Convergence of Slice-Based Block Coordinate Descent Algorithm for Convolutional Sparse Coding

Jing Li,1 Hui Yu,1 Xiao Wei,1 and Jinjia Wang2,3

1School of Science, Yanshan University, Qinhuangdao 066004, China
2School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
3Hebei Key Laboratory of Information Transmission and Signal Processing, Yanshan University, Qinhuangdao, Hebei 066004, China

Correspondence should be addressed to Jinjia Wang; wjj@ysu.edu.cn

Received 4 September 2019; Revised 1 July 2020; Accepted 13 July 2020; Published 30 July 2020

ACADEMIC EDITOR: Javier Martinez Torres

Copyright © 2020 Jing Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Convolutinal sparse coding (CSC) models are becoming increasingly popular in the signal and image processing communities in recent years. Several research studies have addressed the basis pursuit (BP) problem of the CSC model, including the recently proposed local block coordinate descent (LoBCoD) algorithm. This algorithm adopts slice-based local processing ideas and splits the global sparse vector into local vector needles that are locally computed in the original domain to obtain the encoding. However, a convergence theorem for the LoBCoD algorithm has not been given previously. This paper presents a convergence theorem for the LoBCoD algorithm which proves that the LoBCoD algorithm will converge to its global optimum at a rate of \(O(1/k)\). A slice-based multilayer local block coordinate descent (ML-LoBCoD) algorithm is proposed which is motivated by the multilayer basis pursuit (ML-BP) problem and the LoBCoD algorithm. We prove that the ML-LoBCoD algorithm is guaranteed to converge to the optimal solution at a rate \(O(1/k)\). Preliminary numerical experiments demonstrate the better performance of the proposed ML-LoBCoD algorithm compared to the LoBCoD algorithm for the BP problem, and the loss function value is also lower for ML-LoBCoD than LoBCoD.

1. Introduction

Sparse representation models have been widely used in various image processing [1, 2] and computer vision [3, 4] applications. A sparse representation model assumes that signals \(X \in \mathbb{R}^N\) can be expressed as a linear combination of several columns, i.e., \(X = D\Gamma\), where \(D\) is a matrix that forms the dictionary \(D \in \mathbb{R}^{N \times N_m}\), and \(\Gamma \in \mathbb{R}^{N_m}\) is a sparse vector. If \(D\) is assumed to be fixed, it can be considered a basis pursuit (BP) problem to find the sparse vector \(\Gamma\). However, the BP algorithms are only encoded in patches and ignore the relationship between neighboring patches, resulting in a high degree of redundancy in the encoding. The convolution sparse coding (CSC) model [5] has been proposed and extended in the last ten years, and it imposes constraints on the dictionary by using a banded circulant matrix. This model assumes that the signal can be represented as the superposition of a few local filters, convolved with a sparse vector. Several works have presented algorithms for solving the CSC problem [6, 7]. Contemporary BP algorithms for CSC often rely on the Alternating Direction Methods of Multipliers (ADMM) algorithm in the Fourier domain. It is known that algorithms encoded in the Fourier domain are often computationally infeasible. Additionally, algorithms based on the ADMM formula need to introduce auxiliary variables which increases the difficulty of optimization. A recent work proposed by Papyan et al. [8], adopted slice-based local processing ideas and split the global sparse vector \(\Gamma\) into local vector needles that are locally computed in the original domain rather than the Fourier domain to obtain the encoding. While this approach still relies on the ADMM algorithm, its convergence largely depends on the auxiliary variables that were introduced. The LoBCoD algorithm [9] is another algorithm that was proposed for the BP problem.
The advantages of the LoBCoD algorithm are that it is not calculated on the Fourier domain and the calculation does not use the ADMM formula. More precisely, the LoBCoD algorithm optimizes needles of the CSC model in the original domain and operates without any auxiliary variables. Compared with global or local ADMM-based methods, the LoBCoD algorithm achieves better performance to solve the BP problem. However, the literature [9] does not provide convergence theorem for the LoBCoD algorithm. Thus, this paper will present a convergence theorem and proof of the LoBCoD algorithm.

A multilayer convolution sparse coding (ML-CSC) model has been proposed in the last three years by Sulam et al. [10], which is a deep extension of the CSC model. The core assumption of the ML-CSC model is that a signal can be expressed by sparse representations at different layers in terms of nested convolutional filters. The traditional BP problem was recently extended to a multilayer setting, which was motivated by the ML-CSC model [11]. Several methods have been proposed to solve the ML-BP problem. The first method is a layered basis pursuit algorithm [12], which establishes a connection between convolutional neural networks and sparse modeling. However, layered basis pursuit algorithm does not provide a signal that satisfies the assumption of the multilayer model, and the signal reconstruction error increases as the network deepens. Subsequently, the multilayer iterative threshold algorithm (ML-ISTA) and its fast version (ML-FISTA) algorithm [11] were proposed, which only require matrix multiplications and entry-wise operations and will converge well to the global optimal. Unfortunately, both methods operate on patches only and do not utilize slice-based local processing idea. Therefore, the slice-based ML-LoBCoD algorithm is proposed for ML-BP problem. This algorithm employs slice-based local processing idea and the block coordinate descent (BCD) method. Based on the convergence theorem proof of the block coordinate descent algorithm [13], this paper provides a convergence theorem for the ML-LoBCoD algorithm and proves that the ML-LoBCoD algorithm converges to the global optimal value at a rate of $O(1/k)$.

The rest of this paper is organized as follows. We begin by reviewing the slice-based CSC and slice-based LoBCoD algorithms in Section 2. The convergence theorem and proof of the LoBCoD algorithm are given in Section 3. In Section 4, we propose a slice-based ML-CSC model and a slice-based ML-LoBCoD algorithm. The convergence theorem and proof of the slice-based ML-LoBCoD algorithm are given in Section 5. In Section 6, the experimental results of the signal reconstruction and classification accuracy of the two networks inspired by the two algorithms are given. Finally, we conclude this work in Section 7.

2. Background

2.1. Slice-Based Convolutional Sparse Coding. The CSC model assumes that a global signal $X \in \mathbb{R}^N$ can be decomposed as $X = D^T \ast \Gamma$, where $D \in \mathbb{R}^{N \times N_m}$ is a banded convolutional dictionary that consists of all shifted versions of a local dictionary $D_L \in \mathbb{R}^{m \times m}$, $d_i \in \mathbb{R}^m$ are local filters that are extracted from $D_L$, the global sparse vector $\Gamma \in \mathbb{R}^{N \times m}$ contains the interlacing cascades of all the sparse representations $\Gamma_i \in \mathbb{R}^m$, and $\Gamma^0$ is the corresponding sparse representation of local filter $d_i$. Using the above formula, the BP problem can be expressed as

$$\min \frac{1}{2} \| X - D^T \|_2^2 + \lambda \| \Gamma \|_1.$$  

(1)

The global sparse vector $\Gamma$ can be decomposed into $N$ nonoverlapping $m$-dimensional local sparse vectors, $\{\alpha_i\}_{i=1}^N$, which are called needles [8], i.e., $\Gamma = [\alpha_1, \ldots, \alpha_s, \ldots, \alpha_N]$. Thus, the global signal $X$ can be expressed as $X = \sum_{i=1}^N P_i^T D_L \alpha_i$, where $P_i^T \in \mathbb{R}^{N \times m}$ is the operator that places $D_L \alpha_i$ in the $i$th position and pads the remaining entries with zeros. Therefore, the BP problem (1) can be expressed as a local problem:

$$\min \frac{1}{2} \left\| X - \sum_{i=1}^N P_i^T D_L \alpha_i \right\|_2^2 + \lambda \left\| \alpha_i \right\|_1.$$  

(2)

Papyan et al. proposed the slice-based local processing idea and defined $s_i = D_L \alpha_i$ as the $i$th slice. The global signal $X$ can be rewritten as $X = \sum_{i=1}^N P_i^T s_i$. Then, the slice-based BP problem (2) can be expressed as

$$\min \left\{ \alpha_i \right\}_{i=1}^N \frac{1}{2} \left\| X - \sum_{i=1}^N P_i^T s_i \right\|_2^2 + \lambda \sum_{i=1}^N \left\| \alpha_i \right\|_1,$$  

s.t.  

$$s_i = D_L \alpha_i.$$  

(3)

Papyan et al. tackled the BP problem (3) using the ADMM algorithm [8], which minimizes the following augmented Lagrangian problem:

$$\min_{D_L, \left\{ \alpha_i \right\}_{i=1}^N, \left\{ s_i \right\}_{i=1}^N} \frac{1}{2} \left\| X - \sum_{i=1}^N P_i^T s_i \right\|_2^2 + \sum_{i=1}^N \left( \lambda \left\| \alpha_i \right\|_1 + \frac{\rho}{2} \left\| s_i - D_L \alpha_i + u_i \right\|_2^2 \right).$$  

(4)

Here, $\left\{ u_i \right\}_{i=1}^N$ denotes the dual variables of the ADMM formulation.

2.2. Slice-Based Local Block Coordinate Descent Algorithm. The CSC model parameters are represented by the local sparse vectors $\alpha_i$ and the local dictionary $D_L$. Assuming $D_L$ is fixed, a slice-based local processing idea and block coordinate descent method are adopted to update the needles, $f(\alpha^k_1, \ldots, \alpha^k_s, \alpha^k_{s+1}, \ldots, \alpha^k_N)$ is objection function of equation (2). The BCD algorithm [14] will be briefly described below.

Initializtion: choose any $\Gamma^0 = (\alpha^0_1, \ldots, \alpha^0_N)$, $k = 0$

Iteration: choose an index $s \in \{1, \ldots, N\}$ and compute

$$r_s^{k+1} \in \arg \min_{\alpha_s} f(\alpha^k_1, \ldots, \alpha^k_{s-1}, \alpha^k_s, \alpha^k_{s+1}, \ldots, \alpha^k_N),$$

$$\alpha^k_s = \alpha^k_s, \quad \forall j \neq s,$$

$$k = k + 1,$$
until the convergence condition is met.

Output: \( \forall^{k+1} = (a_1^{k+1}, \ldots, a_N^{k+1}) \)

In this paper, each needle \( a_i \) can be treated as a block of coordinates taken from the global vector which can be optimized separately with respect to each block in sequence.

Consequently, the update rule for each needle can be written as

\[
\min_{a_i} \left\| X - \sum_{j=1}^{N} P_j^T D_j \alpha_j - P_i^T D_i \alpha_i \right\|_2^2 + \lambda \| \alpha_i \|_1, \quad i = 1, \ldots, N. \tag{6}
\]

Equation (6) can be decomposed into a local problem:

\[
\min_{a_i} \frac{1}{2} \| P_i R_i - D_i \alpha_i \|_2^2 + \lambda \| \alpha_i \|_1, \quad i = 1, \ldots, N, \tag{7}
\]

where \( R_i = X - \sum_{j \neq i}^{N} P_j^T D_j \alpha_j \) is global variable, representing the residual image without the contribution of needle \( a_i \), and \( P_i \in \mathbb{R}^{n \times N} \) is the transpose of \( P_i \), representing the operator that extracts the \( i \)th \( n \)-dimensional patch from \( X \).

The LoBCoD algorithm is proposed to minimize equation (7) [9]. The function \( f(a_i) = (1/2)\|P_i R_i - D_i \alpha_i\|_2^2 \) can be defined, which is a convex smooth function. The LoBCoD algorithm can be considered to be a generalized gradient algorithm that applies an update in the form \( a_i^{k+1} \leftarrow a_i^k - \nabla f(a_i^k) \), where \( \nabla f(a_i) = D_i^T (D_i \alpha_i - P_i R_i) \). The update rule for each needle \( a_i \) can be expressed as

\[
a_i^{k+1} = S_i(a_i^k + D_i^T (P_i R_i - D_i a_i^k)). \tag{8}
\]

3. Convergence of Slice-Based Local Block Coordinate Descent Algorithm

The convergence theorem will now be proposed and proof of LoBCoD algorithm will be given.

**Lemma 1** (fundamental proximal gradient inequality).

Assume that \( f \) is a convex smooth function, \( g \) is a convex function, and \( T_L \) is proximal gradient operator. For any \( x \in \mathbb{R}^n, y \in \text{int}(\text{dom}(f)), L > 0 \), satisfying

\[
f(T_L(y)) \leq f(y) + \langle \nabla f(y), T_L(y) - y \rangle + \frac{L}{2} \| T_L(y) - y \|^2,
\]

it holds that

\[
F(x) - F(T_L(y)) \geq \frac{L}{2} \| x - T_L(y) \|^2 - \frac{L}{2} \| x - y \|^2 + I_f(x,y),
\]

where

\[
I_f(x,y) = f(x) - f(y) - \langle \nabla f(y),x-y \rangle. \tag{11}
\]

**Theorem 1** (convergence of LoBCoD).

Given a signal \( X \) and local dictionary \( D_L \), the slice-based LoBCoD algorithm is guaranteed to converge to the optimal solution at a rate \( O(1/k) \).

**Proof.** The optimization problem of CSC can be represented as a general minimization model as follows:

\[
\min_{[a_i]} \{ f(a_i) + g(a_i) : a_i \in \mathbb{R}^n, \} \tag{12}
\]

where \( f(a_i) = (1/2)\|P_i R_i - D_i a_i\|_2^2 \) and \( g(a_i) = \lambda \| a_i \|_1 \) are convex functions. The gradient of \( f(a_i) \) is Lipschitz continuous. \( \{a_i^*\} \) is defined to represent the nonempty optimal problem set of problem (12), and the optimal objective function value is represented by \( F(a_i^*) \). According to the proximal gradient method, the general update step of \( a_i \) can be written in the following form:

\[
a_i^{k+1} = \text{prox}_{(1/L)f}(a_i^k + \frac{1}{L}D_i^T (P_i R_i - D_i a_i^k)). \tag{13}
\]

Due to the proximal gradient operator \( T_{L,f}^\vartheta(a_i) = \text{prox}_{(1/L)f}(a_i - (1/L)\nabla f(a_i)) \), \( L \) is the Lipschitz constant of \( f(a_i) \). Therefore, the update step of each needle \( a_i \) can be written in the following form:

\[
a_i^{k+1} = T_{L,f}^\vartheta(a_i^k). \tag{14}
\]

Exploiting the fundamental proximal gradient inequality and making \( x = a_i^*, y = a_i^0 \), we obtain

\[
\frac{2}{L} \bigg( F(a_i^*) - F(T_{L,f}^{\vartheta} (a_i^0)) \bigg) \geq \| a_i^* - a_i^{n+1} \|^2 - \| a_i^* - a_i^n \|^2
\]

\[
+ \frac{2}{L} I_f(a_i^*, a_i^n)
\]

\[
\geq \| a_i^* - a_i^{n+1} \|^2 - \| a_i^* - a_i^n \|^2.
\]

Thus, we obtain

\[
\sum_{n=0}^{k-1} (F(a_i^{n+1}) - F(a_i^n)) \geq \sum_{n=0}^{k-1} \left( \| a_i^* - a_i^{n+1} \|^2 - \| a_i^* - a_i^n \|^2 \right). \tag{16}
\]

The following result is obtained using the scaling method:

\[
\sum_{n=0}^{k-1} (F(a_i^{n+1}) - F(a_i^n)) \leq \frac{L}{2} \| a_i^* - a_i^n \|^2 - \frac{L}{2} \| a_i^* - a_i^{n-1} \|^2
\]

\[
\leq \frac{L}{2} \| a_i^* - a_i^0 \|^2.
\]

Thus, we obtain
\[ k(F(a^k_i) - F(a^*_i)) \leq \sum_{n=0}^{k-1} (F(a^{n+1}) - F(a^*_i)) \leq \frac{L}{2}\|a^*_i - a^0_i\|^2. \]  

Finally, we obtain
\[ F(a^k_i) - F(a^*_i) \leq \frac{L}{2\sqrt{k}}\|a^*_i - a^0_i\|^2. \]  

Since the sparse vector \( \Gamma \) can be decomposed into nonoverlapping \( m \)-dimensional sparse vectors \( \{a_i\}_{i=1}^N \) and \( a_i \) is convergent, we can obtain convergence of \( \Gamma \) and the convergence rate is constant with \( \alpha \). Thus, the LoBCoD algorithm converges to the global optimum at a rate of \( O(1/k) \).

4. Slice-Based Multilayer Local Block Coordinate Descent Algorithm

4.1. Slice-Based Multilayer Convolutional Sparse Coding

The ML-CSC model is a deep extension of the CSC model, since the ML-CSC model assumes that the signal can be represented by a sparse representation of the nested convolution filter on different layers. The ML-CSC model assumes that for \( D_j \in R^{N \times N} \), \( \Gamma_j \) is the corresponding sparse representation and a global signal \( X \) can be expressed as \( X = D_j\Gamma_j \). This model can be cascaded by imposing a similar assumption \( \Gamma_j \), i.e., \( \Gamma_j = D_j\Gamma_j \), for a convolutional dictionary \( D_j \in R^{N \times N} \) and corresponding sparse representations \( \Gamma_j \). The model can be cascaded to \( J \) layers and the final global signal \( X \) can be expressed as \( X = D_J\Gamma_J \), where \( D_{\{1,J\}} = D_1 + D_2 + \ldots + D_J \). Applying the slice-based local processing idea to the ML-CSC model, the slice-based ML-CSC model is proposed. The definition of the slice-based ML-CSC model will now be given. For a set of local convolution dictionaries \( \{D_{L,i}\}_{i=1}^J \) of appropriate dimensions and a global signal \( X \in R^N \), the slice-based ML-CSC model can be expressed as

\[ X = D_{L,1}\Gamma_1 = \sum_{i=1}^{N} p_{i,m}^T D_{L,1}\alpha_{i,m}, \quad \| \sum_{i=1}^{N} \alpha_{i,m} \|_{0,\infty} \leq \lambda_1, \]
\[ \Gamma_1 = D_{L,2}\Gamma_2 = \sum_{i=1}^{N} p_{i,m}^T D_{L,2}\alpha_{i,m}, \quad \| \sum_{i=1}^{N} \alpha_{i,m} \|_{0,\infty} \leq \lambda_2, \]
\[ \vdots \]
\[ \Gamma_{J-1} = D_{L,J}\Gamma_J = \sum_{i=1}^{N} p_{i,m}^T D_{L,J}\alpha_{i,m}, \quad \| \sum_{i=1}^{N} \alpha_{i,m} \|_{0,\infty} \leq \lambda_J, \]

where the \( l_{0,\infty} \) norm is defined the maximal number of nonzeros in vector \( [10] \), \( \Gamma_j \) is the sparse representation of the \( j \)th layer, \( p_{i,m}^T \) is the operator that extracts the \( n \)-dimensional patch from the \( n \)-dimensional sparse representation \( \Gamma_j \), \( D_{L,j} \) is \( j \)th layer local dictionary, \( \alpha_{i,m} \) is \( \alpha \) of the \( j \)th layer, and \( \lambda_j \) is a superparameter. The proposed slice-based multilayer basis pursuit (ML-BP) problem can be expressed as

\[ \min_{\{\alpha_{i,m}\}_{i=1}^{N}} \frac{1}{2} \|X - \sum_{i=1}^{N} p_{i,m}^T D_{L,1}\alpha_{i,m}\|_2^2 + \sum_{j=1}^{J} \lambda_j \| \alpha_{i,m} \|_{1,\infty}. \]  

The \( j \)th layer of the BP problem can be expressed as

\[ \min_{\{\alpha_{i,m}\}_{i=1}^{N}} \frac{1}{2} \|\Gamma_{j-1} - \sum_{m=1}^{N} p_{i,m}^T D_{L,J}\alpha_{i,m}\|_2^2 + \lambda \| \alpha_{i,m} \|_{1,\infty}, \quad i = 1, \ldots, N. \]

4.2. Slice-Based Multilayer Local Block Coordinate Descent Algorithm

Based on the LoBCoD algorithm, this paper extends the LoBCoD algorithm to a multilayer algorithm to solve the ML-BP problem of equation (21). ML-LoBCoD uses slice-based local processing idea to update the needles. However, rather than optimizing with respect to all needles at the same time, we can treat each needle \( \alpha_{i,m} \) as a coordinate block and optimize with respect to each block separately in sequence. Therefore, the \( j \)th layer of the BP problem can be expressed as

\[ \min_{\{\alpha_{i,m}\}_{i=1}^{N}} \frac{1}{2} \|\Gamma_{j-1} - \sum_{m=1}^{N} p_{i,m}^T D_{L,J}\alpha_{i,m}\|_2^2 + \lambda \| \alpha_{i,m} \|_{1,\infty}, \quad i = 1, \ldots, N. \]

(23)

By defining \( R_{j,i} = \Gamma_{j-1} - \sum_{m=1}^{N} p_{i,m}^T D_{L,J}\alpha_{i,m} \) as the residual image of the \( j \)th layer without the contribution of the needles \( \alpha_{i,m} \), the \( j \)th layer of the ML-BP problem of equation (23) can be rewritten as

\[ \min_{\{\alpha_{i,m}\}_{i=1}^{N}} \frac{1}{2} \|R_{j,i} - p_{i,m}^T D_{L,J}\alpha_{i,m}\|_2^2 + \lambda \| \alpha_{i,m} \|_{1,\infty}, \quad i = 1, \ldots, N. \]

(24)

The \( j \)th layer of the ML-BP problem of equation (24) becomes equivalent to solving the following minimization problem:

\[ \min_{\{\alpha_{i,m}\}_{i=1}^{N}} \frac{1}{2} \|p_{i,m} R_{j,i} - D_{L,J}\alpha_{i,m}\|_2^2 + \lambda \| \alpha_{i,m} \|_{1,\infty}, \quad i = 1, \ldots, N. \]

(25)

We define \( f(a) = (1/2)\|p_{i,m} R_{j,i} - D_{L,J}\alpha_{i,m}\|_2^2 \), which is a convex function. The gradient of \( f(a) \) is \( \nabla f(a) = -D_{L,J}^T (p_{i,m} R_{j,i} - D_{L,J}\alpha_{i,m}) \). The ML-LoBCoD algorithm can also be considered to be a generalized gradient algorithm that applies an update of the form...
\[ a_{j}^{k+1} = a_{j}^{k} - \nabla f(a_{j}^{k}). \] The update rule of each needle \( a_{j} \) can be expressed as

\[ a_{j}^{k+1} = S(a_{j}^{k} + D_{L,j}^{T}(P_{j}R_{j} - D_{L,j}a_{j}^{k})). \] (26)

The process of the ML-LoBCoD algorithm is shown as Algorithm 1. The input is the global signal \( X \), the convolution dictionary \( \{D_{L,j}\}_{j=1}^{N} \), and the initial needles \( \{a_{j}^{0}\}_{j=1}^{N} \). The initial parameter values are \( \Gamma_{0} = X \) and \( R_{0} = \Gamma_{0} \). The outputs are the needles \( \{a_{j}^{N}\}_{j=1}^{N} \). For the iterative process, we take the \( j \)th layer as an example, and select a direction needle \( a_{j}^{j} \). The steps of ML-LoBCoD algorithm are as follows. (1) The residuals and needles of the previous iteration are used to calculate the residual \( R_{j}^{k} \). (2) The needles \( a_{j} \) are updated. (3) All needles of the previous iteration are used to calculate the residual \( R_{j}^{k} \). (4) All needles of the previous iteration are used to calculate the residual \( R_{j}^{k} \). (5) All needles of the previous iteration are used to calculate the residual \( R_{j}^{k} \). (6) This process is repeated \( k \) times until convergence occurs.

The natural question that arises is whether the slice-based ML-LoBCoD algorithm is guaranteed to converge to the global minimum of the ML-BP problem, and the answer is positive as will be discussed in Section 5.

5. Convergence of Slice-Based ML-LoBCoD

In this section, we will propose the convergence theorem and provide proof of convergence of the slice-based ML-LoBCoD algorithm.

**Theorem 2** (convergence of ML-LoBCoD).

Given signal \( X \) and local dictionary \( \{D_{L,j}\}_{j=1}^{N} \), the slice-based ML-LoBCoD algorithm will converge with convergence rate \( O(1/k) \).

**Proof.** The definition of nonexpansive operators [15] is employed, which are guaranteed to converge to their fixed point. Firstly, the definition of internal operators and external operators is given as follows:

\[ T_{in}a_{j} = \text{prox}_{g_{j}}(a_{j} - \mu D_{L,j}^{T}(D_{L,j}a_{j} - P_{j}R_{j})), \]

\[ T_{out}a_{j+1} = \text{prox}_{g_{j+1}}(a_{j+1} - \frac{t}{\mu}D_{L,j+1}^{T}(D_{L,j+1}a_{j+1} - T_{in}D_{L,j+1}a_{j+1})). \] (27)

An operator \( T \) is nonexpansive if it is Lipschitz continuous with constant 1, i.e., \( \|T\alpha - T\beta\| \leq \|\alpha - \beta\| \). In addition, if an operator is firmly nonexpansive, then it must be nonexpansive. A firmly nonexpansive operator needs to satisfy the following conditions:

\[ \|T\alpha - T\beta\|_{2} \leq \|T\alpha - T\beta\|_{2} (\alpha - \beta). \] (28)

The proximal operator is firmly nonexpansive. The internal operator will now be analyzed as follows:

\[ \|T_{in}a_{j} - T_{in}a_{j}\|_{2}^{2} \leq (T_{in}a - T_{in}a_{j})^{T}(\alpha_{j} - \mu D_{L,j}^{T}(D_{L,j}a_{j} - P_{j}R_{j})), \]

\[ - (\beta_{j} - \mu D_{L,j}^{T}(D_{L,j}a_{j} - P_{j}R_{j}))) \]

\[ = (T_{out}a_{j+1} - T_{out}a_{j+1})^{T}(1 - \frac{t}{\mu}D_{L,j+1}^{T}(1 - T_{in})D_{L,j+1}), \]

\[ \leq (T_{out}a - T_{out}a_{j+1})^{T}(\alpha_{j+1} - \beta_{j+1}), \] (30)

Therefore, the ML-LoBCoD algorithm is a nonexpansive operator and it converges to a fixed point.

Next, we analyze the fixed point which is defined as \( \alpha_{j+1} = T_{out}a_{j+1} \) for ML-LoBCoD algorithm. Using the second proximal theorem [16], we obtain

\[ \frac{t}{\mu}D_{L,j+1}^{T}(D_{L,j+1}a_{j+1} - T_{in}D_{L,j+1}a_{j+1}) \in \partial g_{j+1}(\alpha_{j+1}). \] (31)

Assuming that there is \( w_{j+1} \in \partial g(\alpha_{j+1}) \), the above expression can be rewritten as

\[ \frac{t}{\mu}D_{L,j+1}^{T}(D_{L,j+1}a_{j+1} - T_{in}D_{L,j+1}a_{j+1}) + w_{j+1} = 0. \] (32)

Using the second proximal theorem, we obtain \( w_{j} \in \partial g_{j}(T_{in}D_{L,j}a_{j+1}) \) and
\textbf{Algorithm 1: Multilayer LoBCoD.}

\begin{align*}
D_{L,j+1}^*\alpha_{j+1}^* - \mu Vf(D_{L,j+1}^*\alpha_{j+1}^*) - T_{in}(D_{L,j+1}^*\alpha_{j+1}^*) = \mu w_{j+1}.
\end{align*}

Combining these equations together, we obtain
\begin{align*}
D_{L,j+1}^*\left(\frac{1}{\mu}D_{L,j+1}^*\alpha_{j+1}^* - D_{L,j+1}^*\alpha_{j+1}^* - \mu Vf(D_{L,j+1}^*\alpha_{j+1}^*) + \mu w_{j+1}\right)
+ w_{j+1} = 0.
\end{align*}

The fixed point satisfies the optimization condition of the convex function, so the algorithm will converge to a fixed point.

Next, the nonexpansive operator $T_{out}$ is used to analyze the convergence rate of this algorithm:
\begin{align*}
\|\alpha_{j}^{k+1} - \alpha_{j}^{*}\|_2^2 &= \left\|\left(\alpha_{j}^{k+1} - \alpha_{j}^{*}\right) - \left(\alpha_{j}^{k} - \alpha_{j}^{*}\right)\right\|_2^2 \\
&= \|\alpha_{j}^{k+1} - \alpha_{j}^{*}\|_2^2 + \|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2 - 2\left(\alpha_{j}^{k+1} - \alpha_{j}^{*}\right)\cdot\left(\alpha_{j}^{k} - \alpha_{j}^{*}\right) \\
&= \|\alpha_{j}^{k+1} - \alpha_{j}^{*}\|_2^2 + \|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2 - 2\left(T_{out}\alpha_{j}^{k} - T_{out}\alpha_{j}^{*}\right)^T\left(\alpha_{j}^{k} - \alpha_{j}^{*}\right) \\
&\leq \|\alpha_{j}^{k+1} - \alpha_{j}^{*}\|_2^2 - \|\alpha_{j}^{k+1} - \alpha_{j}^{*}\|_2^2.
\end{align*}

Summing all the inequalities above from $i = 1, \ldots, k - 1$, we can obtain
\begin{align*}
\sum_{i=0}^{k-1}\|\alpha_{j}^{i+1} - \alpha_{j}^{*}\|_2^2 &\leq \sum_{i=0}^{k-1}\left(\|\alpha_{j}^{i+1} - \alpha_{j}^{*}\|_2^2 - \|\alpha_{j}^{i} - \alpha_{j}^{*}\|_2^2\right) \\
&= \|\alpha_{j}^{0} - \alpha_{j}^{*}\|_2^2 - \|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2.
\end{align*}

Therefore, we obtain
\begin{align*}
k\|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2 &\leq \sum_{i=0}^{k-1}\|\alpha_{j}^{i+1} - \alpha_{j}^{*}\|_2^2 \\
&\leq \sum_{i=0}^{k-1}\left(\|\alpha_{j}^{i+1} - \alpha_{j}^{*}\|_2^2 - \|\alpha_{j}^{i} - \alpha_{j}^{*}\|_2^2\right) \\
&= \|\alpha_{j}^{0} - \alpha_{j}^{*}\|_2^2 - \|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2.
\end{align*}

Finally, the results are obtained:
\begin{align*}
\|\alpha_{j}^{k} - \alpha_{j}^{*}\|_2^2 &\leq \frac{1}{k+1}\|\alpha_{j}^{0} - \alpha_{j}^{*}\|_2^2.
\end{align*}

The $j$th needles $\alpha_{j}$ is convergent, i.e., $\Gamma_{j}$ converges. Since the convolution sparse code $\Gamma$ is an interleaved cascade of all $\Gamma_{j}$, thus $\Gamma$ will also converge, and the convergence rate is constant with $\alpha_{j}$. Thus, ML-LoBCoD algorithm converges to the global optimum solution at a rate $O(1/k)$.

\section{6. Experiment and Discussion}

\textbf{6.1. Methods.} In this section, the LoBCoD algorithm and ML-LoBCoD algorithm for image reconstruction are described. They are inspired by the ML-ISTA [11, 17]. We constructed a one-layer CSC model. The LoBCoD algorithm is iterative unfolded into a layer of convolutional recurrent neural networks. We constructed a three-layer CSC model. The ML-LoBCoD algorithm is iterative unfolded into a multilayer convolutional recurrent neural network. ML-LoBCoD-NET iterates 3 times to form a convolutional recurrent structure because the algorithm requires multiple iterations to obtain the optimal performance. The forward process of ML-LoBCoD is that the images are input to the ML-LoBCoD network to obtain the coding, then which is classified by the classification layer. The backward process of ML-LoBCoD is minimizing the total loss of function to update the convolutional dictionary by the backpropagation algorithm. The loss function is the negative log likelihood function that represents the loss of a classification error between the ground truth labels and the predicted labels by the network.
6.2. Experimental Results. In this section, we perform experiments on the MNIST dataset using both the LoBCoD and ML-LoBCoD model for image reconstruction. We constructed a one-layer CSC model and a three-layer CSC model. The one-layer CSC model contains 64 local filters of size $6 \times 6$. In the three-layer CSC model, the first convolutional layer includes 64 local filters of size $6 \times 6$, the second convolutional layer contains 128 filters of size $6 \times 6$, and the third convolutional layer contains 512 filters of size $4 \times 4$. Its parameters are the same as a traditional CNN, and thus the network parameters remain unchanged.

The cause of selecting the MNIST datasets that this dataset is the most popular and the most frequently used image datasets and are also the entry-level benchmark datasets in deep learning community.

Experiments on MNIST using the proposed networks are run on Pytorch platform and Linux system using a single computer with a Nvidia GeForce 1080Ti GPU. The code used to support the findings of this paper inspired by the ML-ISTA [11] in GitHub website and are available from the corresponding author upon request.

The loss function values of LoBCoD and ML-LoBCoD with the number of iterations in the MNIST dataset are shown in Figure 1. It is obvious that the loss function values of the ML-LoBCoD are lower than that of the LoBCoD.

The results of the original test image and the reconstructed image using the LoBCoD method and the ML-LoBCoD method after 100 iterations are shown in Figures 2 and 3, respectively. It can be clearly seen that the reconstruction quality of the ML-LoBCoD is better than that of LoBCoD.

The loss function value, the training time, and the peak signal-to-noise ratio (PSNR) value of the two networks after 100 iterations are shown in Table 1. It can be observed that the loss function and PSNR values of the ML-LoBCoD network is smaller than the LoBCoD network. The PSNR value of the LoBCoD network is 17.51dB. However, the ML-LoBCoD network has longer training time than LoBCoD network.

The ML-LoBCoD network has more model parameters than the LoBCoD network. The model parameters of the two networks are shown in Table 2. The model parameters and of the ML-ISTA and corresponding CNN are given in [11]. It can be observed that the ML-ISTA, ML-LoBCoD, and corresponding CNN have the similar parameters, more than the LoBCoD network.

The classification accuracy results of the four model are shown in Table 3. It can be observed that ML-LoBCoD has the highest accuracy rate, far superior to the LoBCoD. These experimental results demonstrate the superiority of our proposed method.

7. Conclusion

In this paper, the convergence theorem of LoBCoD algorithm was proposed and it was proved that this method converges to the global optimum at a rate $O(1/k)$. Inspired by ML-CSC and slice-based local processing idea, a slice-based ML-CSC model was proposed. Motivated by ML-ISTA and the LoBCoD algorithm, an ML-LoBCoD algorithm was proposed and it was established that this method is guaranteed to converge to the global optimum of ML-BP at a rate of $O(1/k)$. The experiment compares the loss
function value, training time, and PSNR of the two networks. The experiment shows that the reconstruction quality of the ML-LoBCoD network is better than that of LoBCoD network. However, this paper only studies the pursuit problem of ML-CSC. The dictionary learning problem of CSC needs further study. This paper is a preliminary study of ML-LoBCoD network, and further study is needed to obtain a more complete analysis, which will likely contribute to the further understanding of deep learning based on driven optimization algorithms.

### Data Availability

The data used to support the findings of this study are available from the website http://yann.lecun.com/exdb/mnist/.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

The authors thank the National Natural Science Foundation of China (Grant 61473339) and Special Projects on Basic Research Cooperation of Beijing, Tianjin and Hebei (Grant nos. 19JCZDJC65600Z and F2019203583).

### References

[1] F. I. Miertoiu and B. Dumitrescu, “Feasibility pump algorithm for sparse representation under laplacian noise,” Mathematical Problems in Engineering, vol. 2019, Article ID 5615243, 12 pages, 2019.

[2] L. H. Fernando, F. V. Luis, and M. M. I. Sarria, “Detecting image brush editing using the discarded coefficients and intentions,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 5, pp. 15–21, 2019.

[3] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1031–1044, 2010.

[4] M. Ribeiro and A. Gomes, “Contour enhancement algorithm for improving visual perception of deutan and protan dichromats,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 5, pp. 79–88, 2019.

[5] V. Papyan, J. Sulam, and M. Elad, “Working locally thinking globally: theoretical guarantees for convolutional sparse coding,” IEEE Transactions on Signal Processing, vol. 65, no. 21, pp. 5687–5701, 2017.

[6] H. Bristow, A. Eriksson, and S. Lucey, “Fast convolutional sparse coding,” in Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 391–398, Portland, OR, USA, June 2013.

[7] B. Wohlberg, “Efficient convolutional sparse coding,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7173–7177, IEEE, Florence, Italy, May 2014.

[8] V. Papyan, Y. Romano, J. Sulam, and M. Elad, “Convolutional dictionary learning via local processing,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, pp. 5296–5304, October 2017.

[9] E. Zisselman, J. Sulam, and M. Elad, “A local block coordinate descent algorithm for the convolutional sparse coding model,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[10] J. Sulam, V. Papan, Y. Romano, and M. Elad, “Multi-layer convolutional sparse modeling: pursuit and dictionary learning,” IEEE Transactions on Signal Processing, vol. 66, no. 15, 2018.

[11] J. Sulam, V. Papan, Y. Romano, and M. Elad, “On multi-layer basis pursuit, efficient algorithms and convolutional neural networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 1968–1980, 2020.

[12] V. Papyan, Y. Romano, and M. Elad, “Convolutional neural networks analyzed via convolutional sparse coding,” The Journal of Machine Learning Research, vol. 18, no. 1, pp. 2887–2938, 2017.

[13] A. Beck and L. Tetruashvili, “On the convergence of block coordinate descent type methods,” SIAM Journal on Optimization, vol. 23, no. 4, pp. 2037–2060, 2013.

[14] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” Journal of Optimization Theory and Applications, vol. 109, no. 3, pp. 475–494, 2001.

[15] H. H. Bauschke, S. M. Moffat, and X. Wang, “Firmly non-expansive mappings and maximally monotone operators: correspondence and duality,” Set-Valued and Variational Analysis, vol. 20, no. 1, pp. 131–153, 2012.

[16] A. Beck, The Proximal Operator, First-Order Methods in Optimization, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, US, pp. 157–159, 2017.

[17] J. Wang, J. Xia, Q. Yang, and Y. Zhang, “Research on semi-supervised sound event detection based on mean teacher models using ML-LoBCoD-NET,” IEEE Access, vol. 8, pp. 38032–38044, 2020.