frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 13 June 2023
DOI 10.3389/fams.2023.1176850

OPEN ACCESS

EDITED BY
Housen Li,
University of Göttingen, Germany

REVIEWED BY
Laura Antonelli,
National Research Council (CNR), Italy
Markus Haltmeier,
University of Innsbruck, Austria

*CORRESPONDENCE
Martin Benning
✉ m.benning@qmul.ac.uk

RECEIVED 01 March 2023
ACCEPTED 08 May 2023
PUBLISHED 13 June 2023

CITATION
Wang X and Benning M (2023) A lifted Bregman formulation for the inversion of deep neural networks. *Front. Appl. Math. Stat.* 9:1176850. doi: 10.3389/fams.2023.1176850

A lifted Bregman formulation for the inversion of deep neural networks

Xiaoyu Wang[1] and Martin Benning[2,3]*

[1]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, United Kingdom, [2]Faculty of Science and Engineering, School of Mathematical Sciences, Queen Mary University of London, London, United Kingdom, [3]The Alan Turing Institute, London, United Kingdom

We propose a novel framework for the regularized inversion of deep neural networks. The framework is based on the authors' recent work on training feed-forward neural networks without the differentiation of activation functions. The framework lifts the parameter space into a higher dimensional space by introducing auxiliary variables, and penalizes these variables with tailored Bregman distances. We propose a family of variational regularizations based on these Bregman distances, present theoretical results and support their practical application with numerical examples. In particular, we present the first convergence result (to the best of our knowledge) for the regularized inversion of a single-layer perceptron that only assumes that the solution of the inverse problem is in the range of the regularization operator, and that shows that the regularized inverse provably converges to the true inverse if measurement errors converge to zero.

KEYWORDS

inverse problems, regularization theory, lifted network training, Bregman distance, perceptron, multi-layer perceptron, variational regularization, total variation regularization

1. Introduction

Neural networks are computing systems that have revolutionized a wide range of research domains over the past decade and outperformed many traditional machine learning approaches [cf. [1, 2]]. This performance often comes at the cost of interpretability (or rather a lack thereof) of the outputs that a neural network produces for given inputs. As a consequence, a lot of research has focused on understanding representations of neural networks and on developing strategies to interpret these representations, predominantly with saliency maps [3–6]. An alternative approach focuses on understanding deep image representations by inverting them [7]. The authors propose a total-variation-based variational optimization method that aims to infer the network input from the network output with regularized inversion.

While the concept of inverting neural networks is certainly not new [cf. [8–11]], there has been increasing interest in recent years largely due to developments in nonlinear dimensionality reduction and generative modeling that include (but are not limited to) (variational) Autoencoders [12], Normalizing Flows [13, 14], Cycle-Consistent Generative Adversarial Networks [15], and Probabilistic Diffusion Models [16, 17].

While several approaches for the inversion of neural networks have been proposed especially in the context of generative modeling [see for example [18, 19] in the context of normalizing flows, [20] in the context of generative adversarial networks, and [21] in the context of probabilistic diffusion models] an important aspect, which is often overlooked, is that invertible operations alone are not automatically stable with respect to small variations in the data. For example, computing the solution of the heat equation after a fixed termination time is stable with respect to variations in the initial condition, but estimating the initial condition from the terminal condition of the heat equation is not stable with respect to perturbations in the terminal condition. This issue cannot be resolved without approximation of the inverse with a family of continuous operators, also known as *regularization*. The research field of *Inverse and Ill-posed Problems* and its branch *Regularization Theory* focus strongly on the stable approximation of ill-posed and ill-conditioned inverses via *regularizations* [22] and so-called *variational regularizations* [23, 24] that are a special class of (nonlinear) regularizations. The optimization model proposed in [7] can be considered as a variational regularization method with total variation regularization; however, the work in Mahendran and Vedaldi [7] is purely empirical, and to the best of our knowledge no works exist that rigorously prove that the proposed approach is a variational regularization.

In this work, we propose a novel regularization framework based on lifting with tailored Bregman distances and prove that the proposed framework is a convergent variational regularization for the inverse problem of estimating the inputs from single-layer perceptrons or the inverse problem of estimating hidden variables in a multi-layer perceptron sequentially. While there has been substantial work in previous years that focuses on utilizing neural networks as nonlinear operators in variational regularization methods [25–29], this is the first work that provides theoretical guarantees for the stable, model-based inversion of neural networks to the best of our knowledge.

Our contributions are three-fold. (1) We propose a novel framework for the regularized inversion of multi-layer perceptrons, respectively feed-forward neural networks, that is based on the lifted Bregman framework recently proposed by the authors in [30]. (2) We show that for the single-layer perceptron case, the proposed variational regularization approach is a provably convergent regularization under very mild assumptions. To our knowledge, this is the first time that an inversion method has been proposed that does not just allow to perform inversion empirically, but for which we can prove that the proposed method is a convergent regularization method without overly restrictive assumptions such as differentiability of the activation function and the presence of a tangential cone condition. (3) We propose a proximal first-order optimization strategy to solve the proposed variational regularization method and present several numerical examples that support the effectiveness of the proposed model-based regularization approach.

The paper is structured as follows. In Section 2, we introduce the lifted Bregman formulation for the model-based inversion of feed-forward neural networks. In Section 3, we prove that for the single-layer perceptron case the proposed model is a convergent variational regularization method and provide general error estimates as well as error estimates for a concrete example of a perceptron with ReLU activation function. In Section 4, we discuss how to implement the proposed variational regularization computationally for both the single-layer and multi-layer perceptron setting with a generalization of the primal-dual hybrid gradient method and coordinate descent. Subsequently, we present numerical results that demonstrate empirically that the proposed approach is a model-based regularization in Section 5, before we conclude this work with a brief section on conclusions and outlook in Section 6.

2. Model-based inversion of feed-forward networks

Suppose we are given an L-layer feed-forward neural network $\mathcal{N}:\mathbb{R}^n\times\mathcal{P}\to\mathbb{R}^m$ of the form

$$\mathcal{N}(x,\Theta)=\sigma_L(f(\sigma_{L-1}(f(\dots\sigma_1(f(x,\Theta_1))\dots)),\Theta_L)),\qquad(1)$$

for input data $x\in\mathbb{R}^n$ and pre-trained parameters $\Theta\in\mathcal{P}$. Here, $\{\sigma_l\}_{l=1}^L$ denotes the collection of nonlinear activation functions and f denotes a generic function parameterized by parameters $\{\Theta_l\}_{l=1}^L$. For ease of notation, we use Θ to refer to all parameters $\{\Theta_l\}_{l=1}^L$. For a given network output $y\in\mathbb{R}^m$, our goal is to solve the inverse problem

$$\mathcal{N}(x,\Theta)=y\qquad(2)$$

for the unknown input $x\in\mathbb{R}^n$. The problem (2) is usually ill-posed in the sense that a solution may not exist (especially if $n\ll m$) or is not unique (especially if $m\ll n$, or if information is lost through application of nonlinear activation functions). Moreover, even for a network with identity activation functions σ_l and affine linear transformation f, solving (2) is often ill-conditioned in the sense that errors in y get heavily amplified when solving for x. We therefore, propose to approximate the inverse of this nonlinear, potentially ill-posed inverse problem via the minimization of a lifted Bregman formulation of the form

$$\begin{pmatrix}x^\alpha\\x_1^\alpha\\\vdots\\x_{L-1}^\alpha\end{pmatrix}\in\operatorname*{arg\,min}_{x,x_1,\dots,x_{L-1}}\left\{\sum_{l=1}^L B_{\Psi_l}(x_l,f(x_{l-1},\Theta_l))+\alpha R(x)\right\},\qquad(3)$$

where we assume $x_0=x$ and $x_L=y^\delta$ for simplicity of notation. The data y^δ is a perturbed version of y, for which we assume $B_{\Psi_L}[y^\delta,f(x_{L-1}^\dagger,\Theta_L)]\le\delta^2$, for some constant $\delta\ge0$ and $y=\sigma_L[f(x_{L-1}^\dagger,\Theta_L)]$. Please note that this approach is referred to as "lifted" because the solution space is lifted to a higher dimensional space that also includes auxiliary variables x_l^α for all intermediate layers. The functions B_{Ψ_l} for $l=1,\dots,L$ are defined as

$$B_{\Psi_l}(x,z)=\frac{1}{2}\|x\|^2+\Psi_l(x)+\left(\frac{1}{2}\|\cdot\|^2+\Psi_l\right)^*(z)-\langle x,z\rangle,\qquad(4)$$

for a proper, convex, and lower semi-continuous function $\Psi_l\colon\mathbb{R}^{n_l}\to\mathbb{R}\cup\{\infty\}$. The notation $\left(\frac{1}{2}\|\cdot\|^2+\Psi_l\right)^*$ refers to the convex or Fenchel conjugate of $\frac{1}{2}\|\cdot\|^2+\Psi_l$, i.e., $\left(\frac{1}{2}\|\cdot\|^2+\Psi_l\right)^*(z)=\sup_y\langle z,y\rangle-\frac{1}{2}\|y\|^2-\Psi_l(y)$. Last but not least, the function $R:\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$ is a proper, convex, and

lower semi-continuous function that enables us to incorporate a-priori information into the inversion process. The impact of this is controlled by the parameter $\alpha > 0$.

Please note that the functions B_{Ψ_l} are directly connected to the chosen activation functions $\{\sigma_l\}_{l=1}^L$. Following [30], we observe

$$B_{\Psi_l}(x,z) \geq \frac{1}{2}\|\sigma_l(z) - x\|^2,$$

where $\sigma_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ is the proximal map with respect to Ψ_l, i.e.

$$\sigma_l(z) = \operatorname*{arg\,min}_{y\in\mathbb{R}^{n_l}} \left\{ \frac{1}{2}\|y - z\|^2 + \Psi_l(y) \right\},$$

for all $l \in \{1,\dots,L\}$. This means that we will solely focus on feed-forward neural networks with nonlinear activation functions that are proximal maps.

The advantage of using functions B_{Ψ_l} over more conventional functions such as the squared Euclidean norm of the difference of the network output and the measured output, i.e., $\frac{1}{2}\|\mathcal{N}(x,\Theta) - y\|^2$, is that the functions B_{Ψ_l} are continuously differentiable with respect to their second argument [along with several other useful properties, cf. [30], Theorem 10]. If we define $F_x^l(z) := B_{\Psi_l}(x,z)$, we observe

$$\nabla F_x^l(z) = \sigma_l(z) - x. \tag{5}$$

Please note that the family of objective functions B_{Ψ_l} satisfies several other interesting properties; we refer the interested reader to [30], Theorem 10.

For the remainder of this work, we assume that the parameterized functions f are affine-linear in the first argument, with parameters Θ_l. A concrete example is the affine-linear transformation $f(x,\Theta_l) = W_l x + b_l$, for a (weight) matrix $W_l \in \mathbb{R}^{n_l\times n_{l-1}}$, a (bias) vector $b_l \in \mathbb{R}^{n_l}$ and the collection of parameters $\Theta_l = (W_l, b_l)$.

In the next section, we show that (3) is a variational regularization method for $L = 1$ and prove a convergence rate with which the solution of (3) converges toward the true input of a perceptron when δ converges to zero.

3. Convergence analysis and error estimates

In this section, we show that the proposed model (3) is a convergent variational regularization for the specific choice $L = 1$ and the assumption $f(x,\Theta) = Wx + b$ for $\Theta = (W,b)$, which reduces (3) to a variational regularization model for the perceptron case studied in Wang and Benning [31]. In contrast to Wang and Benning [31] we are not interested in estimating the perceptron parameters W and b but assume that these are fixed, and that we study the regularization operator

$$\begin{aligned} &\mathcal{R}_\alpha :\ \operatorname{dom}(\Psi) \rightrightarrows \mathbb{R}^n, \\ &\mathcal{R}_\alpha : y^\delta \rightrightarrows x_\alpha \in \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \left\{ B_\Psi\left(y^\delta, Wx + b\right) + \alpha R(x) \right\}, \end{aligned} \tag{6}$$

where $\operatorname{dom}(\Psi)$ is defined as $\operatorname{dom}(\Psi) := \{y \in \mathbb{R}^m \mid \Psi(y) < \infty\}$. We first want to establish under which assumptions (6) is well-defined for all y^δ.

3.1. Well-definedness

For simplicity, we focus on the finite-dimensional setting with network inputs in $\mathbb{R}^n$ and outputs in $\operatorname{dom}(\Psi)$. However, the following analysis also extends to more general Banach space settings with additional assumptions on the operator W, see for instance [24], Section 5.1. Following [24], we assume that R is non-negative and the polar of a proper function, i.e., $R = H^*$ for a proper function $H : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$. Note that this automatically implies convexity of R. Moreover, we assume that Ψ is a proper, non-negative and convex function that is continuous on $\operatorname{dom}(\Psi)$, which implies that B_Ψ is proper, non-negative, convex in its second argument and continuous in its first argument for every $y^\delta \in \operatorname{dom}(\Psi)$. Then, for every $g \in \operatorname{dom}(\Psi)$ there exists x with

$$B_\Psi(g, Wx + b) + \alpha R(x) < \infty.$$

Last but not least, we assume that R and Ψ are chosen such that for each $g \in \operatorname{dom}(\Psi)$ and $\alpha > 0$ we have

$$\|x\| \leq c(a,b,\|g\|), \qquad \text{if} \quad B_\Psi(g, Wx + b) \leq a \quad \text{and} \quad \alpha R(x) \leq d,$$

for constants a, d, and a constant c that depends monotonically non-decreasing on all arguments. With these assumptions, we can then verify the following lemma.

Theorem 1. Let the assumptions outlined in the previous paragraph be satisfied.

1. Then, for every $y \in \left\{ g \in \operatorname{dom}(\Psi) \,\middle|\, \operatorname{arg\,min}_{x\in\mathbb{R}^n, R(x)<\infty} B_\Psi(g, Wx + b) \neq \emptyset \right\}$ the selection operator

$$\mathcal{S}(y) = \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \left\{ R(x) \,\middle|\, x \in \operatorname*{arg\,min}_{\tilde{x}\in\mathbb{R}^n} B_\Psi(y, W\tilde{x} + b) \right\}$$

   is well-defined.
2. The regularization operator $\mathcal{R}_\alpha$ as defined in (3) is well-defined in the sense that for every $y \in \operatorname{dom}(\Psi)$ there exists $x_\alpha \in \mathbb{R}^n$ with $x_\alpha \in \mathcal{R}_\alpha(y)$. Moreover, the set $\mathcal{R}_\alpha(y)$ is a convex set.
3. For every sequence $y_n \to y \in \operatorname{dom}(\Psi)$ there exists a subsequence $x_{n_k} \in \mathcal{R}_\alpha(y_{n_k})$ converging to an element $x^* \in \mathcal{R}_\alpha(y)$.

Proof. The results follow directly from [24], Lemma 5.5, Theorem 5.6, and Theorem 5.7. The latter statement originally only implies convergence in the weak-star topology; however, since we are in a finite-dimensional Hilbert space, this automatically implies strong convergence here.

3.2. Error estimates

Having established that (3) is a regularization operator, we now want to prove that it is also a convergent regularization operator in the sense of the estimate

$$D_R(x^\dagger, x^\alpha) \leq C\delta, \tag{7}$$

such that

$$\limsup_{\delta\to 0}\left\{D_R(x^\dagger,x^\alpha)\;\middle|\;x^\alpha\in\mathcal{R}_\alpha(y^\delta),\,y^\delta\in\mathrm{dom}(\Psi),\;B_\Psi(y^\delta,Wx^\dagger+b)\le\delta^2\right\}=0\,.$$

Here, the term D_R denotes the (generalized) Bregman distance [or divergence, [cf. [32, 33]] with respect to R, i.e.,

$$D_R(x,\tilde{x})=R(x)-R(\tilde{x})-\langle\tilde{q},x-\tilde{x}\rangle\,,$$

for two arguments $x,\tilde{x}\in\mathrm{dom}(R)$ and a subgradient $\tilde{q}\in\partial R(\tilde{x})=\{q\in\mathbb{R}^n\,|\,R(x)\ge R(\tilde{x})+\langle q,x-\tilde{x}\rangle,\forall x\in\mathrm{dom}(R)\}$. The vector x^α is a solution of (3) with data y^δ for which we assume $B_\Psi(y^\delta,Wx^\dagger+b)\le\delta^2$, and $C\ge 0$ is a constant. The vector $x^\dagger$ is an element of the selection operator as specified in Lemma 1. 1, i.e., $x^\dagger\in\mathcal{S}(y)$ for $y\in\mathrm{dom}(\Psi)$. Note that $x^\dagger\in\mathcal{S}(y)$ is equivalent to $x^\dagger$ being a R-minimizing vector amongst all vectors that satisfy $0=W^*\left(\sigma(Wx^\dagger+b)-y\right)$, where σ denotes the proximal map with respect to Ψ. This is due to the fact that $x^\dagger\in\arg\min_{\tilde{x}\in\mathbb{R}^n}B_\Psi(y,W\tilde{x}+b)$ is equivalent to $0=\nabla B_\Psi(y,Wx^\dagger+b)=W^*\left(\sigma(Wx^\dagger+b)-y\right)$. Assuming that $\sigma(Wx^\dagger+b)-y$ does not lie in the nullspace of W^*, this further implies $y=\sigma(Wx^\dagger+b)$.

In order to be able to derive error estimates of the form (7), we restrict ourselves to solutions $x^\dagger$ that are in the range of $\mathcal{R}_\alpha$. This means that there exists $y^\dagger$ such that $x^\dagger\in\mathcal{R}_\alpha(y^\dagger)$. Considering the optimality condition of (3) for $y^\dagger$, this implies

$$W^*\left(\frac{y^\dagger-\sigma(Wx^\dagger+b)}{\alpha}\right)\in\partial R(x^\dagger)\,,$$

which for $v^\dagger:=(y^\dagger-\sigma(Wx^\dagger+b))/\alpha=(y^\dagger-y)/\alpha$ is equivalent to the existence of a source condition element $v^\dagger$ that satisfies the source condition [cf. [22, 24]].

$$W^*v^\dagger\in\partial R(x^\dagger)\,,\tag{SC}$$

In the following, we verify that the symmetric Bregman distance with respect to R between a solution of the regularization operator and the solution of the inverse problem is converging to zero if the error in the data is converging to zero. The symmetric Bregman distance or Jeffreys distance between two vectors x and $\tilde{x}$ simply is the sum of two Bregman distances with interchanged arguments, i.e.,

$$D_R^{\mathrm{symm}}(x,\tilde{x}):=D_R(x,\tilde{x})+D_R(\tilde{x},x)=\langle x-\tilde{x},q-\tilde{q}\rangle\,,$$

for $q\in\partial R(x)$ and $\tilde{q}\in\partial R(\tilde{x})$; hence, an error estimate in the symmetric Bregman distance also implies an error estimate in the classical Bregman distance.

Before we begin our analysis, we recall the concept of the *Jensen-Shannon divergence* [34], which for general proper, convex and lower semi-continuous functions $F:\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$ generalizes to so-called *Burbea-Rao divergences* [35–37] and are defined as follows.

Definition 1 (Burbea-Rao divergence). Suppose $F:\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$ is a proper, convex and lower semi-continuous function. The corresponding Burbea-Rao divergence is defined as

$$J_F(x,\tilde{x}):=\frac{1}{2}\left(F(x)+F(\tilde{x})-2F\left(\frac{x+\tilde{x}}{2}\right)\right)\,,\tag{8}$$

for all $x,\tilde{x}\in\mathrm{dom}(F)$.

Another important concept that we need in order to establish error estimates is that of Fenchel conjugates [cf. [38]].

Definition 2 (Fenchel conjugate). The Fenchel (or convex) conjugate $F^*:\mathbb{R}^n\to\mathbb{R}\cup\{-\infty,\infty\}$ of a function $F:\mathbb{R}^n\to\mathbb{R}\cup\{-\infty,+\infty\}$ is defined as

$$F^*(w):=\sup_{x\in\mathbb{R}^n}\langle x,w\rangle-F(x)\,.$$

The Fenchel conjugate that is of particular interest to us is the conjugate of the function $B_\Psi(y,z)$ with respect to the second argument, which we characterize with the following lemma.

Lemma 1. The Fenchel conjugate of $F_y(z):=B_\Psi(y,z)$ with respect to the second argument z reads

$$F_y^*(w)=\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(y+w)-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(y)\,.$$

Proof. From the definition of the Fenchel conjugate we observe

$$\begin{aligned}F_y^*(w)&=\sup_{z\in\mathbb{R}^m}\langle z,w\rangle-F_y(z)\\&=\sup_{z\in\mathbb{R}^m}\langle z,w\rangle-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(y)-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)^*(z)\\&\quad+\langle y,z\rangle\\&=-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(y)+\sup_{z\in\mathbb{R}^m}\langle z,w+y\rangle\\&\quad-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)^*(z)\\&=-\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(y)+\left(\frac{1}{2}\|\cdot\|^2+\Psi\right)(w+y)\,,\end{aligned}$$

which concludes the proof.

Having defined the Burbea-Rao divergence and having established the Fenchel conjugate of $B_\Psi(y,z)$ with respect to the second argument z for fixed y, we can now present and verify our main result that is motivated by [39].

Theorem 2. Suppose R and Ψ satisfy the assumptions outlined in Section 3.1. Then, for data y^δ and $x^\dagger$ that satisfy $B_\Psi(y^\delta,Wx^\dagger+b)\le\delta^2$ with $\delta\ge 0$, a solution $x^\alpha\in\mathcal{R}_\alpha(y^\delta)$ of the variational regularization problem (3), and a solution $x^\dagger$ of the perceptron problem $y=\sigma(Wx^\dagger+b)$ that satisfies $x^\dagger\in\mathcal{S}(y)$ and (SC), we observe the error estimate

$$\begin{aligned}&(1-c)B_\Psi(y^\delta,Wx_\alpha+b)+\alpha D_R^{\mathrm{symm}}(x_\alpha,x^\dagger)\\&\le(1+c)\delta^2+\frac{\alpha^2}{c}\|v^\dagger\|^2,+2cJ_\Psi\left(y^\delta+\frac{\alpha}{c}v^\dagger,y^\delta-\frac{\alpha}{c}v^\dagger\right)\end{aligned}\tag{9}$$

for a constant $c\in(0,1]$.

Proof. Every solution x^α that satisfies $x_\alpha\in\mathcal{R}_\alpha(y^\delta)$ can equivalently be characterized by the optimality condition

$$W^*\left(\sigma(Wx_\alpha+b)-y^\delta\right)+\alpha p_\alpha=0\,,$$

for any subgradient $p_\alpha \in \partial R(x_\alpha)$. Subtracting $p^\dagger \in \partial R(x^\dagger)$ from both sides of the equation and taking a dual product with $x_\alpha - x^\dagger$ then yields

$$\langle \sigma(Wx_\alpha + b) - y^\delta, Wx_\alpha - Wx^\dagger\rangle + \alpha D_R^{\text{symm}}(x_\alpha, x^\dagger) = -\alpha\langle p^\dagger, x_\alpha - x^\dagger\rangle . \tag{10}$$

We easily verify

$$D_{B_\Psi(y^\delta, W\cdot+b)}(x^\dagger, x_\alpha) = B_\Psi(y^\delta, Wx^\dagger + b) - B_\Psi(y^\delta, Wx_\alpha + b) - \langle \sigma(Wx_\alpha + b) - y^\delta, Wx^\dagger - Wx_\alpha\rangle ;$$

hence, we can replace $\langle \sigma(Wx^\alpha + b) - y^\delta, Wx_\alpha - Wx^\dagger\rangle$ with $D_{B_\Psi(y^\delta, W\cdot+b)}(x^\dagger, x_\alpha) + B_\Psi(y^\delta, Wx_\alpha + b) - B_\Psi(y^\delta, Wx^\dagger + b)$ in (10) to obtain

$$D_{B_\Psi(y^\delta, W\cdot+b)}(x^\dagger, x_\alpha) + B_\Psi(y^\delta, Wx_\alpha + b) + \alpha D_R^{\text{symm}}(x_\alpha, x^\dagger) = B_\Psi(y^\delta, Wx^\dagger + b) - \alpha\langle p^\dagger, x_\alpha - x^\dagger\rangle .$$

We know $0 \leq D_{B_\Psi(y^\delta, W\cdot+b)}(x^\dagger, x_\alpha)$ due to the convexity of $B_\Psi(y^\delta, W\cdot + b)$, and we also know that (SC) enables us to choose $p^\dagger = W^* v^\dagger$. Hence, we can estimate

$$B_\Psi(y^\delta, Wx_\alpha + b) + \alpha D_R^{\text{symm}}(x_\alpha, x^\dagger) \leq B_\Psi(y^\delta, Wx^\dagger + b) - \alpha\langle v^\dagger, Wx_\alpha - Wx^\dagger\rangle .$$

Next, we introduce the constant $c \in (0, 1]$ to split the loss functions $B_\Psi(y^\delta, Wx_\alpha + b)$ and $B_\Psi(y^\delta, Wx^\dagger + b)$ into $(1 - c)B_\Psi(y^\delta, Wx_\alpha + b) + cB_\Psi(y^\delta, Wx_\alpha + b)$ and $(1 + c)B_\Psi(y^\delta, Wx^\dagger + b) - cB_\Psi(y^\delta, Wx^\dagger + b)$, respectively. This means we estimate

$$\begin{aligned}(1 - c)B_\Psi(y^\delta, Wx_\alpha + b) + \alpha D_R^{\text{symm}}(x_\alpha, x^\dagger) &\leq (1 + c)B_\Psi(y^\delta, Wx^\dagger + b)\\ &+ \langle \alpha v^\dagger, Wx^\dagger + b\rangle - cB_\Psi(y^\delta, Wx^\dagger + b)\\ &- \langle \alpha v^\dagger, Wx_\alpha + b\rangle - cB_\Psi(y^\delta, Wx_\alpha + b) .\end{aligned}$$

Next, we make use of Lemma 1 to estimate

$$\langle \alpha v^\dagger, Wx^\dagger + b\rangle - cB_\Psi(y^\delta, Wx^\dagger + b) \leq c\left(\left(\frac{1}{2}\|\cdot\|^2 + \Psi\right)\left(y^\delta + \frac{\alpha}{c}v^\dagger\right) - \left(\frac{1}{2}\|\cdot\|^2 + \Psi\right)(y^\delta)\right),$$

and

$$-\langle \alpha v^\dagger, Wx_\alpha + b\rangle - cB_\Psi(y^\delta, Wx_\alpha + b) \leq c\left(\left(\frac{1}{2}\|\cdot\|^2 + \Psi\right)\left(y^\delta - \frac{\alpha}{c}v^\dagger\right) - \left(\frac{1}{2}\|\cdot\|^2 + \Psi\right)(y^\delta)\right).$$

Adding both estimates together yields

$$\begin{aligned}&\langle \alpha v^\dagger, Wx^\dagger + b\rangle - cB_\Psi(y^\delta, Wx^\dagger + b) - \langle \alpha v^\dagger, Wx_\alpha + b\rangle - cB_\Psi(y^\delta, Wx_\alpha + b),\\ &\leq \frac{\alpha^2}{c}\|v^\dagger\|^2 + c\left(\Psi\left(y^\delta + \frac{\alpha}{c}v^\dagger\right) + \Psi\left(y^\delta - \frac{\alpha}{c}v^\dagger\right) - 2\Psi(y^\delta)\right),\\ &= \frac{\alpha^2}{c}\|v^\dagger\|^2 + 2cJ_\Psi\left(y^\delta + \frac{\alpha}{c}v^\dagger, y^\delta - \frac{\alpha}{c}v^\dagger\right),\end{aligned}$$

which together with the error bound $B_\Psi(y^\delta, Wx^\dagger + b) \leq \delta^2$ concludes the proof.

Remark 1. We want to emphasize that for continuous Ψ and $c > 0$ we automatically observe

$$\lim_{\alpha\to 0} J_\Psi\left(y^\delta + \frac{\alpha}{c}v^\dagger, y^\delta - \frac{\alpha}{c}v^\dagger\right) = 0 ,$$

in which case the important question from an error estimate point-of-view is if the term converges quicker to zero than α, as we would need to guarantee $\lim_{\alpha\to 0} J_\Psi\left(y^\delta + \frac{\alpha}{c}v^\dagger, y^\delta - \frac{\alpha}{c}v^\dagger\right)/\alpha = 0$ in order to guarantee that the symmetric Bregman distance in (9) converges to zero for $\alpha \to 0$.

Example 1 (ReLU perceptron). Let us consider a concrete example to demonstrate that (6) is a convergent regularization with respect to the symmetric Bregman distance of R. We know that for $\sigma(z) = \text{prox}_\Psi(z) = \max(0, z)$ to hold true we have to choose $\Psi(z) = \begin{cases} 0 & z \in [0,\infty)^m \\ \infty & \text{else}\end{cases}$. This means that for $B_\Psi(y^\delta, z)$ to be well-defined for any z we require $y_i^\delta \geq 0$ for all $i \in \{1, \ldots, m\}$. In order for the Burbea-Rao divergence to be well-defined, we further require

$$-\frac{c}{\alpha}y_i^\delta \leq v_i^\dagger \leq \frac{c}{\alpha}y_i^\delta ,$$

for all $i \in \{1, \ldots, m\}$, or $\|v^\dagger\|_\infty \leq (c\|y^\delta\|_\infty/\alpha)$ in more compact notation. If $\|v^\dagger\|_\infty \leq (c\|y^\delta\|_\infty/\alpha)$ is guaranteed, we observe $J_\Psi\left(y^\delta + \frac{\alpha}{c}v^\dagger, y^\delta - \frac{\alpha}{c}v^\dagger\right) = 0$. Hence, we can simplify the estimate (9) to

$$\frac{1 - c}{\alpha}B_\Psi(y^\delta, Wx_\alpha + b) + D_R^{\text{symm}}(x_\alpha, x^\dagger) \leq \frac{1 + c}{\alpha}\delta^2 + \frac{\alpha}{c}\|v^\dagger\|^2 ,$$

where we have also divided by α on both sides of the inequality. If we choose $\alpha(\delta) = \sqrt{c(1 + c)}\delta/\|v^\dagger\|$, we obtain the estimate

$$\frac{(1 - c)\|v^\dagger\|}{\delta\sqrt{c(1 + c)}}B_\Psi(y^\delta, Wx_{\alpha(\delta)} + b) + D_R^{\text{symm}}(x_{\alpha(\delta)}, x^\dagger) \leq 2\sqrt{\frac{1 + c}{c}}\|v^\dagger\|\delta ,$$

as long as we can ensure

$$\left\|\frac{v^\dagger}{\|v^\dagger\|}\right\|_\infty \leq \sqrt{\frac{c}{1 + c}}\left\|\frac{y^\delta}{\delta}\right\|_\infty .$$

Together with $D_R^{\text{symm}}(x_{\alpha(\delta)}, x^\dagger) \geq D_R(x^\dagger, x_{\alpha(\delta)})$ we have established an estimate of the form (7), with constant $C = 2\sqrt{\frac{1+c}{c}}\|v^\dagger\|$. Hence, we have verified that the variational regularization method (3) is not only a regularization method but even a convergent regularization method in this specific example.

We want to briefly comment on the extension of the convergence analysis to the general case $L > 1$ with the following remark.

Remark 2. The presented convergence analysis easily extends to a sequential, layer-wise inversion approach. Suppose we have L layers and begin with the final layer, then we can formulate the variational problem

$$x_{L-1}^\alpha \in \arg\min_{x_{L-1}}\left\{B_{\Psi_L}(y^\delta, W_L x_{L-1} + b_L) + \alpha_{L-1}\Psi_{L-1}(x_{L-1})\right\} ,$$

which is also of the form of (6), but where R has been replaced with Ψ_{L-1}. Alternatively, one can also replace Ψ_{L-1} with another function R_{L-1} if good prior knowledge for the auxiliary variable x_{L-1} exists. Once we have estimated x^α_{L-1}, we can recursively estimate

$$x^\alpha_{l-1} \in \arg\min_{x_{l-1}} \left\{ B_{\Psi_l}(x^\alpha_l, W_l x_{l-1} + b_l) + \alpha_{l-1}\Psi_{l-1}(x_{l-1}) \right\},$$

for $l = L-1, \ldots, 2$ and subsequently compute x^α as a solution of (3) but with data x^α_1 instead of y^δ.

The advantage of such a sequential approach is that every individual regularization problem is convex and the previously presented theorems and guarantees still apply. The disadvantage is that for this approach to work in theory, we require bounds for every auxiliary variable of the form $B_{\Psi_l}(x^\alpha_l, W_l x^\alpha_{l-1} + b_l) \leq \delta_l^2$, which is a rather unrealistic assumption. Moreover, it is also not realistic to assume that good prior knowledge for the auxiliary variables exist.

Please note that showing that the simultaneous approach (3) is a (convergent) variational regularization is beyond the scope of this work as it is harder and potentially requires additional assumptions for the following reason. The overall objective function in (3) is no longer guaranteed to be convex with respect to all variables simultaneously, which means that we cannot simply carry over the analysis of the single-layer to the multi-layer perceptron case.

Remark 3 (Infinite-dimensional setting). Please note that almost all theoretical results presented in this section also apply to neural networks that map functions between Banach spaces instead of finite-dimensional vectors. The only result that changes is Theorem 1, Item 3, where the statement in an infinite-dimensional setting only implies convergence in the weak-star topology.

This concludes the theoretical analysis of the perceptron inversion model. In the following section, we focus on how to implement (6) and its more general counterpart (3).

4. Implementation

In this section, we describe how to computationally implement the proposed variational regularization for both the single-layer and the multi-layer perceptron setting. More specifically, we show that the proposed variational regularization can be efficiently solved via a generalized primal-dual hybrid gradient method and a coordinate descent approach.

4.1. Inverting perceptrons

To begin with, we first consider the example of inverting a (single-layer) perceptron. For $L = 1$, Problem (3) reduces to (6), which for a composite regularization function $R \circ K$ reads

$$x^\alpha \in \arg\min_x \left\{ B_\Psi\left(y^\delta, f(x,\Theta)\right) + \alpha R(Kx) \right\}. \tag{11}$$

Here K is a matrix and $\alpha R(Kx)$ denotes the regularization function acting on the argument x. The above Problem (11) can be reformulated to the saddle-point problem

$$\min_x \max_z B_\Psi\left(y^\delta, f(x,\Theta)\right) + \langle z, Kx\rangle - \alpha R^*(z), \tag{12}$$

where R^* denotes the convex conjugate of R. Computationally, we can then solve the saddle-point problem with a generalized version [40] of the popular primal-dual hybrid gradient (PDHG) method [41–45]:

$$x^{k+1} = x^k - \tau_x \left(\left(\text{prox}_\Psi\left(f(x^k,\Theta)\right) - y^\delta \right) \mathcal{J}_f^x(x^k,\Theta) + \alpha K^\top z^k \right) \tag{13a}$$

$$z^{k+1} = \text{prox}_{\tau_z R^*}\left(z^k + \tau_z \alpha K \left(2x^{k+1} - x^k \right) \right). \tag{13b}$$

where we alternate between a descent step in the x variable and an ascent step in the dual variable z. Since (11) is a convex minimization problem, (13) is guaranteed to converge globally for arbitrary starting point, given that τ_x and τ_z are chosen such that $\tau_x\tau_z < 1/\|K\|^2$ and such that (13a) is contractive [check [40], Theorem 5.1 for details].

In this work, we will focus on the discrete total variation $\|\nabla x\|_{p,1}$, [46, 47], as our regularization function $R(Kx)$, but other choices are certainly possible. If we consider a two-dimensional scalar-valued image $x \in \mathbb{R}^{H\times W}$, we can define a finite forward difference discretization of the gradient operator $\nabla : \mathbb{R}^{H\times W} \to \mathbb{R}^{H\times W\times 2}$ as

$$(\nabla x)_{i,j,1} = \begin{cases} x_{i+1,j} - x_{i,j} & \text{if } 1 \leq i < H, \\ 0 & \text{else,} \end{cases},$$

$$(\nabla x)_{i,j,2} = \begin{cases} x_{i,j+1} - x_{i,j} & \text{if } 1 \leq j < W, \\ 0 & \text{else.} \end{cases}.$$

The discrete total variation is defined as the ℓ_1 norm of the p-norm of the pixel-wise image gradients, i.e.

$$\|\nabla x\|_{p,1} = \sum_{i=1}^{H}\sum_{j=1}^{W} |\,(\nabla x)_{i,j}\,|_p = \sum_{i=1}^{H}\sum_{j=1}^{W} \left((\nabla x)^p_{i,j,1} + (\nabla x)^p_{i,j,2} \right)^{1/p}.$$

For our numerical results we consider the isotropic total variation and consequently choose $p = 2$. Hence for a perceptron with affine-linear transformation $f(x,\Theta) = Wx + b$, and with $\sigma = \text{prox}_\Psi$ denoting the activation function, the PDHG approach (13) of solving the perceptron inversion problem (6) can be summarized as

$$x^{k+1} = x^k - \tau_x \left(W^\top \left(\sigma(Wx^k + b) - y \right) - \alpha \text{div} z^k \right), \tag{14a}$$

$$z^{k+1} = \text{prox}_{\tau_z \|\cdot\|^*_{2,1}}\left(z^k + \tau_z \left(2\alpha \nabla x^{k+1} - \alpha \nabla x^k \right) \right). \tag{14b}$$

Please note that we define the discrete approximation of the divergence div such that it satisfies $\text{div} = -\nabla^\top$ in order to be the negative transpose of the discretized finite difference approximation of the gradient in analogy to the continuous case, which is why the sign in (14a) is flipped in comparison to (13a). The proximal map with regards to the convex conjugate of $\|\cdot\|^*_{2,1}$ is simply the argument itself if the maximum of the Euclidean vector-norm per pixel is bounded by one or the projection onto this unit ball.

4.2. Inverting multi-layer perceptrons

We now discuss the implementation of the inversion of multi-layer perceptrons with L layers as described in (3). Note that in this case in order to minimize for x, we also need to optimize with respect to the auxiliary variables $x_1, \ldots, x_{L-1}$.

For the minimization of (3) we consider an alternating minimization approach, also known as *coordinate descent* [48–50]. In this approach we minimize the objective with respect to one variable at a time. In particular, we focus on a semi-explicit coordinate descent algorithm, where we linearize with respect to the smooth functions of the overall objective function. This breaks down the overall minimization problem into L sub-problems, where for x_0 and each x_l variable for $l \in \{1, \ldots, L-1\}$, we have individual minimization problems of the following form:

$$x_0^{t+1} = \arg\min_{x_0} \left\{ \left(\frac{1}{2}\|\cdot\|^2 + \Psi_1 \right)^* (f(x_0, \Theta_1)) - \left\langle x_1^t, f(x_0, \Theta_1) \right\rangle + \alpha R(Kx_0) \right\}, \quad (15a)$$

$$x_l^{t+1} = \arg\min_{x_l} \left\{ \left(\frac{1}{2}\|\cdot\|^2 + \Psi_l \right)(x_l) - \left\langle x_l, f(x_{l-1}^{t+1}, \Theta_l) \right\rangle + \frac{1}{2\tau_{x_l}}\|x_l - x_l^t\|^2 + \left\langle x_l, \left(\text{prox}_{\Psi_{l+1}} \left(f(x_l^t, \Theta_{1+1}) \right) - x_{l+1}^t \right) \mathcal{J}_f^x(x_l^t, \Theta_{1+1}) \right\rangle \right\}. \quad (15b)$$

Note that one advantage for adopting this approach is that we exploit that the overall objective function is convex in each individual variable when all other variables are kept fixed. In the following, we will discuss different strategies to computationally solve each sub-problem.

When optimizing with respect to the input variable x_0, the structure of sub-problem (15a) is identical to the perceptron inversion problem that we have discussed in Section 4.1. Hence, we can approximate x_0^{t+1} with (12), but now with respect to x_1^t instead of y^δ, which yields the iteration

$$x_0^{k+1} = x_0^k - \tau_{x_0} \left(\left(\text{prox}_{\Psi} \left(f(x_0^k, \Theta_1) \right) - x_1^t \right) \mathcal{J}_f^x(x_0^k, \Theta_1) + \alpha K^\top z^k \right), \quad (16a)$$

$$z^{k+1} = \text{prox}_{\tau_z R^*} \left(z^k + \tau_z \alpha K \left(2x_0^{k+1} - x_0^k \right) \right). \quad (16b)$$

For each auxiliary variable x_l with $l \in \{1, \ldots, L-1\}$, the sub-problem associated with (15b) amounts to solving a proximal gradient step with suitable step-size τ_{x_l}, which we can rewrite to

$$x_l^{t+1} = \text{prox}_{\frac{\tau_{x_l}}{1+\tau_{x_l}}\Psi_l} \left(\frac{1}{1+\tau_{x_l}} \left(x_l^t - \tau_{x_l} \left(\left(\text{prox}_{\Psi_l} \left(f(x_l^t, \Theta_{l+1}) \right) - x_{l+1}^t \right) \mathcal{J}_f^x(x_l^t, \Theta_{l+1}) - f(x_{l-1}^{t+1}, \Theta_l) \right) \right) \right). \quad (17)$$

This concludes the discussion on the implementation of the regularized single-layer and multi-layer perceptron inversion. In the next section, we present some numerical results to demonstrate the effectiveness of the proposed approaches empirically.

5. Numerical results

In this section, we present numerical results for the perceptron inversion problem implemented with the PDHG algorithm as outlined in (14), and for the multi-layer perceptron inversion problem implemented with the coordinate descent approach as described in (16) and (17). In the following, we first demonstrate that we can invert a perceptron with random weights and bias terms and ReLU activation function via (11) with total variation regularization and the algorithm described in (14). We then proceed to a more realistic example of inverting the code of a simple autoencoder with perceptron encoder, before we extend the results to the total variation-based inversion of encodings from multi-layer convolutional autoencoders. All results have been computed using PyTorch 3.7 on an Intel Xeon CPU E5-2630 v4.

5.1. The perceptron

We present results for two experiments: the first one is the perceptron inversion of the image of a circle from the noisy output of the perceptron, where we compare the Landweber regularization and the total-variation-based variational regularization (6). For the second experiment, we perform perceptron inversion for samples from the MNIST dataset [51], where we compare the performance of the proposed inversion strategy with the performance of linear and nonlinear decoders on the collection of (approximately) piecewise constant images of hand-written digits.

5.1.1. Circle

We begin with the toy example of recovering the image of a circle from noisy measurements of a ReLU perceptron. To prepare the experiment, we generate a circle image $x^\dagger \in \mathbb{R}^{64\times 64}$, as shown in Figure 1. We construct a perceptron with ReLU activation function using random weights and biases where $W \in \mathbb{R}^{512\times 4,096}$, $b \in \mathbb{R}^{512\times 1}$. The weights operates on the column-vector representation of x, where $x \in \mathbb{R}^{4,096\times 1}$. The noise-free data is generated via the forward operation of the model, i.e., $y = \sigma(Wx^\dagger + b)$. We generate noisy data y^δ by adding Gaussian noise with mean 0 and standard deviation 0.005. Note that we clip all the negative values of y^δ to ensure $y^\delta \in \text{dom}(\Psi)$.

A first attempt to solve this ill-posed perceptron inversion problem is via Landweber regularization [52]. In Figure 2, we see the reconstructed image obtained with Landweber regularization in combination with early stopping following Morozov's discrepancy principle [22, 53]. Even though the Landweber regularized reconstruction matches the data up to the discrepancy value $\|\sigma(Wx^K + b) - y^\delta\|$, the recovered image does not resemble the image $x^\dagger$. We will discuss shortly the reason for this visually poor inversion. In comparison, we see a regularized inversion via the total variation regularization approach following (14) in Figure 3. The regularization parameter for this reconstruction is chosen as $\alpha = 1.5 \times 10^{-2}$. Both x_0 and z are initialized with zero vectors. The stepsize-parameters are chosen as $\tau_x = 1.99/\|W\|_2^2$ and $\tau_z = 1/(8\alpha)$, see [54]. We stop the iterations when changes in x_0 and z in norm are less than a threshold of 10^{-5} or when we reach the

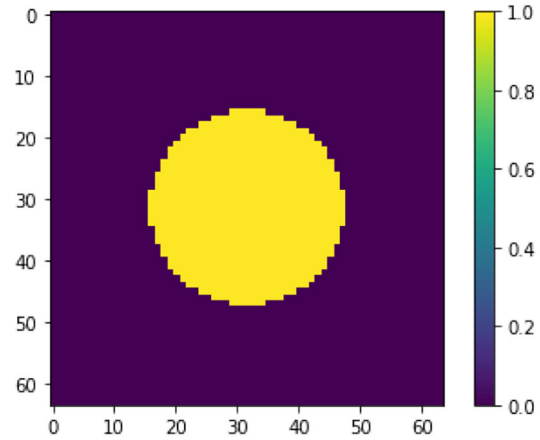

FIGURE 1
Groundtruth image $x^\dagger$ of a circle.

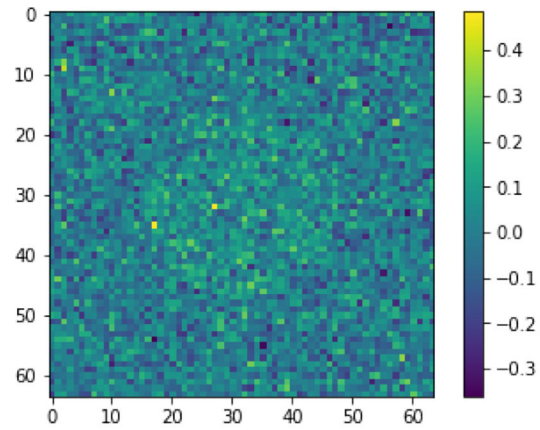

FIGURE 2
Inverted image via Landweber regularization.

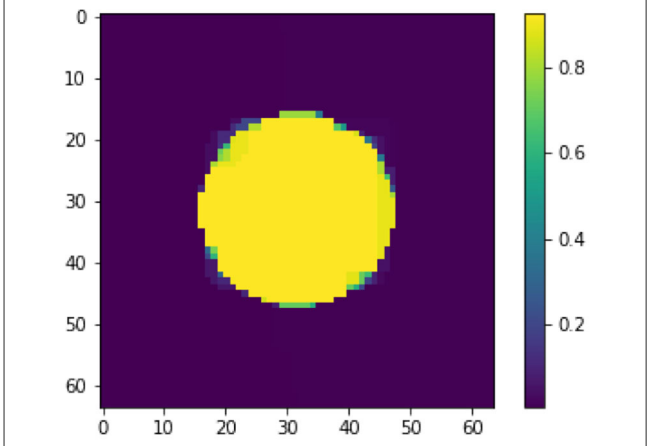

FIGURE 3
Inverted image via TV regularization.

maximum number of iterations, which we set to 10, 000. As shown in Figure 3, the TV-regularization approach is capable of finding a (visually) more meaningful solution.

To explain why the Landweber iteration performs worse compared to the total variation regularization for this specific example, we compare the ℓ_2 norms of each two solutions and the groundtruth image $x^\dagger$. The ℓ_2 norm of the Landweber solution in Figure 2 measures 6.58 while the TV-regularized solution as in Figure 3 and the groundtruth image $x^\dagger$ measure 25.69 and 28.07, respectively. This is not surprising, as the Landweber iteration is known to converge to a minimal Euclidean norm solution if the noise level converges to zero. On the other hand, when we compare the TV semi-norm of each solution, the groundtruth image in measures 128.0, while the Landweber solution in Figure 2 and TV-regularized solution in Figure 3 measure 707.02 and 114.93, respectively, suggesting that the TV-semi-norm is a more suitable regularization function for the inversion of cartoon-like images such as $x^\dagger$.

5.1.2. MNIST

In this second example, we perform perceptron inversion on the MNIST dataset [51]. In particular, we consider the following experimental setup. We first train an autoencoder $\mathcal{A}(x) = \mathcal{D}[\mathcal{E}(x, \Theta_\mathcal{E}), \Theta_\mathcal{D}]$, where $\mathcal{D}(\cdot, \Theta_\mathcal{D})$ and $\mathcal{E}(\cdot, \Theta_\mathcal{E})$ denotes the decoder and the encoder, parameterized by parameters $\Theta_\mathcal{D}$ and $\Theta_\mathcal{E}$, respectively. We pre-train the autoencoder $\mathcal{A}$, compute the code $\mathcal{E}(x, \Theta_\mathcal{E})$ and assign it to the noise-free data variable y, and solve the inverse problem for the input x from the perturbed code y^δ

$$\mathcal{E}(x, \Theta_\mathcal{E}) = y^\delta \ .$$

To be more precise, we first train a two-layer fully connected autoencoder $\tilde{x} = W_2[\sigma(W_1 x + b_1)] + b_2$ using the vanilla stochastic gradient method (SGM) by minimizing the mean squared error (MSE) on the MNIST training dataset. We set the code dimension to 100 and use ReLU as the activation function. Hence $\Theta_\mathcal{E} = (W_1, b_1)$ where $W_1 \in \mathbb{R}^{784\times 100}$ and $b_1 \in \mathbb{R}^{100\times 1}$.

We then invert the code $y^\delta = \sigma(W_1 x + b_1)$ via (11) with Equation (14). All MNIST images are centered as a means of pre-processing. The stepsize-parameters are chosen at $\tau_x = 1.99/\|W_1\|_2^2$ and $\tau_z = 1/(8\alpha)$. We choose the regularization parameter α in the range $[10^{-4}, 10^{-2}]$ and set to 5×10^{-3} for all sample images from the training set, and to $\alpha = 5 \times 10^{-2}$ for all sample images from the validation set. These choices work well with regards to the visual quality of the inverted images.

In Figure 4, we show visualizations of five sample images from the training set, and from the validation set respectively. In comparison, we have also visualized the decoder output. As can be seen, using the code that contains the same compressed information, the inverted images show more clearly defined edges and better visual quality than the decoded outputs. This is to be expected as we compare a nonlinear regularized inversion method with a linear decoder.

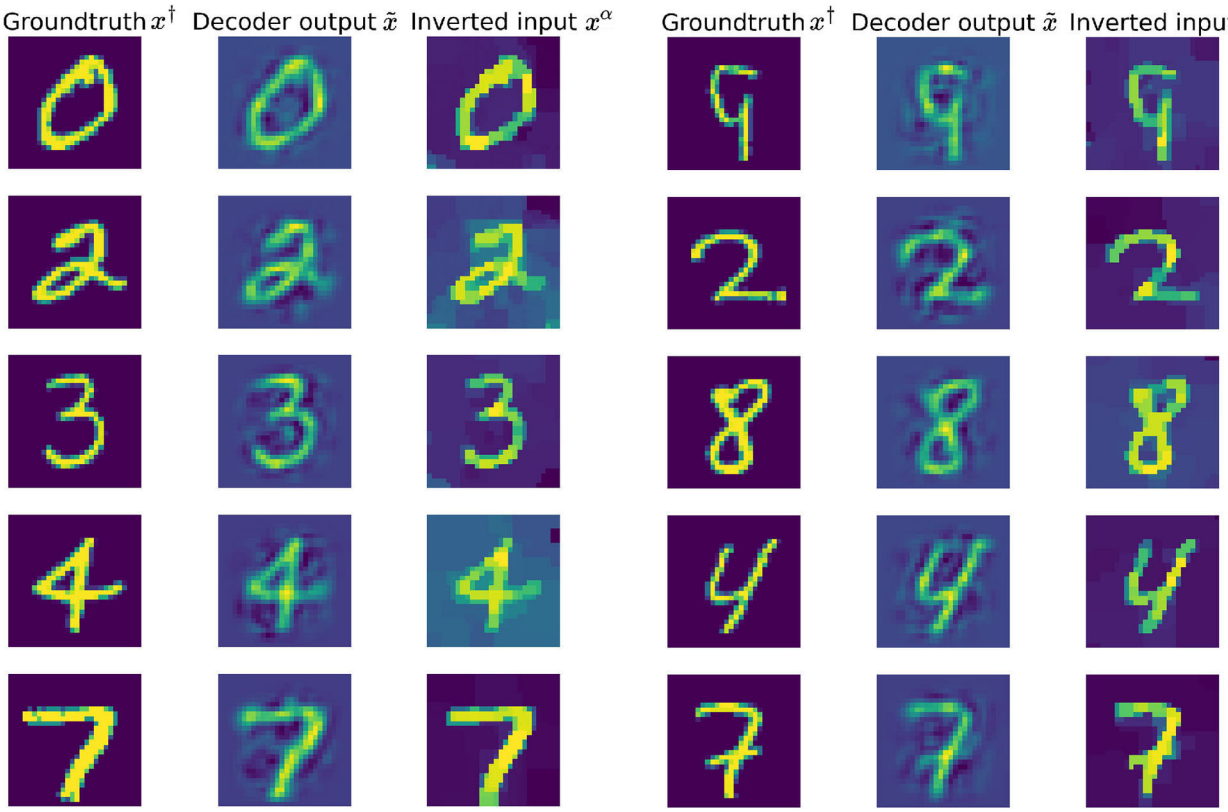

FIGURE 4
Groundtruth input images from the MNIST training dataset **(Left)** and validation dataset **(Right)**, together with the corresponding autoencoder output images $\tilde{x}$ and inversions x^α of the encoding via (11).

5.2. Multi-layer perceptrons

In this section, we present numerical results for inverting multi-layer perceptrons. In particular, we consider feedforward neural networks with convolutional layers (CNN), where in the network architecture two-dimensional convolution operations are used to represent the linear operations in the affine-linear functions $f(x, \Theta)$. Similar to the experimental design described in Section 5.1, we consider a multi-layer neural network inversion problem where we infer input image x from a noise perturbed code y^δ.

More specifically, we first train a six-layer convolutional autoencoder on the MNIST training dataset via stochastic gradient method to minimize the MSE. The encoder $\mathcal{E}(x, \Theta_\mathcal{E})$ consists of two convolutional layers, both with 4×4 convolutions with stride 2, each followed by the application of a ReLU activation function. As image spatial dimension reduces by half, we double the number of feature channels from 8 to 16. We use a fully-connected layer with weights $W_3 \in \mathbb{R}^{300\times784}$ and bias $b_3 \in \mathbb{R}^{300\times1}$ to generate the code. The decoder network first expands the code with an affine-linear transformation with weights $W_4 \in \mathbb{R}^{784\times300}$ and bias $b_4 \in \mathbb{R}^{784\times1}$. This is followed by two layers of transpose convolutions with kernel size 4×4, where each is followed by a ReLU activation function. The number of feature channels halves each time as we double the spatial dimension.

Following the implementation details outlined in Section 4.2, we iteratively compute the update steps (16) and (17) to recover x from $y = \mathcal{E}(x, \Theta)$. For the PDHG method, we choose the stepsize-parameters as $\tau_x = 1.99/\|W_1\|_2^2$ and $\tau_z = 1/(8\alpha)$. The initial values x_0 and z are both zero. The update steps stop either after reaching the maximum iterations of 1, 500 or when the improvements on x_0 and z are $<10^{-5}$ in norm. For the coordinate descent algorithm, the stepsize-parameters are set to $\tau_{x_l} = 1.99/\|W_{l+1}\|^2$ for each layer, where $\|\cdot\|$ denotes the spectral norm.

In Figure 5, we visualize the inverted images, the decoder output images, along with the groundtruth images, from the training dataset and validation dataset, respectively. For each image, α is chosen in the range $[10^{-4}, 10^{-2}]$ and set at 9×10^{-3} for both training sample images and validation sample images for best visual inversion quality.

In Figure 6, we further compare how total variation regularization and decoder respond to different levels of data noise. The noisy data is produced by adding Gaussian noise to perturb the code of each image. We start with zero mean Gaussian noise with standard deviation 0.33 and gradually reduce the noise level, this translates to decreasing δ^2 from 6.80 down to 0.00.

Please note that for each noise level the regularization factor α is manually selected in the range $[10^{-4}, 10^{-2}]$ for the best PSNR value. As we can see, for the noise level with standard deviation 0.33 where δ^2 is at 6.80, the decoder is only capable of producing a blurry

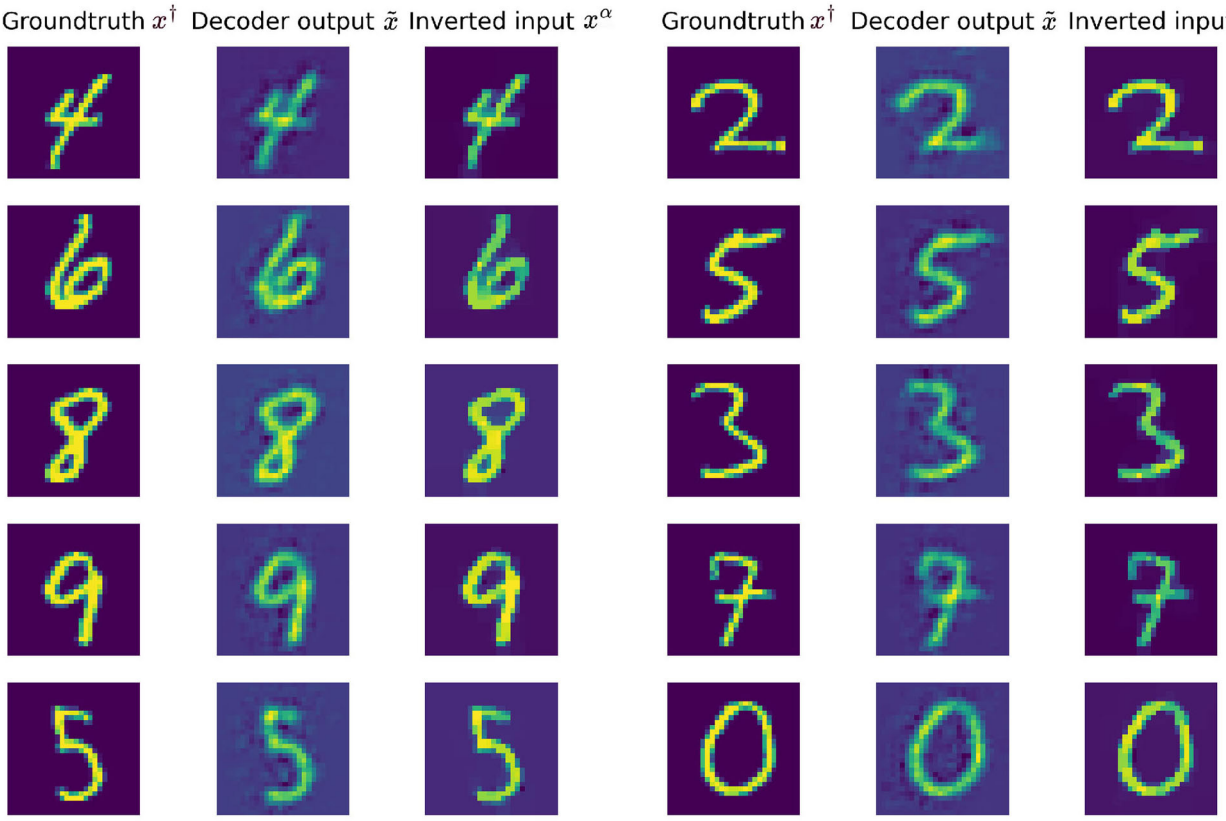

FIGURE 5
Groundtruth input images from the MNIST training dataset **(Left)** and validation dataset **(Right)**, together with the corresponding CNN autoencoder output images $\tilde{x}$ and inversions x^α of the encoding via (3).

distorted output, while the inverted image shows the structure of the digit more clearly. When we decrease the noise level down to 0.00, the inverted image becomes more clean-cut while the decoded image is still less sharply defined.

Figure 7 plots the PSNR value of the decoded and inverted image against decreasing noise level. We want to emphasize that it would be more rigorous to compute and compare $D_R^{\text{symm}}(x_{\alpha(\delta)}, x^\dagger)$ as suggested in the error estimate bound in (9), but empirically the PSNR value does also support the notion of a convergent regularization.

6. Conclusions and outlook

We have introduced a novel variational regularization framework based on a lifted Bregman formulation for the stable inversion of feed-forward neural networks (also known as multi-layer perceptrons). We have proven that the proposed framework is a convergent regularization for the single-layer perceptron case under the mild assumption that the inverse problem solution has to be in the range of the regularization operator. We have derived a general error estimate as well as a specific error estimate for the case that the activation function is the ReLU activation function. We have also addressed the extension of the theory to the multi-layer perceptron case, which can be carried out sequentially, albeit under unrealistic assumptions. We have discussed implementation strategies to solve the proposed scheme computationally, and presented numerical results for the regularized inversion of the image of a circle and piecewise constant images of hand-written digits from single- and multi-layer perceptron outputs with total variation regularization.

Despite all the positive achievements presented in this work, the proposed framework also has some limitations. The framework is currently restricted to feed-forward architectures with affine-linear transformations and proximal activation functions. While it is straight-forward to extend the framework to other architectures such as ResNets [55] or U-Nets [56], it is not straight-forward to include nonlinear operations that cannot be expressed as proximal maps of convex functions, such as max-pooling. However, for many examples there exist remedies, such as using average pooling instead of max-pooling in the previous example.

An open question is how a convergence theory without restrictive, unrealistic assumptions can be established for the multi-layer case. One issue is the non-convexity of the proposed formulation. A remedy could be the use of different architectures that lead to lifted Bregman formulations that are jointly convex in all auxiliary variables.

And last but not least, one would also like to consider other forms of regularization, such as iterative regularization, data-driven regularizations [57], or even combinations of both [58]. However, a convergence analysis for such approaches is currently an open problem.

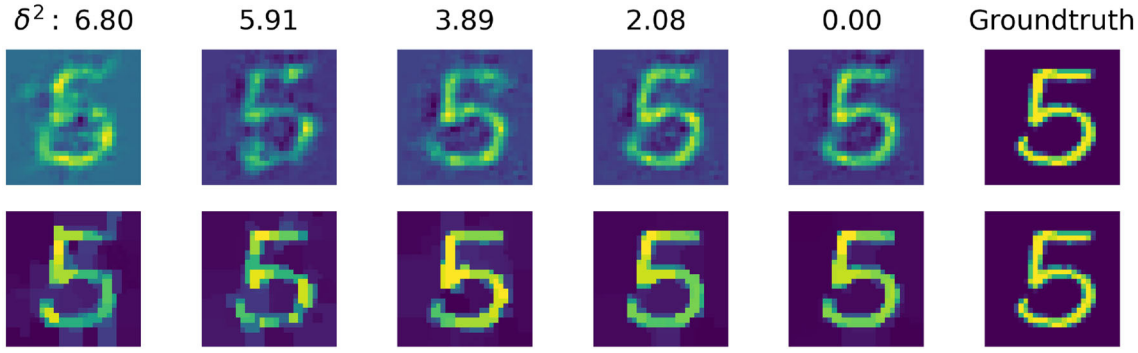

FIGURE 6
Visualization of the comparison between inverted image and decoded image against various levels of noise. **(Top)** Decoded output image from the trained convolutional autoencoder. **(Bottom)** Inverted input image from the CNN with total variation regularization.

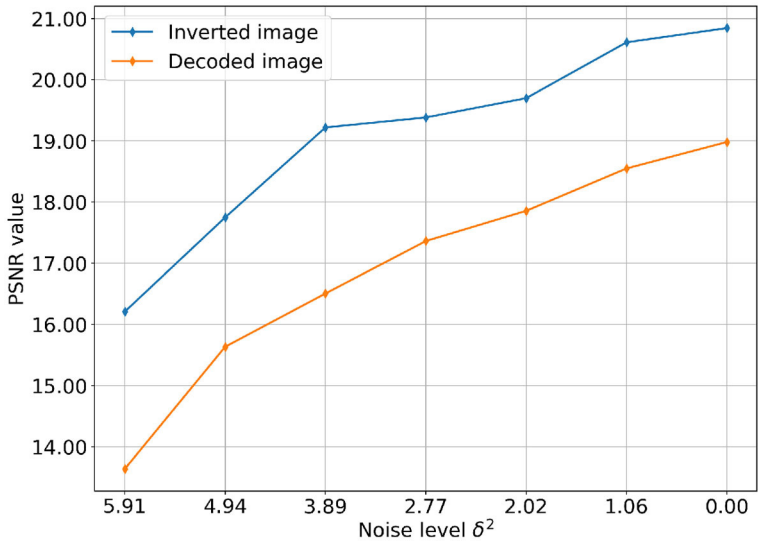

FIGURE 7
Comparison of PSNR values of total variation-based reconstruction and decoder output per noise level. Each curve reports the change of PSNR value over gradually decreasing levels of Gaussian noise, with δ^2 ranging from 0.00 to 6.80.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: http://yann.lecun.com/exdb/mnist/. The programming code for this study can be found in the University of Cambridge data repository at https://doi.org/10.17863/CAM.94404.

Author contributions

XW has programmed and contributed all numerical results as well as Sections 4 and 5. MB has contributed the introduction (Section 1) as well as the theoretical results (Section 3). XW and MB have contributed equally to Sections 2 and 6. Both authors contributed to the article and approved the submitted version.

Funding

The authors acknowledge support from the Cantab Capital Institute for the Mathematics of Information, the Cambridge Centre for Analysis (CCA), and the Alan Turing Institute (ATI).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. (2015) 521:436–44. doi: 10.1038/nature14539

2. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press (2016).

3. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: *Proceedings of the International Conference on Learning Representations* (2014). p. 1–8.

4. Fong RC, Vedaldi A. Interpretable explanations of black boxes by meaningful perturbation. In: *Proceedings of the IEEE International Conference on Computer Vision*. (2017). p. 3429–37. doi: 10.1109/ICCV.2017.371

5. Chang CH, Creager E, Goldenberg A, Duvenaud D. Explaining image classifiers by counterfactual generation. *arXiv [Preprint]*. (2019). arXiv: 1807.08024. Available online at: https://arxiv.org/pdf/1807.08024.pdf

6. Fong R, Patrick M, Vedaldi A. Understanding deep networks via extremal perturbations and smooth masks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (2019). p. 2950–8. doi: 10.1109/ICCV.2019.00304

7. Mahendran A, Vedaldi A. Understanding deep image representations by inverting them. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2015). p. 5188–96. doi: 10.1109/CVPR.2015.7299155

8. Linden A, Kindermann J. Inversion of multilayer nets. In: *Proceedings of International Joint Conference on Neural Networks*. Vol. 2 (1989). p. 425–30. doi: 10.1109/IJCNN.1989.118277

9. Kindermann J, Linden A. Inversion of neural networks by gradient descent. *Parallel Comput*. (1990) 14:277–86. doi: 10.1016/0167-8191(90)90081-J

10. Jensen CA, Reed RD, Marks RJ, El-Sharkawi MA, Jung JB, Miyamoto RT, et al. Inversion of feedforward neural networks: algorithms and applications. *Proc IEEE*. (1999) 87:1536–49. doi: 10.1109/5.784232

11. Lu BL, Kita H, Nishikawa Y. Inverting feedforward neural networks using linear and nonlinear programming. *IEEE Trans Neural Netw*. (1999) 10:1271–90. doi: 10.1109/72.809074

12. Kingma DP, Welling M. Auto-encoding variational Bayes. *arXiv preprint arXiv:13126114*. (2013).

13. Rezende D, Mohamed S. Variational inference with normalizing flows. In: *International Conference on Machine Learning*. (2015). p. 1530–8.

14. Dinh L, Krueger D, Bengio Y. Nice: Non-linear independent components estimation. In: *International Conference on Learning Representations* (2015).

15. Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. (2017). p. 2223–32. doi: 10.1109/ICCV.2017.244

16. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S. Deep unsupervised learning using nonequilibrium thermodynamics. In: *International Conference on Machine Learning*. (2015). p. 2256–65.

17. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inform Process Syst*. (2020) 33:6840–51.

18. Behrmann J, Grathwohl W, Chen RT, Duvenaud D, Jacobsen JH. Invertible residual networks. In: *International Conference on Machine Learning*. (2019). p. 573–82.

19. Behrmann J, Vicol P, Wang KC, Grosse R, Jacobsen JH. Understanding and mitigating exploding inverses in invertible neural networks. In: *International Conference on Artificial Intelligence and Statistics*. (2021). p. 1792–800.

20. Xia W, Zhang Y, Yang Y, Xue JH, Zhou B, Yang MH. GAN inversion: a survey. *IEEE Trans Pattern Anal Mach Intell*. (2022) 45:3121–38. doi: 10.1109/TPAMI.2022.3181070

21. Gal R, Alaluf Y, Atzmon Y, Patashnik O, Bermano AH, Chechik G, et al. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:220801618*. (2022).

22. Engl HW, Hanke M, Neubauer A. *Regularization of Inverse Problems*. Vol. 375. Springer Science & Business Media (1996). doi: 10.1007/978-94-009-1740-8

23. Scherzer O, Grasmair M, Grossauer H, Haltmeier M, Lenzen F. *Variational Methods in Imaging*. (2009). doi: 10.1007/978-0-387-69277-7

24. Benning M, Burger M. Modern regularization methods for inverse problems. *Acta Numer*. (2018) 27:1–111. doi: 10.1017/S0962492918000016

25. Lunz S, Öktem O, Schönlieb CB. Adversarial regularizers in inverse problems. In: *Advances in Neural Information Processing Systems*. (2018). p. 31.

26. Arridge S, Maass P, Öktem O, Schönlieb CB. Solving inverse problems using data-driven models. *Acta Numer*. (2019) 28:1–174. doi: 10.1017/S0962492919000059

27. Schwab J, Antholzer S, Haltmeier M. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Probl*. (2019) 35:025008. doi: 10.1088/1361-6420/aaf14a

28. Li H, Schwab J, Antholzer S, Haltmeier M. NETT: Solving inverse problems with deep neural networks. *Inverse Probl*. (2020) 36:065005. doi: 10.1088/1361-6420/ab6d57

29. Mukherjee S, Dittmer S, Shumaylov Z, Lunz S, Öktem O, Schönlieb CB. Learned convex regularizers for inverse problems. *arXiv preprint arXiv:200802839v2*. (2021).

30. Wang X, Benning M. Lifted Bregman training of neural networks. *arXiv preprint arXiv:220808772*. (2022).

31. Wang X, Benning M. Generalised perceptron learning. In: *12th Annual Workshop on Optimization for Machine Learning*. (2020). Available online at: https://opt-ml.org/papers/2020/paper_68.pdf

32. Bregman LM. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput Math Math Phys*. (1967) 7:200–17. doi: 10.1016/0041-5553(67)90040-7

33. Kiwiel KC. Proximal minimization methods with generalized Bregman functions. *SIAM J Control Optim*. (1997) 35:1142–68. doi: 10.1137/S0363012995281742

34. Lin J. Divergence measures based on the Shannon entropy. *IEEE Trans Inform Theory*. (1991) 37:145–51. doi: 10.1109/18.61115

35. Burbea J, Rao C. On the convexity of higher order Jensen differences based on entropy functions (Corresp.). *IEEE Trans Inform Theory*. (1982) 28:961–3. doi: 10.1109/TIT.1982.1056573

36. Burbea J, Rao C. On the convexity of some divergence measures based on entropy functions. *IEEE Trans Inform Theory*. (1982) 28:489–95. doi: 10.1109/TIT.1982.1056497

37. Nielsen F, Boltz S. The burbea-rao and bhattacharyya centroids. *IEEE Trans Inform Theory*. (2011) 57:5455–66. doi: 10.1109/TIT.2011.2159046

38. Beck A. First-order methods in optimization. *SIAM*. (2017). doi: 10.1137/1.9781611974997

39. Benning M, Burger M. Error estimates for general fidelities. *Electron Trans Numer Anal*. (2011) 38:77.

40. Chambolle A, Pock T. An introduction to continuous optimization for imaging. *Acta Numer*. (2016) 25:161–319. doi: 10.1017/S096249291600009X

41. Zhu M, Chan T. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *Ucla Cam Rep*. (2008) 34:8–34.

42. Pock T, Cremers D, Bischof H, Chambolle A. An algorithm for minimizing the Mumford-Shah functional. In: *2009 IEEE 12th International Conference on Computer Vision*. (2009). p. 1133–40. doi: 10.1109/ICCV.2009.5459348

43. Esser E, Zhang X, Chan TF. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J Imaging Sci*. (2010) 3:1015–46. doi: 10.1137/09076934X

44. Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vis*. (2011) 40:120–45. doi: 10.1007/s10851-010-0251-1

45. Benning M, Riis ES. Bregman methods for large-scale optimization with applications in imaging. In: Chen K, Schönlieb CB, Tai XC, Younces L, editors. *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*. Cham: Springer (2023). doi: 10.1007/978-3-030-03009-4_62-2

46. Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. *Phys D*. (1992) 60:259–68. doi: 10.1016/0167-2789(92)90242-F

47. Chambolle A, Lions PL. Image recovery via total variation minimization and related problems. *Numer Math*. (1997) 76:167–88. doi: 10.1007/s002110050258

48. Beck A, Tetruashvili L. On the convergence of block coordinate descent type methods. *SIAM J Optim*. (2013) 23:2037–60. doi: 10.1137/120887679

49. Wright SJ. Coordinate descent algorithms. *Math Programm*. (2015) 151:3–34. doi: 10.1007/s10107-015-0892-3

50. Wright SJ, Recht B. *Optimization for Data Analysis*. Cambridge University Press (2022). doi: 10.1017/9781009004282

51. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. (1998) 86:2278–324. doi: 10.1109/5.726791

52. Landweber L. An iteration formula for Fredholm integral equations of the first kind. *Am J Math*. (1951) 73:615–24. doi: 10.2307/2372313

53. Morozov VA. *Methods for Solving Incorrectly Posed Problems*. Springer Science & Business Media (2012).

54. Chambolle A. An algorithm for total variation minimization and applications. *J Math Imaging Vis*. (2004) 20:89–97. doi: 10.1023/B:JMIV.0000011321.19549.88

55. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2016). p. 770–8. doi: 10.1109/CVPR.2016.90

56. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference*. Munich: Springer (2015). p. 234–41. doi: 10.1007/978-3-319-24574-4_28

57. Kabri S, Auras A, Riccio D, Bauermeister H, Benning M, Moeller M, et al. Convergent data-driven regularizations for CT reconstruction. *arXiv [Preprint]*. (2022). arXiv: 2212.07786. Available online at: https://arxiv.org/pdf/2212.07786.pdf

58. Aspri A, Banert S, Öktem O, Scherzer O. A data-driven iteratively regularized Landweber iteration. *Numer Funct Anal Optim*. (2020) 41:1190–227. doi: 10.1080/01630563.2020.1740734