Global Convergence of Block Coordinate Descent in Deep Learning

Jinshan Zeng∗†‡ Tim Tsz-Kit Lau∗‡§ Shaobo Lin¶ Yuan Yao‡

January 29, 2019

Abstract

Deep learning has aroused extensive attention due to its great empirical success. The efficiency of the block coordinate descent (BCD) methods has been recently demonstrated in deep neural network (DNN) training. However, theoretical studies on their convergence properties are limited due to the highly nonconvex nature of DNN training. In this paper, we aim at providing a general methodology for provable convergence guarantees for this type of methods. In particular, for most of the commonly used DNN training models involving both two- and three-splitting schemes, we establish the global convergence to a critical point at a $O(1/k)$ rate, where $k$ is the number of iterations. The results extend to general loss functions which have Lipschitz continuous gradients and deep residual networks (ResNets). Our key development adds several new elements to the Kurdyka-Lojasiewicz inequality framework that enables us to carry out the global convergence analysis of BCD in the general scenario of deep learning.

1 Introduction

Tremendous research activities have been dedicated to deep learning due to its great success in some real-world applications such as image classification in computer vision (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012; Sainath et al., 2013), statistical machine translation (Devlin et al., 2014), and especially outperforming human in Go games (Silver et al., 2016).

The practical optimization algorithms for training neural networks can be mainly divided into three categories in terms of the amount of first- and second-order information used, namely, gradient-based, (approximate) second-order and gradient-free methods. Gradient-based methods make use of backpropagation (Rumelhart et al., 1986) to compute gradients of network parameters. Stochastic gradient descent (SGD) method proposed by Robbins and Monro (1951) is the basis. Much of research endeavour is dedicated to adaptive variants of vanilla SGD in recent years, including AdaGrad (Duchi et al., 2011), RMSProp (Tieleman and Hinton, 2012), Adam (Kingma and Ba, 2015) and AMSGrad (Rezende et al., 2018). (Approximate) second-order methods mainly include Newton’s method (LeCun et al., 2012), L-BFGS and conjugate gradient (Le et al., 2011). Despite the great success of these gradient-based methods, they may suffer from the vanishing gradient issue for training deep networks (Goodfellow et al., 2016). As an alternative to overcome this issue, gradient-free methods have been recently adapted to the DNN training, including (but not limited to) block coordinate descent (BCD) methods (Carreira-Perpiñán and Wang, 2014; Zhang and Brand, 2017; Lau et al., 2018; Askari et al., 2018; Gu et al., 2018) and alternating direction method of multipliers (ADMM) (Taylor et al., 2016; Zhang et al., 2016). The main reasons for the surge of attention of these two algorithms are twofold. One reason is that they are gradient-free, thus able to deal with non-differentiable nonlinearities and potentially avoid the vanishing gradient issue (Taylor et al., 2016; Zhang and Brand, 2017). As shown in Figure 1, it is observed that vanilla SGD fails to train a ten-hidden-layer MLPs while BCD still works and achieves a moderate accuracy within a few epochs. The other reason is that BCD and ADMM can be
easily implemented in a distributed and parallel manner (Boyd et al., 2011; Mahajan et al., 2017), therefore in favour of distributed/decentralized scenarios.

![Graph](a) Training accuracy ![Graph](b) Test accuracy

Figure 1: Comparison between BCD and SGD for training ten-hidden-layer MLPs on the MNIST dataset. SGD fails to train such deep neural networks while BCD still works and achieves a moderate accuracy within a few epochs. Refer to Appendix F for details of this experiment.

The BCD methods currently adopted in DNN training run into two categories depending on the specific formulations of the objective functions, namely, the two-splitting formulation and three-splitting formulation (shown in (2.2) and (2.4)), respectively. Examples of the two-splitting formulation include Carreira-Perpiñán and Wang (2014); Zhang and Brand (2017); Askari et al. (2018); Gu et al. (2018), whilst Taylor et al. (2016); Lau et al. (2018) adopt the three-splitting formulation. Convergence studies of BCD methods appeared recently in some restricted settings. In Zhang and Brand (2017), a BCD method was suggested to solve the Tikhonov regularized deep neural network training problem using a lifting trick to avoid the computational hurdle imposed by ReLU. Its convergence was established through the framework of Xu and Yin (2013), where the block multiconvexity and differentiability of the unregularized part of the objective function play central roles in the analysis. However, for other commonly used activations such as sigmoid, the convergence analysis of Xu and Yin (2013) cannot be directly applied since the multiconvexity may be violated. Askari et al. (2018) and Gu et al. (2018) extended the lifting trick introduced by Zhang and Brand (2017) to deal with a class of strictly increasing and invertible activations, and then adapted BCD methods to solve the lifted DNN training models. However, no convergence guarantee was provided in both Askari et al. (2018) and Gu et al. (2018). Following the similar lifting trick as in Zhang and Brand (2017), Lau et al. (2018) proposed a proximal BCD based on the three-splitting formulation of the regularized DNN training problem with ReLU activation. The global convergence was also established through the analysis framework of Xu and Yin (2013). However, similar convergence results for other commonly used activation functions are still lacking.

In this paper, we aim to fill these gaps. Our main contribution is to provide a general methodology to establish the global convergence of these BCD methods in the common DNN training settings, without requiring the block multiconvexity and differentiability assumptions as in Xu and Yin (2013). Instead, our key assumption is the Lipschitz continuity of the activation on any bounded set (see Assumption 1(b)). Specifically, Theorem 1 establishes the global convergence to a critical point at a $O(1/k)$ rate of the BCD methods using the proximal strategy, while extensions to the prox-linear strategy for general losses are provided in Theorem 2 and to residual networks (ResNets) are shown in Theorem 3. Our assumptions are applicable to most cases appeared in the literature. Specifically in Theorem 1, if the loss function, activations, and convex regularizers are lower semicontinuous and either real-analytic (see Definition 1) or semialgebraic
(see Definition 2), and the activations are Lipschitz continuous on any bounded set, then BCD converges to a critical point at the rate of $O(1/k)$ starting from any finite initialization, where $k$ is the iteration number. Note that these assumptions are satisfied by most commonly used DNN training models, where (a) the loss function can be any of the squared, logistic, hinge, exponential or cross-entropy losses, (b) the activation function can be any of ReLU, leaky ReLU, sigmoid, tanh, linear, or polynomial functions, and (c) the regularizer can be any of the squared $\ell_2$ norm, squared Frobenius norm, the elementwise 1-norm, or the sum of squared Frobenius norm and elementwise 1-norm (say, in the vector case, the elastic net by [Zou and Hastie, 2005], or the indicator function of the nonnegative closed half space or a closed interval (see Proposition 1).

Our analysis is based on the Kurdyka-Lojasiewicz (KL) inequality [Lojasiewicz, 1993; Kurdyka, 1998] framework formulated in [Attouch et al., 2013]. However there are several different treatments compared to the state-of-the-art work ([Xu and Yin, 2013]) that enables us to achieve the general convergence guarantee aforementioned. According to [Attouch et al., 2013] Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

Our analysis is based on the Kurdyka-Lojasiewicz (KL) inequality [Lojasiewicz, 1993; Kurdyka, 1998] framework formulated in [Attouch et al., 2013]. However there are several different treatments compared to the state-of-the-art work ([Xu and Yin, 2013]) that enables us to achieve the general convergence guarantee aforementioned. According to [Attouch et al., 2013] Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.

In order to obtain the sufficient descent condition, we exploit the proximal strategy for all non-strongly convex subproblems (see Algorithm 2 and Lemma 4), without requiring block multiconvex assumption used in [Xu and Yin, 2013] Lemma 2.6). In order to establish the relative error condition, we use the Lipschitz continuity of the activation functions and do some careful treatments on the specific updates of the BCD formulation. According to Attouch et al. (2013, Theorem 2.9), the sufficient descent, relative error and continuity conditions, together with the KL assumption yield the global convergence of a nonconvex algorithm.
The main idea of variable splitting is to transform a complicated problem (where the variables are coupled highly nonlinearly) into a relatively simpler one (where the variables are coupled much looser) via introducing some additional variables.

### 2.1.1 Two-splitting formulation.

Considering general deep neural network architectures, the DNN training problem can be naturally formulated as the following model (called two-splitting formulation)\(^4\):

\[
\min_{W, V} L_0(W, V) := R_n(V_N; Y) + \sum_{i=1}^{N} r_i(W_i) + \sum_{i=1}^{N} s_i(V_i)
\]

subject to \( V_i = \sigma_i(W_i V_{i-1}), \quad i = 1, \ldots, N, \) \(^2\)

where \( R_n(V_N; Y) := \frac{1}{n} \sum_{j=1}^{n} \ell((V_N); y_j) \) denotes the empirical risk, \( V := \{V_i\}_{i=1}^{N}, \) \((V_N); j \) is the \( j \)-th column of \( V_N \). In addition, \( r_i \) and \( s_i \) are extended-real-valued, nonnegative functions revealing the priors of the weight variable \( W_i \) and the state variable \( V_i \) (or the constraints on \( W_i \) and \( V_i \) for each \( i = 1, \ldots, N \)), and define \( V_0 := X \). In order to solve the two-splitting formulation (2.2), the following alternative minimization problem was suggested in the literature:

\[
\min_{W, V} L(W, V) := L_0(W, V) + \frac{\gamma}{2} \sum_{i=1}^{N} \| V_i - \sigma_i(W_i V_{i-1}) \|_F^2,
\]

where \( \gamma > 0 \) is a hyperparameter\(^5\).

The DNN training model (2.2) can be very general, where: (a) \( \ell \) can be the squared, logistic, hinge, cross-entropy or other commonly used loss functions; (b) \( \sigma_i \) can be ReLU, leaky ReLU, sigmoid, linear, or other commonly used activation functions; (c) \( r_i \) can be the squared \( \ell_2 \) norm, the \( \ell_1 \) norm, the elastic net (Zou and Hastie 2005), the indicator function of some nonempty closed convex set (such as the nonnegative closed half space or a closed interval \([0, 1]\)), or others; (d) \( s_i \) can be the \( \ell_1 \) norm (Ji et al. 2014), the indicator function of some convex set with simple projection (Zhang and Brand 2017), or others. Particularly, if there is no regularizer or constraint on \( W_i \) (or \( V_i \)), then \( r_i \) (or \( s_i \)) can be zero.

The network architectures considered in this paper exhibit generality to various types of DNNs, including but not limited to the fully (or sparse) connected MLPs (Rosenblatt 1961), convolutional neural networks (CNNs) (Fukushima 1980; LeCun et al. 1998) and residual neural networks (ResNets) (He et al. 2016). For CNNs, the weight matrix \( W_i \) is sparse and shares some symmetry structures represented as permutation invariants, which are linear constraints and up to a linear reparameterization all the main results below are still valid.

Various existing BCD algorithms for DNN training (Carreira-Perpiñán and Wang 2014; Zhang and Brand 2017; Askari et al. 2018; Gu et al. 2018) can be regarded as special cases in terms of the use of the two-splitting formulation (2.2). In fact, Carreira-Perpiñán and Wang (2014) considered a specific DNN training model with squared loss and sigmoid activation function, and proposed the method of auxiliary coordinate (MAC) based on the two-splitting formulation of DNN training (2.2), as a two-block BCD method with the weight variables \( W \) as one block and the state variables \( V \) as the other. For each block, a nonlinear least squares problem is solved by some iterative methods. Furthermore, Zhang and Brand (2017) proposed a BCD type method for DNN training with ReLU and squared loss. To avoid the computational hurdle imposed by ReLU, the DNN training model was relaxed to a smooth multiconvex formulation via lifting ReLU into a higher dimensional space (Zhang and Brand 2017). Such a relaxed BCD is in fact a special case of two-splitting formulation (2.3) with \( \sigma_i \equiv \text{Id}, r_i \equiv 0, s_i(V_i) = \iota_X(V_i), i = 1, \ldots, N, \) where \( X \) is the nonnegative closed half-space with the same dimension of \( V_i \), while Askari et al. (2018) and Gu et al. (2018) extended such lifting trick to more general DNN training settings, of which the activation can be not only

\[\text{Here we consider the regularized DNN training model. The model reduces to the original DNN training model without regularization.}\]

\[\text{In (2.3), we use a uniform hyperparameter } \gamma \text{ for the sum of all quadratic terms for the simplicity of notation. In practice, } \gamma \text{ can be different for each quadratic term and our proof still goes through.}\]

\[\text{The indicator function } \iota_C \text{ of a nonempty convex set } C \text{ is defined as } \iota_C(x) = 0 \text{ if } x \in C \text{ and } +\infty \text{ otherwise.}\]
ReLU, but also sigmoid and leaky ReLU. The general formulations studied in these two papers are also special cases of the two-splitting formulation with different $\sigma_i, r_i$ and $s_i, i = 1, \ldots, N$.

### 2.1.2 Three-splitting formulation.

Note that the variables $W_i$ and $V_{i-1}$ are coupled by the nonlinear activation function in the $i$-th constraint of the two-splitting formulation (2.2), which may bring some difficulties and challenges for solving problem (2.2) efficiently, particularly, when the activation function is ReLU. Instead, the following three-splitting formulation was used in [Taylor et al. 2016; Lau et al. 2018]:

$$\min_{W, V, U} L_0 (W, V) \quad \text{subject to} \quad U_i = W_i V_{i-1}, \quad V_i = \sigma_i(U_i), \quad i = 1, \ldots, N,$$  

(2.4)

where $U := \{U_i\}_{i=1}^N$. From (2.4), the variables are coupled much more loosely, particularly for variables $W_i$ and $V_{i-1}$. As described later, such a three-splitting formulation can be beneficial to designing some more efficient methods, though $N$ auxiliary variables $U_i$’s are introduced. Similarly, the following alternative unconstrained problem was suggested in the literature:

$$\min_{W, V, U} \overline{L}(W, V, U) := L_0 (W, V) + \frac{\gamma}{2} \sum_{i=1}^N \| V_i - \sigma_i(U_i) \|^2 + \| U_i - W_i V_{i-1} \|^2 .$$  

(2.5)

### 2.2 Description of BCD algorithms

In the following, we describe how to adapt the BCD method to Problems (2.3) and (2.5). The main idea of the BCD method of Gauss-Seidel type for a minimization problem with multi-block variables is to update all the variables cyclically while fixing the remaining blocks at their last updated values [Ku and Yin 2013]. In this paper, we consider the BCD method with the backward order (but not limited to this as discussed later) for the updates of variables, that is, the variables are updated from the output layer to the input layer, and for each layer, we update the variables $\{V_i, W_i\}$ cyclically for Problem (2.3) as well as the variables $\{V_i, U_i, W_i\}$ cyclically for Problem (2.5). Since $\sigma_N \equiv \text{Id}$, the output layer is paid special attention. Particularly, for most blocks, we adopt the proximal update strategies for two major reasons: (1) To practically stabilize the training process; (2) To yield the desired “sufficient descent” property for theoretical justification. For each subproblem, we assume that its minimizer can be achieved. The BCD algorithms for Problems (2.3) and (2.5) can be summarized in Algorithms 1 and 2 respectively.

#### Algorithm 1 Two-splitting BCD for DNN Training (2.3)

**Data:** $X \in \mathbb{R}^{d_0 \times n}, Y \in \mathbb{R}^{d_n \times n}$

**Initialization:** $\{W_0, V_N, \}^N_{i=1}, V_0 \equiv V_0 := X$

**Parameters:** $\gamma > 0, \alpha > 0$\footnote{In practice, $\gamma$ and $\alpha$ can vary among blocks and our proof still goes through.}

for $k = 1, \ldots$ do

$V_0 = \text{argmin}_{V_0} \{ s_N(V_N) + \mathcal{R}_a(V_N; Y) + \frac{\gamma}{2} \| V_N - W_N V_{N-1} \|^2 + \frac{\gamma}{2} \| V_N - V_N^{k-1} \|^2 \}$

$W_N = \text{argmin}_{W_N} \{ F_N(W_N) + \frac{\gamma}{2} \| V_0 - W_N V_{N-1} \|^2 + \frac{\gamma}{2} \| W_N - W_N^{k-1} \|^2 \}$

for $i = N - 1, \ldots, 1$ do

$V_i = \text{argmin}_{V_i} \{ s_i(V_i) + \frac{\gamma}{2} \| V_i - \sigma_i(W_i V_{i+1}) \|^2 + \frac{\gamma}{2} \| V_i - V_i^{k-1} \|^2 \}$

$W_i = \text{argmin}_{W_i} \{ r_i(W_i) + \frac{\gamma}{2} \| V_i - \sigma_i(W_i V_{i+1}) \|^2 + \frac{\gamma}{2} \| W_i - W_i^{k-1} \|^2 \}$

end for

end for

One major merit of Algorithm 2 over Algorithm 1 is that in each subproblem, almost all updates are simple proximal updates\footnote{For $V_N^k$-update, we can regard $s_N(V_N) + \mathcal{R}_a(V_N; Y)$ as a new proximal function $\tilde{s}_N(V_N)$.} (or just least squares problems), which usually have closed form solutions to many commonly used DNNs, while a drawback of Algorithm 2 over Algorithm 1 is that more storage memory is required due to the introduction of additional variables $\{U_i\}_{i=1}^N$. Some typical examples leading to the closed form solutions include: (a) $r_i, s_i$ are 0 (i.e., no regularization), or the squared $\ell_2$ norm (a.k.a. weight decay), or the indicator function of a nonempty closed convex set with a simple projection like the nonnegative
Algorithm 2 Three-splitting BCD for DNN training

Samples: $X \in \mathbb{R}^{d_{h} \times n}$, $Y \in \mathbb{R}^{d_{y} \times n}$

Initialization: $\{W_{0}^{i}, V_{0}^{i}, U_{0}^{i}\}_{i=1}^{N}$, $V_{0} \equiv V_{0} := X$

Parameters: $\gamma > 0$, $\alpha > 0$

for $k = 1, \ldots$ do

$V_{N}^{k} = \arg\min_{V_{N}} \{s_{N}(V_{N}) + R_{n}(V_{N}; Y) + \frac{\gamma}{2}\|V_{N} - U_{N}^{k-1}\|_{F}^{2} + \frac{\alpha}{2}\|V_{N} - V_{N}^{k-1}\|_{F}^{2}\}$

$U_{N}^{k} = \arg\min_{U_{N}} \{\frac{\gamma}{2}\|V_{N}^{k} - U_{N}\|_{F}^{2} + \frac{\alpha}{2}\|U_{N} - W_{N}^{k-1}V_{N}^{k-1}\|_{F}^{2}\}$

$W_{N}^{k} = \arg\min_{W_{N}} \{r_{N}(W_{N}) + \frac{\gamma}{2}\|U_{N}^{k} - W_{N}V_{N}^{k-1}\|_{F}^{2} + \frac{\alpha}{2}\|W_{N} - W_{N}^{k-1}\|_{F}^{2}\}$

for $i = N - 1, \ldots, 1$ do

$V_{i}^{k} = \arg\min_{V_{i}} \{s_{i}(V_{i}) + \frac{\gamma}{2}\|V_{i} - \sigma_{i}(U_{i})\|_{F}^{2} + \frac{\alpha}{2}\|U_{i+1} - W_{i+1}V_{i}\|_{F}^{2}\}$

$U_{i}^{k} = \arg\min_{U_{i}} \{\frac{\gamma}{2}\|V_{i}^{k} - \sigma_{i}(U_{i})\|_{F}^{2} + \frac{\alpha}{2}\|U_{i} - W_{i}V_{i}^{k-1}\|_{F}^{2} + \frac{\gamma}{2}\|U_{i} - U_{i}^{k-1}\|_{F}^{2}\}$

$W_{i}^{k} = \arg\min_{W_{i}} \{r_{i}(W_{i}) + \frac{\gamma}{2}\|U_{i}^{k} - W_{i}V_{i}^{k-1}\|_{F}^{2} + \frac{\alpha}{2}\|W_{i} - W_{i}^{k-1}\|_{F}^{2}\}$

end for

end for

3 Global convergence analysis of BCD

In this section, we establish the global convergence of both Algorithm 1 for Problem (2.3), and Algorithm 2 for Problem (2.5), followed by some extensions.

3.1 Main assumptions

First of all, we present our main assumptions, which involve the definitions of real analytic and semialgebraic functions.

Let $h : \mathbb{R}^{p} \to \mathbb{R} \cup \{+\infty\}$ be an extended-real-valued function (respectively, $h : \mathbb{R}^{p} = \mathbb{R}^{q}$ be a point-to-set mapping), its graph is defined by

$$\text{Graph}(h) := \{(x, y) \in \mathbb{R}^{p} \times \mathbb{R} : y = h(x)\},$$

resp.

$$\text{Graph}(h) := \{(x, y) \in \mathbb{R}^{p} \times \mathbb{R}^{q} : y = h(x)\},$$

and its domain by dom($h$) := \{x \in \mathbb{R}^{p} : h(x) < +\infty\} (resp. dom($h$) := \{x \in \mathbb{R}^{p} : h(x) \neq \emptyset\}). When $h$ is a proper function, i.e., when dom($h$) \neq \emptyset, the set of its global minimizers (possibly empty) is denoted by

$$\arg\min h := \{x \in \mathbb{R}^{p} : h(x) = \inf h\}.$$

Definition 1 (Real analytic) A function $h$ with domain an open set $U \subset \mathbb{R}$ and range the set of either all real or complex numbers, is said to be real analytic at $u$ if the function $f$ may be represented by a convergent power series on some interval of positive radius centered at $u$: $h(x) = \sum_{j=0}^{\infty} \alpha_{j}(x - u)^{j}$, for some \{\alpha_{j}\} \subset \mathbb{R}. The function is said to be real analytic on $V \subset U$ if it is real analytic at each $u \in V$ [Krantz and Parks 2002, Definition 1.1.5]. The real analytic function $f$ over $\mathbb{R}^{p}$ for some positive integer $p > 1$ can be defined similarly.

According to [Krantz and Parks 2002], typical real analytic functions include polynomials, exponential functions, and the logarithm, trigonometric and power functions on any open set of their domains. One can verify whether a multivariable real function $h(x)$ on $\mathbb{R}^{p}$ is analytic by checking the analyticity of $g(t) := h(x + ty)$ for any $x, y \in \mathbb{R}^{p}$.

Definition 2 (Semialgebraic)
A set $D \subset \mathbb{R}^p$ is called semialgebraic \cite{Bochnaketal1998} if it can be represented as

$$D = \bigcup_{i=1}^{s} \bigcap_{j=1}^{t} \{x \in \mathbb{R}^p : P_{ij}(x) = 0, Q_{ij}(x) > 0\},$$

where $P_{ij}, Q_{ij}$ are real polynomial functions for $1 \leq i \leq s, 1 \leq j \leq t$.

(b) A function $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ (resp. a point-to-set mapping $h : \mathbb{R}^p \rightrightarrows \mathbb{R}^q$) is called semialgebraic if its graph $\text{Graph}(h)$ is semialgebraic.

According to \textit{Lojasiewicz} (1965), \textit{Bochnak et al.} (1998) and \textit{Shiota} (1997, I.2.9, page 52), the class of semialgebraic sets is stable under the operation of finite union, finite intersection, Cartesian product or complementation. Some typical examples include polynomial functions, the indicator function of a nonempty closed convex set (such as the nonnegative closed half space, box set or a closed interval $[0, 1]$), or no regularization.

3.2 Main theorem

Under Assumption \ref{assumption1}, we state our main theorem as follows.

\textbf{Theorem 1} Let $\{Q^k := (\{W^k_{i1}\}_{i=1}^{N}, \{V^k_{i1}\}_{i=1}^{N})\}_{k \in \mathbb{N}}$ and $\{P^k := (\{W^k_{i1}\}_{i=1}^{N}, \{V^k_{i1}\}_{i=1}^{N}, \{U^k_{i1}\}_{i=1}^{N})\}_{k \in \mathbb{N}}$ be the sequences generated by Algorithms \ref{algorithm1} and \ref{algorithm2} respectively. Suppose that Assumption \ref{assumption1} holds, and that one of the following conditions holds: (i) there exists a convergent subsequence $\{Q^k\}_{j \in \mathbb{N}}$ (resp. $\{P^k\}_{j \in \mathbb{N}}$); (ii) $r_i$ is coercive$^{10}$ for any $i = 1, \ldots, N$; (iii) $\mathcal{L}$ (resp. $\mathcal{Z}$) is coercive. Then for any $\alpha > 0$, $\gamma > 0$ and any finite initialization $Q^0$ (resp. $P^0$), the following hold

(a) $\{\mathcal{L}(Q^k)\}_{k \in \mathbb{N}}$ (resp. $\{\mathcal{Z}(P^k)\}_{k \in \mathbb{N}}$) converges to some $\mathcal{L}^*$ (resp. $\mathcal{Z}^*$).

(b) $\{Q^k\}_{k \in \mathbb{N}}$ (resp. $\{P^k\}_{k \in \mathbb{N}}$) converges to a critical point of $\mathcal{L}$ (resp. $\mathcal{Z}$).

(c) $\frac{1}{K} \sum_{k=1}^{K} \|\mathbf{g}^k\|_F^2 \to 0$ at the rate $O(1/K)$ where $\mathbf{g}^k \in \partial \mathcal{L}(Q^k)$. Similarly, $\frac{1}{K} \sum_{k=1}^{K} \|\mathbf{g}^k\|_F^2 \to 0$ at the rate $O(1/K)$ where $\hat{\mathbf{g}}^k \in \partial \mathcal{Z}(Q^k)$.

Note that the DNN training problems \cite{2.3} and \cite{2.5} in this paper generally do not satisfy such a Lipschitz differentiable property, particularly, when ReLU activation is used. Compared to the existing literature, this theorem establishes the global convergence without the block multicovex and Lipschitz differentiability assumptions used in \textit{Xu and Yin} (2013), which are often violated by the DNN training problems due to the nonlinearity of the activations.

A function $f : X \to \mathbb{R}$ is called lower semicontinuous if $\liminf_{x \to x_0} f(x) \geq f(x_0)$ for any $x_0 \in X$.

An extended-real-valued function $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is called coercive if and only if $h(x) \to +\infty$ as $\|x\| \to +\infty$. 

\end{document}
3.3 Extensions

We extend the established convergence results to the BCD methods for general losses with the prox-linear strategy, and the BCD methods for training ResNets.

3.3.1 Extension to prox-linear

Note that in the \( V_N \)-update of both Algorithms 1 and 2 the empirical risk is involved in the optimization problems. It is generally hard to obtain its closed-form solution except for some special cases such as the case that the loss is the squared loss. For other smooth losses such as the logistic, cross-entropy, and exponential losses, we suggest using the following prox-linear update strategies, that is, for some parameter \( \alpha > 0 \), the \( V_N \)-update in Algorithm 1 is

\[
V_N^k = \arg\min_{V_N} \left\{ s_N(V_N) + \langle \nabla R_n(V_N^{k-1}; Y), V_N - V_N^{k-1} \rangle + \frac{\alpha}{2} \| V_N - V_N^{k-1} \|_F^2 + \frac{\gamma}{2} \| V_N - W_N^{k-1} V_N^{k-1} \|_F^2 \right\},
\]

(3.1)

and the \( V_N \)-update in Algorithm 2 is

\[
V_N^k = \arg\min_{V_N} \left\{ s_N(V_N) + \langle \nabla R_n(V_N^{k-1}; Y), V_N - V_N^{k-1} \rangle + \frac{\alpha}{2} \| V_N - V_N^{k-1} \|_F^2 + \frac{\gamma}{2} \| V_N - U_N^{k-1} \|_F^2 \right\}.
\]

(3.2)

From (3.1) and (3.2), when \( s_N \) is zero or its proximal operator can be easily computed, then \( V_N \)-updates for both BCD algorithms can be implemented with explicit expressions. Therefore, the specific uses of these BCD methods are very flexible, mainly depending on users’ understanding of their own problems.

The claims in Theorem 1 still hold for the prox-linear updates adopted for the \( V_N \)-updates if the loss is smooth with Lipschitz continuous gradient, as stated in the following Theorem 2:

**Theorem 2 (Global convergence for prox-linear update)** Consider adopting the prox-linear updates (3.1), (3.2) for the \( V_N \)-subproblems in Algorithms 1 and 2 respectively. Under the conditions of Theorem 1 if further \( \nabla R_n \) is Lipschitz continuous with a Lipschitz constant \( L_R \) and \( \alpha > \max \left\{ 0, \frac{L_R - \gamma}{2} \right\} \), then all claims in Theorem 1 still hold for both algorithms.

The proof of Theorem 2 is presented in Appendix D. It establishes the global convergence of a BCD method for the commonly used DNN training models with nonlinear losses, such as logistic or cross-entropy losses, etc. Equipped with the prox-linear strategy, all updates of BCD can be implemented easily and allow large scale distributed computations.

3.3.2 Extension to ResNets Training

In this section, we first adapt the BCD method to the residual networks (ResNets) (He et al., 2016), and then extend the established convergence results of BCD to this case. Without loss of generality, similar to (2.2), we consider the following simplified ResNets training problem,

\[
\min_{W, V} \mathcal{R}_n(V_N; Y) + \sum_{i=1}^N r_i(W_i) + \sum_{i=1}^N s_i(V_i) \quad \text{subject to} \quad V_i - V_{i-1} = \sigma_i(W_i V_{i-1}), \quad i = 1, \ldots, N.
\]

(3.3)

Since the ReLU activation is usually used in ResNets, we only consider the three-splitting formulation of (3.3):

\[
\min_{W, V, U} \mathcal{R}_n(V_N; Y) + \sum_{i=1}^N r_i(W_i) + \sum_{i=1}^N s_i(V_i) \quad \text{subject to} \quad U_i = W_i V_{i-1}, \quad V_i - V_{i-1} = \sigma_i(U_i), \quad i = 1, \ldots, N,
\]

and then adapt BCD to the following minimization problem,

\[
\min_{W, V, U} \mathcal{Z}_{\text{res}}(W, V, U),
\]

(3.4)
where $W := \{W_i\}_{i=1}^N$, $V := \{V_i\}_{i=1}^N$, $U := \{U_i\}_{i=1}^N$ as defined before, and 

$$
\mathcal{L}_{\text{res}}(W, V, U) := \mathcal{R}_n(V_N; Y) + \sum_{i=1}^N r_i(W_i) + \sum_{i=1}^N s_i(V_i) + \frac{\gamma}{2} \sum_{i=1}^N \left[ \|V_i - V_{i-1} - \sigma_i(U_i)\|_F^2 + \|U_i - W_iV_{i-1}\|_F^2 \right].
$$

When applied to (3.4), we use the same update order of Algorithm 2 but slightly change the subproblems according to the objective $\mathcal{L}_{\text{res}}$ in (3.4). The specific BCD algorithm for ResNets is presented in Algorithm 3 in Appendix D.

Similarly, we establish the convergence of BCD for the DNN training model with ResNets (3.4) as follows.

**Theorem 3 (Convergence of BCD for ResNets)** Let $\{\{W_i^k, V_i^k, U_i^k\}\}_{i=1}^N$ be a sequence generated by BCD for the DNN training model with ResNets (i.e., Algorithm 3). Let assumptions of Theorem 1 hold. Then all claims in Theorem 1 still hold for BCD with ResNets via replacing $\mathcal{L}$ with $\mathcal{L}_{\text{res}}$.

Moreover, consider adopting the prox-linear update for the $V_N$-subproblem in Algorithm 3 then under the assumptions of Theorem 2, all claims of Theorem 2 still hold for Algorithm 3.

The proof of this theorem is presented in Appendix D. ResNets is one of the most popular network architectures used in the deep learning community and has profound applications in computer vision. How to efficiently train ResNets is thus very important, especially since it is not of a fully-connected structure. The theorem, for the first time, shows that the BCD method might be a good candidate for the training of ResNets with global convergence guarantee.

4 Key stones and discussions

In this section, we present the keystones of our proofs followed by some discussions.

4.1 Main ideas of proofs

Our proofs follow the analysis framework formulated in Attouch et al. (2013), where the establishments of the sufficient descent and relative error conditions and the verifications of the continuity condition and KL property of the objective function are the four key ingredients. In order to establish the sufficient descent and relative error properties, two kinds of assumptions, namely, (a) multiconvexity and differentiability assumption, and (b) (blockwise) Lipschitz differentiability assumption on the unregularized part of objective function are commonly used in the literature, where Xu and Yin (2013) mainly used assumption (a), and the literature (Attouch et al. 2013, Xu and Yin 2017, Bolte et al. 2014) mainly used assumption (b). Note that in our cases, the unregularized part of $\mathcal{L}$ in (2.3),

$$
\mathcal{R}_n(V_N; Y) + \frac{\gamma}{2} \sum_{i=1}^N \|V_i - \sigma_i(W_iV_{i-1})\|_F^2,
$$

and that of $\mathcal{L}$ in (2.5),

$$
\mathcal{R}_n(V_N; Y) + \frac{\gamma}{2} \sum_{i=1}^N \left[ \|V_i - \sigma_i(U_i)\|_F^2 + \|U_i - W_iV_{i-1}\|_F^2 \right]
$$

usually do not satisfy any of the block multiconvexity and differentiability assumption (i.e., assumption (a)), and the blockwise Lipschitz differentiability assumption (i.e., assumption (b)). For instance, when $\sigma_i$ is ReLU or leaky ReLU, the functions $\|V_i - \sigma_i(W_iV_{i-1})\|_F^2$ and $\|V_i - \sigma_i(U_i)\|_F^2$ are non-differentiable and nonconvex with respect to $W_i$-block and $U_i$-block, respectively.

In order to overcome these challenges, in this paper, we first exploit the proximal strategies for all the non-strongly convex subproblems (see Algorithm 2) to cheaply obtain the desired sufficient descent property (see Lemma 1), and then take advantage of the Lipschitz continuity of the activations as well as the specific splitting formulations to yield the desired relative error property (see Lemma 2). Below we present these two key lemmas, while leaving other details in Appendix (where the verification of the KL property for the
concerned DNN training models satisfying Assumption 1 can be referred to Proposition 2 in Appendix C.1 and the verification of the continuity condition is shown by (C.19) in Appendix C.3.2. Based on Lemmas 1 and 2, Proposition 2 and (C.19), we prove Theorem 1 according to Attouch et al. (2013) Theorem 2.9, with details shown in Appendix C.

4.2 Sufficient descent lemma

We state the established sufficient descent lemma as follows.

**Lemma 1 (Sufficient descent)** Let \( \{P^k\}_{k \in \mathbb{N}} \) be a sequence generated by the BCD method (Algorithm 2), under assumptions of Theorem 1, then

\[
\mathcal{Z}(P^k) \leq \mathcal{Z}(P^{k-1}) - a\|P^k - P^{k-1}\|_F^2,
\]

for some constant \( a > 0 \) specified in the proof.

From Lemma 1 the Lagrangian sequence \( \{\mathcal{Z}(P^k)\} \) is monotonically decreasing, and the descent quantity of each iterate can be lower bounded by the discrepancy between the current iterate and its previous iterate. This lemma is crucial for the global convergence of a nonconvex algorithm. It tells at least the following four important items: (i) \( \{\mathcal{Z}(P^k)\}_{k \in \mathbb{N}} \) is convergent if \( \mathcal{Z} \) is lower bounded; (ii) \( \{P^k\}_{k \in \mathbb{N}} \) itself is bounded if \( \mathcal{Z} \) is coercive and \( P^0 \) is finite; (iii) \( \{P^k\}_{k \in \mathbb{N}} \) is square summable, i.e., \( \sum_{k=1}^{\infty} \|P^k - P^{k-1}\|_F^2 < \infty \), implying its asymptotic regularity, i.e., \( \|P^k - P^{k-1}\|_F \to 0 \) as \( k \to \infty \); and (iv) \( \frac{1}{K} \sum_{k=1}^{K} \|P^k - P^{k-1}\|_F^2 \to 0 \) at the rate of \( O(1/K) \). Leveraging on Lemma 1, we can establish the global convergence (i.e., the whole sequence convergence) of BCD in DNN training settings. In contrast, Davis et al. (2019) only establish the subsequence convergence of BCD in DNN training settings. Such a gap between the subsequence convergence of SGD and the whole sequence convergence of BCD in this paper exists mainly because SGD can only achieve the descent property but not the sufficient descent property.

It can be noted from Lemma 1 that neither multiconvexity and differentiability nor Lipschitz differentiability assumption is imposed on the DNN training models to yield this lemma, as required in the literature (Xu and Yin, 2013; Attouch et al., 2013; Xu and Yin, 2017; Bolte et al., 2014). Instead, we mainly exploit the proximal strategy for all non-strongly convex subproblems in Algorithm 2 to establish this lemma.

4.3 Relative error lemma

We now present the obtained relative error lemma.

**Lemma 2 (Relative error)** Under the conditions of Theorem 1, let \( B \) be an upper bound of \( P^{k-1} \) and \( P^k \) for any positive integer \( k \), \( L_B \) be a uniform Lipschitz constant of \( \sigma_i \) on the bounded set \( \{P : \|P\|_F \leq B\} \). Then for any positive integer \( k \), it holds that,

\[
\|g^k\|_F \leq \bar{b}\|P^k - P^{k-1}\|_F, \quad g^k \in \partial \mathcal{L}(P^k)
\]

for some constant \( \bar{b} > 0 \) specified later in the proof, where

\[
\partial \mathcal{L}(P^k) := \{(\partial_{\mathcal{L}}^{\mathcal{Z}})^N_{i=1}, (\partial_{\mathcal{L}}^U)^N_{i=1}, (\partial_{\mathcal{L}}^V)^N_{i=1}(P^k)\}.
\]

Lemma 2 shows that the subgradient sequence of the Lagrangian is upper bounded by the discrepancy between the current and previous iterates. Together with the asymptotic regularity of \( \{P^k\}_{k \in \mathbb{N}} \) yielded by Lemma 1, Lemma 2 shows the critical point convergence. Also, together with the claim (iv) implied by Lemma 1, namely, the \( O(1/K) \) rate of convergence of \( \frac{1}{K} \sum_{k=1}^{K} \|P^k - P^{k-1}\|_F^2 \to 0 \), Lemma 2 yields the \( O(1/K) \) rate of convergence (to a critical point) of BCD, i.e., \( \frac{1}{K} \sum_{k=1}^{K} \|g^k\|_F \to 0 \) at the rate of \( O(1/K) \).

From Lemma 2 both differentiability and (blockwise) Lipschitz differentiability assumptions are not imposed. Instead, we only use the Lipschitz continuity (on any bounded set) of the activations, which is a very mild and natural condition satisfied by most commonly used activation functions. In order to achieve this lemma, we also need to do some special treatments on the specific updates of BCD algorithms as demonstrated in Appendix C.3.1.
5 Conclusion

The empirical efficiency of BCD methods in deep neural network (DNN) training has been demonstrated in the literature. However, the theoretical understanding of their convergence is still very limited and it lacks a general framework due to the fact that DNN training is a highly nonconvex problem. In this paper, we fill this void by providing a general methodology to establish the global convergence of the BCD methods for a class of DNN training models, which encompasses most of the commonly used BCD methods in the literature as special cases. Under some mild assumptions, we establish the global convergence at a rate of $O(1/k)$, with $k$ being the number of iterations, to a critical point of the DNN training models with several variable splittings. Our theory is also extended to residual networks with general losses which have Lipschitz continuous gradients. Such work may lay down a theoretical foundation of BCD methods in their applications to deep learning.

Acknowledgment

The work of Jinshan Zeng is supported in part by the National Natural Science Foundation (NNSF) of China (No. 61603162, 61876074). The work of Yuan Yao is supported in part by HKRGC grant 16303817, 973 Program of China (No. 2015CB85600), NNSF of China (No. 61370004, 11421110001), as well as grants from Tencent AI Lab, Si Family Foundation, Baidu BDI, and Microsoft Research-Asia. The work of Shaobo Lin is supported in part by the NNSF of China (No. 61876133). The work of Tim Tsz-Kit Lau is supported by the University Fellowship funded by The Graduate School at Northwestern University, and part of this work was performed when he was at Department of Mathematics, The Hong Kong University of Science and Technology.
Appendix

A  Implementations on BCD methods

In this section, we provide several remarks to discuss the specific implementations of BCD methods.

Remark 1 (On the initialization of parameters) In practice, the weights \( \{W_i\}_{i=1}^N \) are generally initialized according to some Gaussian distributions with small standard deviations. The bias vectors are usually set as all one vectors scaling by some small constants. Given the weights and bias vectors, the auxiliary variables \( \{U_i\}_{i=1}^N \) and state variables \( \{V_i\}_{i=1}^N \) are usually initialized by a single forward pass through the network.

Remark 2 (On the update order) We suggest such a backward update order in this paper due to the nested structure of DNNs. Besides the update order presented in Algorithm 2, any arbitrary deterministic update order can be incorporated into the BCD methods, and our proofs still go through.

Remark 3 (On the distributed implementation) One major advantage of BCD is that it can be implemented in distributed and parallel manner like in ADMM. Specifically, given \( m \) servers, the total training data are distributed to these servers. Denotes \( S_j \) as the subset of samples at server \( j \). Thus, \( n = \sum_{j=1}^m \#(S_j) \), where \( \#(S_j) \) denotes the cardinality of \( S_j \). For each layer \( i \), the state variable \( V_i \) is divided into \( m \) submatrices by column, that is, \( V_i := ((V_i)_1, \ldots, (V_i)_m) \), where \( (V_i)_j \) denotes the submatrix of \( V_i \) including all the columns in the index set \( S_j \). The auxiliary variables \( U_j \)'s are decomposed similarly. From Algorithm 2, the updates of \( \{V_i\}_{i=1}^N \) and \( \{U_i\}_{i=1}^N \) do not need any communication and thus, can be computed in a parallel way. The difficult part is the update of weight \( W_i \), which is generally hard to parallelize. To deal with this part, there are some effective strategies suggested in the literature like Taylor et al. (2016).

B  Proof of Proposition 1

Proof We verify these special cases as follows.

On the loss function \( \ell \): Since these losses are all nonnegative and continuous on their domains, thus, they are proper lower semicontinuous and lower bounded by 0. In the following, we only verify that they are either real analytic or semialgebraic.

(a1) If \( \ell(t) \) is the squared \( (t^2) \) or exponential \( (e^t) \), then according to Krantz and Parks (2002), they are real analytic.

(a2) If \( \ell(t) \) is the logistic loss \( \log(1 + e^{-t}) \), since it is a composition of logarithm and exponential functions which both are real analytic, thus according to Lemma 3, the logistic loss is real analytic.

(a3) If \( \ell(u; y) \) is the cross-entropy loss, that is, given \( y \in \mathbb{R}^{d_N} \), \( \ell(u; y) := -\frac{1}{d_N} [(y, \log \hat{y}(u)) + (1 - y, \log(1 - \hat{y}(u)))] \), where log is performed elementwise and \( \hat{y}(u)_{1 \leq i \leq d_N} := ((1 + e^{-u_i})^{-1})_{1 \leq i \leq d_N} \) for any \( u \in \mathbb{R}^{d_N} \), which can be viewed as a linear combination of logistic functions, then by (a2) and Lemma 3, it is also real analytic.

(a4) If \( \ell \) is the hinge loss, that is, given \( y \in \mathbb{R}^{d_N} \), \( \ell(u; y) := \max\{0, 1 - u^\top y\} \) for any \( u \in \mathbb{R}^{d_N} \), by Lemma 1, it is semialgebraic, because its graph is \( \text{cl}(D) \), a closure of the set \( D \), where

\[
D = \{(u, z) : 1 - u^\top y - z = 0, 1 - u > 0\} \cup \{(u, z) : z = 0, u^\top y - 1 > 0\}.
\]

On the activation function \( \sigma_i \): Since all the considered specific activations are continuous on their domains, they are Lipschitz continuous on any bounded set. In the following, we only need to check that they are either real analytic or semialgebraic.

(b1) If \( \sigma_i \) is a linear or polynomial function, then according to Krantz and Parks (2002), \( \sigma_i \) is real analytic.
(b2) If \( \sigma_i(t) \) is sigmoid, \((1 + e^{-t})^{-1}\), or hyperbolic tangent, \( \tanh t := \frac{e^t - e^{-t}}{e^t + e^{-t}} \), then the sigmoid function is a composition \( g \circ h \) of these two functions where \( g(u) = \frac{1}{1 + u} \), \( u > 0 \) and \( h(t) = e^{-t} \) (resp. \( g(u) = 1 - \frac{2}{u+1}, u > 0 \) and \( h(t) = e^{2t} \) in the hyperbolic tangent case). According to Krantz and Parks (2002), \( g \) and \( h \) in both cases are real analytic. Thus, according to Lemma 3, sigmoid and hyperbolic tangent functions are real analytic.

(b3) If \( \sigma_i \) is ReLU, i.e., \( \sigma_i(u) := \max\{0, u\} \), then we can show that ReLU is semialgebraic since its graph is \( \text{cl}(D) \), a closure of the set \( D \), where

\[
D = \{(u, z) : u - z = 0, u > 0\} \cup \{(u, z) : z = 0, -u > 0\}.
\]

(b4) Similar to the ReLU case, if \( \sigma_i \) is leaky ReLU, i.e., \( \sigma_i(u) = u \) if \( u > 0 \), otherwise \( \sigma_i(u) = au \) for some \( a > 0 \), then we can similarly show that leaky ReLU is semialgebraic since its graph is \( \text{cl}(D) \), a closure of the set \( D \), where

\[
D = \{(u, z) : u - z = 0, u > 0\} \cup \{(u, z) : au - z = 0, -u > 0\}.
\]

(b5) If \( \sigma_i \) is polynomial as used in Liao and Poggio (2017), then according to Krantz and Parks (2002), it is real analytic.

On \( r_i(W_i), s_i(V_i) \): By the specific forms of these regularizers, they are nonnegative, lower semicontinuous and continuous on their domain. In the following, we only need to verify they are either real analytic and semialgebraic.

(c1) the squared \( \ell_2 \) norm \( \| \cdot \|^2_2 \): According to Bochnak et al. (1998), the \( \ell_2 \) norm is semialgebraic, so is its square according to Lemma 4(2), where \( g(t) = t^2 \) and \( h(W) = \|W\|_2 \).

(c2) the squared Frobenius norm \( \| \cdot \|^2_2 \): The squared Frobenius norm is semialgebraic since it is a finite sum of several univariate squared functions.

(c3) the elementwise 1-norm \( \| \cdot \|_{1,1} \): Note that \( \|W\|_{1,1} = \sum_{i,j} |W_{ij}| \) is the finite sum of absolute functions \( h(t) = |t| \). According to Lemma 4(1), the absolute value function is semialgebraic since its graph is the closure of the following semialgebraic set

\[
D = \{(t, s) : t + s = 0, -t > 0\} \cup \{(t, s) : t - s = 0, t > 0\}.
\]

Thus, the elementwise 1-norm is semialgebraic.

(c4) the elastic net: Note that the elastic net is the sum of the elementwise 1-norm and the squared Frobenius norm. Thus, by (c2), (c3) and Lemma 4(3), the elastic net is semialgebraic.

(c5) If \( r_i \) or \( s_i \) is the indicator function of nonnegative closed half space or a closed interval (box constraints), by Lemma 4(1), any polyhedral set is semialgebraic such as the nonnegative orthant \( \mathbb{R}_{+}^{p \times q} = \{W \in \mathbb{R}^{p \times q}, W_{ij} \geq 0, \forall i, j\} \), and the closed interval. Thus, by Lemma 4(4), \( r_i \) or \( s_i \) is semialgebraic in this case.

C Proof of Theorem 1

To prove Theorem 1, we first show that the Kurdyka-Łojasiewicz (KL) property holds for the considered DNN training models (see Proposition 2), then establish the function value convergence of the BCD methods (see Theorem 1), followed by establishing their global convergence as well as the \( \mathcal{O}(1/k) \) convergence rate to a critical point as shown in Theorem 5. Combining Proposition 2, Theorems 4 and 5 yields Theorem 1.
C.1 The Kurdyka-Lojasiewicz Property in Deep Learning

Before giving the definition of the KL property, we first introduce some notions and notations from variational analysis, which can be found in Rockafellar and Wets (1998).

The notion of subdifferential plays a central role in the following definition of the KL property. For each $x \in \text{dom}(h)$, the Fréchet subdifferential of $h$ at $x$, written $\partial h(x)$, is the set of vectors $v \in \mathbb{R}^p$ which satisfy
\[ \liminf_{y \to x, y \neq x} \frac{h(y) - h(x) - \langle v, y-x \rangle}{\|y-x\|} \geq 0. \]

When $x \notin \text{dom}(h)$, we set $\partial h(x) = \emptyset$. The limiting-subdifferential (or simply subdifferential) of $h$ introduced in Mordukhovich (2006), written $\partial h(x)$ at $x \in \text{dom}(h)$, is defined by
\[ \partial h(x) := \{ v \in \mathbb{R}^p : \exists x^k \to x, h(x^k) \to h(x), v^k \to \partial h(x^k) \to v \}. \] (C.1)

A necessary (but not sufficient) condition for $x \in \mathbb{R}^p$ to be a minimizer of $h$ is $0 \in \partial h(x)$. A point that satisfies this inclusion is called limiting-critical or simply critical. The distance between a point $x$ to a subset $S$ of $\mathbb{R}^p$, written $\text{dist}(x, S)$, is defined by $\text{dist}(x, S) = \inf\{\|x-s\| : s \in S\}$, where $\| \cdot \|$ represents the Euclidean norm.

The KL property (Lojasiewicz 1963, 1993; Kurdyka 1998; Bolte et al., 2007a,b) plays a central role in the convergence analysis of nonconvex algorithms (see e.g., Attouch et al., 2013; Xu and Yin, 2013; Wang et al., 2018). The following definition is adopted from Bolte et al. (2007a).

**Definition 3 (Kurdyka-Lojasiewicz property)** A function $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is said to have the Kurdyka-Lojasiewicz (KL) property at $x^* \in \text{dom}(\partial h)$ if there exist a neighborhood $U$ of $x^*$, a constant $\eta$, and a continuous concave function $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in (0, 1)$ such that the Kurdyka-Lojasiewicz inequality holds: For all $x \in U \cap \text{dom}(\partial h)$ and $h(x) < h(x^*) + \eta$,
\[ \phi'(h(x) - h(x^*)) \cdot \text{dist}(0, \partial h(x)) \geq 1, \] (C.2)
where $\theta$ is called the KL exponent of $h$ at $x^*$. Proper lower semi-continuous functions which satisfy the Kurdyka-Lojasiewicz inequality at each point of $\text{dom}(\partial h)$ are called KL functions.

Note that we have adopted in the definition of the KL inequality (C.2) the following notational conventions: $0^0 = 1$, $\infty/\infty = 0/0 = 0$. Such property was firstly introduced by Lojasiewicz (1993) on real analytic functions (Krantz and Parks, 2002) for $\theta \in [\frac{1}{2}, 1)$, then was extended to functions defined on the o-minimal structure in Kurdyka (1998), and later was extended to nonsmooth subanalytic functions in Bolte et al. (2007a).

By the definition of the KL property, it means that the function under consideration is sharp up to a reparametrization (Attouch et al., 2013). Particularly, when $h$ is smooth, finite-valued, and $h(x^*) = 0$, the inequality (C.2) can be rewritten
\[ \|\nabla(\phi \circ h)(x)\| \geq 1, \]
for each convenient $x \in \mathbb{R}^p$. This inequality may be interpreted as follows: up to the reparametrization of the values of $h$ via $\phi$, we face a sharp function. Since the function $\phi$ is used here to turn a singular region—a region in which the gradients are arbitrarily small—into a regular region, i.e., a place where the gradients are bounded away from zero, it is called a desingularizing function for $h$. For theoretical and geometrical developments concerning this inequality, see Bolte et al. (2007b). KL functions include real analytic functions (see Definition 1), semialgebraic functions (see Definition 2), tame functions defined in some o-minimal structures (Kurdyka, 1998), continuous subanalytic functions (Bolte et al., 2007a,b) and locally strongly convex functions (Xu and Yin, 2013).

In the following, we establish the KL properties\(^\dagger\) of the DNN training models with variable splitting, i.e., the functions $\mathcal{L}$ defined in (2.3) and $\mathcal{Z}$ defined in (2.5).

**Proposition 2 (KL properties of deep learning)** Suppose that Assumption 1 holds. Then the functions $\mathcal{L}$ defined in (2.3), and $\mathcal{Z}$ defined in (2.5) are KL functions.

\(^\dagger\)It should be pointed out that we need to use the vectorization of the matrix variables involved in $\mathcal{L}$, $\mathcal{Z}$ and $\mathcal{Z}_{\text{vec}}$ in order to adopt the existing definitions of KL property, real analytic functions and semialgebraic functions. We still use the matrix notation for the simplicity of notation.
This proposition shows that most of the DNN training models with variable splitting have some nice geometric properties, i.e., they are amenable to sharpness at each point in their domains. In order to prove this theorem, we need the following lemmas. The first lemma shows some important properties of real analytic functions.

**Lemma 3 (Krantz and Parks, 2002)** The sums, products, and compositions of real analytic functions are real analytic functions.

Then we present some important properties of semialgebraic sets and mappings, which can be found in Bochnak et al. (1998).

**Lemma 4** The following hold:

1. Finite union, finite intersection, or complementation of semialgebraic sets is semialgebraic. The closure and the interior of a semialgebraic set are semialgebraic (Bochnak et al., 1998, Proposition 2.2.2).
2. The composition $g \circ h$ of semialgebraic mappings $h : A \to B$ and $g : B \to C$ is semialgebraic (Bochnak et al., 1998, Proposition 2.2.6).
3. The sum of two semialgebraic functions is a semialgebraic function (can be referred to the proof of (Bochnak et al., 1998, Proposition 2.2.6)).
4. The indicator function of a semialgebraic set is semialgebraic (Bochnak et al., 1998).

**Lemma 5** The following hold:

1. Both real analytic functions and semialgebraic functions (mappings) are subanalytic (Shiota, 1997).
2. Let $f_1$ and $f_2$ are both subanalytic functions, then the sum of $f_1 + f_2$ is a subanalytic function if at least one of them map a bounded set to a bounded set or if both of them are nonnegative (Shiota, 1997, p.43).

Moreover, we still need the following important lemma from Bolte et al. (2007a), which shows that the subanalytic function is a KL function.

**Lemma 6 (Bolte et al., 2007a, Theorem 3.1)** Let $h : \mathbb{R}^p \to \mathbb{R} \cup \{-\infty\}$ be a subanalytic function with closed domain, and assume that $h$ is continuous on its domain, then $h$ is a KL function.

**Proof (Proof of Proposition 2)** We first verify the KL property of $\mathcal{L}$, then similarly show that of $\mathcal{L}$. From (2.5),

$$
\overline{L}(\{W_i\}_{i=1}^N, \{V_i\}_{i=1}^N, \{U_i\}_{i=1}^N) := \mathcal{R}_n(V_N; Y) + \sum_{i=1}^N r_i(W_i) + \sum_{i=1}^N s_i(V_i) + \frac{\gamma}{2} \sum_{i=1}^N \left( \|V_i - \sigma_i(U_i)\|_F^2 + \|U_i - W_i V_{i-1}\|_F^2 \right),
$$

which mainly includes the following types of functions, i.e.,

$$
\mathcal{R}_n(V_N; Y), \ r_i(W_i), \ s_i(V_i), \ \|V_i - \sigma_i(U_i)\|_F^2, \ \|U_i - W_i V_{i-1}\|_F^2.
$$

To verify the KL property of the function $\overline{L}$, we consider the above functions one by one under the hypothesis of Proposition 2.

- **On $\mathcal{R}_n(V_N; Y)$:** Note that given the output data $Y$, $\mathcal{R}_n(V_N; Y) := \frac{1}{n} \sum_{j=1}^n \ell((V_N)_j, y_j)$, where $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+ \cup \{0\}$ is some loss function. If $\ell$ is real analytic (resp. semialgebraic), then by Lemma 5 (resp. Lemma 4), $\mathcal{R}_n(V_N; Y)$ is real-analytic (resp. semialgebraic).

- **On $\|V_i - \sigma_i(U_i)\|_F^2$:** Note that $\|V_i - \sigma_i(U_i)\|_F^2$ is a finite sum of simple functions of the form $(v - \sigma_i(u))^2$ for any $u, v \in \mathbb{R}$. If $\sigma_i$ is real analytic (resp. semialgebraic), then $v - \sigma_i(u)$ is real analytic (resp. semialgebraic).
and further by Lemma 3 (resp. Lemma 4(2)), \( |v - \sigma_i(u)|^2 \) is also real analytic (resp. semialgebraic) since \( |v - \sigma_i(u)|^2 \) can be viewed as the composition \( g \circ h \) of these two functions where \( g(t) = t^2 \) and \( h(u, v) = v - \sigma_i(u) \).

On \( \| U_i - W_i V_{i-1} \|_F^2 \): Note that the function \( \| U_i - W_i V_{i-1} \|_F^2 \) is a polynomial function with the variables \( U_i, W_i \) and \( V_{i-1} \), and thus according to Krantz and Parks (2002) and Bochnak et al. (1998), it is both real analytic and semialgebraic.

On \( r_i(W_i), s_i(V_i) \): All \( r_i \)'s and \( s_i \)'s are real analytic or semialgebraic by the hypothesis of Proposition 2.

Since each part of the function \( \mathcal{Z} \) is either real analytic or semialgebraic, then by Lemma 3, \( \mathcal{Z} \) is a subanalytic function. Furthermore, by the continuity hypothesis of Proposition 2, \( \mathcal{Z} \) is continuous in its domain. Therefore, \( \mathcal{Z} \) is a KL function according to Lemma 6.

Similarly, we can verify the KL property of \( \mathcal{L} \) by checking each part is either real analytic or semialgebraic. The major task is to check the KL properties of the functions \( \| V_i - \sigma_i(W_i V_{i-1}) \|_F^2, (i = 1, \ldots, N) \). This reduces to check the function \( h : \mathbb{R} \times \mathbb{R}^{d_{i-1}} \times \mathbb{R}^{d_i} \to \mathbb{R}, h(u, v, w) := |u - \sigma_i(w^T v)|^2 \). Similar to the case \( |v - \sigma_i(u)|^2 \) for any \( u, v \in \mathbb{R} \) in \( \mathcal{Z} \), \( h \) is real analytic (resp. semialgebraic) if \( \sigma_i \) is real analytic (resp. semialgebraic) by Lemma 3 (resp. Lemma 4(2)). As a consequence, each part of the function \( \mathcal{L} \) is either real analytic or semialgebraic, so \( \mathcal{L} \) is a subanalytic function, and further by the continuity hypothesis of Proposition 2, \( \mathcal{L} \) is a KL function according to Lemma 6. This completes the proof.

\( \square \)

### C.2 Value convergence of BCD

We show the value convergence of both algorithms as follows.

**Theorem 4** Let \( \{Q^k\} := \{\{W_i^k\}_{i=1}^N, \{V_i^k\}_{i=1}^N\} \) and \( \{P^k\} := \{\{W_i^k\}_{i=1}^N, \{V_i^k\}_{i=1}^N, \{U_i^k\}_{i=1}^N\} \) be the sequences generated by Algorithms 1 and 2 respectively. Under Assumption 1 and finite initializations \( Q^0 \) and \( P^0 \), then for any positive \( \alpha \) and \( \gamma \), \( \{\mathcal{L}(Q^k)\} \) is nonincreasing and converges to some finite \( \mathcal{L}^* \) (resp. \( \mathcal{Z}(P^k) \)).

In order to prove Theorem 4, we first show the convergence of Algorithm 2 and then show that of Algorithm 1 similarly. We restate Lemma 1 precisely as follows.

**Lemma 7 (Restate of Lemma 1)** Let \( \{P^k\} \) be a sequence generated by the BCD method (Algorithm 2), then

\[
\mathcal{Z}(P^k) \leq \mathcal{Z}(P^{k-1}) - a\| P^k - P^{k-1} \|_F^2, \tag{C.3}
\]

where

\[
a := \min \left\{ \alpha \frac{\gamma}{2}, \alpha \frac{\gamma - L_R}{2} \right\} \tag{C.4}
\]

for the case that \( V_N \) is updated via the proximal strategy, or

\[
a := \min \left\{ \alpha \frac{\gamma}{2}, \alpha + \frac{\gamma - L_R}{2} \right\} \tag{C.5}
\]

for the case that \( V_N \) is update via the prox-linear strategy.

According to Algorithm 2, the decreasing property of the sequence \( \{\mathcal{Z}(P^k)\} \) is obvious. However, establishing the sufficient descent inequality \( (C.3) \) for the sequence \( \{\mathcal{Z}(P^k)\} \) is nontrivial. To achieve this, we should take advantage of the specific update strategies and also the form of \( \mathcal{Z} \) as shown in the following proofs.

**Proof** The descent quantity in \( (C.3) \) can be developed via considering the descent quantity along the update of each block variable. From Algorithm 2, each block variable is updated either by the proximal strategy with parameter \( \alpha/2 \) (say, updates of \( V_i \)) or by minimizing a strongly convex function \( (\ref{eq:strongly_convex}) \) with parameter \( \gamma > 0 \) (say, updates of \( U_N, U_i \)). Hence, we will consider both cases one by one.

\[\text{The function } h \text{ is called a strongly convex function with parameter } \gamma > 0 \text{ if } h(u) \geq h(v) + \langle \nabla h(v), u - v \rangle + \frac{\gamma}{2}\| u - v \|^2.\]
(a) **Proximal update case:** In this case, we take the $W_i^k$-update case for example. By Algorithm\[2\]

$W_i^k \leftarrow \arg\min_{W_i} \left\{ r_i(W_i) + \frac{\gamma}{2} \| U_i^k - W_i \|_2 \|^2 + \frac{\alpha}{2} \| W_i - W_i^{k-1} \|_F^2 \right\}. \quad (C.6)$

Let $h^k(W_i) = r_i(W_i) + \frac{\gamma}{2} \| U_i^k - W_i \|_2 \|^2$ and $\bar{h}^k(W_i) = r_i(W_i) + \frac{\gamma}{2} \| U_i^k - W_i \|_2 \|^2 + \frac{\alpha}{2} \| W_i - W_i^{k-1} \|_F^2$.

By the optimality of $W_i^k$, it holds

$$\bar{h}^k(W_i^{k-1}) \geq \bar{h}^k(W_i^k),$$

which implies

$$h^k(W_i^{k-1}) \geq h^k(W_i^k) + \frac{\alpha}{2} \| W_i^k - W_i^{k-1} \|_F^2. \quad (C.7)$$

Note that the $W_i^k$-update [C.6] is equivalent to the following original proximal BCD update, i.e.,

$$W_i^k \leftarrow \arg\min_{W_i} \bar{Z}(W_{<i}^k, W_i, W_{>i}^k, V_{<i}^k, V_{>i}^k, U_{<i}^k, U_{>i}^k, \alpha) + \frac{\alpha}{2} \| W_i^k - W_i^{k-1} \|_F^2,$$

where $W_{<i} := (W_1, W_2, \ldots, W_{i-1})$, $W_{>i} := (W_{i+1}, W_{i+2}, \ldots, W_N)$, and $V_{<i}, V_{>i}, U_{<i}, U_{>i}$ are defined similarly. Thus, by [C.7], we establish the descent part along the $W_i$-update $(i = 1, \ldots, N - 1)$, that is,

$$\bar{Z}(W_{<i}^k, W_i^k, V_{<i}^k, V_{>i}^k, U_{<i}^k, U_{>i}^k) \geq \bar{Z}(W_{<i}^{k-1}, W_i^{k-1}, V_{<i}^{k-1}, V_{>i}^{k-1}, U_{<i}^{k-1}, U_{>i}^{k-1}) + \frac{\alpha}{2} \| W_i^k - W_i^{k-1} \|_F^2. \quad (C.8)$$

Similarly, we can establish the similar descent estimates of [C.8] for the other blocks using the proximal updates including $V_i^k, \{U_i^k\}_{i=1}^{N-1}$ and $W_i^k$ blocks.

Specifically, for the $V_i^k$-block, the following holds

$$\bar{Z}(W_{<i}^k, W_i^{k-1}, V_{<i}^k, V_{>i}^k, U_{<i}^k, U_{>i}^k) \geq \bar{Z}(W_{<i}^{k-1}, W_i^{k-1}, V_{<i}^{k-1}, V_{>i}^{k-1}, U_{<i}^{k-1}, U_{>i}^{k-1}) + \frac{\alpha}{2} \| V_i^k - V_i^{k-1} \|_F^2. \quad (C.9)$$

For the $U_i^k$-block, $i = 1, \ldots, N - 1$, the following holds

$$\bar{Z}(W_{<i}^k, W_i^{k-1}, V_{<i}^k, V_{>i}^k, U_{<i}^k, U_{>i}^k) \geq \bar{Z}(W_{<i}^{k-1}, W_i^{k-1}, V_{<i}^{k-1}, V_{>i}^{k-1}, U_{<i}^{k-1}, U_{>i}^{k-1}) + \frac{\alpha + \gamma}{2} \| U_i^k - U_i^{k-1} \|_F^2. \quad (C.10)$$

For the $W_i^k$-block, the following holds

$$\bar{Z}(W_{<N}, W_i^k, V_{<N}, V_{>i}, U_{<N}, U_{>i}) \geq \bar{Z}(W_{<N}, W_i^{k-1}, V_{<N}, V_{>i}, U_{<N}, U_{>i}) + \frac{\alpha}{2} \| W_i^k - W_i^{k-1} \|_F^2. \quad (C.11)$$

(b) **Minimization of a strongly convex case:** In this case, we take $V_i^k$-update case for example. From Algorithm\[3\]

$$V_i^k \leftarrow \arg\min_{V_i} \left\{ s_i(V_i) + \frac{\gamma}{2} \| V_i - \sigma_i(U_i^{k-1}) \|_2 \|^2 + \frac{\gamma}{2} \| U_{i+1}^k - W_{i+1}^k \|_F^2 \right\}. \quad (C.12)$$

Let $h^k(V_i) = s_i(V_i) + \frac{\gamma}{2} \| V_i - \sigma_i(U_i^{k-1}) \|_2 \|^2 + \frac{\gamma}{2} \| U_{i+1}^k - W_{i+1}^k \|_F^2$. By the convexity of $s_i$, the function $h^k(V_i)$ is a strongly convex function with modulus no less than $\gamma$. By the optimality of $V_i^k$, it holds that

$$h^k(V_i^{k-1}) \geq h^k(V_i^k) + \frac{\gamma}{2} \| V_i^k - V_i^{k-1} \|_2^2. \quad (C.13)$$
Noting the relation between $h^k(V_i)$ and $\overline{Z}(W_{\leq i}^{k-1}, W_{\geq i}^{k-1}, V_{\leq i}, V_{\geq i}, U_{\leq i}^{k-1}, U_{\geq i}^{k-1})$, and by (C.13), it yields for $i = 1, \ldots, N - 1$,

$$\overline{Z}(W_{\leq i}^{k-1}, W_{\geq i}^{k-1}, V_{\leq i}, V_{\geq i}, U_{\leq i}^{k-1}, U_{\geq i}^{k-1}) \geq \overline{Z}(W_{\leq i}^{k-1}, W_{\leq i}^{k-1}, V_{\leq i}, V_{\geq i}, U_{\leq i}^{k-1}, U_{\geq i}^{k-1}) + \frac{\gamma}{2} \|V_i^k - V_{i}^{k-1}\|_F^2. \quad (C.14)$$

Similarly, we can establish the similar descent estimates for the $U_N^k$-block, that is,

$$\overline{Z}(W_{\leq N}^{k-1}, W_{\leq N}^{k-1}, V_{\leq N}^{k-1}, U_{\leq N}^{k-1}, U_N^{k-1}) \geq \overline{Z}(W_{\leq N}^{k-1}, W_{\leq N}^{k-1}, V_{\leq N}^{k-1}, U_{\leq N}^{k-1}, U_N^{k-1}) + \gamma \|U_N^k - U_{N}^{k-1}\|_F^2. \quad (C.15)$$

Summing (C.8)–(C.11) and (C.14)–(C.15) yields the descent inequality (C.3).

(c) **Prox-linear case for $V_N$, i.e., (3.2):** From (3.2), similarly, we let $h^k(V_N) := s_N(V_N) + R_n(V_N; Y) + \frac{\gamma}{2} \|V_N - U_N^{k-1}\|_F^2$ and $\overline{h}^k(V_N) = s_N(V_N) + R_n(V_N^{k-1}; Y) + \langle \nabla R_n(V_N^{k-1}; Y), V_N - V_N^{k-1} \rangle + \frac{\gamma}{2} \|V_N - V_N^{k-1}\|_F^2 + \frac{\gamma}{2} \|V_N - U_N^{k-1}\|_F^2$. By the optimality of $V_N^k$ and the strong convexity of $\overline{h}^k(V_N)$ with modulus at least $\alpha + \gamma$, it holds

$$\overline{h}^k(V_N^{k-1}) \geq \overline{h}^k(V_N^k) + \frac{\alpha + \gamma}{2} \|V_N^k - V_N^{k-1}\|_F^2. \quad (C.16)$$

After some simplifications and noting the relation between $h^k(V_N)$ and $\overline{h}^k(V_N)$, we have

$$h^k(V_N^{k-1}) \geq h^k(V_N^k) - \langle \nabla R_n(V_N^{k-1}; Y), V_N^{k-1} - V_N^{k-1} \rangle + \langle \nabla R_n(V_N^{k-1}; Y), V_N^{k-1} - V_N^{k-1} \rangle + \frac{\gamma}{2} \|V_N^k - V_N^{k-1}\|_F^2. \quad (C.16)$$

where the last inequality holds for the $L_R$-Lipschitz continuity of $\nabla R_n$, that is, the following inequality by Nesterov [2018],

$$R_n(V_N^k; Y) \leq R_n(V_N^{k-1}; Y) + \langle \nabla R_n(V_N^{k-1}; Y), V_N^k - V_N^{k-1} \rangle + \frac{L_R}{2} \|V_N^k - V_N^{k-1}\|_F^2. \quad (C.16)$$

Summing (C.8)–(C.11), (C.14)–(C.15) and (C.9) yields the descent inequality (C.3). \qed

**Proof (of Theorem 4)** By (C.3), $\overline{Z}(\mathcal{P}^k)$ is monotonically nonincreasing and lower bounded by 0 since each term of $\overline{Z}$ is nonnegative, thus, $\overline{Z}(\mathcal{P}^k)$ converges to some nonnegative, finite $\overline{Z}^*$. Similarly, we can show the claims in Theorem 4 holds for Algorithm 1. \qed

Based on Lemma 7, we can obtain the following corollary.

**Corollary 1 (Square summable)** The following hold:

(a) \( \sum_{k=1}^{\infty} \|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F^2 < \infty, \)

(b) \( \frac{1}{K} \sum_{k=1}^{K} \|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F^2 \to 0 \) at the rate of $\mathcal{O}(1/K)$, and

(c) \( \|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F \to 0 \) as $k \to \infty$.

**Proof** Summing (C.3) over $k$ from 1 to $\infty$ yields

$$\sum_{k=1}^{\infty} \|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F^2 \leq \overline{Z}(\mathcal{P}^0) < \infty,$$

which directly implies $\|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F \to 0$ as $k \to \infty$. Similarly, summing (C.3) over $k$ from 1 to $K$ yields

$$\frac{1}{K} \sum_{k=1}^{K} \|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F^2 \leq \frac{1}{K} \overline{Z}(\mathcal{P}^0),$$

which implies claim (b) of this corollary. \qed
C.3 Global convergence of BCD

Theorem 3 implies that the quality of the generated sequence is gradually improving during the iterative procedure in the sense of the descent of the objective value, and eventually achieves some level of objective value, then keeps stable. However, the convergence of the generated sequence \(\{Q^k\}_{k \in \mathbb{N}}\) (resp. \(\{P^k\}_{k \in \mathbb{N}}\)) itself is still unclear. In the following, we will show that under some natural conditions, the whole sequence converges to some critical point of the objective, and further if the initial point is sufficiently close to some global minimum, then the generated sequence can converge to this global minimum.

**Theorem 5 (Global convergence and rate)** Under assumptions of Theorem 1, the following hold

(a) \(\{Q^k\}_{k \in \mathbb{N}}\) (resp. \(\{P^k\}_{k \in \mathbb{N}}\)) converges to a critical point of \(\mathcal{L}\) (resp. \(\mathcal{Z}\)).

(b) If further the initialization \(Q^0\) (resp. \(P^0\)) is sufficiently close to some global minimum \(Q^*\) of \(\mathcal{L}\) (resp. \(P^*\) of \(\mathcal{Z}\)), then \(Q^k\) (resp. \(P^k\)) converges to \(Q^*\) (resp. \(P^*\)).

(c) Let \(\theta\) be the KL exponent of \(\mathcal{L}\) (resp. \(\mathcal{Z}\)) at \(Q^*\) (resp. \(P^*\)). There hold: (a) if \(\theta = 0\), then \(\{Q^k\}_{k \in \mathbb{N}}\) converges in a finite number of steps; (b) if \(\theta \in (0, \frac{1}{2}]\), then \(\|Q^k - Q^*\|_F \leq C \tau^k\) for all \(k \geq k_0\), for certain \(k_0 > 0, C > 0, \tau \in (0, 1)\); and (c) if \(\theta \in \left(\frac{1}{2}, 1\right]\), then \(\|Q^k - Q^*\|_F \leq C k^{-\frac{\theta - 1}{2 \theta}}\) for \(k \geq k_0\), for certain \(k_0 > 0, C > 0\). The same claims hold for the sequence \(\{P^k\}\).

(d) \(\frac{1}{K} \sum_{k=1}^{K} \|g^k\|_F^2 \to 0\) at the rate \(\mathcal{O}(1/K)\) where \(g^k \in \partial \mathcal{L}(Q^k)\). Similarly, \(\frac{1}{K} \sum_{k=1}^{K} \|g^k\|_F^2 \to 0\) at the rate \(\mathcal{O}(1/K)\) where \(g^k \in \partial \mathcal{L}(P^k)\).

Our proof is mainly based on the Kurdyka-Lojasiewicz framework established in \cite{Attouch2013}. Some other pioneer work can be also found in \cite{Attouch2010}. According to \cite{Attouch2013}, three key conditions including the sufficient decrease condition, relative error condition and continuity condition, together with the KL property are required to establish the global convergence of a descent algorithm from the subsequence convergence, where the sufficient decrease condition and the KL property have been established in Lemma 7 and Proposition 2 respectively, and the relative error condition is developed in Lemma 8 while the continuity condition holds naturally due to the continuity assumption. In the following, we first prove Theorem 5 under the subsequence convergence assumption, i.e., condition (a) of this theorem, then show that both condition (b) and condition (c) can imply the boundedness of the sequence (see Lemma 10), and thus the subsequence convergence as required in condition (a). The rate of convergence results follow the same argument as in the proof of \cite{Attouch2009} Theorem 2.

### C.3.1 Establishing relative error condition

We restate Lemma 2 precisely as follows.

**Lemma 8 (Restatement of Lemma 2)** Under conditions of Theorem 3, let \(B\) be an upper bound of \(P^{k-1}\) and \(P^k\) for any positive integer \(k\), \(L_B\) be a uniform Lipschitz constant of \(\sigma\) on the bounded set \(\{P : \|P\|_F \leq B\}\), and

\[
b := \max\{\gamma, \alpha + \gamma B, \alpha + \gamma L_B, \gamma B + 2\gamma B^2, 2\gamma B + \gamma B^2\},
\]

(or, for the prox-linear case, \(b := \max\{\gamma, \alpha + \gamma B, \alpha + \gamma L_B, \gamma B + 2\gamma B^2, 2\gamma B + \gamma B^2\}\)), then for any positive integer \(k\), there holds,

\[
dist(0, \partial \mathcal{L}(P^k)) \leq b \sum_{i=1}^{N} \|W_i^k - W_i^{k-1}\|_F + \|V_i^k - V_i^{k-1}\|_F + \|U_i^k - U_i^{k-1}\|_F \leq \tilde{b}\|P^k - P^{k-1}\|_F,
\]

where \(\tilde{b} := b \sqrt{3N}\), \(\text{dist}(0, S) := \inf_{s \in S} \|s\|_F\) for a set \(S\), and

\[
\partial \mathcal{L}(P^k) := (\{\partial_W \mathcal{L} \}_i^N, \{\partial_V \mathcal{L} \}_i^N, \{\partial_U \mathcal{L} \}_i^N)(P^k).
\]
Proof The inequality (C.18) is established via bounding each term of $\partial \mathcal{L}(\mathcal{P}^k)$. By the optimality conditions of all updates in Algorithm 4, the following hold

$$0 \in \partial s_N(V^k_N) + \partial r_N(V^k_N; Y) + \gamma(V^k_N - U^{k-1}_N) + \alpha(V^k_N - V^{k-1}_N),$$

(or for prox-linear, $0 \in \partial s_N(V^k_N) + \nabla r_N(V^{k-1}_N; Y) + \gamma(V^k_N - U^{k-1}_N) + \alpha(V^k_N - V^{k-1}_N),$

$$0 = \gamma(U^k_N - V^{k-1}_N) + \gamma(U^k_N - W^{k-1}_N V^{k-1}_{N-1}),$$

$$0 \in \partial r_N(W^k_N) + \gamma(W^k_N V^{k-1}_{N-1} - U^k_N) V^{k-1}_{N-1}^\top + \alpha(W^k_N - W^{k-1}_N),$$

for $i = N - 1, \ldots, 1$,

$$0 \in \partial s_i(V^k_i) + \gamma(V^k_i - \sigma_i(U^{i-1}_i)) + \gamma W^k_{i+1} \top (W^k_{i+1} V^k_i - U^k_i),$$

$$0 \in \gamma([\sigma_i(U^k_i) - V^k_i] \odot \sigma_i(U^k_i)) + \gamma(U^k_i - W^{k-1}_i V^{k-1}_i) + \alpha(U^k_i - U^{k-1}_i),$$

$$0 \in \partial r_i(W^k_i) + \gamma(W^k_i V^{k-1}_i - U^k_i) V^{k-1}_{i-1}^\top + \alpha(W^k_i - W^{k-1}_i),$$

where $V^k_0 \equiv V_0 = X$ for all $k$, and $\odot$ is the Hadamard product. By the above relations, we have

$$-\alpha(V^k_N - V^{k-1}_N) - \gamma(U^k_N - U^{k-1}_N) \in \partial s_N(V^k_N) + \partial r_N(V^k_N; Y) + \gamma(V^k_N - U^k_N) = \partial \mathcal{V}_N \mathcal{L}(\mathcal{P}^k),$$

$$-\gamma(W^k_N - W^{k-1}_N) V^{k-1}_{N-1} - \gamma W^{k-1}_N (V^{k-1}_{N-1} - V^{k-1}_{N-1}) = \gamma(U^k_N - V^k_N + (U^k_N - W^k_N) V^{k-1}_{N-1}) = \partial U_N \mathcal{L}(\mathcal{P}^k),$$

$$\gamma W^k_N [V^k_{N-1} (V^{k-1}_{N-1} - V^{k-1}_{N-1})^\top + (V^k_{N-1} - V^{k-1}_{N-1}) V^{k-1}_{N-1}^\top] - \gamma U^k_N (V^k_N - V^{k-1}_N)^\top - \alpha(W^k_N - W^{k-1}_N)$$

$$\in \partial r_N(W^k_N) + \gamma(W^k_N V^{k-1}_{N-1} - U^k_N) V^{k-1}_{N-1}^\top = \partial W_N \mathcal{L}(\mathcal{P}^k),$$

for $i = N - 1, \ldots, 1$,

$$-\gamma(\sigma_i(U^k_i) - \sigma_i(U^{i-1}_i)) \in \partial s_i(V^k_i) + \gamma(V^k_i - \sigma_i(U^{i-1}_i)) + \gamma W^k_{i+1} \top (W^k_{i+1} V^k_i - U^k_i) = \partial \mathcal{V}_i \mathcal{L}(\mathcal{P}^k),$$

$$-\gamma W^{k-1}_i [V^{k-1}_i - V^{k-1}_{i-1}] - \gamma(W^k_i - W^{k-1}_i) V^{k-1}_{i-1}^\top - \alpha(U^k_i - U^{k-1}_i)$$

$$\in \gamma ([\sigma_i(U^k_i) - V^k_i] \odot \sigma_i(U^k_i)) + \gamma(U^k_i - W^k_i) V^{k-1}_{i-1} = \partial U_i \mathcal{L}(\mathcal{P}^k),$$

$$\gamma W^k_i [V^{k-1}_{i-1} (V^{k-1}_{i-1} - V^{k-1}_{i-1})^\top + (V^{k-1}_{i-1} - V^{k-1}_{i-1}) V^{k-1}_{i-1}^\top] - \gamma U^k_i (V^k_i - V^{k-1}_{i-1})^\top - \alpha(W^k_i - W^{k-1}_i)$$

$$\in \partial r_i(W^k_i) + \gamma(W^k_i V^{k-1}_{i-1} - U^k_i) V^{k-1}_{i-1}^\top = \partial W_i \mathcal{L}(\mathcal{P}^k).$$

Based on the above relations, and by the Lipschitz continuity of the activation function on the bounded set \{\mathcal{P} : \|\mathcal{P}\|_F \leq B\} and the bounded assumption of both $\mathcal{P}^{k-1}$ and $\mathcal{P}^k$, we have

$$\|G_{V_N}^k\|_F \leq \alpha \|V^k_N - V^{k-1}_N\|_F + \gamma \|U^k_N - U^{k-1}_N\|_F,$$

$$\|G_{U_N}^k\|_F \leq (L_R + \alpha) \|V^k_N - V^{k-1}_N\|_F + \|U^k_N - U^{k-1}_N\|_F,$$

$$\|G_{W_N}^k\|_F \leq 2\beta^2 \|V^{k-1}_{N-1} - V^{k-1}_{N-1}\|_F + \gamma \|V^k_N - V^{k-1}_N\|_F + \alpha \|W^k_N - W^{k-1}_N\|_F,$$

and for $i = N - 1, \ldots, 1$,

$$\|G_{V_i}^k\|_F \leq \alpha \|U^k_i - U^{k-1}_i\|_F,$$

$$\|G_{U_i}^k\|_F \leq \alpha \|U^k_i - U^{k-1}_i\|_F + \gamma \|W^k_i - W^{k-1}_i\|_F + \alpha \|U^k_i - U^{k-1}_i\|_F,$$

$$\|G_{W_i}^k\|_F \leq (\gamma \beta^2 + \gamma \beta) \|V^k_i - V^{k-1}_i\|_F + \alpha \|W^k_i - W^{k-1}_i\|_F,$$

and

Summing the above inequalities and after some simplifications, we obtain (C.18).
C.3.2 Proof of Theorem 5 under condition (a)

Based on Theorem 4 and under the hypothesis that $\mathcal{L}$ is continuous on its domain and there exists a convergent subsequence (i.e., condition (a)), the continuity condition required in [Attouch et al.] (2013) holds naturally, that is, there exists a subsequence $\{P^k\}_{j \in \mathbb{N}}$ and $P^*$ such that

$$P^{k_j} \to P^* \quad \text{and} \quad \mathcal{L}(P^{k_j}) \to \mathcal{L}(P^*), \quad \text{as} \quad j \to \infty.$$  \hfill (C.19)

Based on Lemmas 7 and 8 and (C.19), we can justify the global convergence of $P^k$ stated in Theorem 5 following the proof idea of Attouch et al. (2013). For the completeness of the proof, we still present the detailed proof as follows.

Before presenting the main proof, we establish a local convergence result of $P^k$, that is, the convergence of $P^k$ when $P^0$ is sufficiently close to some point $P^*$. Specifically, let $(\varphi, \eta, U)$ be the associated parameters of the KL property of $\mathcal{L}$ at $P^*$, where $\varphi$ is a continuous concave function, $\eta$ is a positive constant, and $U$ is a neighborhood of $P^*$. Let $\rho$ be some constant such that $\mathcal{N}(P^*, \rho) := \{P : \|P - P^*\|_F \leq \rho\} \subset U$, $B := \rho + \|P^*\|_F$, and $L_B$ be the uniform Lipschitz constant for $\sigma_i, i = 1, \ldots, N - 1$, within $\mathcal{N}(P^*, \rho)$. Assume that $P^0$ satisfies the following condition

$$\frac{\bar{b}}{a} \varphi(\mathcal{L}(P^0)) + 3 \sqrt{\frac{\mathcal{L}(P^0)}{a}} + \|P^0 - P^*\|_F < \rho,$$  \hfill (C.20)

where $\bar{b} = b/\sqrt{3N}$, $b$ and $a$ are defined in (C.17) and (C.4), respectively.

**Lemma 9 (Local convergence)** Under the conditions of Theorem 5, suppose that $P^0$ satisfies the condition (C.20), and $\mathcal{L}(P^k) > \mathcal{L}(P^*)$ for $k \in \mathbb{N}$, then

$$\sum_{i=1}^{k} \|P^i - P^{i-1}\|_F \leq 2 \sqrt{\frac{\mathcal{L}(P^0)}{a}} + \frac{\bar{b}}{a} \varphi(\mathcal{L}(P^0)) - \varphi(\mathcal{L}(P^*)) + \sqrt{\frac{\mathcal{L}(P^0)}{a}} + \|P^0 - P^*\|_F < \rho,$$  \hfill (C.21)

As $k$ goes to infinity, (C.21) yields

$$\sum_{i=1}^{\infty} \|P^i - P^{i-1}\|_F < \infty,$$

which implies the convergence of $\{P^k\}_{k \in \mathbb{N}}$.

**PROOF** We will prove $P^k \in \mathcal{N}(P^*, \rho)$ by induction on $k$. It is obvious that $P^0 \in \mathcal{N}(P^*, \rho)$. Thus, (C.22) holds for $k = 0$. For $k = 1$, we have from (C.3) and the nonnegativeness of $\{\mathcal{L}(P^k)\}_{k \in \mathbb{N}}$ that

$$\mathcal{L}(P^0) \geq \mathcal{L}(P^0) - \mathcal{L}(P^1) \geq a\|P^0 - P^1\|_F^2,$$

which implies $\|P^0 - P^1\|_F \leq \sqrt{\frac{\mathcal{L}(P^0)}{a}}$. Therefore,

$$\|P^1 - P^*\|_F \leq \|P^0 - P^1\|_F + \|P^0 - P^*\|_F \leq \sqrt{\frac{\mathcal{L}(P^0)}{a}} + \|P^0 - P^*\|_F,$$

which indicates $P^1 \in \mathcal{N}(P^*, \rho)$.

Suppose that $P^k \in \mathcal{N}(P^*, \rho)$ for $0 \leq k \leq K$. We proceed to show that $P^{K+1} \in \mathcal{N}(P^*, \rho)$. Since $P^k \in \mathcal{N}(P^*, \rho)$ for $0 \leq k \leq K$, it implies that $\|P^k\|_F \leq B := \rho + P^*$ for $0 \leq k \leq K$. Thus, by Lemma 8 for $1 \leq k \leq K$,

$$\text{dist}(0, \partial \mathcal{L}(P^k)) \leq \|P^k - P^{k-1}\|_F,$$

which together with the KL inequality (C.2) yields

$$\frac{1}{\varphi'(\mathcal{L}(P^k) - \mathcal{L}(P^*))} \leq \frac{\bar{b}}{\eta} \|P^k - P^{k-1}\|_F,$$  \hfill (C.23)
By (C.3), the above inequality and the concavity of $\varphi$, for $k \geq 2$, it holds

$$a\|P^k - P^{k-1}\|_F^2 \leq \mathcal{L}(P^{k-1}) - \mathcal{L}(P^k) = (\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - (\mathcal{L}(P^k) - \mathcal{L}(P^*))$$

$$\leq \varphi(\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - \varphi(\mathcal{L}(P^k) - \mathcal{L}(P^*))$$

$$\leq \frac{\beta}{a} \|P^k - P^{k-2}\|_F \cdot [\varphi(\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - \varphi(\mathcal{L}(P^k) - \mathcal{L}(P^*))],$$

which implies

$$\|P^k - P^{k-1}\|_F^2 \leq \|P^k - P^{k-2}\|_F \cdot \frac{\beta}{a} [\varphi(\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - \varphi(\mathcal{L}(P^k) - \mathcal{L}(P^*))].$$

Taking the square root on both sides and using the inequality $2\sqrt{\alpha\beta} \leq \alpha + \beta$, the above inequality implies

$$2\|P^k - P^{k-1}\|_F \leq \|P^k - P^{k-2}\|_F + \frac{\beta}{\alpha} [\varphi(\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - \varphi(\mathcal{L}(P^k) - \mathcal{L}(P^*))].$$

Summing the above inequality over $k$ from 2 to $K$ and adding $\|P^1 - P^0\|_F$ to both sides, it yields

$$\|P^K - P^{K-1}\|_F + \sum_{k=1}^K \|P^k - P^{k-1}\|_F \leq 2\|P^1 - P^0\|_F + \frac{\beta}{\alpha} \sum_{k=1}^K [\varphi(\mathcal{L}(P^{k-1}) - \mathcal{L}(P^*)) - \varphi(\mathcal{L}(P^k) - \mathcal{L}(P^*))]$$

which implies

$$\sum_{k=1}^K \|P^k - P^{k-1}\|_F \leq 2\sqrt{\frac{\mathcal{L}(P^0)}{\alpha}} + \frac{\beta}{\alpha} \varphi(\mathcal{L}(P^0) - \mathcal{L}(P^*)), \quad (C.24)$$

and further,

$$\|P^{K+1} - P^*\|_F \leq \|P^{K+1} - P^K\|_F + \sum_{k=1}^K \|P^k - P^{k-1}\|_F + \|P^0 - P^*\|_F$$

$$\leq \sqrt{\frac{\mathcal{L}(P^0) - \mathcal{L}(P^{K+1})}{\alpha}} + 2\sqrt{\frac{\mathcal{L}(P^0)}{\alpha}} + \frac{\beta}{\alpha} \varphi(\mathcal{L}(P^0) - \mathcal{L}(P^*)) + \|P^0 - P^*\|_F$$

$$\leq 3\sqrt{\frac{\mathcal{L}(P^0)}{\alpha}} + \frac{\beta}{\alpha} \varphi(\mathcal{L}(P^0) - \mathcal{L}(P^*)) + \|P^0 - P^*\|_F < \rho,$$

where the second inequality holds for (C.3) and (C.24), the third inequality holds for $\mathcal{L}(P^k) - \mathcal{L}(P^{K+1}) \leq \mathcal{L}(P^K) \leq \mathcal{L}(P^0)$. Thus, $P^{K+1} \in \mathcal{N}(P^*, \rho)$. Therefore, we prove this lemma. $\square$

**Proof (Proof of Theorem 5)** We prove the whole sequence convergence stated in Theorem 5 according to the following two cases.

**Case 1:** $\mathcal{L}(P^{k_0}) = \mathcal{L}(P^*)$ at some $k_0$. In this case, by Lemma 7 it holds $P^k = P^{k_0} = P^*$ for all $k \geq k_0$, which implies the convergence of $P^k$ to a limit point $P^*$.

**Case 2:** $\mathcal{L}(P^k) > \mathcal{L}(P^*)$ for all $k \in \mathbb{N}$. In this case, since $P^*$ is a limit point and $\mathcal{L}(P^k) \to \mathcal{L}(P^*)$, by Theorem 4 there must exist an integer $k_0$ such that $P^{k_0}$ is sufficiently close to $P^*$ as required in Lemma 9 (see the inequality (C.20)). Therefore, the whole sequence $\{P^k\}_{k \in \mathbb{N}}$ converges according to Lemma 9. Since $P^*$ is a limit point of $\{P^k\}_{k \in \mathbb{N}}$, we have $P^k \to P^*$.

Next, we show $P^*$ is a critical point of $\mathcal{L}$. By Corollary 1(c), $\lim_{k \to \infty} \|P^k - P^{k-1}\|_F = 0$. Furthermore, by Lemma 8

$$\lim_{k \to \infty} \text{dist}(0, \partial \mathcal{L}(P^k)) = 0,$$

which implies that any limit point is a critical point. Therefore, we prove the global convergence of the sequence generated by Algorithm 2.

The convergence to a global minimum is a straightforward variant of Lemma 9.
The $O(1/k)$ rate of convergence is a direct claim according to the proof of Lemma 8 and Corollary 1(e). The proof of the convergence of Algorithm 1 is similar to that of Algorithm 2. We give a brief description about this. Note that in Algorithm 1, all blocks of variables are updated via the proximal strategies (or, prox-linear strategy for $V_{N}$-block). Thus, it is easy to show the similar descent inequality, i.e.,

$$\mathcal{L}(Q_{k-1}) - \mathcal{L}(Q_k) \geq a \|Q_k - Q^{k-1}\|_{F}^{2},$$

for some $a > 0$. Then similar to the proof of Lemma 8, we can establish the following inequality via checking the optimality conditions of all subproblems in Algorithm 1, that is,

$$\text{dist}(0, \partial \mathcal{L}(Q_k)) \leq b \|Q_k - Q^{k-1}\|_{F},$$

for some $b > 0$. By (C.25), (C.26) and the KL property of $\mathcal{L}$ (by Proposition 2), the global convergence of Algorithm 1 can be proved via a similar proof procedure of Algorithm 2.

C.3.3 Condition (b) or (c) implies condition (a)

**Lemma 10** Under condition (b) or (c) of Theorem 5, $P^k$ is bounded for any $k \in \mathbb{N}$, and thus, there exists a convergent subsequence.

**Proof** We first show the boundedness of the sequence as well as the subsequence convergence under condition (b) of Theorem 5 then under condition (c) of Theorem 5.

1. **Condition (b) implies condition (a):** We first establish the boundedness of $W^k_i$, $i = 1, \ldots, N$, then recursively, we establish the boundedness of $U^k_i$ via the boundedness of $W^k_i$ and $V^k_{i-1}$ (noting that $V^k_0 \equiv X$), followed by that of $V^k_i$ via the boundedness of $U^k_i$ from $i = 1$ to $N$.

   (1) Boundedness of $W^k_i$ ($i = 1, \ldots, N$): By Lemma 7, $\mathcal{Z}(P^k) < \infty$ for all $k \in \mathbb{N}$. Noting that each term of $\mathcal{Z}$ is nonnegative, thus, $0 \leq r_i(W^k_i) < \infty$ for any $k \in \mathbb{N}$ and $i = 1, \ldots, N$. By the coercivity of $r_i$, $W^k_i$ is bounded for any $k \in \mathbb{N}$ and $i = 1, \ldots, N$.

   In the following, we establish the boundedness of $U^k_i$ for any $k \in \mathbb{N}$ and $i = 1, \ldots, N$.

   (2) $i = 1$: Since $\mathcal{Z}(P^k) < \infty$, then $\|U^k_1 - W^k_1 X\|_{F}^2 < \infty$ for any $k \in \mathbb{N}$. By the boundedness of $W^k_1$ and the coercivity of the function $\|\cdot\|_{F}^2$, we have the boundedness of $U^k_1$ for any $k \in \mathbb{N}$. Then we show the boundedness of $V^k_1$ by the boundedness of $U^k_1$. Due to $\mathcal{Z}(P^k) < \infty$, then $\|V^k_1 - \sigma_1(U^k_1)\|_{F}^2 < \infty$ for any $k \in \mathbb{N}$. By the Lipschitz continuity of $\sigma_1$ and the boundedness of $U^k_1$, $\sigma_1(U^k_1)$ is uniformly bounded for any $k \in \mathbb{N}$. Thus, by the coercivity of $\|\cdot\|_{F}^2$, $V^k_1$ is bounded for any $k \in \mathbb{N}$.

   (3) $i > 1$: Recursively, we show that the boundedness of $W^k_i$ and $V^k_{i-1}$ implies the boundedness of $U^k_i$, and then the boundedness of $V^k_i$ from $i = 2$ to $N$.

   Now, we assume that the boundedness of $V^k_{i-1}$ has been established. Similar to (2), the boundedness of $U^k_i$ is guaranteed by $\|U^k_i - W^k_i V^k_{i-1}\|_{F}^2 < \infty$ and the boundedness of $W^k_i$ and $V^k_{i-1}$, and the boundedness of $V^k_i$ is guaranteed by $\|V^k_i - \sigma_i(U^k_i)\|_{F}^2 < \infty$ and the boundedness of $U^k_i$, as well as the Lipschitz continuity of $\sigma_i$.

   As a consequence, we prove the boundedness of $\{P^k\}_{k \in \mathbb{N}}$ under condition (b), which implies the subsequence convergent.

2. **Condition (c) implies condition (a):** By Lemma 7 and the finite initialization assumption, we have

$$\mathcal{Z}(P^k) \leq \mathcal{Z}(P^0) < \infty,$$

which implies the boundedness of $P^k$ due to the coercivity of $\mathcal{Z}$ (i.e., condition (c)), and thus, there exists a convergent subsequence.

This completes the proof of this lemma. □
D Proof of Theorem 3

The proof of Theorem 3 is very similar to those of Lemma 7 and Theorem 5 via noting that the updates are slightly different. In the following, we present the proof of Theorem 3.

Lemma 7 still holds for Algorithm 3 via replacing $L$ with $L_{res}$, which is stated as the following lemma.

**Lemma 11** Let $\{(W_i^k, V_i^k, U_i^k)_{i=1}^N\}_{k \in \mathbb{N}}$ be a sequence generated by the BCD method (Algorithm 3) for the DNN training with ResNets, then for any $\gamma > 0, \alpha > 0$,

$$
\mathcal{Z}_{res}\left(\{W_i^k, V_i^k, U_i^k\}_{i=1}^N\right) \leq \mathcal{Z}_{res}\left(\{W_i^{k-1}, V_i^{k-1}, U_i^{k-1}\}_{i=1}^N\right) - \alpha \sum_{i=1}^N \left[\|W_i^k - W_i^{k-1}\|_F^2 + \|V_i^k - V_i^{k-1}\|_F^2 + \|U_i^k - U_i^{k-1}\|_F^2\right],
$$

(D.1)

and $\{\mathcal{Z}_{res}\left(\{W_i^k, V_i^k, U_i^k\}_{i=1}^N\right)\}_{k \in \mathbb{N}}$ converges to some $\mathcal{Z}_{res}^*$, where $a := \min \left\{\frac{\gamma}{2}, \frac{\alpha}{2}\right\}$.

**Proof** The proof of this lemma is the same as that of Lemma 7.

In the ResNets case, Lemma 8 should be revised as the following lemma.

**Lemma 12** Let $\{(W_i^k, V_i^k, U_i^k)_{i=1}^N\}_{k \in \mathbb{N}}$ be a sequence generated by Algorithm 3. Under Assumptions of Theorem 3 let $\hat{b} := \max\{\alpha + \gamma L, \alpha + \gamma B, 2\gamma (1 + B^2), \gamma (1 + LB + 2B^2)\}$, then

$$
dist(0, \partial \mathcal{Z}_{res}(\{W_i^k, V_i^k, U_i^k\}_{i=1}^N)) \leq \hat{b} \sum_{i=1}^N \left[\|W_i^k - W_i^{k-1}\|_F + \|V_i^k - V_i^{k-1}\|_F + \|U_i^k - U_i^{k-1}\|_F\right],
$$

(D.2)

where

$$
\partial \mathcal{Z}_{res}(\{W_i^k, V_i^k, U_i^k\}_{i=1}^N) := (\partial W_i^k, \partial L_{res}, \partial V_i, \partial L_{res}, \partial U_i, \partial L_{res})_{i=1}^N(\{W_i^k, V_i^k, U_i^k\}_{i=1}^N).
$$

**Proof** From updates of Algorithm 3

$0 \in \partial s_N(V_N^k) + \partial r_{\alpha}(V_N^k; Y) + \gamma (V_N^k - V_{N-1}^k - U_N^{k-1}) + \alpha (V_N^k - V_{N-1}^k),$

$0 = \gamma (U_N^k + V_{N-1}^k - U_N^{k-1}) + \alpha (U_N^k - W_{N-1}^{k-1}V_{N-1}^{k-1}),$

$0 \in \partial r_N(W_N^k) + \gamma (W_N^kV_{N-1}^{k-1} - U_N^k)V_{N-1}^{k-1} + \alpha (W_N^k - W_{N-1}^k),$

for $i = N - 1, \ldots, 1$.

$0 \in \partial s_i(V_i^k) + \gamma (V_i^k - V_{i-1}^k - \sigma_i(U_{i-1}^{k-1})) - \gamma (V_{i+1}^k - V_i^k - \sigma_{i+1}(U_{i+1}^{k-1})) + \gamma W_{i+1}^k (W_{i+1}^k V_i^k - U_{i+1}^k),$

$0 \in \gamma [\sigma_i(U_i^k) + V_{i-1}^{k-1} - V_i^k] + \partial \sigma_i(U_{i-1}^{k-1}) + \gamma (U_i^k - W_i^kV_{i-1}^{k-1}) + \alpha (U_i^k - U_{i-1}^k),$

$0 \in \partial r_i(W_i^k) + \gamma (W_i^kV_{i-1}^{k-1} - U_i^k)V_{i-1}^{k-1} + \alpha (W_i^k - W_{i-1}^k).$
where \( V_0^k \equiv V_0 = X \), for all \( k \), and \( \odot \) is the Hadamard product. By the above relations, we have

- \( \alpha(V_N^k - V_N^{k-1}) - \gamma(V_N^k - V_N^{k-1}) - \gamma(U_N^k - U_N^{k-1}) \)
- \( \in \partial s_N(V_N^k) + \partial R_n(V_N^k, Y) + \gamma(V_N^k - V_N^{k-1} - U_N^k) = \partial V_N \mathcal{L}_{\text{res}}(\{ W_i^k, V_i^k, U_i^k \}_{i=1}^N) \),

From \( \alpha \), \( \gamma \), \( \in \), \( \gamma \) are adopted for the \( k \), \( \gamma \) of the \( \gamma \) property of \( \gamma \), and the \( \gamma \) property of \( \gamma \), we can prove this corollary by Attouch et al. (2013, Theorem 2.9).

For \( i = N - 1, \ldots, 1 \),

- \( \gamma(V_{i-1}^k - V_{i-1}^{k-1}) - \gamma(U_{i-1}^k) \)
- \( \in \partial s_i(V_i^k) + \gamma(V_i^k - V_{i-1}^k) - \gamma(V_i^k - V_{i+1}^k) = \partial V_i \mathcal{L}_{\text{res}}(\{ W_i^k, V_i^k, U_i^k \}_{i=1}^N) \)

From the above relations, the uniform boundedness of the generated sequence (where its bound is \( B \)) and the Lipschitz continuity of the activation function by the hypothesis of this lemma, then \( \| \xi_i^k \| \leq L B \), and further we get (D.2).

\[ \Box \]

**Proof (Proof of Theorem 3)** The proof of this theorem is very similar to that of Theorem 5. First, similar to Proposition 2, it is easy to show that \( \mathcal{L}_{\text{res}} \) is also a KL function. Then based on Lemma 11 and Lemma 12 and the KL property of \( \mathcal{L}_{\text{res}} \), we can prove this corollary by Attouch et al. (2013, Theorem 2.9).

The other claims of this theorem follow from the same proof of Theorem 5. When the prox-linear strategy is adopted for the \( V_N \)-update, the claims of Theorem 3 can be proved via following the same proof of Theorem 2.

\[ \Box \]

### E Closed form solutions of some subproblems

In this section, we provide the closed form solutions to the ReLU involved subproblem and hinge loss involved subproblem.

#### E.1 Closed form solution to ReLU-subproblem

From Algorithm 2 when \( \sigma_i \) is ReLU, then the \( U_i^k \)-update actually reduces to the following one-dimensional minimization problem,

\[
u^* = \arg \min_u f(u) := \frac{1}{2}(\sigma(u) - a)^2 + \frac{\gamma}{2}(u - b)^2,
\]

where \( \sigma(u) = \max\{0, u\} \) and \( \gamma > 0 \). The solution to the above one-dimensional minimization problem can be represented in the following lemma.
Lemma 13 \textit{The optimal solution to Problem (E.1) is shown as follows.}

\[
\text{prox}_{\frac{1}{2}(\sigma(\cdot)-a)^2}(b) = \begin{cases} 
\frac{a + \gamma b}{1 + \gamma}, & \text{if } a + \gamma b \geq 0, \ b \geq 0, \\
\frac{a + \gamma b}{1 + \gamma}, & \text{if } -(\sqrt{\gamma(\gamma+1)} - \gamma)a \leq \gamma b < 0, \\
b, & \text{if } a \leq \gamma b \leq -\sqrt{\gamma(\gamma+1)} - \gamma a < 0, \\
\min\{b, 0\}, & \text{if } a + \gamma b < 0.
\end{cases}
\]

\textbf{Proof} In the following, we divide this into two cases.

(a) \(u \geq 0\): In this case,

\[
f(u) = \frac{1}{2}(u - a)^2 + \frac{\gamma}{2}(u - b)^2.
\]

It is easy to check that

\[
u^* = \begin{cases} 
\frac{a + \gamma b}{1 + \gamma}, & \text{if } a + \gamma b \geq 0 \\
0, & \text{if } a + \gamma b < 0
\end{cases}
\] (E.2)

and

\[
f\left(\frac{a + \gamma b}{1 + \gamma}\right) = \frac{\gamma}{2(1 + \gamma)}(b - a)^2, \quad f(0) = \frac{1}{2}a^2 + \frac{\gamma}{2}b^2.
\]

(b) \(u < 0\): In this case,

\[
f(u) = \frac{1}{2}a^2 + \frac{\gamma}{2}(u - b)^2.
\]

It is easy to check that

\[
u^* = \begin{cases} 
0, & \text{if } b \geq 0 \\
b, & \text{if } b < 0
\end{cases}
\] (E.3)

and

\[
f(b) = \frac{1}{2}a^2, \quad f(0) = \frac{1}{2}a^2 + \frac{\gamma}{2}b^2.
\]

Based on (E.2) and (E.3), we obtain the solution to Problem (E.1) by considering the following four cases.

1. \(a + \gamma b \geq 0, \ b \geq 0\): In this case, we need to compare the values \(f\left(\frac{a + \gamma b}{1 + \gamma}\right) = \frac{\gamma}{2(1 + \gamma)}(b - a)^2\) and \(f(0) = \frac{1}{2}a^2 + \frac{\gamma}{2}b^2\). It is obvious that

\[
u^* = \frac{a + \gamma b}{1 + \gamma}.
\]

2. \(a + \gamma b \geq 0, \ b < 0\): In this case, we need to compare the values \(f\left(\frac{a + \gamma b}{1 + \gamma}\right) = \frac{\gamma}{2(1 + \gamma)}(b - a)^2\) and \(f(b) = \frac{1}{2}a^2\).

By the hypothesis of this case, it is obvious that \(a > 0\). We can easily check that

\[
u^* = \begin{cases} 
\frac{a + \gamma b}{1 + \gamma}, & \text{if } -(\sqrt{\gamma(\gamma+1)} - \gamma)a \leq \gamma b < 0, \\
b, & \text{if } a \leq \gamma b \leq -\sqrt{\gamma(\gamma+1)} - \gamma a < 0.
\end{cases}
\]

3. \(a + \gamma b < 0, \ b \geq 0\): It is obvious that

\[
u^* = 0.
\]

4. \(a + \gamma b < 0, \ b < 0\): It is obvious that

\[
u^* = b.
\]
Thus, the solution to Problem (E.1) is
\[
\text{prox}_{\frac{1}{\gamma}(\cdot - a)^2}(b) = \begin{cases} 
\frac{a + \gamma b}{1 + \gamma}, & \text{if } a + \gamma b \geq 0, \ b \geq 0, \\
\frac{a + \gamma b}{1 + \gamma}, & \text{if } -(\sqrt{\gamma(\gamma + 1)} - \gamma)a \leq \gamma b < 0, \\
b, & \text{if } -a \leq \gamma b \leq -(\sqrt{\gamma(\gamma + 1)} - \gamma)a < 0, \\
\min\{b, 0\}, & \text{if } a + \gamma b < 0.
\end{cases}
\]

E.2 The closed form of the proximal operator of hinge loss

Consider the following optimization problem
\[
u^* = \arg\min_u g(u) := \max\{0, 1 - a \cdot u\} + \frac{\gamma}{2}(u - b)^2, \quad (E.4)
\]
where \(\gamma > 0\).

**Lemma 14** The optimal solution to Problem (E.4) is shown as follows
\[
hinge_\gamma(a, b) = \begin{cases} 
b, & \text{if } a = 0, \\
b + \gamma^{-1} a, & \text{if } a \neq 0 \text{ and } ab \leq 1 - \gamma^{-1} a^2, \\
a^{-1}, & \text{if } a \neq 0 \text{ and } 1 - \gamma^{-1} a^2 < ab < 1, \\
b, & \text{if } a \neq 0 \text{ and } ab \geq 1.
\end{cases}
\]

**Proof** We consider the problem in the following three different cases: (1) \(a > 0\), (2) \(a = 0\) and (3) \(a < 0\).

(1) \(a > 0\): In this case,
\[
g(u) = \begin{cases} 
1 - au + \frac{\gamma}{2}(u - b)^2, & \text{if } u < a^{-1}, \\
\frac{\gamma}{2}(u - b)^2, & \text{if } u \geq a^{-1}.
\end{cases}
\]

It is easy to show that the solution to the problem is
\[
u^* = \begin{cases} 
b + \gamma^{-1} a, & \text{if } a > 0 \text{ and } b \leq a^{-1} - \gamma^{-1} a, \\
a^{-1}, & \text{if } a > 0 \text{ and } a^{-1} - \gamma^{-1} a < b < a^{-1}, \\
b, & \text{if } a > 0 \text{ and } b \geq a^{-1}.
\end{cases} \quad (E.5)
\]

(2) \(a = 0\): It is obvious that
\[
u^* = b. \quad (E.6)
\]

(3) \(a < 0\): Similar to (1),
\[
g(u) = \begin{cases} 
1 - au + \frac{\gamma}{2}(u - b)^2, & \text{if } u \geq a^{-1}, \\
\frac{\gamma}{2}(u - b)^2, & \text{if } u < a^{-1}.
\end{cases}
\]

Similarly, it is easy to show that the solution to the problem is
\[
u^* = \begin{cases} 
b + \gamma^{-1} a, & \text{if } a < 0 \text{ and } b \geq a^{-1} - \gamma^{-1} a, \\
a^{-1}, & \text{if } a < 0 \text{ and } a^{-1} < b < a^{-1} - \gamma^{-1} a, \\
b, & \text{if } a < 0 \text{ and } b \leq a^{-1}.
\end{cases} \quad (E.7)
\]

Thus, we finish the proof of this lemma. □
BCD vs. SGD for training ten-hidden-layer MLPs

In this experiment, we attempt to verify the capability of BCD for training MLPs with many layers. Specifically, we consider the DNN training model (2.2) with ReLU activation, the squared loss, and the network architecture being an MLPs with ten hidden layers, on the MNIST data set. The specific settings were summarized as follows:

(a) For the MNIST data set, we implemented a $784-(600 \times 10)-10$ MLPs (that is, the input dimension $d_0 = 28 \times 28 = 784$, the output dimension $d_{11} = 10$, and the numbers of hidden units are all 600), and set $\gamma = \alpha = 1$ for BCD. The sizes of training and test samples are 60000 and 10000, respectively.

(b) The learning rate of SGD is 0.001 (a very conservative learning rate to see if SGD can train the DNNs). More greedy learning rates such as 0.01 and 0.05 have also been used, and similar failure of training is also observed.

(c) For each experiment, we used the same mini-batch sizes (512) and initializations for all algorithms. Specifically, all the weights $\{W_i\}_{i=1}^N$ are initialized from a Gaussian distribution with a standard deviation of 0.01 and the bias vectors are initialized as vectors of all 0.1, while the auxiliary variables $\{U_i\}_{i=1}^N$ and state variables $\{V_i\}_{i=1}^N$ are initialized by a single forward pass.

Under these settings, we plot the curves of training accuracy (acc.) and test accuracy (acc.) of BCD and SGD as shown in Figure 1. According to Figure 1, vanilla SGD usually fails to train such deeper MLPs since it suffers from the vanishing gradient issue (Goodfellow et al., 2016), whereas BCD still works and achieves a moderate accuracy within a few epochs.

References

A. Askari, G. Negiar, R. Sambharya, and L. E. Ghaoui. Lifted neural networks. arXiv preprint arXiv:1805.01532, 2018.

H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Mathematical Programming, 116:5–16, 2009.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality. Mathematics of Operations Research, 35(2):438–457, 2010.

H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. Mathematical Programming, 137:91–129, 2013.

J. Bochnak, M. Coste, and M.-F. Roy. Real algebraic geometry, volume 3. Ergeb. Math. Grenzgeb. Springer-Verlag, Berlin, 1998.

J. Bolte, A. Daniilidis, and A. Lewis. The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization, 17:1205–1223, 2007a.

J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clark subgradients of stratifiable functions. SIAM Journal on Optimization, 18:556–572, 2007b.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 146(1):459–494, 2014.

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. In Foundations and Trends® in Machine Learning, volume 3, pages 1–122. Now Publishers, Inc., 2011.

M. Carreira-Perpiñán and W. Wang. Distributed optimization of deeply nested systems. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS), 2014.
D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, to appear, 2019.

J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

K. Fukushima. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. [http://www.deeplearningbook.org](http://www.deeplearningbook.org).

F. Gu, A. Askari, and L. El Ghaoui. Fenchel lifted networks: a Lagrange relaxation of neural network training. *arXiv preprint arXiv:1811.08039*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

N.-N. Ji, J.-S. Zhang, and C.-X. Zhang. A sparse-response deep belief network based on rate distortion theory. *Pattern Recognition*, 47(9):3179–3191, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

S. Krantz and H. R. Parks. *A primer of real analytic functions*. Birkhäuser, second edition, 2002.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

K. Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l’institut Fourier*, 48:769–783, 1998.

T. T.-K. Lau, J. Zeng, B. Wu, and Y. Yao. A proximal block coordinate descent algorithm for deep neural network training. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2018.

Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In G. Montavon, G. B. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Q. Liao and T. Poggio. Theory II: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*, 2017.

S. Lojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In: *Les Équations aux dérivées partielles. Éditions du centre National de la Recherche Scientifique, Paris*, pages 87–89, 1963.

S. Lojasiewicz. *Ensembles semi-analytiques*. Institut des Hautes Études Scientifiques, 1965.
S. Lojasiewicz. Sur la geometrie semi-et sous-analytique. *Annales de l’institut Fourier*, 43:1575–1595, 1993.

D. Mahajan, S. S. Keerthi, and S. Sundararajan. A distributed block coordinate descent method for training $\ell_1$ regularized linear classifiers. *Journal of Machine Learning Research*, 18:1–35, 2017.

B. S. Mordukhovich. *Variational analysis and generalized differentiation I: Basic Theory*. Springer, 2006.

Y. Nesterov. *Lectures on convex optimization*. Springer, second edition, 2018.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Grundlehren Math. Wiss. 317, Springer-Verlag, New York, 1998.

F. Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan Books, Washington DC, 1961.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for LVCSR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

M. Shiota. *Geometry of subanalytic and semialgebraic sets*, volume 150 of *Progress in Mathematics*. Birkhäuser, Boston, 1997.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein. Training neural networks without gradients: A scalable ADMM approach. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

T. Tieleman and G. Hinton. Lecture 6.5-RMSProp: Divide the gradient by running average of its recent magnitude. *Coursera Neural Networks for Machine Learning*, 4, 2012.

Y. Wang, W. Yin, and J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 2018.

Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

Y. Xu and W. Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72:700–734, 2017.

Z. Zhang and M. Brand. Convergent block coordinate descent for training Tikhonov regularized deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Z. Zhang, Y. Chen, and V. Saligrama. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.