DeComFL: Federated Learning with Dimension-Free Communication

Zhe Li [1,*] Bicheng Ying [2,*] Zidong Liu [3] Chaosheng Dong [4]
Haibo Yang [1]

[1]Rochester Institute of Technology [2]Google Inc. [3]ComboCurve Inc. [4]Amazon.com Inc.

[*]Equal contribution.

December 15, 2024

Motivation

The communication cost in federated learning becomes a bottleneck in the large language model era. In each round, the communication costs **scale linearly** with the model dimension.

Using FedAvg as an example

$$\mathbf{x}_{i,r}^{1} = \mathbf{x}_r, \quad \forall i \in C_r \qquad \text{(Pull Model)}$$

$$\mathbf{x}_{i,r}^{k+1} = \mathbf{x}_{i,r}^{k} - \eta \nabla f_i(\mathbf{x}_{i,r}^{k}), \quad k = 1, 2, \cdots, K, \qquad \text{(Local Update)}$$

$$\mathbf{x}_{r+1} = \frac{1}{|C_r|} \sum_{i \in C_r} \mathbf{x}_{i,r}^{K}, \qquad \text{(Aggregate Model)}$$

We want to eliminate this linear scale relationship between the model dimension and the communication cost.

Preliminary of Zeroth-order Optimization

Zeroth-order (ZO) optimization is a **gradient-free** technique for solving optimization problems that uses function values instead of gradients.

$$x_{r+1} = x_r - \eta \cdot g \cdot z \qquad \text{ZO-SGD}$$

$$\text{where} \quad g = \frac{1}{\mu}\big(f(x+\mu z) - f(x)\big) \qquad \text{Gradient Scalar}$$

$$z \sim \mathcal{N}(0, I_d) \qquad \text{Gaussian Distribution}$$

The model update term $-\eta \cdot g \cdot z$ can be constructed by two scalars:

- g is just **a scalar**
- z can be reproduced given a **seed**

Algorithm Design

The key design idea is transforming the algorithm into **an update-based form without losing the global model synchronization.**

Eliminating Dimension-Dependent Communication in the Uplink:

$$\mathbf{x}_{i,r}^{k+1} = \mathbf{x}_{i,r}^{k} - \eta \cdot g_{i,r}^{k} \cdot \mathbf{z}_r^k, \quad k = 1, 2, \cdots, K, \qquad \text{(Local Update)}$$

$$\mathbf{x}_{r+1} = \mathbf{x}_r - \eta \sum_{k=0}^{K-1} \underbrace{\left(\frac{1}{|C_r|} \sum_{i \in C_r} g_{i,r}^{k} \right)}_{:= g_r^k} \cdot \mathbf{z}_r^k \qquad \text{(Aggregate Model Update)}$$

Key points:

- $\mathbf{z}_r^k$ instead of $\mathbf{z}_{i,r}^k$ because of a shared seed determined by the server.
- Server model update only requires several scalars $\{g_{i,r}^k\}$.

Algorithm Design (Cont'd)

Eliminating Dimension-Dependent Communication in the Downlink:

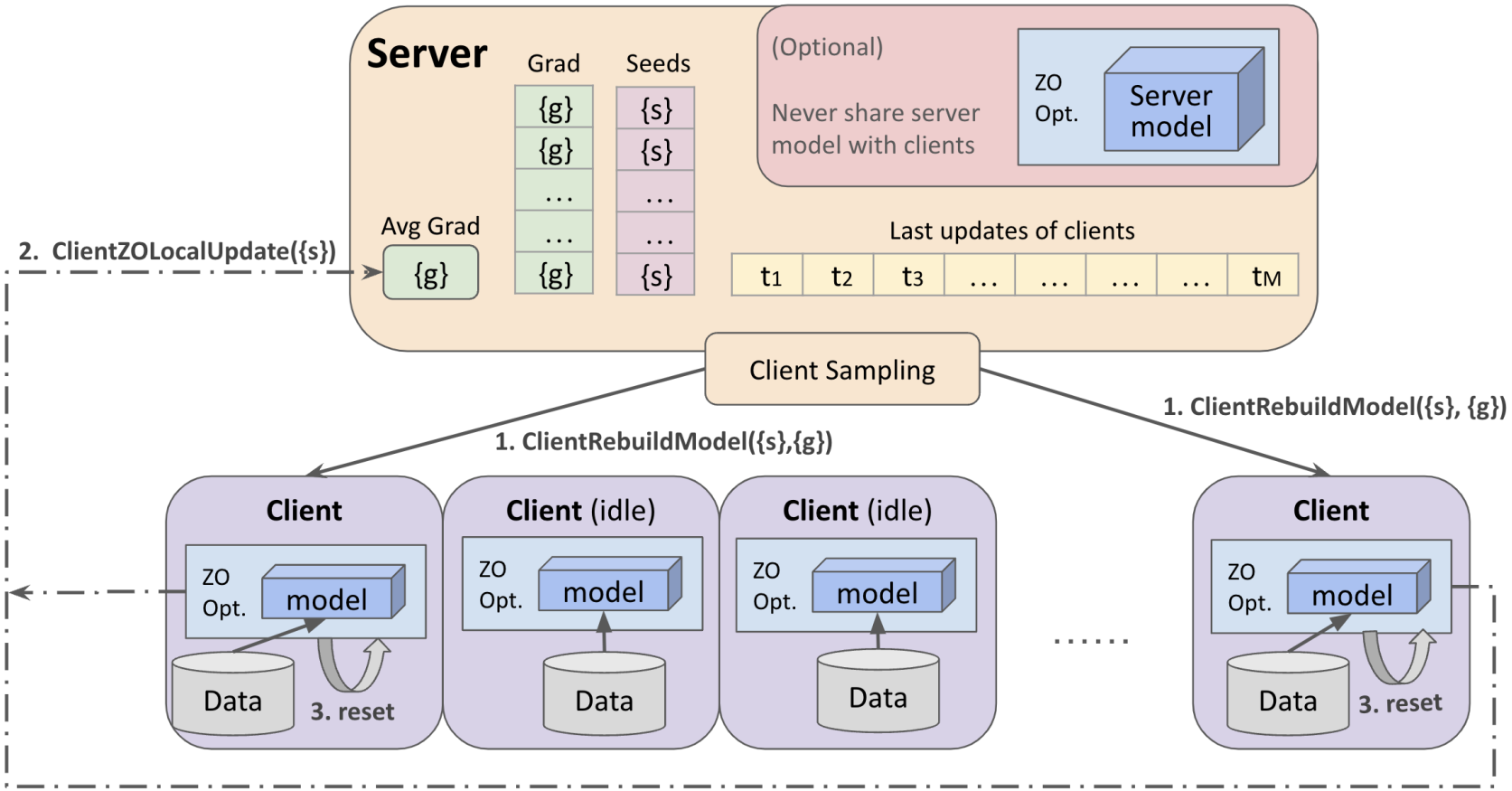
$$\mathbf{x}_{i,r}^{1} = \mathbf{x}_{i,r'}^{K} - \eta \sum_{j=r'}^{r-1} \sum_{k=1}^{K} g_j^k \cdot \mathbf{z}_j^k, \qquad \text{(Reconstruct Model)} \qquad (1)$$

where r' is the last participation round. Above is a valid model only if $\mathbf{x}_{i,r'}^{K}$ is the same as server model $\mathbf{x}_{r'}$, but at the end of the local update, the client model $\mathbf{x}_{i,r'}^{K}$ **deviates** from the server model.

One straightforward solution

- **take a snapshot** of the model at the beginning of the local update
- **revert** to the snapshot after completing the local update.

Workflow and Communication Pattern of DeComFL



Experimental Results

Table: Test Accuracy and Communication Costs on Fine-Tuning Tasks

Model	Dataset \ Task	MeZO	FedZO with $P = 5$	DeComFL with $P = 5$	DeComFL with $P = 10$
OPT-125M	SST-2	83.99%	84.11% (0.68 TB)	84.02% (0.18 MB)	**85.08%** (0.36 MB)
	CB	72.49%	73.97% (0.23 TB)	74.28% (0.06 MB)	**75.00%** (0.12 MB)
	WSC	55.18%	59.43% (0.68 TB)	59.13% (0.18 MB)	**59.59%** (0.36 MB)
	WIC	53.25%	53.31% (0.68 TB)	53.28% (0.18 MB)	**53.38%** (0.36 MB)
	RTE	52.91%	53.42% (0.45 TB)	54.33% (0.12 MB)	**57.05%** (0.24 MB)
	BoolQ	61.46%	61.20% (0.45 TB)	61.36% (0.12 MB)	**61.60%** (0.24 MB)
OPT-1.3B	SST-2	90.23%	90.17% (4.73 TB)	90.02% (0.12 MB)	**90.78%** (0.24 MB)
	CB	74.01%	74.41% (7.09 TB)	74.40% (0.18 MB)	**75.71%** (0.36 MB)
	WSC	58.21%	59.95% (7.09 TB)	60.41% (0.18 MB)	**64.16%** (0.36 MB)
	WIC	55.95%	56.06% (4.73 TB)	55.97% (0.12 MB)	**56.14%** (0.24 MB)
	RTE	57.57%	58.88% (3.55 TB)	59.42% (0.90 MB)	**60.89%** (1.80 MB)
	BoolQ	61.98%	62.01% (3.55 TB)	62.17% (0.90 MB)	**62.50%** (1.80 MB)

The value enclosed in parentheses represents the total bytes of the vector transferred between the server and a client throughout the entire fine-tuning phase.

Thank you!

Q&A