Approximate Gradient Coding for Distributed Learning with Heterogeneous Stragglers

Heekang Song[1], Wan Choi[2]

[1] Korea Advanced Institute of Science and Technology

[2] Seoul National University

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction and Motivation of Approximate Gradient Coding for Distributed Learning

- Recent large-scale AI models like ChatGPT and Gemini necessitate distributed learning, which operates through the following three phases.

① Data distribution: n data partitions $D_1, D_2, \dots, D_n$ to k workers $W_1, W_2, \dots, W_k$.

② Local gradient computation: compute $g_i^{(t)} = \nabla L(D_i, \beta^{(t)})$ and transmit partial gradients.

③ Gradient sum retrieval: compute $\beta^{(t+1)} = \beta^{(t)} - \gamma_t \cdot \sum_{i=1}^{n} g_i^{(t)}$ and distribute updated model.

At t-th epoch, operate ② and ③

- The overall performance of a distributed system is bottlenecked by the slowest worker – "**straggler**".
- Without coding and data redundancy, the gradient updates are performed using only a subset of the gradients in the presence of stragglers. (Fig. 1 (left))
- With gradient coding and data redundancy, the gradient updates can be performed using the full gradient. ➡ Computation redundancy provides the coding opportunities. (Fig. 1 (right))

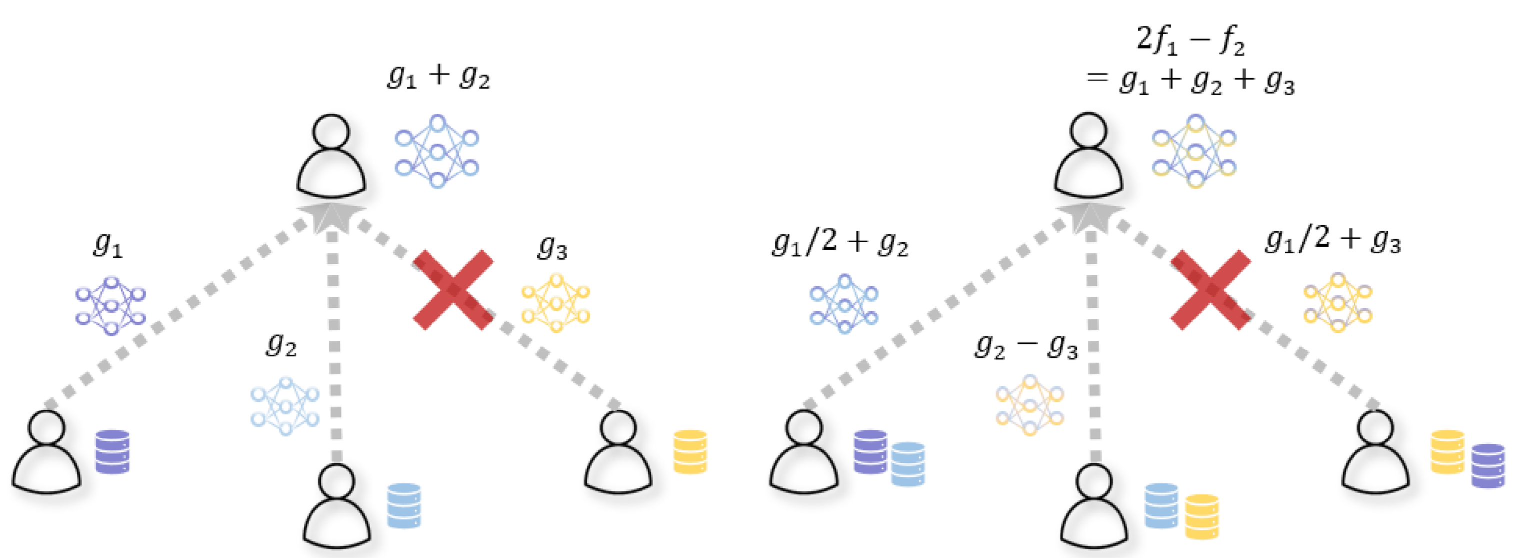

Fig .1: Motivating example of gradient coding.

- ***Gradient coding*** explores coding techniques that ensure the recovery of the aggregated gradient at master node, in the presence of stragglers.
- Limitation of prior work:
  - ✓ <u>Exact gradient coding</u>: Requires knowing the number of stragglers in advance and suffers from high data replication (computation load).
  - ✓ <u>Approximate gradient coding</u>: More practical, but most methods focus on only one of two goals: (1) minimizing residual error or (2) ensuring unbiasedness.

Estimated Gradient Update

- ✓ Let $\mathcal{I}_i$ be the indicator variable, where $\mathcal{I}_i = 1$ if worker i is non-straggler, with straggler probability p_i.
- ✓ Worker i computes encoded message $f_i^{(t)}$ with encoding coefficient a_{ij} ($a_{ij} = 0$ if worker i has no data j): $f_i^{(t)} = \sum_{j=1}^{n} \mathcal{I}_i \cdot a_{ij} \cdot g_j^{(t)}$.
- ✓ The master node recovers the estimated gradient $\hat{g}^{(t)}$ at iteration t (instead of true gradient $g^{(t)}$): $\hat{g}^{(t)} = \sum_{i=1}^{k} w_i \cdot f_i^{(t)}$,

  where w_i is decoding coefficient, and update the model parameters as $\beta^{(t+1)} = \beta^{(t)} - \gamma_t \cdot \hat{g}^{(t)}$.

Optimally Structured Gradient Coding

Optimal Structure of Gradient Coding

- Our main idea lies in the minimization of residual error under unbiased gradient estimator:

$$\min_{A,w} \ \mathbb{E}\left[\left\|g^{(t)} - \hat{g}^{(t)}\right\|_2^2\right] \quad s.t. \ \mathbb{E}\left[\hat{g}^{(t)}\right] = g^{(t)}$$

What is true gradient?

- ✓ Impractical to obtain true gradient and optimize codes at each iteration.
- ✓ Suppose that there exists a constant C such that $\left\|g_j^{(t)}\right\|_2^2 \le C, \forall j \in [1:n]$, and gradient estimator is unbiased. Then, we have

$$\mathbb{E}\left[\left\|g^{(t)} - \hat{g}^{(t)}\right\|_2^2\right] \le C\left[\sum_{i=1}^{k} p_i(1-p_i) \cdot w_i^2 \left(\sum_{j=1}^{n} a_{i,j}\right)^2\right].$$

- Convex transformation:

Non-convex
$$\min_{A,w} \sum_{i=1}^{k} \delta_i \tilde{w}_i^2 \left(\sum_{j=1}^{n} a_{i,j}\right)^2 \quad s.t. \sum_{i=1}^{k} \tilde{w}_i a_{i,j} = 1, \forall j,$$

$\alpha_i^j = \tilde{w}_i a_{ij}$ ➡

$$\min_{\alpha} \sum_{i=1}^{k} \delta_i \left(\sum_{j=1}^{n} \alpha_i^j\right)^2 \quad s.t. \sum_{i=1}^{k} \alpha_i^j = 1, \forall j,$$
Convex !

where $\tilde{w}_i = (1-p_i) \cdot w_i$ and $\delta_i = \frac{p_i}{(1-p_i)}$

- By using Karush-Kuhn-Tucker (KKT) conditions, the optimal structure of optimization problem satisfies the conditions:

1) $\sum_{j=1}^{n} \alpha_i^j = Y_i, \forall i \in [1:k]$ and 2) $\sum_{i=1}^{k} \alpha_i^j = 1, \forall j \in [1:n]$

where $Y_i = \delta_i^{-1} \cdot \frac{n}{\sum_j \delta_j^{-1}}$ and $\delta_i^{-1} = \frac{1-p_i}{p_i}$ for all $i \in [1:k]$.

Optimally Structured Gradient Code Construction

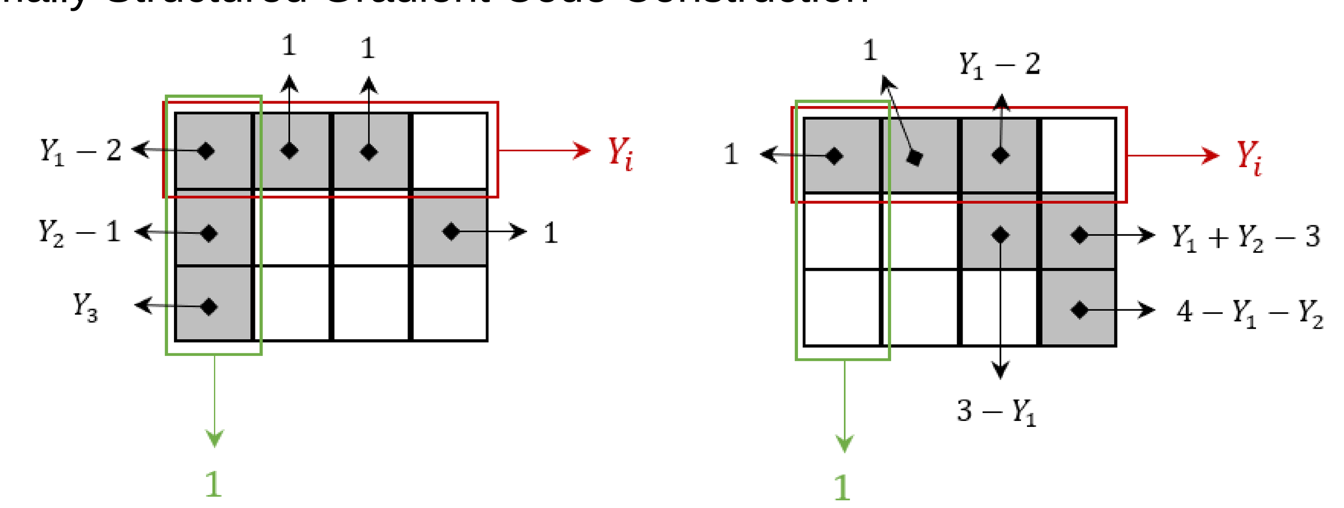

Fig. 2: Illustrative example of proposed schemes: (left) Scheme I and (right) Scheme II.

- **Scheme I**: A single, specific data partition (D_1) is shared by all workers and all other partitions are assigned exclusively to individual workers.
- **Scheme II**: Consecutive workers share a single overlapping data point, and the final worker has only one data partition.
  - ✓ Set $\alpha_i^j = 1$ for exclusive partitions, and set values for shared partitions to satisfy the row-sum constraint $\sum_j \alpha_i^j = Y_i$.
- **Computation load** (data replication for each worker): $\frac{n+k-1}{n} < 2$ ($\because n > k$).
- **Construction of A and w**
  - ✓ Since $\alpha_i^j = \tilde{w}_i a_{ij}$, we can construct $a_{ij} = \alpha_i^j / \tilde{w}_i$ and $w_i = \tilde{w}_i/(1-p_i)$ using random generation of $\tilde{w}$.

Experiments

Convergence Graphs for COCO Dataset (MobileNetv3)

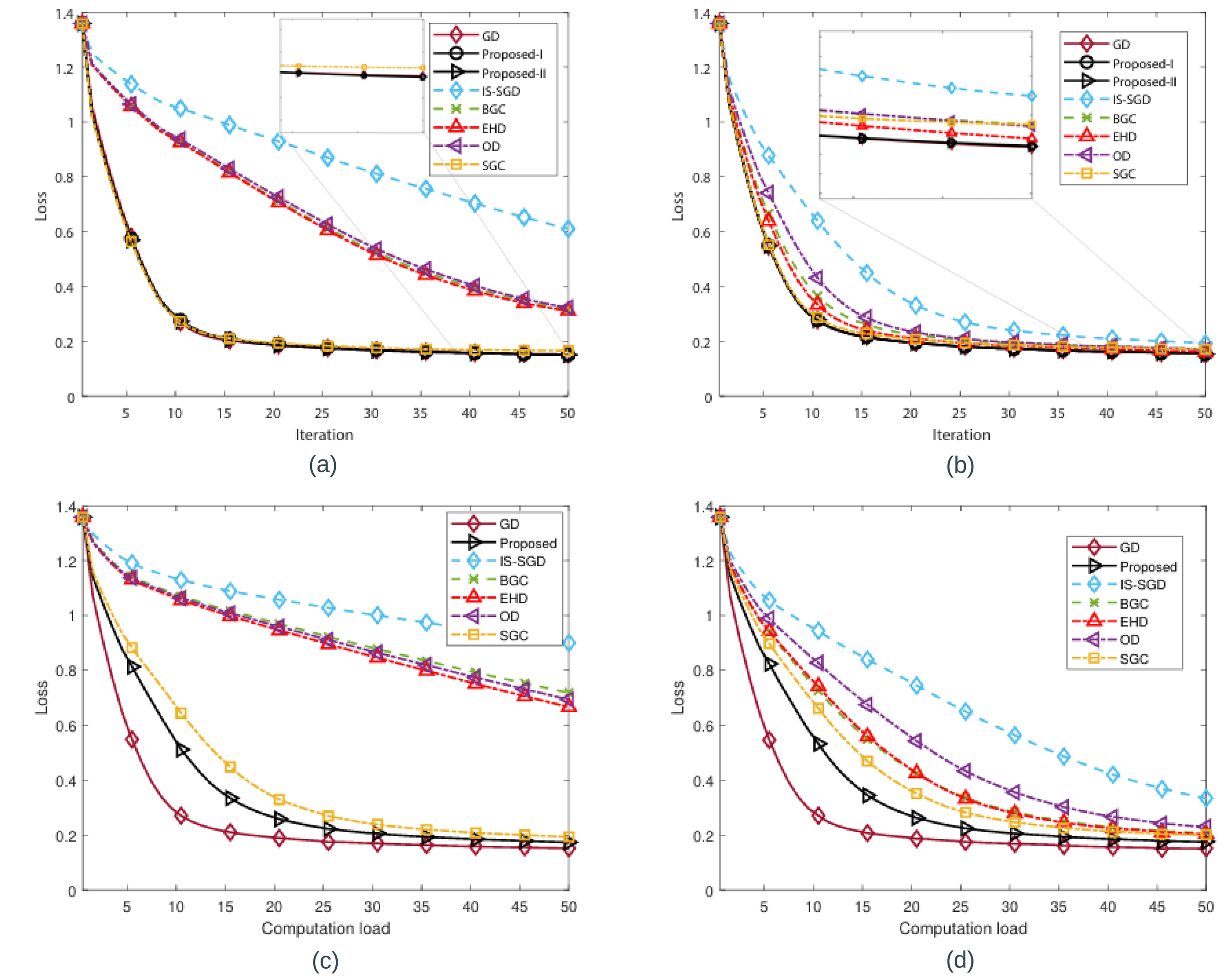

Fig. 3: Convergence graph with respect to the iterations ((a) $\tau_{th} = 1.1$ and (b) $\tau_{th} = 1.5$), and with respect to the computation load ((c) $\tau_{th} = 1.1$ and (d) $\tau_{th} = 1.5$) when $k = 10$.

- **Straggler model**: suppose τ_{th} denote the response time limit for each training iteration. → worker i straggles if local processing time $\tau_i > \tau_{th}$

$$p_i = e^{-\psi_i(\tau_{th} - 1)}$$

where ψ_i represents the straggling parameter obtained by Uniform rand.

- **Baselines**: Centralized learning-based GD, Ignore-Stragglers SGD (IS-SGD), Bernoulli Gradient Coding (BGC) [8], ERASUREHEAD (EHD) [9], Optimal Decoding (OD) [11], Stochastic Gradient Coding (SGC) [12].

Visual Representations: COCO Object Detection

Fig. 4: Detected objects of sampled image.