DNN Training as Distributed Optimization

Setting

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

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

- $\ell_i(\mathbf{w}) = \mathbb{E}_{\mathbf{z}_i\sim D_i} f(\mathbf{w};\mathbf{z}_i) + r(\mathbf{w})$,
- $r(\mathbf{w})$ is a non-differentiable regularized function,
- 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) Subgradient Method

Main Idea: In each iteration,

- Each node randomly picks up a sample and perform stochastic sub-gradient method,
- The local subgradients are averaged at the server (coordinator),
- The subgradinet average are used to update the model parameters.

Advantages

- The computation load are divided.

Distributed (Parallel) Stochastic Subgradient Method

Learning Objective:

$$\max_{\mathbf{w}\in\mathbb{R}^d} \quad \mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\ell_i(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\left[f_i(\mathbf{w}) + r(\mathbf{w})\right],$$

In the t-th iteration, the stochastic subgradient generated by the i-th device is calculated by, e.g, proximal gradient method:

$$\mathbf{g}_{i,t} = \mathbf{w}_t - \mathrm{Prox}_{r,\eta}(\mathbf{w}_t - \eta\nabla f_i(\mathbf{w}_t)),$$

Distributed subgradient method iteration:

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

Practical updating:

$$\mathbf{w}_{t+1} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{Prox}_{r,\eta}(\mathbf{w}_t - \eta\nabla f_i(\mathbf{w}_t)).$$

Convergence Analysis

Assumptions

Assumption

(A1: μ-Strongly Convex) $f(\mathbf{w})$ *is* μ*-strongly congex.*

(A2: Unbiased Estimation) $\mathbb{E}\left[\mathbf{g}_{i,t}|\mathbf{w}_t\right] = \mathbf{g}_t$.

(A3: Bounded Subgradient Norm) $\mathbb{E}\left[\|\mathbf{g}_{i,t}\|^2 |\mathbf{w}_t\right] \leq B^2$.

Question: How to show the convergece using the above assumptions?
Answer: Similar to that of stochastic subgradient method.

Thank you!

wendzh@shanghaitech.edu.cn