Latency Minimization for Split Federated Learning in Resource-Constrained MEC Networks

Yujin Hong, Gusun Joung, and Inkyu Lee
Department of Electrical & Electronic Engineering, Korea University, Seoul, Republic of Korea
yujinnx@korea.ac.kr, rntjs300@korea.ac.kr, inkyu@korea.ac.kr

***Abstract*—We propose an optimization algorithm for split federated learning (SFL) that jointly optimizes the model split point, aggregation interval, and uplink transmit power to minimize total training latency for ensuring convergence accuracy. On CIFAR-10 with VGG16, our approach achieves the lowest convergence latency compared to baseline methods, highlighting that the proposed method improves both convergence speed and energy efficiency for the SFL framework, validating its practicality for resource-constrained mobile edge computing (MEC) networks.**

***Index Terms*—Split federated learning, mobile edge computing.**

I. Introduction

As learning models grow larger and more complex, training them on resource-constrained edge devices (EDs) has become increasingly challenging. Federated learning (FL) is a decentralized framework where participating EDs train models locally on their data, and the edge server (ES) aggregates the model parameters transmitted by EDs to construct the global model [1]. Split learning (SL) alleviates the computational burden on EDs by partitioning the model between the EDs and ES, but its sequential training procedure across EDs incurs excessive training latency [2]. To address these limitations, Split federated learning (SFL) combines FL and SL, enabling parallel training across EDs while reducing computation on EDs by partitioning the model [3]. In SFL, the aggregation interval mainly affects the communication overhead and the convergence rate. In addition, the model split point influences not only the communication and computational overheads but also the convergence rate of training. Therefore, we develop an optimization algorithm for SFL, which jointly determines the model split point, the aggregation interval, and the uplink transmit power to minimize the total training latency while ensuring the training accuracy in resource-constrained model edge computing (MEC) networks.

II. System Model

We consider a typical SFL framework [4] in MEC networks, consisting of ED set $\mathcal{N} \triangleq \{1, 2, ..., N\}$ and a ES. Each ED operates in parallel and is responsible for executing the forward propagation (FP) with a computing latency $T_i^{\mathrm{F}} = \frac{b\,\Phi_{c,i}^{\mathrm{F}}(\boldsymbol{\mu})}{f_d}$, where $\Phi_{c,i}^{\mathrm{F}}(\boldsymbol{\mu}) = \sum_{j=1}^{L} \mu_{i,j}\rho_j$. Here, ρ_j denotes the FP computing workload of the propagating jth layer, f_d is the computing capability of ED and b is the mini batch size. Similarly, the backward propagation (BP) is performed with a computing latency of $T_i^B = \frac{b\,\Phi_{c,i}^B(\boldsymbol{\mu})}{f_d}$, where $\Phi_{c,i}^B(\boldsymbol{\mu}) = \sum_{j=1}^{L} \mu_{i,j}\varpi_j$ with ϖ_j denoting the computing workload of ED.

At the ES, FP is performed upon receiving activation data from the participating EDs, with a computing latency given by $T_s^F = \frac{b\,\Phi_s^F(\mu)}{f_s}$, where $\Phi_s^F(\boldsymbol{\mu}) = \sum_{i=1}^{N}\sum_{j=1}^{L} \mu_{i,j}\,(\rho_L - \rho_j)$ with f_s denoting the computing capability of ES. Similarly, BP is performed with a computing latency of $T_s^B = \frac{b\,\Phi_s^B(\mu)}{f_s}$, where $\Phi_s^B(\boldsymbol{\mu}) = \sum_{i=1}^{N}\sum_{j=1}^{L} \mu_{i,j}\,(\varpi_L - \varpi_j)$.

After completing FP at the ED, the cut-layer model parameters are transmitted with an uplink communication latency given by $T_{a,i}^U = \frac{b\Gamma_{a,i}(\boldsymbol{\mu})}{R_i^U(p_{a,i}^U)}$ where $\Gamma_{a,i}(\boldsymbol{\mu}) = \sum_{j=1}^{L} \mu_{i,j}\psi_j$. The uplink channel rate R_i^U is defined as $R_i^U = \frac{W_U}{N}\log_2\left(1 + \frac{p_{a,i}^U d^{-\nu}|h_1|^2}{\frac{W_U}{N}N_0}\right)$. Meanwhile, at the ES, after completing BP, the model parameters are delivered with a downlink communication latency of $T_{g,i}^D = \frac{b\Gamma_{g,i}(\boldsymbol{\mu})}{R_i^D}$ where $\Gamma_{g,i}(\mu) = \sum_{j=1}^{L} \mu_{i,j}\chi_j$. The downlink channel rate R^D is given by $R^D = \frac{W_D}{N}\log_2\left(1 + \frac{p_s d^{-\nu}|h_2|^2}{\frac{W_D}{N}N_0}\right)$. Here, W_U and W_D denote the uplink and downlink channel bandwidths, respectively. p_a, and p_s are the transmit power of ED and ES. d is for distance from ED to ES, v is a path loss exponent, and N_0 is for noise power spectral density. Moreover, the uplink and downlink channel fading coefficients are denoted by h_1 and h_2.

After I training rounds, the locally trained models are simultaneously offloaded from the EDs to the ES with an uplink latency $T_{m,i}^U = \frac{\Lambda_{m,i}(\mu)}{R_i^U(p_{m,i}^U)}$, where $\Lambda_{m,i}(\mu) = \sum_{j=1}^{L} \mu_{i,j}\delta_j$, and δ_j denotes the model data size at ED when the cut layer is at layer j. Subsequently, the EDs receive the aggregated model from the ES with a downlink latency $T_{m,i}^D = \frac{\Lambda_{m,i}(\mu)}{R_i^D}$.

III. Problem Formulation

We jointly optimize the split point $\boldsymbol{\mu} \triangleq \{\mu_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{L}\}$, the aggregation interval I, the uplink transmit power $\boldsymbol{p}^U \triangleq \{p_{a,i}^U, p_{m,i}^U, \forall i \in \mathcal{N}\}$, and the uplink latency $\boldsymbol{T}^U \triangleq \{T_{a,i}^U, T_{m,i}^U, \forall i \in \mathcal{N}\}$ to minimize the total training latency. The total training latency is given by $M \cdot f(I, \boldsymbol{\mu}, \boldsymbol{p}^U, \boldsymbol{T}^U) \triangleq M \cdot [I(\max_i\{T_i^F + T_{a,i}^U\} + T_s^F + T_s^B + \max_i\{T_{g,i}^D + T_i^B\}) + \max_i\{T_{m,i}^U\} + \max_i\{T_{m,i}^D\}]$ where $M \triangleq \frac{R}{I}$ denotes the number of communication rounds required to reach convergence.

The resulting optimization problem is formulated as

$$\text{(P1)}: \underset{I,\boldsymbol{\mu},\boldsymbol{p}^U,\boldsymbol{T}^U}{\text{minimize}} \quad M \cdot f\left(I, \boldsymbol{\mu}, \boldsymbol{p}^U, \boldsymbol{T}^U\right)$$

$$\text{s.t.} \quad \frac{1}{R}\sum_{t=1}^{R}\mathbb{E}\left[\left\|\nabla_{\mathbf{w}} f(\mathbf{w}^{t-1})\right\|^2\right] \leq \epsilon, \tag{1a}$$

$$\mu_{i,j} \in \{0,1\}, \ \forall i \in \mathcal{N}, \ j \in \mathcal{L}, \tag{1b}$$

$$\sum_{j\in\mathcal{L}} \mu_{i,j} = 1, \ \forall i \in \mathcal{N}, \tag{1c}$$

$$I \in \mathbb{N}^+, \tag{1d}$$

$$0 \leq p_{a,i}^U \leq p_{\max} \text{ and } 0 \leq p_{m,i}^U \leq p_{\max}, \ \forall i \in \mathcal{N} \tag{1e}$$

where (1a) guarantees the global convergence accuracy. (P1) is a mixed-integer nonlinear problem, for which obtaining an optimal solution in polynomial time is infeasible.

IV. Proposed Optimization Scheme

In this section, we propose an efficient iterative optimization algorithm for (P1). First, following [4], we reformulate the expression of M in accordance with the convergence accuracy of (1a). Second, noting the one-to-one correspondence between the uplink transmit powers $\{p_{a,i}^U, p_{m,i}^U\}$ and the uplink transmission latencies $\{T_{a,i}^U, T_{m,i}^U\}$, we replace the uplink transmit power variables with their time equivalents. Third, to handle the max functions in the objective of (P1), we introduce the auxiliary variables $\boldsymbol{T} \triangleq [T_1, T_2, T_3, T_4, T_5]$ and use an epigraph reformulation to move the max functions into constraints. Therefore, (P1) can be reformulated as

$$\text{(P2)}: \underset{I,\boldsymbol{\mu},\boldsymbol{T^U},\boldsymbol{T}}{\text{minimize}} \quad \Theta(I, \boldsymbol{\mu}, \boldsymbol{T})$$

$$\text{s.t.} \quad \text{(1b), (1c), (1d)},$$

$$\frac{b\Gamma_{a,i}(\mu)}{R_i^U(p_{\max})} \leq T_{a,i}^U, \quad \forall i \in \mathcal{N}, \tag{2a}$$

$$\frac{\Lambda_{c,i}(\mu)}{R_i^U(p_{\max})} \leq T_{m,i}^U, \quad \forall i \in \mathcal{N}, \tag{2b}$$

$$\sum_{j=1}^{L}\left(\mu_{i,j}\sum_{k=1}^{j} G_k^2\right) \leq T_1, \quad \forall i \in \mathcal{N}, \tag{2c}$$

$$T_i^F + T_{a,i}^U \leq T_2, \quad \forall i \in \mathcal{N}, \tag{2d}$$

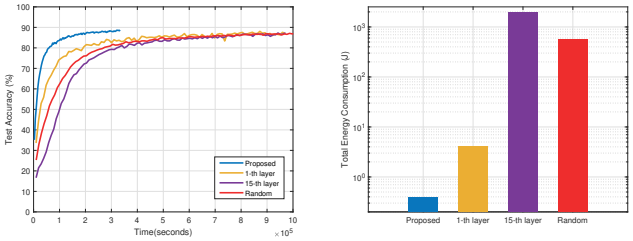
$$T_{g,i}^D + T_i^B \leq T_3, \quad \forall i \in \mathcal{N}, \tag{2e}$$

$$T_{m,i}^U \leq T_4, \quad \forall i \in \mathcal{N}, \tag{2f}$$

$$T_{m,i}^D \leq T_5, \quad \forall i \in \mathcal{N}. \tag{2g}$$

where $\Theta(I, \boldsymbol{\mu}, \boldsymbol{T}) \triangleq \frac{2\vartheta\left\{I\left(T_2+T_s^F+T_s^B+T_3\right)+T_4+T_5\right\}}{\gamma I\left(\varepsilon-\frac{\beta\gamma}{N}\sum_{j=1}^{L}\sigma_j^2-4\beta^2\gamma^2I^2T_1\right)}$.

To solve (P2) efficiently, we decompose it into three subproblems and alternately solve each while keeping the others fixed until convergence. First, to determine $\boldsymbol{T^U}$, we formulate and solve a convex problem that minimizes the squared deviation from their prior values. Second, to determine I, we solve the subproblem of (P2) via the Newton-Raphson method. Third, to determine $\boldsymbol{\mu}$ and $\boldsymbol{T}$, the subproblem of (P2) is solved by using the Dinkelbach algorithm, where $\boldsymbol{\mu}$ and $\boldsymbol{T}$ are jointly optimized.



(a) Test accuracy versus time. (b) Energy consumed over the total training latency.

Fig. 1. Training performance and total transmit energy consumption for CIFAR-10 datasets using VGG-16.

V. Simulation Results

We evaluate the proposed optimization method on the CIFAR-10 dataset using the VGG16 model distributed across 5 EDs. Simulation parameters follow [4]. For performance comparison, we consider baselines where the split point of all EDs is fixed at the first layer, fixed at the 15th layer, and randomly assigned. Under the proposed optimization method, the split points of all EDs are distributed between the 9th and 11th layers. Fig. 1(a) presents the proposed method reaches high accuracy much earlier than all baselines, exhibiting a clearly faster convergence rate throughout training. Fig. 1(b) reports the total transmit energy consumption over reaching convergence. The proposed method yields substantially lower energy than fixed or random split configurations. Overall, the results demonstrate that the proposed method improves both convergence speed and energy efficiency for the SFL framework.

VI. Conclusions

In this paper, we propose an optimization algorithm for the SFL framework, which jointly determines the split point of the model, the aggregation interval, and the uplink transmit power to minimize the total training latency while ensuring the accuracy of the training in resource-constrained MEC networks. Our convergence-aware total training latency optimization algorithm significantly reduces end-to-end training latency while guaranteeing target accuracy.

Acknowledgment

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II210467,Intelligent 6G Wireless Access System)

References

[1] H. Brendan McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in AISTATS, 2017.

[2] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," in NIPS, 2018.

[3] Z. Lin, G. Qu, X. Chen and K. Huang, "Split learning in 6G edge networks, in IEEE Wireless Communications, vol. 31, no. 4, pp. 170-176, Aug. 2024.

[4] Zheng Lin et al., "AdaptSFL: Adaptive split federated learning in resource-constrained edge networks" arxiv:2403.13101.