北京大學
PEKING UNIVERSITY

Optimization for Deep Learning

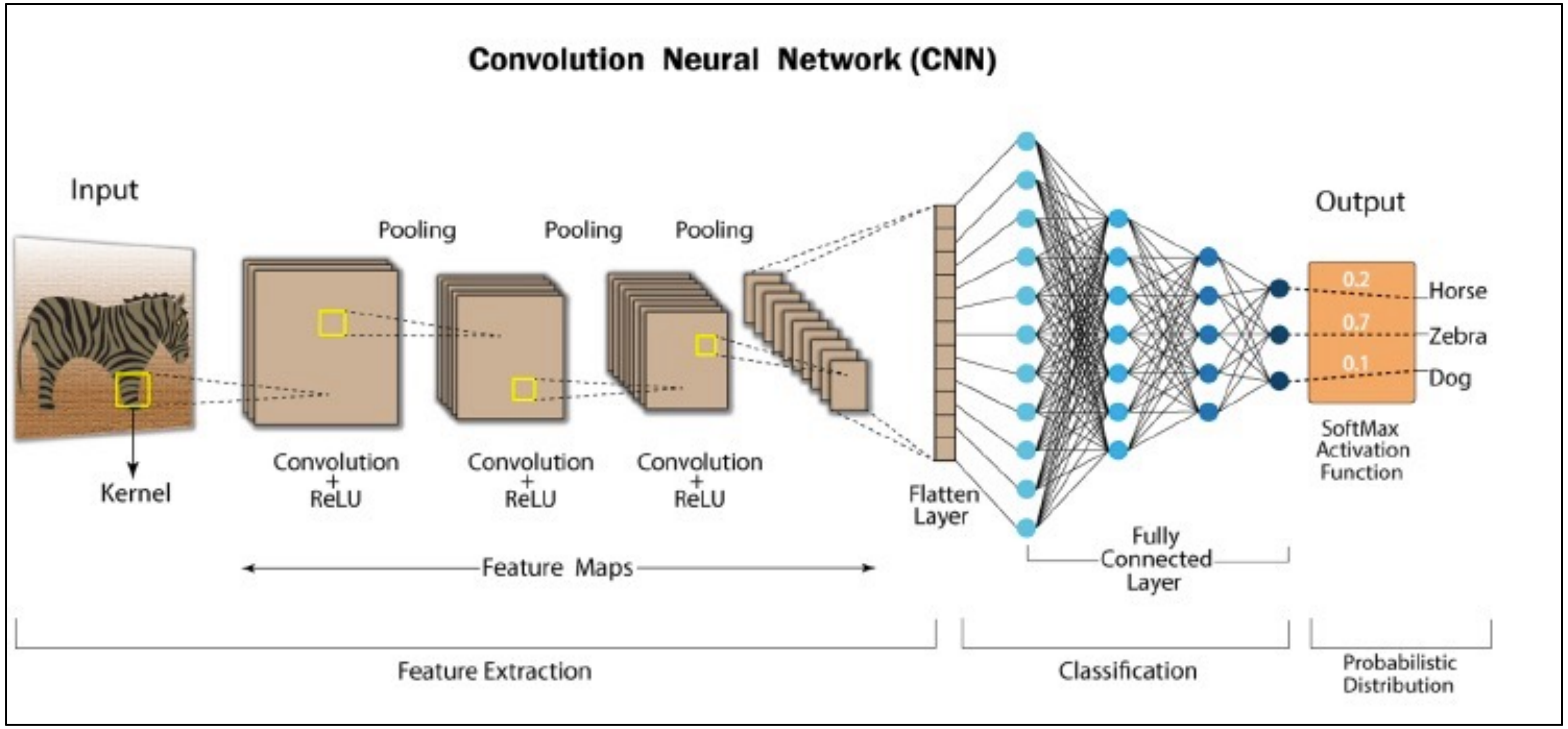
Lecture 13-1: Introduction to Distributed Learning

Kun Yuan

Part 1

Distributed Stochastic Optimization

Training deep neural network is notoriously difficult



DNN training = non-convexity + **massive dataset** + huge models

Distributed learning

- 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:

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

  - 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

Part 2

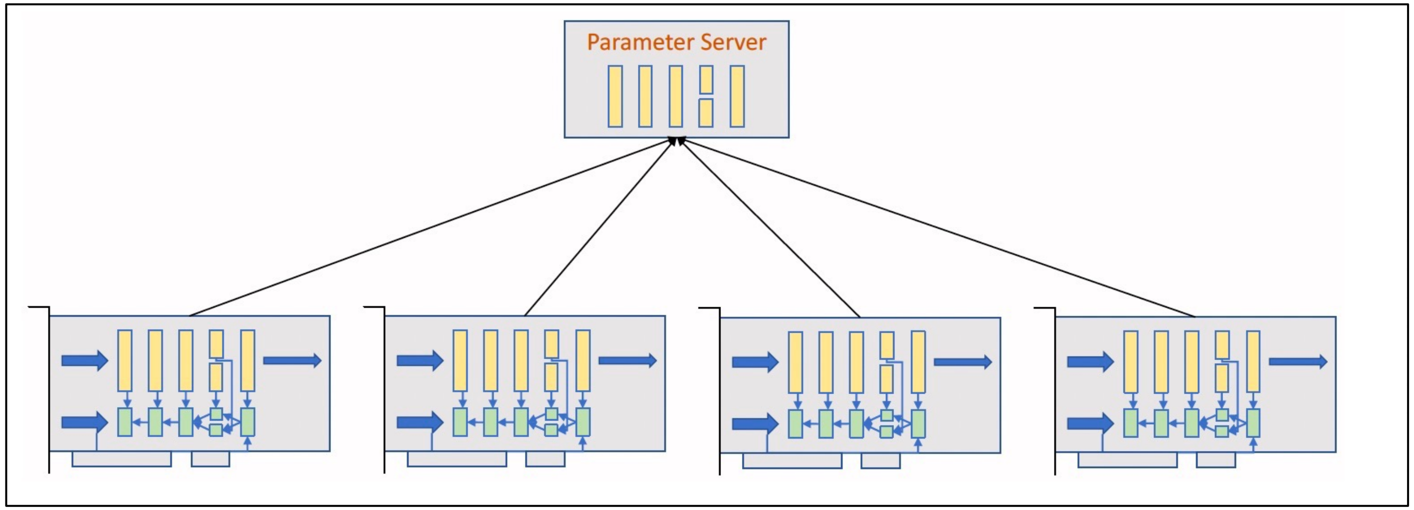
Parallel Stochastic Gradient Descent

Vanilla parallel stochastic gradient descent (PSGD)

$$
\begin{aligned}
g_i^{(k)} &= \nabla F(x^{(k)}; \xi_i^{(k)}) && \text{(Local compt.)} \\
x^{(k+1)} &= x^{(k)} - \frac{\gamma}{n}\sum_{i=1}^{n} g_i^{(k)} && \text{(Global comm.)}
\end{aligned}
$$

- 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)



- The figure shows a parameter-server framework
- More advanced distributed learning framework: Tree-Allreduce, Ring-Allreduce, etc.

Convergence

$$
\begin{aligned}
g_i^{(k)} &= \nabla F(x^{(k)}; \xi_i^{(k)}) && \text{(Local compt.)} \\
x^{(k+1)} &= x^{(k)} - \frac{\gamma}{n}\sum_{i=1}^{n} g_i^{(k)} && \text{(Global comm.)}
\end{aligned}
$$

- PSGD is essentially a SGD algorithm with batch-size n

- The convergence analysis of PSGD follows that of mini-batch SGD

Convergence

Theorem [PSGD Convergence Property]

Assume each $f_i(x)$ is L-smooth and each stochastic gradient $\nabla F(x;\xi_i)$ is unbiased and has bounded variance σ^2, parallel SGD converges as follows

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla f(x^{(k)})\|^2 = \mathcal{O}\left(\sqrt{\frac{L\sigma^2}{nK}} + \frac{L}{K}\right)$$

- The iteration complexity of PSGD is $\mathcal{O}(1/(n\epsilon^2))$
- Achieves **linear speedup**! The number of iterations to reach ϵ decreases linearly as n increases

Part 3

Empirical Studies

Experiments in deep training (image classification)

ImageNet-1K dataset

1.3M training images

50K test images

1K classes

DNN model: ResNet-50 (25.5M parameters)

GPU: Up to 256 Tesla V100 GPUs

- **Wall-clock time** to finish 90 epochs of training; measures per-iter communication
- **Validation accuracy** after 90 epochs of training; measures convergence rate

DSGD over one-peer Exp. achieves better linear speedup

nodes	4(4x8 GPUs)		8(8x8 GPUs)		16(16x8 GPUs)		32(32x8 GPUs)	
topology	acc.	time	acc.	time	acc.	time	acc.	time
P-SGD	76.32	11.6	76.47	6.3	76.46	3.7	76.25	2.2

PSGD almost achieves a linear speedup

However, the linear speedup is not perfect

[YYC+21]B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin, ``Exponential Graph is Provably Efficient for Deep Training", NeurIPS 2021

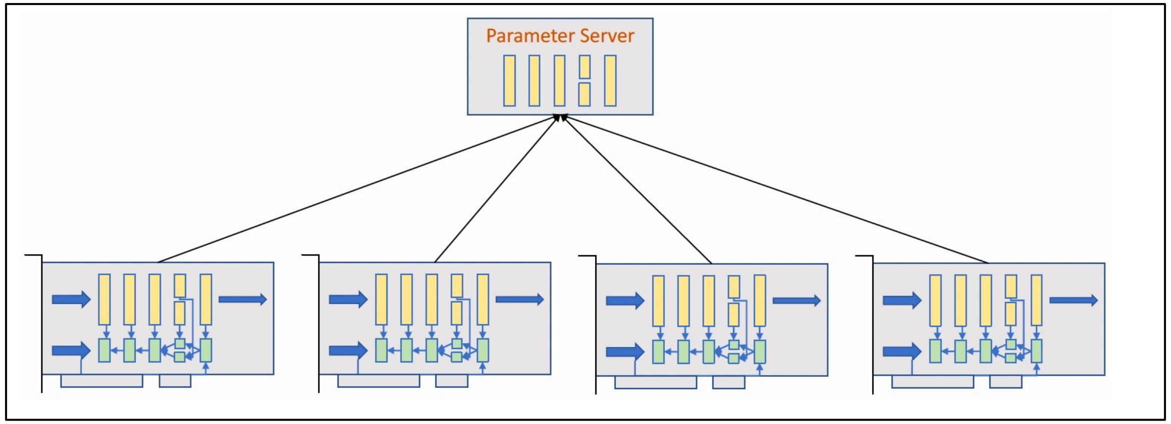
Part 4

Communication overhead hinders perfect linear speedup

PSGD incurs significant communication overhead

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

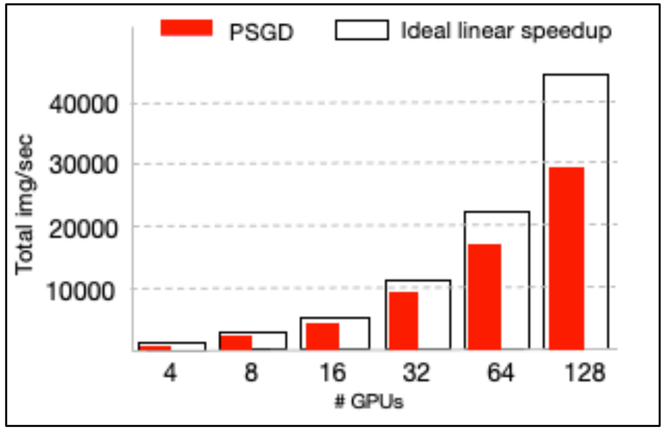
$$x^{(k+1)} = x^{(k)} - \frac{\gamma}{n}\sum_{i=1}^{n} g_i^{(k)} \qquad \text{(Global comm.)}$$



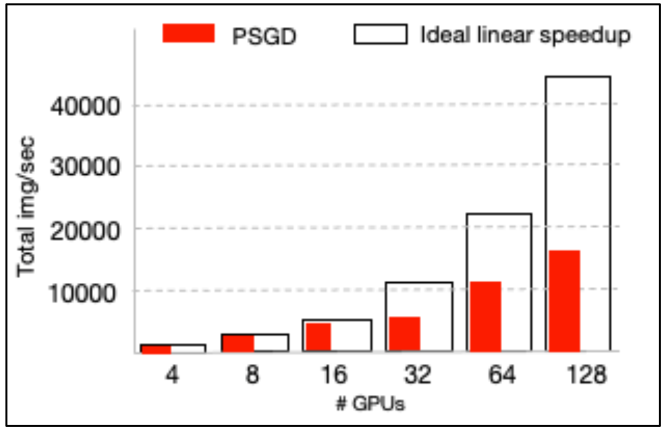
- PSGD requires a **global average** per communication; incurs O(n) worst-case overhead
- PSGD communicates a **full-dimensional** vector per communication; incurs O(d) overhead
- PSGD communicates **at every iteration**; incurs O(K) overhead

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? **We will discuss it in the next few lectures**

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

Summary

- Data-Parallel distributed learning is essential to handle massive dataset
- Vanilla parallel SGD (PSGD) achieves theoretical linear speedup in convergence
- However, the linear speedup in real implementations is not perfect due to communication overhead
- This motivates us to develop communication-efficient algorithms to reduce communication overhead