Loss and Energy Tradeoff in Multi-access Edge Computing Enabled Federated Learning

Chit Wutyee Zaw, Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University,
Yongin-si, Republic of Korea
Email: {cwyzaw, cshong}@khu.ac.kr

Abstract—Federated learning (FL) encourages users to train statistical models on their local devices. Since mobile devices have the limited power and computing capabilities, the users are rational in minimizing their energy consumption with the cost of the model’s accuracy. Multi-access Edge Computing (MEC) enabled FL is a prominent approach where users can offload a fraction of their dataset to the MEC server where the training of the statistical model is performed with the help of the powerful MEC server in parallel with the local training at the mobile users. With the size of dataset offloaded to the MEC server, both the performance of the model and the energy consumption of the system are varied. We analyze this tradeoff between the performance of the system and the energy consumption at the MEC server and mobile users. The time consumption can also be saved by managing the size of the dataset offloaded to the MEC server. Since the MEC server and mobile users have the conflicting interest in saving the energy consumption with the constraint on the time taken for one computing round where the performance of the model fluctuates across the size of offloaded dataset, we analyze the tradeoff by formulating the resource management problem as a penalized convex optimization problem. We propose a distributed resource management problem for MEC enabled FL system where the global model is responsible for radio resource management and each local model performs a dataset offloading decision. Then, we perform the simulation to show the tradeoff between the performance of the model jointly where the convergence of the algorithm is considered the wireless resource allocation and the learning training is widely addressed in recent works. Authors in [5] considered the wireless resource allocation and the learning model jointly where the convergence of the algorithm is derived. An optimization model for FL is proposed in [6] in which authors discussed a tradeoff between learning time and the energy consumption of users. The edge association of users to the servers is considered in [7] where the energy and time are jointly minimized by allocating the wireless bandwidth and local computing resources. Authors in [8] considered a joint data sampling and participant selection for FL in mobile edge computing system where the model training loss is jointly minimized with the cost of the training. However, authors did not take the time consumption into account which is an important factor to consider in the resource allocation. Moreover, the distributed resource allocation approach for the MEC system is discussed in [9] and [10].

A. Related Works

The wireless resource allocation with respect to the model training is widely addressed in recent works. Authors in [5] considered the wireless resource allocation and the learning model jointly where the convergence of the algorithm is derived. An optimization model for FL is proposed in [6] in which authors discussed a tradeoff between learning time and the energy consumption of users. The edge association of users to the servers is considered in [7] where the energy and time are jointly minimized by allocating the wireless bandwidth and local computing resources. Authors in [8] considered a joint data sampling and participant selection for FL in mobile edge computing system where the model training loss is jointly minimized with the cost of the training. However, authors did not take the time consumption into account which is an important factor to consider in the resource allocation. Moreover, the distributed resource allocation approach for the MEC system is discussed in [9] and [10].

B. Our Contribution

In this paper, we propose a MEC enabled federated learning model by utilizing the computation capability of the edge server. The participating users decide the fraction of the local dataset to offload to the edge server to train a statistical model while training the local model on the remaining dataset simultaneously. The edge server trains a model on the datasets
offloaded from the users in parallel with the local training. We formulate a distributed resource management problem where the edge server controls the uplink bandwidth allocation for the dataset offloading and weight transmission while the mobile users decide the datasets to offload to the edge server. The objective of the edge server and mobile users is to minimize the cost which is the combination of the model training loss and the energy consumption within the specified time duration. A distributed resource management algorithm is proposed to solve the formulated problem. The simulation is performed to show the performance of the algorithm.

II. SYSTEM MODEL

In this paper, we consider the MEC enabled federated learning model in which an edge server is responsible for training a statistical model simultaneously with the participating mobile users. By utilizing the edge sever in MEC system, users can offload a portion of their dataset to train a model at the edge server so that the users can save the energy consumption of their devices. Our MEC enabled FL system consists of an edge server, $E$, and a set of mobile users, $I = \{1, 2, \cdots, n\}$. Each user, $i$, has a local dataset $D_i$ to train a statistical model by participating in FL. The MEC enable FL system model is shown in Fig. 1 where user $i$ offloads a portion of their dataset, $\delta_i$, to the edge server, $E$, and train the local model with the remaining portion of dataset $(1 - \delta_i)$. User $i$ determines $\delta_i$ independently with other users and the edge server, $E$. The bandwidth allocation is performed by the edge server, $E$, to offload the dataset and parameters of the local training model at users.

The time consumption of our proposed MEC enabled FL model and traditional FL model is shown in Fig. 2. In traditional FL, mobile users participating in FL train their local models and transfer the weights of local model after training. The edge node, which is considered the edge server in our paper, aggregates the weights from all participating users after receiving from all users. In our proposed model, user $i$ can upload $\delta_i$ portion of the local dataset and train the local model with $(1 - \delta_i)$ portion of dataset simultaneously. After receiving the offloaded dataset from all users, the edge server trains a model with the offloaded dataset from all users. The model aggregation is performed after the edge server and mobile users finish the training, and the weight transmission. For both traditional and proposed MEC enabled FL, the synchronous update model is considered.

A. Federated Learning Model

In FL, user $i$ trains a statistical model with the local dataset, $D_i$, by minimizing the loss function which is defined as follows.

$$\text{minimize}_{w_i} \sum_{j \in D_i} l(w_i, x_j, y_j),$$

where $w_i$ is the weight vector which captures the local model, $x_j$ and $y_j$ are the features and label of sample $j$ in the local dataset $D_i$. In this paper, we perform the linear regression model where the Mean Squared Error is considered as the loss function. After training the local model, users transfer the model which is the weight vector to the edge server. The edge server then aggregate the weight vectors from all participating users and the final model, $w_f$ is obtained as follows.

$$w_f = \frac{\sum_{i \in I} |D_i| w_i}{\sum_{i \in I} |D_i|},$$

where the local weights, $w_i$, are adjusted by the contribution of the user $i$ in the model training which is the proportion of local dataset with respect to sum of all local datasets of users. However, in our proposed MEC enabled FL system, the edge server perform the model training as well with the data offloaded from the participating users which is described in following subsections.

B. Local Model

In this section, we define the local model training, energy and time consumption of the participating mobile devices. In our proposed MEC enabled FL system, user $i$ is allowed to offload $\delta_i$ portion of the local dataset to the edge server $E$. The remaining $(1 - \delta_i)$ portion of dataset is used for the local training. Let $D_i$ be the dataset offloaded to the edge server $E$ where the data samples are randomly chosen from $D_i$ and $|D_i| = \delta_i|D_i|$. Let $\bar{D}_i$ be the dataset to train the local model of user $i$ where $\bar{D}_i = D_i - D_i$, $|\bar{D}_i| = (1 - \delta_i)|D_i|$ and $D_i = \bar{D}_i \cup D_i$. Thus, user $i$ performs the training of the local model as follows.

$$\text{minimize}_{w_i} \sum_{j \in D_i} l(w_i, x_j, y_j).$$

To participate in our MEC enabled FL system, user $i$ needs to perform three steps which are the dataset transmission, local model training and model (weight) transmission. In this paper, we consider Orthogonal Frequency Division Multiple Access
for communication. Thus, the energy consumption of user $i$ is calculated as follows.

$$
e_i(\tilde{\beta}_i, \hat{\beta}_i, \delta_i) = p_i \left( \frac{\delta_i c_d |D_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})} + \frac{c_w |w_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})} \right) + \xi(1 - \delta_i) c_d |D_i| \tilde{c}_f^2, \tag{2}$$

where $p_i$ is the transmission power of user $i$, $h_i$ is the channel gain, $n_0$ is noise power, $\xi$ is chip capacitance, $c_f$ is the CPU cycles required for computing one byte data, $c_d$ and $c_w$ are the size of a data sample and weight vector respectively. $\beta_i$ and $\hat{\beta}_i$ are the uplink bandwidth allocation for the dataset offloading and weight transmission respectively.

The time consumption for the dataset offloading, local training and weight transmission is defined as follows. The time taken for offloading the dataset of size $\delta_i |D_i|$ is

$$t_{\text{offload}}^i(\tilde{\beta}_i, \delta_i) = \frac{\delta_i c_d |D_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})}. \tag{3}$$

The time required to perform the local training with the dataset of size $(1 - \delta_i) c_d |D_i|$ is

$$t_{\text{training}}^i(\delta_i) = \frac{(1 - \delta_i) c_d |D_i| c_f}{F_i}, \tag{4}$$

where $F_i$ is the CPU cycle speed of the mobile device of user $i$. The time consumption for the weight transmission is calculated as follows.

$$t_{\text{trans}}^i(\hat{\beta}_i) = \frac{c_w |w_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})}. \tag{5}$$

C. Edge Model

After receiving the offloaded dataset, $\tilde{D}_i, \forall i \in I$, the edge server performs the model training where the loss function with respect to the offloaded datasets is as follows.

$$\text{minimize}_{w_i} \sum_{j \in \tilde{D}_i} l(w_i, x_j, y_j), \tag{6}$$

where $\tilde{D}_E = \bigcup_{i \in I} \tilde{D}_i$. The energy consumption of the edge server depends on the traffic and computing load at the edge server. The traffic load on the uplink transmission for the dataset offloading and weight transmission is defined as $
abla = \frac{\sum_{i \in I} \delta_i c_d |D_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})}$ and $\tilde{\nabla} = \frac{\sum_{i \in I} c_w |w_i|}{\beta_i \log_2 (1 + \frac{p_i h_i}{n_0})}$ respectively. The computing load at the edge server for training the model is same as the traffic load for dataset offloading because the offloaded datasets are used in the model training.

$$e_{E}(\tilde{\beta}, \beta, \delta) = (\nabla + \tilde{\nabla}) P_{\text{BS}} + \tilde{\nabla} P_{\text{MEC}}, \tag{7}$$

where $P_{\text{BS}}$ and $P_{\text{MEC}}$ are the maximum available power of the edge base station and MEC edge server, and $\beta = [\beta]_{i \in I}, \tilde{\beta} = [\tilde{\beta}]_{i \in I}, \delta = [\delta]_{i \in I}$ which are the vector representation of the resource management variables. The time consumption for training the model at the edge server with respect to all the offloaded dataset is calculated as follows.

$$t_{\text{training}}^E(\delta) = \frac{\sum_{j \in T} (1 - \delta_j) c_d |D_j| c_f}{F_E}, \tag{8}$$

where $F_E$ is CPU cycle speed of the edge server.

D. Model Aggregation

Once the mobile users and the edge server finish the training their models, the model aggregation is performed which is the summation of the weights with respect to their contribution in FL. The model aggregation is performed as follows.

$$w_f = \frac{1}{\sum_{i \in I} |D_i|} \left( \sum_{i \in I} |D_i| w_i + \sum_{i \in I} \tilde{D}_i w_E \right).$$

III. Problem Formulation

In this section, the resource management problem is formulated where the mobile users and the edge server minimize their cost which is defined by the loss function and their energy consumption. The mobile users and the edge server manage their resources which are the portion of dataset offloaded to the edge server and the uplink bandwidth allocation respectively so as to minimize their cost. Let $t_i$ be the time taken for one computing round which is the time taken for the traditional FL approach.

$$t_i(\tilde{\beta}_i, \delta_i) = t_{\text{training}}^i(\delta_i) + t_{\text{trans}}^i(\hat{\beta}_i). \tag{9}$$

Let $t_E$ be the time taken for the edge server to train the model with all the offloaded datasets.

$$t_E(\tilde{\beta}, \delta) = \max_{\tilde{\beta}, \delta} \left\{ t_i(\tilde{\beta}_i, \delta_i) \right\} + t_{\text{training}}^E(\delta), \tag{10}$$

where $\max_{\tilde{\beta}, \delta} \left\{ t_i(\tilde{\beta}_i, \delta_i) \right\}$ is taken since the edge server can begin the model training after receiving all the datasets offloaded from the mobile users. Thus, the total time taken for the proposed MEC enabled FL model is defined as

$$T(\tilde{\beta}, \beta, \delta) = \max_{i \in E} \left\{ t_i(\tilde{\beta}_i, \delta_i), t_{\text{training}}^E(\delta) \right\}. \tag{11}$$

Thus, the resource management problem for the MEC enabled FL to minimize the cost function by managing the uplink bandwidth, the offloaded dataset is formulated as follows.

The objective of the edge server is to minimize the cost function which is the sum of the model training loss and energy consumption by allocating the uplink bandwidth for dataset offloading and weight transmission within the specified time period, $\Delta t$, which is defined as follows.

$$\text{minimize}_{w_i, \delta_i} \sum_{j \in \tilde{D}_i} l(w_i, x_j, y_j) + e_{E}(\tilde{\beta}, \beta, \delta) \tag{12}$$

subject to $T(\tilde{\beta}, \beta, \delta) \leq \Delta t$.

User $i$ management the portion of dataset offloaded to the edge server, $\delta_i$, so as to minimize the loss of the model training and energy consumption. The resource management problem for user $i$ is formulated as follows.

$$\text{minimize}_{w_i, \delta_i} \sum_{j \in \tilde{D}_i} l(w_i, x_j, y_j) + e_i(\tilde{\beta}_i, \hat{\beta}_i, \delta_i) \tag{13}$$

subject to $T(\tilde{\beta}, \beta, \delta) \leq \Delta t$,
The formulated resource management problem is hard to solve due to the coupling in $T(\bar{\beta}, \tilde{\beta}, \delta)$ between the edge server and users. In addition, the loss function is not guaranteed to be convex with respect to the dataset used for training which is later shown in section V. Thus, we propose a penalized distributed algorithm where the model learning and the resource management problem are solved alternatively.

IV. DISTRIBUTED RESOURCE MANAGEMENT ALGORITHM

In this section, a distributed resource management algorithm is proposed for the MEC enabled FL system. Since there is a coupling between the model training and the portion of datasets offloaded to the edge server, $\delta_i$, we decouple the model training and the resource management problem.

User $i, \forall i \in \mathcal{I}$ and the edge server perform the model training by solving (1) and (6) with the fixed $\delta$. The offloaded decision, $\delta$ is updated by solving the following problems which are solved by the edge server and user $i, \forall i \in \mathcal{I}$ simultaneously. The resource management problem for the edge server is defined as follows.

$$\min_{\beta, \bar{\beta}} e_E(\bar{\beta}, \tilde{\beta}, \delta)$$
subject to $T(\bar{\beta}, \tilde{\beta}, \delta) \leq \Delta t$. (14)

The resource management problem for the user $i$ is defined as

$$\min_{\delta_i} e_i(\tilde{\beta}_i, \bar{\beta}_i, \delta_i)$$
subject to $T(\tilde{\beta}_i, \bar{\beta}_i, \delta) \leq \Delta t$. (15)

Since there is a resource coupling in $T(\bar{\beta}, \tilde{\beta}, \delta)$ in (14) and (15), the formulated resource management problem is transformed by setting a penalty parameter to the time constraint as follows. Let $\bar{T}(\bar{\beta}, \tilde{\beta}, \delta) = T(\tilde{\beta}, \bar{\beta}, \delta) - \Delta t$. The penalized resource management problem for the edge server is

$$\min_{\beta, \bar{\beta}} e_E(\bar{\beta}, \tilde{\beta}, \delta) + \lambda_E \bar{T}(\bar{\beta}, \tilde{\beta}, \delta),$$ (16)

where $\lambda_E$ is the penalty parameter controlled by the edge server with respect to the time constraint. (16) is convex with respect to $\bar{\beta}, \tilde{\beta}$ for the fixed $\delta$ and $\lambda_E$. The penalized resource management problem for user $i$ is formulated as

$$\min_{\delta_i} e_i(\tilde{\beta}_i, \bar{\beta}_i, \delta_i) + \lambda_i \bar{T}(\bar{\beta}_i, \tilde{\beta}_i, \delta),$$ (17)

where $\lambda_i$ is the penalty parameter controlled by user $i$ to prevent the violation of time constraint. (17) is a convex problem with respect to $\delta_i$ for the fixed $\tilde{\beta}_i, \bar{\beta}_i$ and $\lambda_i$. Thus, (16) and (17) can be solved efficiently. The distributed resource management algorithm is proposed in algorithm 1.

Let $\epsilon_i^k$ and $\epsilon_i^k$ be the difference of the objective values between iteration $k$ and $k - 1$ for the edge server and user $i, \forall i \in \mathcal{I}$ respectively which are calculated as follows.

$$\epsilon_i^k = |(\sum_{j \in D_i^{k-1}} l(\mathbf{w}_i, x_j, y_j) + e_i(\tilde{\beta}_i^{k-1}, \bar{\beta}_i^{k-1}, \delta_i^{k-1})) - (\sum_{j \in D_i^k} l(\mathbf{w}_i, x_j, y_j) + e_i(\tilde{\beta}_i^k, \bar{\beta}_i^k, \delta_i^k))|$$

The penalty parameters, $\lambda_E$ and $\lambda_i \forall i \in \mathcal{I}$ are updated as follows.

$$\lambda_E^k = \begin{cases} \lambda_E^{k-1}, & \text{if } T(\tilde{\beta}_E^{k-1}, \bar{\beta}_E^{k-1}, \delta^{k-1}) \leq \Delta t \\ \lambda_E^{k-1} + \gamma_E, & \text{otherwise} \end{cases}$$ (18)

$$\lambda_i^k = \begin{cases} \lambda_i^{k-1}, & \text{if } T(\tilde{\beta}_i^{k-1}, \bar{\beta}_i^{k-1}, \delta^*) \leq \Delta t \\ \lambda_i^{k-1} + \gamma_i, & \text{otherwise} \end{cases}$$ (19)

where $\gamma_E$ and $\gamma_i$ are the increment for the penalty parameters so that the penalty values will keep increasing if the constraint is violated.

V. SIMULATION RESULTS

In this paper, we consider a single cell scenario with an edge server is deployed together with a macro basestation. We perform a linear regression model on 100 mobile users which are randomly located in 1 km radius. The total available uplink bandwidth is 20 MHz and the power density thermal noise is used which is -174 dBm/Hz. The edge server is equipped with a 16 GHz CPU and the value of $\Delta t$ is 30 s. The edge server is denoted as global model and the mobile users are denoted as local model in the following figures.

Fig. 3 shows the training loss and time consumption between the traditional and proposed FL. Since the data samples
for the offloaded dataset is chosen randomly, the training loss of the models varied. The fraction of offloaded data of value 0.0 means that the local dataset is not offloaded to the edge server and trained at the local model whereas the value 1.0 means the whole dataset is offloaded to the edge server. The training loss of the most cases, except for values 0.8 and 0.9, is less than the traditional FL in which the dataset is not offloaded to the edge server. When a small fraction of dataset is offloaded to the edge server, the resulting time consumption is less than that of traditional FL because the edge server is equipped with powerful computation capability and the local model is training the model with less data samples. However, as the fraction of offloaded dataset increases after a threshold, the time consumption also increases due to the increased load on the uplink transmission.

Fig. 3 shows the training model loss and energy consumption of the global and local model. The energy consumption of the global model increases with the fraction of offloaded dataset while the local model’s energy consumption decreases. In some cases, loss is also decreasing with the energy consumption of local model where the global model’s energy consumption is compromised. Since both the edge server and mobile users want to minimize the energy consumption, the fraction of dataset offloaded needs to be optimized to satisfy the objectives.

Fig. 5 shows the convergence of the algorithm on the training loss and energy consumption of the global and local model. The training loss has to be increased by a small amount in order to save the energy consumption due to the tradeoff between them. The algorithm converges in a few iteration. Fig. 6 shows the fraction of offloaded dataset and bandwidth allocation of the cell-edge and cell-center users. Since the cell-edge user is situated far from the edge server, only a small fraction of the dataset is offloaded to the edge server so that the only few bandwidth resources are needed for the offloading. However, for the cell-center user, almost all the dataset is offloaded to the edge server in which a large fraction of bandwidth resources is required for the transmission.

VI. CONCLUSION

In this paper, we propose a Multi-access Edge Computing enabled Federated Learning where the computing power of the edge server is utilized by performing the model training at the server with the datasets offloaded by the participating users. The resource management problem is formulated where the edge server optimizes the uplink bandwidth allocation for dataset offloading and weights transmission by minimizing its cost function which is the combination of the training model loss and energy consumption. The participating users control the fraction of dataset offloaded to the edge server where the local training loss and energy consumption are minimized. For the formulated problem, we propose a distributed resource management algorithm by setting a penalty parameter to the coupling constraints so as to solve the formulated problem independently. The simulation is then performed to show the performance of the proposed algorithm, the coupling in the resource management, time and the energy consumption.

REFERENCES


