Can Decentralized Algorithms Outperform Centralized Algorithms?

A Case Study for Decentralized Parallel Stochastic Gradient Descent

Paper ID 2767

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu

University of Rochester · IBM · ETH Zürich · UC Davis University of California · Tencent AI Lab

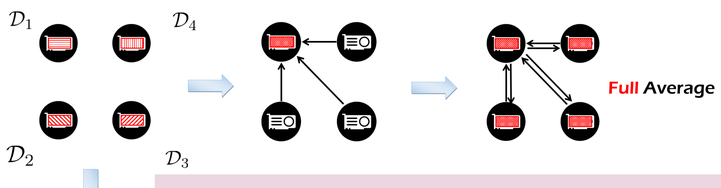
Executive summary

- In large scale machine learning, instead of using only one machine, we distribute data into multiple machine and let them collaborate on solving the following optimization problem:

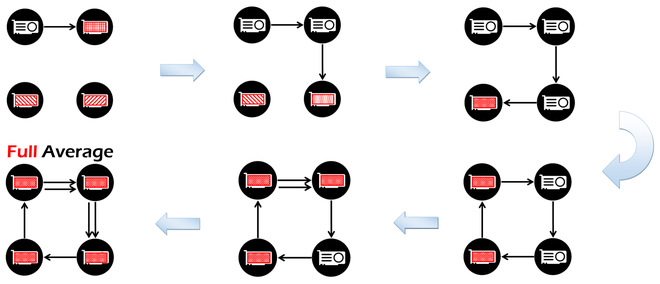
$$\min_{\mathbf{x}} \quad f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{a\sim\mathcal{D}_i} F_i(\mathbf{x};a)}_{=:f_i(\mathbf{x})}$$

- Based on our case study, the decentralized algorithm may not need more iterations to converge than its centralized counterpart
- Decentralized algorithms outperform centralized algorithms for networks with low bandwidth and high latency

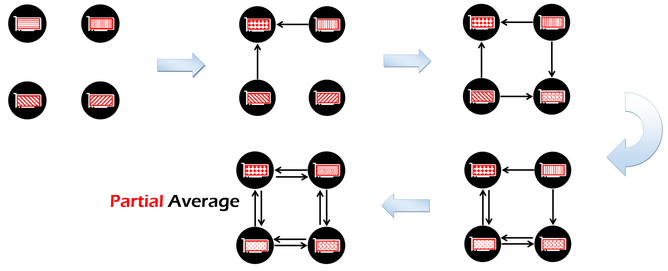
Background: Centralized Method 1 (Parameter Server)



Background: Centralized Method 2 (AllReduce)



Our Proposal: Decentralized Method



	Communication cost per iteration
Single Machine PS	O(N * alpha + NB * beta)
AllReduce	O(log N * alpha + B * beta)
Decentralization (ring)	O(alpha + B * beta)

- alpha: latency per message
- beta: transfer time per byte
- N: # workers
- B: # bytes of the message

How does the decentralized algorithm compare to the centralized one?

Algorithm (Decentralized Parallel SGD)

Shared model · Local model · Local sample

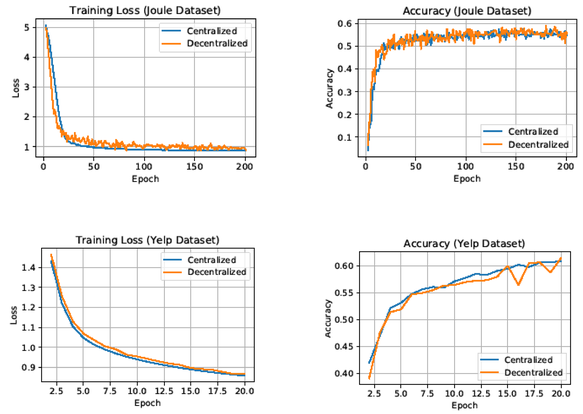
C-PSGD:
$$\mathbf{x} \leftarrow \mathbf{x} - \gamma \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}; a^{(i)})$$

D-PSGD:
$$\begin{bmatrix}\mathbf{x}^{(1)}\\ \mathbf{x}^{(2)}\\ \cdots\\ \mathbf{x}^{(N)}\end{bmatrix} \leftarrow W \begin{bmatrix}\mathbf{x}^{(1)}\\ \mathbf{x}^{(2)}\\ \cdots\\ \mathbf{x}^{(N)}\end{bmatrix} - \gamma \begin{bmatrix} g(\mathbf{x}^{(1)}; a^{(1)})\\ g(\mathbf{x}^{(2)}; a^{(2)})\\ \cdots\\ g(\mathbf{x}^{(N)}; a^{(N)})\end{bmatrix}$$

N individual models

Confusion matrix: symmetric, doubly stochastic
(W1 = 1, W^T1=1, negative, W=W^T)

Evaluation: Proprietary dataset and model (IBM Watson Natural Language Classifier)



Future work

- Asynchronous parallelism for decentralized algorithms
- Investigate new topologies to improve communication efficiency

This Paper

Theoretical Results

Assumptions

- **Lipschitzian** All $f_i(\cdot)$ are with L-Lipschitzian gradient
- **Bounded variance**

$$\mathbb{E}_{a\sim\mathcal{D}_i}\|\nabla F(\mathbf{x};a) - \nabla f_i(\mathbf{x})\|^2 \le \sigma^2,\ \forall i, \forall \mathbf{x}$$
(data variance **within** each worker)

$$\mathbb{E}_{i\sim\mathcal{U}([N])}\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \le \varsigma^2,\ \forall \mathbf{x}$$
(data variance **among** workers)

- **Spectral gap**

$$\rho := \max(|\lambda_2(W)|, |\lambda_n(W)|)^2 < 1$$
(the smaller the better)

Theorem [Convergence rate for D-PSGD]
Choosing the learning rate satisfying $\gamma \lesssim \frac{1-\sqrt{\rho}}{L\sqrt{N}}$, the following rate holds

$$\mathbb{E}\|\nabla f(\bar{\mathbf{x}}_K)\|^2 \lesssim \frac{f(\bar{\mathbf{x}}_0) - f^*}{\gamma K} + \left(\frac{\gamma L}{N} + \frac{\gamma^2 L^2 N}{1-\rho}\right)\sigma^2 + \frac{\gamma^2 L^2 N}{(1-\sqrt{\rho})^2}\varsigma^2$$

average all local x's at iteration K · # of iterations · # of workers

Corollary
Choose the learning rate to be $\gamma = \Theta\left(\left(1+\sqrt{\frac{K}{N}}(\sigma+\varsigma)\right)^{-1}\right)$

The following rate holds

$$\mathbb{E}\|\nabla f(\bar{\mathbf{x}}_K)\|^2 \lesssim \frac{\sigma+\varsigma}{\sqrt{KN}}$$
(consistent with C-PSGD)

if the number of iterations is sufficiently large

$$K \gtrsim \frac{N^5}{(1-\sqrt{\rho})^4(\sigma^2+\varsigma^2)}$$
(imply linear speedup: complexity per worker is $\mathcal{O}\left(\frac{1}{N\epsilon^2}\right)$)

Fully connected network

$$W = \frac{\mathbf{1}\mathbf{1}^\top}{N}, \quad \rho = 0$$

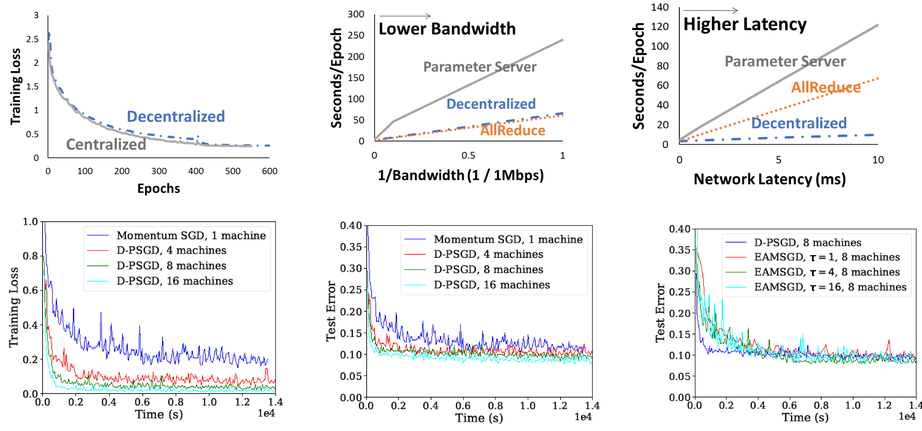
Ring network

$$W = \begin{pmatrix} 1/3 & 1/3 & & & 1/3 \\ 1/3 & 1/3 & 1/3 & & \\ & \ddots & \ddots & \ddots & \\ & & 1/3 & 1/3 & 1/3 \\ 1/3 & & & 1/3 & 1/3 \end{pmatrix}, \quad \rho \approx \left(1 - \frac{16\pi^2}{3N^2}\right)$$

Disconnected network

$$W = \begin{pmatrix} D & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \quad \rho = 1$$

Evaluation: Public dataset and model (CIFAR10/ResNet)



Decentralized algorithms outperform centralized algorithms for networks with low bandwidth and high latency