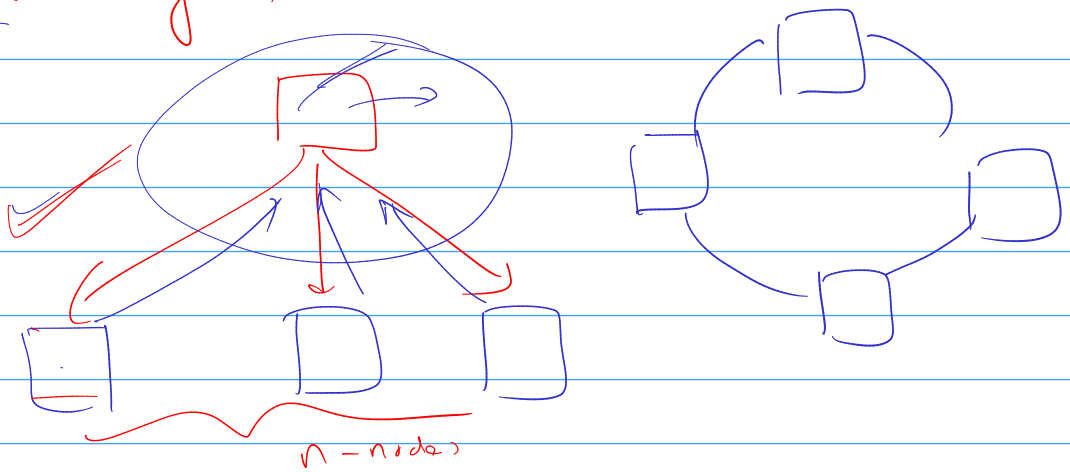
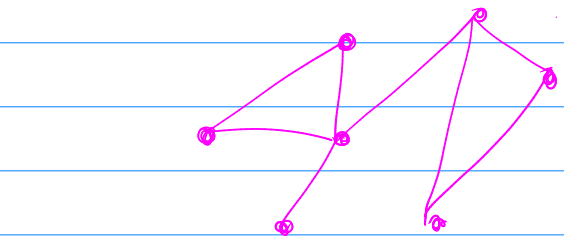
* Parameter Server Approach ✓
* Ring All Reduce Algorithm ✓

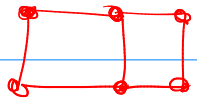


* Decentralized-SGD (Stochastic Gradient Descent) Algorithm

Accelerate convergence speed by choosing the right topology

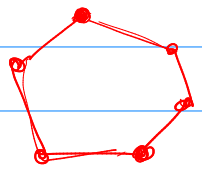


d=5 Line graph

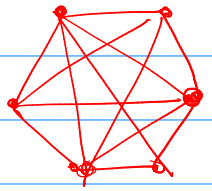


Grid Graph

d=3



d=3 Ring Graph



Complete Graph

d=1

Parallel SGD:

↳ Single-node training (SGD):

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x^{(k)})\|^2\right] = \mathcal{O}\left(\frac{\sigma}{\sqrt{T}}\right)$$

↳ n-node parallel training:

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x^{(k)})\|^2\right] = \mathcal{O}\left(\frac{\sigma}{\sqrt{nT}}\right)$$

To achieve ε-accurate solution:

Single node — $\frac{\sigma}{\sqrt{T}} = \varepsilon \Rightarrow T = \frac{\sigma^2}{\varepsilon^2}$

n-node (parallel) — $\frac{\sigma}{\sqrt{nT}} = \varepsilon \Rightarrow T = \frac{\sigma^2}{n\varepsilon^2}$

Linear speedup with the number of workers

Parallel SGD has linear speedup!

$$g_i^{(k)} = \nabla F(x^{(k)}; \xi_i^{(k)}) \quad \text{(Local computation)}$$

$$x^{(k+1)} = x^{(k)} - \frac{\gamma}{n} \sum_{i=1}^{n} g_i^{(k)} \quad \text{(Global communication)}$$

Algorithm	Bandwidth Cost	Latency	Total cost
RingAllReduce	$\Omega(1)$ ✓	$\Omega(n)$	$\Omega(n+1)$
Parameter Server	$\Omega(n)$	$\Omega(1)$	$\Omega(n+1)$

In both approaches, total cost is $\Omega(n)$.

* D-SGD (Decentralized SGD)

No global synchronization

D-SGD = local SGD update + partial averaging

$$x_i^{(k+1/2)} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)})$$

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} W_{ij} x_j^{(k+1/2)}$$

Per-iteration communication cost?

$\Omega(d_{max}) << \Omega(n)$

↳ largest degree

$\Omega(1)$ is the communication overhead for line graphs.

However, D-SGD has slower convergence.

Convergence speed depends on spectral gap or Fiedler value.

Spectral gap $\rho = \|W - \frac{1}{n}\mathbf{1}\mathbf{1}^T\|_2$

If W is doubly-stochastic $\Rightarrow \rho \in (0,1)$

Well connected, $\rho \to 0$

Sparsely connected, $\rho \to 1$

for eg: for ring graphs $\rho = O\left(1 - \frac{1}{n^2}\right)$

Theorem:

A1: $F(x;\xi_i)$ is L-smooth in terms of x.

A2: Local stochastic gradients are unbiased, and have bounded variance.

$$\mathbb{E}[g_i^{(k)}] = \nabla f_i(x^{(k)})$$

$$\mathbb{E}\left[\|g_i^{(k)} - \nabla f_i(x^{(k)})\|^2\right] \le \sigma^2$$

where, $\nabla f_i(x) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[\nabla F(x;\xi_i)]$

A3: Local stochastic gradients are independent of each other.

A4: Data heterogeneity is bounded.

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla f(x)\|^2 \le b^2$$

where, $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x)$ ($b=0$ if iid)

Under the above assumptions and $\gamma = O\left(\frac{1}{\sqrt{T}}\right)$, we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x^{(k)})\|^2\right] = O\left(\frac{\sigma}{\sqrt{nT}}\right) + \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}} + \frac{\rho^{2/3}b^{2/3}}{T^{2/3}(1-\rho)^{2/3}}$$

$\to 0$ if iid

Extra overhead because of data heterogeneity

* When data is iid:

P-SGD: $\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}}\right)$

D-SGD: $\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}}\right) + \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}}$ → $T \to \infty$

* Transient iterations: # iterations before D-SGD achieves linear speedup

↳ Measures the convergence gap b/w P-SGD and D-SGD

Transient Iteration Complexity

iid data: $\frac{\sigma}{\sqrt{nT}} \geq \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}} \Rightarrow T = \Omega\left(\frac{\rho^4 n^3}{(1-\rho)^2}\right)$

non-iid data: $\frac{\sigma}{\sqrt{nT}} \geq \frac{\rho^{2/3} b^{2/3}}{T^{2/3}(1-\rho)^{2/3}} \Rightarrow T = \Omega\left(\frac{\rho^4 n^3}{(1-\rho)^4}\right)$

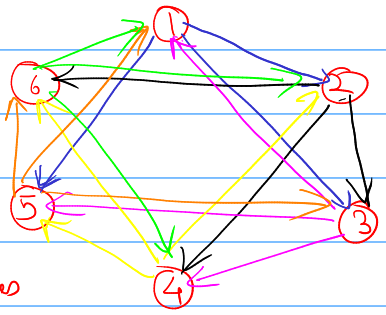
How can we make D-SGD practical?

↳ Removing data heterogeneity

↳ Static exponential graphs are provably efficient

↳ Directed graph

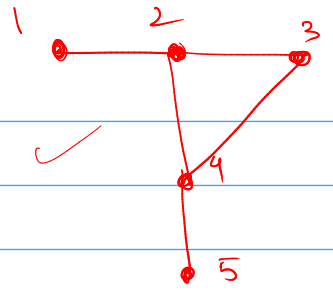
$\lfloor \log_2 n \rfloor$



Static exponential graph with 6 nodes

$$W_{ij} = \begin{cases} \frac{1}{\lceil \log_2 n \rceil + 1} & \text{if } \log_2(\text{mod}(j-i,n)) \text{ is an integer or } i=j \\ 0 & \text{else.} \end{cases}$$

Resilient distributed optimization in presence of Byzantine adversary.



$$f_i(x) := \frac{1}{2}(x-i)^2$$

$$\nabla f_i(x) = (x-i)$$

$$\min_x \sum f_i(x) = 3$$

Agent 5 is byzantine:

$$\{ x_5(k+1) = x_5(k) - \eta \nabla f_5(x_5(k))$$

Every other agent in the network runs the DGD algorithm:

$$i \in \{1,2,3,4\} \begin{cases} x_i(k+1/2) = \sum_{j=1}^{5} W_{ij}\, x_j(k) \\ x_i(k+1) = x_i(k+1/2) - \eta \nabla f_i(x_i(k+1/2)) \end{cases}$$

ClippedGossip Algorithm:

$$\text{Clip}(z,\tau) := \min\left(1, \frac{\tau}{\|z\|}\right) \cdot z$$

$$x_i(k+1/2) = \sum_{j=1}^{5} W_{ij}\left(x_i(k) + \text{Clip}(x_j(k) - x_i(k), \tau)\right)$$