
Differentiable Linearized ADMM

Xingyu Xie^{* 1} Jianlong Wu^{* 1} Zhisheng Zhong¹ Guangcan Liu² Zhouchen Lin¹

Abstract

Recently, a number of learning-based optimization methods that combine data-driven architectures with the classical optimization algorithms have been proposed and explored, showing superior empirical performance in solving various ill-posed inverse problems, but there is still a scarcity of rigorous analysis about the convergence behaviors of learning-based optimization. In particular, most existing analyses are specific to unconstrained problems but cannot apply to the more general cases where some variables of interest are subject to certain constraints. In this paper, we propose Differentiable Linearized ADMM (D-LADMM) for solving the problems with linear constraints. Specifically, D-LADMM is a K -layer LADMM inspired deep neural network, which is obtained by firstly introducing some learnable weights in the classical Linearized ADMM algorithm and then generalizing the proximal operator to some learnable activation function. Notably, we rigorously prove that there exist a set of learnable parameters for D-LADMM to generate globally converged solutions, and we show that those desired parameters can be attained by training D-LADMM in a proper way. To the best of our knowledge, we are the first to provide the convergence analysis for the learning-based optimization method on constrained problems.

1. Introduction

Numerous problems solving at the core of statistics, learning and vision areas rely on well-designed optimization algorithms, and especially so for the recently prevalent deep

learning. Provided with some well-deigned optimization strategies such as (Kingma & Ba, 2014; Zeiler, 2012; Li et al., 2019), the researchers can focus on the design of task-oriented loss functions without being encumbered by the solving methods. In addition, optimization can also help the deep neural network (DNN) design. For example, Li et al. (2018) show that optimization algorithms can in fact inspire the architectures of DNN, and they connect the classical optimization algorithms with some prevalent DNN architectures, e.g., ResNet (He et al., 2016) and DenseNet (Huang et al., 2017). While it is apparent that optimization does benefit learning, the converse of the statement is not so affirmative. That is, can the well-developed learning methods also benefit the optimization? If so, in what sense?

To answer the highlighted question, some techniques have been proposed to combine data-driven learning frameworks with the traditional optimization algorithms, so called as *learning-based optimization* (Gregor & LeCun, 2010; Liu et al., 2016; Chen & Pock, 2017; Liu et al., 2018a; Peng et al., 2018). Usually, the combination is achieved by introducing learnable parameters into the classical numerical solvers at first, then performing discriminative learning on collected training data so as to obtain some task-specific (but possibly inconsistent) optimization schemes. Due to the success of deep learning in a wide variety of application fields, many researchers choose to consider DNN as the learnable units for being combined with the optimization procedure. For example, Sprechmann et al. (2015); Liu et al. (2019); Chen et al. (2018) resemble a recurrent neural network (RNN) to solve the LASSO problem, and Zhou et al. (2018) show the connection between sparse coding and long short term memory (LSTM). The empirical results in these studies illustrate that the computational efficiency of optimization is dramatically improved by the incorporation of DNN. However, there is only few work that analyzes the convergence properties of these algorithms in theory. Chen et al. (2018) prove that there exist a sequence of parameters for their learning-based optimization procedure to converge linearly to the optimal solution of the LASSO problem. But this result is specific to LASSO and may not apply to the other problems.

While most existing methods and theories in learning-based optimization are made specific to unconstrained problems, many optimization problems arising from modern appli-

^{*}Equal contribution.

¹Key Lab. of Machine Perception, School of EECS, Peking University.

²B-DAT and CICAET, School of Automation, Nanjing University of Information Science and Technology.

Correspondence to: Guangcan Liu <gcliu@nuist.edu.cn>, Zhouchen Lin <zlin@pku.edu.cn>.

cations may contain some constraints. In this paper, we would like to take a step towards learning-based *constrained* optimization. To be more precise, we shall consider the following linearly constrained problem:

$$\min_{\mathbf{Z}, \mathbf{E}} f(\mathbf{Z}) + g(\mathbf{E}), \quad \text{s.t. } \mathbf{X} = \mathbf{AZ} + \mathbf{BE}, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{m \times d_1}$, $\mathbf{B} \in \mathbb{R}^{m \times d_2}$, $\mathbf{X} \in \mathbb{R}^{m \times n}$, and $f(\cdot)$ and $g(\cdot)$ are convex functions. Many problems in the learning field can be formulated as (1), e.g., matrix recovery (Zhang et al., 2018; 2015; Liu & Li, 2016; Liu et al., 2017), subspace clustering (You et al., 2016; Liu et al., 2013), image deblurring (Liu et al., 2014) and so on. To solve the problem in (1), the Linearized ADMM (LADMM) algorithm established by (Lin et al., 2011) is a desirable choice. But LADMM generally needs hundreds or more iterations to converge and is therefore time consuming; this motivates us to seek a learning-based version of LADMM. However, due to the presence of the equality constraint, existing theories are no longer applicable. As a consequence, we need to invent new algorithm design and theoretical analysis to address properly the following questions:

1. How to combine the deep learning strategy with LADMM so as to solve the constrained problem in (1)?
2. What is the relation between the learning-based LADMM and original LADMM? Specifically, does the output of learning-based LADMM still obey the linear constraint? And, most importantly, can the learning based LADMM still ensure convergence rate?

To make LADMM learnable, first of all, we convert the proximal operator in LADMM to a special neural network structure. Then we replace the given matrix \mathbf{A} and \mathbf{B} with some learnable weights and, meanwhile, expand the dimension of the penalty parameter such that the penalties on different directions are learnable as well, resulting in a novel method termed Differentiable LADMM (D-LADMM). What is more, we prove that, under some mild conditions, there do exist a set of learnable parameters that ensure D-LADMM to achieve a linear rate of convergence, and we show that those desired parameters can be attained by training D-LADMM properly. Interestingly, our results illustrate that it is possible for D-LADMM to possess a decay rate of linear convergence smaller than that of LADMM, which means that D-LADMM could converge faster than LADMM (note that LADMM is not linearly convergent unless the objective function is strongly convex). In summary, the main contributions of this paper include:

- We propose a learning-based method called D-LADMM for solving the constrained optimization

problem in (1). It is worth noting that our techniques, mainly including the proximal operator inspired network structure and the proposed policies for dealing with the linear constraint, would be useful for solving the other constrained problems.

- As for convergence, due to the high flexibility of the learnable modules, it is difficult to assert the convergence of learning-based optimization. Remarkably, we establish a rigorous analysis on the convergence properties of D-LADMM. Our analysis shows that D-LADMM still satisfies the linear constraint and may converge faster than LADMM in some cases.

To the best of our knowledge, we are the *first* to provide convergence analysis for learning-based optimization method under the context of constrained problems.

The remainder of the paper is organized as follows. We review some related work in Section 2. In Section 3, we start with a warm-up case to introduce how to convert the proximal operator in LADMM as a shallow neural network, and, accordingly, we establish the so-called D-LADMM. We analyze the convergence properties of D-LADMM in Section 4. Finally, empirical results that verify the proposed theories are given in Sections 5.

2. Related Work

When ignoring the equality constraint of the problem in (1), there already exist some learning-based algorithms equal to the task, but most of them provide no convergence analysis. We have spotted only one theoretical article; namely, Chen et al. (2018) unroll the optimization procedure of LASSO as a RNN and prove that the resulted learning-based algorithm can achieve a linear rate of convergence. This result is significant but, to our knowledge, there is no similar conclusion available for constrained problems. The majority of the literature is consisting of empirical studies, e.g., (Ulyanov et al., 2018; Zhang et al., 2017; Diamond et al., 2017) consider DNN as implicit priors for image restoration. The problems addressed by these methods are in fact special cases of problem (1); namely, their formulations can be obtained by removing the constraint as well as some regularization term from (1). Due to the lack of theoretical analysis, it is unclear when or where their DNN dominant solution sequences should be terminated.

Yang et al. (2016) recast the ADMM procedure as some learnable network, called ADMM-Net, and they apply it to a compressive sensing based Magnetic Resonance Imaging (MRI) problem that is indeed a special case of problem (1). The authors show that ADMM-Net performs well in MRI, but there is no guarantee for ensuring the convergence of their algorithm. Notice that our proposed D-LADMM is built upon LADMM rather than ADMM. Comparing to

Table 1. Summary of notations in this paper.

a	A scalar.	\mathbf{A}	A matrix.
\mathbf{a}	A vector.	$\mathcal{A}(\cdot)$	An operator.
$\mathbf{A} \succ 0$	A positive-definite matrix.	$\mathcal{A} \succ 0$	A positive-definite operator, $\langle \mathcal{A}(\mathbf{Z}), \mathbf{Z} \rangle > 0, \forall \mathbf{Z} \neq 0$.
\mathbf{I}_m	$\mathbf{I}_m \in \mathbb{R}^{m \times m}$ identity matrix.	\circ	Hadamard product (entrywise product).
$\ \mathbf{a}\ _2$	$\ \mathbf{a}\ _2 = \sqrt{\sum_i a_i^2}$.	$\ \mathbf{A}\ _F$	$\ \mathbf{A}\ _F = \sqrt{\sum_{ij} A_{ij}^2}$.
$\ \mathbf{A}\ _1$	$\ \mathbf{A}\ _1 = \sum_{ij} A_{ij} $.	$\ \mathbf{A}\ $	Maximum singular value.
$\ \omega\ _{\mathcal{H}}^2$	$\langle \omega, \mathcal{H}(\omega) \rangle$.	$\text{dist}_{\mathcal{H}}^2(\omega, \Omega^*)$	$\min_{\omega^* \in \Omega^*} \ \omega - \omega^*\ _{\mathcal{H}}^2$.
\mathcal{D}	Linear operator defined in (12).	β	Positive matrix with $\beta_{ij} > 0$.
ω	$(\mathbf{Z}, \mathbf{E}, -\lambda)^\top$.	\mathbf{u}	$(\mathbf{Z}, \mathbf{E})^\top$.
$\beta^{-1}, 1/\beta$	The ij -th entry being $1/\beta_{ij}$.	$\mathcal{H}(\omega)$	$\mathcal{H}(\omega) = (\mathcal{D}(\mathbf{Z}), \beta \circ \mathbf{E}, -(\beta)^{-1} \circ \lambda)^\top$.
$h(\mathbf{u})$	$f(\mathbf{Z}) + g(\mathbf{E})$.	$\mathcal{F}_k(\omega)$	$(\mathbf{W}_k^\top \lambda, \lambda, \mathbf{A}\mathbf{Z} + \mathbf{B}\mathbf{E} - \mathbf{X})^\top$.
$\mathcal{G}_k(\cdot)$	$(\mathbf{W}_k, \mathbf{I}, 0)^\top \beta_k \circ (\cdot)$.	$\phi(\omega)$	$(\mathbf{A}^\top \lambda + \partial f(\mathbf{Z}), \lambda + \partial g(\mathbf{E}), \mathbf{A}\mathbf{Z} + \mathbf{B}\mathbf{E} - \mathbf{X})^\top$.
$d^*(\cdot)$	Lagrange dual function of (1).	Ω^*	The solution set of (1).

ADMM, LADMM needs fewer auxiliary variables to solve the constrained problems like (1). This detail is important, because fewer auxiliary variables means fewer learnable parameters while recasting the algorithm to DNN and, consequently, the reduction in the number of learnable parameters can accelerate the training process.

3. Differentiable Linearized ADMM

In this section, we shall begin with a warm-up case to show the conversion from a proximal iteration to some DNN block. Then we introduce the proposed D-LADMM in detail. The main notations used throughout this paper are summarized in Table 1.

3.1. Warm-Up: Differentiable Proximal Operator

We first show how to differentialize the proximal operator as a network block. Consider the following unconstrained problem with two objective components:

$$\min_{\mathbf{z}} f(\mathbf{z}) + \frac{1}{2} \|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2, \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{m \times d}$, $\mathbf{z} \in \mathbb{R}^d$, $\mathbf{b} \in \mathbb{R}^m$, and $f(\cdot)$ is a real-valued convex function. The proximal gradient algorithm solves problem (2) as follows:

$$\mathbf{z}_k = \text{prox}_{t f}(\mathbf{z}_{k-1} - t \mathbf{A}^\top (\mathbf{A}\mathbf{z}_{k-1} - \mathbf{b})), \quad (3)$$

where $t > 0$ is the step size and prox is the proximal operator given by

$$\text{prox}_f(x) = \underset{z}{\text{argmin}} f(z) + \frac{1}{2} \|z - x\|_2^2.$$

As pointed out by (Lin et al., 2011; Zhang et al., 2010; Blumensath & Davies, 2008; Bot & Nguyen, 2018), the strict convergence of the proximal gradient procedure (3) relies on the condition $t \|\mathbf{A}\|^2 < 1$; that is,

$$\frac{1}{t} \mathbf{I} - \mathbf{A}^\top \mathbf{A} \succ 0. \quad (4)$$

The iteration (3) is quite similar to a local block of DNN, i.e., $\mathbf{z}_k = \Phi(\mathbf{W}_k \mathbf{z}_{k-1} + \mathbf{b}_k)$, where $\Phi(\cdot)$ is an NN block. Actually, the proximal operator connects tightly to the non-linear activation function in DNN. In some cases, they share the same form. For example, given $\mathbf{a} \in \mathbb{R}^d$, we may set the function $f(\cdot)$ as $f(\mathbf{x}) = \sum_{i=1}^d f_i(x_i)$, where

$$f_i(x_i) = \begin{cases} \frac{1}{2}(x_i - a_i)^2 + c, & x_i < 0, \\ 0, & x_i \geq 0. \end{cases}$$

Then the proximal operator coincides with ReLU, a commonly used non-linear activation function for DNN; namely, $\text{prox}_f(\mathbf{a}) = \text{ReLU}(\mathbf{a})$, where $\text{ReLU}(a_i) = \max\{a_i, 0\}$. We can see that the proximal operator shares almost the same role as the activation function. Inspired by this, we can transform the iteration (3) into a network structure, namely Differentiable Proximal Operator:

$$\mathbf{z}_k = \zeta(\mathbf{z}_{k-1} - \mathbf{W}_{k-1}^\top (\mathbf{A}\mathbf{z}_{k-1} - \mathbf{b})), \quad (5)$$

where $\zeta(\cdot)$ is some non-linear activation function and \mathbf{W}_{k-1} is the learnable parameter. As shown above, with proper $\zeta(\cdot)$ and function $f(\cdot)$, (3) and (5) can be the same.

3.2. Differentiable Linearized ADMM

First of all, we shall revisit the Linearized ADMM (LADMM) algorithm (Lin et al., 2011). Given $\mathbf{A} \in \mathbb{R}^{m \times d_1}$, $\mathbf{B} \in \mathbb{R}^{m \times d_2}$, $\mathbf{X} \in \mathbb{R}^{m \times n}$ as well as two real-valued convex functions, $f(\cdot)$ and $g(\cdot)$, the iterative scheme of LADMM for solving (1) reads as:

$$\begin{cases} \mathbf{T}_{k+1} = \mathbf{A}\mathbf{Z}_k + \mathbf{B}\mathbf{E}_k - \mathbf{X}, \\ \mathbf{Z}_{k+1} = \text{prox}_{\frac{f}{L_1}} \left\{ \mathbf{Z}_k - \frac{1}{L_1} \mathbf{A}^\top (\lambda_k + \beta \mathbf{T}_{k+1}) \right\}, \\ \hat{\mathbf{T}}_{k+1} = \mathbf{A}\mathbf{Z}_{k+1} + \mathbf{B}\mathbf{E}_k - \mathbf{X}, \\ \mathbf{E}_{k+1} = \text{prox}_{\frac{g}{L_2}} \left\{ \mathbf{E}_k - \frac{1}{L_2} \mathbf{B}^\top (\lambda_k + \beta \hat{\mathbf{T}}_{k+1}) \right\}, \\ \lambda_{k+1} = \lambda_k + \beta (\mathbf{A}\mathbf{Z}_{k+1} + \mathbf{B}\mathbf{E}_{k+1} - \mathbf{X}), \end{cases} \quad (6)$$

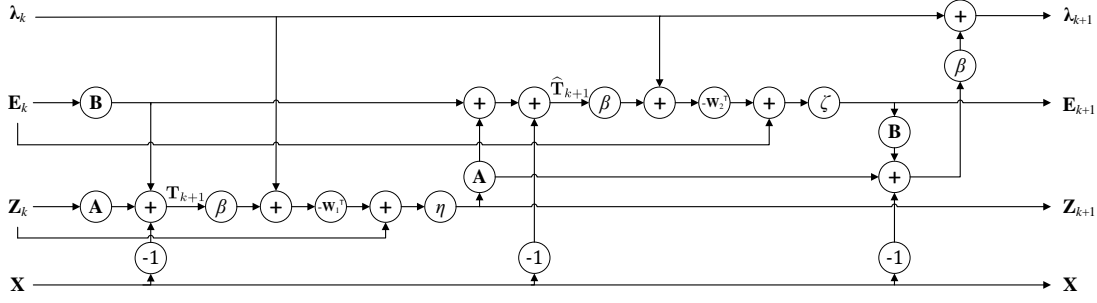


Figure 1. One block structure of the proposed D-LADMM. As we can see, such a LADMM inspired differentiable block reflects some prevalent structures, such as residual connection (He et al., 2016) and dense connection (Huang et al., 2017).

where λ is Lagrange multiplier, $L_1 > 0$ and $L_2 > 0$ are Lipsitz constants, and $\beta > 0$ is penalty parameter.

Following the spirits of learning-based optimization, we propose a DNN named Differentiable LADMM (D-LADMM). One iteration of the original LADMM is generalized as a block of neural network. Specifically, we retain the updating rule for \mathbf{T}_{k+1} , $\hat{\mathbf{T}}_{k+1}$ and the Lagrange multiplier λ , replace the two proximal steps in (6) by differentiable proximal operators and, meanwhile, expand the dimension of the penalty parameter β such that the penalties on different directions are learnable as well. In summary, one block of our D-LADMM is given by

$$\begin{cases} \mathbf{T}_{k+1} = \mathbf{A}\mathbf{Z}_k + \mathbf{B}\mathbf{E}_k - \mathbf{X}, \\ \mathbf{Z}_{k+1} = \eta_{(\theta_1)_k} (\mathbf{Z}_k - (\mathbf{W}_1)_k^\top (\lambda_k + \beta_k \circ \mathbf{T}_{k+1})), \\ \hat{\mathbf{T}}_{k+1} = \mathbf{A}\mathbf{Z}_{k+1} + \mathbf{B}\mathbf{E}_k - \mathbf{X}, \\ \mathbf{E}_{k+1} = \zeta_{(\theta_2)_k} (\mathbf{E}_k - (\mathbf{W}_2)_k^\top (\lambda_k + \beta_k \circ \hat{\mathbf{T}}_{k+1})), \\ \lambda_{k+1} = \lambda_k + \beta_k \circ (\mathbf{A}\mathbf{Z}_{k+1} + \mathbf{B}\mathbf{E}_{k+1} - \mathbf{X}), \end{cases} \quad (7)$$

where $\Theta = \{(\mathbf{W}_1)_k, (\mathbf{W}_2)_k, (\theta_1)_k, (\theta_2)_k, \beta_k\}_{k=0}^K$ are learnable matrices, and \circ is the element-wise product. In addition, $\eta(\cdot)$ and $\zeta(\cdot)$ are some non-linear functions parameterized by θ_1 and θ_2 , respectively.

It is worth mentioning that we intentionally keep the matrices \mathbf{A} and \mathbf{B} in the updating step for \mathbf{T}_{k+1} , $\hat{\mathbf{T}}_{k+1}$ and λ . The reason is that, instead of leaving everything for NN to learn, our scheme can in fact benefit the compliance of the equality constraint in (1) so as to help reveal the connection between D-LADMM and LADMM.

D-LADMM, a block of which is illustrated in Figure 1, actually corresponds to a K -layer feed-forward neural network with side connections. Many empirical results, e.g., (Gregor & LeCun, 2010; Wang et al., 2016; Yang et al., 2016; Peng et al., 2018), show that a well-trained K -layer differentiable optimization inspired model—compared with the original optimization algorithm—can obtain almost the same good solution within one or even two order-of-magnitude fewer iterations. Moreover, the quality of the output from each layer will be gradually improved.

Among the other things, it is also feasible to expand the parametric space of D-LADMM by introducing more learnable units. For example, we can generalize the linear transformation \mathbf{A} to some non-linear mapping $\mathcal{A}_{\theta_1}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ parameterized by θ_1 , so as to learn adaptively some proper representation from data. But a theoretical analysis for such models has to be much more involved.

Training Strategy: Different from LADMM which has no parameter to learn, D-LADMM is treated as a special structured neural network and trained using stochastic gradient descent (SGD) over the observation \mathbf{X} , and all the parameters Θ are subject to learning. Depending on whether the ground truth (i.e., true optimal solution to (1)) is given, we present two different training strategies. Without the ground truth, the training process is actualized by

$$\min_{\Theta} f(\mathbf{Z}_K) + g(\mathbf{E}_K) - d^*(\lambda_K), \quad (8)$$

where $d^*(\lambda_K)$ is the dual function of (1) defined as $d^*(\lambda_K) = \inf_{\mathbf{Z}, \mathbf{E}} f(\mathbf{Z}) + g(\mathbf{E}) + \langle \lambda_K, \mathbf{A}\mathbf{Z} + \mathbf{B}\mathbf{E} - \mathbf{X} \rangle$. When the functions $f(\cdot)$ and $g(\cdot)$ are given, we can obtain explicitly the dual function $d^*(\cdot)$, which is a concave function bounded from above by $f(\cdot) + g(\cdot)$; this means the objective in (8) is always non-negative. Moreover, due to the convexity of $f(\cdot) + g(\cdot)$, the minimum attainable value of (8) is exactly zero. In other words, the global optimum is attained whenever the objective reaches zero.

In the cases where the ground-truth \mathbf{Z}^* and \mathbf{E}^* are provided along with the training samples, D-LADMM can be trained by simply minimizing the following square loss:

$$\min_{\Theta} \|\mathbf{Z}_K - \mathbf{Z}^*\|_F^2 + \|\mathbf{E}_K - \mathbf{E}^*\|_F^2. \quad (9)$$

4. Convergence Analysis

In this section, we formally establish the convergence of the proposed D-LADMM, based on some specific settings. Although learning-based optimization methods have achieved a great success in practice, their merits have not been validated sufficiently from the theoretical perspective. Chen et al. (2018) and Liu et al. (2019) provide the convergence analysis for Unfolded Iterative Shrinkage and Thresholding

Algorithm (ISTA). However, their techniques are applicable only to unconstrained problems and thus not very helpful for analyzing D-LADMM, which contains an equality constraint and some complex updating rules. In general, due to the presence of non-linear learnable functions, it is hard, if not impossible, to analyze theoretically the model in (7). Fortunately, with some proper settings, we can still accomplish a rigorous analysis for D-LADMM.

4.1. Settings

Please notice that the analytical approach remains the same while one of \mathbf{A} and \mathbf{B} is suppressed as the identity matrix. Thus, for the sake of simplicity, we omit the given matrix \mathbf{B} from problem (1). We shall also focus on the cases where η and ζ are respectively the proximal operators of the functions $f(\cdot)$ and $g(\cdot)$. In other words, we consider the following simplified D-LADMM for analysis:

$$\begin{cases} \mathbf{T}_{k+1} = \mathbf{A}\mathbf{Z}_k + \mathbf{E}_k - \mathbf{X}, \\ \mathbf{Z}_{k+1} = \text{prox}_{f_{\theta_k}} \left(\mathbf{Z}_k - \frac{1}{\theta_k} \circ [\mathbf{W}_k^\top (\lambda_k + \beta_k \circ \mathbf{T}_{k+1})] \right), \\ \mathbf{E}_{k+1} = \text{prox}_{g_{\beta_k}} \left(\mathbf{X} - \mathbf{A}\mathbf{Z}_{k+1} - \frac{1}{\beta_k} \circ \lambda_k \right), \\ \lambda_{k+1} = \lambda_k + \beta_k \circ (\mathbf{A}\mathbf{Z}_{k+1} + \mathbf{E}_{k+1} - \mathbf{X}), \end{cases} \quad (10)$$

where $\text{prox}_{g_{\beta}}(\mathbf{R}) = \arg\min_{\mathbf{E}} \{g(\mathbf{E}) + \frac{\beta}{2} \circ \|\mathbf{E} - \mathbf{R}\|_F^2\}$ and $\beta_k, \theta_k > 0$.

By the definition of the proximal operator, we rewrite the above (10) as follows:

$$\begin{cases} \mathbf{Z}_{k+1} = \arg\min_{\mathbf{Z}} \left\{ f(\mathbf{Z}) + \frac{\theta_k}{2} \circ \|\mathbf{Z} - \mathbf{Z}_k + (\theta_k)^{-1} \circ \mathbf{W}_k^\top (\lambda_k + \beta_k \circ (\mathbf{A}\mathbf{Z}_k + \mathbf{E}_k - \mathbf{X}))\|_F^2 \right\}, \\ \mathbf{E}_{k+1} = \arg\min_{\mathbf{E}} \left\{ g(\mathbf{E}) + \frac{\beta_k}{2} \circ \|\mathbf{E} - \mathbf{X} + \mathbf{A}\mathbf{Z}_{k+1} + (\beta_k)^{-1} \circ \lambda_k\|_F^2 \right\}, \\ \lambda_{k+1} = \lambda_k + \beta_k \circ (\mathbf{A}\mathbf{Z}_{k+1} + \mathbf{E}_{k+1} - \mathbf{X}). \end{cases} \quad (11)$$

The above is indeed a special case of D-LADMM (7), with learnable parameters $\Theta = \{(\mathbf{W}_k \in \mathbb{R}^{m \times d}, \theta_k \in \mathbb{R}^{d \times n}, \beta_k \in \mathbb{R}^{m \times n})\}_{k=0}^K$.

As aforementioned, the success of proximal operator based methods may rely on the condition in (4). Given $(\mathbf{W}, \theta, \beta)$, the positive-definite matrix in (4) becomes a linear operator $\mathcal{D} : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}$ given by

$$\mathcal{D}(\mathbf{Z}) = \theta \circ (\mathbf{Z}) - \mathbf{W}^\top \beta \circ (\mathbf{A}\mathbf{Z}), \quad \forall \mathbf{Z} \in \mathbb{R}^{d \times n}. \quad (12)$$

It is known that, to ensure the convergence of LADMM, the positive-definiteness condition in (4) has to be obeyed. For the same reason, the convergence of D-LADMM necessitates the positive-definiteness of the operator \mathcal{D} . Based on

this, we define the following set:

$$\mathcal{S}(\sigma, \mathbf{A}) \triangleq \{(\mathbf{W}, \theta, \beta) \mid \|\mathbf{W} - \mathbf{A}\| \leq \sigma, \mathcal{D} \succ 0, \beta, \theta > 0\}, \quad (13)$$

where $\mathcal{D} \succ 0$ means that the operator \mathcal{D} is a positive-definite operator. The above set, $\mathcal{S}(\sigma, \mathbf{A})$, could be non-empty if a proper \mathbf{A} is given. For example, when $\mathbf{W} = \mathbf{A}$ and both β and θ degenerate to scalars, the non-emptiness of $\mathcal{S}(\sigma, \mathbf{A})$ is actually equivalent to the classical condition (4). In general, $\|\mathbf{W} - \mathbf{A}\| < \sigma$ ensures that the learnable weight is close to \mathbf{A} , which guarantees that the first minimization problem in (11) has an analogous optimization landscape with the original one, and $\mathcal{D} \succ 0$ ensures that the local approximation in the proximal operator is a strict upper bound of the original objective. Thus, we can treat the non-emptiness of the set $\mathcal{S}(\sigma, \mathbf{A})$ as a generalization of the condition (4), and we would make an assumption as follows.

Assumption 1. *There exists a constant c such that $\mathcal{S}(\sigma, \mathbf{A})$ is non-empty for any σ that satisfies $0 \leq \sigma \leq c$, namely the given \mathbf{A} is proper.*

4.2. Convergence Property

Before proving the main theorem, we shall establish some basic lemmas by which the candidate parameters set for $\Theta = \{\mathbf{W}_k, \theta_k, \beta_k\}_{k=0}^K$ can be derived. Note that, for all the lemmas and theorems in this section, we assume that the Assumption 1 is satisfied.

Define an operator \mathcal{H}_k as

$$\mathcal{H}_k(\cdot) = \begin{pmatrix} \mathcal{D}_k & 0 & 0 \\ 0 & \beta_k \circ & 0 \\ 0 & 0 & (\beta_k)^{-1} \circ \end{pmatrix} (\cdot), \quad (14)$$

where \mathcal{D}_k is the operator (12) given $(\mathbf{W}_k, \theta_k, \beta_k)$.

Firstly, we prove an inequality for the objective $h(\mathbf{u}_k)$.

Lemma 4.1. *Let the sequence $\{\omega_k\}$ be generated by (11). Then we have:*

$$\begin{aligned} h(\mathbf{u}) - h(\mathbf{u}_{k+1}) + \langle \omega - \omega_{k+1}, \mathcal{F}_k(\omega_{k+1}) + \mathcal{G}_k(\mathbf{E}_k - \mathbf{E}_{k+1}) + \mathcal{H}_k(\omega_{k+1} - \omega_k) \rangle &\geq 0, \quad \forall \omega, \end{aligned} \quad (15)$$

where $\mathcal{F}_k(\cdot)$ and $\mathcal{G}_k(\cdot)$ are simply two linear operators as defined in Table 1.

Lemma 4.1 suggests that the quantity $\|\omega_{k+1} - \omega_k\|_{\mathcal{H}_k}^2$ could be used to measure the distance between the iterate ω_{k+1} to the solution set Ω^* . In other words, when $\|\omega_{k+1} - \omega_k\|_{\mathcal{H}_k}^2 = 0$, the positive-definiteness of $\mathcal{H}_k(\cdot)$ gives $\omega_{k+1} - \omega_k = 0$. If $\mathbf{W}_k = \mathbf{A}$, we have

$$h(\mathbf{u}) - h(\mathbf{u}_{k+1}) + \langle \omega - \omega_{k+1}, \mathcal{F}_k(\omega_{k+1}) \rangle \geq 0, \quad \forall \omega,$$

which implies that ω_{k+1} is a solution of problem (1).

Lemma 4.2. *Let the sequence $\{\omega_k\}$ be generated by (11). Suppose that, for any point $\omega^* \in \Omega^*$, there exists proper $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$ such that:*

$$\langle \omega_{k+1} - \omega^*, \mathcal{H}_k(\omega_k - \omega_{k+1}) \rangle \geq 0, \quad \forall k \geq 0, \quad (16)$$

where $\mathcal{H}_k(\cdot)$ is given in (14). Then $\|\omega_k\|_F < \infty$ holds for all k , and we have:

$$\|\omega_k - \omega^*\|_{\mathcal{H}_k}^2 \geq \|\omega_{k+1} - \omega^*\|_{\mathcal{H}_k}^2 + \|\omega_k - \omega_{k+1}\|_{\mathcal{H}_k}^2. \quad (17)$$

Lemma 4.2 shows that there exist proper learnable parameters that make $\{\omega_k\}$ strictly contractive with respect to the solution set Ω^* . It is worth mentioning that the proof of Lemma 4.2 may partly explain why D-LADMM converges faster than LADMM. Denote $\|\mathbf{W}_k - \mathbf{A}\| = \sigma_k$. From the proof process, we find that, when $\|\mathbf{E}_{k+1} - \mathbf{E}_k\|_F$ is large and $\|\mathbf{Z}_{k+1} - \mathbf{Z}^*\|_F$ is small, σ_k can be set as a large value, which means the feasible space of the learnable weight is large as well. Conversely, the better weight retrieved from the larger feasible space can also promote the convergence speed. In one word, the sequence $\{\sigma_k\}_{k=0}^K$ is somehow learnt adaptably so as to benefit global convergence.

We now show that, whenever there is no solution that satisfies $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$ and $\omega_{k+1} \neq \omega_k$, the optimum is attained.

Lemma 4.3. *Let the sequence $\{\omega_k\}$ be generated by (11). Given ω_k , if the updating rule in (11) achieves $\omega_{k+1} = \omega_k$ for all $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$, then $\omega_k = \omega_{k+1} \in \Omega^*$.*

4.3. Main Results

Equipped with the above lemmas, we establish a couple of theorems to guarantee the convergence of D-LADMM (11).

Theorem 1 (Convergence of D-LADMM). *Let the sequence $\{\omega_k\}$ be generated by (11). There exists proper $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$ such that $\{\omega_k\}$ converges to a solution $\omega^* \in \Omega^*$.*

So, in general, there exist proper parameters such that the outputs of our D-LADMM converge to the optimal solution of problem (1). In the rest of this subsection, we will further investigate its convergence rate, which is measured by a point-to-set distance $\text{dist}_{\mathcal{H}}^2(\omega, \Omega^*)$. The following theorem shows that we can find a set of parameters to make the distance decrease monotonically.

Theorem 2 (Monotonicity of D-LADMM). *Let the sequence $\{\omega_k\}$ be generated by (11). There exists proper $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$ such that $\text{dist}_{\mathcal{H}_k}^2(\omega_k, \Omega^*)$ decreases monotonically when k is large enough.*

The above theorem proves the monotonic decreasing property of the deviation between the produced solution ω_k and the true solution set Ω^* . This is different from Lemma 4.2 in which the distance is measured by a point-to-point metric.

For convenience, we combine all the updating rules in (11) into a single operator \mathcal{T} :

$$\mathcal{T}(\mathbf{W}_k, \theta_k, \beta_k)(\omega_k) = \omega_{k+1}.$$

Next, we will show that, under some condition imposed on \mathcal{T} , D-LADMM can attain a linear rate of convergence.

Theorem 3 (Convergence Rate of D-LADMM). *Let the sequence $\{\omega_k\}$ be generated by (11). Suppose that there exist $(\mathbf{A}, \theta^*, \beta^*)$ and $K_0 > 0$ such that for any $k \geq K_0$ (i.e., k is large enough) the following holds:*

$$(\text{EBC}): \quad \text{dist}_{\mathcal{H}^*}^2(\tilde{\omega}, \Omega^*) \leq \frac{\kappa}{16} \|\tilde{\omega} - \omega_k\|_{\mathcal{H}^*}^2, \quad (18)$$

where $\mathcal{H}^*(\cdot)$ is given in (14) by setting $(\mathbf{W}_k, \theta_k, \beta_k)$ as $(\mathbf{A}, \theta^*, \beta^*)$ and $\tilde{\omega} = \mathcal{T}(\mathbf{A}, \theta^*, \beta^*)(\omega_k)$. Then there exists proper $(\mathbf{W}_k, \theta_k, \beta_k) \in \mathcal{S}(\sigma, \mathbf{A})$ such that $\text{dist}_{\mathcal{H}_k}^2(\omega_k, \Omega^*)$ converges to zero linearly; namely,

$$\text{dist}_{\mathcal{H}_{k+1}}^2(\omega_{k+1}, \Omega^*) < \gamma \text{dist}_{\mathcal{H}_k}^2(\omega_k, \Omega^*), \quad (19)$$

where γ is some positive constant smaller than 1.

From Theorem 3, we know that if the Error Bound Condition (EBC) in (18) is satisfied, then there exists a sequence $\{(\mathbf{W}_k, \theta_k, \beta_k)\}_{k=0}^K \rightarrow (\mathbf{A}, \theta^*, \beta^*)$ that enables the linear decay rate of $\text{dist}_{\mathcal{H}_k}^2(\omega_k, \Omega^*)$. Actually, our learning based D-LADMM does obtain a faster convergence speed than fixing all parameters as $(\mathbf{A}, \theta^*, \beta^*)$. To confirm this, we give a specific example in the following lemma.

Lemma 4.4. *Consider the case where the prox operator of the function $f(\cdot)$ is bijective. For any $\omega \notin \Omega^*$ and $\omega^* \in \Omega^*$, there exists a residue $(\Delta_w, \Delta_\theta, \Delta_\beta)$ such that $\|\mathcal{T}(\mathbf{A} + \Delta_w, \theta^* + \Delta_\theta, \beta^* + \Delta_\beta)(\omega) - \omega^*\|_F < \|\mathcal{T}(\mathbf{A}, \theta^*, \beta^*)(\omega) - \omega^*\|_F$.*

Lemma 4.4 implies that, at each iteration, we can find appropriate parameter to construct a solution that is closer to Ω^* than the solution produced by fixed parameters. Hence, it is entirely possible for the proposed D-LADMM to achieve a convergence rate higher than the traditional LADMM. The first experiment in Section 5 verifies the superiority of D-LADMM over LADMM, in terms of convergence speed.

It is worth noting that, although the convergence speed of LADMM can be improved by setting suitable penalty parameter β for each dimension, it is really difficult to do this in practice. In contrast, the proposed D-LADMM provides a convenient way to learn all parameters adaptively from data so as to gain high convergence speed, as will be confirmed in the experiments.

4.4. Discussions on D-LADMM

First, our analysis for the convergence of D-LADMM is very general and can actually include the traditional LADMM as a special case. In effect, some already known properties

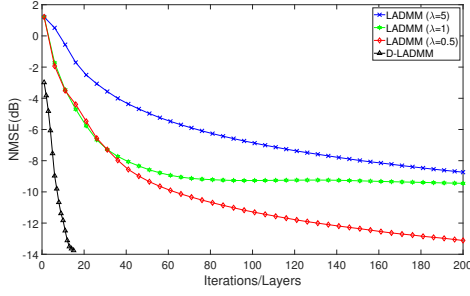


Figure 2. NMSE comparison among D-LADMM and LADMM with different λ on the simulation dataset.

of LADMM can be deduced from our results, but not vice versa. The analysis techniques designed for traditional optimization cannot be directly applied to our case, and it is considerably more challenging to establish the convergence properties of D-LADMM.

Second, our EBC is indeed weaker than the conditions assumed by the previous studies (e.g., (Han & Yuan, 2013; Han et al., 2015; Yang & Han, 2016)), which prove linear convergence rate for ADMM or LADMM. More precisely, as pointed out by Liu et al. (2018b), all the EBCs in (Han & Yuan, 2013; Han et al., 2015; Yang & Han, 2016) can be equivalently expressed as

$$\text{dist}(\omega, \Omega^*) \leq \kappa \cdot \|\phi(\omega)\|_F, \forall \omega, \text{dist}(\omega, \Omega^*) \leq \epsilon, \quad (20)$$

where $\phi(\cdot)$ is a mapping given in Table 1. Notably, the following lemma confirms the generality of our EBC.

Lemma 4.5. *The EBC in (20) suffices to ensure the validity of our EBC given in (18).*

One may have noticed that our EBC is somewhat similar to the condition by Liu et al. (2018b). Yet the work of (Liu et al., 2018b) did not reveal the merit of learning-based optimization; that is, as shown in Lemma 4.2, all the parameters are learnt automatically from data with the purpose of accelerating the convergence speed.

5. Experiments

Different from LADMM which contains no learnable parameter, the proposed D-LADMM is treated as a structured neural network and trained using stochastic gradient descent (SGD) over the observation \mathbf{X} . All the parameters, denoted as Θ , are subject to learning. For convenience, we mainly consider the following ℓ_1 -norm constrained problem for empirical validations:

$$\min_{\mathbf{Z}, \mathbf{E}} \lambda \|\mathbf{Z}\|_1 + \|\mathbf{E}\|_1, \quad \text{s.t. } \mathbf{X} = \mathbf{AZ} + \mathbf{E}, \quad (21)$$

where λ is a parameter to balance the contribution of each term. We use both LADMM and the proposed D-LADMM to solve the above problem, and compare their results on both synthetic datasets and natural images¹. Our D-LADMM is implemented on the PyTorch platform.

¹Code: <https://github.com/zzs1994/D-LADMM>

5.1. Simulation Experiments

We first experiment with synthetic data, using similar experimental settings as (Chen et al., 2018). Specifically, we set $m = 500$ and $d = 250$. The numbers of training and testing samples are set to 10,000 and 1,000, respectively. Elements of the dictionary matrix \mathbf{A} are sampled from i.i.d. Gaussian, namely $A_{i,j} \sim N(0, 1/d)$. The columns of \mathbf{A} are normalized have a unit ℓ_2 norm. To make a fair comparison, \mathbf{A} is fixed and shared by all considered methods. The sparse coefficient matrix \mathbf{Z} is generated by using a Bernoulli sampling operator (with probability 0.1) to randomly select values from the standard Gaussian distribution, i.e., $\mathbf{Z} = \text{Ber}(0.1) \circ N(0, 1)$. The sparse matrix \mathbf{E} is generated in the same way as \mathbf{Z} , and the data samples for training and testing are constructed as $\mathbf{X} = \mathbf{AZ} + \mathbf{E}$.

For the proposed D-LADMM, the number of layers is set to $K = 15$. SGD is adopted to update the parameters with learning rate $lr = 0.01$. Regarding the activation function, we use the softshrink operator by Beck & Teboulle (2009). In these experiments, the ground-truth \mathbf{Z}^* and \mathbf{E}^* of training samples are known, thereby the second strategy in (9) is adopted to train our D-LADMM network. The results are evaluated by a measure of NMSE (normalized mean square error in dB), defined in terms of both \mathbf{Z} and \mathbf{E} :

$$\text{NMSE} = 10 \log_{10} \left(\frac{\|\mathbf{Z}_K - \mathbf{Z}^*\|_F^2}{\|\mathbf{Z}^*\|_F^2} + \frac{\|\mathbf{E}_K - \mathbf{E}^*\|_F^2}{\|\mathbf{E}^*\|_F^2} \right). \quad (22)$$

In Figure 2, we compare the proposed D-LADMM with LADMM. As one can see, the NMSE achieved by D-LADMM decreases linearly as the layer number k grows and is much smaller than that by LADMM. These results confirm our main results stated in Theorems 1, 2 and 3. Note here that, unlike D-LADMM, LADMM needs a proper λ to produce correct solutions, thereby we test several different λ 's for LADMM. One can see that the choice of λ has dramatic influences on the performance of LADMM, and smaller λ may lead to better results.

5.2. Natural Image Denoising

We also evaluate the considered methods on the task of natural image denoising, which is to remove the noise term \mathbf{E} from the noisy observation \mathbf{X} , or recover the noise-free image \mathbf{AZ} from \mathbf{X} as equal. The experimental data is a classic dataset consisting of 12 natural images, called Waterloo BragZone Greyscale set2, in which a fraction of $r\%$ Salt-and-pepper noise is added to each image. Furthermore, the rectangle of each image is divided into non-overlapping patches of size 16×16 . We use the patch-dictionary method (Xu & Yin, 2014) to learn a 256×512 dictionary \mathbf{A} and use it to initialize our D-LADMM. The network and parameter settings are the same as in the simulation experiments. Since the ground truth is unknown, we

Table 2. PSNR comparison on 12 images with noise rate 10%. For LADMM, we examine its performance at a couple of different iterations. LADMM is comparable to D-LADMM only when it undergoes a large number of iterations.

PSNR	Images											
	Barb	Boat	France	Frog	Goldhill	Lena	Library	Mandrill	Mountain	Peppers	Washesat	Zelda
Baseline	15.4	15.3	14.5	15.6	15.4	15.4	14.2	15.6	14.4	15.1	15.1	15.2
LADMM (iter=15)	22.1	24.2	18.0	23.1	25.2	25.6	15.0	21.7	17.7	25.1	30.6	29.7
LADMM (iter=150)	27.9	29.8	21.6	26.5	30.4	31.3	17.8	24.3	20.5	30.0	34.5	35.7
LADMM (iter=1500)	29.9	31.1	22.2	26.9	31.8	33.2	18.0	25.1	20.7	32.8	36.2	37.8
D-LADMM ($K=15$)	29.5	31.3	21.9	25.9	32.5	35.1	18.8	24.5	19.3	34.3	35.6	38.9

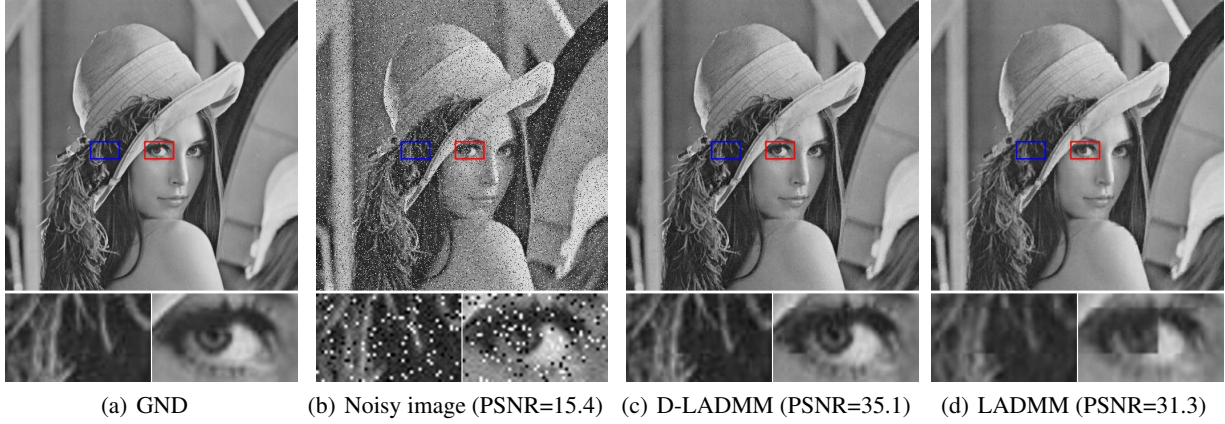


Figure 3. Denoising results the Lena image. (a) The ground-truth images. (b) The noisy images with salt-and-pepper noise of rate 10%. (c) Denoised images by our D-LADMM. (d) Denoised images by LADMM with 150 iterations. Best view on screen!

use the objective (8) as the loss function for training. The parameter in problem (21) is set as $\lambda = 0.5$, and Peak signal-to-noise ratio (PSNR) is used to evaluate the performance of various methods.

Table 2 shows the comparison results at noise rate 10%. For LADMM, which usually needs a large of iterations to converge, we examine its performance at 15-th, 150-th and 1,500-th iterations—note that one iteration of LADMM corresponds to one block/layer of our D-LADMM. As can be seen, our 15-layers D-LADMM achieves much higher PSNR than the LADMM at 15-th iteration, and LADMM requires 1,500 iterations to obtain results comparable to our D-LADMM. In other words, compared with LADMM, D-LADMM achieves almost the same good solution within two order-of-magnitude fewer iterations. In Figure 3, we compare the visual quality of the denoised images by D-LADMM and LADMM, using the Lena image as the experimental data. It can be seen that the quality of the images recovered by D-LADMM is visibly higher than that of LADMM, especially for the areas full of rich details. All the above results demonstrate the superiorities of our proposed D-LADMM over LADMM.

5.3. Complexity Comparison

Suppose that $\mathbf{A} \in \mathbb{R}^{m \times d_1}$, $\mathbf{B} \in \mathbb{R}^{m \times d_2}$ and $\mathbf{X} \in \mathbb{R}^{m \times n}$, then the complexity of training D-LADMM is $\mathcal{O}((d_1 + d_2)m n K p)$, where K is the number of layers. Whereas

the complexity of LADMM is $\mathcal{O}((d_1 + d_2)m n t)$, where t is the number of iterations. Usually, t is one or two order-of-magnitude greater than K . Notice, that D-LADMM achieves comparable performance with LADMM only when $K p \ll t$, e.g., $t = 1,500$ and $K p = 225$ as shown in Table 2. In experiments, to both achieve an NMSE of -13dB with $n = 10,000$ and $20,000$, D-LADMM needs 5 and 9 minutes (including training time), while LADMM needs 12 and 22 minutes, respectively. Therefore, even considering the training computational load, D-LADMM is still much faster than LADMM which needs a large iteration number. In addition, it is worth noting that, when a new data point comes, D-LADMM only needs a very low computational load for forward propagation.

6. Conclusion

In this paper, inspired by LADMM, we propose D-LADMM, a deep-learning-based optimization method for solving the constrained problem (1). Specifically, we first convert the proximal operator in LADMM to a special NN structure and replace the given matrix \mathbf{A} in the proximal operator by learnable weights. Furthermore, we generalize the scalar to some elements-wise operations. From the theoretical perspective, we prove that, under some mild technical conditions, D-LADMM can obtain the linear convergence and converge faster than original LADMM. Experiments on simulative and real applications verify the superiority of D-LADMM.

Acknowledgments

The work of Guangcan Liu is supported in part by NSF of China under Grant 61622305 and Grant 61502238, in part by Natural Science Foundation of Jiangsu Province of China under Grant BK20160040, in part by SenseTime Research Fund. The work of Zhouchen Lin is supported in part by 973 Program of China under Grant 2015CB352502, in part by NSF of China under Grant 61625301 and Grant 61731018, in part by Qualcomm and Microsoft Research Asia.

References

- Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- Blumensath, T. and Davies, M. E. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.
- Bot, R. I. and Nguyen, D.-K. The proximal alternating direction method of multipliers in the nonconvex setting: convergence analysis and rates. *arXiv preprint arXiv:1801.01994*, 2018.
- Chen, X., Liu, J., Wang, Z., and Yin, W. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *arXiv preprint arXiv:1808.10038*, 2018.
- Chen, Y. and Pock, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.
- Diamond, S., Sitzmann, V., Heide, F., and Wetzstein, G. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.
- Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *Proceedings of the International Conference on Machine Learning*, pp. 399–406, 2010.
- Han, D. and Yuan, X. Local linear convergence of the alternating direction method of multipliers for quadratic programs. *SIAM Journal on Numerical Analysis*, 51(6):3446–3457, 2013.
- Han, D., Sun, D., and Zhang, L. Linear rate convergence of the alternating direction method of multipliers for convex composite quadratic and semi-definite programming. *arXiv preprint arXiv:1508.02134*, 2015.
- He, B. and Yuan, X. On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, 2017.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Li, H., Yang, Y., Chen, D., and Lin, Z. Optimization algorithm inspired deep neural network structure design. *arXiv preprint arXiv:1810.01638*, 2018.
- Li, J., Fang, C., and Lin, Z. Lifted proximal operator machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- Lin, Z., Liu, R., and Su, Z. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 612–620, 2011.
- Liu, G. and Li, P. Low-rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing*, 64(21):5623–5633, 2016.
- Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.
- Liu, G., Chang, S., and Ma, Y. Blind image deblurring using spectral properties of convolution operators. *IEEE Transactions on Image Processing*, 23(12):5047–5056, 2014.
- Liu, G., Liu, Q., and Li, P. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):47–60, 2017.
- Liu, J., Chen, X., Wang, Z., and Yin, W. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations*, 2019.
- Liu, Q., Liu, G., Li, L., Yuan, X.-T., Wang, M., and Liu, W. Reversed spectral hashing. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2441–2449, 2018a.
- Liu, R., Zhong, G., Cao, J., Lin, Z., Shan, S., and Luo, Z. Learning to diffuse: A new perspective to design pdes for visual analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2457–2471, 2016.

- Liu, Y., Yuan, X., Zeng, S., and Zhang, J. Partial error bound conditions and the linear convergence rate of the alternating direction method of multipliers. *SIAM Journal on Numerical Analysis*, 56(4):2095–2123, 2018b.
- Peng, X., Zhou, J. T., and Zhu, H. K-meansNet: When k-means meets differentiable programming. *arXiv preprint arXiv:1808.07292*, 2018.
- Sprechmann, P., Bronstein, A. M., and Sapiro, G. Learning efficient sparse and low rank models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1821–1833, 2015.
- Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.
- Wang, Z., Liu, D., Chang, S., Ling, Q., Yang, Y., and Huang, T. S. D3: Deep dual-domain based fast restoration of jpeg-compressed images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2764–2772, 2016.
- Xu, Y. and Yin, W. A fast patch-dictionary method for whole image recovery. *arXiv preprint arXiv:1408.3740*, 2014.
- Yang, W. H. and Han, D. Linear convergence of the alternating direction method of multipliers for a class of convex optimization problems. *SIAM Journal on Numerical Analysis*, 54(2):625–640, 2016.
- Yang, Y., Sun, J., Li, H., and Xu, Z. Deep ADMM-Net for compressive sensing MRI. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 10–18, 2016.
- You, C., Robinson, D., and Vidal, R. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3918–3927, 2016.
- Zeiler, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- Zhang, H., Lin, Z., Zhang, C., and Chang, E. Y. Exact recoverability of robust PCA via outlier pursuit with tight recovery bounds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3143–3149, 2015.
- Zhang, K., Zuo, W., Gu, S., and Zhang, L. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.
- Zhang, X., Burger, M., Bresson, X., and Osher, S. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3): 253–276, 2010.
- Zhang, X., Wang, L., Yu, Y., and Gu, Q. A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *Proceedings of the International Conference on Machine Learning*, pp. 5857–5866, 2018.
- Zhou, J. T., Di, K., Du, J., Peng, X., Yang, H., Pan, S. J., Tsang, I. W., Liu, Y., Qin, Z., and Goh, R. S. M. SC2Net: Sparse LSTMs for sparse coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.