\mathbf{A}\mathbf{x}, \mathbf{y}) + r(\mathbf{x}) \tag{4.3}$$

The first term in (4.3) encourages the solution to be consistent with the observations whereas the second, regularization one encourages solutions that conform to a certain postulated data prior. There are a large number of model-based approaches in the literature: Popular variational methods use a regularizer that promotes smoothness of the solutions [93, 94] whereas sparsity-driven methods use regularizers that promote sparsity of the solutions in some transform domain [95, 96, 97]. In addition to the challenging task of determining a suitable data prior, these traditional approaches tend to require relatively complex solvers inevitably restricting their applicability.

[2] The regularization techniques and empirical results are presented in Chapters 5 and 6.

4.2.2 Data-driven techniques for inverse problems

The recent years have witnessed a surge of interest in data-driven approaches – with a focus on deep learning ones – to solve inverse problems [12]. In particular, inspired by the success of deep learning in classification tasks, such approaches typically "solve" an inverse problem by using a neural network that has learnt how to map the model output to the model input based on a number of input-output examples [90]. Such approaches have been applied to a large number of inverse problems such as image denoising [98, 1], image super-resolution [99], MRI reconstruction [100, 101], CT reconstruction [102], and many more. However, these data-driven approaches typically require rich enough datasets – which are not always available in various domains such as medical imaging – in order to learn how to solve the inverse problem [103].

4.2.3 Model-aware data driven approaches

In view of the fact that the underlying physical model is known in various scenarios, there are been an increased interest in model-aware data-driven approaches to inverse problems. Some approaches leverage knowledge of the forward model to provide a rough estimate of the inverse problem solution (e.g. using some form of pseudo-inverse of the forward operator) that is then further processed using a neural network [5, 104, 105]

Another approach that is becoming increasingly popular relies on algorithm unfolding or unrolling [106, 107, 17]. By starting with a typical optimization based formulation to tackle the underlying inverse problem – where knowledge of the physical model is explicitly used – unfolding then maps iterative solvers onto a neural network architecture whose parameters can be further tuned in a data-driven manner.

Finally there is also a new suite of techniques that leverage the knowledge of forward operator as follows: the reconstruction of the desired data vector given the measurements vector is carried out using a (regularized) optimization problem using the underlying model; however, the regularizer within such an optimization problem is itself learnt directly from a set of data examples. One such recent (unsupervised) approach relies on the use of adversarially learnt data dependent regularizers [4]. Another suite of techniques uses instead data representations learnt directly from data in any underlying model based optimization problem. For example, in [108], the authors propose to learn the underlying low dimensional manifold of the latent signal of interest using a generative adversarial network (GAN). This allows them to constrain, in any optimization problem, the reconstructed data to conform to such learnt manifold. While this method yields powerful representations, its training hinges upon the acquisition of a sufficient amount of training data for it to generalize well enough to the test data. A similar approach which employs the structure of a GAN as an implicit regularizer was proposed in [109]. The work shows that a hand crafted network architecture inherently favours solutions that look like natural images – hence can serve as a suitable prior in image restoration tasks. Finally there are approaches where a learned denoising autoencoder is treated as a regularization step in an iterative reconstruction method [110, 111].

Our work departs from these contributions in the sense that – whereas we also use a deep network to solve an inverse problem – we leverage knowledge of the underlying forward operator model via appropriate regularization strategies deriving from a principled generalization error analysis. The proposed approach gives rise to a prior which is tailored to a particular inverse problem without requiring additional preprocessing.

4.2.4 Other related work

There is a considerable volume of literature offering analysis of the generalization ability of deep neural networks demonstrating that the generalization error of highly paramterized models can be bounded in terms of certain parameter norms [19, 20]. To the best of knowledge, all of these bounds are applicable to classification problems or model-agnostic regression based ones. Our current work addresses this issue, offering a study of the generalization ability of deep neural networks based inverse problems solvers, leading to entirely new gradient based regularization strategies allowing to incorporate knowledge into the learning process.

Finally, various works have already proposed approaches to efficiently introduce Lipschitz regularity in deep neural networks. However, as stated earlier, none of these results specifically consider problems for which the knowledge of data generation model is available and therefore fail to leverage it. The reader should refer to Section 3.1 for a detailed summary of such contributions.

4.3 The Setup

We consider the linear observation model in eq. (4.1), with the following additional assumptions: the input space $\mathcal{X} \subseteq \mathbb{R}^p$ is compact with respect to the ℓ_2 metric; the noise space $\mathcal{E} = \{\mathbf{n} : \|\mathbf{n}\|_2 \leq \eta\} \subseteq \mathbb{R}^q$ is also compact with respect to the ℓ_2 metric; and the output space – which is defined as $\mathcal{Y} = \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} : \mathbf{x} \in \mathcal{X}, \mathbf{n} \in \mathcal{E}\} \subseteq \mathbb{R}^q$ – can also be shown to be compact with respect to the ℓ_2 metric. Finally, we also define the sample space $\mathcal{D} = \{(\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) : \mathbf{x} \in \mathcal{X}, \mathbf{n} \in \mathcal{E}\}$ that is compact with respect to the ℓ_2 metric.

Our approach to solve this problem is based on the standard supervised learning paradigm. We assume access to a training set $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i)\}_{i \leq m}$ consisting of m data points drawn independently and identically distributed (IID) from the sample space $\mathcal{D}$ according to the unknown data distribution μ, consistent with the forward model in (4.1). We use such a training set to learn a hypothesis $f_{\mathcal{S}} : \mathcal{Y} \rightarrow \mathcal{X}$ mapping the measurement variables to variables of interest. We then use such a hypothesis to map new measurement variables $\mathbf{y} \in \mathcal{Y}$ to the variables of interest $\mathbf{x} \in \mathcal{X}$ that were not necessarily originally present in the training set.

We restrict our attention to mappings based on feed-forward neural networks $f_{\mathcal{S}}$ for this work (2.1). One is typically interested in the performance of the learnt neural network not only on the training data but also on (previously unseen) testing data. Therefore, it is useful to quantify the generalization error associated with the learnt neural network given by:

$$GE(f_{\mathcal{S}}) = |l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}})| \tag{4.4}$$

where $l_{\text{exp}}(f_{\mathcal{S}}) = \mathbb{E}_{\mathbf{s}\sim\mu}[l(f_{\mathcal{S}},\mathbf{s})]$ represents the expected error, $l_{\text{emp}}(f_{\mathcal{S}}) = \frac{1}{m}\sum_i l(f_{\mathcal{S}},\mathbf{s}_i)$ represents the empirical error, and the loss function $l : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}_0^+$ — which measures the discrepancy between the neural network prediction and the ground truth — is taken to be the ℓ_2 distance given by:

$$l(f_{\mathcal{S}},\mathbf{s}) = \|f_{\mathcal{S}}(\mathbf{y}) - \mathbf{x}\|_2 \tag{4.5}$$

Our ensuing analysis offers bounds to the generalization error in (4.4) of deep feed-forward neural networks based inverse problems solvers as a function of a number of relevant quantities. These quantities include the covering number of the sample space $\mathcal{D}$, the size of the training set $\mathcal{S}$, and properties of the network encapsulated in its input-output Jacobian matrix given in eq. (2.5). The bounds also depend on quantities associated with the linear model in eq. (4.1) such as the forward operator and the noise bound. Our analysis will therefore also inform how to import knowledge about the forward-operator associated with the inverse problem onto the learning procedure in order to improve the generalization error.

4.4 Generalization Error Bounds

Our analysis builds upon the *algorithmic robustness* framework in [22].

Definition 4.1. A learning algorithm is said to be $(K,\epsilon(\mathcal{S}))$-robust if the sample space $\mathcal{D}$ can be partitioned into K disjoint sets $\mathcal{K}_k$, $k = 1,\ldots,K$, such that for all$(\mathbf{x}_i,\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i) \in \mathcal{S}$ and all $(\mathbf{x},\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) \in \mathcal{D}$

$$\begin{aligned}&(\mathbf{x}_i, \mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i), (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) \in \mathcal{K}_k \\ &\qquad \Longrightarrow \left| l(f_{\mathcal{S}}, (\mathbf{x}_i, \mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i)) - l(f_{\mathcal{S}}, (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n})) \right| \leq \epsilon(\mathcal{S})\end{aligned}$$

This notion has already been used to analyse the performance of deep neural networks in [38, 33, 30] and previously in Chapter 3. However, such analyses applicable to classification tasks do not carry over immediately to inverse problems based tasks where – in addition to exploiting knowledge about the forward operator associated with the inverse problem for the computation of $\epsilon(\mathcal{S})$ and K – there are some technical complications that may arise due the fact that the loss functions are typically unbounded [3].

We begin addressing these challenges by offering a simple smoothness based result that showcases how the distance between the neural network estimates of the variables of interest depends on the distance between the variables of interest themselves and, importantly, the Jacobian of the network, the Jacobian of the composition of the network with the forward model associated with the inverse problem, and the noise power associated with the inverse problem.

This result is important because it shows that in the presence of bounded Lipschitz constants of the relevant mappings and noise value, the DNN learns a smooth mapping.

Theorem 4.1. Consider a neural network $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ based solver of the inverse problem in (4.1), learnt using a training set $\mathcal{S}$. Then, for any $(\mathbf{x}_1, \mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{n}_1)$, $(\mathbf{x}_2, \mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{n}_2) \in \mathcal{D}$, it follows that

$$||f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)||_2 \leq \Lambda_{f\circ a}||\mathbf{x}_2 - \mathbf{x}_1||_2 + 2\eta\Lambda_f$$

where $\Lambda_{f\circ a}$ and Λ_f are upper bounds to the Lipschitz constants of the neural network and the composition of the neural network and the forward

[3] Existing work applies to uniformly bounded loss function (e.g. [22, 38]).

operator respectively, given by:

$$\Lambda_{f\circ a} = \sup_{\mathbf{y}\in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2, \qquad \Lambda_f = \sup_{\mathbf{y}\in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_2 \tag{4.6}$$

Proof. We first note that the line between $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{n}_1$ and $\mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{n}_2$ is given by $\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2$ where $\theta \in (0,1)$ and $\bar{\theta} = 1-\theta$. Let us now define a function $h(\theta)$ as follows:

$$h(\theta) = f_{\mathcal{S}}(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2) = f_{\mathcal{S}}\left(\mathbf{A}\left(\bar{\theta}\mathbf{x}_1 + \theta\mathbf{x}_2\right) + \bar{\theta}\mathbf{n}_1 + \theta\mathbf{n}_2\right)$$

By the generalized fundamental theorem of calculus, it can be shown that:

$$f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1) = \int_0^1 \frac{dh(\theta)}{d\theta} d\theta$$

where

$$\frac{d}{d\theta}(h(\theta)) = \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\left[\mathbf{A}\left(\mathbf{x}_2 - \mathbf{x}_1\right) + \left(\mathbf{n}_2 - \mathbf{n}_1\right)\right]$$

Then, from the sub-multiplicative property of matrix norms, it is immediate to show that:

$$\begin{aligned}
&\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\
&= \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\left[\mathbf{A}\left(\mathbf{x}_2 - \mathbf{x}_1\right) + \left(\mathbf{n}_2 - \mathbf{n}_1\right)\right] d\theta \right\|_2 \\
&\leq \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) d\theta \right\|_2 + \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)(\mathbf{n}_2 - \mathbf{n}_1) d\theta \right\|_2 \\
&\leq \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A} d\theta \right\|_2 \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right) d\theta \right\|_2 \|\mathbf{n}_2 - \mathbf{n}_1\|_2
\end{aligned}$$

It is also possible to show that:

$$\left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A} d\theta \right\|_2 \overset{(a)}{\leq} \int_0^1 \|\mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A}\|_2 d\theta \leq \sup_{\substack{\mathbf{y}_1,\mathbf{y}_2\in\mathcal{Y} \\ \theta\in[0,1]}} \|\mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A}\|_2$$

Therefore, given that $\bar{\theta}\mathbf{y}_1+\theta\mathbf{y}_2$ is in convex-hull of $\mathcal{Y}$ for $\theta\in[0,1]$, it follows immediately that:

$$\begin{aligned}\|f_{\mathcal{S}}(\mathbf{y}_2)-f_{\mathcal{S}}(\mathbf{y}_1)\|_2 &\le \sup_{\mathbf{y}\in conv(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2\|\mathbf{x}_2-\mathbf{x}_1\|_2 + \sup_{\mathbf{y}\in conv(\mathcal{Y}})\|\mathbf{J}(\mathbf{y})\|_2\|\mathbf{n}_2-\mathbf{n}_1\|_2\\ &\le \sup_{\mathbf{y}\in conv(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2\|\mathbf{x}_2-\mathbf{x}_1\|_2 + 2\eta \sup_{\mathbf{y}\in conv(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\|_2\end{aligned} \tag{4.7}$$

where $conv(\mathcal{Y})$ represents the convex hull of $\mathcal{Y}$. □

We now state another theorem – building upon Theorem 1 – articulating about the robustness of a deep neural network based solver of an inverse problem.

Theorem 4.2. A neural network trained to solve the inverse problem in (4.1) based on a training set $\mathcal{S}$ is $(K,\epsilon(\mathcal{S}))$-robust such that for any $\delta>0$,

$$K\le \mathcal{N}_{\mathcal{X}}(\delta,\ell_2), \qquad \epsilon(\mathcal{S})\le 2(1+\Lambda_{f\circ a})\delta+2\Lambda_f\eta$$

where $\mathcal{N}_{\mathcal{X}}(\delta,\ell_2)$ is the covering number of $\mathcal{X}$.

Proof. We can construct a finite δ-cover $\mathcal{X}'=\{\mathbf{x}'_i, i=1,\dots,K\}$ of the compact space $\mathcal{X}$ with $K\le\mathcal{N}_{\mathcal{X}}(\delta,\ell_2)$. We can therefore also construct a finite cover $\mathcal{D}'=\{(\mathbf{x}'_i,\mathbf{A}\mathbf{x}'_i),\mathbf{x}'_i\in\mathcal{X}', i=1,\dots K\}$ of the space $\mathcal{D}=\{(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}):\mathbf{x}\in\mathcal{X},\mathbf{n}\in\mathcal{E}\}$.

This implies that the sample space $\mathcal{D}$ can be partitioned into K disjoint subsets $\mathcal{K}_i, i=1,\dots,K$ where $\mathcal{K}_i$ corresponds to the Voronoi region of $(\mathbf{x}'_i,\mathbf{y}'_i=\mathbf{A}\mathbf{x}'_i),\mathbf{x}'_i\in\mathcal{X}'$. Consequently, for a point $(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n})$ taken from the subset $\mathcal{K}_i$ we can guarantee:

$$\|\mathbf{x}'_i-\mathbf{x}\|_2\le\delta \tag{4.8}$$

Let us now choose $(\mathbf{x}_1,\mathbf{y}_1=\mathbf{A}\mathbf{x}_1+\mathbf{n}_1)\in\mathcal{S}$ and $(\mathbf{x}_2,\mathbf{y}_2=\mathbf{A}\mathbf{x}_2+\mathbf{n}_2)\in\mathcal{D}$ from a particular subset in our partitioned $\mathcal{D}$. Then,

$$\begin{aligned}|l(f_{\mathcal{S}},(\mathbf{x}_2,\mathbf{y}_2))-l(f_{\mathcal{S}},(\mathbf{x}_1,\mathbf{y}_1))| &= |\|\mathbf{x}_2-f_{\mathcal{S}}(\mathbf{y}_2)\|_2-\|\mathbf{x}_1-f_{\mathcal{S}}(\mathbf{y}_1)\|_2| \\ &\overset{(a)}{\leq} \|\mathbf{x}_2-\mathbf{x}_1\|_2+\|f_{\mathcal{S}}(\mathbf{y}_2)-f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\ &\overset{(b)}{\leq} \|\mathbf{x}_2-\mathbf{x}_1\|_2+\Lambda_{f\circ a}\|\mathbf{x}_2-\mathbf{x}_1\|_2+\Lambda_f\|\mathbf{n}_2-\mathbf{n}_1\|_2 \\ &\overset{(c)}{\leq} 2(1+\Lambda_{f\circ a})\delta+2\Lambda_f\eta \qquad (4.9)\end{aligned}$$

where the inequality (a) is due to reverse triangle inequality and Minkowski-inequality, (b) holds because of Theorem 1. Finally (c) holds due to (4.8) and because η upper bounds the ℓ_2 norm of noise.

We have therefore shown that we can partition the set $\mathcal{D}$ onto K non-overlapping subsets so that if a training sample $(\mathbf{x}_1,\mathbf{y}_1)\in\mathcal{S}$ and another sample $(\mathbf{x}_2,\mathbf{y}_2)\in\mathcal{D}$ belong to the same subset then (4.9) holds [4]. □

We now state our main theorem relating to the generalization error of a deep neural network trained to solve an inverse problem.

Theorem 4.3. A neural network trained to solve the inverse problem in (4.1) based on a training set $\mathcal{S}$ consisting of m i.i.d. training samples obeys with probability $1-\zeta$, for any $\zeta>0$, the *GE* bound given by:

$$GE(f_{\mathcal{S}})\leq 2(1+\Lambda_{f\circ a})\delta+2\Lambda_f\eta+M\sqrt{\frac{2n_{\mathcal{X}}(\delta,\ell_2)\log(2)+2\log(1/\zeta)}{m}}$$

for $\max_{(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n})}|l(f_{\mathcal{S}},(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}))|\leq M<\infty$ and any $\delta>0$.

Proof. We first establish a simple Lemma.

Lemma 4.1. The Lipschitz constant of a differentiable function f on a compact set $\mathcal{Z}$ is bounded.

[4] Note that this bounding technique produces a slightly different bound than an alternative one where we would bound the second term $\|f_{\mathcal{S}}(\mathbf{y}_2)-f_{\mathcal{S}}(\mathbf{y}_1)\|_2$ on the right hand side of 4.9(a) by $\Lambda_f\|\mathbf{y}_2-\mathbf{y}_1\|$ (instead of $\Lambda_{f\circ a}\|\mathbf{x}_2-\mathbf{x}_1\|$ which is possible via Theorem 1). However, the proposed bounding technique results in a tighter characterization of $\epsilon(\mathcal{S})$ since $\Lambda_{f\circ a}=\sup_{\mathbf{y}\in conv(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2\leq\sup_{\mathbf{y}\in conv(\mathcal{Y})}\|\mathbf{J}(\mathbf{y})\|_2\|\mathbf{A}\|_2=\Lambda_f\Lambda_a$.

Proof. Let $f : \mathbb{R}^p \rightarrow \mathbb{R}^q$ be a differentiable function, defined on a compact set $\mathcal{Z} \subseteq \mathbb{R}^p$. Let also $g(\theta) = f(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))$, for some $\theta \in [0,1]$, so that $g(0) = f(\mathbf{z}')$ and $g(1) = f(\mathbf{z}'')$ where $\mathbf{z}', \mathbf{z}''$ are any two fixed points taken for $\mathcal{Z}$. Then, by the fundamental theorem of calculus, we have

$$f(\mathbf{z}') - f(\mathbf{z}'') = \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta(\mathbf{z}' - \mathbf{z}'')$$

where $\mathbf{J}(\mathbf{z})$ is the Jacobian matrix of f at $\mathbf{z}$.

From the multiplicative property of norms, we also have that

$$||f(\mathbf{z}') - f(\mathbf{z}'')|| \leq \left\| \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta \right\|_2 ||\mathbf{z}' - \mathbf{z}''||_2$$

Next, by the triangle inequality for integrals, it can be shown that

$$\begin{aligned} \left\| \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta \right\|_2 &\leq \sup_{\substack{\mathbf{z}',\mathbf{z}'' \in \mathcal{Z} \\ \theta \in [0,1]}} ||\mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))||_2 \\ &\leq \sup_{\mathbf{z} \in conv(\mathcal{Z})} ||\mathbf{J}(\mathbf{z})||_2 \end{aligned}$$

where $conv(\mathcal{Z})$ represents the convex hull of the compact set $\mathcal{Z}$. Note that the Carathéodory's theorem of convex hulls can be used to prove that the convex hull of compact set in a finite dimensional space $\mathbb{R}^p$ is also compact [112].

Next, for a continuous function f defined on a compact set, there exists a finite λ_0 such that [113, 89].

$$\left| \frac{\partial}{dz_j}(f(\mathbf{z})_i) \right| \leq \lambda_0 \tag{4.10}$$

where $\frac{\partial}{dz_j}(f(\mathbf{z})_i)$ is the element at row (i,j)-th element of the Jacobian matrix $\mathbf{J}$. This, then leads to the following

$$\sup_{\mathbf{z} \in conv(\mathcal{Z})} ||\mathbf{J}(\mathbf{z})||_2 \overset{(a)}{\leq} \sup_{\mathbf{z} \in conv(\mathcal{Z})} c||\mathbf{J}(\mathbf{z})||_\infty \overset{(b)}{\leq} cp\lambda_0$$

where (a) is due to the equivalence of matrix norms and c is a constant dependent on the dimensions of the Jacobian matrix [114]. Finally the last inequality follows from the definition of the $||.||_\infty$ matrix norm [115]. □

We are now in a position to prove the Theorem. In particular, it can be shown that the GE of a $(K, \epsilon(\mathcal{S}))$-robust deep neural network, with probability greater than $1-\zeta$, obeys [22]

$$GE \leq \epsilon(\mathcal{S}) + \max_{(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n})} |l(f_{\mathcal{S}},(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}))| \sqrt{\frac{2K\log(2)+2\log(1/\zeta)}{m}} \tag{4.11}$$

We can immediately use the robustness result in Theorem 2 to determine two quantities in this generalization error bound: $\epsilon(\mathcal{S})$ and K. However – in contrast with existing results that assume that the loss function is uniformly bounded so that $\max_{(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n})} |l(f_{\mathcal{S}},(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}))| \leq M < \infty$ (e.g. see [22]) – the loss function associated with our inverse problem is not necessarily bounded. However, it is still possible to show that $\max_{(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n})} |l(f_{\mathcal{S}},(\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}))|$ is finite.

In particular, let us observe that $\forall\ (\mathbf{x},\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{n}),(\mathbf{x}',\mathbf{y}'=\mathbf{A}\mathbf{x}'+\mathbf{n}') \in \mathcal{D}$

$$\begin{aligned} |l(f_{\mathcal{S}},(\mathbf{x},\mathbf{y}))-l(f_{\mathcal{S}},(\mathbf{x}',\mathbf{y}'))| &= \Big| ||\mathbf{x}-f_{\mathcal{S}}(\mathbf{y})||_2 - ||\mathbf{x}'-f_{\mathcal{S}}(\mathbf{y}')||_2 \Big| \\ &\leq ||\mathbf{x}-\mathbf{x}'||_2 + ||f_{\mathcal{S}}(\mathbf{y})-f_{\mathcal{S}}(\mathbf{y}')||_2 \\ &\overset{(a)}{\leq} ||\mathbf{x}-\mathbf{x}'||_2 + \Lambda_f ||\mathbf{y}-\mathbf{y}'||_2 \\ &\overset{(b)}{\leq} (1+\Lambda_f)||(\mathbf{x},\mathbf{y})-(\mathbf{x}',\mathbf{y}')||_2 \end{aligned}$$

where (a) is due to Corollary 2 in [38] and (b) holds because the metric on $\mathcal{D}$ upper bounds the metrics on constituent metric spaces $\mathcal{X}$ and $\mathcal{Y}$.

Let us also observe that the Lipschitz constant of the loss function is finite because – via Lemma 4.1 – the Lipschitz constant of the neural network Λ_f is also finite.

This immediately implies that the loss function is Lipschitz continuous hence continuous, and – by the Extreme Value theorem [89] – that it is also bounded on $\mathscr{D}$, so that $\max_{(\mathbf{x},\mathbf{y}=\mathbf{Ax}+\mathbf{n})} |l(f_{\mathcal{S}}, (\mathbf{x}, \mathbf{y} = \mathbf{Ax} + \mathbf{n}))| \leq M < \infty$.

The Theorem follows immediately from Theorem 4.2. □

One can derive various insights from this theorem that are applicable to any differentiable feed forward neural network along with any linear forward map: (1) first, in line with traditional bounds [19, 21], the generalization error depends on the size of training set $\mathcal{S}$; (2) second, in line with more recent bounds [38, 33, 77], the generalization error also depends on the complexity of the data space $\mathscr{D}$; [5] (3) third, although the ℓ_2-loss is unbounded in nature, on a compact sample space, the DNN is able to predict samples such that the loss is finite and therefore the GE is provably bounded; 4) Finally, Theorem 4.3 also reveals that the operator norm of the Jacobian of the network and the composite map also play a critical role: the lower the value of these norms, the lower the generalization error. More importantly, the proposed generalization bound is also non-vacuous in the network parameters because as opposed to the product of the norms of layer-wise weight matrices appearing in other generalization error bounds such as [33, 18, 20], the norm of the network Jacobian matrix does not seem to exhibit exponential dependence on network depth. This is in sharp contrast with existing generalization bounds that typically contain a term that deteriorates exponentially with depth.

It should also be noted that for the linear inverse problems in imaging, the input space $\mathcal{X}$ can be assumed to be a C_M regular k-dimensional manifold [116]. The constant C_M varies for different manifolds and represents their "intrinsic" properties. This is a reasonable assumption for the visual data and has previously been used to represent such input spaces. The covering number for such manifolds can be bounded via $\left(\frac{2C_M}{\delta}\right)^k$ [116, 30].

The bound in Theorem 4.3 reveals that a small $||\mathbf{JA}||_2$ should translate to an improvement in the generalization performance of the network on the

[5] The complexity of the sample space – which can be captured via its covering number – is often small in view of the fact that in various applications data lies on a manifold with small intrinsic dimension [38].

inverse problem. We exploit our analysis to propose regularization strategies in Chapter 5 and show that penalizing the norms of **J** and **JA** indeed results in better reconstruction performance on the held-out data. A number of model based deep learning techniques have been proposed in literature that propose to leverage the knowledge of forward map in the reconstruction process. However, our proposed techniques are different in that we propose a plug and play prior which can be incorporated at run time and stems from a principled generalization error analysis.

We now specialize our bounds to the special case of inverse problems when the signal of interest is assumed to be sparse.

4.5 Specialization for Sparse Signals

Sparsity is a desirable quality for the recovery of signals that have localized energy and can be compressed by means of an appropriate basis expansion. In a large number of inverse problems such as tomography, image super-resolution, denoising and deblurring [117, 118, 119, 120], it is reasonable to assume that the signal of interest has sparse representation in some basis. Therefore, in this section we specialize the *GE* bounds described in the previous section for the case when the space $\mathcal{X}$ consists of unit ℓ_2-norm k-sparse vectors [6], i.e.

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p : ||\mathbf{x}||_0 \leq k, ||\mathbf{x}||_2 \leq 1\} \tag{4.12}$$

where $||\mathbf{x}||_0$ computes the cardinality of non-zero elements of $\mathbf{x}$.

We also consider that an appropriately trained network – using a training set $\mathcal{S}$ – is employed to deliver an estimate of the sparse vector $\mathbf{x}$ given the measurement vector $\mathbf{y}$.

We can now immediately specialize the results appearing in Theorems 4.2 and 4.3 to this particular setting. The following uper bound on the

[6] The analysis can be easily extended to the settings where the signal of interest, $\mathbf{x}$ is sparse in some basis.

covering number of the input space will be very useful [121]:

$$n_{\mathcal{X}}\left(\frac{\delta}{2},\ell_2\right) \leq \left(\frac{pe}{k}\right)^k \left(1+\frac{4}{\delta}\right)^k \tag{4.13}$$

We are now in a position to present our main result that bounds the generalization error for the sparse signal recovery by DNNs.

Corollary 1. Consider again the spaces $\mathcal{X}$ and $\mathcal{Y}$, equipped with a ℓ_2 metric, the space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ compact with the ρ, and the Lipschitz continuous mapping in (4.1). It follows that a d-layer DNN based regressor $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \rightarrow \mathcal{X}$ trained on a training set $\mathcal{S}$ consisting of m i.i.d. training samples obeys with probability $1-\zeta$, for any $\zeta > 0$, the generalization error bound given by:

$$GE(f_{\mathcal{S}}) \leq \left(1+\Lambda_{f\circ a}\right)\delta + 2\eta\Lambda_f + M\sqrt{\frac{2\,(pe/k)^k\,(1+4/\delta)^k \log(2) + 2\log(1/\zeta)}{m}}$$

for any $\delta > 0$ and $M < \infty$.

Proof. The result follows directly from Theorem 4.3 and eq. (4.13). □

The results embodied in Corollary 1 can be used to illuminate further the performance of sparse approximation based on deep learning networks. In particular, let us assume we employ a regularization strategy during the training phase constraining the Lipschitz constant of the network to be less than one [36].

This leads immediately to another generalization error bound holding with probability $1-\zeta$

$$GE(f_{\mathcal{S}}) \leq \left(1+\Lambda_{f\circ a}\right)\delta + 2\eta\Lambda_f + M\sqrt{\frac{2\,(pe/k)^k\,(1+4/\delta)^k \log(2) + 2\log(1/\zeta)}{m}} \tag{4.14}$$

for any $\zeta > 0$ and any $\delta > 0$, and by setting $\delta = o\left(m^{-\frac{1}{k}}\right)$ and by setting trivially ζ to be a function of m such that $\log(1/\zeta)/m = o(1)$, to another generalization bound behaving as follows

$$GE(f_{\mathcal{S}}) \leq 2 \cdot \eta + o(1) \tag{4.15}$$

This suggests that – with the increase of the number of training samples m – the generalization ability of a deep neural network is limited only by the level of the noise independently of the parameter values of the linear observation model, namely q, p, k, Λ_f and $\Lambda_{f \circ a}$. Instead, these parameters mainly influence the speed at which the generalization error asymptotics kick-in. In turn, in view of the fact that the generalization error is upper bounded by the sum of the expected and empirical error, it is also possible to upper bound the expected sparse approximation error associated with a deep neural network as follows:

$$l_{\text{exp}}(f_{\mathcal{S}}) \leq l_{\text{emp}}(f_{\mathcal{S}}) + GE(f_{\mathcal{S}}) \leq l_{\text{emp}}(f_{\mathcal{S}}) + 2 \cdot \eta + o(1) \tag{4.16}$$

Recent results suggest that deep neural networks – with a sufficient number of parameters – tend to memorize the training dataset [122] suggesting that

$$l_{\text{exp}}(f_{\mathcal{S}}) \leq GE(f_{\mathcal{S}}) \leq 2 \cdot \eta + o(1) \tag{4.17}$$

4.5.1 Comparison with Basis Pursuit Denoising (BPDN)

We conclude by comparing how the performance of a deep neural network compares to the performance of a well-known algorithm – BPDN – in sparse approximation problems.

Theorem 4.4 ([95])**.** Consider the linear observation model in (4.1) where $\mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p : ||\mathbf{x}||_0 \leq k\}$ and $\mathbf{y} \in \mathcal{Y} = \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \in \mathbb{R}^q : ||\mathbf{x}||_0 \leq k, ||\mathbf{n}||_2 \leq \eta\}$. Consider also the sparse approximation algorithm delivering an estimate of $\mathbf{x}$ from $\mathbf{y}$ given knowledge of $\mathbf{A}$:

$$\widetilde{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^p} ||\mathbf{x}||_1 \qquad \text{subject to,} \qquad ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2 \leq \varepsilon$$

where $\epsilon \geq \eta$. It follows – under the assumption that $k \leq (1+\mu)/4\mu$ – the error of the approximation delivered by this algorithm can be bounded as follows:

$$\|\widetilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \frac{\eta + \varepsilon}{\sqrt{1 - \mu(4k-1)}}$$

where μ corresponds to the mutual coherence of the matrix $\mathbf{A}$.

This sparse approximation algorithm – along with other sparse approximation algorithms based on convex optimization approaches or greedy approaches (see [123] and references within) – are known to exhibit a phase transition. Here, when the data sparsity $k \leq (1+\mu)/4\mu$, the algorithm provides a reconstruction that scales with the amount of noise η; this is akin to the behaviour of the sparse approximation delivered by a deep neural network. On the other hand, when the data sparsity $k > (1+\mu)/4\mu$ the algorithm does not give any reconstruction guarantees but the deep neural network may still be able to deliver an appropriate reconstruction of the sparse vector given its under-sampled linear observation. In fact, reference [16] has empirically demonstrated that the performance of a DNN degrades gradually as q decreases in relation to p and k.

4.6 Summary

In this chapter, we presented a generalisation error analysis for deep learning methods applicable to inverse problems. Our bound contains key ingredients associated with the learning problem namely the complexity of the data space, the size of the training set, upper bound on the Lipschitz constant of the deep neural network; Lipschitz constant of the forward operator; and the upper bound on the Lipschitz constant of the composition of the forward operator with the neural network. We also specialized our analysis for the setting where the signal of interest is sparse. We also shed light on how the bounds that we propose compare with the classical results present in compressed sensing literature.

Chapter 5

Regularization for Robust Learning

Our generalization error bounds in Chapter 3 demonstrate a direct dependence between the generalization ability and the operator norm of the network input-output Jacobian matrix. This motivates an entirely new regularization strategy that can outperform existing neural network regularization techniques such as weight decay, weight orthogonalization [33] and penalizing the Frobenius norm of the Jacobian [38, 39] as shown in the sequel.

5.1 Model Agnostic Regularization

Building upon the insights associated with our bounds applicable to classification or regression problems in Chapter 3, we now propose a new regularization strategy applicable to scenarios where one adopts an ℓ_2-metric both on the network input and output.

In particular, by adopting such an Euclidean metric, the bounds suggest that the generalization ability of a deep neural network depends on the maximum value of the spectral norm of the network input-output Jacobian matrix over the convex hull of the input space. We therefore propose a new regularization technique that penalizes the sum of the spectral norms of the network Jacobian matrix evaluated at the various training samples within the training set, leading immediately to the training objective given by

$$\frac{1}{m}\sum_{i=1}^{m} l(f_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i) + \lambda \sum_{i=1}^{m} ||(\mathbf{J}(\mathbf{y}_i)||_2^2 \tag{5.1}$$

where $||\cdot||_2$ represents the spectral norm of its matrix argument. The hyperparameter λ balances between the desire to minimize the empirical error and the desire to minimize the spectral norm of the network Jacobian

Note that this new training objective encourages the network to explore solutions in regions in the parameter space associated with Jacobians with small spectral norms which – owing to the nonlinear nature of a deep neural network – allows for a broader parameter search space. This is in contrast to more restrictive techniques such as (1) weight decay that constrains the network weights to exhibit small norms (hence, these techniques do not take into account the correlation between the rows of weight matrices) or else (2) weight orthogonalization that constrains the weights to lie on Stiefel manifold [33, 30].

5.1.1 Experiments

We now conduct a series of experiments to gauge the effectiveness of the proposed regularizer on classification as well as regression tasks. We consider both fully connected as well as convolutional neural networks regularized with our proposed gradient based regularizer (5.1) (referred to as JS) or other conventional weight based regularizers such as weight decay (WD), weight orthogonalization (WO) [33, 81], Frobenius norm of the Jacobian (JF) [38, 39].

5.1.1.1 Classification

We consider a standard image classification problem involving the MNIST dataset. We train feedforward networks composed of 3 fully connected hidden layers of size 784 and a classification layer, taking images from the MNIST dataset unrolled into a 784 dimensional vector. We use an SGD based optimizer to train the network on 2000, 10000 and 40000 samples taken from

Table 5.1: Generalization error and test accuracy(%) for a 4 layer feed-forward DNN classifier, trained on MNIST.

	2000 samples		10000 samples		40000 samples	
	acc	GE	acc	GE	acc	GE
WD	92.3	0.32	96.3	0.14	97.7	0.04
WO	92.2	0.25	96.2	0.11	97.7	0.05
JF	94.5	0.15	96.6	0.13	98.2	0.03
JS	94.5	0.16	97.0	0.06	98.4	0.03

the MNIST training set.

Table 5.1 reports experimental results under WD, WO, JF and JS. These results suggest that Jacobian based regularizations strategies and specially JS have a huge advantage over the weight based regularizers in terms of test accuracy and generalization ability. For small training sets, JS regularization clearly outperforms the other competing regularizers. For larger training sets, JS regularization also outperforms other regularizers, though weight based regularizers appear to become increasingly competitive. Overall, in line with our analysis, our proposed regularizer appears to generalize much better by making the network robust to overfitting specially in scenarios where one has access to limited training data.

5.1.1.2 Regression

We now consider a classical image denoising problem involving the reconstruction of a clean image given a noisy version of the image (corrupted with Gaussian noise). The average Peak Signal to Noise Ratio (PSNR) of the noisy dataset is 25.01 dB. We use the 3-layer version of the classical DnCNN model from [1] with 32 filters of size 3×3 followed by ReLU in first layer, 32 filters of dimension 3×3 followed by batch normalization and ReLU in the second layer and 1 filter of size 3×3 in the third layer. We use 64×64 cutouts of the greyscale images taken from BSD300 dataset for training and testing purposes. We also use an SGD based optimizer to train the network on 40, 200 and 400 samples.

We report the performance of the network trained with our proposed regularizer against networks trained with WD, WO, JF in Table 5.2. Our

Table 5.2: Generalization error and test PSNR (dB) for a 3-layer DnCNN, trained for 140 epochs on BSD300.

	40 samples		200 samples		400 samples	
	PSNR	GE	PSNR	GE	PSNR	GE
WD	25.56	2.05	27.11	1.68	28.17	2.12
WO	25.42	2.18	26.91	1.69	27.72	2.28
JF	26.78	2.63	26.92	1.77	27.55	2.31
JS	27.22	2.49	27.26	1.74	27.94	2.14

results also show that JS outbeat the competing regularizers not only in terms of PSNR but also in optimization speed. Note that for large training samples the performance of JS and WD eventually become comparable if the network is trained for sufficiently large number of epochs. However, the convergence speed of JS is much faster than that of WD. This not only shows that JS regularization improves generalization capability but also – importantly – it shows that our regularizer also improves on the learning process (in the relevant data-limited regime). A visual comparison of the reconstructed images has been presented in Figure 5.1.

To conclude, in line with our analysis, we offer one additional result showcasing that generalization appears to be intimately related to the spectral norm of the network Jacobian. In particular, Figure 5.2 compares the value of the network input-output Jacobian spectral norm for a deep neural network trained under standard SGD with no regularization and a deep neural network trained under SGD for various regularization strategies. It can be seen that in both cases the spectral norm of the Jacobians decreases gradually with the increase in the number of training epochs. These trends apply not only to this denoising tasks but also other regression and classification ones.

5.2 Model-Aware Jacobian Regularization

Our approach to leverage knowledge about the inverse problem model onto the learning process involves regularization. In particular, Theorem 4.3 suggests that penalizing the spectral norm of the Jacobian of the neural network

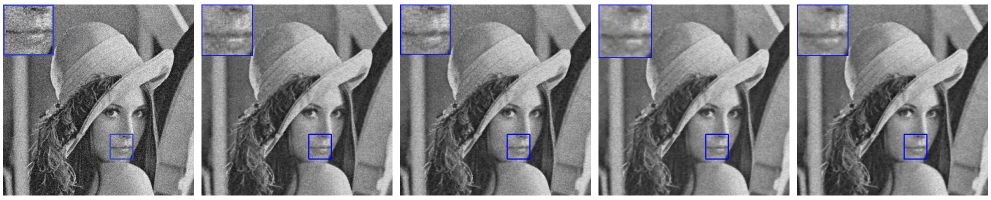


Figure 5.1: Sample results of the denoised images for $m = 40$ using a 5-layer DnCNN [1]. (Left to Right) noisy (PSNR = 27.02), WD (PSNR = 28.08), WO (PSNR = 28.04), JF (PSNR = 28.63), JS (PSNR = 29.03).

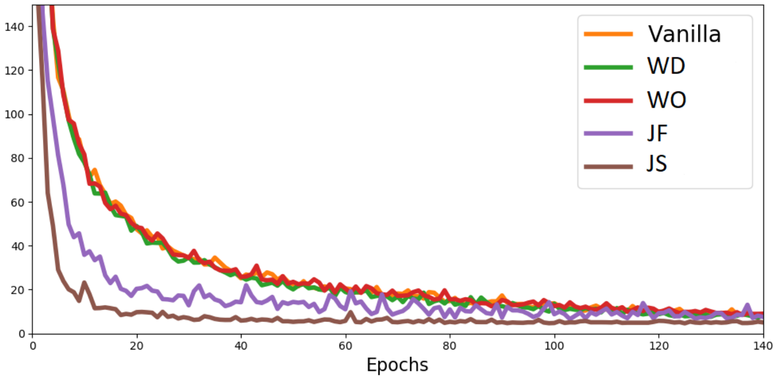


Figure 5.2: Sum of the spectral norms of the network input-output Jacobian evaluated on training samples $\sum_{i=1}^{m} \|\mathbf{J}(\mathbf{y}_i)\|_2^2$ versus number of training epochs. The neural network is trained using SGD under different regularizers, for the image denoising task.

and the spectral norm of the Jacobian of the composition of the neural network with the inverse problem forward operator, which incidentally also serves as an upper bound to the Lipschitz constants of these mappings, should improve the generalization ability of a neural network based inverse problem solver.

The use of Lipschitz regularization to improve the generalization ability of deep neural networks has already been recognized by various works [33]-[38]. However, the fact that introducing Lipschitz regularity in the end-to-end mapping involving the composition of the neural network and the inverse problem forward map may also control generalization does not appear to have been acknowledged in previous works pertaining to deep learning approaches to inverse problems. We therefore propose two model-aware regularization strategies:

Model-Aware Spectral Norm Based Regularization: Our first regularization strategy directly penalizes the operator norm of the Jacobians for the neural network and for the composition of the neural network and the forward map.Training in a minibatch stochastic gradient setup, where the optimization is carried out over minibatches $\mathcal{B} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{|\mathcal{B}|}, \mathbf{y}_{|\mathcal{B}|})\}$, leads to the following objective:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i) + \lambda_1 \max_{(\mathbf{y},\mathbf{x})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2 + \lambda_2 \max_{(\mathbf{y},\mathbf{x})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\|_2 \tag{5.2}$$

where λ_1, λ_2 are hyper-parameters. Note that $\lambda_2 = 0$ in a noise free setting. **Model-Aware Frobenius Norm Based Regularization:** Our second regularization strategy stems from the fact that the Frobenius norm upper bounds the Spectral norm. Regularisation strategies that punish the Frobenius norm of the network Jacobian have been associated with significant improvement in robustness of DNN classifiers [54, 38, 39]. Therefore, our cost function in (5.2) directly gives rise to the following objective function:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i) + \lambda_1 \max_{(\mathbf{y},\mathbf{x})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_F + \lambda_2 \max_{(\mathbf{y},\mathbf{x})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\|_F \tag{5.3}$$

We, however propose to regularize the following upper bound on (5.3) given by:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i) + \lambda_1 \sum_{i=1}^{|\mathcal{B}|} \|\mathbf{J}(\mathbf{y}_i)\mathbf{A}\|_F^2 + \lambda_2 \sum_{i=1}^{|\mathcal{B}|} \|\mathbf{J}(\mathbf{y}_i)\|_F^2 \tag{5.4}$$

This is mainly because the sum of square of the Frobenius norm results in simpler gradient computation. Additionally the regularization terms in (5.4) can be approximated in a computationally efficient setting as explained in the sequel.

Algorithm 1: Estimation of the $||\mathbf{JA}||_F^2$

```
Input: Mini-batch $\mathcal{B}$,number of projections $n$.
Output: Square of the Frobenius norm of the Jacobian $\mathcal{A}_F$.
$\mathcal{A}_F \leftarrow 0$
for $(\mathbf{y},\mathbf{x}) \in \mathcal{B}$ do
    $i \leftarrow 0$
    while $i < n$ do
        Initialize $\{\mathbf{z}\} \sim \mathcal{N}(0,\mathbf{I})$
        $\mathbf{z} \leftarrow \mathbf{z}/||\mathbf{z}||$
        $\mathcal{A}_F \leftarrow \mathcal{A}_F + p||vjp(f(\mathbf{y}),\mathbf{y},\mathbf{z}) \cdot \mathbf{A}||_2^2/(n|\mathcal{B}|)$
```

5.2.1 Efficient Computation of the Jacobian Based Regularizers

The challenge associated with the use of the training objectives in (5.2) and (5.4) relates to the computation of the Spectral norm and Frobenious norm of both **J** and **JA** because computing and storing the Jacobian matrix of deep neural networks incurs a huge cost. There are already computationally efficient algorithms to approximate the Frobenius and spectral norm of the Jacobian [39, 35]. Here, for completeness, we illustrate how to re-purpose these algorithms within our set-up; we also illustrate that these algorithms lead to efficient approximation.

5.2.1.1 Frobenious Norm Regularization of JA

The random projection based method proposed in [39] used to approximate the square of the Frobenius norm of the network Jacobian matrix **J** can be immediately extended to approximate the square of the Frobenius norm of the **JA** as shown in Algorithm 1. The technique leverages the reverse mode automatic differentiation to compute vector Jacobian product – the $vjp(\cdot,\cdot,\cdot)$ – of random vector sampled from the unit sphere of dimension $p-1$ with the network Jacobian. It has been shown in [39] that the proposed technique converges to the true value as $\mathcal{O}(n^{-1/2})$ where n is the number of random projections used for the estimation of the Frobenious norm. The algorithm when used for regularization, has also been shown to result in

Algorithm 2: Estimation of the spectral norm of **JA**

```
Input: Mini-batch $\mathcal{B}$, number of power iterations $n$.
Output: Maximum singular value, $\sigma$, of the matrix $\mathbf{JA}$.
for $(\mathbf{y},\mathbf{x}) \in \mathcal{B}$ do
    Initialize $\{\mathbf{u}\} \in \mathbb{R}^p$
    $i \leftarrow 0$
    while $i < n$ do
        $\mathbf{v} \leftarrow \mathbf{A}^T vjp(f(\mathbf{y}),\mathbf{y},\mathbf{u})$
        $\mathbf{u} \leftarrow jvp(f(\mathbf{y}),\mathbf{y},\mathbf{Av})$
        $i \leftarrow i+1$.
    $\sigma \leftarrow \|\mathbf{u}\|_2/\|\mathbf{v}\|_2$
```

Algorithm 3: Computation of the *jvp*.

```
Input: Mini-batch $\mathcal{B}$, model outputs $f(\mathbf{y})$, vector $\mathbf{Av}$.
Output: $\mathbf{JAv}$
Initialize a dummy tensor $\mathbf{d}$.
$\mathbf{g} \leftarrow vjp(f(\mathbf{y}),\mathbf{y},\mathbf{d})$
$\mathbf{u} \leftarrow vjp(\mathbf{g},\mathbf{d},\mathbf{Av})$
return $\mathbf{u}$
```

only an inconsequential overhead in compute requirements [39].

5.2.1.2 Spectral Norm Regularization of **JA**

In turn, the method in [35] used to approximate the spectral norm of the network Jacobian **J** can also be immediately re-purposed to approximate the spectral norm of **JA** as shown in Algorithm 2. The procedure leverages the power method [124] to approximate the spectral norm of the Jacobian based regularization terms in (5.2). It starts by choosing (randomly) an initial (nonzero) approximation of the left singular vector $\mathbf{u}$ in $\mathbb{R}^p$ associated with the highest singular value of the matrix **JA**. It then leverages the automatic differentiation to iteratively compute the Jacobian vector product and vector Jacobian product as follows:

$$\mathbf{v} \leftarrow \mathbf{A}^T \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right]^T \mathbf{u}, \qquad \mathbf{u} \leftarrow \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right] \mathbf{Av}$$

The spectral norm σ is then equal to $\|\mathbf{u}\|_2/\|\mathbf{v}\|_2$.

The algorithm exploits the reverse and forward mode automatic differ-

Table 5.3: Time and memory requirements for training a 4-layer fully connected NN and 5-layer CNN [1] on the full training set of MNIST with a batch size of 100 and $p = q = 784$.

	4-layer FC NN		5-layer DnCNN	
	time	memory	time	memory
SGD	29m	595Mb	1h8m	1057Mb
Alg. 2 ($n = 1$)	47.5m	659Mb	3h,42m	1825Mb
Alg. 2 ($n = 2$)	1h,1m	787Mb	5h,4m	2849Mb
Alg. 2 ($n = 3$)	1h,13m	787Mb	6h,31m	4897Mb
Alg. 2 ($n = 4$)	1h,18m	787Mb	9h,42m	4897Mb
tf batch J ($n = 3$)	63h,7m	4659Mb	−−	≈ 160Gb

entiation to compute the vector Jacobian product $vjp(\cdot,\cdot,\cdot)$, and the Jacobian vector products $jvp(\cdot,\cdot,\cdot)$ respectively. All major deep learning frameworks offer support for the computation of reverse mode vector Jacobian product. The forward mode Jacobian vector product can easily be computed via the reverse mode automatic differentiation using the method described in Algorithm 3 [125][1].

Note again that the merit of Algorithms 1 and 2 lies in computing the Frobenius and spectral norms of Jacobians without explicitly computing the Jacobians themselves that is prohibitive in high-dimensional settings.

5.2.1.3 Algorithm Accuracy and Complexity

We now study the efficacy offered by Algorithm 2 via a simple experiment involving the reconstruction of MNIST data from its noisy version. We generate the noisy MNIST data by passing the clean data through the linear model in (4.1) with the forward operator set to be equal to an identity one. We also further contaminate the MNIST data with a noise sampled uniformly from a ℓ_2-sphere of radius 0.3. We then reconstruct the data from the noisy version using two neural networks, a 4-layer fully connected neural network and a 5-layer convolutional neural network. These networks are trained using ADAM optimizer for 300 epochs using the ℓ_2 loss function in (4.5).

Our experiments have two main goals:

[1] The spectral norm of the Jacobian matrix can be computed by directly substituting **A** with an Identity matrix. A similar technique has been proposed in [35].

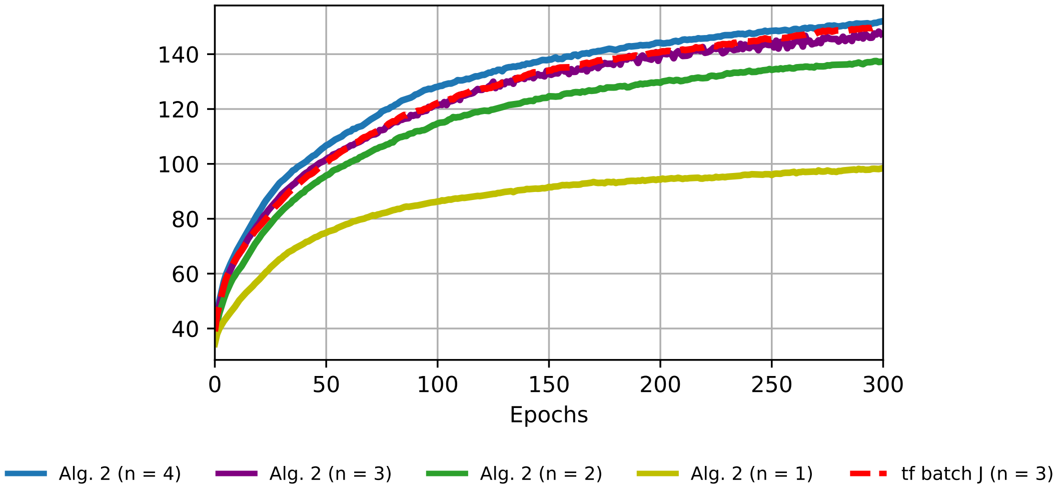


Figure 5.3: Maximum singular values of the batch Jacobians for a 4-layer fully connected network with $p = q = 784$.

1. First, we want to test that Algorithm 2 indeed results in a faithful estimate of the spectral norm of the network Jacobian. To this end, we compare the output of the Algorithm 2 with the spectral norm computed using power method applied to a Jacobian matrix explicitly computed using Tensorflow. It can be seen in Fig. 5.3 that for equal number of power iterations ($n = 3$) the results obtained using both methods are almost identical.

2. Second, we want to quantify the computational benefit afforded to us by Algorithm 2 – owing to its implicit matrix vector products computation – in contrast to estimating the spectral norm via explicit matrix vector products. In particular, Table 5.3 compares computation and memory requirements of the algorithm against alternatives associated with the training of both the fully connected and the convolutional neural networks. It can also be seen that our algorithm provides considerable gains in relation to the alternatives.

In summary, both for fully connected and convolutional neural networks, our experiments suggest that regularizing the network using Algorithm 2, offers considerable computational gains in comparison to direct computation of the spectral norm. In fact, the explicit computation of the network Jacobian would be practically impossible even for a modestly sized network. For example, for convolutional neural networks, even a minibatch Jacobian of 10 samples occupies 16GB of memory making it infeasible to approximate any norm. In contrast, with Algorithm 2 both *jvp* and *vjp* can be computed approximately in linear time using most major deep learning frameworks.

5.3 Experiments

We now conduct a series of experiments in order to assess the efficacy of our proposed model-aware deep learning regularization strategy on range of popular inverse problems. These include (a) the reconstruction of images from low-dimensional Gaussian measurements (b) the generation of high-resolution images from a low-resolution version. These various inverse problems involve different measurement operators, exhibiting different condition numbers, enabling us to verify the merit of our proposed regularizers under various settings.

5.3.1 Image Reconstruction in the Presence of Gaussian Measurements

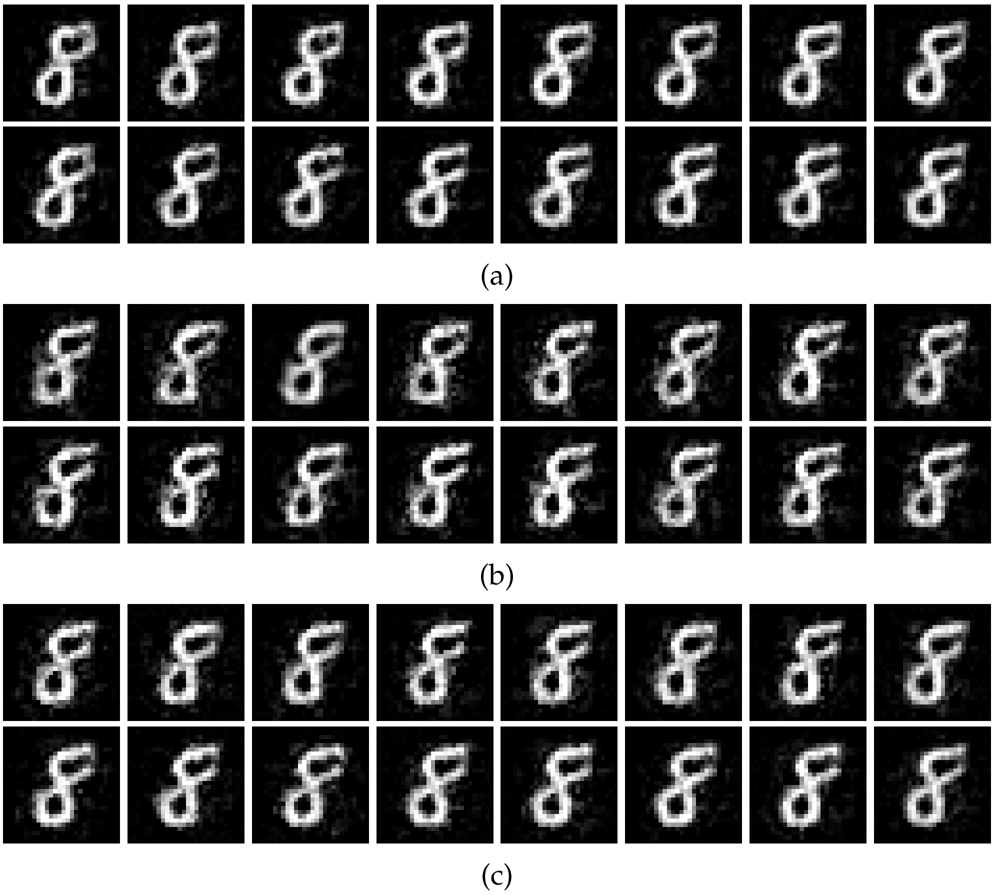
5.3.1.1 Experimental procedure

Our first set of experiments involves the reconstruction of images from noisy compressive Gaussian measurements. In particular, we consider our linear model in (4.1) where $\mathbf{A}$ is a (wide) random Gaussian matrix with i.i.d. entries sampled from a Gaussian distribution with mean zero and variance $1/q$ and the noise is sampled uniformly from a sphere of radius η. We consider 28×28 greyscale images of handwritten digits taken from the MNIST dataset [126]. We construct a dataset $\{\mathbf{y}_i, \mathbf{x}_i\}_{i=1}^{n}$ whereby the q-dimensional measurement vector $\mathbf{y}_i$ is obtained from the p-dimensional vector $\mathbf{x}_i$ – which is derived by converting a 28×28 greyscale image onto a 784 dimensional vector – via the linear model in (4.1). We also scale the pixel values in the images to the range $[0,1]$ prior to the application of the linear operator.

For the reconstruction of the original images from the noisy compressive measurements, we consider a 4-layer fully connected neural network consisting of an input layer of width equal to the measurement size – q, followed by three layers, each containing neurons equal to the dimension of the ground truth – p. All the layers except the last one have an associated Rectified Linear Unit (ReLU) activation function.

The reconstruction network is trained using the ADAM optimizer for 600 epochs using various regularization strategies. These strategies include: (a) model aware Spectral norm regularization of Jacobian in (5.2) which is denoted by SJA&SJ ($\lambda_1, \lambda_2 > 0$) or only SJA ($\lambda_1, \lambda_2 = 0$); (b) model-aware Frobenius norm regularization in (5.4) which is denoted by FJA&FA ($\lambda_1, \lambda_2 > 0$) or only FJA ($\lambda_1, \lambda_2 = 0$); and (c) model agnostic regularization approaches such as weight decay (WD), spectral norm regularization of weights (SW) [28], Spectral norm regularization of Jacobian (SJ) and Frobenius norm regularization of Jacobian (FJ) [39]. Note that comparing our regularization strategies with WD, SW, SJ and FJ will allow us to assess the benefits of model-aware regularization since WD, SW, SJ and FJ do not take into account the presence of the linear operator. The regularization parameters appearing in the various strategies (including λ_1 and λ_2 for our regularizers) are always fine-tuned using a grid search method.

To assess the efficacy of the proposed regularizers on inverse problems with different levels of ill-posedness and corruption, we conduct various experimental studies. Specifically, we look at the performance of networks trained under different regularized loss functions when q is varied such that it takes values in the set $\{80, 160, 320, 640\}$ for $m = 500$ and η fixed at 0.3. Likewise, we also observe the performance of different regularizers when the noise level η is gradually increased from 0 to 0.6 while keeping m and q fixed at 500 and 160 respectively. Finally we also gauge how different regularizers behave under the training sets of size 200, 400, 600 and 800 while keeping q fixed at 160 and $\eta = 0.3$.



(a)

(b)

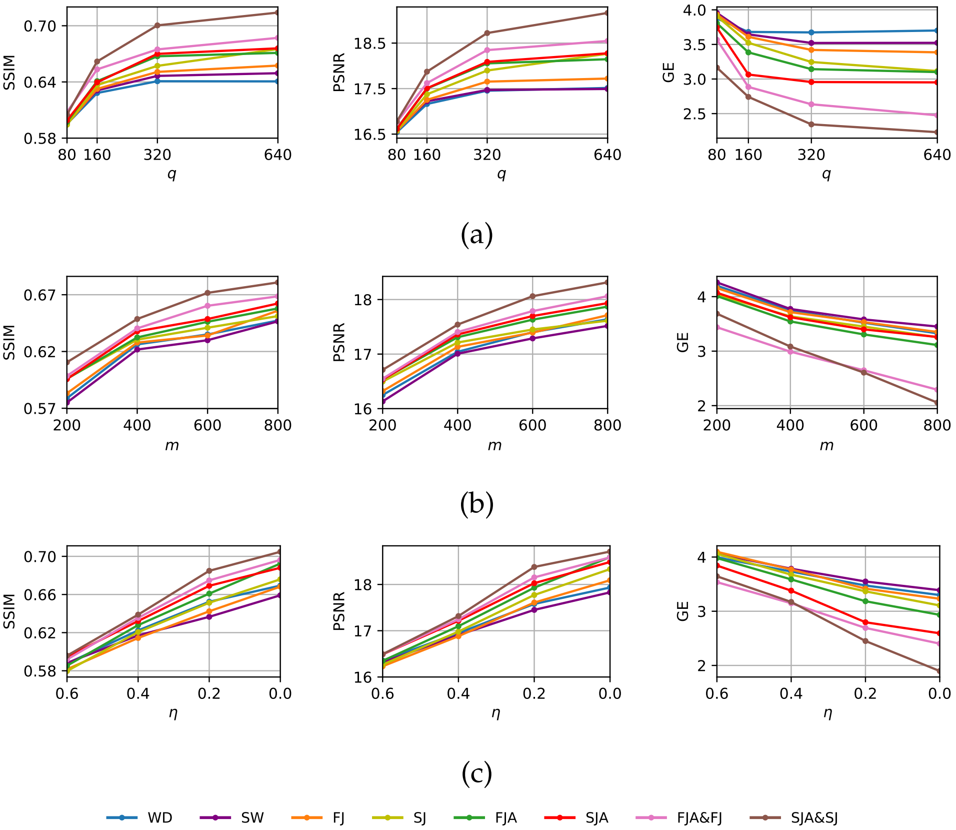
(c)

Figure 5.4: Sample results from the reconstruction from compressed Gaussian measurements using a 4-layer FC neural network regularized with WD, SW, FJ, SJ, FJA, SJA, FJA&FJ and SJA&SJ. (5.4a) $\eta = 0.3$ $m = 500$ and $q = 320$(top row), $q = 640$(bottom row). (5.4b) $\eta = 0.3$, $q = 160$ and $m = 600$ (top row), $m = 800$ (bottom row). (5.4c) $m = 500$, $q = 160$ and $\eta = 0.2$ (top row), $\eta = 0$ (bottom row).

The reconstruction performance of the various regularization schemes is compared in terms of visualizations and various quality metrics such as the generalization gap determined using the generalization error in eq. (4.4) and other quality metrics such as Structural Similarity Index (SSIM) and PSNR.

5.3.1.2 Results

Fig. 5.5 and 5.4 present a performance comparison of networks regularized with our model aware Jacobian regularizers and the baseline techniques for various training scenarios.



(a)

(b)

(c)

Figure 5.5: Reconstruction of MNIST images given Gaussian compressive measurements using a fully connected neural network. (5.5a) (Left) SSIM versus number of Gaussian measurements, (Centre) PSNR versus number of Gaussian measurements, (Right) GE versus number of Gaussian measurements for various regularization strategies such that $\eta = 0.3$ and $m = 500$. (5.5b) (Left) SSIM versus number of training examples, (Centre) PSNR versus number of training examples, (Right) GE versus number of training examples for various regularization strategies such that $\eta = 0.3$ and $q = 160$. (5.5c) (Left) SSIM versus noise level, (Centre) PSNR versus noise level, (Right) GE versus noise level for various regularization strategies such that $m = 500$ and $q = 160$.

In Fig. 5.5a, we plot the test set SSIM, PSNR and GE of the reconstructed MNIST images versus the number of measurements q. It can be seen that our proposed model-aware strategies lead to performance gains in comparison with existing ones, where the gains are more pronounced with the increase in the number of measurements. This shows that – owing to the explicit exploitation of the forward map – model aware regularizers are better able to leverage the additional measurements. The generalization error between the training and the test set for different measurement sizes also shows a similar trend with model aware regularizers consistently outbeating the competing baseline techniques. A visual comparison of the quality of the reconstructed images under different number of measurements is also offered in Fig. 5.4a. It is clear that the images recovered with model aware regularizers are perceptually far more refined in comparison with the ones reconstructed using the model agnostic techniques.

In Fig. 5.5b, we plot the test set SSIM, PSNR and GE of the reconstructed MNIST images versus number of training examples. Here again, the regularizers that incorporate the knowledge of the forward map outperform the regularization techniques that do not. This result also reinforces the hypothesis that even in situations where we may only have small number of training examples, model-aware regularization can result in a better generalization performance. The gain in performance metrics also translates to the superior reconstruction as depicted in Fig. 5.4b

Finally, in Fig. 5.5c, we study the effect of different regularizers in the presence of different levels of noise. For measurements with high noise levels, the model agnostic and model aware regularizers show similar performance. This is because in low SNR conditions the effect of noise may dominate the effect of the forward operator. In contrast, for measurements with low levels of noise, model aware regularizers show superior performance to the existing model agnostic ones. The GE plot for these experiments again shows that the proposed regularizers results in superior generalization behaviour when the noise levels are low. This effect can be visually observed in Fig. 5.4c which shows sample reconstructions using the competing regularizers.

It should be noted that SJA&SJ consistently outperforms FJA&FJ. This is because Frobenious norm regularization minimizes the sum of squres of all the elements in the matrix – not taking into account the correlation between the rows of the Jacobian – and thus is more restrictive than the Spectral norm regularization.

These results support our analysis that model induced regularizers improve the performance of neural networks over model agnostic regularization translating into better reconstructions.

5.3.2 Image Super-resolution

5.3.2.1 Experimental procedure

We now study the performance of our regularizers on the classical super resolution (SR) problem involving the recovery of high resolution images from their low resolution versions. The SR problem can be mathematically formulated via the linear model in (4.1) where $\mathbf{n}$ represents the measurement noise and the forward operator $\mathbf{A}$ can be defined as the product of a blur matrix $\mathbf{H} \in \mathbb{R}^{p \times p}$ and a subsampling matrix $\mathbf{L} \in \mathbb{R}^{p \times q}$. The point spread function (PSF) of the matrix $\mathbf{H}$ can be uniform, Gaussian or bicubic and is assumed to be known in advance [127]. In our experiments we sample the noise uniformly from a sphere of radius η and assume the PSF to be a 5×5 Gaussian kernel. For our training procedure, we sample images from the BSD300 database [128]. The dataset $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{m}$ is generated by obtaining $128 \times 128 \times 3$ cutouts from these images; vectorizing them; and then obtaining the q-dimensional measurement $\mathbf{y}_i$ via the linear model in (4.1). The exact value of q depends on the subsampling ratio p/q. We test our regularizers for subsampling ratios of 2 and 4. The training set size, m, is fixed to 500 samples, while the regularization parameters are tuned on a validation set of size 3500. We also apply an adaptive policy taking into account feedback from training in order to tune the regularization parameters – as opposed to keeping them fixed – using the approach summarized in Algorithm 4 [2].

To reconstruct the high resolution (HR) images from the low resolution

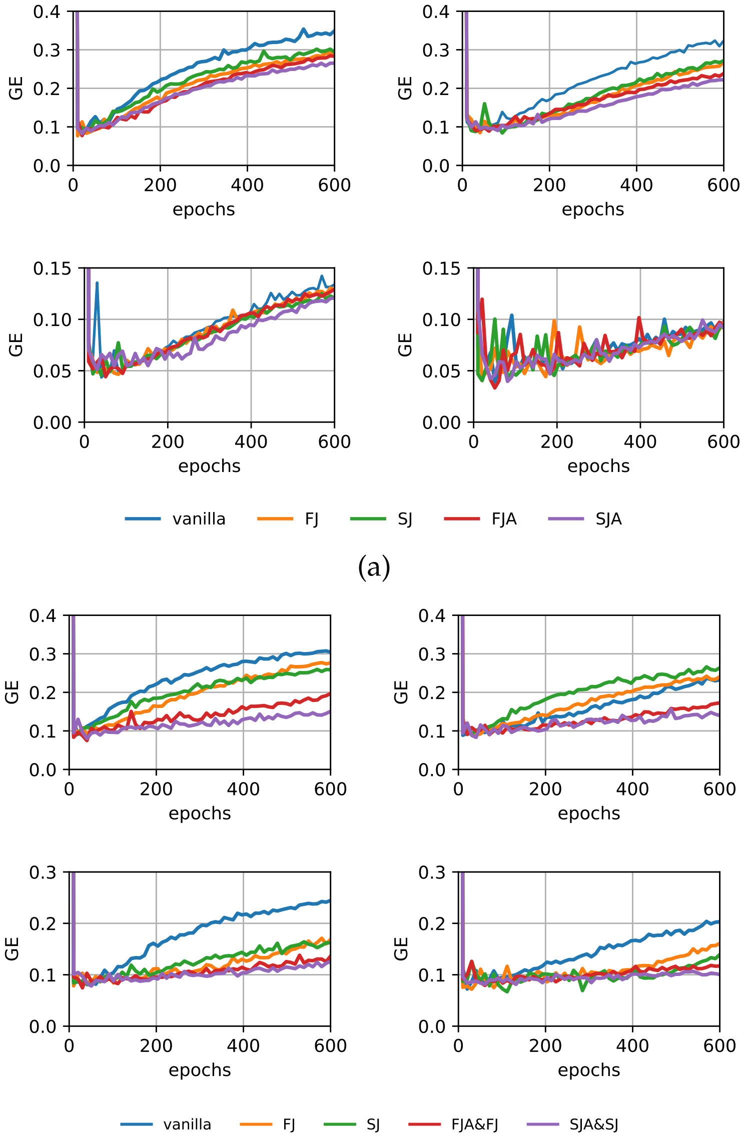
[2] Our empirical results show that using such an adaptive technique results in better validation performance and lesser training time. Since this technique takes into account the training performance to compute the regularization coefficients at each step and does not rely on a hit and trial method to find the 'best' hyperparamter, it results in lesser overall training time for the model.

Table 5.4: Reconstruction performance of EDSR and WDSR on various test datasets.

			$p/q = 2$						$p/q = 4$					
			Set5		Set14		Urban100		Set5		Set14		Urban100	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	$\eta = 0$	Vanilla	32.63	0.902	29.57	0.849	27.44	0.873	25.87	0.701	24.05	0.655	21.43	0.629
		FJ	32.98	0.913	29.96	0.868	27.92	0.888	26.61	0.731	24.81	0.680	22.28	0.663
		SJ	33.32	0.916	30.33	0.861	28.18	0.885	26.33	0.723	24.54	0.671	22.00	0.650
		FJA	33.05	0.913	30.01	0.862	27.91	0.884	26.65	0.729	24.84	0.675	22.29	0.659
		SJA	**33.60**	**0.922**	**30.66**	**0.876**	**28.68**	**0.910**	**26.90**	**0.745**	**24.97**	**0.685**	**22.37**	**0.665**
	$\eta = 3$	Vanilla	28.60	0.833	26.01	0.744	23.04	0.716	24.75	0.654	23.09	0.575	20.42	0.522
		FJ	28.68	0.822	26.23	0.738	23.35	0.720	24.78	0.656	23.12	0.574	20.47	0.522
		SJ	28.72	0.823	26.14	0.737	23.18	0.709	24.98	0.677	23.26	0.592	20.68	0.542
		FJA+FJ	29.34	0.848	26.73	0.759	23.98	0.751	25.38	0.680	23.59	0.599	20.80	0.547
		SJA+SJ	**29.50**	**0.849**	**26.95**	**0.768**	**24.00**	**0.753**	**25.50**	**0.692**	**23.61**	**0.603**	**20.83**	**0.551**
WDSR	$\eta = 0$	Vanilla	33.76	0.921	30.17	0.869	28.49	0.893	26.19	0.739	24.08	0.670	21.52	0.648
		FJ	34.20	0.925	30.57	0.877	28.76	0.900	26.80	0.757	24.75	0.685	22.11	0.663
		SJ	34.05	0.928	30.69	0.880	28.88	0.902	26.77	0.752	24.71	0.686	22.10	0.662
		FJA	34.25	**0.929**	30.66	0.881	28.94	0.906	27.00	**0.765**	24.85	0.689	**22.22**	**0.669**
		SJA	**34.55**	0.927	**30.79**	**0.881**	**29.01**	**0.905**	**27.06**	0.762	**24.89**	**0.690**	22.20	0.665
	$\eta = 3$	Vanilla	28.63	0.827	26.20	0.751	23.27	0.730	24.95	0.666	23.23	0.588	20.53	0.532
		FJ	29.15	0.841	26.56	0.759	23.85	0.749	25.34	0.699	23.35	0.598	20.74	0.549
		SJ	29.31	0.842	26.64	0.760	23.88	0.752	25.36	0.704	23.39	0.604	20.80	0.558
		FJA+FJ	29.73	0.853	26.87	0.762	24.15	0.757	25.52	0.708	23.60	0.607	20.91	0.563
		SJA+SJ	**29.91**	**0.858**	**27.19**	**0.858**	**24.37**	**0.769**	**25.54**	**0.711**	**23.65**	**0.612**	**20.93**	**0.565**

(LR) measurements, we train two state-of-the-art ResNet architectures – the Enhanced Deep Residual Networks (EDSR) [2] and the Wide Activation Residual Networks (WDSR) [3]. These architectures have specially been designed for solving the SR problem leading up to exceptional performance on various datasets and SR challenges. We train these networks using the ADAM optimizer for 600 epochs. In these set of experiments, we compare the performance of the proposed model aware regularizers in eqs. (5.2) and (5.4) against their model agnostic counterparts; we also demonstrate with the help of generalization error curves how incorporating the knowledge of the forward operator also induces generalization gains.

In order to evaluate the impact of incorporating the knowledge of the forward operator in the proposed regularizers, we test the performance of our trained networks on various publicly available datasets such as Set5, Set14 and Urban 100 dataset. The reconstruction performance of the various



(a)

(b)

Figure 5.6: Generalization error Vs number of epoch plots for the SR problem using different regularization strategies. (5.6a) GE plots for EDSR (left) and WDSR (right) when $\eta = 0$. (Top) $p/q = 4$ (Bottom) $p/q = 2$ SR task. (5.6b) GE plots for EDSR (left) and WDSR (right) when $\eta = 3$. (Top) $p/q = 4$ (Bottom) $p/q = 2$.

schemes is compared in terms of visualizations and quality metrics such as GE, Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR).

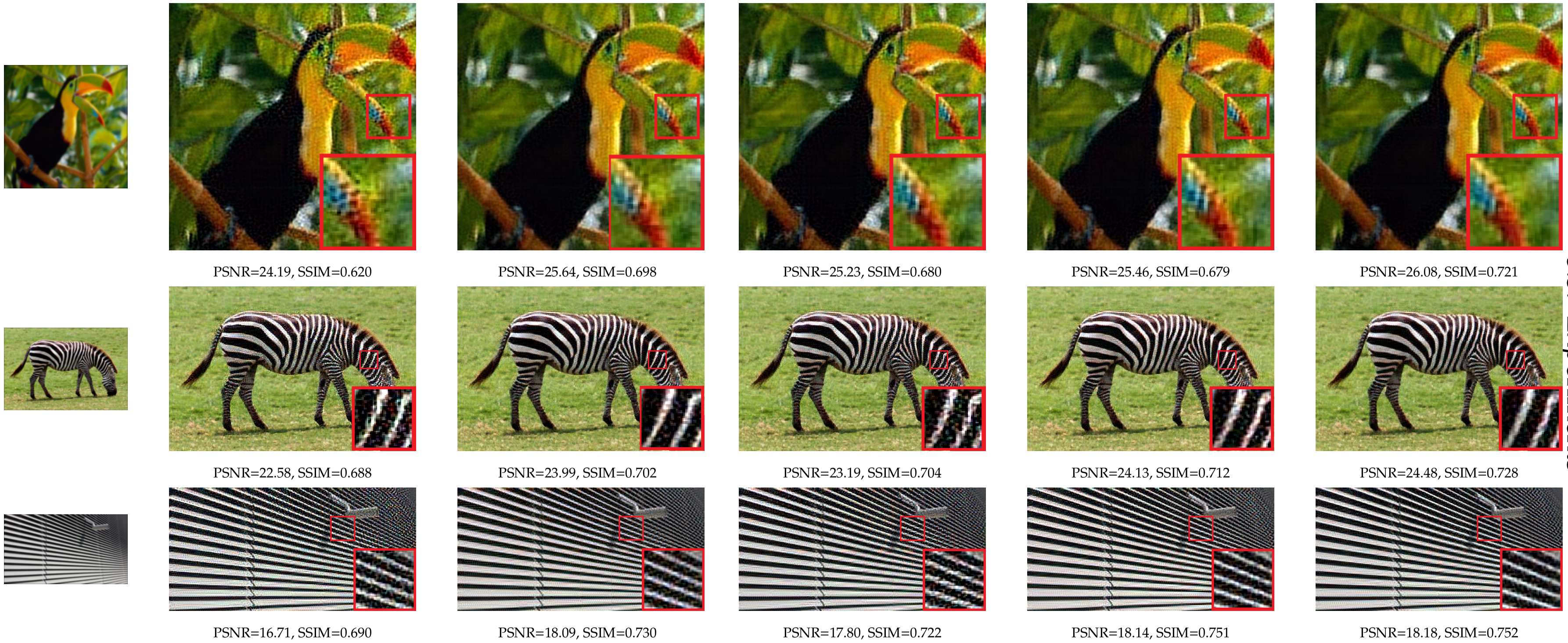


Figure 5.7: Sample reconstruction results for the image SR task under a noiseless setting ($\eta = 0$) recovered using EDSR [2]. (Left to Right) vanilla, FJ, SJ, FJA and SJA.

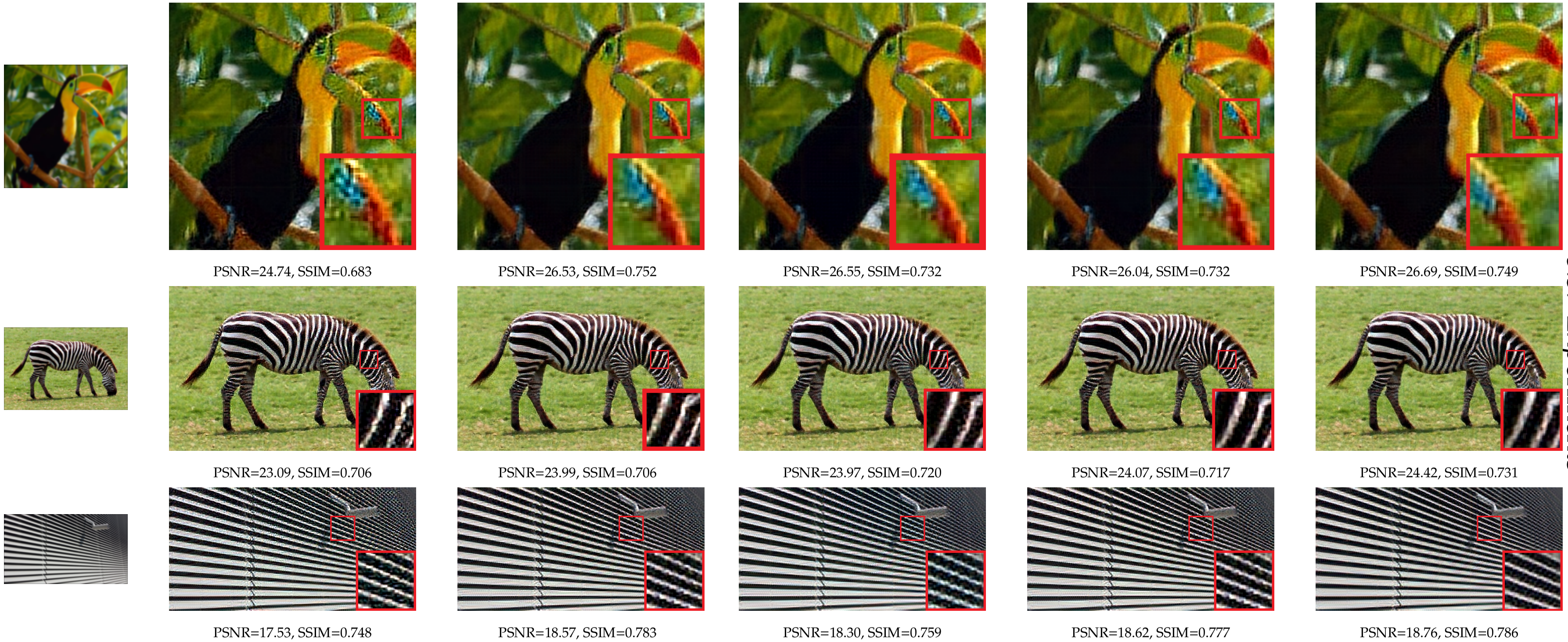


Figure 5.8: Sample reconstruction results for the image SR task under a noiseless setting ($\eta = 0$) recovered using WDSR [3]. (Left to Right) vanilla, FJ, SJ, FJA and SJA.

5.3.2.2 Results

In order to investigate the improvement in the generalization behaviour induced by the proposed model aware regularizers, we have plotted the GE between the training and validation set – computed via eq. (4.4) for solving the SR tasks in Fig. 5.6. It can be seen in Figs 5.6a and 5.6b that for the subsampling ratio of $p/q = 4$, the regularization methods which incorporate the knowledge of the forward operator outperform the regularization techniques that do not. Our results for the noise free SR problems when $p/q = 2$, – presented in Fig 5.6a do not exhibit significant gaps in the generalization performance. This is expected since for $p/q = 2$, SR is a comparatively easier recovery problem and therefore exploiting the knowledge of the forward model may not provide much benefit. However, in the presence of noise, regularizing the networks shows improved generalization performance even for $p/q = 2$, as shown in Fig. 5.6b. These results validate our theory that model aware regularization techniques induce performance gains by reducing the effect of overfitting – resulting in a better generalization behaviour.

Fig. 5.7 and 5.8 present a visual comparison of the outputs achieved using model aware Jacobian regularizers and the baseline techniques on various datasets. It can be seen that for both EDSR and WDSR, our propsed regularized leads to perceptual gains in contrast to the standard (unregularized) training. Although the model agnostic regularization techniques also result in improved visualizations, a close inspection of the recovered images reveals that the model aware regularizers are able to recover finer image details. It should be noted that the reconstruction results are achieved with only 500 training samples.

Finally, in Table 5.4, we demonstrate the effectiveness of the proposed regularizers on various out of sample datasets. On the $p/q = 2$ SR task – in comparison to vanilla training for both the EDSR and WDSR – the model aware regularization techniques result in a gain of up to 1.24 dB and 0.04 in terms of the PSNR and SSIM respectively. The proposed model aware regularizers also show improvement over their model agnostic counterparts. A similar trend can be observed in the $p/q = 4$ SR task where model aware regularizers achieve a performance gain of upto 0.97dB and 0.036 in terms of PSNR and SSIM respectively . The performance improvement over the vanilla training in the WDSR network are slightly less pronounced than EDSR but still noticeable. This is because WDSR is a more competetive network than EDSR.

These results support our analysis that model induced regularizers improve the performance of neural networks over model agnostic regularization translating into better reconstructions.

5.4 Summary

In this chapter – motivated by our analysis in Chapters 3 and 4 – we have proposed new regularization strategies that penalize the spectral norm of the Jacobian matrices of the relevant mappings. Our experiments on both classification and regression problems show that the proposed regularization strategies outperform the current state of the art ones.

Chapter 6

Case Studies

A number of biomedical problems require the investigation of human organs via images obtained using non-invasive techniques such as Magnetic Resonance Imaging (MRI) or different types of tomographic approaches. A common feature in these diagnostic techniques is that the obtained measurements – in their raw form – are noisy and usually belong to modalities other than imaging. Although the acquisition models for these inverse problems are usually well postulated – the under-determined nature of data renders them ill-posed in the sense of Hadamard [8] i.e., the solution either does not exist, is not unique or is not stable with respect to measurements. This lends the retrieval of meaningful information a non-trivial task. Therefore, appropriate regularization or constraints reflecting the domain knowledge are necessary to obtain reliable solutions.

A variety of iterative, funtional analytic and data-driven methods have been shown to perform well on these biomedical imaging problems. In this chapter, we evaluate the effectiveness of the proposed model Jacobian regularization proposed in Chapter 5 on a few clinically significant inverse problems and compare them with state of art methods in literature.

6.1 Magnetic Resonance Imaging

Our first set of experiments involve the reconstruction of MRI images from sub-sampled Fourier measurements. The frequency domain sub-sampling

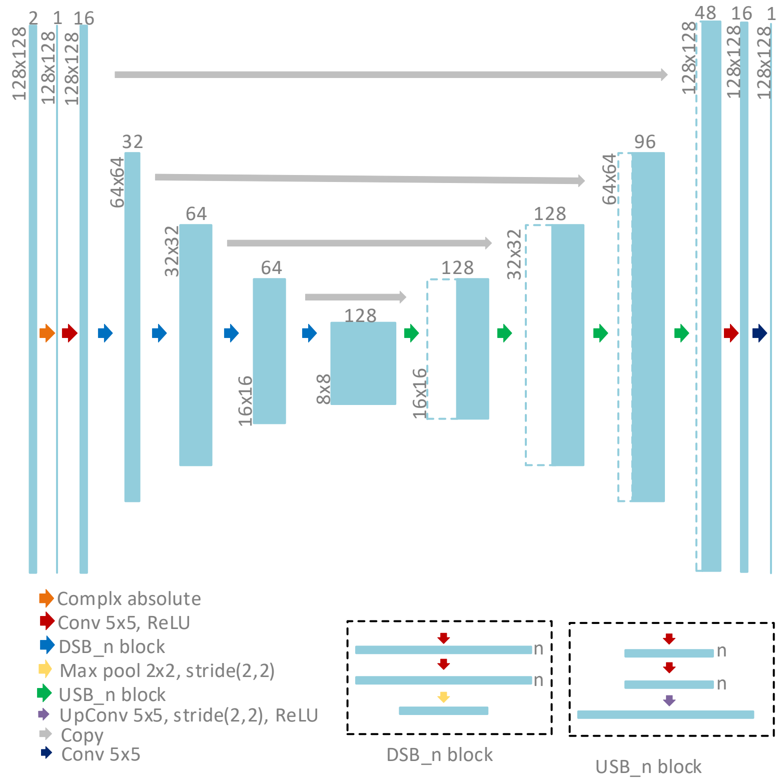


Figure 6.1: UNet Architecture.

operator can be mathematically represented as

$$\mathbf{A} = \mathbf{F}^{-1}\mathbf{M}\mathbf{F}$$

where $\mathbf{F}$ and $\mathbf{F}^{-1}$ are the Fourier and inverse Fourier transform matrices. The mask $\mathbf{M}$ is diagonal matrix containing binary entries on its diagonal where the number of non-zero entries signify the sampling density s. These types of forward mappings are particularly important for applications such as accelerated MRI reconstruction where the field of view is scanned by obtaining sparse measurements in k-space domain leading to reduced MRI acquisition periods.

Our linear model is also such that the noise for each sample in (4.1) is sampled uniformly from an ℓ_2 sphere of radius η. We construct our dataset by retrospectively under-sampling the Fourier transform of the ground truth

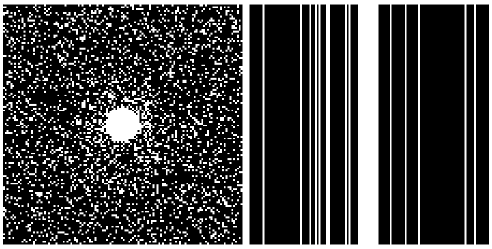


Figure 6.2: k-space acquisition masks (Left): Random 2D 5-fold subsampling mask with the centre fully sampled.(Right) Random 1D 4-fold subsampling mask.

images, obtained from the NYU fastMRI's knee database [6]. The subsampling is achieved by the Cartesian 1D and 2D random sampling masks in k-space, shown in Fig. 6.2 – retaining only 25% and 20% of the total Fourier samples, respectively. We also normalize the images to the range $[0,1]$ before applying the forward transform and adding noise with level $\eta = 5$. The training was achieved using a set of 500 samples and a minibatch of size 5.

We consider the state-of-the-art UNet architecture [129] under different regularization strategies (including SJA&SJ and FJA&FJ) to reconstruct the original images from the noisy under-sampled Fourier measurements. A schematic of the network architecture is shown in Fig. 6.1. This network is trained using the ADAM optimizer for 300 epochs using the different regularization strategies. However, we only apply the regularization in 10% of the steps per epoch in order to speed up the optimization. We also apply an adaptive policy taking into account feedback from training in order to tune the hyperparameters λ_1 and λ_2 – as opposed to keeping them fixed – using the approach summarized Algorithm 4. [1]

We also consider, for comparison purposes, competing techniques such as (a) wavelet sparsity regularized reconstruction [92]; (b) adversarial regularizers [4]; and (c) postprocessing via UNet method [5]. For a fair comparison, the UNet architecture and training routines are kept the same for our

[1] Our empirical results show that using such an adaptive technique results in better validation performance and lesser training time. Since this technique takes into account the training performance to compute the regularization coefficients at each step and does not rely on a hit and trial method to find the 'best' hyperparamter, it results in lesser overall training time for the model.

Algorithm 4: Estimation of the regularization coefficient λ for Jacobian regularizer.

```
Input: magnitude $r$ of the regularization term and $l$ of the loss over
       Mini-batch $\mathcal{B}$, scaling factor $s$
Output: Value of the regularization coefficient
$\alpha \leftarrow \text{floor}(\log(l/r))$ ;        // $l$ is the unregularized empirical loss
  $1/|\mathcal{B}| \sum_i l(f_{\mathcal{S}},(\mathbf{x}_i,\mathbf{y}_i))$
$\lambda \leftarrow 10^{\alpha}/s$ ;   // The values of 10,20 and 30 were tested for $s$.  20
  usually gave the best results.
```

Table 6.1: Comparison of different SOTA approaches used to reconstruct the MRI measurements obtained from the NYU fastMRI knee dataset [6] under PSNR (dB) and SSIM.

	2D mask ($s = 0.2$)		1D mask ($s = 0.25$)	
	PSNR	SSIM	PSNR	SSIM
Wavelet sparsity reg	28.49	0.72	24.56	0.50
Adversarial Regularizer [4]	29.89	0.77	25.44	0.54
UNet as post-processor [5]	30.01	0.79	28.36	0.74
UNet w FJA&FJ	30.80	0.80	28.96	0.75
UNet w SJA&SJ	30.89	0.81	29.30	0.78

work and the postprocessing method. Note also that unlike [5], we train the post processing UNet on ℓ_2 loss function. For the Adversarial regularization method, we modify the official implementation of the adversarial regularizer, present on Github [4], provided by the authors of the publication to suit the forward model used in this work. The batch size and other hyperparameters such as the step size and the choice of the adversarial regularizer network were kept the same as in the original implementation. Both the postprocessing and the adversarial regularization method involve a 'preprocessing' step. That is, both techniques obtain an initial course estimate of the signal of interest by applying a classical regularized reconstruction method, $\mathbf{A}^{\dagger}(\cdot)$ to the measurement $\mathbf{y}$. For our experiments, we use the output of the wavelet sparsity regularized reconstruction method as this initial estimate.

We compare once again the reconstruction performance of the various approaches in terms of visualizations and quality metrics such as Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR).

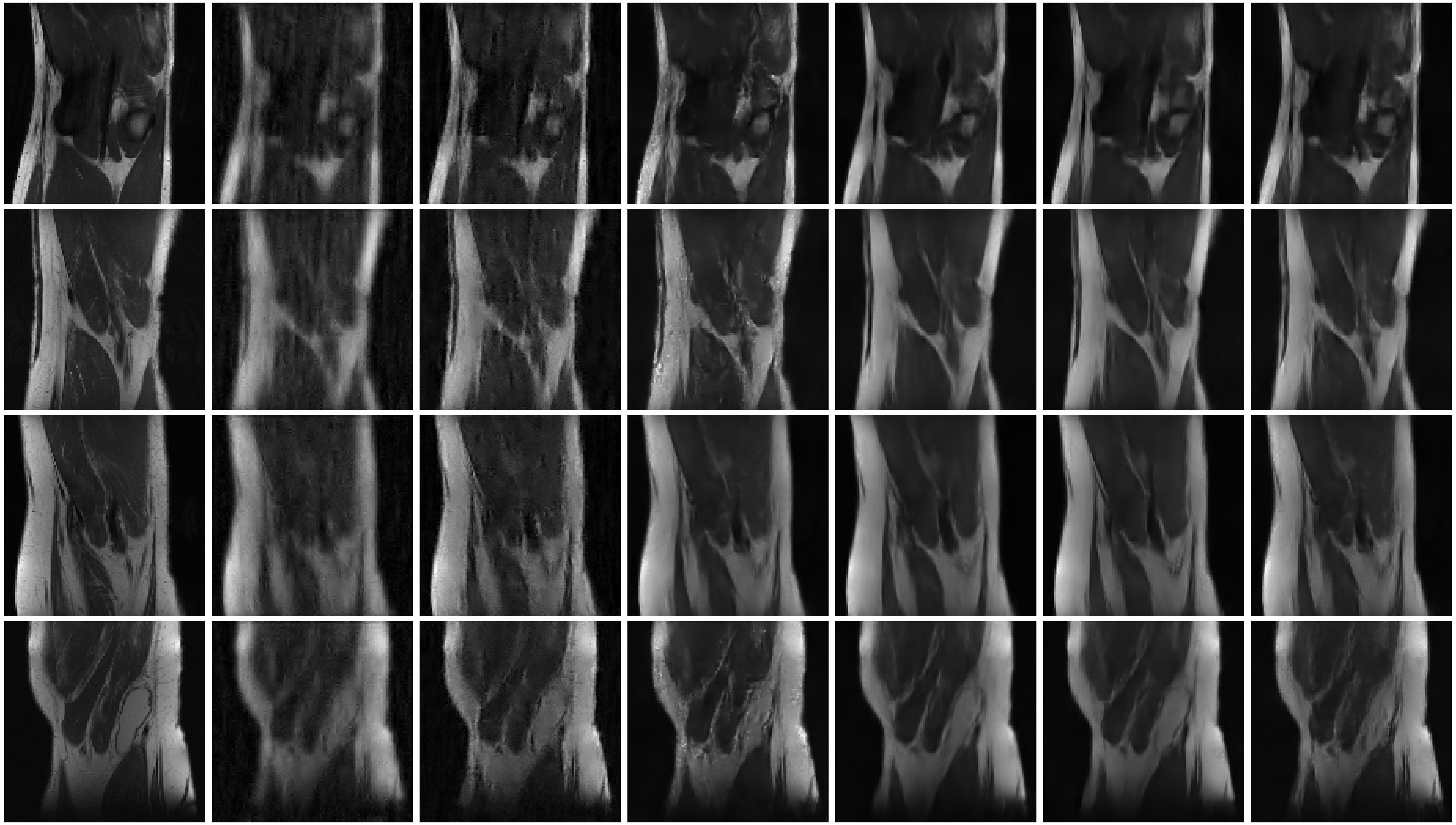


Figure 6.3: Sample results from the reconstruction from k-space subsampled measurements acquired using a 2D acquisition mask in Fig. 6.2. (Left to Right) ground truth, reconstruction using Wavelet sparsity regularization, Adversarial Regularizer [4], postprocessing using UNet [5], UNet with FJA&FJ and UNet with SJA&SJ, postprocessing using UNet [5] regularized with SJA&SJ.

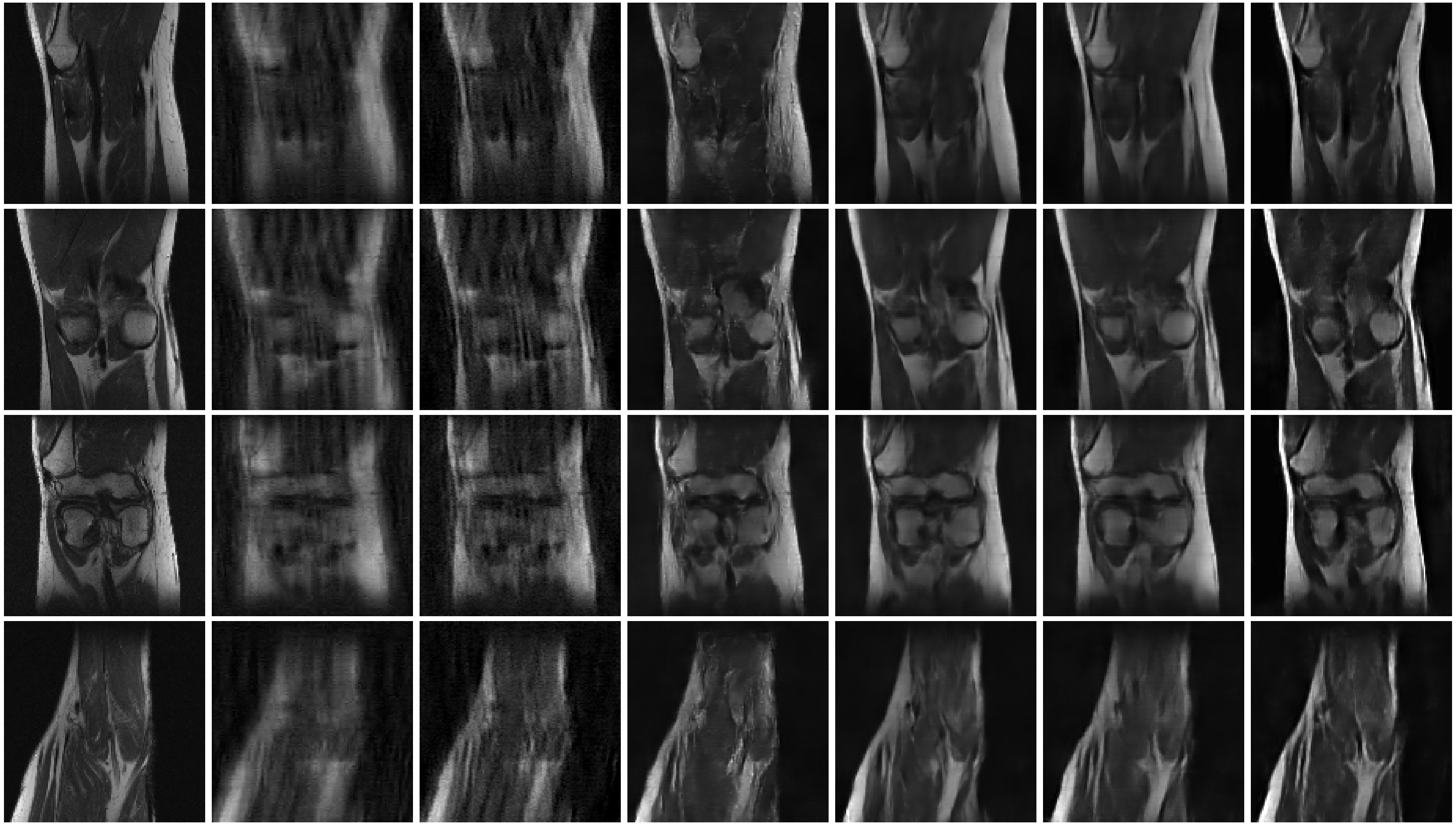


Figure 6.4: Sample results from the reconstruction from k-space subsampled measurements acquired using a 1D acquisition mask in Fig. 6.2. (Left to Right) ground truth, reconstruction using Wavelet sparsity regularization, Adversarial Regularizer [4], postprocessing using UNet [5], UNet with FJA&FJ and UNet with SJA&SJ, postprocessing using UNet [5] regularized with SJA&SJ.

6.1.1 Results

Table 6.1 compares the performance of the different reconstruction approaches. The proposed regularizers consistently outbeat all the other methods in terms of PSNR and SSIM. The performance gains are more pronounced for 1D sampling mask in view of the fact that these introduce aliasing artifacts that appear to be better dealt with our approaches in relation to competing ones.

Figs. 6.3 and 6.4 in turn offer a visual comparison of the quality of the reconstructed images for the different approaches. It can be seen that our proposed regularization approaches appear to lead to better reconstruction quality in relation to the competing methods. Since the Jacobian regularization method can be directly used with any deep learning based reconstruction method, we also include reconstruction results when a post-processing UNet is regularized via SJA&SJ regularizer. It can be seen that perceptually the reconstruction achieved through this method outperforms all the other techniques. However there is no improvement in terms of PSNR and SSIM over the UNet with SJA&SJ (without the preprocessing). A close inspection of the reconstructed images reveals that the proposed method introduces less artifacts than the other reconstructions. This is specially important for data available in medical applications, since any artificial noise artificant introduced as a result of the reconstruction process can severely hinder the diagnostic process.

6.2 Computed Tomography

In this section, we demonstrate performance of our model aware regularizer (5.2) on the recovery of the desired image from tomographic 2D parallel beam measurements. An important application of these problems is the Computed Tomography (CT) from X-ray beams which involves recovering the signal of interest by observing the attenuation patterns of X-rays passing through a body. The underlying physics of a CT problem can compactly be represented via the standard formulation of the inverse problem (4.1) :

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

where the forward operator $\mathbf{A}$ is the ray transform [130] and $\mathbf{n}$ represents the degradation associated with the acquisition setup such as metal and ring artifacts.

For our experiments, we fix $\mathbf{A}$ to be an under-sampled ray transform with 30 parallel beam projections. The latent vector $\mathbf{x}$ is sampled from the publicly available LIDC/IDRI database containing lung scans [131]. The dataset is created by resizing the images to the dimensions of 128×128 prior to the application of the forward operator. The transformed data is then corrupted with white Gaussian noise[2]. We utilize 500 samples for training, 200 for validation and the trained network is tested on 500 test data points.

We test the performance of our regularizer on the UNet architecture and tune the regularization coefficients using the adaptive approach given in Algorithm 4. For these experiments, the neural network takes $\mathbf{A}^{\dagger}(\mathbf{y})$ as input. Here $\mathbf{A}^{\dagger}$ encapsulates the psuedo inverse of the ray transform and can easily be computed using the Filterd Back Projection (FBP) algorithm. We benchmark our model aware regularizers FJA+FJ (eq. 5.4) against the knowledge based Total Variation (TV) [93] denoising and data-driven post-processing UNet [5]. We also compare the model aware regularizer with model agnostic regularization, FJ, to observe the benefit of incorporating the knowledge of forward model in the regularization process.

The simulations are performed with the help of the Tensorflow and Operator Discretization Library.

6.2.1 Results

We have summarized the qualitative results for our tests on the LIDC dataset in Table 6.2 while the visual comparisons are given in Figs. 6.5 and 6.6.

[2] Although our theoretical analysis applies to bounded noise – experiments on Gaussian noise show that the proposed regularizer performs well in other settings as wel.

Table 6.2: Performance comparison of the proposed model aware regularizers (5.4) and (5.2) on the tomographic reconstruction problems in terms of PSNR(dB) and SSIM.

	noise variance = 0.01		noise variance = 0.05	
	PSNR	SSIM	PSNR	SSIM
FBP	15.07	0.3425	8.01	0.0477
TV	29.18	0.8310	27.43	0.8010
UNet post-processing [5]	33.31	0.9020	29.06	0.8316
UNet post-processing w FJ	34.53	0.9177	30.05	0.8612
UNet as post-processing w FJA&FJ	35.66	0.9338	31.11	0.8701

It can be seen from Table 6.2 that the proposed mapping induced regularizers consistently out perform the baselines. The PSNR and SSIM values also indicate that regularizing the state of the art data-driven methods [5] augments the quality of the original reconstructions. In high SNR regimes, importing the knowledge of the forward mapping via our regularizers results in an advantage of up to 2dB in PSNR and 0.03 in terms of the SSIM. Even, in settings, where the SNR is low, the quality of the reconstruction for model aware regularizers is superior to the networks that are unregularized or regularized with a model-agnostic penalty (FJ). It can be noted from the sample reconstruction results in Fig 6.5 that under high levels of noise, the model based FBP algorithm performs very poorly which is expected since the filtering process in the FBP algorithm usually results in the amplification of the measurement noise. In contrast, the TV regularization gives results that are comparable to the learnt methods. This is also not surprising since TV has been shown to perform well in such scenarios [132]. The neural network based reconstructions are however still superior to the model based reconstructions and are perceptually more appealing. Finally, although the learned reconstructions are of similar quality in general, the model aware regularizers recover images with lesser artifacts – a quality which is highly desirable when these methods are used during the diagnostic processes.

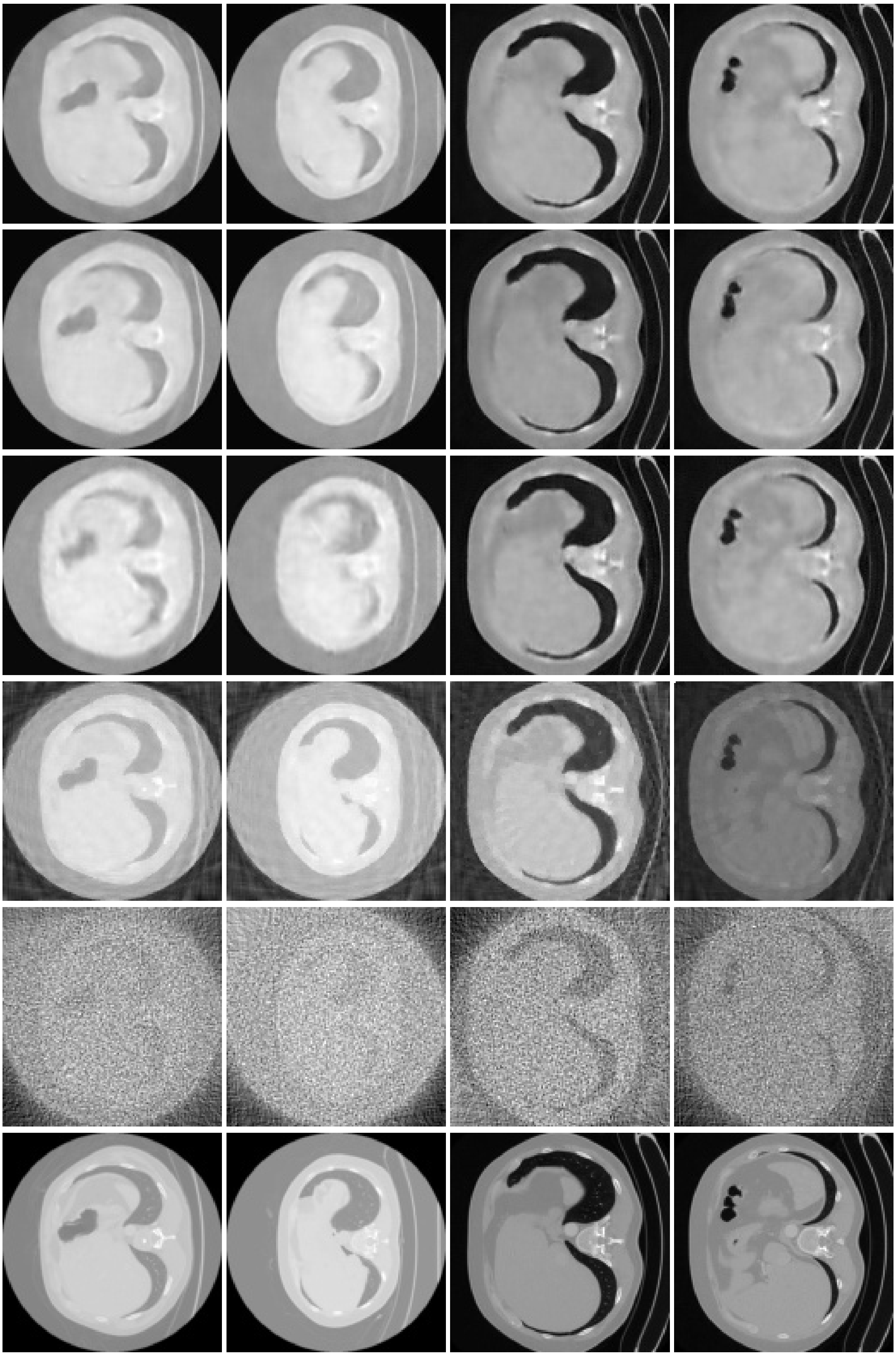


Figure 6.5: Reconstruction results of the tomographic data, when the noise variance is set to 0.05. (Left to Right) ground truth, reconstruction using FBP, TV, postprocessing using UNet [5], UNet postprocessing with FJ and UNet postprocessing with FJA&FJ.

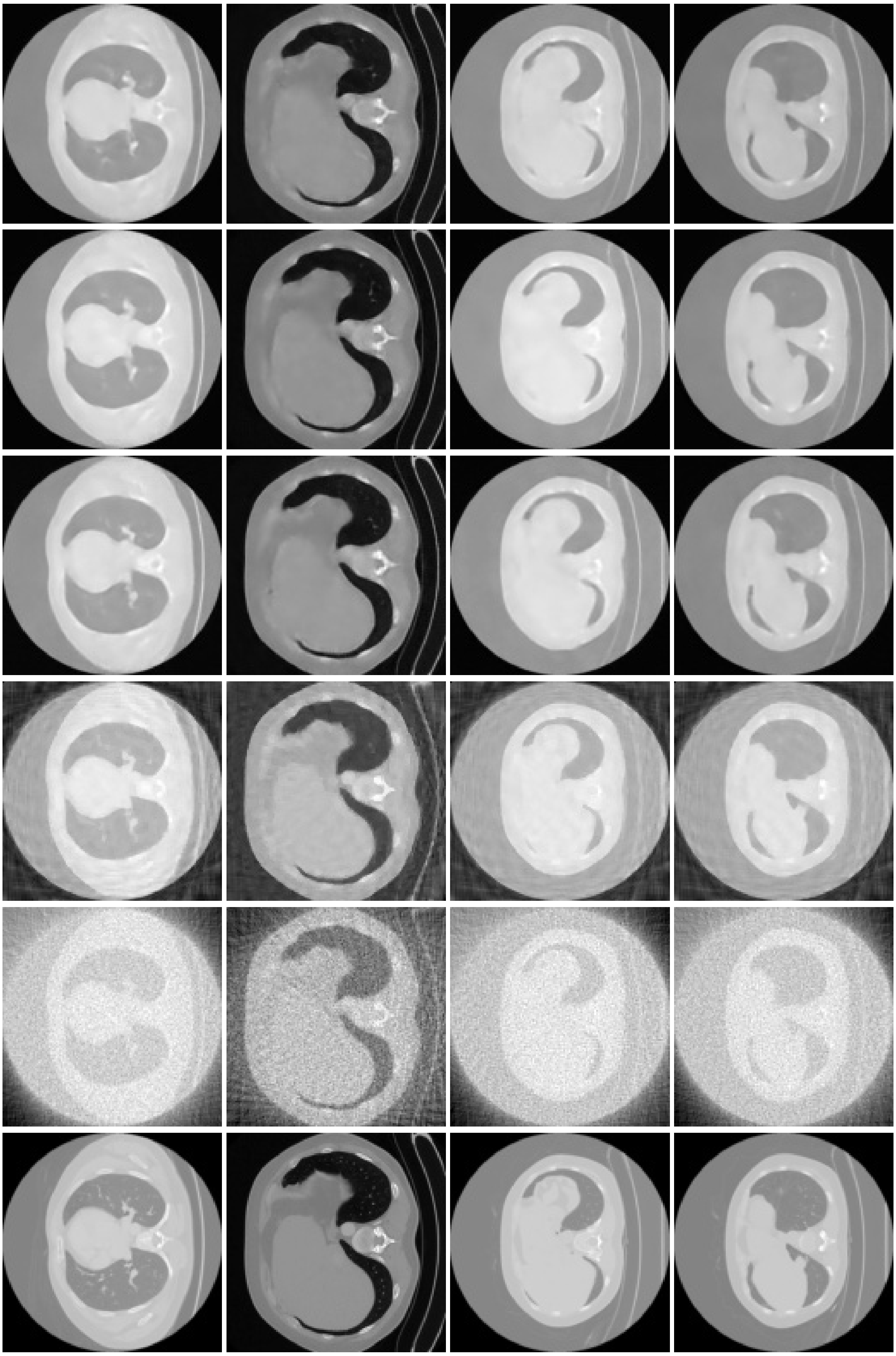


Figure 6.6: Reconstruction results of the tomographic data, when the noise variance is set to 0.01. (Left to Right) ground truth, reconstruction using FBP, TV, post-processing using UNet [5], UNet post-processing with FJ and UNet post-processing with FJA&FJ.

6.3 Summary

In this chapter we have demonstrated the efficacy of our model aware regularizers on a few notable biomedical imaging problems such as MRI and CT.

Chapter 7

Conclusions and Future Directions

7.1 Conclusion

In spite of the tremendous empirical achievements of DNNs, the generalization behaviour of these networks on a given task is still not clearly understood. In this work, by drawing on the robustness framework introduced by Xu and Mannor, we put forth generalization bounds for deep neural network based reconstruction that can be specialized for a wide range of regression and classification settings. Our analysis, in particular, tries to show how the hypothesis complexity – captured via network Jacobian and the generalization error are related.

We further extend our analysis to a special set of model based learning tasks, the linear inverse problem, occurring in various signal and image processing tasks. We bound the generalization error of deep neural network based learning algorithms on these tasks in terms of norm of the network Jacobian, norm of the product of the network Jacobian and the forward operator of the problem, the number of examples and the covering number of the space spanning the ground truth. These bounds motivate a new neural network learning procedure involving the use of cost functions in capturing knowledge of the underlying inverse problem model via appropriate regularization.

Motivated by our analysis, we propose a plug-and-play regularizer that penalized the Jacobian matrix of the relevant mappings and can be integrated into any deep learning based solver of inverse problems without additional complications. Empirical results on a variety of problems demonstrate that our proposed regularization approach can outperform considerably standard model agnostic regularizers and reconstruction schemes specialized for inverse problems. In particular, incorporating the knowledge of the forward map in the data-driven reconstruction methods is a problem of longstanding interest in literature [133]. This work adds to recent ones by showing there is much value incorporating model knowledge onto data-driven approaches.

Possible directions for future work include investigating how to marry data-driven approaches with model-based ones in the presence of model or distribution uncertainty.

7.2 Emerging Themes & Extensions

In the present work, we have shown that the robustness framework [22] can be leveraged to gain insights and develop efficient reconstruction methods for high dimensional input-output model based deep learning methods. We strongly believe that this is just a first step towards understanding and exploiting the structure underlying the data and the processes responsible for data generation as many crucial concerns that may arise in actual applications still need to be addressed.

7.2.1 Operator Mismatch in Inverse Problems

The reconstruction accuracy of the latent variable in inverse problems is highly dependent on the amount of noise or uncertainty present in the received measurements. Since inverse problems are typically ill-posed and therefore sensitive to errors, whether originating from data collection systems or from modelling, the discrepancy between the forward and inverse models may have a dramatic effect on the quality of the solution. A look at the convergence guarantees present in literature will reveal immediately that in a non-blind setting (when the decoder has the knowledge of the forward model), the reconstruction performance is upper bounded by the corruption in the measurement [134, 95]. Therefore, having an accurate knowledge of the forward operator and underlying physics is imperative for reliable recovery of the signal of interest. However, such desired accuracy is not always possible due to problems caused by stochasticity and high dimensions involved in these problems. Such issues of model mismatch may also arise due to discretization and model reduction. Therefore, in many of these applications, we have to resort to the use of approximations to these operators in order to restrict time and money consumption. This may lead to degradation in the quality of reconstructed variable [135].

Mathematically, such problems can be represented using the following model:

$$\mathbf{y} = \mathbf{A}'\mathbf{x} = (\mathbf{A} + d\mathbf{A})\mathbf{x} + \mathbf{n} \tag{7.1}$$

where $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^q$ is the measured data, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ is the unknown vector of interest, $\mathbf{A}$ is a presumably known noisy version of the true forward operator $\mathbf{A}'$, $d\mathbf{A}$ is a perturbation matrix representing the uncertainty in the model and finally $\mathbf{n}$ is the measurement noise. We assume $||d\mathbf{A}||_F \leq \alpha$ and $||\mathbf{n}||_2 \leq \eta$.

Various works in literature have proposed approaches that make appropriate corrections to account for the full physical phenomena [136, 137]. More recently, deep learning techniques have also been put forth as a means to 'learn' the corrections in the forward operators [138, 139]. A promising future research direction that can be of interest to the research community is to characterize the generalization performance of such deep learning algorithms applicable to inverse problems where we have mismatch in the knowledge of the forward model. Next we present some preliminary results that leverage the proof technique presented in Chapters 3 and 4 to derive upper bounds on the generalization error of such systems.

7.2.1.1 Preliminary Analysis

We consider an approach, where a regressor $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ has been trained for the inversion from the noisy observation $\mathbf{y}$ in (7.1), on a set of m examples $\mathcal{S} = \{(\mathbf{y}_i, \mathbf{x}_i)\}_{i \leq m}$, drawn independently and identically distributed (IID) from the sample space $\mathcal{D} = \mathcal{Y} \times \mathcal{X}$ according to an unknown distribution μ underlying the data.

We start by proving the following upper bound on the distance between the outputs of the network $f_{\mathcal{S}}$.

Theorem 7.1. Consider the linear map $\mathbf{A}' : \mathcal{X} \to \mathcal{Y}$ in (7.1). Now, consider a learning algorithm $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$. Then, for any $(\mathbf{y}_1, \mathbf{x}_1), (\mathbf{y}_2, \mathbf{x}_2) \in \mathcal{D}$, it follows that

$$\begin{aligned}\left\| f_{\mathcal{S}}(\mathbf{y}_1) - f_{\mathcal{S}}(\mathbf{y}_2) \right\|_2 \leq & \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} \left\| \mathbf{J}(\mathbf{y}) \mathbf{A} \right\|_{2,2} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\ & \qquad + \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} \left\| \mathbf{J}(\mathbf{y}) \right\|_{2,2} (2\eta + \alpha \|\mathbf{x}_2 - \mathbf{x}_1\|_2)\end{aligned}$$

where $\|.\|_{2,2}$ represents the spectral norm of a matrix.

Proof. See Appendix A. □

The *algorithmic robustness* framework (Definition 2.6 [22]) can then be used to show that the regressor is robust under the ℓ_2 loss.

Theorem 7.2. Consider that $\mathcal{X}$ and $\mathcal{Y}$ are compact spaces with respect to the ℓ_2. Consider also the sample space $\mathcal{D} = \mathcal{Y} \times \mathcal{X}$ compact with the metric ρ. Then, the DNN based regressor $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ trained on the training set $\mathcal{S}$ is (K,ϵ)-robust such that $K \leq \mathcal{N}(\psi/2; \mathcal{X}, \|.\|)$ and $\epsilon \leq \left(1 + \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} \left\| \mathbf{J}(\mathbf{y}) \mathbf{A} \right\|_{2,2} + \eta \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} \left\| \mathbf{J}(\mathbf{y}) \right\|_{2,2}\right) \psi + 2\eta \sup_{\mathbf{y} \in \text{conv}(\mathcal{Y})} \left\| \mathbf{J}(\mathbf{y}) \right\|_{2,2}$.

Proof. See Appendix A. □

The generalization error can then be bounded as following the techniques used in Chapters 3 and 4:

$$GE(f_{\mathcal{S}}) \leq \left(1 + \sup_{\mathbf{y}\in \mathrm{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2} + \alpha \sup_{\mathbf{y}\in \mathrm{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2}\right)\psi \tag{7.2}$$
$$+2\eta \sup_{\mathbf{y}\in \mathrm{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2} + M\sqrt{\frac{2\mathcal{N}\left(\psi/2;\mathcal{X},\|.\|\right)\log(2) + 2\log\left(1/\zeta\right)}{m}}$$

for any $M < \infty$.

The bound in 7.3 is quite similar to the results derived in Theorem 4.3 in that the generalization error depends on the norms of Jacobian of the network $f_{\mathcal{S}}$, norms of the product of the incorrect forward model $\mathbf{A}$ and the network Jacobian, the corruption levels η and α and the sample space complexity.

7.2.2 Mismatch in the Distribution

One of the main factors responsible for the tremendous success of DL techniques on a myriad of tasks [140, 141, 15] is the abundance of and the ubiquitous access to training datasets. The availability of such huge datasets, however comes with its own set of caveats. For instance, it has been shown that different image datasets have strong inherent biases that make it impossible for models trained on one dataset to generalize to another dataset [142]. This cognizance also raises important questions on the effectiveness of state of the art deep learning methods on real data that may come from a different population. This is because in such a situation, the independent and identically distributed (IID) assumption no more holds true.

The speed with which deep learning techniques are being integrated in sensitive applications – such as healthcare and privacy – and the realization that the distribution mismatch across source data (that is used to train the learning algorithm), and target data (on which the model is applied) may results in a dramatically sub-optimal performance has attracted significant attention from machine learning researchers. These investigations – mostly empirical – have subsequently given rise to domains such as transfer learning

(TL) and domain adaptation (DA) [143]. A few works – mainly addressing classification tasks – have also leveraged popular techniques from classical generalization theory to provide theoretical guarantees for TL and DA [144, 145, 146, 147].

In such problems, it is assumed that the training samples, $\mathcal{S}$ are generated according to a distribution $\mu_{\mathcal{S}}$ which is slightly different from μ – the one form which the testing samples are generated. Typically μ represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure $\mu_{\mathcal{S}}$ typically represents a theory, model, description, or approximation of μ. The two distributions are assumed to have a small distance $D \leq \delta$. This notion of discrepancy can be measured in terms of suitably defined Wasserstein, optimal transport distances or a divergence that depends on the likelihood of the distributions – such as the Kullback-Leibler divergence [148]. The generalization error framework discussed in previous chapters can be extended to include such scenarios leading to the following bound on the GE.

Theorem 7.3. Let two probability measure $\mu_{\mathcal{S}}$ and μ be defined on a compact measurable space $\mathcal{D}$. Consider now a DNN $f_{\mathcal{S}}$ trained on a set $\mathcal{S} = \{(\mathbf{y}_i, \mathbf{x}_i)\}_{i \leq m} \subset \mathcal{D}$ such that $(\mathbf{y}_i, \mathbf{x}_i) \sim \mu_{\mathcal{S}}$ for all $(\mathbf{y}_i, \mathbf{x}_i) \in \mathcal{S}$. Consider also that $f_{\mathcal{S}}$ is now tested on samples $(\mathbf{y}, \mathbf{x}) \in \mathcal{D}$ such that $(\mathbf{y}, \mathbf{x}) \sim \mu$. Then, with a probability $1 - \zeta$, the *GE* of $f_{\mathcal{S}}$ is bounded by:

$$GE(f_{\mathcal{S}}) \leq \epsilon(\mathcal{S})\psi + M\sqrt{2D_{KL}(\mu \| \mu_{\mathcal{S}})} + M\sqrt{\frac{2\mathcal{N}(\psi/2; \mathcal{D}, \rho)\log(2) + 2\log(1/\zeta)}{m}}$$

for any $M < \infty$.

Proof. See Appendix A. □

The bound in Theorem 7.3 is similar to the bound derived in Theorem 3.4 with an additional term accounting for the shift in the joint distribution of the train and test set. The formulation of our problem and the resulting bound obtained is subject to many factors and therefore we would like to make some observations.

Observation 7.1. The bound in Theorem 7.3 considers KL-divergence as the measure of discrepancy between μ and $\mu_{\mathcal{S}}$. A possible shortcoming of such likelihood based distance measure is that these ultimately require μ and $\mu_{\mathcal{S}}$ to have same support and therefore may result in an undesirable behaviour on out-of-sample sets where this criteria is unmet. Therefore, a potential extension to this work could be to utilize other more generic distance measures such as optimal transport.

Observation 7.2. There is a vast body of literature dedicated to theory and algorithms proposed in order to understand and obviate issues caused by distribution mismatch – giving rise to domain such as TL, DA and multitask learning. An approach that has recently emerged to tackle problems of similar essence is Distributionally Robust Optimization (DRO) [149]. Unlike TL – which involves knowledge transfer via fine tuning the model on the new/out-of-sample data – DRO refers to a generalization of empirical risk minimization. It follows a training approach which minimizes the worst case loss by solving a minmax problem. In recent times, some of the techniques in these areas have been extended and applied to study distribution shift in medical imaging problems such as tumor detection [150]. However, the advances in these applications are quite limited and therefore suggests the need for additional research.

Observation 7.3. In many inverse problems such as the one appearing in astronomical imaging, we may not have any 'real' ground truth and have to rely on training data generated through simulations leading to distributional differences with the test data. In other settings such as encountered in medicines – where training data is limited – we may combine data from multiple sources or acquisition hardware resulting in variability in image resolution, contrast, signal-to-noise ratio or mapping function restricting the applicability of learning algorithms in both research and clinical settings. Therefore, it'll be interesting to see research techniques such as TL and DRO applied to such scenarios.

Appendix A

Proofs of Chapter 7

proof of Theorem 7.1. Let $\mathcal{G} = f_{\mathcal{S}}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))$. Then the generalized fundamental theorem of calculus shows that $f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1) = \int_0^1 \frac{\partial \mathcal{G}}{d\theta} d\theta$, where

$$\begin{aligned}\frac{\partial \mathcal{G}}{d\theta} &= \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))(\mathbf{y}_2 - \mathbf{y}_1) \\ &= \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))[\mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) + d\mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) + (\mathbf{n}_2 - \mathbf{n}_1)]\end{aligned}$$

Then, from the sub-multiplicative property of matrix norms

$$\begin{aligned}\left\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_2)\right\|_2 \leq &\left\|\int_0^1 \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))\mathbf{A} d\theta\right\|_{2,2} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\ &+ \left\|\int_0^1 \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1)) d\theta\right\|_{2,2} \|d\mathbf{A}\|_F \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\ &+ \left\|\int_0^1 \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1)) d\theta\right\|_{2,2} \|\mathbf{n}_2 - \mathbf{n}_1\|_2\end{aligned}$$

where $||.||_{2,2}$ is the ℓ_2 induced matrix norm. Next, it is easy to show that

$$\begin{aligned}\left\|\int_0^1 \mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))\mathbf{A} d\theta\right\|_{2,2} &\overset{(a)}{\leq} \int_0^1 \left\|\mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))\mathbf{A}\right\|_{2,2} d\theta \\ &\leq \sup_{\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}, \theta \in [0,1]} \left\|\mathbf{J}(\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1))\mathbf{A}\right\|_{2,2}\end{aligned}$$

where (a) is because of the triangular inequality.

For $\theta \in [0,1]$, $\mathbf{y}_1 + \theta(\mathbf{y}_2 - \mathbf{y}_1)$ lies is in convex-hull of $\mathcal{Y}$. Hence

$$\begin{aligned}
&\left\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\right\|_2 \\
&\leq \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2} (2\eta + \|d\mathbf{A}\| \, \|\mathbf{x}_2 - \mathbf{x}_1\|_2) \\
&\leq \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2} (2\eta + \alpha \|\mathbf{x}_2 - \mathbf{x}_1\|_2)
\end{aligned}$$

□

proof of Theorem 7.2. Let $(\mathbf{y}_1,\mathbf{x}_1),(\mathbf{y}_2,\mathbf{x}_2) \in \mathcal{D}$. Then

$$\begin{aligned}
&|l(f_{\mathcal{S}}(\mathbf{y}_2),\mathbf{x}_2) - l(f_{\mathcal{S}}(\mathbf{y}_1),\mathbf{x}_1)| \\
&= \left| \|\mathbf{x}_2 - f_{\mathcal{S}}(\mathbf{y}_2)\|_2 - \|\mathbf{x}_1 - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \right| \\
&\overset{(a)}{\leq} \|\mathbf{x}_2 - f_{\mathcal{S}}(\mathbf{y}_2) - \mathbf{x}_1 + f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\
&\overset{(b)}{\leq} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\
&\overset{(c)}{\leq} \left(1 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2}\right) \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2} (2\eta + \alpha \|\mathbf{x}_2 - \mathbf{x}_1\|_2) \\
&\leq 2\eta \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2} \\
&\qquad + \left(1 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2} + \alpha \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2}\right) \|\mathbf{x}_2 - \mathbf{x}_1\|_2
\end{aligned}$$

where we leverage the reverse triangle inequality and Minkowski-inequality in (a) and (b) respectively. The inequality (c) holds because of the result proved in Theorem 7.1.

Then, for a $\psi/2$-cover the sample space $\mathcal{X}$ and for all $(\mathbf{y}_1,\mathbf{x}_1) \in \mathcal{S}, (\mathbf{y}_2,\mathbf{x}_2) \in \mathcal{D}$ falling in the same partition, we have

$$\begin{aligned}
|l(f_{\mathcal{S}}(\mathbf{y}_1),\mathbf{x}_1) - l(f_{\mathcal{S}}(\mathbf{y}_2),\mathbf{x}_2)| \leq &\left(1 + \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\mathbf{A}\right\|_{2,2} + \alpha \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2}\right)\psi \\
&+ 2\eta \sup_{\mathbf{y}\in\operatorname{conv}(\mathcal{Y})} \left\|\mathbf{J}(\mathbf{y})\right\|_{2,2}
\end{aligned}$$

leading directly to the result. □

Proof of Theorem 7.3. For $i \in \{1,2\ldots,K\}$, let $\mathcal{K}_i$ be a partition of $\mathcal{D}$. Define, $N_i = \{j|(\mathbf{y}_j,\mathbf{x}_j) \in \mathcal{S} \wedge \mathcal{K}_i\}$. Note that $(|N_1|,|N_2|\ldots,|N_K|)$ is a multinomial random variable with parameteres m and $(\mu_\mathcal{S}(\mathcal{K}_1),\mu_\mathcal{S}(\mathcal{K}_2)\ldots,\mu_\mathcal{S}(\mathcal{K}_K))$. Breteganolle-Huber-Carol inequality [151] is a a very useful result which quantifies the total variation distance $\sum_{i=1}^K \left|N_i - m\mu_\mathcal{S}(\mathcal{K}_i)\right|$ giving rise to the following inequality:

$$P\left(\sum_{i=1}^{K}\left|\frac{|N_i|}{m} - \mu_\mathcal{S}(\mathcal{K}_i)\right| \leq \sqrt{\frac{2K\log(2)+2\log(1/\zeta)}{m}}\right) \geq 1-\zeta \tag{A.1}$$

The generalization error is now given by:

$$\begin{aligned}
&|l_{\text{exp}}(f_\mathcal{S}) - l_{\text{emp}}(f_\mathcal{S})| \\
&= \left|\sum_{i=1}^{K}\mathbb{E}_{(\mathbf{y},\mathbf{x})\sim\mu}[l(f_\mathcal{S}(\mathbf{y}),\mathbf{x})|(\mathbf{y},\mathbf{x})\in\mathcal{K}_i]\mu(\mathcal{K}_i) - \frac{1}{m}\sum_{i=1}^{m} l(f_\mathcal{S}(\mathbf{y}_i),\mathbf{x}_i)\right| \\
&\overset{(a)}{\leq} \left|\sum_{i=1}^{K}\mathbb{E}_{(\mathbf{y},\mathbf{x})\sim\mu}[l(f_\mathcal{S}(\mathbf{y}),\mathbf{x})|(\mathbf{y},\mathbf{x})\in\mathcal{K}_i]\frac{|N_i|}{m} - \frac{1}{m}\sum_{i=1}^{m} l(f_\mathcal{S}(\mathbf{y}_i),\mathbf{x}_i)\right| \\
&\qquad + \left|\sum_{i=1}^{K}\mathbb{E}_{(\mathbf{y},\mathbf{x})\sim\mu}[l(f_\mathcal{S}(\mathbf{y}),\mathbf{x})|(\mathbf{y},\mathbf{x})\in\mathcal{K}_i]\mu(\mathcal{K}_i) - \sum_{i=1}^{K}\mathbb{E}_{(\mathbf{y},\mathbf{x})\sim\mu}[l(f_\mathcal{S}(\mathbf{y}),\mathbf{x})|(\mathbf{y},\mathbf{x})\in\mathcal{K}_i]\frac{|N_i|}{m}\right| \\
&\leq \left|\frac{1}{m}\sum_{i=1}^{K}\sum_{j\in N_i}\max_{(\mathbf{x}',\mathbf{y}')\in\mathcal{K}_i}|l(f_\mathcal{S},(\mathbf{x}_j,\mathbf{y}_j)) - l(f_\mathcal{S},(\mathbf{x}',\mathbf{y}'))|\right| + \left|\max_{(\mathbf{y},\mathbf{x})\in\mathcal{D}} l(f_\mathcal{S}(\mathbf{y}),\mathbf{x})\sum_{i=1}^{K}\left|\mu(\mathcal{K}_i) - \frac{|N_i|}{m}\right|\right| \\
&\overset{(b)}{\leq} \epsilon(\mathcal{S}) + M\sum_{i=1}^{K}\left|\mu(\mathcal{K}_i) - \frac{|N_i|}{m}\right| \\
&\overset{(c)}{\leq} \epsilon(\mathcal{S}) + M\sum_{i=1}^{K}\left|\mu(\mathcal{K}_i) - \mu_\mathcal{S}(\mathcal{K}_i)\right| + M\sum_{i=1}^{K}\left|\mu_\mathcal{S}(\mathcal{K}_i) - \frac{|N_i|}{m}\right| \\
&\overset{(d)}{\leq} \epsilon(\mathcal{S}) + M\sqrt{2D_{KL}(\mu||\mu_\mathcal{S})} + M\sum_{i=1}^{K}\left|\mu_\mathcal{S}(\mathcal{K}_i) - \frac{|N_i|}{m}\right|
\end{aligned}$$

Here (a) and (c) are due to the triangle inequality. (b) holds because of Definition 2.6 and (d) is because of Lemma A.1

Substituting (A.1) in (d) and replacing K with its upper bound – the covering number on $\mathcal{D}$, $\mathcal{N}(\frac{\psi}{2};\mathcal{D},\rho)$ – directly leads to the result:

$$GE(f_{\mathcal{S}}) \le \epsilon(\mathcal{S})\psi + M\sqrt{2D_{KL}(\mu\|\mu_{\mathcal{S}})} + M\sqrt{\frac{2\mathcal{N}(\psi/2;\mathcal{D},\rho)\log(2) + 2\log(1/\zeta)}{m}}$$

□

Lemma A.1. Let the compact metric space $\mathcal{D}$ be partitioned into K disjoint partions $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_K$ such that

$$\cup_{i=1}^{K}\mathcal{K}_i = \mathcal{D}, \qquad \text{and} \qquad \mathcal{K}_i \cap \mathcal{K}_j = \emptyset, \quad \text{for } i \neq j \tag{A.2}$$

Then

$$\sum_{i=1}^{K} |\mu(\mathcal{K}_i) - \mu_{\mathcal{S}}(\mathcal{K}_i)| \le \sqrt{2D_{KL}(\mu\|\mu_{\mathcal{S}})} \tag{A.3}$$

Proof. Let us first prove a lower bound to the total variation distance:

$$\begin{aligned}\sum_{i=1}^{K} |\mu(\mathcal{K}_i) - \mu_{\mathcal{S}}(\mathcal{K}_i)| &= \sum_{i=1}^{K} \left| \sum_{(\mathbf{y},\mathbf{x})\in\mathcal{K}_i} (\mu((\mathbf{y},\mathbf{x})) - \mu_{\mathcal{S}}((\mathbf{y},\mathbf{x}))) \right| \\ &\overset{(a)}{\le} \sum_{i=1}^{K} \sum_{(\mathbf{y},\mathbf{x})\in\mathcal{K}_i} \left|\mu((\mathbf{y},\mathbf{x})) - \mu_{\mathcal{S}}((\mathbf{y},\mathbf{x}))\right| \\ &\overset{(b)}{=} \sum_{(\mathbf{y},\mathbf{x})\in\mathcal{D}} \left|\mu((\mathbf{y},\mathbf{x})) - \mu_{\mathcal{S}}((\mathbf{y},\mathbf{x}))\right|\end{aligned}$$

where (a) is because of the triangle inequality. Note that the Total Variation (TV) distance, $V(\mu,\mu_{\mathcal{S}})$ between μ and $\mu_{\mathcal{S}}$ is given by:

$$V(\mu,\mu_{\mathcal{S}}) = \frac{1}{2} \sum_{(\mathbf{y},\mathbf{x})\in\mathcal{D}} \left|\mu((\mathbf{y},\mathbf{x})) - \mu_{\mathcal{S}}((\mathbf{y},\mathbf{x}))\right| \tag{A.4}$$

Substituting (A.4) in (b) gives:

$$\sum_{i=1}^{K} |\mu(\mathcal{K}_i) - \mu_{\mathcal{S}}(\mathcal{K}_i)| \le 2V(\mu,\mu_{\mathcal{S}}) \tag{A.5}$$

We can now leverage the Kullback–Csizšar–Kemperman inequality that links the TV distance $V(\mu, \mu_{\mathcal{S}})$ and the KL divergence $D_{KL}(\mu \| \mu_{\mathcal{S}})$ and is given by [152, 153, 154, 155]:

$$V(\mu, \mu_{\mathcal{S}}) \leq \sqrt{\frac{D_{KL}(\mu \| \mu_{\mathcal{S}})}{2}} \tag{A.6}$$

Substituting the (A.5) in (A.6) concludes the proof. □

Bibliography

[1] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[2] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[3] Jiahui Yu, Yuchen Fan, and Thomas Huang. Wide activation for efficient image and video super-resolution. In Kirill Sidorov and Yulia Hicks, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 52.1–52.13. BMVA Press, September 2019.

[4] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems*, pages 8507–8516, 2018.

[5] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[6] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnel-

son, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated mri, 2018.

[7] Dong Liang, Jing Cheng, Ziwen Ke, and Leslie Ying. Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Processing Magazine*, 37(1):141–151, 2020.

[8] J. HADAMARD. Sur les problemes aux derivees partielles et leur signification physique. *Princeton university bulletin*, pages 49–52, 1902.

[9] Frank Natterer. *The mathematics of computerized tomography*. SIAM, 2001.

[10] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[11] Ralph A. Willoughby. *Solutions of Ill-Posed Problems (A. N. Tikhonov and V. Y. Arsenin)*, volume 21. 1979.

[12] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[13] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[16] Ali Mousavi, Ankit B Patel, and Richard G Baraniuk. A deep learning approach to structured signal recovery. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 1336–1343. IEEE, 2015.

[17] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[18] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[19] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[20] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[21] Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *arXiv preprint arXiv:1905.03684*, 2019.

[22] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

[23] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[26] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[27] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[28] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[29] Kui Jia, Dacheng Tao, Shenghua Gao, and Xiangmin Xu. Improving training of deep neural networks via singular value bounding. In *Conf Comp Vis Pattern Recognit*, volume 2017, pages 3994–4002, 2017.

[30] Kui Jia, Shuai Li, Yuxin Wen, Tongliang Liu, and Dacheng Tao. Orthogonal deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[31] D. Plaut, S. Nowlan, and Geoffrey E. Hinton. Experiments on learning by back propagation. *Technical Report*, 1986.

[32] Peter M Williams. Bayesian regularization and pruning using a laplace prior. *Neural computation*, 7(1):117–143, 1995.

[33] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

[34] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *Advances in Neural Information Processing Systems*, pages 4262–4272, 2018.

[35] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301, 2019.

[36] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[37] Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017.

[38] Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 2017.

[39] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[41] Yoshua Bengio, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Dumitru Erhan, Jeremy Eustache, Xavier Glorot, Xavier Muller, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 164–172, 2011.

[42] Nelson Morgan and Hervé Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in neural information processing systems*, pages 630–637, 1990.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[44] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*, 2016.

[45] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[46] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[47] Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37. San Matteo, CA: Morgan Kaufmann, 1988.

[48] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[49] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[50] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] Dozat Timothy. Incorporating nesterov momentum into adam. *Natural Hazards*, 3(2):437–453, 2016.

[53] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[54] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[55] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[56] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 888–896. PMLR, 2019.

[57] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[58] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[59] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[60] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[61] Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1064–1068. PMLR, 07–10 Jul 2017.

[62] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[63] Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pages 1651–1686, 1998.

[64] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[65] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[66] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

[67] Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-dependent coresets for compressing neural

networks with applications to generalization bounds. *arXiv preprint arXiv:1804.05345*, 2018.

[68] Taiji Suzuki, Hiroshi Abe, Tomoya Murata, Shingo Horiuchi, Kotaro Ito, Tokuma Wachi, So Hirai, Masatoshi Yukishima, and Tomoaki Nishimura. Spectral pruning: Compressing deep neural networks via spectral analysis and its generalization error. *arXiv preprint arXiv:1808.08558*, 2018.

[69] David Pollard. *Convergence of stochastic processes*. Springer Science & Business Media, 2012.

[70] Daniel Jakubovitz, Raja Giryes, and Miguel RD Rodrigues. Generalization error in deep learning. *arXiv preprint arXiv:1808.01174*, 2018.

[71] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[72] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

[73] Harris Drucker and Yann Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

[74] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

[75] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[76] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

[77] Jaweria Amjad, Jure Sokolić, and Miguel RD Rodrigues. On deep learning for inverse problems. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1895–1899. IEEE, 2018.

[78] Christian Etmann. A closer look at double backpropagation. *arXiv preprint arXiv:1906.06637*, 2019.

[79] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[80] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[81] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[82] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

[83] Farzan Farnia, Jesse M Zhang, and David Tse. Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457*, 2018.

[84] Hanie Sedghi, Vineet Gupta, and Philip M. Long. The singular values of convolutional layers. *CoRR*, abs/1805.10408, 2018.

[85] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128, 2016.

[86] Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. On orthogonality and learning recurrent networks with long term dependencies. *arXiv preprint arXiv:1702.00071*, 2017.

[87] Nik Weaver. *Lipschitz algebras*. World Scientific, 1999.

[88] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[89] Maurice Fréchet. *Généralisation d'un théorème de Weierstrass*. gauthier-Villars, 1904.

[90] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.

[91] Jennifer L Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10. Siam, 2012.

[92] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[93] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[94] Andrey N Tikonov and Vasily Y Arsenin. Solutions of ill-posed problems. *New York: Winston*, 1977.

[95] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006.

[96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[97] Gaofeng Wang, Jun Zhang, and Guang-Wen Pan. Solution of inverse problems in image processing by wavelet expansion. *IEEE Transactions on Image Processing*, 4(5):579–593, 1995.

[98] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[99] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[100] Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled mri reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.

[101] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[102] Junyoung Park, Donghwi Hwang, Kyeong Yun Kim, Seung Kwan Kang, Yu Kyeong Kim, and Jae Sung Lee. Computed tomography super-resolution using deep convolutional neural network. *Physics in Medicine & Biology*, 63(14):145011, 2018.

[103] Sarah B Scruggs, Karol Watson, Andrew I Su, Henning Hermjakob, John R Yates III, Merry L Lindsey, and Peipei Ping. Harnessing the heart of big data. *Circulation research*, 116(7):1115–1119, 2015.

[104] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[105] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018.

[106] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

[107] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *arXiv preprint arXiv:1912.10557*, 2019.

[108] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

[109] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[110] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[111] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[112] Jean Gallier. Notes on convex sets, polytopes, polyhedra, combinatorial topology, voronoi diagrams and delaunay triangulations, 2008.

[113] B Bolzano. Functionenlehre, edited by k. *Rychlik. Royal Bohemian Academy of Sciences, Prague*, 1930.

[114] Andrew Tonge. Equivalence constants for matrix norms: a problem of goldberg. *Linear Algebra and its Applications*, 306(1-3):1–13, 2000.

[115] Andrew D Lewis. A top nine list: Most popular induced matrix norms. *Queen's University, Kingston, Ontario, Tech. Rep*, 2010.

[116] Nakul Verma. Distance preserving embeddings for general n-dimensional manifolds. *The Journal of Machine Learning Research*, 14(1):2415–2448, 2013.

[117] Ingrid Daubechies and Gerd Teschke. Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising. *Applied and Computational Harmonic Analysis*, 19(1):1–16, 2005.

[118] Michael Elad, J-L Starck, Philippe Querre, and David L Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

[119] Jean-Luc Starck, Mai K Nguyen, and Fionn Murtagh. Wavelets and curvelets for image deconvolution: a combined approach. *Signal processing*, 83(10):2279–2283, 2003.

[120] Brian D Jeffs and Metin Gunsay. Restoration of blurred star field images by maximally sparse optimization. *IEEE Transactions on Image Processing*, 2(2):202–211, 1993.

[121] Raja Giryes, Guillermo Sapiro, and Alexander M Bronstein. Deep neural networks with random gaussian weights: a universal classification strategy? *IEEE Trans. Signal Processing*, 64(13):3444–3457, 2016.

[122] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[123] Joel A Tropp and Stephen J Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[124] RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.

[125] Jamie Townsend. A new trick for calculating Jacobian vector products. `https://j-towns.github.io/2017/06/12/A-new-trick.html`, 2017. Accessed: 2020-01-17.

[126] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[127] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE transactions on image processing*, 23(6):2569–2582, 2014.

[128] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[130] Fritz John. The ultrahyperbolic differential equation with four independent variables. In *Fritz John*, pages 79–101. Springer, 1985.

[131] Samuel G. Armato, Geoffrey McLennan, Luc Bidaut, Michael F. McNitt-Gray, Charles R. Meyer, Anthony P. Reeves, Heber MacMahon, Roger M. Engelmann, Rachael Y. Roberts, Adam Starkey, Philip Caligiuri, Denise R. Aberle, Matthew S. Brown, Richard C. Pais, David P-Y Qing, Poonam Batra, C. Matilda Jude, Iva Petkovska, Alberto M. Biancardi, Binsheng Zhao, Claudia I. Henschke, David Yankelevitz, Daniel Max, Ali Farooqi, Eric A. Hoffman, Edwin J. R. van Beek, Amanda R. Smith, Ella A. Kazerooni, Peyton H. Bland, Gary E. Laderach, Gregory W. Gladish, Reginald F. Munden, Leslie E. Quint, Lawrence H. Schwartz, Baskaran Sundaram, Lori E. Dodd, Charles Fenimore, David Gur, Nicholas Petrick, John Freymann, Justin Kirby, Brian Hughes, Alessi Vande Casteele, Sangeeta Gupte, Maha Sallam, Michael D. Heath, Michael H. Kuhn, Ekta Dharaiya, Richard Burns, David S. Fryd, Marcos Salganicoff, Vikram Anand, Uri Shreter, Stephen Vastagh, Barbara Y. Croft, and Laurence P. Clarke. The lung image database consortium, (lidc) and image database resource initiative (idri):: a completed reference database of lung nodules on ct scans. *Medical Physics*, 38(2):915–931, February 2011.

[132] Zhen Tian, Xun Jia, Kehong Yuan, Tinsu Pan, and Steve B Jiang. Low-dose ct reconstruction via edge-preserving total variation regularization. *Physics in Medicine & Biology*, 56(18):5949, 2011.

[133] Nir Shlezinger, Jay Whang, Yonina C. Eldar, and Alexandros G. Dimakis. Model-based deep learning, 2020.

[134] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[135] Jeffrey H Siewerdsen and David A Jaffray. Cone-beam computed tomography with a flat-panel imager: magnitude and effects of x-ray scatter. *Medical physics*, 28(2):220–231, 2001.

[136] Ruola Ning, Xiangyang Tang, and David Conover. X-ray scatter correction algorithm for cone beam ct imaging. *Medical physics*, 31(5):1195–1202, 2004.

[137] Lei Zhu, Yaoqin Xie, Jing Wang, and Lei Xing. Scatter correction for cone-beam ct in radiation therapy. *Medical physics*, 36(6Part1):2258–2268, 2009.

[138] Sebastian Lunz, Andreas Hauptmann, Tanja Tarvainen, Carola-Bibiane Shonlieb, and Simon Arridge. On learned operator correction in inverse problems. *SIAM Journal on Imaging Sciences*, 14(1):92–127, 2021.

[139] Andreas Hauptmann, Ben Cox, Felix Lucka, Nam Huynh, Marta Betcke, Paul Beard, and Simon Arridge. Approximate k-space models and deep learning for fast photoacoustic reconstruction. In *International Workshop on Machine Learning for Medical Image Reconstruction*, pages 103–111. Springer, 2018.

[140] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[141] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[142] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[143] Lei Zhang. Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687*, 2019.

[144] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[145] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[146] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[147] Zirui Wang. Theoretical guarantees of transfer learning. *arXiv preprint arXiv:1810.05986*, 2018.

[148] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[149] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review, 2019.

[150] Lucas Fidon, Sebastien Ourselin, and Tom Vercauteren. Generalized wasserstein dice score, distributionally robust deep learning, and ranger for brain tumor segmentation: Brats 2020 challenge, 2021.

[151] Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak Convergence and Empirical Processes*, pages 16–28. Springer, 1996.

[152] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[153] Solomon Kullback. A lower bound for discrimination information in terms of variation (corresp.). *IEEE transactions on Information Theory*, 13(1):126–127, 1967.

[154] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[155] Johannes HB Kemperman. On the optimum rate of transmitting information. In *Probability and information theory*, pages 126–169. Springer, 1969.