Robust Training in High Dimensions via Block Coordinate Geometric Median Descent

Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit Dhillon, Ufuk Topcu

Robust DNN Training

- (Gross Corruption) Adversary can *inspect all samples and replace* $0 \leq \psi \leq 1/2$ fraction of them with ***arbitrary*** points. If G and B are sets of good and bad points $\alpha = \frac{|B|}{|G|} = \frac{\psi}{\psi - 1} \leq 1$.

- The goal of this paper is to design an efficient first-order optimization method to solve *smooth non-convex optimization* problems with finite-sum structure (ERM formulation of DNN training), under gross corruption, *without any prior knowledge about the malicious samples*.

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{|\mathbb{G}|} \sum_{i \in \mathbb{G}} f_i(\mathbf{x}) \right]$$

Vulnerability of SGD

- SGD: $\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \tilde{\mathbf{g}}^{(t)}, \quad \tilde{\mathbf{g}}^{(t)} = \frac{1}{|\mathcal{D}_t|} \sum_{i \in \mathcal{D}_t} \nabla f_i(\mathbf{x}_t).$

- Breakdown Point: smallest fraction of contamination that must be introduced to cause an estimator to produce arbitrarily wrong estimates.

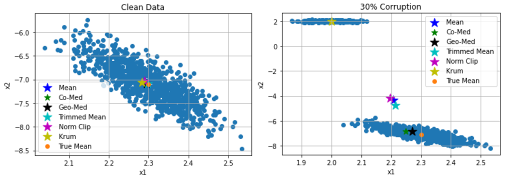
- A single corrupt sample can lead SGD to an *arbitrarily poor solution.* Consider a single malicious gradient: $\mathbf{g}_j^{(t)} = -\sum_{i \in \mathcal{D}_t \setminus j} \mathbf{g}_i^{(t)}$

- SGD has lowest possible *asymptotic breakdown of 0* under gross corruption due to the *linear gradient aggregation* step.

Robust SGD

- Replace Mean with *Robust Mean Estimator.*
- Geometric Median: *Optimal Breakdown point of ½*

$$\mathbf{x}_* = \mathrm{GM}(\{\mathbf{x}_i\}) = \arg\min_{\mathbf{y} \in \mathbb{X}} \left[g(\mathbf{x}) := \sum_{i=1}^{n} \|\mathbf{y} - \mathbf{x}_i\| \right]$$



- GM Descent: $x_{t+1} = \mathrm{x_t} - \eta \hat{g}_t \,, \; \hat{g}_t = \mathrm{GM}(\{\mathrm{g_i}\})$

- Finding ϵ –approximate GM of n points in $\mathrm{R^d}$ requires at least $\mathrm{O}(d/\epsilon^2)$ compute making GM-SGD computationally intractable for DNN training e.g. $d \approx$ 60M Alexnet, $d \approx$ 175B GPT3

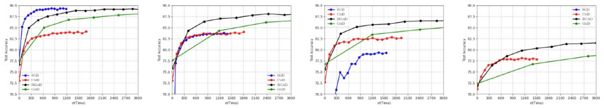
Block Geometric Median Descent

- gradient distribution of overparameterized DNNs are long tailed *implying performing gradient aggregation in low dimensions* should have *little impact* in the downstream optimization task.

- Judiciously *sample an informative block of k dimensions* to perform GM in $\mathrm{R^k}$ (k << d)

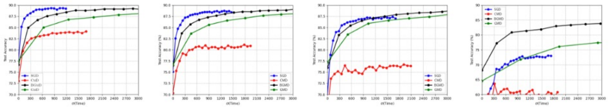
- Keep track of Residual information loss due to dimensionality reduction and add back to gradient estimate in future iterations.

Empirical Evidence



(a) No corruption (b) 10% Corruption (c) 20% Corruption (d) 40% Corruption

Feature Corruption Test accuracy as a ***function of wall clock time*** for training Fashion-MNIST using LeNet (1.16 M params) in presence of impulse noise.



(a) No corruption (b) 10% Corruption (c) 20% Corruption (d) 40% Corruption

Label Corruption. Test accuracy under label noise.

Theoretical Guarantees

Algorithm	Aggregation Operator*	Iteration Complexity†	Breakdown Point†‡
SGD	MEAN(·)	$\boldsymbol{\mathcal{O}(bd)}$	0
(Yang et al., 2019; Yin et al., 2018)	CM(·)	$\mathcal{O}(bd \log b)$	**1/2**
(Wu et al., 2020)	GM(·)	$\mathcal{O}(d\epsilon^{-2} + bd)$	**1/2**
BGmD (This work)	BGM(·)	$\mathcal{O}(k\epsilon^{-2} + bd)$	**1/2**
(Data and Diggavi, 2020)	(Steinhardt et al., 2017)	$\mathcal{O}(db^2 \min(d, b) + bd)$	1/4
(Blanchard et al., 2017)	KRUM(·)	$\mathcal{O}(b^2 d)$	$\lfloor \beta \rfloor$
(Yin et al., 2018)	$\mathrm{CTM}_\beta(\cdot)$	$\mathcal{O}(bd(1 - 2\beta) + bd \log b)$	$\lfloor \beta \rfloor$
(Ghosh et al., 2019; Gupta et al., 2020)	$\mathrm{NC}_\beta(\cdot)$	$\mathcal{O}(bd(2 - \beta) + b \log b)$	$\lfloor \beta \rfloor$

- Non-Convex and Smooth: Suppose f_i corresponding to non-corrupt samples i.e. $i \in G$ are L smooth and non-convex. Run BGMD with ϵ approx. GM oracle and $\gamma = \frac{1}{2L}$ in presence of α corruption for T iterations. Sample any iteration τ uniformly at random then:

$$\mathbb{E}\|\nabla f(\mathbf{x}_\tau)\|^2 = \mathcal{O}\left(\frac{LR_0}{T} + \frac{\sigma^2 \xi^{-2}}{(1-\alpha)^2} + \frac{L^2 \epsilon^2}{|\mathbb{G}|^2 (1-\alpha)^2} \right)$$