Nr. 71
06. June 2020

universität
innsbruck
Institut für Mathematik

Preprint-Series: Department of Mathematics - Applied Mathematics

# Regularization of Inverse Problems by Neural Networks

M. Haltmeier, L. V. Nguyen

APPLIEDMATHEMATICS

Technikerstraße 13 - 6020 Innsbruck - Austria
Tel.: +43 512 507 53803 Fax: +43 512 507 53898
https://applied-math.uibk.ac.at

# Regularization of Inverse Problems by Neural Networks

Markus Haltmeier

Department of Mathematics. University of Innsbruck Technikerstrasse 13, 6020 Innsbruck, Austria. markus.haltmeier@uibk.ac.at

Linh Nguyen

Department of Mathematics, University of Idaho, Moscow, ID 83844, US, lnguyen@uidaho.edu

May 31, 2020

**Abstract**

Inverse problems arise in a variety of imaging applications including computed tomography, non-destructive testing, and remote sensing. The characteristic features of inverse problems are the non-uniqueness and instability of their solutions. Therefore, any reasonable solution method requires the use of regularization tools that select specific solutions and at the same time stabilize the inversion process. Recently, data-driven methods using deep learning techniques and neural networks demonstrated to significantly outperform classical solution methods for inverse problems. In this chapter, we give an overview of inverse problems and demonstrate the necessity of regularization concepts for their solution. We show that neural networks can be used for the data-driven solution of inverse problems and review existing deep learning methods for inverse problems. In particular, we view these deep learning methods from the perspective of regularization theory, the mathematical foundation of stable solution methods for inverse problems. This chapter is more than just a review as many of the presented theoretical results extend existing ones.

**Keywords:** Inverse problems, deep learning, neural networks, regularization theory, ill-posedness, stability, theoretical foundation.

## Contents

# 1 Introduction

The solution of inverse problems arises in a variety of practically important applications including medical imaging, computer vision, geophysics as well as many other branches of pure and applied sciences. Inverse problems are most efficiently formulated as an estimation problem of the form

$$\text{recover} \quad \mathbf{x}^* \in \mathbb{X} \quad \text{from data} \quad \mathbf{y} = \mathbf{A}(\mathbf{x}^*) + \xi \in \mathbb{Y}\,. \tag{1.1}$$

Here $\mathbf{A} \colon \mathbb{X} \to \mathbb{Y}$ is a mapping between normed spaces, $\mathbf{x}^* \in \mathbb{X}$ is the true unknown solution, $\mathbf{y}$ represents the given data, and $\xi$ is an unknown data perturbation. In this context, the application of the operator $\mathbf{A}$ is referred to as the forward operator or forward problem, and solving (1.1) is the corresponding inverse problem. In the absence of noise where $\xi = 0$ we refer to $\mathbf{y} = \mathbf{A}(\mathbf{x}^*)$ as exact data, and in the case where $\xi \neq 0$ we refer to $\mathbf{y}$ as noisy data.

Prime examples of inverse problems are image reconstruction problems, where the forward operator describes the data generation process depending on the image reconstruction modality. For example, in X-ray computed tomography (CT), the forward operator is the sampled Radon transform whereas in magnetic resonance imaging (MRI) the forward operator is the sampled Fourier transform. Reconstructing the diagnostic image from experimentally collected data amounts to solving an inverse problem of the form (1.1). In these and other applications, the underlying forward operator is naturally formulated between infinite dimensional spaces, because the object to be reconstructed is a function of a continuous spatial variable. Even though the numerical solution is performed in a finite dimensional discretization, the mathematical properties of the continuous formulation are crucial for understanding and improving image formation algorithms.

## 1.1 Ill-posedness

The inherent character of inverse problems is their ill-posedness. This means that even in the case of exact data, the solution of (1.1) is either not unique, not existent, or does not stably depend on the given data. More formally, for an inverse problem at least one of the following three unfavorable properties holds:

(I1) NON-UNIQUENESS: For some $\mathbf{x}_1^* \neq \mathbf{x}_2^* \in \mathbb{X}$ we have $\mathbf{A}(\mathbf{x}_1^*) = \mathbf{A}(\mathbf{x}_2^*)$.

(I2) NON-EXISTENCE: For some $\mathbf{y} \in \mathbb{Y}$, the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has no solution.

(I3) INSTABILITY: Smallness of $\|\mathbf{A}(\mathbf{x}_1^*) - \mathbf{A}(\mathbf{x}_2^*)\|$ does not imply smallness of $\|\mathbf{x}_1^* - \mathbf{x}_2^*\|$.

These conditions imply that the forward operator does not have a continuous inverse which could be used to directly solve (1.1). Instead, regularization methods have to be applied which result in stable methods for solving inverse problem.

Regularization methods approach the ill-posedness by two steps. Firstly, to address non-uniqueness and non-existence issues (I1), (I2), one restricts the image and preimage space of the forward operator to sets $\mathbb{M} \subseteq \mathbb{X}$ and $\operatorname{ran}(\mathbf{A}) \subseteq \mathbb{Y}$ such that the restricted forward operator $\mathbf{A}_{\mathrm{res}} \colon \mathbb{M} \to \operatorname{ran}(\mathbf{A})$ becomes bijective. For any exact data, the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ then has a unique solution in $\mathbb{M}$ which is given by the inverse of the restricted forward operator applied to $\mathbf{y}$. Secondly, in order to address the instability issue (I3), one considers a family of continuous operators $\mathbf{B}_\alpha \colon \mathbb{Y} \to \mathbb{X}$ for $\alpha > 0$ that converge to $\mathbf{A}_{\mathrm{res}}^{-1}$ in a suitable sense; see Section 2 for precise definitions.

Note that the choice of the set $\mathbb{M}$ is crucial as it represents the class of desired reconstructions and acts as selection criteria for picking a particular solution of the given inverse problem. A main challenge is that this class is actually unknown or at least it cannot be described properly. For example, in CT for medical imaging, set of desired solutions represents the set of all functions corresponding to spatially attenuation inside patients, a function class that is clearly challenging, if not impossible, to describe in mathematical terms.

Variational regularization and variants [43] have been the most successful class of regularization methods for solving inverse problems. Here, $\mathbb{M}$ is defined as solutions having a small value of a certain regularization functional that can be interpreted as a measure for the deviation from desired solutions. Various regularization functionals have been analyzed for inverse problems, including Hilbert space norms [17], total variation [1] and sparse $\ell^q$-penalties [14, 21]. Such handcrafted regularization functionals have limited complexity and are unlikely to accurately model complex signal classes arising in applications such as medical imaging. On the other hand, their regularization effects are well understood, efficient numerical algorithms have been developed for their realization, they work reasonably well in practice, and they have been rigorously analyzed mathematically.

## 1.2 Data-driven reconstruction

Recently data-driven methods based on neural networks and deep learning demonstrated to significantly outperform existing variational and iterative reconstruction algorithms for solving inverse problems. The essential idea is to use neural networks to define a class $(\mathbf{R}_\theta)_{\theta\in\Theta}$ of reconstruction networks $\mathbf{R}_\theta\colon \mathbb{Y} \to \mathbb{X}$ and to select the parameter vector $\theta \in \Theta$ of the network in data-driven manner. The selection is based on a set of training data $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ where $\mathbf{x}_i \in \mathbb{M}$ are desired reconstructions and $\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i^*) + \xi_i \in \mathbb{Y}$ are corresponding data. Even if the set $\mathbb{M}$ of desired reconstructions is unknown, the available samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ can be used to select the particular reconstruction method. A typical selection strategy is to minimize a penalized least squares functional having the form

$$\theta^* \in \operatorname*{arg\,min}_\theta \left\{ \frac{1}{N} \sum_{i=1}^N \bigl\| \mathbf{x}_i - \mathbf{R}_\theta(\mathbf{y}_i) \bigr\|^2 + P(\theta) \right\} . \tag{1.2}$$

The final neural network based reconstruction method is then given by $\mathbf{R}_{\theta^*}\colon \mathbb{Y} \to \mathbb{X}$ and is such that in average it performs well on the given training data set.

Existing deep learning-based methods include post-processing networks [23, 27], null-space networks [45, 46], variational networks [28], iterative networks [51, 2, 3], network cascades [29, 44] and learned regularization functional [32, 34, 38]. We refer to the review [6] for other data-driven reconstruction methods such as GANs [9, 35], dictionary learning, deep basis pursuit [47] or deep image priors [48, 49, 15], that we do not touch in this chapter. Post-processing networks and null-space networks are explicit, where the reconstruction network is given explicitly and its parameters are trained to fit the given training data. Methods using learned regularizers are implicit and the reconstruction network $\mathbf{R}_\theta(\mathbf{y}) = \arg\min \mathcal{T}_{\theta,\mathbf{y}}$ is defined by minimizing a properly trained Tikhonov functional $\mathcal{T}_{\theta,\mathbf{y}}\colon \mathbb{X} \to [0,\infty]$. Variational networks and iterative networks are in between, where $\arg\min \mathcal{T}_{\theta,\mathbf{y}}$ is approximated via an iterative scheme using $L$ steps.

Any reasonable method for solving an inverse problem, including all learned reconstruction schemes, has to include some form of regularization. However, regularization may be imposed implicitly, even without noticing by the researcher developing the algorithm. Partially, this is the case because discretization, early stopping, or other techniques to numerically stabilizing an optimization algorithm at the same time has a regularization effect on the underlying inverse problem. Needless to say, understanding and analyzing where exactly the regularization effect comes from, will increase reliability of any algorithm and allows its further improvement. In conclusion, any data-driven reconstruction method has to include either explicitly or implicitly a form of regularization. In this chapter, we will analyze the regularization properties of various deep learning methods for solving inverse problems.

### 1.3 Outline

The outline of this chapter is as follows. In Section 2 we present the background of inverse problems and deep learning. In Section 3 we analyze direct neural network based reconstructions whereas in Section 4 we study variational and iterative reconstruction methods based on neural networks. The chapter concludes with a discussion and some final remarks given in Section 5. While the concepts presented in the subsequent sections are known, most of the presented results extend existing ones. Therefore, this chapter is much more than just a review over existing results.

For the sake of clarity, in this chapter we only study linear inverse problems even several results can be extended non-linear problems as well. We will provide remarks pointing to such results. Throughout, we allow an infinite dimensional setting, because in many applications the unknowns to be recovered as well as the data are most naturally modeled as functions which lie in infinite dimensional spaces $\mathbb{X}$ and $\mathbb{Y}$. However, everything said in this chapter applies to finite dimensional spaces as well. In limited data problems, such as sparse view CT, the finite dimension of the data space $\mathbb{Y}$ is even an intrinsic part of the forward model. Therefore, the reader not familiar with infinite dimensional vector space can think of $\mathbb{X}$ and $\mathbb{Y}$ as finite dimensional vector spaces each equipped with a standard vector norm.

## 2 Preliminaries

In this section, we provide necessary background on linear inverse problems, their regularization and their solution with neural networks.

Throughout the following, $\mathbb{X}$ and $\mathbb{Y}$ are Banach spaces. We study solving inverse problems of the form (1.1) in a deterministic setting with a bounded linear forward operator $\mathbf{A}\colon \mathbb{X} \to \mathbb{Y}$. Hence we aim for estimating the unknown signal $\mathbf{x}^* \in \mathbb{X}$ from the available data $\mathbf{y} = \mathbf{A}(\mathbf{x}^*) + \xi$, where $\xi \in \mathbb{Y}$ is the noise that is assumed to satisfy an estimate of the form $\|\xi\| \leq \delta$. Here $\delta \geq 0$ is called the noise level and in the case $\delta = 0$ we call $\mathbf{y} = \mathbf{A}(\mathbf{x}^*)$ the exact data.

## 2.1 Right inverses

As we have explained in the introduction, the main feature of inverse problems is their ill-posedness. Regularization methods approach the ill-posedness by two steps. In the first step, they address (I1) and (I2) by restricting the image and the pre-image spaces which gives a certain right inverse defined on $\operatorname{ran}(\mathbf{A})$. In order to address the instability issue (I3), regularization methods are applied for stabilization. We first consider right inverse and their instability, and consider the regularization in the following subsection.

**Definition 2.1** (Right inverse)**.** A possibly non-linear mapping $\mathbf{B} \colon \operatorname{ran}(\mathbf{A}) \subseteq \mathbb{Y} \to \mathbb{X}$ is called right inverse of $\mathbf{A}$ if $\mathbf{A}(\mathbf{B}(\mathbf{y})) = \mathbf{y}$ for all $\mathbf{y} \in \operatorname{ran}(\mathbf{A})$.

Clearly, a right inverse always exists because for any $\mathbf{y} \in \operatorname{ran}(\mathbf{A})$ there exists an element $\mathbf{B}\mathbf{y} := \mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$. However, in general no continuous right inverse exists. More precisely, we have the following result (compare [36]).

**Proposition 2.2** (Continuous right inverses)**.** *Let $\mathbf{B} \colon \operatorname{ran}(\mathbf{A}) \to \mathbb{X}$ be a continuous right inverse. Then $\operatorname{ran}(\mathbf{A})$ is closed.*

*Proof.* By continuity, $\mathbf{B}$ can be extended in a unique way to a mapping $\mathbf{H} \colon \overline{\operatorname{ran}(\mathbf{A})} \to \mathbb{X}$. Let $\mathbf{x}_n \in \operatorname{ran}(\mathbf{A})$ with $\mathbf{x} = \lim_{n\to\infty} \mathbf{x}_n \in \overline{\operatorname{ran}(\mathbf{A})}$. The continuity of $\mathbf{H}$ and $\mathbf{A}$ implies $\mathbf{A} \circ \mathbf{H} = \operatorname{Id}_{\overline{\operatorname{ran}(\mathbf{A})}}$. Therefore $\overline{\operatorname{ran}(\mathbf{A})} = \operatorname{ran}(\mathbf{A} \circ \mathbf{H}) \subseteq \operatorname{ran}(\mathbf{A}) \subseteq \overline{\operatorname{ran}(\mathbf{A})}$ which shows that $\operatorname{ran}(\mathbf{A})$ is closed. □

Proposition 2.2 implies that whenever $\operatorname{ran}(\mathbf{A})$ is non-closed, then $\mathbf{A}$ does not have a continuous right inverse.

The next question we study is the existence of a linear right inverse. For that purpose recall that a mapping $\mathbf{P} \colon \mathbb{X} \to \mathbb{X}$ is called projection if $\mathbf{P}^2 = \mathbf{P}$. If $\mathbf{P}$ is a linear bounded projection, then $\operatorname{ran}(\mathbf{P})$ and $\ker(\mathbf{P})$ are closed subspaces and $\mathbb{X} = \operatorname{ran}(\mathbf{P}) \oplus \ker(\mathbf{P})$.

**Definition 2.3** (Complemented subspace)**.** A closed subspace $\mathbb{V}$ of $\mathbb{X}$ is called complemented in $\mathbb{X}$, if there exists a bounded linear projection $\mathbf{P}$ with $\operatorname{ran}(\mathbf{P}) = \mathbb{V}$

A closed subspace $\mathbb{V} \subseteq \mathbb{X}$ is complemented if and only if there is another closed subspace $\mathbb{U} \subseteq \mathbb{X}$ with $\mathbb{X} = \mathbb{U} \oplus \mathbb{V}$. In a Hilbert space, any closed subspace is complemented, and $\mathbb{X} = \mathbb{V}^\perp \oplus \mathbb{V}$ with the orthogonal complement $\mathbb{V}^\perp := \{u \in \mathbb{X} \mid \forall v \in \mathbb{V} \colon \langle u, v \rangle = 0\}$. However, as shown in [33], in every Banach space that is not isomorphic to a Hilbert space there exist closed subspaces which are not complemented.

**Proposition 2.4** (Linear right inverses)**.**

*(a)* $\mathbf{A}$ *has a linear right inverse if and only if* $\ker(\mathbf{A})$ *is complemented.*

*(b) A linear right inverse is continuous if and only* $\operatorname{ran}(\mathbf{A})$ *is closed.*

*Proof.* (a) First suppose that $\mathbf{A}$ has a linear right inverse $\mathbf{B}\colon \operatorname{ran}(\mathbf{A}) \to \mathbb{X}$. For any $\mathbf{x} \in \mathbb{X}$ we have $(\mathbf{B} \circ \mathbf{A})^2(\mathbf{x}) = \mathbf{B} \circ (\mathbf{A} \circ \mathbf{B})(\mathbf{A}(\mathbf{x})) = (\mathbf{B} \circ \mathbf{A})(\mathbf{x})$. Hence $\mathbf{B} \circ \mathbf{A}$ is a linear bounded projection. This implies the topological decomposition $\mathbb{X} = \operatorname{ran}(\mathbf{B} \circ \mathbf{A}) \oplus \ker(\mathbf{B} \circ \mathbf{A})$ with closed subspaces $\operatorname{ran}(\mathbf{B} \circ \mathbf{A})$ and $\ker(\mathbf{B} \circ \mathbf{A})$. It holds $\ker(\mathbf{B} \circ \mathbf{A}) \supseteq \ker(\mathbf{A}) = \ker(\mathbf{A} \circ \mathbf{B} \circ \mathbf{A}) \supseteq \operatorname{ran}(\mathbf{B} \circ \mathbf{A})$ which shows that $\ker(\mathbf{A}) = \operatorname{ran}(\mathbf{B} \circ \mathbf{A})$ is complemented. Conversely let $\ker(\mathbf{A})$ be complemented and write $\mathbb{X} = \mathbb{X}_1 \oplus \ker(\mathbf{A})$. Then $\mathbf{A}_{\mathrm{res}}\colon \mathbb{X}_1 \to \operatorname{ran}(\mathbf{A})$ is bijective and therefore has a linear inverse $\mathbf{A}_{\mathrm{res}}^{-1}$ defining a right inverse for $\mathbf{A}$.

(b) For any continuous right inverse, $\operatorname{ran}(\mathbf{A})$ is closed according to Proposition 2.2. Conversely, let $\mathbf{B}\colon \operatorname{ran}(\mathbf{A}) \to \mathbb{X}$. be linear right inverse and $\operatorname{ran}(\mathbf{A})$ closed. In particular, $\ker(\mathbf{A})$ is complemented and we can write $\mathbb{X} = \mathbb{X}_1 \oplus \ker(\mathbf{A})$. The restricted mapping $\mathbf{A}_{\mathrm{res}}\colon \mathbb{X}_1 \to \overline{\operatorname{ran}(\mathbf{A})}$ is bijective therefore bounded according to the bounded inverse theorem. This implies that $\mathbf{B}$ is bounded, too. □

In a Hilbert space $\mathbb{X}$ the kernel $\ker(\mathbf{A})$ of a bounded linear operator is complemented, as any other closed subspace of $\mathbb{X}$. Therefore, according to Proposition 2.4, any bounded linear operator defined on a Hilbert space has a linear right inverse. However, in a general Banach space this is not the case, as the following example shows.

**Example 2.5** (Bounded linear operator without linear right inverse)**.** *Consider the set* $c_0(\mathbb{N})$ *of all sequences converging to zero as a subspace of the space* $\ell^\infty(\mathbb{N})$ *of all bounded sequences* $\mathbf{x}\colon \mathbb{N} \to \mathbb{R}$ *with the supremum norm* $\|\mathbf{x}\|_\infty := \sup_{n\in\mathbb{N}}|\mathbf{x}(n)|$. *Note that* $c_0(\mathbb{N}) \subseteq \ell^\infty(\mathbb{N})$ *is a classic example for a closed subspace that is not complemented in a Banach space, as first shown in [39]. Now consider the quotient space* $\mathbb{Y} = {}^{\ell^\infty(\mathbb{N})}\!/\!_{c_0(\mathbb{N})}$ *where elements in* $\ell^\infty(\mathbb{N})$ *are identified if their difference is contained in* $c_0(\mathbb{N})$. *Then the quotient map* $\mathbf{A}\colon \ell^\infty(\mathbb{N}) \to \mathbb{Y}\colon \mathbf{x} \mapsto [\mathbf{x}]$ *is clearly linear, bounded, and onto with* $\ker(\mathbf{A}) = c_0(\mathbb{N})$. *It is clear that a right inverse of* $\mathbf{A}$ *exists which can be constructed by simply choosing any representative in* $[\mathbf{x}]$. *However, because* $c_0(\mathbb{N})$ *is not complemented, the kernel of* $\mathbf{A}$ *is not complemented and according to Proposition 2.4 no linear right inverse of* $\mathbf{A}$ *exists.*

At first glance it might be surprising that bounded linear forward operators do not always have linear right inverses. However, following Example 2.5 one constructs bounded linear operators without linear right inverses for every Banach space that is not isomorphic to a Hilbert space. This in particular includes the function spaces $L^p(\Omega)$ with $p \neq 2$, where inverse problems are often formulated on.

**Proposition 2.6** (Right inverses in Hilbert spaces)**.** *Let* $\mathbb{X}$ *be a Hilbert space and let* $\mathbf{P}_{\ker(\mathbf{A})}\colon \mathbb{X} \to \mathbb{X}$ *denote the orthogonal projection onto* $\ker(\mathbf{A})$.

*(a)* $\mathbf{A}$ *has a unique linear right inverse* $\mathbf{A}^\dagger\colon \operatorname{ran}(\mathbf{A}) \to \mathbb{X}$ *with* $\mathbf{A} \circ \mathbf{A}^\dagger = \operatorname{Id} - \mathbf{P}_{\ker(\mathbf{A})}$.

*(b)* $\forall \mathbf{y} \in \operatorname{ran}(\mathbf{A})\colon \mathbf{A}^\dagger(\mathbf{y}) = \arg\min\{\|\mathbf{x}\| \mid \mathbf{A}(\mathbf{x}) = \mathbf{y}\}$.

*(c)* $\mathbf{A}^\dagger$ *is continuous if and only if* $\operatorname{ran}(\mathbf{A})$ *is closed.*

*(d) If* $\operatorname{ran}(\mathbf{A})$ *is non-closed, then any right inverse is discontinuous.*

*Proof.* In a Hilbert space the orthogonal complement $\ker(\mathbf{A})^\perp$ defines a complement of $\ker(\mathbf{A})$ and therefore (a), (c), (d) follow from Propositions 2.2 and 2.4. Item (b) holds because any solution of the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has the form $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 \in \ker(\mathbf{A})^\perp \oplus \ker(\mathbf{A})$ and we have $\|\mathbf{x}\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2$ according to Pythagoras theorem. □

In the case that $\mathbb{X}$ and $\mathbb{Y}$ are both Hilbert spaces there is a unique extension $\mathbf{A}^\dagger\colon \operatorname{ran}(\mathbf{A}) \oplus \operatorname{ran}(\mathbf{A})^\perp \to \mathbb{X}$ such that $\mathbf{A}^\dagger(\mathbf{y}_1 \oplus \mathbf{y}_2) = \mathbf{A}^\dagger(\mathbf{y}_1)$ for all $\mathbf{y}_1 \oplus \mathbf{y}_2 \in \operatorname{ran}(\mathbf{A}) \oplus \operatorname{ran}(\mathbf{A})^\perp$. The operator $\mathbf{A}^\dagger$ is referred to as the Moore-Penrose inverse of $\mathbf{A}$. For more background on generalized in inverses in Hilbert and Banach spaces see [36].

## 2.2 Regularization methods

Let $\mathbf{B}\colon \operatorname{ran}(\mathbf{A}) \subseteq \mathbb{Y} \to \mathbb{X}$ be a right inverse of $\mathbf{A}$, set $\mathbb{M} := \operatorname{ran}(\mathbf{B})$ and suppose $\mathbb{M}^* \subseteq \mathbb{M}$. Moreover, let $\mathcal{D}\colon \mathbb{Y} \times \mathbb{Y} \to [0, \infty]$ be some functional measuring closeness in the data space. The standard choice is the squared norm distance $\mathbf{d}_{\mathbb{Y}}(\mathbf{y}, \mathbf{y}^\delta) = \|\mathbf{y} - \mathbf{y}^\delta\|$ but also other choices will be considered in this chapter.

**Definition 2.7** (Regularization method)**.** A function $\mathbf{R}\colon (0, \infty) \times \mathbb{Y} \to \mathbb{X}$ with

$$\forall \mathbf{x} \in \mathbb{M}^*\colon \lim_{\delta \to 0} \sup\Big\{\|\mathbf{x} - \mathbf{R}(\delta, \mathbf{y}^\delta)\| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) \leq \delta\Big\} = 0\,. \tag{2.1}$$

is called (convergent) regularization method for (1.1) on the signal class $\mathbb{M}^* \subseteq \mathbb{M}$ with respect to the similarity measure $\mathcal{D}$. We also write $(\mathbf{R}_\delta)_{\delta > 0}$ instead of $\mathbf{R}$.

The following lemma gives a useful guideline for creating regularization methods based on point-wise approximations of $\mathbf{B}$.

**Proposition 2.8** (Point-wise approximations are regularizations)**.** *Let* $(\mathbf{B}_\alpha)_{\alpha > 0}$ *be a family of continuous operators* $\mathbf{B}_\alpha\colon \mathbb{Y} \to \mathbb{X}$ *that converge uniformly to* $\mathbf{B}$ *on* $\mathbf{A}(\mathbb{M}^*)$ *as* $\alpha \to 0$. *Then, there is a function* $\alpha_0\colon (0, \infty) \to (0, \infty)$ *such that*

$$\mathbf{R}\colon (0, \infty) \times \mathbb{Y} \to \mathbb{X}\colon (\delta, \mathbf{y}^\delta) \mapsto \mathbf{R}(\delta, \mathbf{y}^\delta) := \mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta) \tag{2.2}$$

*is a regularization method for* (1.1) *on the signal class* $\mathbb{M}^*$ *with respect to the similarity measure* $\mathbf{d}_{\mathbb{Y}}$. *One calls* $\alpha_0$ *an a-prior parameter choice over the set* $\mathbb{M}^*$.

*Proof.* For any $\epsilon > 0$ choose $\alpha(\epsilon)$ such $\|\mathbf{B}_{\alpha(\epsilon)}(\mathbf{y}) - \mathbf{x}\| \leq \epsilon/2$ for all $\mathbf{x} \in \mathbb{M}^*$. . Moreover, choose $\tau(\epsilon)$ such that for all $\mathbf{z} \in \mathbb{Y}$ with $\|\mathbf{y} - \mathbf{z}\| \leq \tau(\epsilon)$ we have $\|\mathbf{B}_{\alpha(\epsilon)}(\mathbf{y}) - \mathbf{B}_{\alpha(\epsilon)}(\mathbf{z})\| \leq \epsilon/2$. Without loss of generality we can assume that $\tau(\epsilon)$ is strictly increasing and continuous with $\tau(0+) = 0$. We define $\alpha_0 \coloneqq \alpha \circ \tau^{-1}$. Then, for every $\delta > 0$ and $\|\mathbf{y} - \mathbf{y}^\delta\| \leq \delta$,

$$\begin{aligned}\|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta) - \mathbf{x}\| &\leq \|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}) - \mathbf{x}\| + \|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}) - \mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta)\| \\ &= \|\mathbf{B}_{\alpha\circ\tau^{-1}(\delta)}(\mathbf{y}) - \mathbf{x}\| + \|\mathbf{B}_{\alpha\circ\tau^{-1}(\delta)}(\mathbf{y}) - \mathbf{B}_{\alpha\circ\tau^{-1}(\delta)}(\mathbf{y}^\delta)\| \\ &\leq \tau^{-1}(\delta)/2 + \tau^{-1}(\delta)/2 = \tau^{-1}(\delta)\,.\end{aligned}$$

Because $\tau^{-1}(\delta) \to 0$ as $\delta \to 0$ this completes the proof. □

A popular class of regularization methods is convex variational regularization defined by a convex functional $\mathcal{R}\colon \mathbb{X} \to [0, \infty]$. These methods approximate right inverses, given by the $\mathcal{R}$-minimizing solutions of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. Such solutions are elements in $\arg\min\{\mathcal{R}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{X} \wedge \mathbf{A}(\mathbf{x}) = \mathbf{y}\}$. Note that an $\mathcal{R}$-minimizing solution exists whenever $\mathbb{X}$ is reflexive, $\mathcal{R}$ is coercive and weakly lower semi-continuous, and the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has at least one solution in the domain of $\mathcal{R}$. Moreover, $\mathcal{R}$-minimizing solution are unique if $\mathcal{R}$ is strictly convex. In this case this immediately defines a right inverse for $\mathbf{A}$. Convex variational regularization is defined by minimizing the Tikhonov functional $\mathcal{T}_{\mathbf{y}^\delta,\alpha}\colon \mathbb{X} \to [0, \infty]\colon \mathbf{x} \mapsto \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha\mathcal{R}(\mathbf{x})$ for data $\mathbf{y}^\delta \in \mathbb{Y}$ and regularization parameter $\alpha > 0$. In Section 4 we will study a more general form including non-convex regularizers defined by a neural network. At his point, we only state one result on convex variational regularization.

**Theorem 2.9** (Tikhonov regularization in Banach spaces)**.** *Let $\mathbb{X}$ be reflexive, strictly convex and $p, q > 1$. Moreover, suppose that $\mathbb{X}$ satisfies the Radon–Riesz property; that is, for any sequence $(\mathbf{x}_k)_{k\in\mathbb{N}} \in \mathbb{X}^\mathbb{N}$ the weak convergence $\mathbf{x}_k \rightharpoonup \mathbf{x} \in \mathbb{X}$ together with the convergence in the norm $\|\mathbf{x}_k\| \to \|\mathbf{x}\|$ implies $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}$ in the norm topology. Then the following holds:*

*(a)* $\mathbf{A}^\dagger\colon\ \mathrm{ran}(\mathbf{A}) \to \mathbb{X}\colon \mathbf{y} \mapsto \arg\min\{\|\mathbf{x}\| \mid \mathbf{x} \in \mathbb{X} \wedge \mathbf{A}(\mathbf{x}) = \mathbf{y}\}$ *is well-defined.*

*(b) For all $\alpha > 0$ the mapping $\mathbf{B}_\alpha\colon \mathbb{Y} \to \mathbb{X}\colon \mathbf{y}^\delta \mapsto \arg\min\{\|\mathbf{A}(\mathbf{x}) - \mathbf{y}^\delta\|^p + \|\mathbf{x}\|^q \mid \mathbf{x} \in \mathbb{X}\}$ is well defined and continuous.*

*(c) For any $\alpha_0\colon (0, \infty) \to (0, \infty)$ with $\alpha_0 \to 0$ and $\delta^p/\alpha_0(\delta) \to 0$ as $\delta \to 0$, the mapping defined by* (2.2) *and (b) is a regularization method for* (1.1) *on $\mathbf{A}^\dagger(\mathbb{X})$ with respect to the norm distance $\mathbf{d}_\mathbb{X}$*

*Proof.* See [26, 43]. □

In the Hilbert space setting, the mapping $\mathbf{A}^\dagger$ defined In Theorem 2.9 is given by the Moore-Penrose inverse, see Proposition 2.6 and the text below this Proposition.

## 2.3 Deep learning

In this subsection, we give a brief review of neural networks and deep learning. Deep learning can be characterized as the field where deep neural networks are used to solve various learning problems [31, 20]. Several such methods recently appeared as a new paradigm for solving inverse problems. In deep learning literature, neural networks are often formulated in a finite dimensional setting. To allow a unified treatment we consider here a general setting including the finite dimensional as well as the infinite dimensional setting.

**Problem 2.10** (The Supervised learning problem)**.** *Suppose the aim is to find an unknown function $\mathbf{\Phi}\colon \mathbb{Y} \to \mathbb{X}$ between two Banach spaces. Similar to classical regression, we are given data $(\mathbf{y}_i, \mathbf{x}_i) \in \mathbb{Y} \times \mathbb{X}$ with $\mathbf{\Phi}(\mathbf{y}_i) \simeq \mathbf{x}_i$ for $i = 1, \dots, N$. From this data, we aim to estimate the function $\mathbf{\Phi}$. For that purpose one chooses a certain class $(\mathbf{\Phi}_\theta)_{\theta\in\Theta}$ of functions $\mathbf{\Phi}_\theta\colon \mathbb{Y} \to \mathbb{X}$ and defines $\mathbf{\Phi} \coloneqq \mathbf{\Phi}_{\theta^*}$ where $\theta^*$ minimizes the penalized empirical risk functional*

$$R_N\colon \Theta \to [0, \infty]\colon \theta \mapsto \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{\Phi}_\theta(\mathbf{y}_i), \mathbf{x}_i) + P(\theta)\,. \tag{2.3}$$

*Here $L\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is the so called loss function which is a measure for the error made by the function $\mathbf{\Phi}_\theta$ on the training examples, and $P$ is a penalty that prevents overfitting of the network and also stabilizes the training process.*

Both, the numerical minimization of the functional (2.3) and investigating properties of $\theta^*$ as $N \to \infty$ are of interest in its own [19, 13], but not subject of our analysis. Instead, most theory in this chapter is developed under the assumption of suitable trained prediction functions.

**Definition 2.11** (Neural network)**.** Let $\Theta$ be a parameter set and $H_{\ell,\theta}\colon \mathbb{X}_0 \times \cdots \times \mathbb{X}_{\ell-1} \to \mathbb{X}_\ell$, for $\ell = 1, \dots, L$ and $\theta \in \Theta$, be mappings between Banach spaces with $\mathbb{X}_0 = \mathbb{Y}$ and $\mathbb{X}_L = \mathbb{X}$. We call a family $(\mathbf{\Phi}_\theta)_{\theta\in\Theta}$ of recursively defined mappings

$$\mathbf{\Phi}_\theta \coloneqq a_{L,\theta}\colon \mathbb{Y} \to \mathbb{X} \quad \text{where } \forall \ell \in \{1, \dots, L\}\colon a_{\ell,\theta} = H_{\ell,\theta}(\mathrm{Id}, a_{1,\theta}, \dots, a_{\ell-1,\theta}) \tag{2.4}$$

a neural network. In that context, $\mathbb{X}_1, \dots, \mathbb{X}_{L-1}$ are called the hidden spaces. We refer to the individual members $\mathbf{\Phi}_\theta$ of a neural network as neural network functions.

A neural network in finite dimension can be seen as discretization of $(\mathbf{\Phi}_\theta)_{\theta\in\Theta}$ where $\mathbb{Y}$ and $\mathbb{X}$ are discretized using any standard discretization approach.

**Example 2.12** (Layered neural network)**.** *As a typical example for a neural network consider a layered neural network $(\mathbf{\Phi}_\theta)_{\theta\in\Theta}$ with $L$ layers between finite dimensional spaces.*

*In this case, each network function has the form* $\mathbf{\Phi}_\theta \colon \mathbb{R}^p \to \mathbb{R}^q \colon \mathbf{y} \mapsto (\sigma_L \circ \mathcal{V}_L^\theta) \circ \cdots \circ (\sigma_1 \circ \mathcal{V}_1^\theta)(\mathbf{y})$*, where* $\mathcal{V}_\ell^\theta \colon \mathbb{R}^{d(\ell-1)} \to \mathbb{R}^{d(\ell)}$ *are affine mappings and* $\sigma_\ell \colon \mathbb{R}^{d(\ell)} \to \mathbb{R}^{d(\ell)}$ *are nonlinear mappings with* $d(0) = p$ *and* $d(L) = q$*. The notion indicates that the affine mappings depend on the parameters* $\theta \in \Theta$ *while the nonlinear mappings are taken fixed. Although this is standard in neural networks, modifications where the nonlinearities contain trainable parameters have been proposed [4, 40]. The affine parts* $\mathcal{V}_\ell^\theta$ *which are the learned parts in the neural network can be represented by a* $d(\ell) \times d(\ell-1)$ *matrix for the linear part and a vector of size* $1 \times d(\ell)$ *for the translation part.*

In standard neural networks, the entries of the matrix and the bias vector are taken as independent parameters. For typical inverse problems where the dimensions $p$ and $q$ are large, learning all these numbers is challenging and perhaps impossible task. For example, the matrix describing the linear part of a layer mapping a $200 \times 200$ image to an image of the same size already contains $1.6$ billion parameters. Learning these parameters from data seems challenging. Recent neural networks and in particular convolutional neural networks (CNNs) use the concepts of sparsity and weight sharing to significantly reduce the number of parameters to be learned.

**Example 2.13** (CNNs using sparsity and weight sharing)**.** *In order to reduce the number of free parameter between a linear mapping between images, say of sizes* $q = n \times n$ *and* $p = n \times n$*, CNNs implement sparsity and weight sharing via convolution operators. In fact, a convolution operation* $\mathbf{K} \colon \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ *with kernel size* $k \times k$ *is represented by* $k^2$ *numbers which clearly enormously reduces the number* $n^4$ *of parameters required to represent a general linear mapping on* $\mathbb{R}^{n \times n}$*. To enrich the expressive power of the neural network, actual CNN architectures use multiple-input multiple-output convolutions* $\mathbf{K} \colon \mathbb{R}^{n \times n \times c} \to \mathbb{R}^{n \times n \times d}$ *which uses one convolution kernel for each pair in* $\{1, \dots, c\} \times \{1, \dots, d\}$ *formed between each input channel and each output channel. This now increases the number of learnable parameters to* $cdk^2$ *but overall the number of parameters remains much smaller than for a full dense layer between large images. Moreover, the use of multiple-input multiple-output convolutions in combination with typical nonlinearities introduces a flexible and complex structure which demonstrated to give state of the art results in various imaging tasks.*

## 3 Regularizing networks

Throughout this section let $\mathbf{A} \colon \mathbb{X} \to \mathbb{Y}$ be a linear forward operator between Banach spaces and $\mathbf{B} \colon \operatorname{ran}(\mathbf{A}) \to \mathbb{X}$ a linear right inverse with $\mathbb{U} \coloneqq \operatorname{ran}(\mathbf{B})$. In particular, the kernel of $\mathbf{A}$ is complemented and we can write $\mathbb{X} = \mathbb{U} \oplus \ker(\mathbf{A})$. The results in this section generalize the methods and some of the results of [45] from the Hilbert case to the Banach space case.

## 3.1 Null-space networks

The idea of post-processing networks is to improve a given right inverse by applying a network. Standard networks however will destroy data consistency of the initial reconstruction. Null space networks are the natural class of neural networks restoring data consistency.

**Definition 3.1** (Null space network)**.** We call the family $(\mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)_{\theta\in\Theta}$ a null space network if $(\mathbf{N}_\theta)_{\theta\in\Theta}$ is any network of Lipschitz continuous functions $\mathbf{N}_\theta \colon \mathbb{X} \to \mathbb{X}$. We will also refer to individual functions $\mathbf{\Phi}_\theta = \mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ as null space networks.

Any null space network $\mathbf{\Phi}_\theta = \mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ preserves data consistency in the sense that $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ implies $\mathbf{A}(\mathbf{\Phi}_\theta(\mathbf{x})) = \mathbf{y}$, which can be seen from

$$\mathbf{A} \circ \big(\mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta\big)(\mathbf{x}) = \mathbf{A}(\mathbf{x}) + \mathbf{A} \circ \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta(\mathbf{x}) = \mathbf{y}\,. \tag{3.1}$$

A standard residual network $\mathrm{Id}_{\mathbb{X}} + \mathbf{N}_\theta$ often used as post processing network in general does not satisfy this property (3.1).

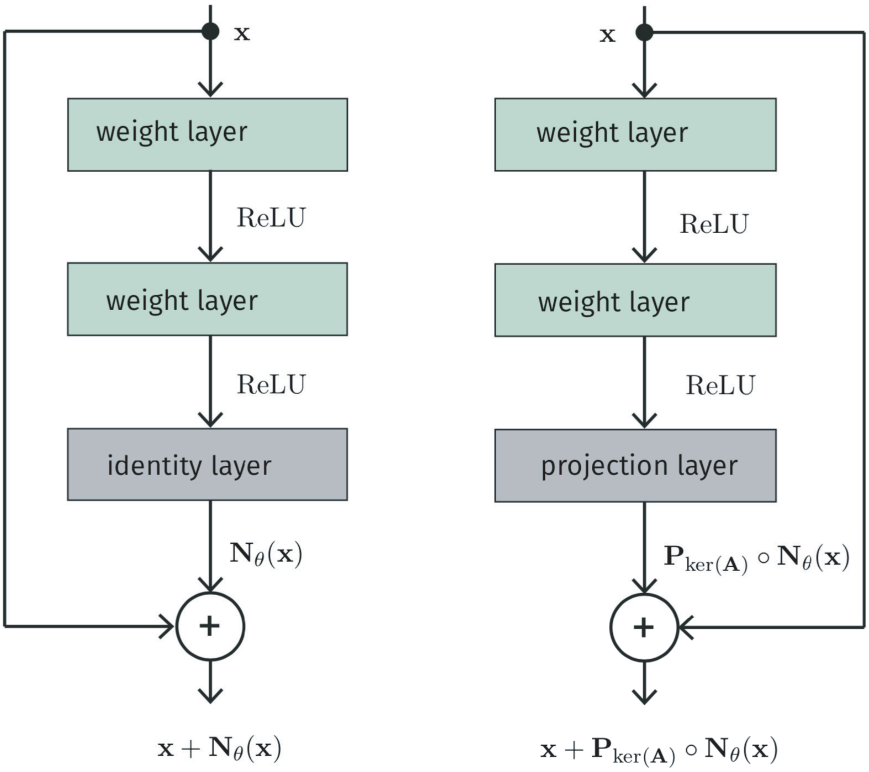
$\mathbf{x}$
weight layer
ReLU
weight layer
ReLU
identity layer
$\mathbf{N}_\theta(\mathbf{x})$
+
$\mathbf{x} + \mathbf{N}_\theta(\mathbf{x})$
$\mathbf{x}$
weight layer
ReLU
weight layer
ReLU
projection layer
$\mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta(\mathbf{x})$
+
$\mathbf{x} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta(\mathbf{x})$

Figure 3.1: Residual network $\mathrm{Id} + \mathbf{N}_\theta$ (left) versus null space network $\mathrm{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ (right). The difference between the two architectures is the projection layer $\mathbf{P}_{\ker(\mathbf{A})}$ in the null space network after the last weight layer.

**Remark 3.2** (Computation of the projection layer)**.** *A main ingredient in the null-space* *network is the computation of the projection layer $\mathbf{P}_{\ker(\mathbf{A})}$. In some cases, it can be computed explicitly. For example, if $\mathbf{A} = \mathbf{S}_I \circ \mathbf{F}$ is the subsampled Fourier transform, then $\mathbf{P}_{\ker(\mathbf{A})} = \mathbf{F}^* \circ \mathbf{S}_I \circ \mathbf{F}$. For a general forward operator between Hilbert spaces the projection $\mathbf{P}_{\ker(\mathbf{A})}\mathbf{z}$ can be implemented via standard methods for solving linear equation. For example, using the starting value $\mathbf{z}$ and solving the equation $\mathbf{A}(\mathbf{x}) = 0$ with the CG (conjugate gradient) method for the normal equation or Landwebers methods gives a sequence that converges to the projection $\mathbf{P}_{\ker(\mathbf{A})}\mathbf{z} = \arg\min\{\|\mathbf{x} - \mathbf{z}\| \mid \mathbf{A}(\mathbf{x}) = 0\}$.*

An example comparing a standard residual network $\mathrm{Id}_{\mathbb{X}} + \mathbf{N}_\theta$ and a null space network $\mathrm{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ both with two weight layers are shown in Figure 3.1.

**Proposition 3.3** (Right inverses defined by null space networks)**.** *Let $\mathbf{B}\colon \mathbb{Y} \to \mathbb{X}$ be a given linear right inverse and $\mathbf{\Phi}_\theta = \mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ be a null space network. Then the composition*

$$\mathbf{\Phi}_\theta \circ \mathbf{B}\colon\ \mathrm{ran}(\mathbf{A}) \to \mathbb{X}\colon \mathbf{y} \mapsto (\mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)(\mathbf{B}\mathbf{y}) \tag{3.2}$$

*is right inverse of $\mathbf{A}$. Moreover, the following assertions are equivalent:*

*(i)* $\mathbf{\Phi}_\theta \circ \mathbf{B}$ *is continuous*

*(ii)* $\mathbf{B}$ *is continuous*

*(iii)* $\mathrm{ran}(\mathbf{A})$ *is closed.*

*Proof.* Because $\mathbf{B}$ is a right inverse we have $(\mathbf{A} \circ \mathbf{B})(\mathbf{x}) = \mathbf{y}$ for all $\mathbf{y} \in \mathrm{ran}(\mathbf{A})$. Hence, the data consistence property (3.1) implies $\mathbf{A}(((\mathrm{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta) \circ \mathbf{B})(\mathbf{x})) = \mathbf{y}$, showing that $\mathbf{\Phi}_\theta \circ \mathbf{B}$ is a right inverse of $\mathbf{A}$. The implication (i) ⇒ (ii) follows from the identity $\mathbf{P}_{\mathbb{U}} \circ \mathbf{\Phi}_\theta \circ \mathbf{B} = \mathbf{B}$ and the continuity of the projection. The implication (ii) ⇒ (iii) follows from the continuity of $\mathbf{\Phi}_\theta$. Finally, the equivalence (ii) ⇔ (iii) has been established in Proposition 2.4. □

A benefit of non-linear right inverses defined by null space networks is that they can be adjusted to a given image class. A possible network training is given as follows

**Remark 3.4** (Possible training strategy)**.** *The null space network $\mathbf{\Phi}_\theta = \mathrm{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ can be trained to map elements in $\mathbb{M}$ to the elements from the desired class of images. For that purpose, select training data pairs $(\mathbf{x}_1, \mathbf{z}_1), \dots, (\mathbf{x}_N, \mathbf{z}_N)$ with $\mathbf{z}_i = \mathbf{B} \circ \mathbf{A}(\mathbf{x}_i)$ and minimize the regularized empirical risk,*

$$R_N\colon \Theta \to [0, \infty]\colon \theta \mapsto \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_i - \mathbf{\Phi}_\theta(\mathbf{z}_i)\|^2 + \|\theta\|_2^2 \,. \tag{3.3}$$

*Note that for our analysis it not required that (3.3) is exactly minimized. Any null space network $\mathbf{\Phi}_\theta$ where $\sum_{n=1}^{N} \|\mathbf{x}_i - \mathbf{\Phi}_\theta(\mathbf{z}_i)\|^2$ is small yields right inverse $\mathbf{\Phi}_\theta \mathbf{B}$ does a better job in estimating $\mathbf{x}_n$ from data $\mathbf{A}\mathbf{x}_n$ than the original right inverse $\mathbf{B}$.*

Proposition 3.3 implies that the solution of ill-posed problems by null space networks requires the use of stabilization methods similar to the case of classical methods. In the following subsection, we show that a the combination of null-space network with a regularization of $\mathbf{B}$ in fact yields a regularization method on a signal class related to the null-space network.

## 3.2 Convergence analysis

Throughout the following, let $\mathbf{\Phi}_\theta = \operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ be null-space network and $(\mathbf{R}_\delta)_{\delta>0}$ be a regularization method for (1.1) on the signal class $\mathbb{U}^* \subseteq \mathbb{U}$ with respect to the similarity measure $\mathcal{D}$ as introduced in Definition 2.7. As as illustrated in Figure 3.2, we consider the family $(\mathbf{\Phi}_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ of compositions of the regularization method with the null space network.

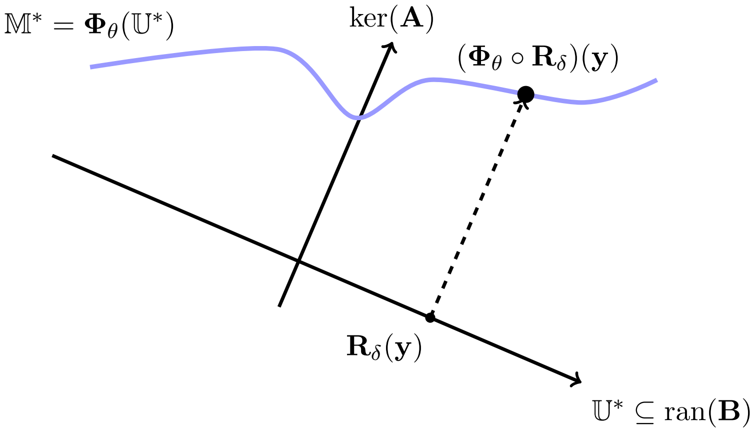
$\mathbb{M}^* = \mathbf{\Phi}_\theta(\mathbb{U}^*)$
$\ker(\mathbf{A})$
$(\mathbf{\Phi}_\theta \circ \mathbf{R}_\delta)(\mathbf{y})$
$\mathbf{R}_\delta(\mathbf{y})$
$\mathbb{U}^* \subseteq \operatorname{ran}(\mathbf{B})$

Figure 3.2: Linear Regularization $(\mathbf{R}_\delta)_{\delta>0}$ combined with a null space network $\mathbf{\Phi}_\theta = \operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$. We start with a linear regularization $\mathbf{R}_\delta \mathbf{y}$ and the nullspace network $\mathbf{\Phi}_\theta = \operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ adds missing parts along the null space $\ker(\mathbf{A})$.

**Theorem 3.5** (Regularizing null-space network)**.** *For a given null-space network $\mathbf{\Phi}_\theta = \operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ and a given regularization method $(\mathbf{R}_\delta)_{\delta>0}$ on the signal class $\mathbb{U}^*$, the family $(\mathbf{\Phi}_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ is regularization method for* (1.1) *on the signal class $\mathbf{\Phi}_\theta(\mathbb{U}^*)$ with respect to the similarity measure $\mathcal{D}$. We call $(\mathbf{\Phi}_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ a regularizing null-space network.*

*Proof.* Let $L$ be a Lipschitz constant of $\mathbf{\Phi}_\theta$ and recall $\mathbf{\Phi}_\theta = \operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$. For any $\mathbf{x} \in \mathbf{\Phi}_\theta(\mathbb{U}^*)$ and $\mathbf{y}^\delta \in \mathbb{Y}$ we have

$$\begin{aligned}\|\mathbf{x} - \mathbf{\Phi}_\theta \circ \mathbf{R}_\delta(\mathbf{y}^\delta)\| &= \|(\operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)(\mathbf{B} \circ \mathbf{A}(\mathbf{x})) - (\operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)(\mathbf{R}_\delta(\mathbf{y}^\delta))\| \\ &\leq L \|(\mathbf{B} \circ \mathbf{A})(\mathbf{x}) - \mathbf{R}_\delta(\mathbf{y}^\delta)\| \,.\end{aligned}$$

Here we have uses the identity $\mathbf{x} = (\operatorname{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)((\mathbf{B} \circ \mathbf{A})(\mathbf{x}))$ for $\mathbf{x} \in \operatorname{ran}(\boldsymbol{\Phi}_\theta)$. Consequently

$$\begin{aligned} &\sup \{ \| \mathbf{x} - (\boldsymbol{\Phi}_\theta \circ \mathbf{R}_\delta)(\mathbf{y}^\delta) \| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{y}^\delta, y) \leq \delta \} \\ &\qquad\qquad \leq L \sup \{ \| (\mathbf{B} \circ \mathbf{A})(\mathbf{x}) - \mathbf{R}_\delta(\mathbf{y}^\delta) \| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{y}^\delta, y) \leq \delta \} \to 0 \,. \end{aligned}$$

In particular, $(\boldsymbol{\Phi}_\theta \circ \mathbf{R}_\delta)_{\delta > 0}$ is regularization method for (1.1) on $\boldsymbol{\Phi}_\theta(\mathbb{U}^*)$ with respect to the similarity measure $\mathcal{D}$ □

In Hilbert spaces a wide class of regularizing reconstruction networks can be defined by regularizing filters.

**Example 3.6** (Regularizations defined by filters). *Let $\mathbb{X}$ and $\mathbb{Y}$ be Hilbert spaces. A family $(g_\alpha)_{\alpha>0}$ of functions $g_\alpha \colon [0, \|\mathbf{A}^* \circ \mathbf{A}\|] \to \mathbb{R}$ is called a regularizing filter if it satisfies*

- *For all $\alpha > 0$, $g_\alpha$ is piecewise continuous;*
- $\exists C > 0 \colon \sup \{ |\lambda g_\alpha(\lambda)| \mid \alpha > 0 \wedge \lambda \in [0, \|\mathbf{A}^* \circ \mathbf{A}\|] \} \leq C$.
- $\forall \lambda \in (0, \|\mathbf{A}^* \circ \mathbf{A}\|] \colon \lim_{\alpha \to 0} g_\alpha(\lambda) = 1/\lambda$.

*For a given regularizing filter $(g_\alpha)_{\alpha>0}$ define $\mathbf{B}_\alpha := g_\alpha(\mathbf{A}^* \circ \mathbf{A})\mathbf{A}^*$. Then for a suitable parameter choice $\alpha = \alpha(\delta, \mathbf{y})$ the family $(\mathbf{B}_{\alpha(\delta, \cdot)})_{\delta>0}$ is a regularization method on $\operatorname{ran}(\mathbf{A}^\dagger)$. Therefore, according to Theorem 3.5, the family $(\boldsymbol{\Phi}_\theta \circ \mathbf{B}_{\alpha(\delta, \cdot)})_{\delta>0}$ is a regularization method on $\boldsymbol{\Phi}_\theta(\operatorname{ran}(\mathbf{A}^\dagger))$. Note that in this setting one can derive derive quantitative error estimates (convergence rates); we the refer the interested reader to the original paper [45].*

## 3.3 Extensions

The regularizing null-space networks defined in Theorem 3.5 are of the form $\boldsymbol{\Phi}_\theta \circ \mathbf{R}_\delta$ where $\mathbf{R}_\delta$ is a classical regularization and $\boldsymbol{\Phi}_\theta$ only acts in null space of $\mathbf{A}$. In order to better accounts for noise, it is beneficial allowing the networks to modify $\mathbf{R}_\delta$ also on the complement $\mathbb{U}$.

**Definition 3.7** (Regularizing family of networks). Let $(\mathbf{R}_\delta)_{\delta>0}$ be a regularization method for (1.1) on the signal class $\mathbb{U}^* \subseteq \mathbb{U}$ with respect to the similarity measure $\mathcal{D}$ as introduced in Definition 2.7. A family $(\boldsymbol{\Phi}_{\theta(\delta)} \circ \mathbf{R}_\delta)_{\delta>0}$ is called regularizing family of networks if $(\boldsymbol{\Phi}_\theta)_{\theta \in \Theta}$ is a neural network such that the network functions $\boldsymbol{\Phi}_{\theta(\delta)} \colon \mathbb{X} \to \mathbb{X}$, for $\delta > 0$, are uniformly Lipschitz continuous and

$$\forall \mathbf{z} \in \operatorname{ran}(\mathbf{B}) \colon \qquad \lim_{\delta \to 0} \boldsymbol{\Phi}_{\theta(\delta)}(\mathbf{R}_\delta \circ \mathbf{A}(\mathbf{z})) = \mathbf{N}(\mathbf{z}) \,,$$

for some null-space network $\mathbf{N}$.

Regularizing families of networks have been introduced in [46] where it has been shown that a regularizing family of networks defines a regularization method together with convergence rates. Moreover, an example in the form of a data-driven extension of truncated SVD regularization has been given. In a finite dimensional setting, related extension of null-space networks named deep decomposition learning has been introduced in [12]. A combination of null-space learning with shearlet reconstruction for limited angle tomography has been introduced in [11]. In [16], a neural network based projection approach based on approximate data consistency sets has been studied. Relaxed versions of null-space networks, where approximate data consistency is incorporated via a confidence region or a soft penalty are proposed in [25, 30]. Finally, extensions of the null-space approach to non-linear problems are studied in [8].

# 4 The NETT approach

Let us recall that convex variational regularization of the inverse problem (1.1) consists in minimizing the generalized Tikhonov functional $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha\mathcal{R}(\mathbf{x})$, where $\mathcal{R}$ is a convex functional and $\mathcal{D}$ a similarity measure (see Section 2.2). The regularization term $\mathcal{R}$ is traditionally a semi-norm defined on a dense subspace of $\mathbb{X}$. In this section, we will extend this setup by using deep learning techniques with learned regularization functionals.

## 4.1 Learned regularization functionals

We assume that the regularizer takes the form

$$\forall \mathbf{x} \in \mathbb{X}\colon \quad \mathcal{R}(\mathbf{x}) = \mathcal{R}_\theta(\mathbf{x}) := \psi_\theta(\mathbf{\Phi}_\theta(\mathbf{x}))\,. \tag{4.1}$$

Here $\psi_\theta\colon \mathbf{\Xi} \to [0, \infty]$ is a scalar functional and $\mathbf{\Phi}_\theta(\,\cdot\,)\colon \mathbb{X} \to \mathbf{\Xi}$ a neural network where $\theta \in \Theta$, for some vector space $\Theta$ containing free parameters that can be adjusted by available training data. From the representation learning point of view [7], $\mathbf{\Phi}_\theta(\mathbf{x})$ can be interpreted as a learned representation of $\mathbf{x}$. It could be constructed in such a way that $\psi_\theta \circ \mathbf{\Phi}_\theta$ is minimal for a low dimensional manifold where the true signals $\mathbf{x}$ are clustered around. Finding such manifold for biomedical images has been an active research topic on manifold learning [18, 50]. Deep learning has also been used for this purpose [10]. A learned regularizer $\mathcal{R}_\theta = \psi_\theta \circ \mathbf{\Phi}_\theta$ reflects the statistics of the signal space, which penalizes those who deviate from the data manifold.

The similarity measure is taken as $\mathcal{D}_\theta\colon \mathbf{C} \times \mathbf{C} \to [0, \infty]$, where $\mathbf{C}$ is a conic closed subset in $\mathbb{Y}$. It is not necessarily symmetric in its arguments. One may take $\mathcal{D}_\theta(\mathbf{A}(\mathbf{x}), \mathbf{y})$ to be common hard-coded consistency measure such as $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}) = \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|^2$ or the

Kullback-Leibler divergence (which, among others, is used in emission tomography). On the other hand, it can be a learned measure, defined via a neural network. A learned consistency measure $\mathcal{D}_\theta$ reflects the statistics in the data (measurement) space. It can learn to reduce uncertainty in the data measurement process, e.g. by identifying non-functional transducers. It can also learn to reduce the error in the forward model [5]. Finally, it may encode the range description of the forward operator, which has not been successfully exploited in inverse problems by traditional methods. In summary, it can be said that learned consistency measures have potentially high impact in solving inverse problems.

Using the neural network based learned regularizer (4.1) and a learned discrepancy measure as discussed above, results in the following optimization problem

$$\arg\min_{\mathbf{x}\in\mathbb{D}} \mathcal{T}_\theta(\mathbf{x}) := \mathcal{D}_\theta(\mathbf{A}(\mathbf{x}), \mathbf{y}) + \alpha \mathcal{R}_\theta(\mathbf{x})\,. \tag{4.2}$$

Solving (4.2) can be seen as a neural networks-based variant of generalized Tikhonov regularization for solving (1.1). Following [32] we therefore call (4.2) the network Tikhonov (NETT) approach for solving inverse problems. Currently, there are two main approaches for integrating neural networks in the NETT approach (4.2): (T1) training the neural networks simultaneously with solving the optimization problem, and (T2) training the network independently before solving the optimization problem.

Approach (T1) fuses the data with a solution method of the optimization problem (4.2). The resulting neural networks, therefore, depend on the method to solve the optimization. This approach enforces the neural networks to learn particular representations that are useful for the chosen optimization technique. These representations will be called *solver-dependent*. The biggest advantage of this end-to-end approach is to provide a direct and relatively fast solution $\mathbf{x}$ for given new data $\mathbf{y}$. It is commonly realized by unrolling an iterative process [6]. The resulting neural network is a cascade of relatively small neural networks, each of them is, possibly a variant of, those appearing in the data consistency or regularization term. It is worth noting that the neural network does not aim for representation learning. Each layer or block serves to move the approximate solution closer to the exact solution. In contrast to typical iterative methods, each block in a unrolled neural network can be different from others. This is explained to speed up the convergence of the learned iterative method. The success of this approach is an interesting phenomenon that needs further investigation. The use of neural networks to implement and accelerate iterative methods to solve traditional regularization methods has been intensively studied. We refer the reader to [6] and the references contained therein.

The approach (T2) is more modular [32, 34] and results in smaller training problems and is closer to the meaning of representation learning. The training of the regularizer may or may not depend on the forward operator $\mathbf{A}$. In the former case, the resulting representation is called *model-dependent* while the latter is *model-independent*. Model-dependent representation seems to be crucial in inverse problem for two reasons. The first reason is that it aligns with the inverse problem (and better serves any solution approach). Secondly, in medical imaging applications, the training signals are often not the ground truth signals. They are normally obtained with a reconstruction method from high quality data. Therefore, while training the regularizer, one should also keep in mind the reconstruction mechanism of the training data. A possible approach is to first train a baseline neural network to learn model-independent representation. Then an additional block is added on top to train for model-dependent representation. This has been shown in [37] to be a very efficient strategy.

Let us mention that approach (T1) has richer literature than (T2), but less (convergence) analysis. In this section, we focus more on (T2), where we establish the convergence analysis and convergence rate in Section 4.2. This is an extension of our works [22, 37]. In Section 4.3, we review a few existing methods that are most relevant to our discussion, including some works in approach (T1). We also propose INDIE, which can be regarded as an operator inversion-free variant of the MODL technique [3] and can make better use of parallel computation.

## 4.2 Convergence analysis

Analysis for regularization with neural networks has been studied in [32] and [22]. In this section, we further investigate the issue. To this end, we consider the approach (T2), where the neural networks are trained independently of the optimization problem (4.2). That is, $\theta = \theta^*$ is already fixed a-priori. For the sake of simplicity, we will drop $\theta$ from the notation of $\mathcal{R}_\theta$ and $\mathcal{D}_\theta$. We focus on how the problem depends on the regularization parameter $\alpha$ and noise level $\delta$ in the data. Such analysis in standard situations is well-studied, see, e.g., [43]. However, we need to extend the analysis to more general cases to accommodate the fact that $\mathcal{R}$ comes from a neural network and is likely non-convex.

Let us make several assumptions on the regularizer and fidelity term.

**Condition 4.1.**

(A1) Network regularizer $\mathcal{R}$: $\mathbb{X} \to [0, \infty]$ satisfies

- (a) $0 \in \operatorname{dom}(\mathcal{R}) := \{\mathbf{x} \mid \mathcal{R}(\mathbf{x}) < \infty\}$;
- (b) $\mathcal{R}$ is lower semi-continuous;
- (c) $\mathcal{R}(\,\cdot\,)$ is coercive, that is $\mathcal{R}(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$.

(A2) Data consistency term $\mathcal{D}$: $\mathbf{C} \times \mathbf{C} \to [0, \infty]$ satisfies

- (a) $\operatorname{dom}(\mathcal{D}(0, \cdot)) = \mathbf{C}$;

(b) If $\mathcal{D}(\mathbf{y}_0, \mathbf{y}_1) < \infty$ and $\mathcal{D}(\mathbf{y}_1, \mathbf{y}_2) < \infty$ then $\mathcal{D}(\mathbf{y}_0, \mathbf{y}_2) < \infty$;

(c) $\mathcal{D}(\mathbf{y}, \mathbf{y}') = 0 \iff \mathbf{y} = \mathbf{y}'$;

(d) $\mathcal{D}(\mathbf{y}, \mathbf{y}') \geq C\|\mathbf{y} - \mathbf{y}'\|^2$ holds in any bounded subset of $\operatorname{dom}(\mathcal{D})$;

(e) For any $\mathbf{y}$, the function $\mathcal{D}(\mathbf{y}, \,\cdot\,)$ is continuous and coercive on its domain;

(f) The functional $(\mathbf{x}, \mathbf{y}) \mapsto \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y})$ is sequentially lower semi-continuous in the weak topology of $\mathbb{X}$ and strong topology of $\mathbb{Y}$.

For (A1), the coercivity condition (c) is the most restrictive. However, it can be accommodated. One such regularizer is proposed in our recent work [22] as follows

$$\mathcal{R}(\mathbf{x}) = \phi(\mathbf{E}(\mathbf{x})) + \frac{\beta}{2}\|\mathbf{x} - (\mathbf{D} \circ \mathbf{E})(\mathbf{x})\|_2^2 . \tag{4.3}$$

Here, $\mathbf{D} \circ \mathbf{E} \colon \mathbb{X} \to \mathbb{X}$ is an encoder-decoder network. The regularizer $\mathcal{R}$ is to enforce that a reasonable solution $\mathbf{x}$ satisfies $\mathbf{x} \simeq (\mathbf{D} \circ \mathbf{E})(\mathbf{x})$ and $\phi(\mathbf{E}(\mathbf{x}))$ is small. The term $\phi(\mathbf{E}(\mathbf{x}))$ implements learned prior knowledge, which is normally a sparsity measure in a non-linear basis. The second term $\|\mathbf{x} - (\mathbf{D} \circ \mathbf{E})(\mathbf{x})\|_2^2$ forces $\mathbf{x}$ to be close to data manifold $\mathcal{M}$. Their combination also guarantees the coercivity of the regularization functional $\mathcal{R}$. Another choice for $\mathcal{R}$ was suggested in [32].

For (A2), $\mathbf{C}$ is a conic set in $\mathbb{Y}$. For any $\mathbf{y} \in \mathbf{C}$, we define $\operatorname{dom}(\mathcal{D}(\mathbf{y}, \,\cdot\,)) = \{\mathbf{y}' \mid \mathcal{D}(\mathbf{y}, \mathbf{y}') < \infty\}$. The data consistency conditions in (A2) are flexible enough to be satisfied by a few interesting cases. The first example is that $\mathcal{D}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}\|^2$, which is probably the most popular data consistency measure. Another case is the Kullback-Leibler divergence, which reads as follows. Let $\mathbb{Y} = \mathbb{R}^n$ and $\mathbf{A} : \mathbb{X} \to \mathbb{Y}$ is a bounded linear positive operator.[1] Consider nonnegative cone $\mathbf{C} = \{(\mathbf{y}_1, \ldots, \mathbf{y}_n) \mid \forall i \colon \mathbf{y}_i \geq 0\}$. We define $\mathcal{D} \colon \mathbf{C} \times \mathbf{C} \to [0, \infty]$ by

$$\mathcal{D}(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^{n} \mathbf{y}_i \log \frac{\mathbf{y}_i}{\mathbf{y}_i'} + \mathbf{y}_i' - \mathbf{y}_i.$$

It is straight forward to check that Condition (A2) is satisfied in this case. In particular, item (d) has been verified in [41, Equation (13)].

To emphasize the fact that our data is the noisy version $\mathbf{y}^\delta$ of $\mathbf{y}$, we rewrite (4.2) as follows

$$\operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{D}} \mathcal{T}_{\mathbf{y}^\delta, \alpha}(\mathbf{x}) := \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha \mathcal{R}(\mathbf{x}). \tag{4.4}$$

Here, $\mathbb{D}$ is a weakly-closed conic set in $\mathbb{X}$ such that $\mathbf{A}(\mathbb{D}) \subseteq \mathbf{C}$.

**Theorem 4.2** (Well-posedness and convergence). *Let Condition 4.1 be satisfied. Then the following assertions hold true:*

*(a)* Existence: *For all $\mathbf{y} \in \mathbf{C}$ and $\alpha > 0$, there exists a minimizer of $\mathcal{T}_{\alpha;\mathbf{y}}$ in $\mathbb{D}$.*

[1] $\mathbf{A}$ is positive if: $\mathbf{y} \geq 0 \Rightarrow \mathbf{A}\mathbf{y} \geq 0$.

*(b)* Stability: *If $\mathbf{y}_k \to \mathbf{y}$, $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) < \infty$ and $\mathbf{x}_k \in \arg\min \mathcal{T}_{\alpha;\mathbf{y}_k}$, then weak accumulation points of $(\mathbf{x}_k)_{k\in\mathbb{N}}$ exist and are minimizers of $\mathcal{T}_{\alpha;\mathbf{y}}$.*

*(c)* Convergence: *Let $\mathbf{y} \in \operatorname{ran}(\mathbf{A}) \cap \mathbf{C}$ and $(\mathbf{y}_k)_{k\in\mathbb{N}}$ satisfy $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) \le \delta_k$ for some sequence $(\delta_k)_{k\in\mathbb{N}} \in (0,\infty)^{\mathbb{N}}$ with $\delta_k \to 0$. Suppose $\mathbf{x}_k \in \arg\min_{\mathbf{x}} \mathcal{T}_{\mathbf{y}_k,\alpha(\delta_k)}(\mathbf{x})$, and let the parameter choice $\alpha\colon (0,\infty) \to (0,\infty)$ satisfy*

$$\lim_{\delta\to 0} \alpha(\delta) = \lim_{\delta\to 0} \frac{\delta}{\alpha(\delta)} = 0\,. \tag{4.5}$$

*Then the following holds:*

*(1) All weak accumulation points of $(\mathbf{x}_k)_{k\in\mathbb{N}}$ are $\mathcal{R}$-minimizing solutions of the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$;*

*(2) $(\mathbf{x}_k)_{k\in\mathbb{N}}$ has at least one weak accumulation point $\mathbf{x}^\dagger$;*

*(3) Every subsequence $(\mathbf{x}_{k(n)})_{n\in\mathbb{N}}$ that weakly converges to $\mathbf{x}^\dagger$ satisfies $\mathcal{R}(\mathbf{x}_{k(n)}) \to \mathcal{R}(\mathbf{x}^\dagger)$;*

*(4) If the $\mathcal{R}$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ is unique, then $\mathbf{x}_k \rightharpoonup \mathbf{x}^\dagger$.*

Before starting the proof, we recall that $\mathbf{x}^\dagger$ is an $\mathcal{R}$-minimizing solution of the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ if $\mathbf{x}^\dagger \in \arg\min \{\mathcal{R}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{D} \wedge \mathbf{A}\mathbf{x} = \mathbf{y}\}$.

*Proof.* (a) Firstly, we observe that $c := \inf_{\mathbf{x}} \mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) \le \mathcal{T}_{\mathbf{y},\alpha}(0) < \infty$. Let $(\mathbf{x}_k)_k$ be a sequence such that $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}_k) \to c$. There exists $M > 0$ such that $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}_k) \le M$, which implies $\alpha\mathcal{R}(\mathbf{x}_k) \le M$. Since $\mathcal{R}$ is coercive, we obtain $(\mathbf{x}_k)_k$ is bounded. By passing into a subsequence, $\mathbf{x}_{k_i} \rightharpoonup \mathbf{x}^* \in \mathbb{D}$. Due to the lower semi-continuity of $\mathcal{T}_{\alpha,\,\cdot\,}(\,\cdot\,)$, we have $\mathbf{x}^* \in \arg\min \mathcal{T}_{\mathbf{y},\alpha}$.

(b) Since $\mathbf{x}_k \in \arg\min \mathcal{T}_{\mathbf{y},\alpha}$, it holds $\mathcal{T}_{\mathbf{y}_k,\alpha}(\mathbf{x}_k) \le \mathcal{T}_{\mathbf{y}_k,\alpha}(0) = \mathcal{D}(0, \mathbf{y}_k) + \alpha\mathcal{R}(0)$. Thanks to the continuity of $\mathcal{D}(0,\,\cdot\,)$ on $\mathbf{C}$, $(\mathcal{D}(0, \mathbf{y}_k))_k$ is a bounded sequence. Therefore, $\alpha\mathcal{R}(\mathbf{x}_k) \le \mathcal{T}_{\mathbf{y}_k,\alpha}(\mathbf{x}_k) \le M$, for a constant $M$ independent of $k$. Since $\mathcal{R}$ is coercive, $(\mathbf{x}_k)_k$ is bounded and hence has a weakly-convergent subsequence $\mathbf{x}_{k_i} \rightharpoonup \mathbf{x}^\dagger$.

Let us now prove that $\mathbf{x}^\dagger$ is a minimizer of $\mathcal{T}_{\mathbf{y},\alpha}$. Since $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x})$ is lower semi-continuous in $\mathbf{x}$ and $\mathbf{y}$,

$$\liminf_{k_i\to\infty} \mathcal{T}_{\mathbf{y}_{k_i},\alpha}(\mathbf{x}_{k_i}) \ge \mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}^\dagger). \tag{4.6}$$

On the other hand, let $\mathbf{x} \in \mathbb{D}$ be such that $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) < \infty$. We obtain $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}) < \infty$ and $\mathcal{R}(\mathbf{x}) < \infty$. Condition (A2)(d) and $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) < \infty$ give $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}_k) < \infty$. That is, $\mathbf{y}_k \in \operatorname{dom}(\mathcal{D}(\mathbf{A}(\mathbf{x}), \cdot)$. The continuity of $\mathcal{D}(\mathbf{A}(\mathbf{x}), \cdot)$ on its domain implies $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}_k) \to \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y})$. Since $\mathbf{x}_k$ is the minimizer of $\mathcal{T}_{\mathbf{y}_k,\alpha}$, $\mathcal{T}_{\mathbf{y}_k,\alpha}(\mathbf{x}_k) \le \mathcal{T}_{\mathbf{y}_k,\alpha}(\mathbf{x})$. Taking the limit, we obtain $\limsup_k \mathcal{T}_{\mathbf{y}_k,\alpha}(\mathbf{x}_k) \le \mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x})$. From (4.6), $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}^\dagger) \le \mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{D}$. We conclude that $\mathbf{x}^\dagger$ is a minimizer of $\mathcal{T}_{\mathbf{y},\alpha}$.

(c) We prove the properties item by item.

(1) Since $\mathbf{y} \in \mathbb{R}(\mathbf{A})$, we can pick be a solution $\bar{\mathbf{x}}$ of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. We have

$$\mathcal{D}(\mathbf{A}(\mathbf{x}_k), \mathbf{y}_k) + \alpha_k \mathcal{R}(\mathbf{x}_k) \leq \mathcal{D}(\mathbf{y}, \mathbf{y}_k) + \alpha_k \mathcal{R}(\bar{\mathbf{x}}) \leq \delta_k + \alpha_k \mathcal{R}(\bar{\mathbf{x}}). \tag{4.7}$$

Assume that $\mathbf{x}^\dagger$ is a weak accumulation point of $\mathbf{x}_k$, then

$$\mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}) \leq \lim_{k\to\infty} \inf \mathcal{D}(\mathbf{A}(\mathbf{x}_k), \mathbf{y}_k) \leq \lim_{k\to\infty} \inf(\delta_k + \alpha_k \mathcal{R}(\bar{\mathbf{x}})) = 0.$$

Therefore, $\mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}) = 0$ or $\mathbf{A}(\mathbf{x}^\dagger) = \mathbf{y}$. Moreover, $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\bar{\mathbf{x}})$, which implies $\mathcal{R}(\mathbf{x}^\dagger) \leq \liminf \mathcal{R}(\mathbf{x}_k) \leq \mathcal{R}(\bar{\mathbf{x}})$. Since this holds for all possible solution $\bar{\mathbf{x}}$ of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, we conclude that $\mathbf{x}^\dagger$ is a $\mathcal{R}$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$.

(2) Using again the inequality $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\bar{\mathbf{x}})$ and $\mathcal{R}$ is coercive, we obtain $\{\mathbf{x}_k\}$ is bounded. Therefore, $\{\mathbf{x}_k\}$ has a weak accumulation point $\mathbf{x}^\dagger$.

(3) Using (4.7) again for $\bar{\mathbf{x}} = \mathbf{x}^\dagger$, we obtain $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\mathbf{x}^\dagger)$, which gives $\limsup_k \mathcal{R}(\mathbf{x}_k) \leq \mathcal{R}(\mathbf{x}^\dagger)$. This together with the fact that $\mathcal{R}$ is lower semi-continuous gives $\mathcal{R}(\mathbf{x}_{k(n)}) \to \mathcal{R}(\mathbf{x}^\dagger)$.

(4) The last conclusion follows straight forwardly from the above three. □

Let us proceed to obtain some convergence results in the norm. Following [32], we introduce the absolute Bregman distance.

**Definition 4.3** (Absolute Bregman distance). Let $\mathbb{F} \colon \mathbb{D} \subseteq \mathbb{X} \to \mathbb{R}$ be Gâteaux differentiable at $\mathbf{x} \in \mathbb{X}$. The *absolute Bregman distance* $\mathbf{\Delta}_{\mathbb{F}}(\,\cdot\,, \mathbf{x}) \colon \mathbb{D} \to [0, \infty]$ with respect to $\mathbb{F}$ at $\mathbf{x}$ is defined by

$$\forall \tilde{\mathbf{x}} \in \mathbb{X}\colon \quad \mathbf{\Delta}_{\mathbb{F}}(\tilde{\mathbf{x}}, \mathbf{x}) := \left|\mathbb{F}(\tilde{\mathbf{x}}) - \mathbb{F}(\mathbf{x}) - \mathbb{F}'(\mathbf{x})(\tilde{\mathbf{x}} - \mathbf{x})\right|. \tag{4.8}$$

Here $\mathbb{F}'(\mathbf{x})$ denotes the Gâteaux derivative of $\mathbb{F}$ at $\mathbf{x}$.

From Theorem 4.2 we can conclude convergence of $\mathbf{x}_\alpha^\delta$ to the exact solution in the absolute Bregman distance $\mathbf{\Delta}_{\mathcal{R}}$. Below we show that this implies strong convergence under some additional assumption on the regularization functional. For this purpose, we define the concept of total non-linearity, which was introduce in [32].

**Definition 4.4** (Total non-linearity). Let $\mathbb{F} \colon \mathbb{D} \subseteq \mathbb{X} \to \mathbb{R}$ be Gâteaux differentiable at $\mathbf{x} \in \mathbb{D}$. We define the *modulus of total non-linearity* of $\mathbb{F}$ at $\mathbf{x}$ as $\nu_{\mathbb{F}}(\mathbf{x}, \cdot) : [0, \infty) \to [0, \infty]$,

$$\forall t > 0\colon \quad \nu_{\mathbb{F}}(x, t) := \inf \left\{ \mathbf{\Delta}_{\mathbb{F}}(\tilde{\mathbf{x}}, \mathbf{x}) \mid \tilde{\mathbf{x}} \in \mathbb{D} \wedge \|\tilde{\mathbf{x}} - \mathbf{x}\| = t \right\}. \tag{4.9}$$

The function $\mathbb{F}$ is called *totally non-linear* at $\mathbf{x}$ if $\nu_{\mathbb{F}}(\mathbf{x}, t) > 0$ for all $t \in (0, \infty)$.

The following result, due to [32], connects the convergence in absolute Bregman distance and in norm

**Proposition 4.5.** *For $\mathbf{F}\colon D \subseteq \mathbb{X} \to \mathbb{R}$ and and any $\mathbf{x} \in D$, the followings are equivalent:*

*(i) The function $\mathbf{F}$ is totally nonlinear at at $\mathbf{x}$;*

*(ii) $\forall (\mathbf{x}_n)$: $(\lim_{n\to\infty} \mathcal{B}_{\mathbf{F}}(\mathbf{x}_n, \mathbf{x}) = 0 \wedge (\mathbf{x}_n)$ bounded) $\Rightarrow \lim_{n\to\infty} \|\mathbf{x}_n - \mathbf{x}\| = 0$.*

As a consequence, we have the following convergence result in the norm topology.

**Theorem 4.6** (Strong Convergence)**.** *Assume that $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has a solution, let $\mathcal{R}_\theta$ be totally nonlinear at all $\mathcal{R}_\theta$-minimizing solutions of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, and let $(\mathbf{x}_k)_{k\in\mathbb{N}}$, $(y_k)_{k\in\mathbb{N}}$, $(\alpha_k)_{k\in\mathbb{N}}$, $(\delta_k)_{k\in\mathbb{N}}$ be as in Theorem 4.2. Then there is a subsequence $(\mathbf{x}_{k(\ell)})_{\ell\in\mathbb{N}}$ of $(\mathbf{x}_k)_{k\in\mathbb{N}}$ and an $\mathcal{R}_\theta$-minimizing solution $\mathbf{x}^\dagger$ of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ such that $\lim_{\ell\to\infty} \|\mathbf{x}_{k(\ell)} - \mathbf{x}^\dagger\| = 0$. Moreover, if the $\mathcal{R}_\theta$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ is unique, then $\mathbf{x} \to \mathbf{x}^\dagger$ in the norm topology.*

We now focus on the convergence rate. To this end, we make the following assumptions. following assumptions:

(B1) $\mathbb{Y}$ is a finite dimensional space;

(B2) $\mathcal{R}$ is coercive and weakly sequentially lower semi-continuous;

(B3) $\mathcal{R}$ is Lipschitz;

(B4) $\mathcal{R}$ is Gâteaux differentiable.

The most restrictive condition in the above list is that $\mathbf{A}$ has finite-dimensional range. However, this assumption holds true in practical applications such as sparse data tomography, which is a main focus of deep learning techniques for inverse problems. For infinite dimensional space result see [32].

We start our analysis with the following result.

**Proposition 4.7.** *Let (B1)-(B4) be satisfied and assume that $\mathbf{x}^\dagger$ is an $\mathcal{R}$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. Then there exists a constant $C > 0$ such that*

$$\forall \mathbf{x} \in \mathbb{X}\colon \quad \mathbf{\Delta}_{\mathcal{R}}(\mathbf{x}, \mathbf{x}^\dagger) \leq \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + C\|\mathbf{A}(\mathbf{x}) - \mathbf{A}(\mathbf{x}^\dagger)\| .$$

The proof follows [37]. We present it here for the sake of completeness.

*Proof.* Let us first prove that for some constant $\gamma \in (0, \infty)$ it holds

$$\forall \mathbf{x} \in \mathbb{X}\colon \quad \mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \leq \gamma \|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\| . \tag{4.10}$$

Indeed, let $\mathbf{P}$ be the orthogonal projection onto $\ker(\mathbf{A})$ and define $\mathbf{x}_0 = (\mathbf{x}^\dagger - \mathbf{P}(\mathbf{x}^\dagger)) + \mathbf{P}(\mathbf{x})$. Then, we have $\mathbf{A}(\mathbf{x}_0) = \mathbf{A}(\mathbf{x}^\dagger)$ and $\mathbf{x} - \mathbf{x}_0 \in \ker(\mathbf{A})^\perp$. Since the restricted operator

$\mathbf{A}|_{\ker(\mathbf{A})^\perp} \colon \ker(\mathbf{A})^\perp \to \mathbb{Y}$ is injective and has finite-dimensional range, it is bounded from below by a constant $\gamma_0$. Therefore,

$$\|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\| = \|\mathbf{A}(\mathbf{x}_0) - \mathbf{A}(\mathbf{x})\| = \|\mathbf{A}(\mathbf{x}_0 - \mathbf{x})\| \geq \gamma_0 \|\mathbf{x}_0 - \mathbf{x}\| \,. \tag{4.11}$$

On the other hand, since $\mathbf{x}^\dagger$ is the $\mathcal{R}$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ and $\mathcal{R}$ is Lipschitz, we have $\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \leq \mathcal{R}(\mathbf{x}_0) - \mathcal{R}(x) \leq L\|\mathbf{x}_0 - \mathbf{x}\|$. Together with (4.11) we obtain (4.10).

Next we prove that there is a constant $\gamma_1$ such that

$$\left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x} \right\rangle \leq \gamma_1 \|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\| \,. \tag{4.12}$$

Indeed, since $\mathbf{x}^\dagger$ is an $\mathcal{R}$-minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, we obtain $\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x}_0 \rangle \leq 0$. Therefore,

$$\begin{aligned}
\left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x} \right\rangle &= \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x}_0 \right\rangle + \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}_0 - \mathbf{x} \right\rangle \\
&\leq \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}_0 - \mathbf{x} \right\rangle \leq \|\mathcal{R}'(\mathbf{x}^\dagger)\| \|\mathbf{x}_0 - \mathbf{x}\|.
\end{aligned}$$

Using (4.11) again we obtain (4.12).

To finish the proof, we consider two cases:

- $\mathcal{R}(\mathbf{x}^\dagger) \leq \mathcal{R}(\mathbf{x}) \Rightarrow |\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x})| = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger)$
- $\mathcal{R}(\mathbf{x}^\dagger) \geq \mathcal{R}(\mathbf{x}) \Rightarrow |\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x})| = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + 2(\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}))$.

Therefore, using (4.10) and (4.12), we obtain

$$\begin{aligned}
\mathbf{\Delta}_{\mathcal{R}}(\mathbf{x}, \tilde{\mathbf{x}}) &\leq |\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x})| + |\left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x} - \mathbf{x}^\dagger \right\rangle| \\
&\leq \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + (2\gamma + \gamma_1)\|\mathbf{A}(\mathbf{x}) - \mathbf{A}(\mathbf{x}^\dagger)\| \,,
\end{aligned}$$

which concludes our proof with $C := 2\gamma_0 + \gamma_1$. □

Here is our convergence rates result, which is an extension of [37, Theorem 3.1].

**Theorem 4.8** (Convergence rates results). *Let (B1)-(B4) be satisfied and suppose $\alpha \sim \delta$. Then $\mathbf{\Delta}_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) = \mathcal{O}(\delta)$ as $\delta \to 0$.*

*Proof.* From Proposition 4.7, we obtain

$$\begin{aligned}
\alpha \mathbf{\Delta}_{\mathbf{F}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) &\leq \alpha\mathcal{R}(\mathbf{x}_\alpha^\delta) - \alpha\mathcal{R}(\mathbf{x}^\dagger) + C\alpha\|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{A}(\mathbf{x}^\dagger)\| \\
&= \mathcal{T}_{\alpha,\delta}(\mathbf{x}_\alpha^\delta) - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) - \left(\mathcal{T}_{\alpha,\delta}(\mathbf{x}^\dagger) - \mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}^\delta)\right) \\
&+ C\alpha\|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{A}(\mathbf{x}^\dagger)\| \\
&\leq \delta^2 + C\alpha\delta - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) + C\alpha\|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{y}^\delta\|
\end{aligned}$$

$$\leq \quad \delta^2 + C\alpha\delta - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) + C\alpha\sqrt{\mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta)}\,.$$

Cauchy's inequality gives $\alpha\mathbf{\Delta}_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) \leq \delta^2 + C\alpha\delta + C^2\alpha^2/4$. For $\alpha \sim \delta$, we easily conclude $\mathbf{\Delta}_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) = \mathcal{O}(\delta)$. □

## 4.3 Related methods

The use of neural networks as regularizers or similarity measures is an active research directions. Many interesting works have been done. We briefly review several techniques: variational networks [28], deep cascaded networks [29, 44] and the MODL approach [3]. Further, we propose INDIE as a new operator-inversion free variant of MODL. As opposed to the discussion in Section 4.2, these works make use of the approach (T1): employing solver-dependent training. Finally, we will discuss a synthesis variant of the NETT framework.

**Variational networks:** Variational networks [28] connect variational methods and deep learning. They are based on the fields of experts model [42] and consider the Tikhonov functional

$$\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) = \sum_{c=1}^{N_c} \mathcal{T}_c(\mathbf{x}) \coloneqq \sum_{c=1}^{N_c} \Big( \sum_j \sum_i \phi_i^c((\bar{K}_i^c \mathbf{x})_j) + \alpha \sum_j \sum_i \psi_i^c((K_i^c(\mathbf{A}(\mathbf{x}) - \mathbf{y}))_j) \Big)\,,$$

where $\bar{K}_i^c$ and, $K_i^c$ are learnable convolutional operators, and $\phi_i, \psi_i$ are learnable functionals. Alternating gradient descent method for minimizing $\mathcal{T}_{\mathbf{y},\alpha}$ provides the update formula

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla_\theta \mathcal{T}_{c(n)}(\mathbf{x}_n) \quad \text{where } c(n) = 1 + (n \mod N_c)\,. \tag{4.13}$$

Direct calculations show $\nabla_\theta \mathcal{T}_c(\mathbf{x}) = \sum_i (\bar{K}_i^c)^T (\phi_i^c)'(K_i^c \mathbf{x}) + \mathbf{A}^T \sum_i (K_i^c)^T (\psi_i^c)'(\bar{K}_i^c(\mathbf{A}(\mathbf{x}) - \mathbf{y}))$. Minimizing the $\mathcal{T}_{\mathbf{y},\alpha}$ is then replaced by training the neural network that consists of a $L$ blocks realizing the iterative update (4.13).

**Network cascades:** Deep network cascades [29, 44] alternate between the application of post-processing networks and so-called data consistency layers. The data consistency condition proposed in [29] for sparse data problems $\mathbf{A} = \mathbf{S} \circ \mathbf{A}_{\mathrm{F}}$ where $\mathbf{S}$ is a sampling operator and $\mathbf{A}_{\mathrm{F}}$ a full data forward operator (such as the fully sampled Radon transform) takes the form

$$\mathbf{x}_{n+1} = \mathbf{B}_{\mathrm{F}} \left( \arg\min_{\mathbf{z}} \|\mathbf{z} - \mathbf{A}_{\mathrm{F}}(\mathcal{N}_{\theta(n)}(\mathbf{x}_n))\|_2^2 + \alpha \|\mathbf{y} - \mathbf{S}(\mathbf{z})\|_2^2 \right)\,, \tag{4.14}$$

with initial reconstruction $\mathbf{x}_0 = (\mathbf{B}_{\mathrm{F}} \circ \mathbf{S}^*)(\mathbf{y})$, where $\mathbf{B}_{\mathrm{F}} \colon \mathbb{Y} \to \mathbb{X}$ is a reconstruction method for the full data forward operator and $\mathcal{N}_{\theta(n)}$ are networks. For example, in MRT the operator $\mathbf{B}_{\mathrm{F}}$ is the inverse Fourier transform [44] and in CT the operator $\mathbf{B}_{\mathrm{F}}$ can be implemented by the filtered backprojection [29]. The resulting neural network consists of $L$ steps of (4.14) that can be trained end-to-end.

**MODL approach:** The model based deep learning (MODL) approach of [3] starts with the Tikhonov functional $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) = \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|_2^2 + \alpha\|\mathbf{x} - \mathcal{N}_\theta(\mathbf{x})\|_2^2$ where $\mathcal{N}_\theta(\mathbf{x})$ is interpreted as denoising network. By designing $\mathcal{N}_\theta$ as a convolutional block, then $\mathbf{x} - \mathcal{N}_\theta(\mathbf{x})$ is a small residual network [24]. The authors of [3] proposed the following heuristic iterative scheme $\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|^2 + \alpha\|\mathbf{x} - \mathcal{N}_\theta(\mathbf{x}_n)\|_2^2$ based on $\mathcal{T}_{\mathbf{y},\alpha}$ whose closed-form solution is

$$\mathbf{x}_{n+1} = (\mathbf{A}^\intercal\mathbf{A} + \alpha\,\mathrm{Id})^{-1}(\mathbf{A}^\intercal\mathbf{y} + \alpha\,\mathcal{N}_\theta(\mathbf{x}_n))\,. \tag{4.15}$$

Concatenating these steps together one arrives at a deep neural network. Similar to network cascades, each block (4.15) consists of a trainable layer $\mathbf{z}_n = \mathbf{A}^\intercal\mathbf{y} + \alpha\mathcal{N}_\theta(\mathbf{x}_n)$ and a non-trainable data consistency layer $\mathbf{x}_{n+1} = (\mathbf{A}^\intercal\mathbf{A} + \lambda\,\mathrm{Id})^{-1}(\mathbf{z}_n)$.

**INDIE approach:** Let us present an alternative to the above procedures, inspired by [14]. Namely we propose the iterative update

$$\begin{aligned}
&\mathbf{x}_{n+1} = \arg\min \mathcal{L}_n(\mathbf{x})\\
&\mathcal{L}_n(\mathbf{x}) \coloneqq \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|^2 + \alpha\|\mathbf{x} - \mathcal{N}_\theta(\mathbf{x}_n)\|^2 + C\|\mathbf{x} - \mathbf{x}_n\|^2 - \|\mathbf{A}(\mathbf{x} - \mathbf{x}_n)\|^2\,.
\end{aligned}$$

Here the constant $C > 0$ is an upper bound for the operator norm $\|\mathbf{A}\|$. Elementary manipulations show the identity

$$\begin{aligned}
\mathcal{L}_n(\mathbf{x}) = -2\Big\langle \mathbf{A}^\intercal(\mathbf{y} - \mathbf{A}(\mathbf{x}_n)) + \alpha\mathcal{N}_\theta(\mathbf{x}_n) + C\mathbf{x}_n, \mathbf{x}\Big\rangle&\\
+ (\alpha + C)\|\mathbf{x}\|^2 - (\alpha\|\mathcal{N}_\theta(\mathbf{x}_n)\| + C\|\mathbf{x}_n\|^2) - \|\mathbf{A}(\mathbf{x}_n)\|^2 + \|\mathbf{y}\|^2\,.&
\end{aligned}$$

The minimizer of $\mathcal{L}_n$ can therefore be computed explicitly by setting the gradient of the latter expression to zero. This results in the proposed network block

$$\mathbf{x}_{n+1} = \frac{1}{\alpha + C}\Big(\mathbf{A}^\intercal(\mathbf{y} - \mathbf{A}(\mathbf{x}_n)) + \alpha\mathcal{N}_\theta(\mathbf{x}_n) + C\mathbf{x}_n\Big)\,. \tag{4.16}$$

This results at a deep neural network similar to the MODL iteration. However, each block in (4.16) is clearly simpler than the blocks in (4.15). In fact, opposed to MODL our proposed learned iterative scheme does not require costly matrix inversion. We name the resulting iteration INDIE (for **in**version-free **d**eep **ite**rative) cascades. We consider the numerical comparison of MODL and INDIE as well as the theoretical analysis of both architectures interesting lines of future research.

**Learned synthesis regularization:** Let us finish this section by pointing out that regularization by neural network is not restricted to the form (4.2). For example, one can consider the synthesis version, which reads [38]

$$\mathbf{x}^{\mathrm{syn}} = \mathbf{D}_\theta\Big(\arg\min_{\xi} \|\mathbf{A} \circ \mathbf{D}_\theta(\xi) - \mathbf{y}\|^2 + \alpha \sum_{\lambda \in \Lambda} \omega_\lambda |\xi_\lambda|^p\Big) , \tag{4.17}$$

where $\Lambda$ is a countable set, $1 \leq p < 2$, and $\mathcal{D}_\theta \colon \ell^2(\Lambda) \to \mathbb{X}$ is a learned operator that performs nonlinear synthesis of $\mathbf{x}$. Rigorous analysis of the above formulation was derived in [38].

Finally, note that one can generalize the frameworks (4.2) and (4.17) by allowing the involved neural networks to depend on the regularization parameter $\alpha$ or the noise level $\delta$. The dependence on $\alpha$ has been studied in, for example, [38]. The dependence on $\delta$ can be realized by mimicking the Morozov's stopping criteria when training the neural networks, either independently or together with the optimization problem. In the later case, $\delta$ can help to decide the depth of the unrolled neural network.

## 5 Conclusion and outlook

Inverse problems are central to the solving a wide range of important practical problems within and outside of imaging and computer vision. Inverse problems are characterized by the ambiguity and instability of their solution. Therefore, stabilizing solution methods based on regularization techniques are necessary to solve them in reasonable way. In recent years, neural networks and deep learning have emerged as the rising stars for the solution of inverse problems. In this chapter, we have developed the mathematical foundations for solving inverse problems with deep learning. In addition, we have shown stability and convergence for selected neural networks to solve inverse problems. The investigated methods, which combine the strengths of both worlds, are regularizing null-space networks and the NETT (Network-Tikhonov) approach for inverse problems.

## References

[1] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Probl.*, 10(6):1217–1229, 1994.

[2] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.*, 33(12):124007, 2017.

[3] H. K. Aggarwal, M. P. Mani, and M. Jacob. MoDL: model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imaging*, 38(2):394–405, 2018.

[4] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi. Learning activation functions to improve deep neural networks. *arXiv:1412.6830*, 2014.

[5] R. Aljadaany, D. K. Pal, and M. Savvides. Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, pages 10235–10244, 2019.

[6] S. Arridge, P. Maass, O. Öktem, and C. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[7] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal.*, 35(8):1798–1828, 2013.

[8] Y. E. Boink, M. Haltmeier, S. Holman, and J. Schwab. Data-consistent neural networks for solving nonlinear inverse problems. *arXiv:2003.11253*, 2020.

[9] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546, 2017.

[10] T. Brosch, R. Tam, et al. Manifold learning of brain MRIs by deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 633–640. Springer, 2013.

[11] T. A. Bubba, G. Kutyniok, M. Lassas, M. Maerz, W. Samek, S. Siltanen, and V. Srinivasan. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.*, 35(6):064002, 2019.

[12] D. Chen and M. E. Davies. Deep decomposition learning for inverse imaging problems. *arXiv preprint arXiv:1911.11028*, 2019.

[13] Ti. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Adv. Neural Inf. Process. Syst.*, pages 6571–6583, 2018.

[14] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

[15] S. Dittmer, T. Kluth, P. Maass, and D. O. Baguer. Regularization by architecture: A deep prior approach for inverse problems. *J. Math. Imaging Vis.*, pages 1–15, 2019.

[16] S. Dittmer and P. Maass. A projectional ansatz to reconstruction. *arXiv:1907.04675*, 2019.

[17] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Kluwer Academic Publishers Group, Dordrecht, 1996.

[18] M. Georg, R. Souvenir, A. Hope, and R. Pless. Manifold learning for 4D CT reconstruction of the lung. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, pages 1–8. IEEE, 2008.

[19] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. 13th international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[20] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[21] M. Grasmair, M. Haltmeier, and O. Scherzer. Sparse regularization with $l^q$ penalty term. *Inverse Probl.*, 24(5):055020, 13, 2008.

[22] M. Haltmeier, L. Nguyen, D. Obmann, and J. Schwab. Sparse $\ell^q$-regularization of inverse problems with deep learning, 2019.

[23] Y. Han, J. J. Yoo, and J. C. Ye. Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis, 2016. http://arxiv.org/abs/1611.06391.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, pages 770–778, 2016.

[25] Y. Huang, A. Preuhs, M. Manhart, G. Lauritsch, and A. Maier. Data consistent ct reconstruction from insufficient data with learned prior images. *arXiv:2005.10034*, 2020.

[26] V. K. Ivanov, V. V. Vasin, and V. P. Tanana. *Theory of linear ill-posed problems and its applications*. Inverse and Ill-posed Problems Series. VSP, Utrecht, second edition, 2002. Translated and revised from the 1978 Russian original.

[27] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.*, 26(9):4509–4522, 2017.

[28] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock. Variational networks: connecting variational methods and deep learning. In *German Conference on Pattern Recognition*, pages 281–293. Springer, 2017.

[29] A Kofler, M Haltmeier, C Kolbitsch, M Kachelrieß, and M Dewey. A U-Nets cascade for sparse view computed tomography. In *Proc. 1st Workshop on Machine Learning for Medical Image Reconstruction*, pages 91–99. Springer, 2018.

[30] A. Kofler, M. Haltmeier, T. Schaeffter, M. Kachelrieß, M Dewey, C. Wald, and C. Kolbitsch. Neural networks-based regularization of large-scale inverse problems in medical imaging. *Phys. Med. Biol.*, 2020, in press.

[31] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[32] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. NETT: Solving inverse problems with deep neural networks. *Inverse Problems*, 2020. in press.

[33] J. Lindenstrauss and L. Tzafriri. On the complemented subspaces problem. *Isr. J. Math.*, 9(2):263–269, 1971.

[34] S. Lunz, O. Öktem, and C. Schönlieb. Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems 31*, pages 8507–8516. 2018.

[35] M. Mardani, E. Gong, J. Y. Cheng, S. S. Vasanawala, G. Zaharchuk, L. Xing, and J. M. Pauly. Deep generative adversarial neural networks for compressive sensing MRI. *IEEE Trans. Med. Imag.*, 38(1):167–179, 2018.

[36] M. Z. Nashed. Inner, outer, and generalized inverses in banach and hilbert spaces. *Numer. func. anal. opt.*, 9(3-4):261–325, 1987.

[37] D. Obmann, L. Nguyen, J. Schwab, and M. Haltmeier. Sparse aNETT for solving inverse problems with deep learning. *arXiv:2004.09565*, 2020.

[38] D. Obmann, J. Schwab, and M. Haltmeier. Deep synthesis regularization of inverse problems. *arXiv:2002.00155*, 2020.

[39] R. S. Phillips. On linear transformations. *Trans. Am. Math. Soc.*, 48(3):516–541, 1940.

[40] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv:1710.05941*, 2017.

[41] E. Resmerita and R. S. Anderssen. Joint additive Kullback–Leibler residual minimization and regularization for linear inverse problems. *Math. Meth. Appl. Sci.*, 30(13):1527–1544, 2007.

[42] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, volume 2, pages 860–867. IEEE, 2005.

[43] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*, volume 167 of *Applied Mathematical Sciences*. Springer, New York, 2009.

[44] J. Schlemper, J. Caballero, J. V. Hajnal, A. Price, and D. Rueckert. A deep cascade of convolutional neural networks for mr image reconstruction. In *Proc. Inf. Process. Med. Imaging*, pages 647–658. Springer, 2017.

[45] J. Schwab, S. Antholzer, and M. Haltmeier. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Probl.*, 35(2):025008, 2019.

[46] J. Schwab, S. Antholzer, and M. Haltmeier. Big in Japan: Regularizing networks for solving inverse problems. *J. Math. Imaging Vis.*, pages 445–455, 2020.

[47] J. Sulam, A. Aberdam, A. Beck, and M. Elad. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[48] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, pages 9446–9454, 2018.

[49] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis. Compressed sensing with deep image prior and learned regularization. *arXiv:1806.06438*, 2018.

[50] C. Wachinger, M. Yigitsoy, E. Rijkhorst, and N. Navab. Manifold learning for image-based breathing gating in ultrasound and MRI. *Med. Image Anal.*, 16(4):806–818, 2012.

[51] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep ADMM-net for compressive sensing MRI. In *Proc. 30th International Conference on Neural Information Processing Systems*, pages 10–18, 2016.