ICML 2019
Long Beach, CA

Global Convergence of Block Coordinate Descent in Deep Learning

Jinshan Zeng[1 2 *] Tim Tsz-Kit Lau[3 *] Shaobo Lin[4] Yuan Yao[2]

[1]Jiangxi Normal Univ. [2]HKUST [3]Northwestern [4]CityU HK

[*]Equal contribution

Tim Tsz-Kit Lau

Department of Statistics **Northwestern University**

INTRODUCTION

MOTIVATION OF BLOCK COORDINATE DESCENT (BCD) IN DEEP LEARNING

- Gradient-based methods are commonly used in training deep neural networks
- But gradient-based methods may suffer from various problems for deep networks
- Gradients of the loss function w.r.t. parameters of earlier layers involve those of later layers
  - $\Rightarrow$ **Gradient vanishing**
  - $\Rightarrow$ **Gradient exploding**
- **First-order gradient-based** methods does not work well

MOTIVATION OF BLOCK COORDINATE DESCENT (BCD) IN DEEP LEARNING

- **Gradient-free** methods have recently been adapted to training DNNs:
  - *Block Coordinate Descent (BCD)*
  - *Alternating Direction Method of Multipliers (ADMM)*
- Advantages of Gradient-free Methods:
  - Deal with non-differentiable nonlinearities
  - Potentially avoid **vanishing gradient**
  - Can be easily implemented in a *distributed* and *parallel* fashion

BLOCK COORDINATE DESCENT IN DEEP LEARNING

BLOCK COORDINATE DESCENT IN DEEP LEARNING

- 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
  - $\mathfrak{L} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+ \cup \{0\}$: 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 ways of variable splitting appear in the literature

BCD IN DEEP LEARNING: TWO-SPLITTING FORMULATION

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

$$\min_{\mathcal{W},\mathcal{V}} \mathcal{L}_0(\mathcal{W},\mathcal{V}) := \mathcal{R}_n(\boldsymbol{V}_L;\boldsymbol{Y}) + \sum_{\ell=1}^L r_\ell(\boldsymbol{W}_\ell) + \sum_{\ell=1}^L s_\ell(\boldsymbol{V}_\ell)$$

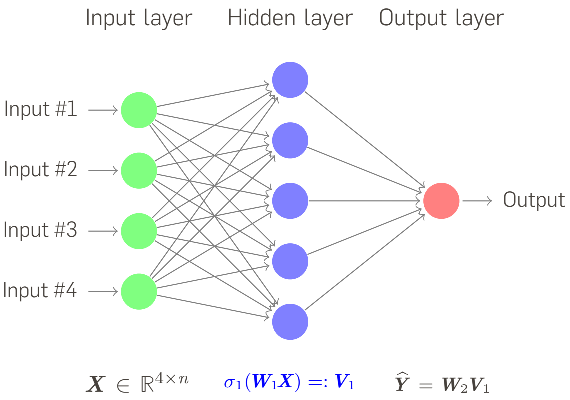
$$\text{subject to} \quad \boldsymbol{V}_\ell = \sigma_\ell(\boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1}),\ \ell \in \{1,\ldots,L\}$$

- The functions r_ℓ and s_ℓ are regularizers
- Rewritten as unconstrained optimization:

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

- $\gamma > 0$ is a hyperparameter

TWO-SPLITTING FORMULATION: GRAPHICAL ILLUSTRATION



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

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

BCD IN DEEP LEARNING: THREE-SPLITTING FORMULATION

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

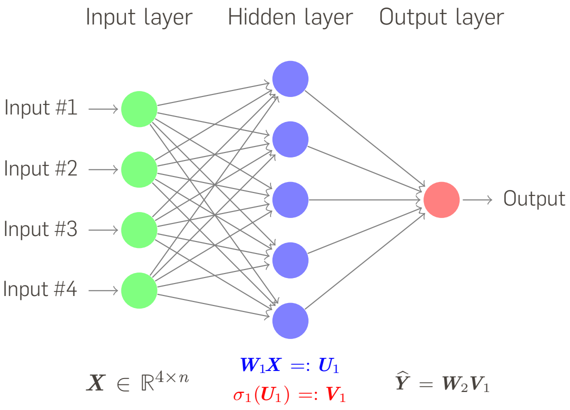
$$\min_{\mathcal{W},\mathcal{V},\mathcal{U}} \mathcal{L}_0(\mathcal{W},\mathcal{V}) \quad \text{subject to} \quad \boldsymbol{U}_\ell = \boldsymbol{W}_\ell \boldsymbol{V}_{\ell-1},\ \boldsymbol{V}_\ell = \sigma_\ell(\boldsymbol{U}_\ell),\ \ell \in \{1,\ldots,L\}$$

- Rewritten as unconstrained optimization:

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

- Variables **more loosely coupled** than those in two-splitting

THREE-SPLITTING FORMULATION: GRAPHICAL ILLUSTRATION



- Jointly minimize the *distances* (in terms of **squared 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 $\boldsymbol{V}_0 := \boldsymbol{X}$,

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

BLOCK COORDINATE DESCENT (BCD) ALGORITHMS

BLOCK COORDINATE DESCENT (BCD) ALGORITHMS

- Devise algorithms for training DNNs based on the two formulations
- Update all the variables cyclically while fixing the remaining blocks
- Update in a backward order as in **backpropagation**
- Adopt the *proximal update strategies*

BCD ALGORITHM (TWO-SPLITTING)

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

BCD ALGORITHM (THREE-SPLITTING)

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

$$\boldsymbol{V}_L^{(t)} = \mathrm{argmin}_{\boldsymbol{V}_L} \{s_L(\boldsymbol{V}_L) + \mathcal{R}_n(\boldsymbol{V}_L;\boldsymbol{Y}) + \tfrac{\gamma}{2}\|\boldsymbol{V}_L - \boldsymbol{U}_L^{(t-1)}\|_F^2 + \tfrac{\alpha}{2}\|\boldsymbol{V}_L - \boldsymbol{V}_L^{(t-1)}\|_F^2\}$$

$$\boldsymbol{U}_L^{(t)} = \mathrm{argmin}_{\boldsymbol{U}_L} \{\tfrac{\gamma}{2}\|\boldsymbol{V}_L^{(t)} - \boldsymbol{U}_L\|_F^2 + \tfrac{\gamma}{2}\|\boldsymbol{U}_L - \boldsymbol{W}_L^{(t-1)}\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2\}$$

$$\boldsymbol{W}_L^{(t)} = \mathrm{argmin}_{\boldsymbol{W}_L} \{r_L(\boldsymbol{W}_L) + \tfrac{\gamma}{2}\|\boldsymbol{U}_L^{(t)} - \boldsymbol{W}_L\boldsymbol{V}_{L-1}^{(t-1)}\|_F^2 + \tfrac{\alpha}{2}\|\boldsymbol{W}_L - \boldsymbol{W}_L^{(t-1)}\|_F^2\}$$

for $\ell = L-1, \ldots, 1$ do

$$\boldsymbol{V}_\ell^{(t)} = \mathrm{argmin}_{\boldsymbol{V}_\ell} \{s_\ell(\boldsymbol{V}_\ell) + \tfrac{\gamma}{2}\|\boldsymbol{V}_\ell - \sigma_\ell(\boldsymbol{U}_\ell^{(t-1)})\|_F^2 + \tfrac{\gamma}{2}\|\boldsymbol{U}_{\ell+1}^{(t)} - \boldsymbol{W}_{\ell+1}^{(t)}\boldsymbol{V}_\ell\|_F^2\}$$

$$\boldsymbol{U}_\ell^{(t)} = \mathrm{argmin}_{\boldsymbol{U}_\ell} \{\tfrac{\gamma}{2}\|\boldsymbol{V}_\ell^{(t)} - \sigma_\ell(\boldsymbol{U}_\ell)\|_F^2 + \tfrac{\gamma}{2}\|\boldsymbol{U}_\ell - \boldsymbol{W}_\ell^{(t-1)}\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2 + \tfrac{\alpha}{2}\|\boldsymbol{U}_\ell - \boldsymbol{U}_\ell^{(t-1)}\|_F^2\}$$

$$\boldsymbol{W}_\ell^{(t)} = \mathrm{argmin}_{\boldsymbol{W}_\ell}\{r_\ell(\boldsymbol{W}_\ell) + \tfrac{\gamma}{2}\|\boldsymbol{U}_\ell^{(t)} - \boldsymbol{W}_\ell\boldsymbol{V}_{\ell-1}^{(t-1)}\|_F^2 + \tfrac{\alpha}{2}\|\boldsymbol{W}_\ell - \boldsymbol{W}_\ell^{(t-1)}\|_F^2\}$$

end for

end for

GLOBAL CONVERGENCE ANALYSIS

ASSUMPTIONS OF THE FUNCTIONS FOR CONVERGENCE GUARANTEES

Assumption

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.

[1] A function $f : \mathcal{X} \to \mathbb{R}$ is called *lower semicontinuous* if $\liminf_{\boldsymbol{x}\to\boldsymbol{x}_0} f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0)$ for any $\boldsymbol{x}_0 \in \mathcal{X}$.

EXAMPLES OF THE FUNCTIONS

Proposition

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

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[2] 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 $\bar{\boldsymbol{g}}^t \in \partial\overline{\mathcal{L}}(\mathcal{P}^t)$.

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

EXTENSIONS

Extensions

1. *Prox-linear updates* instead of proximal update strategies
2. *Residual Networks (ResNets)* with skip connections

Global convergence of both extensions are also proved

PROOF IDEAS

PROOF IDEAS

Four key ingredients:

- The *sufficient descent* condition
- The *relative error* condition
- The *continuity* condition of the objective function
- 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

PROOF IDEAS

- 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

PROOF IDEAS

To overcome these challenges:

(i) Exploit the **proximal strategies** for all the *non-strongly convex* subproblems to cheaply obtain the desired *sufficient descent* property

(ii) Take advantage of the **Lipschitz continuity** of the activations as well as the specific splitting formulations to yield the desired *relative error* property

SUMMARY OF THEORETICAL RESULTS OF THIS PAPER

Theoretical Results

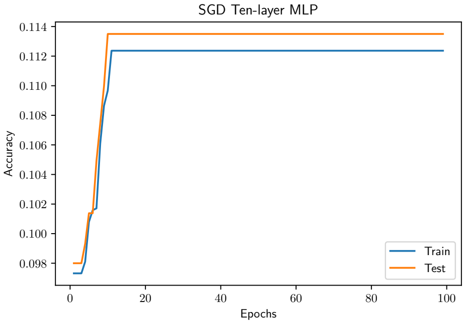
1. Global convergence to a critical point at a rate of $\mathcal{O}(1/T)$, where T is the number of iterations
2. Further, if the initialization is sufficiently close to some global minimum of $\mathcal{L}$ or $\overline{\mathcal{L}}$, then both the sequences generated by Algorithms 1 and 2 converges to their corresponding global minima
3. **Comparison with the convergence of SGD/stochastic subgradient method**:
   - *BCD*: **Global (whole sequence)** convergence
   - *SGD* (Davis et al., 2019): **Subsequence** convergence

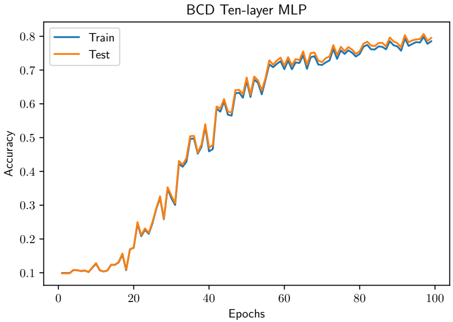
[Davis et al., **Stochastic subgradient method converges on tame functions**, *FOCM* (2019)]

DEMONSTRATION

DEMONSTRATION

- 10-class classification for the MNIST handwritten digit (0–9) dataset (with 60K training samples; 10K test samples)
- Fully-connected neural network (MLP)
- 10 hidden layers
- Comparison of training and test accuracies (after 100 epochs)





Poster #78

Paper: **http://proceedings.mlr.press/v97/zeng19a.html**

GitHub: **https://github.com/timlautk/BCD-for-DNNs-PyTorch**

The End

Thank you!