Distributed Stochastic Gradient Methods

Dr. Dingzhu Wen

School of Information Science and Technology (SIST)
ShanghaiTech University

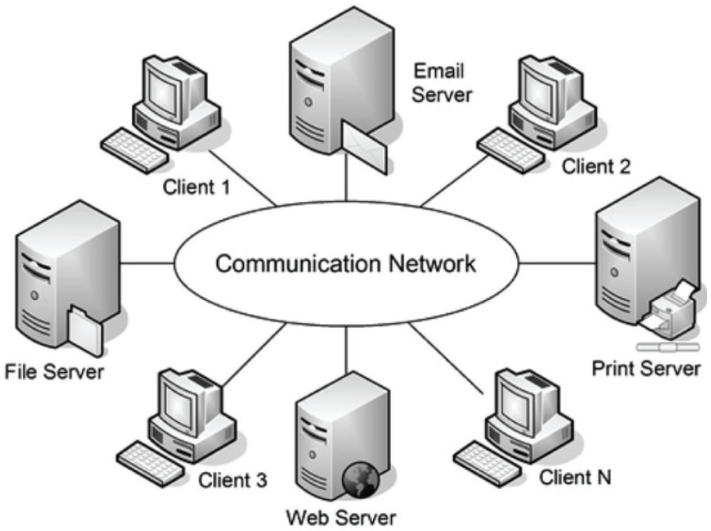
wendzh@shanghaitech.edu.cn

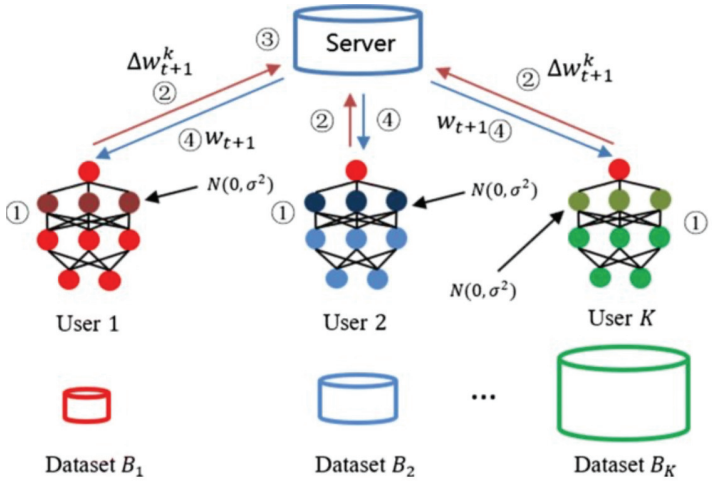
March 20, 2024

Overview

Distributed Optimization Systems

Distributed optimization is an optimization process that is used in networked systems with a large number of nodes.





Motivations

Parallel/Distributed training is necessary, especially for deep neural netowrks(DNNs).

- Scale to Larger Models and Bigger Data
  - Large model: ResNet50 (> 4 millions of parameters), AlexNet (≈ 8 millons of parameters),
  - Big dataset: ImageNet ($14,197,122$ $256 \times 256 \times 3$ images),
- Multi-Core Computing: Bring down training time from days to hours.
- Distributed Datasets with Privacy Preservation (Federated Learning).

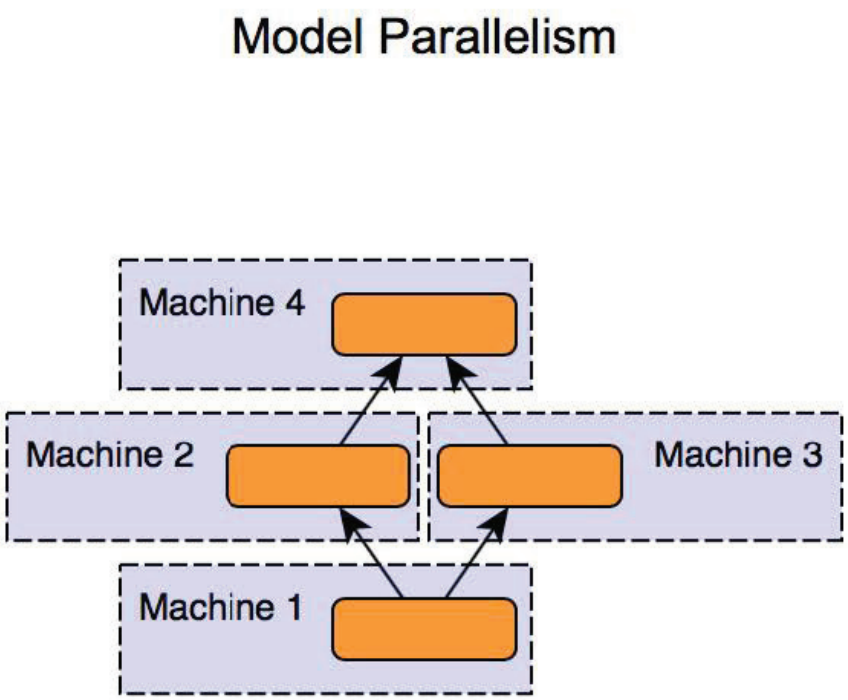
Types

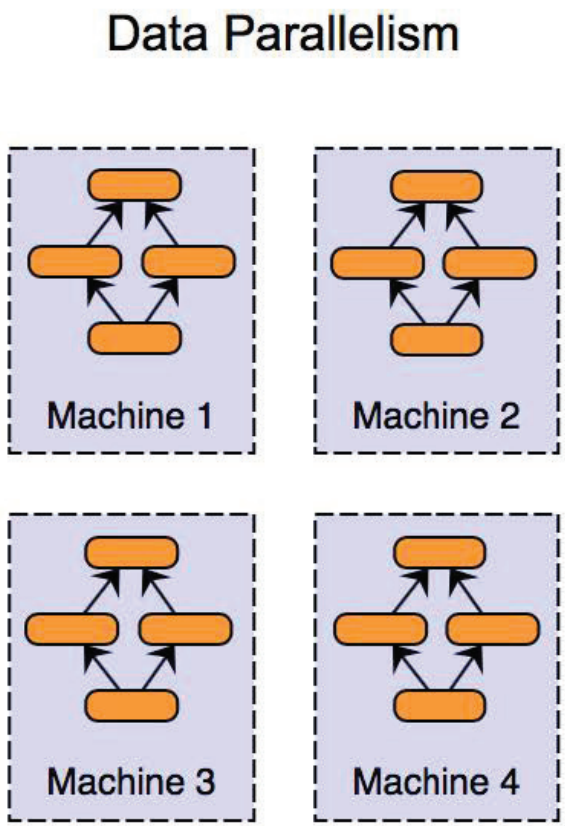
Types of Distributed Training:

- Data-parallel training: share the model; partition the data;
- Model-parallel training: share the data; partition the model;
- Data-parallel and model-parallel mixed training.

In this lecture, the most common type, say SGD based data-parallel training, is the focus.

Types


Model Parallelism
Machine 4
Machine 2
Machine 3
Machine 1


Data Parallelism
Machine 1
Machine 2
Machine 3
Machine 4

DNN Training as Distributed Optimization

Setting

- A network of N modes (GPUs) collaborate to solve the problem:

$$\max_{\mathbf{w}\in\mathbb{R}^d} \quad \mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(\mathbf{w}),$$

- $\ell_i(\mathbf{w}) = \mathbb{E}_{\mathbf{z}_i\sim D_i} f(\mathbf{w}; \mathbf{z}_i)$,
- Each component $\ell_i(\cdot)$ is local and private to node i,
- Random variable $\mathbf{z}_i$ denotes the local data that follows distribution D_i,
- Each local distribution D_i may be different, i.e., data heterogeneity.

DNN Training as Distributed Optimization

Setting

- Training in a server with multiple cores (GPUs),
  - All GPUs are connected with high-bandwidth channels,
  - Network topology can be fully controlled,
  - Communication is highly reliable; no occasional link failure,
  - In summary: Communication problem is ignored.
- Different from the mobile AI applications, or Federated Learning where
  - Nodes are connected with low-bandwidth channels,
  - Network topology can not be controlled,
  - Communication is highly fragile; occasional link failures.

Distributed (Parallel) Gradient Descent

Main Idea [R1]: In each iteration,

- each node solves the problem using full-batch local data,
- all local gradients are averaged to derive the true gradient,
- the true gradinet are used to update the model parameters.

Advantages

- The computation load is divided,
- True gradient is derived without variance.

Shortage

- The number of processers is not comparable to the size of the global dataset. Local full-batch GD is computaionally expensive.

[R1] G. Mann, R. McDonald, M. Mohri, N. Silberman, and D. Walker, "Efficient large-scale distributed training of conditional maximum entropy models," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1231-1239. 2009.

Distributed (Parallel) Stochastic Gradient Descent

Main Idea

- Similar to distributed GD except each node using a single or a mini-batch of data sample for updating in this case,
- The single or mini-batch sample is (are) ramdonly chosed.

In the t-th iteration, the stochastic gradient generated by the i-th device is

$$\mathbf{g}_{i,t} = \nabla f(\mathbf{w}_t, \mathbf{z}_{i_t}),$$

Distributed SGD iteration:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{N} \sum_{i=1}^{N} \mathbf{g}_i.$$

Distributed (Parallel) Stochastic Gradient Descent

Description

- Each node i randomly samples data i_t and computes the stochastic gradient $\mathbf{g}_i$,

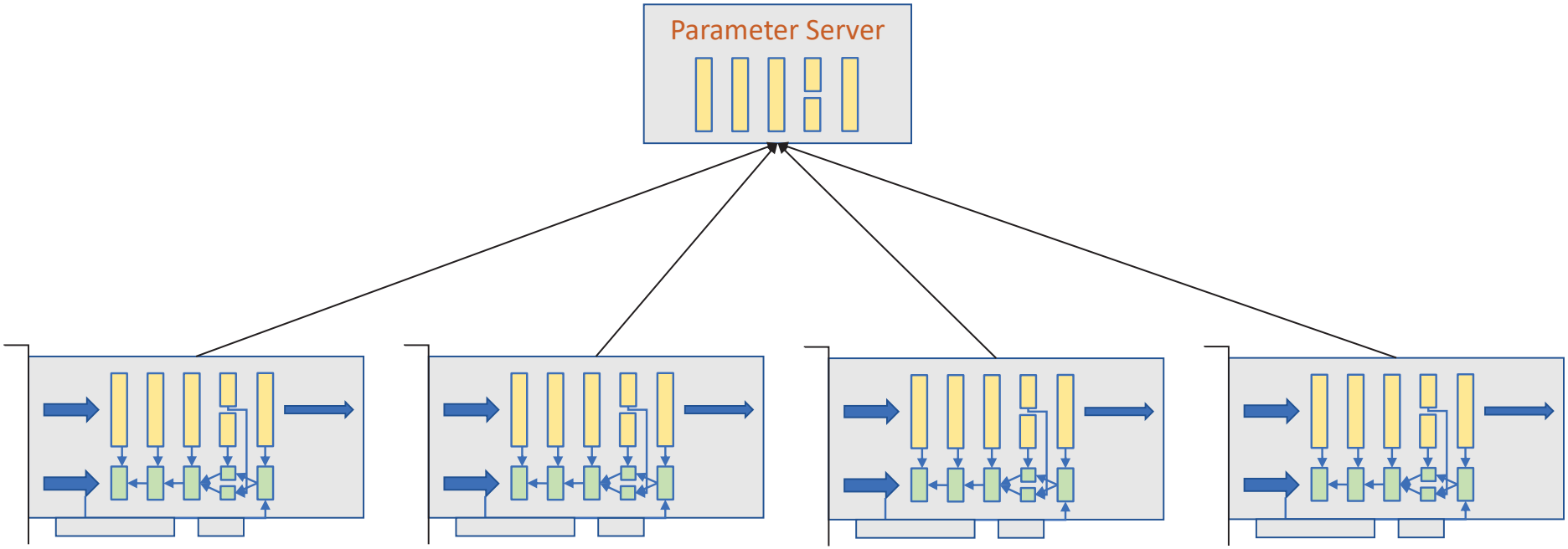
$$\mathbf{g}_{i,t} = \nabla f(\mathbf{w}_t, \mathbf{z}_{i_t}),$$

- All nodes synchronize (i.e. global averaged) to update model $\mathbf{w}$:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{N} \sum_{i=1}^{N} \mathbf{g}_i,$$

- Global average incurs significant communication cost, which hinders training scalability.

Parameter-Server Framework for Stochastic Gradient Average



Covergence Analysis of Distributed SGD

Assumption

*(**A1:Unbiased Estimation**)*

$$\mathbb{E}\left[\mathbf{g}_{i,t}\middle|\mathbf{w}_t\right] = \frac{1}{N}\sum_{i=1}^{N}\nabla_{\mathbf{w}_t}\ell(\mathbf{w}_t;\mathbf{z}_i) = \nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t).$$

*(**A2: Bounded Stochastic Gradient Variance**)*

$$\mathbb{E}\left[\left\|\mathbf{g}_{i,t}\right\|^2\middle|\mathbf{w}_t\right] - \left\|\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)\right\|^2 \leq \sigma^2.$$

*(**A3: L-Smoothness**) $\{\ell(\mathbf{w};\mathbf{z}_i),\ 1 \leq i \leq M\}$ are L-smooth.*

*(**A4: Independence**) Each local stochastic gradient $\mathbf{g}_{i,t}$ is independent of each other.*

Question: How to use these assumptions to show the convergence?

Covergence Analysis of Distributed SGD

Descent Lemma of L-smooth:

$$\mathcal{L}(\mathbf{w}_{t+1}) \leq \mathcal{L}(\mathbf{w}_t) + \nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)^T(\mathbf{w}_{t+1} - \mathbf{w}_t) + \frac{L}{2}\left\|\mathbf{w}_{t+1} - \mathbf{w}_t\right\|^2.$$

For the t-th distributed SGD iteration,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{N}\sum_{i=1}^{N} g_{i,t}.$$

Then,

$$\mathcal{L}(\mathbf{w}_{t+1}) \leq \mathcal{L}(\mathbf{w}_t) - \frac{\eta}{N}\sum_{i=1}^{N}\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)^T\mathbf{g}_{i,t} + \frac{L\eta^2}{2N^2}\left\|\sum_{i=1}^{N}\mathbf{g}_{i,t}\right\|^2.$$

Covergence Analysis of Distributed SGD

Conditioned on all past iterations,

$$\mathbb{E}\left[\mathcal{L}(\mathbf{w}_{t+1})\middle|\mathbf{w}_t\right] \leq \mathcal{L}(\mathbf{w}_t) - \eta \left\|\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)\right\|^2 + \frac{L\eta^2}{2N^2}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\mathbf{g}_{i,t}\right\|^2\middle|\mathbf{w}_t\right].$$

By using the assumption of bounded stochastic gradient variance:

$$\mathbb{E}\left[\mathcal{L}(\mathbf{w}_{t+1})\middle|\mathbf{w}_t\right] \leq \mathcal{L}(\mathbf{w}_t) - \eta\left(1-\frac{L\eta}{2}\right)\left\|\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)\right\|^2 + \frac{L\eta^2\sigma^2}{2N}.$$

The variance of the globally averaged gradient is remarkably reduced.

Covergence Analysis of Distributed SGD

Theorem (Convergence of Distributed SGD with Fixed Step Size)

Using Assumptions (A1)-(A4), then the sequence $\{\mathbf{w}_t\}_{t\in\mathbb{N}}$ *generated by SGD with step size* $\eta = \frac{1}{L}$ *satisifies*

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)\|^2 \leq \frac{2L}{T+1}\left\{\mathbb{E}\left[\mathcal{L}(\mathbf{w}_{t+1})\middle|\mathbf{w}_t\right] - \mathcal{L}_*\right\} + \frac{\sigma^2}{N}.$$

Covergence Analysis of Distributed SGD

Theorem (Convergence of Distributed SGD with Adaptive Step Size)

Using Assumptions (A1)-(A4), then the sequence $\{\mathbf{w}_t\}_{t\in\mathbb{N}}$ *generated by SGD with step size* $\eta = \frac{1}{\sqrt{(T+1)L}}$ *satisifies*

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla_{\mathbf{w}_t}\mathcal{L}(\mathbf{w}_t)\|^2 = \mathcal{O}\left(\frac{\sigma}{\sqrt{N(T+1)}}\right).$$

Distributed SGD Can Achieve Linear Speedup

Convergence Rate

- Single node: $\mathcal{O}\left(\frac{\sigma}{\sqrt{(T+1)}}\right)$,
- Multiple nodes: $\mathcal{O}\left(\frac{\sigma}{\sqrt{N(T+1)}}\right)$.

To achieve an ϵ-accurate solution, i.e., $\frac{1}{T+1}\sum_{t=0}^{T}\nabla\mathcal{L}(\mathbf{w}) \leq \epsilon$,

- Single-node training requires $\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$,
- Multi-node training requires $\mathcal{O}\left(\frac{\sigma^2}{N\epsilon^2}\right)$,
- Iteration complexity is inversely proportional to n, i.e., distributed SGD has a linear sppedup.

ImageNet Classification

Settings

- ImageNet-1K dataset
- 1.3M training images
- 50K test images
- 1K classes
- DNN Model: ResNet-50
- GPU: Tesla V100 clusters
- Framework: Pytorch DDP

ImageNet Classification

Table: Comparison of Training Time

Number of GPUs	32	64	128	256
Test Accuracy	76.32	76.47	76.46	76.25
Time (Hours)	11.6	6.3	3.7	2.2

- Cannot achieve ideal linear speedup due to comm. cost.
- Global average incurs significant comm. cost; hinders training scalability.

Thank you!

wendzh@shanghaitech.edu.cn