北京大学
PEKING UNIVERSITY

Optimization for Deep Learning

Lecture 13-2: Decentralized Communication and Average Consensus

Kun Yuan

北京大学
PEKING UNIVERSITY

Part 1

Vanilla distributed learning

Distributed learning

北京大学 PEKING UNIVERSITY

- Training deep neural networks typically requires **massive** datasets; efficient and scalable distributed optimization algorithms are in urgent need

- A network of n nodes (devices such as GPUs) collaborate to solve the problem:

$$\min_{x\in\mathbb{R}^d} \quad f(x)=\frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where} \quad f_i(x)=\mathbb{E}_{\xi_i\sim D_i} F(x;\xi_i)$$

  - Each component $f_i : \mathbb{R}^d \to \mathbb{R}$ is local and private to node i
  - Random variable ξ_i denotes the local data that follows distribution D_i
  - Each local distribution D_i is different; data heterogeneity exists

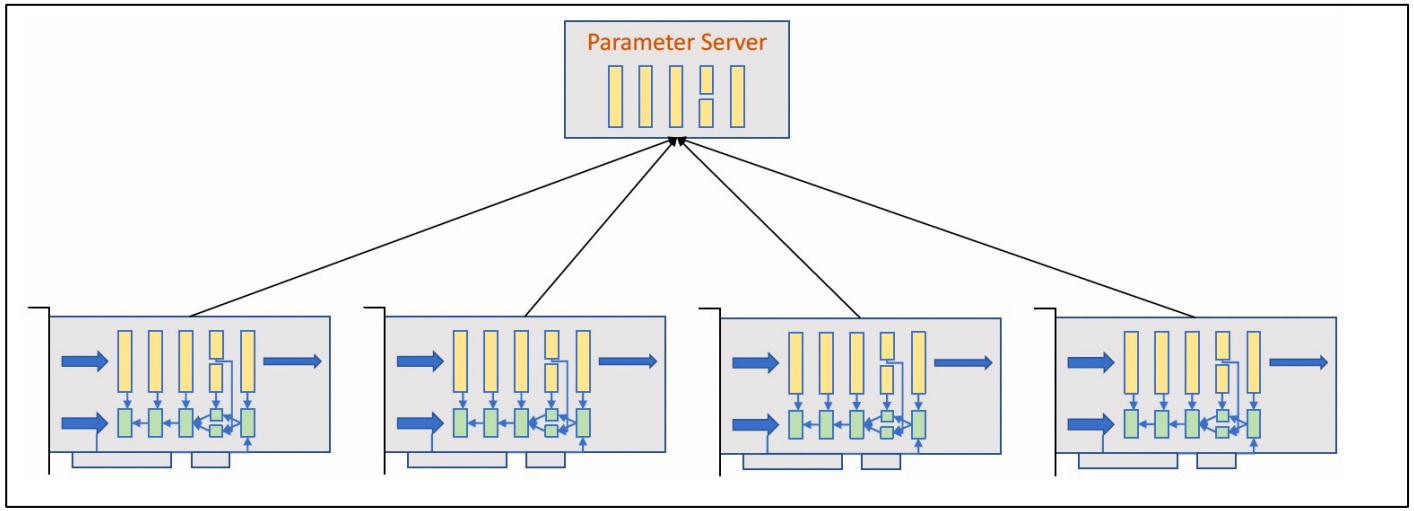
Vanilla parallel stochastic gradient descent (PSGD)

$$g_i^{(k)} = \nabla F(x^{(k)}; \xi_i^{(k)}) \qquad \text{(Local compt.)}$$

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

- Each node i samples data $\xi_i^{(k)}$ and computes gradient $\nabla F(x^{(k)}; \xi_i^{(k)})$
- All nodes synchronize (i.e. globally average) to update model x per iteration

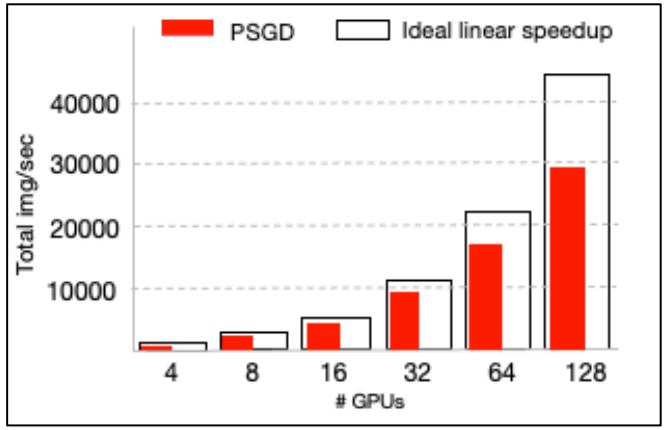
Vanilla parallel stochastic gradient descent (PSGD)



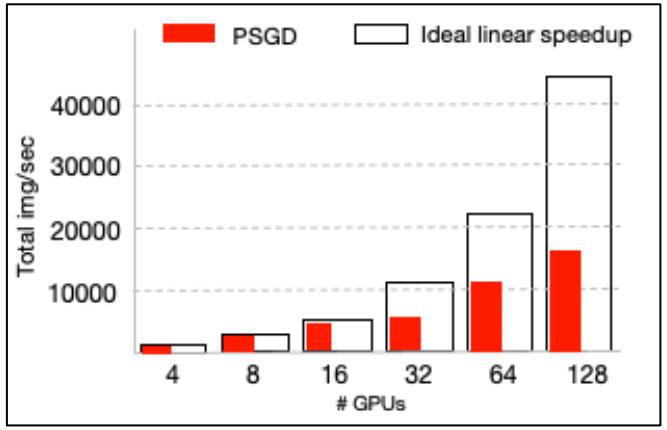
- Global average incurs $O(n)$ comm. overhead; **proportional to network size n**
- When network size n is large, PSGD suffers severe communication overhead

PSGD cannot achieve linear speedup due to comm. overhead

- PSGD cannot achieve ideal linear speedup in throughput due to comm. overhead
- Larger comm-to-compt ratio leads to worse performance in PSGD



Small comm.-to-compt. ratio



Large comm.-to-compt. ratio

- How can we accelerate PSGD? **Decentralized SGD is a promising paradime.**

B. Ying, K. Yuan, H. Hu, Y. Chen and W. Yin, “BlueFog: Make decentralized algorithms practical for optimization and deep learning”, arXiv: 2111. 04287, 2021

北京大学
PEKING UNIVERSITY

Part 2

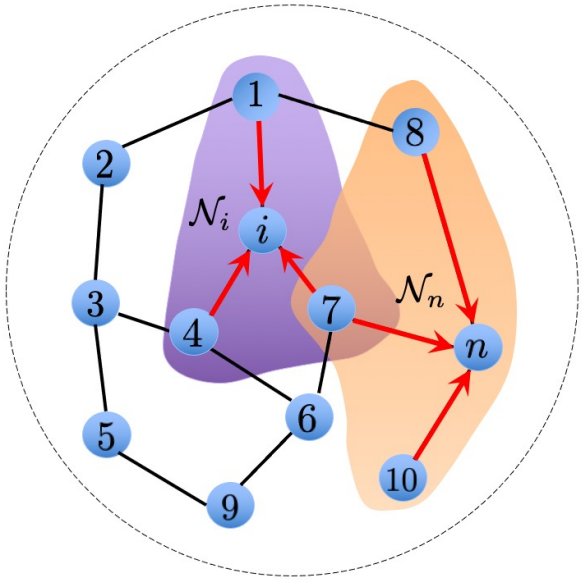
Decentralized learning with partial averaging

Decentralized SGD (DSGD)

- To break $O(n)$ comm. overhead, we replace global average with partial average

$$x_i^{(k+\frac{1}{2})} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) \quad \text{(Local update)}$$

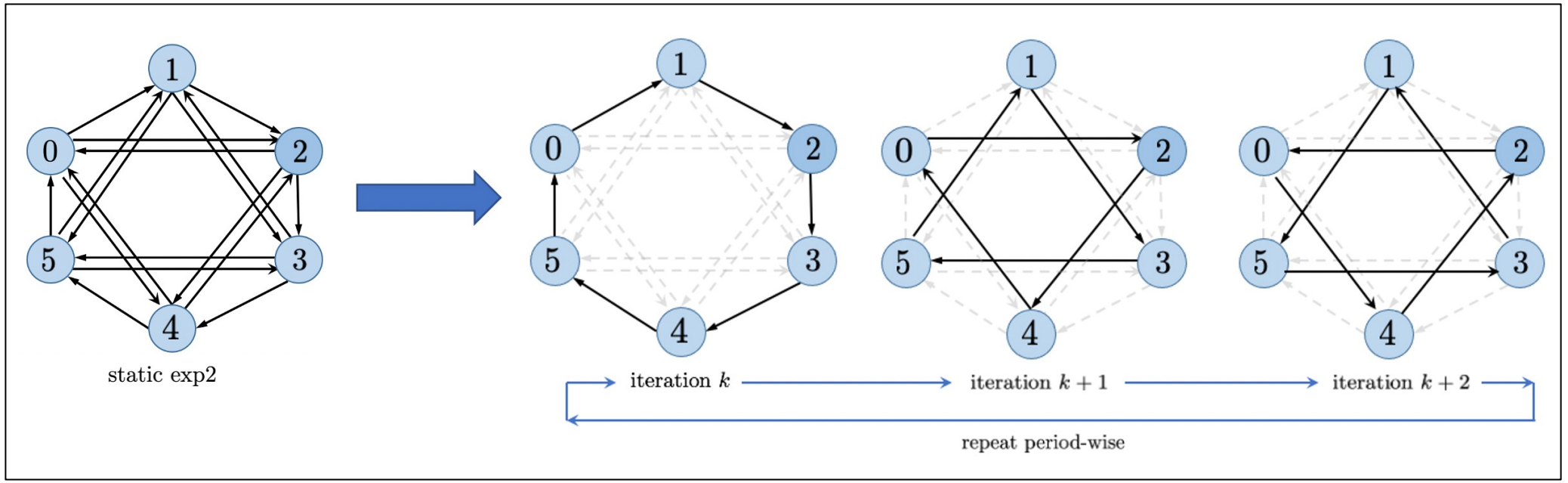
$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k+\frac{1}{2})} \quad \text{(Partial averaging)}$$



- DSGD = local SGD update + partial averaging [LS08]
- $\mathcal{N}_i$ is the set of neighbors at node i ; w_{ij} scales information from j to i and satisfies $\sum_{j \in \mathcal{N}_i} w_{ij} = 1$
- Incurs $O(d_{\max})$ comm. overhead per iteration where $d_{\max} = \max_i \{|\mathcal{N}_i|\}$ is the graph maximum degree

DSGD is more communication-efficient than PSGD

- Incurs $O(1)$ comm. overhead on **sparse** topologies; much less than global average $O(n)$



One-peer exponential graph incurs O(1) comm. overhead

B. Ying, K. Yuan, Y. Chen, H. Hu, and W. Yin, "Exponential Graph is Provably Efficient for Decentralized Deep Training", NeurIPS 2021

DSGD is more communication-efficient than PSGD

北京大学 PEKING UNIVERSITY

- A real experiment on a 256-GPUs cluster [CYZ+21]

Model	Ring-Allreduce	Partial average
ResNet-50 (25.5M)	278 ms	150 ms

Table. Comparison of per-iter comm. time in terms of runtime with 256 GPUs

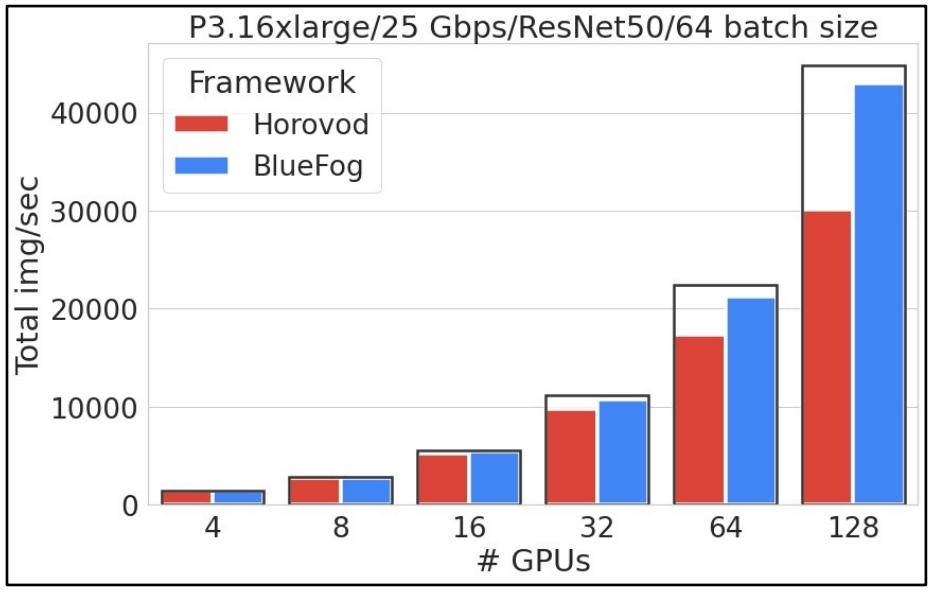
- DSGD saves more communications per iteration for larger models

[CYZ+21] Y. Chen*, K. Yuan*, Y. Zhang, P. Pan, Y. Xu, and W. Yin, ``Accelerating Gossip SGD with Periodic Global Averaging", ICML 2021

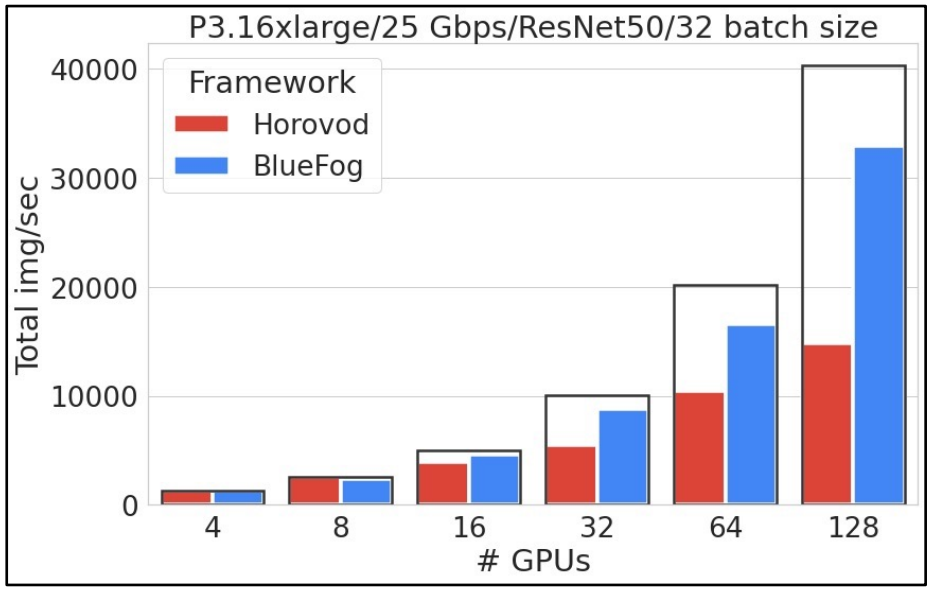
DSGD is more communication-efficient than PSGD

北京大学 PEKING UNIVERSITY

- DSGD (BlueFog) has **better linear speedup** than PSGD (Horovod) due to its small comm. overhead



Small comm.-to-compt. ratio



Large comm.-to-compt. ratio

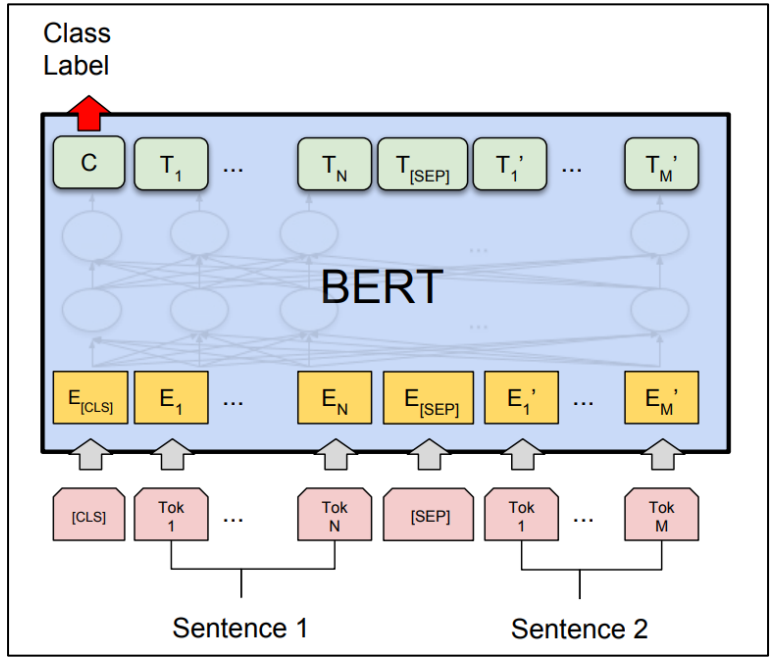
B. Ying, K. Yuan, H. Hu, Y. Chen and W. Yin, “BlueFog: Make decentralized algorithms practical for optimization and deep learning”, arXiv: 2111. 04287, 2021

DSGD achieves better linear speedup

nodes topology	4(4x8 GPUs) acc.	 time	8(8x8 GPUs) acc.	 time	16(16x8 GPUs) acc.	 time	32(32x8 GPUs) acc.	 time
P-SGD	76.32	11.6	76.47	6.3	76.46	3.7	76.25	2.2
Decentralized	76.34	11.1	76.52	5.7	76.47	2.8	76.27	1.5

DSGD shows very impressive linear speedup performance and saves more time than PSGD!

Experiments in deep learning (language modeling)



Model: BERT-Large (330M parameters)

Dataset: Wikipedia (2500M words) and BookCorpus (800M words)

Hardware: 64 GPUs

Table. Comparison in loss and training time [CYZ+21]

Method	Final Loss	Wall-clock Time (hrs)
P-SGD	1.75	59.02
D-SGD	1.77	30.4

[CYZ+21] Y. Chen*, K. Yuan*, Y. Zhang, P. Pan, Y. Xu, and W. Yin, ``Accelerating Gossip SGD with Periodic Global Averaging", ICML 2021

北京大学
PEKING UNIVERSITY

Part 3

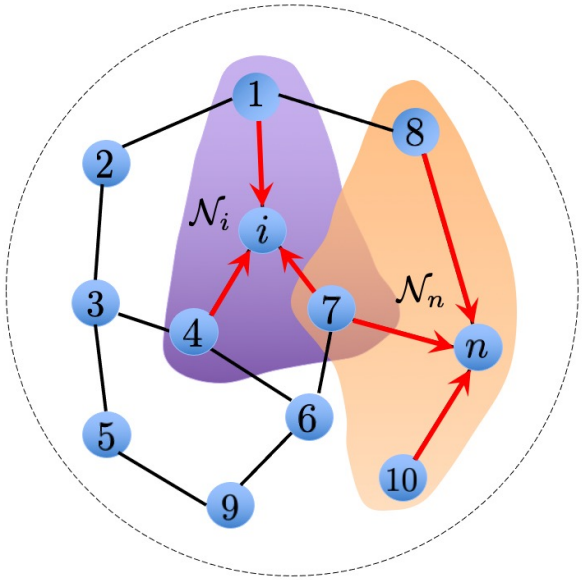
Average consensus

Recall the decentralized SGD approach

- To break $O(n)$ comm. overhead, we replace global average with partial average

$$x_i^{(k+\frac{1}{2})} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) \quad \text{(Local update)}$$

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k+\frac{1}{2})} \quad \text{(Partial averaging)}$$



- DSGD = local SGD update + partial averaging [LS08]
- $\mathcal{N}_i$ is the set of neighbors at node i ; w_{ij} scales information from j to i and satisfies $\sum_{j \in \mathcal{N}_i} w_{ij} = 1$
- DSGD relieves on the partial average $x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k)}$

Why does decentralized learning work?

北京大学 PEKING UNIVERSITY

- The **average consensus** algorithm:

$$x_i^{(0)} = z_i, \quad x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k)}, \quad \forall\, i \in [n]$$

- We stack all weights into a weight matrix $W = [w_{ij}]_{i=1,j=1}^{n} \in \mathbb{R}^{n \times n}$ and rewrite average consensus as

$$x^{(k+1)} = W x^{(k)} \quad \text{where} \quad x^{(0)} = z \in \mathbb{R}^n$$

- We next prove that average consensus converges to global average asymptotically!

$$\lim_{k \to \infty} x_i^{(k)} = \bar{z} = \frac{1}{n} \sum_{j=1}^{n} z_j, \quad \forall\, i \in [n]$$

Weight matrix: assumptions

Assumption 1 [Graph-induced weight matrix]

Each entry w_{ij} in W reflects the connectedness of the edge (j, i), i.e.

$$w_{ij} \begin{cases} > 0 & \text{if node } j \text{ can talk to } i \\ = 0 & \text{if node } j \text{ cannot talk to } i \end{cases}$$

Assumption 2 [Doubly-stochasticity]

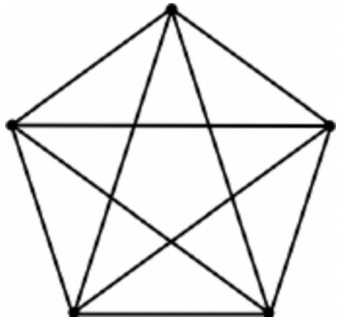
The weight matrix W is primitive and doubly-stochastic, i.e.

$$W\mathbf{1}_n = \mathbf{1}_n \quad \text{and} \quad \mathbf{1}_n^T W = \mathbf{1}_n^T$$

Weight matrix: illustrations

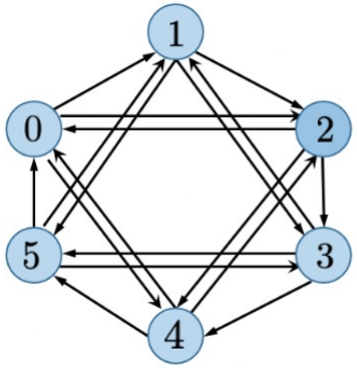
北京大学 PEKING UNIVERSITY

Fully-connected weight matrix



$$W = \frac{1}{5}\mathbb{1}\mathbb{1}^T = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

Exponential weight matrix



$$W = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Weight matrix: properties

- Weight matrix W reflects the connectivity of the underlying decentralized network

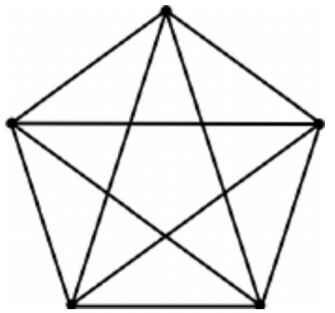
Lemma 1 Under Assumptions 1-2, it holds that $\|W - \mathbf{1}_n \mathbf{1}_n^T/n\| \leq \rho \in [0, 1)$

- Quantity ρ measures the gap between W and the fully-connected weight matrix $\mathbf{1}_n \mathbf{1}_n^T/n$

- For well-connected network, we have $\rho \to 0$

  For sparsely-connected network, we have $\rho \to 1$

- We define the spectral gap of W as $1 - \rho$



$$W = \frac{1}{5}\mathbb{1}\mathbb{1}^T = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

Convergence rate of the average consensus algorithm

北京大学 PEKING UNIVERSITY

- Recall the average consensus algorithm

$$x^{(k+1)} = Wx^{(k)} \quad \text{where} \quad x^{(0)} = z \in \mathbb{R}^n$$

- Partial average converges to the global average as follows

$$\begin{aligned}
\|x^{(k)} - \mathbf{1}_n \mathbf{1}_n^T z/n\| &= \|(W^k - \mathbf{1}_n \mathbf{1}_n/n)z\| \\
&\leq \|W^k - \mathbf{1}_n \mathbf{1}_n/n\|\|z\| \\
&= \|(W - \mathbf{1}_n \mathbf{1}_n/n)^k\|\|z\| \qquad \text{(Assumption 1)} \\
&\leq \|W - \mathbf{1}_n \mathbf{1}_n/n\|^k \|z\| \leq \rho^k \|z\| \qquad \text{(Lemma 1)}
\end{aligned}$$

- Quantity ρ measures how fast the average consensus converges to the global average; also known as **consensus rate**; well-connected network produces small ρ and hence achieves faster consensus

Convergence rate of the average consensus algorithm

北京大学 PEKING UNIVERSITY

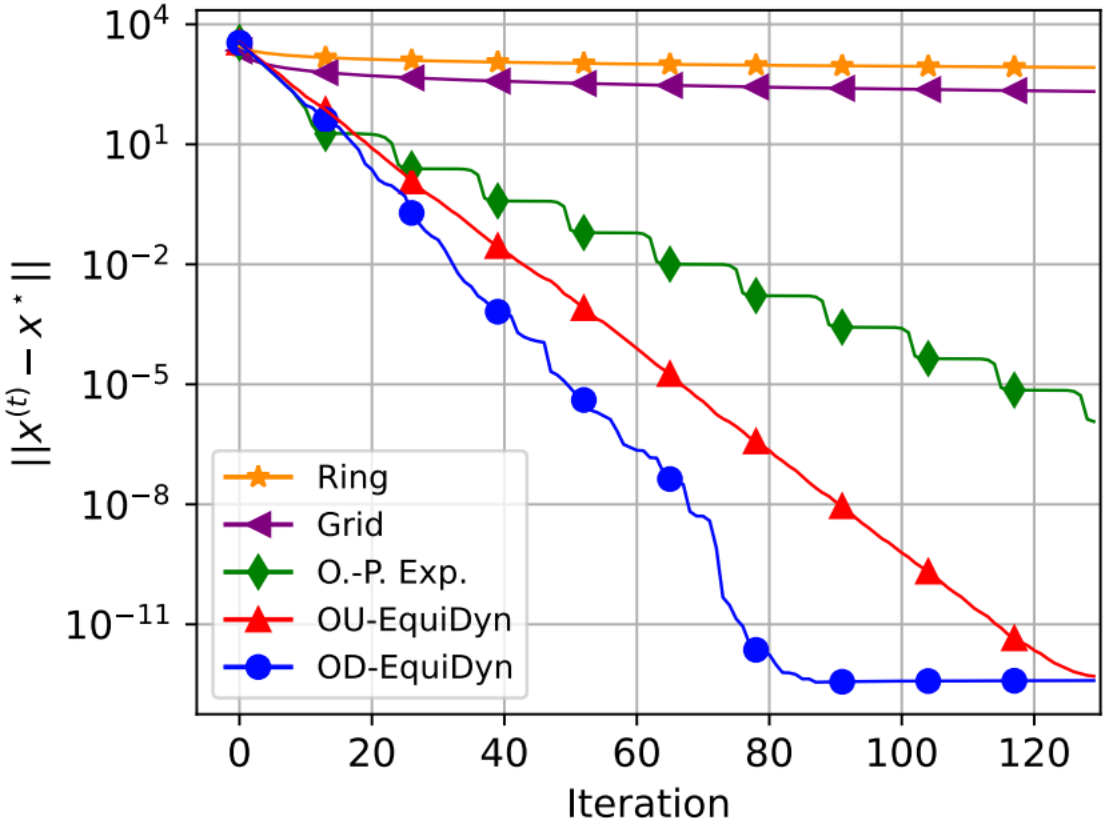
Theorem 1 [Convergence rate of average consensus]

Under Assumptions 1-2, the average consensus algorithm will converge exponentially fast to the global average, i.e,

$$\|x^{(k)} - \mathbf{1}_n \mathbf{1}_n^T z/n\| \leq \rho^k \|z\|$$

Convergence rate of the average consensus algorithm

Network topology	Consensus rate ρ
Ring	$O(1-\frac{1}{n^2})$
Grid	$O(1-\frac{1}{n\ln(n)})$
Torus	$O(1-\frac{1}{n})$
ExpoGraph	$O(1-\frac{1}{\ln(n)})$
GeoMedian	$O(1-\frac{\ln(n)}{n})$
Erdos-Renyi	$O(1)$
EquiGraph	$O(1)$



Simulation results are from [Song et. al., NeurIPS 2022]

北京大学
PEKING UNIVERSITY

Part 4

Dynamic Average consensus

Dynamic average consensus

北京大学 PEKING UNIVERSITY

- Recall the average consensus algorithm

$$x^{(k+1)} = Wx^{(k)} \quad \text{where} \quad x^{(0)} = z \in \mathbb{R}^n$$

- Average consensus tracks the global average of the **static** input $z \in \mathbb{R}^n$

- If the input is **changing** with iteration, can we sill be able to track the global average of the dynamic $z^{(k)}$?

- Dynamic average consensus

$$x^{(k+1)} = Wx^{(k)} + z^{(k+1)} - z^{(k)} \quad \text{where} \quad x^{(0)} = z^{(0)}$$

Dynamic average consensus

$$x^{(k+1)} = Wx^{(k)} + z^{(k+1)} - z^{(k)} \quad \text{where} \quad x^{(0)} = z^{(0)}$$

- Reduce to vanilla average consensus when $z^{(k+1)} = z^{(k)} = \cdots = z^{(0)}$

- Left-multiplying $(1/n)\mathbf{1}_n^T$ to both sides of the above recursion, we have

$$\begin{aligned}(\frac{1}{n}\sum_{j=1}^{n} x_j^{(k+1)})\mathbf{1}_n &= (\frac{1}{n}\sum_{j=1}^{n} x_j^{(k)})\mathbf{1}_n + (\frac{1}{n}\sum_{j=1}^{n} z_j^{(k+1)})\mathbf{1}_n - (\frac{1}{n}\sum_{j=1}^{n} z_j^{(k)})\mathbf{1}_n \\ &= (\frac{1}{n}\sum_{j=1}^{n} z_j^{(k+1)})\mathbf{1}_n\end{aligned}$$

This implies that the global average of $x^{(k)}$ tracks the global average of the dynamic input $z^{(k)}$

Dynamic average consensus

$$x^{(k+1)} = Wx^{(k)} + z^{(k+1)} - z^{(k)} \quad \text{where} \quad x^{(0)} = z^{(0)}$$

- Now we define $\bar{x}^{(k)} = (\frac{1}{n}\sum_{j=1}^{n} x_j^{(k)})\mathbf{1}_n \in \mathbb{R}^n$ and $\bar{z}^{(k)} = (\frac{1}{n}\sum_{j=1}^{n} z_j^{(k)})\mathbf{1}_n \in \mathbb{R}^n$
- With the recursion of $\bar{x}^{(k)}$ in the last page, we have

$$\begin{aligned}
\|x^{(k+1)} - \bar{x}^{(k+1)}\|^2 &= \|W(x^{(k)} - \bar{x}^{(k)}) + \Delta^{(k+1)} - \bar{\Delta}^{(k+1)}\|^2 \\
&= \|\left(W - (1/n)\mathbf{1}_n\mathbf{1}_n^T\right)(x^{(k)} - \bar{x}^{(k)}) + \Delta^{(k+1)} - \bar{\Delta}^{(k+1)}\|^2 \\
&\leq \rho\|x^{(k)} - \bar{x}^{(k)}\|^2 + \frac{1}{1-\rho}\|\Delta^{(k+1)}\|^2
\end{aligned}$$

where $\Delta^{(k)} = z^{(k)} - z^{(k-1)}$ and $\bar{\Delta}^{(k)} = \bar{z}^{(k)} - \bar{z}^{(k-1)}$

Dynamic average consensus

$$x^{(k+1)} = Wx^{(k)} + z^{(k+1)} - z^{(k)} \quad \text{where} \quad x^{(0)} = z^{(0)}$$

- If the dynamic input oscillates in a small range, i.e., $\|\Delta^{(k)}\|^2 = e^2$, it holds that

$$\lim_{k\to\infty} \|x^{(k)} - \bar{x}^{(k)}\| = \frac{e}{1-\rho} \quad \text{where} \quad \bar{x}^{(k)} = \bar{z}^{(k)} = (\frac{1}{n}\sum_{j=1}^{n} z_j^{(k)})\mathbf{1}_n \in \mathbb{R}^n$$

  Dynamic average consensus converges to a small neighborhood around $\bar{z}^{(k)}$

- If the dynamic input converges to stationary points, i.e.,$\|\Delta^{(k)}\| \to 0$, it holds that

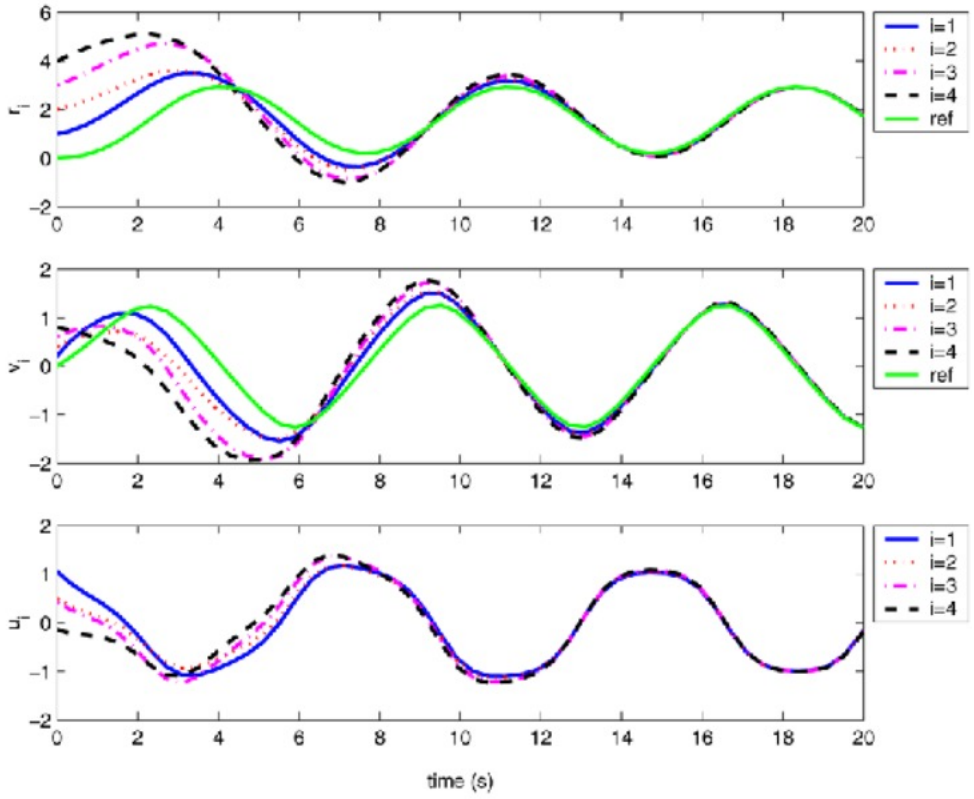
$$\lim_{k\to\infty} \|x^{(k)} - \bar{x}^{(k)}\| \to 0$$

  The convergence rate is determined by both ρ and the rate at which $\Delta^{(k)}$ approaches 0

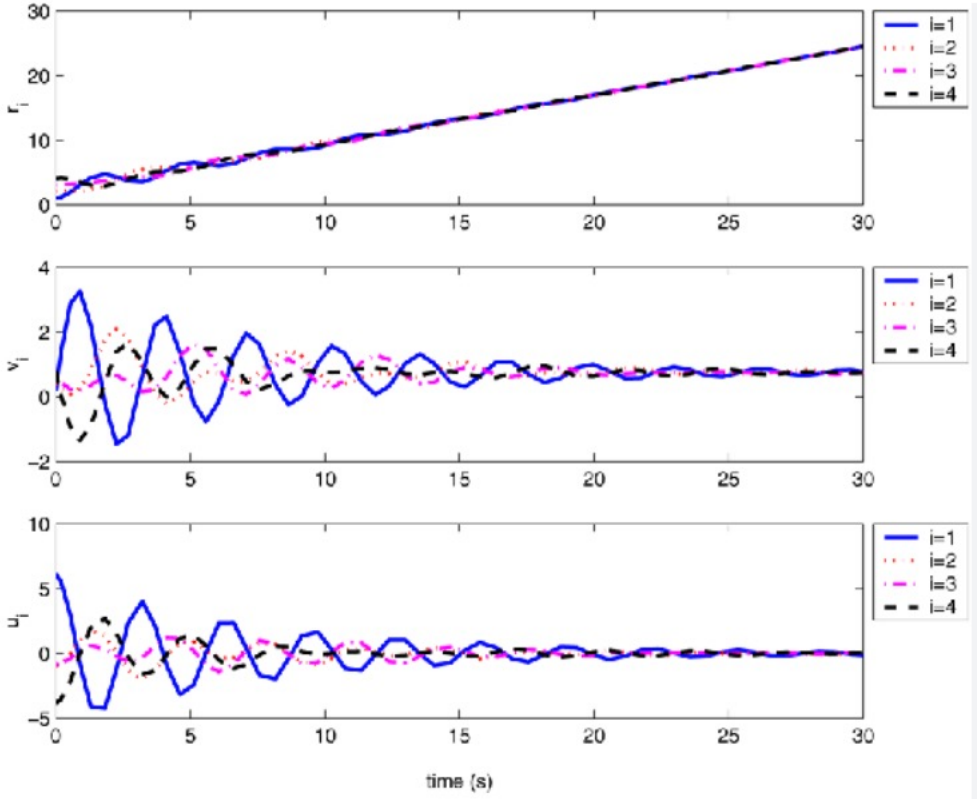
Simulations

北京大学 PEKING UNIVERSITY

Small oscillation in dynamic input



Dynamic input converges to a stationary point



Simulation results are from [Ren IEEE TAC 2007]

Summary

北京大学 PEKING UNIVERSITY

- Decentralized communication can save significant communication overhead
- Decentralized learning significantly speeds up DNN training
- Decentralized learning algorithms build upon partial averaging (also known as average consensus)
- Partial averaging converges to global averaging exponentially fast
- Dynamic average consensus can track the average of the dynamic input