ECE 8101: Nonconvex Optimization for Machine Learning

Lecture Note 3-2: Decentralized Optimization for Learning

Jia (Kevin) Liu

Assistant Professor
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH, USA

Autumn 2024

Outline

In this lecture:

- Key Idea of Decentralized Nonconvex Optimization for Learning
- Representative Techniques
- Convergence Results

Revisit the Distributed/Federated Learning Problem

- Consider the problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}) \triangleq \min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}),$$

where $f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[F_i(\mathbf{x},\xi_i)]$ is nonconvex

- Distributed/Federated Learning: The "summation" in the mini-batched SGD, which implies a decomposable and distributed implementation:
  - Each stochastic gradient $\nabla f(\mathbf{x}_k,\xi_i)$ can be computed by a "worker/client" i
  - B_k workers can compute such stochastic gradients in parallel
  - A server collects the stochastic gradients returned by workers and aggregate

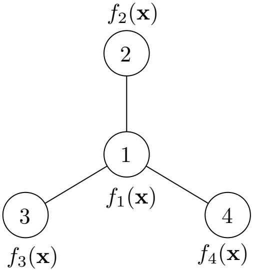
But what if we don't have a server?

Reasons for "Not Having a Server" in Distributed Learning

- Networks Having No Infrastructure
  - Networking protocols based on random access (CSMA, ALOHA, etc.)
  - Ad hoc sensor networks for environmental monitoring
  - Multi-agent systems (autonomous driving, UAVs/UGVs, robotics, etc.)
  - Autonomous swarms on battle field
  - In-situ disaster recovery
- Security/Robustness/Privacy Concerns
  - Avoid single point of failure
  - Avoid having a single target under cyber-attacks
  - Avoid communication/networking bottleneck
  - Need for information privacy preservation
  - Need for decentralization to avoid being controlled by a single party
- Economics Motivations
  - Competition/collaboration among entities
  - Build trust between multiple parties
  - Fairness guarantees
  - Promote personalization and diversity...

Decentralization Optimization for Learning: The Setup

- A network represented by a connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, with $|\mathcal{N}| = N$, $|\mathcal{L}| = L$
- $\mathbf{x} \in \mathbb{R}^d$: a global learning model
- Each node/agent i can only evaluate a local objective function $f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[F_i(\mathbf{x}, \xi_i)]$
- Global objective function is: $\frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x})$
- Goal: To learn the global model collaboratively in a decentralized fashion (i.e., w/o needing any server)



Example: Decentralized Learning in Multi-Agent Systems

- A multi-agent system (drones, robots, soldiers, etc.). Each agent collects high-resolution images $\{\mathbf{u}_{ij}, \mathbf{v}_{ij}, \theta_{ij}\}_{j=1}^{N_i}$
- $\mathbf{u}_{ij}$, $\mathbf{v}_{ij}$, θ_{ij}: pixels, geographical information, ground-truth label of the j-th image at agent i.

- Agents collaboratively perform image regression based on linear model with parameters $\mathbf{x} = [\mathbf{x}_1^\top \; \mathbf{x}_2^\top]^\top$
- This problem can be written as: $\min_{\mathbf{x}} f(\mathbf{x}) \triangleq \min_{\mathbf{x}} \sum_{i=1}^{N} f_i(\mathbf{x})$, where $f_i(\mathbf{x}) \triangleq \frac{1}{N_i} \sum_{j=1}^{N_i} (\theta_{ij} - \mathbf{u}_{ij}^\top \mathbf{x}_1 - \mathbf{v}_{ij}^\top \mathbf{x}_2)^2$

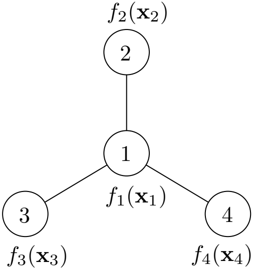
Consensus Reformulation: The First Step

- Goal: To solve the following optimization problem distributively & collaboratively

$$\min_{x \in \mathbb{R}^d} f(\mathbf{x}) = \min_{x \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x})$$

- Clearly, this problem can be rewritten in a consensus form:

$$\min_{\mathbf{x}_i \in \mathbb{R}^d, \forall i} \left\{ \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}_i) \middle| \mathbf{x}_i = \mathbf{x}_j, \forall (i,j) \in \mathcal{L} \right\}$$



The consensus reformulation shares the same spirit with distributed/federated learning that each node maintains a local copy of the global model

Recall What We Did When We Have a Server

- In distributed/federated learning: Each node/client i computes

$$\mathbf{x}_{i,k+1} = \bar{\mathbf{x}}_k - s_k \mathbf{g}_{i,k} \qquad \text{NET-FLEET}$$

where $\bar{\mathbf{x}}_k \triangleq \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_{i,k}$ is the node/client average in iteration k

- This prompts the following natural idea for decentralized learning

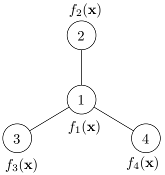
$$\mathbf{x}_{i,k+1} = \text{“Some approximation of } \bar{\mathbf{x}}_k\text{”} - s_k \mathbf{g}_{i,k}$$

- This idea turns out to the foundation of decentralized consensus optimization
  - Note: This is an insight in hindsight. Decentralized consensus optimization traces its roots to the seminal work [Tsitsiklis, Ph.D. Thesis@MIT, 1984]!

A Decentralized Method for Computing Average

Consider a consensus matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ that satisfies:

- Doubly stochastic: $\sum_{i=1}^{N} [\mathbf{W}]_{ij} = \sum_{j=1}^{N} [\mathbf{W}]_{ij} = 1$.
- Sparsity pattern defined by network topology: $[\mathbf{W}]_{ij} > 0$ for $\forall\ (i,j) \in \mathcal{L}$ and $[\mathbf{W}]_{ij} = 0$ otherwise
- Symmetric and hence real eigenvalues in $(-1, 1]$ (thus can be sorted). Moreover, easy to see that $\lambda_{\max} = 1$ with corresponding eigenvector $\mathbf{1}_N$.
- W.l.o.g., denote eigenvalues as $-1 < \lambda_N \leq \cdots \leq \lambda_1 = 1$. Let $\beta \triangleq \max\{|\lambda_2|, |\lambda_N|\}$ (i.e., 2nd-largest eigenvalue in magnitude).



$$\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 3/4 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 \\ 1/4 & 0 & 0 & 3/4 \end{bmatrix}$$

A Decentralized Method for Computing Average

Goal: $\underbrace{\frac{1}{N}\sum_{i=1}^{N} x_{i,0}}$.

1. $k = 0$. Each node has initial value $\mathbf{x}_{i,0}$ to be averaged with other nodes

2. In k-th iteration: Each node shares its local copy to its neighbors.

3. Upon reception of all local copies from its neighbors, each node performs the local updates:

$$\mathbf{x}_{i,k+1} = \sum_{j \in \underline{\mathcal{N}_i}} \overset{w_{ij}}{[\mathbf{W}]_{ij}} \mathbf{x}_{j,k},$$

   where $\underline{\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i,j) \in \mathcal{L}\}}$.

4. Let $k \leftarrow k + 1$ and go to Step 2

A Decentralized Method for Computing Average

- Define a stacked matrix of all local copies:

$$\mathbf{X}_k \triangleq \left[\ \underset{\in\mathbb{R}^d}{\mathbf{x}_{1,k}} \quad \mathbf{x}_{2,k} \quad \cdots \quad \mathbf{x}_{N,k}\ \right] \in \mathbb{R}^{d\times N}.$$

- Then the algorithm in the previous slide can be compactly written as

$$\mathbf{X}_{k+1} = \mathbf{X}_k \mathbf{W}, \qquad X_{k+1}^T = W X_k^T$$

(i.e., $\mathbf{X}_k = \mathbf{X}_0\mathbf{W}^k$). Similar to a discrete-time finite-state Markov chain.

Perron – Frobenius Thm:

- Fact: The stationary distribution of an irreducible aperiodic finite-state Markov chain is uniform iff its transition matrix is doubly stochastic.

- Convergence rate of "averaging": Let $\mathbf{W}^\infty = \lim_{k\to\infty}\mathbf{W}^k$. Then, we have $\mathbf{W}^\infty = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top$. Further, it holds that

Markov chain mixing time

$$= \begin{bmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \vdots \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix}$$

i-th basis vector in $\mathbb{R}^d$

$$\left\|\mathbf{W}^\infty \underset{\begin{bmatrix}\frac{1}{N}\\ \vdots \\ \frac{1}{N}\end{bmatrix}}{\mathbf{e}_i} - \mathbf{W}^k\mathbf{e}_i\right\| \le \beta^k, \quad \forall i \in \{1,\ldots,N\}, k\in\mathbb{N}. \qquad \left(\frac{\lambda_2}{\lambda_1}\right)$$

WTS: $\| \underline{\underline{W}}^{\infty} \cdot \underline{e}_i - \underline{\underline{W}}^{k} \cdot \underline{e}_i \| \leq \beta^k$

Proof: $\| \underline{\underline{W}}^{\infty} \underline{e}_i - \underline{\underline{W}}^{k} \underline{e}_i \| = \| (\underline{\underline{W}}^{\infty} - \underline{\underline{W}}^{k}) \underline{e}_i \|$

Cauchy-Schwartz

$\leq \| \underline{\underline{W}}^{\infty} - \underline{\underline{W}}^{k} \| \cdot \underbrace{\| \underline{e}_i \|}_{=1} = \| \underline{\underline{W}}^{\infty} - \underline{\underline{W}}^{k} \| \qquad (1).$

induced norm.

Note $\underline{\underline{W}}$ is symmetric, then it has real eigenvalues

$$\underline{\underline{W}} = \underline{\underline{U}} \, \underline{\underline{\Lambda}} \, \underline{\underline{U}}^T, \quad \text{where } \underline{\underline{\Lambda}} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

unitary. $\underline{\underline{U}} \, \underline{\underline{U}}^T = \underline{\underline{U}}^T \underline{\underline{U}} = \underline{\underline{I}}$

Moreover, $\lambda_1 = 1$, and the corresp. eigvec. is $\underline{1}_N$.

$$\underline{\underline{W}}^k = \underbrace{\underline{\underline{U}} \, \underline{\underline{\Lambda}} \, \cancel{\underline{\underline{U}}^T} \, \cancel{\underline{\underline{U}}} \, \underline{\underline{\Lambda}} \, \cancel{\underline{\underline{U}}^T} \cdots \cancel{\underline{\underline{U}}} \, \underline{\underline{\Lambda}} \, \underline{\underline{U}}^T}_{k \text{ terms.}} = \underline{\underline{U}} \, \underline{\underline{\Lambda}}^k \underline{\underline{U}}^T = \underline{\underline{U}} \begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_N^k \end{bmatrix} \underline{\underline{U}}^T.$$

Also, $\underline{\underline{W}}^{\infty} = \frac{1}{N} \underline{1}_N \underline{1}_N^T \leftarrow$ rank-1 matrix, it has eigenvalue 1.

and eigenvec. $\underline{1}_N$

Claim: Can rewrite $\underline{\underline{W}}^{\infty} = \underline{\underline{U}} \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \underline{\underline{U}}^T = \sum_{i=1}^{N} \lambda_i \underline{u}_i \underline{u}_i^T$

So, $(1) = \left\| \underline{\underline{U}} \left(\begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} - \begin{bmatrix} 1 & & & \\ & \lambda_2^k & & \\ & & \ddots & \\ & & & \lambda_N^k \end{bmatrix} \right) \underline{\underline{U}}^T \right\|$

$= \left\| \sum_{i=2}^{N} \lambda_i^k \underline{u}_i \underline{u}_i^T \right\| \leq \beta^k \left\| \sum_{i=1}^{N} \underline{u}_i \underline{u}_i^T \right\| = \beta^k \| \underbrace{\underline{\underline{U}} \, \underline{\underline{U}}^T}_{= \underline{\underline{I}}} \| = \beta^k.$ ▨

replace $\lambda_i, \forall i$ by β, factor out β, add $\beta \underline{u}_1 \underline{u}_1^T$

Decentralized Stochastic Gradient Descent (DSGD)

The DSGD algorithm [Nedic and Ozdaglar, TAC'09]:

1. Initialization: Let $k = 1$. Choose initial values for $x_{i,1}$ and step-size α_1.

2. In k-th iteration: Each node sends its local copy to its neighbors.

3. Upon reception of all local copies from its neighbors, each node updates its local copy:

$$\mathbf{x}_{i,k+1} = \underbrace{\sum\nolimits_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij} \mathbf{x}_{j,k}}_{\text{Avg consensus step}} - \underbrace{s_k \nabla F_i(\mathbf{x}_{i,k}, \xi_{i,k})}_{\text{Local SGD step}},$$

   where $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i,j) \in \mathcal{L}\}$.

4. Let $k \leftarrow k + 1$ and go to Step 2

Convergence Results of DSGD

Assumptions:

- $f_i(\cdot)$, $\forall i$ are L-smooth
- Unbiased stochastic gradients: $\mathbb{E}_{\xi_{i,k} \sim \mathcal{D}_i}[\nabla F_i(\mathbf{x}_{i,k}, \xi_{i,k})] = \nabla f_i(\mathbf{x}_{i,k})$, $\forall i, k$
- Bounded local stochastic gradient variance:

$$\mathbb{E}[\|\nabla F_i(\mathbf{x}, \xi) - \nabla f_i(\mathbf{x})\|^2] \leq \sigma^2, \quad \forall i, \mathbf{x}$$

- Bounded gradient dissimilarity: Non-i.i.d.

$$\mathbb{E}_{i \sim \mathcal{U}([n])}[\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2] \leq \zeta^2, \quad \forall \mathbf{x}$$

- Start from $\mathbf{0}$: $\mathbf{X}_0 = \mathbf{0}$ (not necessary, but simplifies the proof w.l.o.g.)

Convergence Results of DSGD

- Let $s_k = s$, $\forall k$, and define two constants:

$$D_1 := \left(\frac{1}{2} - \frac{9s^2L^2N}{(1-\beta)^2 D_2}\right), \text{ and } D_2 := \left(1 - \frac{18s^2}{(1-\beta)^2}NL^2\right)$$

Theorem 1 ([Lian et al. NeurIPS'17]) $[\nabla f(x_{1,k}) \cdots \nabla f(x_{N,k})]_{d\times N}$

Under the stated assumptions, the following convergence rate holds for DSGD:

$= [x_{1,k} \cdots x_{N,k}]_{d\times N}$

$$\frac{1}{K}\left(\frac{1-sL}{2}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\frac{\partial f(\mathbf{X}_k)\mathbf{1}_N}{N}\right\|^2\right] + D_1\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f\left(\frac{\mathbf{X}_k\mathbf{1}_N}{N}\right)\right\|^2\right]\right)$$

$$\leq \frac{f(\mathbf{0}) - f^*}{sK} + \frac{sL}{2N}\sigma^2 + \frac{s^2L^2N\sigma^2}{(1-\beta^2)D_2} + \frac{9s^2L^2N\zeta^2}{(1-\beta)^2 D_2}$$

$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

$\sum_{i=1}^{N}\|\nabla f_{i,k}(x_{i,k})\|^2$

Convergence Results of DSGD

Corollary 2 ([Lian et al. NeurIPS'17])

Under the same assumptions as in Theorem 5, if $s = \frac{1}{2L+\sigma\sqrt{K/N}}$, *then DSGD achieves the following convergence rate:*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f\left(\frac{\mathbf{X}_k\mathbf{1}_N}{N}\right)\right\|^2\right] \leq \frac{8(f(\mathbf{0})-f^*)}{K} + \frac{(8f(\mathbf{0})-8f^*+4L)\sigma}{\sqrt{KN}}.$$

Remark 1

If K *is sufficiently large such that*

$$K \geq \frac{4L^4N^5}{\sigma^2(f(\mathbf{0})-f^*+L)^2}\left(\frac{\sigma^2}{1-\beta^2}+\frac{9\zeta^2}{(1-\beta)^2}\right) \text{ and } K \geq \frac{72L^2N^2}{\sigma^2(1-\beta)^2},$$

then the convergence rate of DSGD is $O\left(\frac{1}{K}+\frac{1}{\sqrt{NK}}\right)$.

Convergence Results of DSGD

Theorem 3 ([Lian et al. NeurIPS'17])

With $s = \frac{1}{2L+\sigma\sqrt{K/N}}$ *and under the same assumptions in Theorem 5, it holds that*

$$\frac{1}{KN}\mathbb{E}\left[\sum_{k=0}^{K-1}\sum_{i=1}^{N}\left\|\overbrace{\frac{\sum_{i'=1}^{N}\mathbf{x}_{i',k}}{N}}^{=\bar{x}_k}-\mathbf{x}_{i,k}\right\|^2\right] \leq Ns^2\frac{A}{D_2},$$

where the constant A *is defined as:*

$$A := \frac{2\sigma^2}{1-\beta^2} + \frac{18\zeta^2}{(1-\beta)^2} + \frac{L^2}{D_1}\left(\frac{\sigma^2}{1-\beta^2} + \frac{9\zeta^2}{(1-\beta)^2}\right) + \frac{18}{(1-\beta)^2}\left(\frac{f(\mathbf{0})-f^*}{sK} + \frac{sL\sigma^2}{2ND_1}\right).$$

Remark 2

The local copies achieve consensus at the rate $O(1/K)$

Prelim:

$$\underline{\underline{X}}_k \triangleq \begin{bmatrix} \vdots & & \vdots \\ x_{1,k} & \cdots & x_{N,k} \\ \vdots & & \vdots \end{bmatrix}_{d\times N}, \quad \underline{\underline{W}} \triangleq \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & & \\ w_{N1} & \cdots & w_{NN} \end{bmatrix}.$$

$$\partial \underline{F}(\underline{\underline{X}}_k, \xi_k) \triangleq \begin{bmatrix} \vdots & & \vdots \\ \nabla F_1(x_{1,k}, \xi_{1,k}) & \cdots & \nabla F_N(x_{N,k}, \xi_{N,k}) \\ | & & | \end{bmatrix}_{d\times N}.$$

Recall: $\underline{x}_{i,k+1} = \sum_{j=1}^{N} \underline{w}_{i,j}\, x_{j,k} - s \nabla F_i(x_{i,k}, \xi_{i,k})$

Concatenating $x_{i,k+1}$, $\forall i$, we have:

$$\begin{bmatrix} \vdots & & \vdots \\ x_{1,k+1} & \cdots & x_{N,k+1} \\ | & & | \end{bmatrix}_{d\times N} = \begin{bmatrix} \vdots & & \vdots \\ x_{1,k} & \cdots & x_{N,k} \\ | & & | \end{bmatrix} \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & & \\ w_{N1} & \cdots & w_{NN} \end{bmatrix} - s \begin{bmatrix} | & & | \\ \nabla F_1(x_{1,k}, \xi_{1,k}) & \cdots & \nabla F_N(x_{N,k}, \xi_{N,k}) \\ | & & | \end{bmatrix}$$

In matrix form:

$$\underline{\underline{X}}_{k+1} = \underline{\underline{X}}_k \underline{\underline{W}} - s \underline{\underline{F}}(\underline{\underline{X}}_k, \xi_k)$$

Right-multiply both sides by $\frac{1}{N}\underline{1}_N$

$$\frac{1}{N}\underline{\underline{X}}_{k+1}\underline{1}_N = \frac{1}{N}\underline{\underline{X}}_k \underbrace{\underline{\underline{W}}\,\underline{1}_N}_{\underline{1}_N} - \frac{s}{N}\partial \underline{\underline{F}}(\underline{\underline{X}}_k, \xi_k)\,\underline{1}_N.$$

$$\Rightarrow \frac{1}{N}\underline{\underline{X}}_{k+1}\underline{1}_N = \frac{1}{N}\underline{\underline{X}}_k \underline{1}_N - \frac{s}{N}\partial \underline{\underline{F}}(\underline{\underline{X}}_k, \xi_k)\,\underline{1}_N.$$

$$\Rightarrow \bar{x}_{k+1} = \bar{x}_k - \underbrace{\frac{s}{N}\sum_{i=1}^{N}\nabla F_i(x_{i,k}, \xi_{i,k})}_{\text{"Grad"}}$$ $\leftarrow$ Dynamic of $\bar{x}_k$.

descent lemma of $\bar{x}$

Quad → $\boxed{B_\Theta}$

Cross → T_1 → Agent Drift "$x_{i,k} - \bar{x}_k$"

Proof of Thm 1:

From descent lemma:

$$\mathbb{E}[f(\bar{x}_{k+1})] \le \mathbb{E}[f(\bar{x}_k)] - \frac{s}{N}\underbrace{\mathbb{E}\left[\nabla f(\bar{x}_k)^T \sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k})\right]}_{\text{Cross}}$$

$$+ \underbrace{\frac{s^2L}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k})\right\|^2\right]}_{\text{Quad}}$$

Consider the Quad term: $\pm \sum_{i=1}^N \nabla f_i(x_{i,k})$

$$\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k})\right\|^2\right] = \mathbb{E}\left[\left\|\frac{1}{N}\left(\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k}) - \sum_{i=1}^N \nabla f_i(x_{i,k})\right) + \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right]$$

$$= \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k}) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right] + \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right]$$

$$+ 2\,\mathbb{E}\left[\left\langle \underbrace{\cancel{\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k}) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})}}_{\text{unbiasedness}} \to 0,\ \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\rangle\right]$$

$$\Rightarrow \mathbb{E}[f(\bar{x}_{k+1})] \le \mathbb{E}[f(\bar{x}_k)] - \frac{s}{N}\mathbb{E}\left[\nabla f(\bar{x}_k)^T \sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k})\right] +$$

$$\underline{\frac{s^2L}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k}) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right]} + \frac{s^2L}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right]$$

$$\hookrightarrow \underline{\frac{s^2L}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \nabla F_i(x_{i,k}, \xi_{i,k}) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})\right\|^2\right]}$$

$$= \frac{s^2 L}{2N^2} \sum_{i=1}^{N} \mathbb{E}\Big[\underbrace{\| \nabla F_i(x_{i,k}, \xi_{i,k}) - \nabla f_i(x_{i,k}) \|^2}_{\le \sigma^2} \Big]$$

$$+ \frac{s^2 L}{N^2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \ne i}}^{N} \mathbb{E}\Big[\underbrace{\cancel{\langle \nabla F_i(x_{i,k}, \xi_{i,k}) - \nabla f_i(x_{i,k}), \nabla F_j(x_{j,k}, \xi_{j,k}) - \nabla f_j(x_{j,k}) \rangle}}_{\text{unbiasedness}} \Big] \to 0$$

Thus:

$$\mathbb{E}[f(\bar{x}_{k+1})] \le \mathbb{E}[f(\bar{x}_k)] - \frac{s}{N} \underbrace{\mathbb{E}\Big[\nabla f(\bar{x}_k)^T \overset{\parallel\ \frac{1}{N}\sum f_i(\cdot\cdot)}{\sum_{i=1}^{N}} \nabla F_i(x_{i,k}, \xi_{i,k}) \Big]}_{a^T b = \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 - \frac{\|a-b\|^2}{2}} + \frac{s^2 L \sigma^2}{2N}$$

$$+ \mathbb{E}\Big[\Big\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(x_{i,k}) \Big\|^2 \Big]$$

$$= \mathbb{E}[f(x_k)] - \frac{s - s^2 L}{2} \underline{\mathbb{E}\Big[\Big\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(x_{i,k}) \Big\|^2 \Big]} - \frac{s}{2} \underline{\mathbb{E}\big[\|\nabla f(\bar{x}_k)\|^2 \big]}$$

$$+ \frac{s^2 L \sigma^2}{2N} + \frac{s}{2} \underbrace{\mathbb{E}\Big[\Big\| \Big[\frac{1}{N} \sum_{i=1}^{N} \nabla F_i(x_{i,k}, \xi_{i,k}) \Big] - \nabla f(\bar{x}_k) \Big\|^2 \Big]}_{T_1}$$

Now, let's bound T_1:

$$\mathbb{E}\Big[\Big\| \Big[\frac{1}{N} \sum_{i=1}^{N} \nabla F_i(x_{i,k}, \xi_{i,k}) \Big] - \underset{\nearrow\ \frac{1}{N}\sum_{i=1}^{N} f_i(\cdot)}{\nabla f(\bar{x}_k)} \Big\|^2 \Big] = \frac{1}{N^2} \mathbb{E}\Big[\Big\| \sum_{i=1}^{N} \big(\nabla f_i(\bar{x}_k) - \nabla F_i(x_{i,k}, \xi_{i,k}) \big) \Big\|^2 \Big]$$

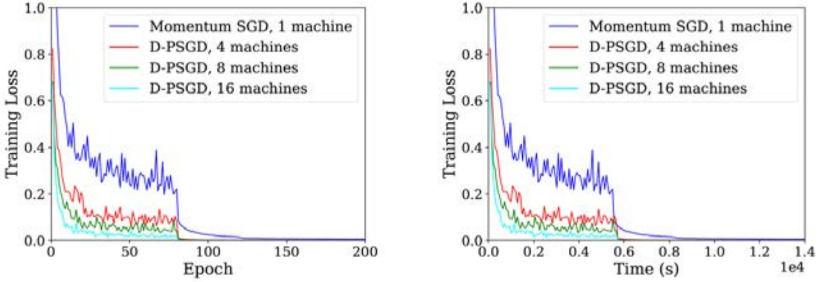
$$\le \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\big[\| \nabla f_i(\bar{x}_k) - \nabla F_i(x_{i,k}, \xi_{i,k}) \|^2 \big]$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\Big[\underbrace{\| \nabla f_i(\bar{x}_k) - \nabla f_i(x_{i,k}) \|^2}_{\text{Lipschitz: } \le L^2 \|\bar{x}_k - x_{i,k}\|^2} \Big] + \mathbb{E}\Big[\underbrace{\| \nabla f_i(x_{i,k}) - \nabla F_i(x_{i,k}, \xi_{i,k}) \|^2}_{\le \sigma^2} \Big]$$

$$\le \frac{L^2}{N} \sum_{i=1}^{N} \underbrace{\mathbb{E}\big[\| \bar{x}_k - x_{i,k} \|^2 \big]}_{?\ \text{"Agent drift"}} + \sigma^2$$

Numerical Results of DSGD

- Linear Speedup Effect
  - 32-layer residual network and CIFAR-10 dataset
  - Up to 16 machines; each machine includes two Xeon E5-2680 8-core processors and a NVIDIA K20 GPU



(a) Iteration vs Training Loss

(b) Time vs Training Loss

A "Tug of War" in DSGD

Revisit the DSGD algorithm:

- The algorithmic update at each agent is:

$$\mathbf{x}_{i,k+1} = \underbrace{\sum_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij} \mathbf{x}_{j,k}}_{\text{Avg consensus step}} - \underbrace{s_k \nabla F_i(\mathbf{x}_{i,k}, \xi_{i,k})}_{\text{Local SGD step}},$$

where $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i,j) \in \mathcal{L}\}$.

The average consensus step and the local SGD step "conflict" with each other. Can we do better?

The Gradient Tracking Idea

Gradient-Tracking DSGD: [Lu et al., DSW'19]:

1. Initialization: Let $k = 1$. Choose initial values for $\mathbf{x}_{i,1}$ and step-size s_1. Define an auxiliary variable $\mathbf{y}_{i,k}$ with $\mathbf{y}_{i,1} = \nabla F_i(\mathbf{x}_{i,1}, \xi_{i,1})$.

2. In k-th iteration: Each node sends its local copy to its neighbors.

3. Upon reception of all local copies from its neighbors, each node updates its local copy:

$$\mathbf{x}_{i,k+1} = \sum\nolimits_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij} \mathbf{x}_{j,k} - s_k \mathbf{y}_{i,k},$$
$$\mathbf{y}_{i,k+1} = \sum\nolimits_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij} \mathbf{y}_{j,k} + \nabla F_i(\mathbf{x}_{i,k+1}, \xi_{i,k+1}) - \nabla F_i(\mathbf{x}_{i,k}, \xi_{i,k}).$$

   where $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i,j) \in \mathcal{L}\}$.

4. Let $k \leftarrow k + 1$ and go to Step 2

Convergence Results for GT-DSGD

- Define $P^k \triangleq \mathbb{E}[f(\bar{\mathbf{x}}_k)] + \mathbb{E}[\|\mathbf{x}_k - \mathbf{1}_N \otimes \bar{\mathbf{x}}_k\|^2] + Q\mathbb{E}[\|\mathbf{y}_k - \mathbf{1}_N \otimes \bar{\mathbf{y}}_k\|^2]$

Theorem 4 (Convergence of Agent-Average [Lu et al. DSW'19])

If the step-size is set to $\frac{C_0}{\sqrt{T}}$, then it holds that:

$$C_1\mathbb{E}[\|\bar{\mathbf{y}}_k\|^2] + \frac{C_2}{C_0}\mathbb{E}[\|\mathbf{x}_t - \mathbf{1}_N \otimes \bar{\mathbf{x}}_t\|^2] \leq \left(\frac{P^0 - P^*}{C_0} + C_4 C_0 \sigma^2\right)\frac{1}{\sqrt{T}}$$

Convergence Results for GT-GSGD

Theorem 5 (Contration of Consensus Gap [Lu et al. DSW'19])

Let ρ *be some constant such that* $(1+\rho)\beta^2 < 1$*. It holds that:*

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{1}_N \otimes \bar{\mathbf{x}}_{k+1}\|] \leq{}& (1+\rho)\beta^2 \mathbb{E}[\|\mathbf{x}_k - \mathbf{1}_N \otimes \bar{\mathbf{x}}_k\|^2] \\
&+ 3\left(1+\frac{1}{\rho}\right) s^2 \mathbb{E}[\|\mathbf{y}_k - \mathbf{1}_N \otimes \bar{\mathbf{y}}_k\|^2] + 6\left(1+\frac{1}{\rho}\right) s^2 \kappa\sigma^2, \\
\mathbb{E}[\|\mathbf{y}_k - \mathbf{1}_N \otimes \bar{\mathbf{y}}_k\|] \leq{}& \frac{4L^2 s^2}{N}\left(1+\frac{1}{\beta}\right)^2 \|\bar{\tilde{\mathbf{y}}}_k\|^2 \\
&+ \left(\frac{L^2}{N^2}\beta^2(1+\rho)\left(1+\frac{1}{\rho}\right) + \frac{4L^2}{N^2}\left(1+\frac{1}{\rho}\right)^2\right) \mathbb{E}[\|\mathbf{x}_k - \mathbf{1}_N \otimes \bar{\mathbf{x}}_k\|^2] \\
&+ \left((1+\rho)\beta^2 + \frac{4L^2 s^2}{N^2}\left(1+\frac{1}{\rho}\right)^2\right) \mathbb{E}[\|\mathbf{y}_k - \mathbf{1}_N \otimes \bar{\mathbf{y}}_k\|^2] \\
&\frac{4L^2 s^2}{N^2}\left(1+\frac{1}{\rho}\right)^2 \kappa\sigma^2.
\end{aligned}
$$

Next Class

Zeroth-Order Methods