Global Convergence of Block Coordinate Descent in Deep Learning

Jinshan Zeng [1 2 *], Tim Tsz-Kit Lau [3 *], Shao-Bo Lin [4], Yuan Yao [2] [1] Jiangxi Normal Univ. [2] HKUST [3] Northwestern [4] CityU HK [*] Equal contribution

Provide a general methodology to establish the global convergence of BCD methods in training deep neural networks to a critical point at a rate of $\mathcal{O}(1/k)$, where k is the number of iterations, without the *block multiconvexity* and *differentiability* assumptions

INTRODUCTION

Motivation of Block Coordinate Descent (BCD) in Deep Learning

- Gradient-based methods are commonly used in training deep neural networks (DNNs), but may suffer from various problems, especially when neural networks become deeper (i.e., more layers):
  Since the gradients of the loss function w.r.t. parameters of earlier layers involve those of later layers
  1. if one of the later layer gradients is zero, all gradients of previous layers are zero (known as **gradient vanishing**);
  2. if one of the later layer gradients is $\pm\infty$ (in terms of computer precision), all gradients of previous layers are $\pm\infty$ (known as **gradient exploding**)
- Gradient-free methods have recently been adapted to training DNNs:
  1. Block Coordinate Descent (BCD)
  2. Alternating Direction Method of Multipliers (ADMM)
- Advantages of Gradient-free Methods:
  1. Ability to deal with non-differentiable nonlinearities and potentially avoid **vanishing gradient**
  2. Can be easily implemented in a *distributed* and *parallel* fashion

DNN TRAINING VIA BLOCK COORDINATE DESCENT

Variable Splitting

- View parameters of hidden layers and the output layer as variable blocks
- **Variable splitting**: Split the highly coupled network layer-wise to compose a surrogate loss function
- Notations:
  - $\mathcal{W} := \{\boldsymbol{W}_\ell\}_{\ell=1}^L$: the set of layer parameters (bias vectors are absorbed)
  - $\mathfrak{L} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+ \cup \{0\}$: some loss function
  - $\Phi(\boldsymbol{x}_i; \mathcal{W}) := \sigma_L(\boldsymbol{W}_L \sigma_{L-1}(\boldsymbol{W}_{L-1} \cdots \boldsymbol{W}_2 \sigma_1(\boldsymbol{W}_1 \boldsymbol{x}_i)))$: the neural network
- **Empirical risk minimization**:

$$\min_{\mathcal{W}} \mathcal{R}_n(\Phi(\boldsymbol{X}; \mathcal{W}), \boldsymbol{Y}) := \frac{1}{n} \sum_{i=1}^n \mathfrak{L}(\Phi(\boldsymbol{x}_i; \mathcal{W}), \boldsymbol{y}_i)$$

Two-Splitting Formulation

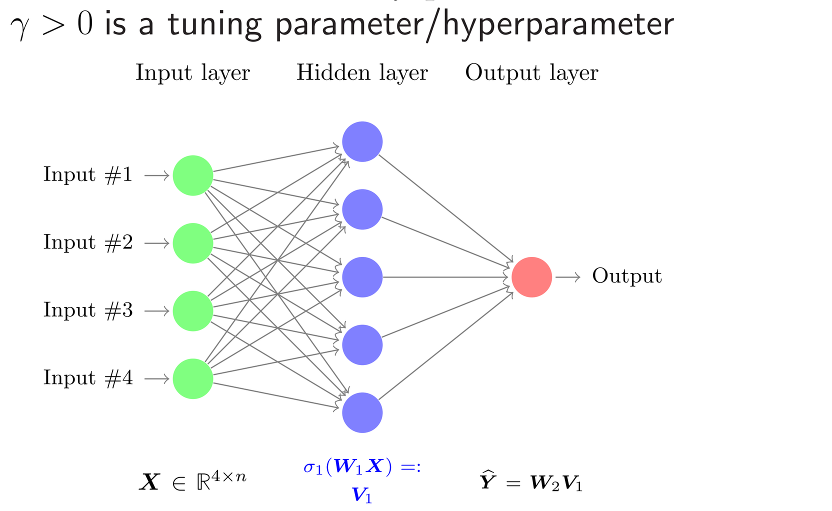
- Introduce one set of auxiliary variables $\mathcal{V} := \{\boldsymbol{V}_\ell\}_{\ell=1}^L$

$$\min_{\mathcal{W},\mathcal{V}} \mathcal{L}_0(\mathcal{W}, \mathcal{V}) := \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \sum_{\ell=1}^L r_\ell(\boldsymbol{W}_\ell) + \sum_{\ell=1}^L s_\ell(\boldsymbol{V}_\ell)$$
$$\text{subject to} \quad \boldsymbol{V}_\ell = \sigma_\ell(\boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1}),\ \ell \in \{1, \ldots, L\}$$

- The functions r_ℓ and s_ℓ are nonnegative functions revealing the priors of the weight variable $\boldsymbol{W}_\ell$ and the state variable $\boldsymbol{V}_\ell$ (i.e., regularizers)
- The constrained optimization is usually rewritten as an unconstrained one through:

$$\min_{\mathcal{W},\mathcal{V}} \mathcal{L}(\mathcal{W}, \mathcal{V}) := \mathcal{L}_0(\mathcal{W}, \mathcal{V}) + \frac{\gamma}{2} \sum_{\ell=1}^L \|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1})\|_F^2,$$

where $\gamma > 0$ is a tuning parameter/hyperparameter



- Jointly minimize the *squared distances* (in terms of **Frobenius norms**) between the input and the output of hidden layers
- E.g., define $\boldsymbol{V}_0 := \boldsymbol{X}$,

$$\|\boldsymbol{V}_1 - \sigma_1(\boldsymbol{W}_1 \boldsymbol{V}_0)\|_F^2$$

Three-Splitting Formulation

- Introduce two sets of auxiliary variables $\mathcal{U} := \{\boldsymbol{U}_\ell\}_{\ell=1}^L$, $\mathcal{V} := \{\boldsymbol{V}_\ell\}_{\ell=1}^L$

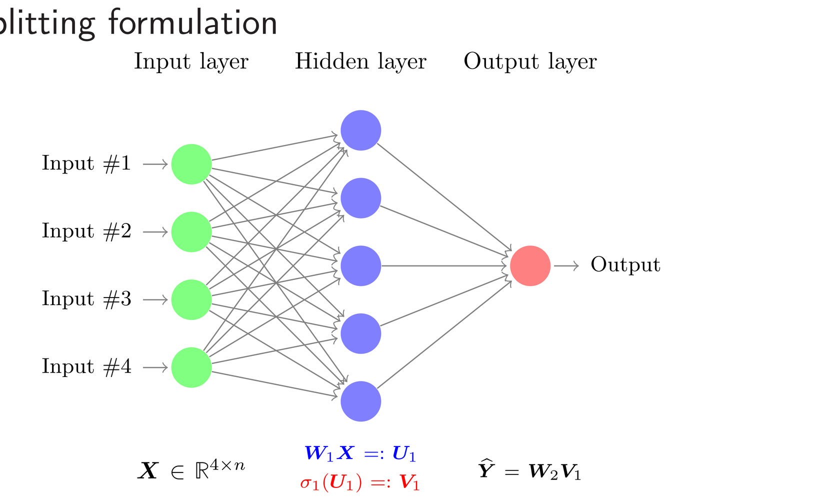
$$\min_{\mathcal{W},\mathcal{V},\mathcal{U}} \mathcal{L}_0(\mathcal{W}, \mathcal{V})$$
$$\text{subject to} \quad \boldsymbol{U}_\ell = \boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1},\ \boldsymbol{V}_\ell = \sigma_\ell(\boldsymbol{U}_\ell),\ \ell \in \{1, \ldots, L\}$$

- The constrained optimization is usually rewritten as an unconstrained one through:

$$\min_{\mathcal{W},\mathcal{V}} \mathcal{L}(\mathcal{W}, \mathcal{V}) := \mathcal{L}_0(\mathcal{W}, \mathcal{V}) + \frac{\gamma}{2} \sum_{\ell=1}^L \|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1})\|_F^2,$$

where $\gamma > 0$ is a tuning parameter/hyperparameter

- Variables are more loosely coupled than those in the two-splitting formulation



- Jointly minimize the *squared distances* (in terms of **Frobenius norms**) between
  1. the input and the *pre-activation* output of hidden layers
  2. the *pre-activation* output and the *post-activation* output of hidden layers
- E.g., define $\boldsymbol{V}_0 := \boldsymbol{X}$,

$$\|\boldsymbol{U}_1 - \boldsymbol{W}_1 \boldsymbol{V}_0\|_F^2 + \|\boldsymbol{V}_1 - \sigma_1(\boldsymbol{U}_1)\|_F^2$$

Algorithms

Algorithm 1 Two-splitting BCD for DNN Training

Data: $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, $\boldsymbol{Y} \in \mathbb{R}^{k\times n}$
Initialization: $\{\boldsymbol{W}_\ell^{(0)}, \boldsymbol{V}_\ell^{(0)}\}_{\ell=1}^L$, $\boldsymbol{V}_0^{(t)} \equiv \boldsymbol{V}_0 := \boldsymbol{X}$
Hyperparameters: $\gamma > 0$, $\alpha > 0$
for $t = 1, \ldots$ **do**
$\quad \boldsymbol{V}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_L} \{s_L(\boldsymbol{V}_L) + \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \frac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{W}_L^{(t-1)}\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2\}$
$\quad \boldsymbol{W}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_L} \{r_L(\boldsymbol{W}_L) + \frac{\gamma}{2}\|\boldsymbol{V}_L^{(t)} - \boldsymbol{W}_L\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_L - \boldsymbol{W}_L^{(t-1)}\|_F^2\}$
$\quad$ **for** $\ell = L-1, \ldots, 1$ **do**
$\quad\quad \boldsymbol{V}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_\ell} \{s_\ell(\boldsymbol{V}_\ell) + \frac{\gamma}{2}\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{W}_\ell^{(t-1)}\boldsymbol{V}_{\ell-1}^{(t-1)})\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{V}_{\ell+1}^{(t)} - \sigma_{\ell+1}(\boldsymbol{W}_{\ell+1}^{(t)}\boldsymbol{V}_\ell)\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{V}_\ell - \boldsymbol{V}_\ell^{(t-1)}\|_F^2\}$
$\quad\quad \boldsymbol{W}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_\ell} \{r_\ell(\boldsymbol{W}_\ell) + \frac{\gamma}{2}\|\boldsymbol{V}_\ell^{(t)} - \sigma_\ell(\boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}^{(t-1)})\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_\ell - \boldsymbol{W}_\ell^{(t-1)}\|_F^2\}$
$\quad$ **end for**
end for

Algorithm 2 Three-splitting BCD for DNN training

Samples: $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, $\boldsymbol{Y} \in \mathbb{R}^{k\times n}$
Initialization: $\{\boldsymbol{W}_\ell^{(0)}, \boldsymbol{V}_\ell^{(0)}, \boldsymbol{U}_\ell^{(0)}\}_{\ell=1}^L$, $\boldsymbol{V}_0^{(t)} \equiv \boldsymbol{V}_0 := \boldsymbol{X}$
Hyperparameters: $\gamma > 0$, $\alpha > 0$
for $t = 1, \ldots$ **do**
$\quad \boldsymbol{V}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_L} \{s_L(\boldsymbol{V}_L) + \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \frac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{U}_L^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2\}$
$\quad \boldsymbol{U}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{U}_L} \{\frac{\gamma}{2}\|\boldsymbol{V}_L^{(t)} - \boldsymbol{U}_L\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{U}_L - \boldsymbol{W}_L^{(t-1)}\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2\}$
$\quad \boldsymbol{W}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_L} \{r_L(\boldsymbol{W}_L) + \frac{\gamma}{2}\|\boldsymbol{U}_L^{(t)} - \boldsymbol{W}_L\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_L - \boldsymbol{W}_L^{(t-1)}\|_F^2\}$
$\quad$ **for** $\ell = L-1, \ldots, 1$ **do**
$\quad\quad \boldsymbol{V}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_\ell} \{s_\ell(\boldsymbol{V}_\ell) + \frac{\gamma}{2}\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{U}_\ell^{(t-1)})\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{U}_{\ell+1}^{(t)} - \boldsymbol{W}_{\ell+1}^{(t)}\boldsymbol{V}_\ell\|_F^2\}$
$\quad\quad \boldsymbol{U}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{U}_\ell} \{\frac{\gamma}{2}\|\boldsymbol{V}_\ell^{(t)} - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{U}_\ell - \boldsymbol{W}_\ell^{(t-1)}\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{U}_\ell - \boldsymbol{U}_\ell^{(t-1)}\|_F^2\}$
$\quad\quad \boldsymbol{W}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_\ell} \{r_\ell(\boldsymbol{W}_\ell) + \frac{\gamma}{2}\|\boldsymbol{U}_\ell^{(t)} - \boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_\ell - \boldsymbol{W}_\ell^{(t-1)}\|_F^2\}$
$\quad$ **end for**
end for

GLOBAL CONVERGENCE ANALYSIS

Main Assumptions

Assumption 1 *Suppose that*

(a) *the loss function $\mathcal{L}$ is a proper lower semicontinuous[1] and nonnegative function,*
(b) *the activation functions σ_ℓ ($\ell = 1 \ldots, L-1$) are Lipschitz continuous on any bounded set,*
(c) *the regularizers r_ℓ and s_ℓ ($\ell = 1 \ldots, L-1$) are nonegative lower semicontinuous convex functions, and*
(d) *all these functions $\mathcal{L}$, σ_ℓ, r_ℓ and s_ℓ ($\ell = 1 \ldots, L-1$) are either real analytic or semialgebraic, and continuous on their domains.*

Proposition 1 *Examples satisfying Assumption 1 include:*

(a) *$\mathfrak{L}$ is the squared, logistic, hinge, or cross-entropy losses;*
(b) *σ_ℓ is ReLU, leaky ReLU, sigmoid, hyperbolic tangent, linear, polynomial, or softplus activations;*
(c) *r_ℓ and s_ℓ are the squared ℓ_2 norm, the ℓ_1 norm, the elastic net, the indicator function of some nonempty closed convex set (such as the nonnegative closed half space, box set or a closed interval $[0, 1]$), or 0 if no regularization.*

Main Theorem

Theorem 1 *Let $\{\mathcal{Q}^t := (\{\boldsymbol{W}_\ell^t\}_{\ell=1}^L, \{\boldsymbol{V}_\ell^t\}_{\ell=1}^L)\}_{t\in\mathbb{N}}$ and $\{\mathcal{P}^t := (\{\boldsymbol{W}_\ell^t\}_{\ell=1}^L, \{\boldsymbol{V}_\ell^t\}_{\ell=1}^L, \{\boldsymbol{U}_\ell^t\}_{\ell=1}^L)\}_{t\in\mathbb{N}}$ be the sequences generated by Algorithms 1 and 2, respectively. Suppose that Assumption 1 holds, and that one of the following conditions holds: (i) there exists a convergent subsequence $\{\mathcal{Q}^{t_j}\}_{j\in\mathbb{N}}$ (resp. $\{\mathcal{P}^{t_j}\}_{j\in\mathbb{N}}$); (ii) r_ℓ is coercive for any $\ell = 1, \ldots, L$; (iii) $\mathcal{L}$ (resp. $\overline{\mathcal{L}}$) is coercive. Then for any $\alpha > 0$, $\gamma > 0$ and any finite initialization $\mathcal{Q}^0$ (resp. $\mathcal{P}^0$), the following hold*

(a) *$\{\mathcal{L}(\mathcal{Q}^t)\}_{t\in\mathbb{N}}$ (resp. $\{\overline{\mathcal{L}}(\mathcal{P}^t)\}_{t\in\mathbb{N}}$) converges to some $\mathcal{L}^\star$ (resp. $\overline{\mathcal{L}}^\star$).*
(b) *$\{\mathcal{Q}^t\}_{t\in\mathbb{N}}$ (resp. $\{\mathcal{P}^t\}_{t\in\mathbb{N}}$) converges to a critical point of $\mathcal{L}$ (resp. $\overline{\mathcal{L}}$).*
(c) *$\frac{1}{T}\sum_{t=1}^T \|\boldsymbol{g}^t\|_F^2 \to 0$ at the rate $\mathcal{O}(1/T)$ where $\boldsymbol{g}^t \in \partial\mathcal{L}(\mathcal{Q}^t)$. Similarly, $\frac{1}{T}\sum_{t=1}^T \|\bar{\boldsymbol{g}}^t\|_F^2 \to 0$ at the rate $\mathcal{O}(1/T)$ where $\bar{\boldsymbol{g}}^t \in \partial\overline{\mathcal{L}}(\mathcal{P}^t)$.*

Extensions

Extension to Prox-Linear

- In the $\boldsymbol{V}_L$-update of both Algorithms 1 and 2, the empirical risk is involved in the optimization problems
- Generally hard to obtain its closed-form solution except for the square loss
- Use *prox-linear* update strategies for other smooth losses such as the logistic, cross-entropy, and exponential losses
- For some parameter $\alpha > 0$, the $\boldsymbol{V}_L$-update in Algorithm 1 is

$$\boldsymbol{V}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_L} \left\{s_L(\boldsymbol{V}_L) + \langle \nabla\mathcal{R}_n(\boldsymbol{V}_L^{(t-1)}; \boldsymbol{Y}), \boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\rangle + \frac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{W}_L^{(t-1)}\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2\right\}, \tag{1}$$

- The $\boldsymbol{V}_L$-update in Algorithm 2 is

$$\boldsymbol{V}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_L} \left\{s_L(\boldsymbol{V}_L) + \langle \nabla\mathcal{R}_n(\boldsymbol{V}_L^{(t-1)}; \boldsymbol{Y}), \boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\rangle + \frac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2 + \frac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{U}_L^{(t-1)}\|_F^2\right\}. \tag{2}$$

Theorem 2 (Global convergence for prox-linear update) *Consider adopting the prox-linear updates (1), (2) for the $\boldsymbol{V}_L$-subproblems in Algorithms 1 and 2, respectively. Under the conditions of Theorem 1, if further $\nabla\mathcal{R}_n$ 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 Theorem 1 still hold for both algorithms.*

Extension to ResNet Training

- Consider the simplified ResNet training problem (**two-splitting formulation**):

$$\min_{\mathcal{W},\mathcal{V}} \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \sum_{\ell=1}^L r_\ell(\boldsymbol{W}_\ell) + \sum_{\ell=1}^L s_\ell(\boldsymbol{V}_\ell)$$
$$\text{subject to} \quad \boldsymbol{V}_\ell - \boldsymbol{V}_{\ell-1} = \sigma_\ell(\boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1}),\ \ell \in \{1, \ldots, L\} \tag{3}$$

- **Three-splitting formulation**:

$$\min_{\mathcal{W},\mathcal{V},\mathcal{U}} \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \sum_{\ell=1}^L r_\ell(\boldsymbol{W}_\ell) + \sum_{\ell=1}^L s_\ell(\boldsymbol{V}_\ell)$$
$$\text{subject to} \quad \boldsymbol{U}_\ell = \boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1},\ \boldsymbol{V}_\ell - \boldsymbol{V}_{\ell-1} = \sigma_\ell(\boldsymbol{U}_\ell),\ \ell \in \{1, \ldots, L\}$$

$$\overline{\mathcal{L}}_{\text{res}}(\mathcal{W}, \mathcal{V}, \mathcal{U}) := \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \sum_{\ell=1}^L r_\ell(\boldsymbol{W}_\ell) + \sum_{\ell=1}^L s_\ell(\boldsymbol{V}_\ell) + \frac{\gamma}{2}\sum_{\ell=1}^L \left[\|\boldsymbol{V}_\ell - \boldsymbol{V}_{\ell-1} - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2 + \|\boldsymbol{U}_\ell - \boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}\|_F^2\right]$$

Algorithm 3 BCD for DNN Training with ResNets

Samples: $\boldsymbol{X} \in \mathbb{R}^{d_0\times n}$, $\boldsymbol{Y} \in \mathbb{R}^{d_N\times n}$, $\boldsymbol{V}_0^{(t)} \equiv \boldsymbol{V}_0 := \boldsymbol{X}$
Initialization: $\{\boldsymbol{W}_\ell^0, \boldsymbol{V}_\ell^0, \boldsymbol{U}_\ell^0\}_{\ell=1}^L$
Parameters: $\gamma > 0$, $\alpha > 0$
for $t = 1, \ldots$ **do**
$\quad \boldsymbol{V}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_L} \{s_L(\boldsymbol{V}_L) + \mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \frac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_{L-1}^{(t-1)} - \boldsymbol{U}_L^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2\}$
$\quad \boldsymbol{U}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{U}_L} \frac{\gamma}{2}[\|\boldsymbol{V}_L^{(t)} - \boldsymbol{V}_{L-1}^{(t-1)} - \boldsymbol{U}_L\|_F^2 + \|\boldsymbol{U}_L - \boldsymbol{W}_L^{(t-1)}\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2]$
$\quad \boldsymbol{W}_L^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_L} \{r_L(\boldsymbol{W}_L) + \frac{\gamma}{2}\|\boldsymbol{U}_L^{(t)} - \boldsymbol{W}_L\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_L - \boldsymbol{W}_L^{(t-1)}\|_F^2\}$
$\quad$ **for** $\ell = L-1, \ldots, 1$ **do**
$\quad\quad \boldsymbol{V}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{V}_\ell} \{s_\ell(\boldsymbol{V}_\ell) + \frac{\gamma}{2}[\|\boldsymbol{V}_\ell - \boldsymbol{V}_{\ell-1}^{(t-1)} - \sigma_\ell(\boldsymbol{U}_\ell^{(t-1)})\|_F^2 + \|\boldsymbol{V}_{\ell+1}^{(t)} - \boldsymbol{V}_\ell - \sigma_{\ell+1}(\boldsymbol{U}_{\ell+1}^{(t)})\|_F^2 + \|\boldsymbol{U}_{\ell+1}^{(t)} - \boldsymbol{W}_{\ell+1}^{(t)}\boldsymbol{V}_\ell\|_F^2]\}$
$\quad\quad \boldsymbol{U}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{U}_\ell} \{\frac{\gamma}{2}[\|\boldsymbol{V}_\ell^{(t)} - \boldsymbol{V}_{\ell-1}^{(t-1)} - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2 + \|\boldsymbol{U}_\ell - \boldsymbol{W}_\ell^{(t-1)}\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2] + \frac{\alpha}{2}\|\boldsymbol{U}_\ell - \boldsymbol{U}_\ell^{(t-1)}\|_F^2\}$
$\quad\quad \boldsymbol{W}_\ell^{(t)} = \operatorname{argmin}_{\boldsymbol{W}_\ell} \{r_\ell(\boldsymbol{W}_\ell) + \frac{\gamma}{2}\|\boldsymbol{U}_\ell^{(t)} - \boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{W}_\ell - \boldsymbol{W}_\ell^{(t-1)}\|_F^2\}$
$\quad$ **end for**
end for

Theorem 3 (Convergence of BCD for ResNets) *Let $\{\{\boldsymbol{W}_\ell^{(t)}, \boldsymbol{V}_\ell^{(t)}, \boldsymbol{U}_\ell^{(t)}\}_{\ell=1}^L\}_{t\in\mathbb{N}}$ 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 by replacing $\overline{\mathcal{L}}$ with $\overline{\mathcal{L}}_{\text{res}}$.*

Moreover, consider adopting the prox-linear update for the $\boldsymbol{V}_L$-subproblem in Algorithm 3, then under the assumptions of Theorem 2, all claims of Theorem 2 still hold for Algorithm 3.

PROOF IDEAS

Four key ingredients:

1. The *sufficient descent* condition
2. The *relative error* condition
3. The *continuity condition* of the objective function
4. The *Kurdyka-Łojasiewicz* property of the objective function

Establishing the *sufficient descent* and the *relative error* conditions require two kinds of assumptions:

(a) multiconvexity and differentiability assumptions, and
(b) (blockwise) Lipschitz differentiability assumption on the unregularized part of objective function

- In our cases, the unregularized part of $\mathcal{L}$ in **two-splitting formulation**,

$$\mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \frac{\gamma}{2}\sum_{\ell=1}^L \|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1})\|_F^2,$$

and that of $\overline{\mathcal{L}}$ in **three-splitting formulation**,

$$\mathcal{R}_n(\boldsymbol{V}_L; \boldsymbol{Y}) + \frac{\gamma}{2}\sum_{\ell=1}^L \left[\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2 + \|\boldsymbol{U}_\ell - \boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}\|_F^2\right]$$

usually do not satisfy any of **assumption (a)** and **assumption (b)**

- E.g., when σ_ℓ is ReLU or leaky ReLU, the functions $\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1})\|_F^2$ and $\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2$ are non-differentiable and nonconvex with respect to $\boldsymbol{W}_\ell$-block and $\boldsymbol{U}_\ell$-block, respectively
- To overcome these challenges:
  (i) 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)
  (ii) 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)

Suffcient Descent Lemma

Lemma 1 (Sucient descent) *Let $\{\mathcal{P}^t\}_{t\in\mathbb{N}}$ be a sequence generated by the BCD method (Algorithm 2). Then, under the assumptions of Theorem 1,*

$$\overline{\mathcal{L}}(\mathcal{P}^t) \le \overline{\mathcal{L}}(\mathcal{P}^{t-1}) - a\|\mathcal{P}^t - \mathcal{P}^{t-1}\|_F^2,$$

for some constant $a > 0$ specified in the proof.

Lemma 1 tells:

(i) $\{\overline{\mathcal{L}}(\mathcal{P}^t)\}_{t\in\mathbb{N}}$ is convergent if $\overline{\mathcal{L}}$ is lower bounded;
(ii) $\{\mathcal{P}^t\}_{t\in\mathbb{N}}$ itself is bounded if $\overline{\mathcal{L}}$ is coercive and $\mathcal{P}^0$ is finite;
(iii) $\{\mathcal{P}^t\}_{t\in\mathbb{N}}$ is square summable, i.e., $\sum_{t=1}^\infty \|\mathcal{P}^t - \mathcal{P}^{t-1}\|_F^2 < \infty$, implying its asymptotic regularity, i.e., $\|\mathcal{P}^t - \mathcal{P}^{t-1}\|_F \to 0$ as $t \to \infty$; and
(iv) $\frac{1}{T}\sum_{t=1}^T \|\mathcal{P}^t - \mathcal{P}^{t-1}\|_F^2 \to 0$ at a rate of $\mathcal{O}(1/T)$.

Relative Error Lemma

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

$$\|\bar{\boldsymbol{g}}^t\|_F \le \bar{b}\|\mathcal{P}^t - \mathcal{P}^{t-1}\|_F, \quad \bar{\boldsymbol{g}}^t \in \partial\overline{\mathcal{L}}(\mathcal{P}^t)$$

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

$$\partial\overline{\mathcal{L}}(\mathcal{P}^t) := (\{\partial_{\boldsymbol{W}_\ell}\overline{\mathcal{L}}\}_{\ell=1}^L, \{\partial_{\boldsymbol{V}_\ell}\overline{\mathcal{L}}\}_{\ell=1}^L, \{\partial_{\boldsymbol{U}_\ell}\overline{\mathcal{L}}\}_{\ell=1}^L)(\mathcal{P}^t).$$

- The subgradient sequence of the Lagrangian is upper bounded by the discrepancy between the current and previous iterates
- Together with the asymptotic regularity of $\{\mathcal{P}^t\}_{t\in\mathbb{N}}$ yielded by Lemma 1, Lemma 2 shows the **critical point convergence**
- Also, together with the claim **(iv)** implied by Lemma 1, Lemma 2 yields the $\mathcal{O}(1/T)$ rate of convergence (to a critical point) of BCD, i.e., $\frac{1}{T}\sum_{t=1}^T \|\bar{\boldsymbol{g}}^t\|_F \to 0$ at the rate of $\mathcal{O}(1/T)$
- Both *differentiability* and *(blockwise) Lipschitz differentiability* assumptions are not imposed
- Only use the *Lipschitz continuity* (on any bounded set) of the activations (mild and natural condition)

DEMONSTRATION

- 10-class classification for the MNIST dataset (with 60K training samples; 10K test samples)
- 784-(600 × 10)-10 MLPs
- 10 hidden layers (rather deep; usually < 5 hidden layers)
- Comparison of training and test accuracies (after 100 epochs)
- SGD fails to train deeper MLPs due to vanishing gradient