Fast-Convergent Federated Learning with Adaptive Weighting

Hongda Wu, Ping Wang
Department of Electrical Engineering and Computer Science, York University, Canada
hwu1226@eecs.yorku.ca, pingw@yorku.ca

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500890

Abstract—**Federated learning (FL) enables resource-constrained edge nodes to collaboratively learn a global model under the orchestration of a central server while keeping privacy-sensitive data locally. The non-independent-and-identically-distributed (non-IID) data samples across participating nodes slow model training and impose additional communication rounds for FL to converge. In this paper, we propose `Federated Adaptive` Weighting (`FedAdp`) algorithm that aims to accelerate model convergence under the presence of nodes with non-IID dataset. Through mathematical and empirical analysis, we observe the implicit connection between the gradient of local training and data distribution on local node. We then propose to assign different weight for updating global model based on node contribution adaptively through each training round, which is measured by the angle between local gradient vector and global gradient vector, and is quantified by a designed non-linear mapping function. The simple yet effective strategy can reinforce positive (suppress negative) node contribution dynamically, that results in communication round reduction drastically. With extensive experiments performed in Pytorch and PySyft, we show that FL training with `FedAdp` can reduce the number of communication rounds by up to 54.1% on MNIST dataset and up to 45.4% on FashionMNIST dataset, as compared to the commonly adopted Federated Averaging (`FedAvg`) algorithm.**

I. Introduction

The rapid advancement of mobile devices equipped with enhanced computational capability is constantly generating unprecedented amount of data. A model learned on such data has the prospect of greatly improving the user experience. However, collecting data for centralized model training is unrealistic from a privacy, security, regulatory or necessity point of view. Federated Learning (FL) has emerged as an attractive paradigm for model training, where local nodes collaboratively train a task model under the orchestration of a central server without accessing end-user data [1]. FL has been deployed by major service providers and plays an important role in supporting privacy-sensitive applications including computer vision, natural language processing, and medical database [2].

Even though good convergence performance is shown, owing to limited connectivity of wireless network and availability of local nodes, communication cost becomes a critical bottleneck in FL context since generally serval iterations are involved for model converging [2]. Generally, FL algorithm adopts synchronous aggregation and selects a subset of nodes to participate in each round randomly to avoid long-tailed waiting time due to the network uncertainty and straggler. To boost convergence and reduce the communication rounds, McMahan *et al.* [1] presented the vanilla Federated Averaging (`FedAvg`) algorithm, which increases the number of local updates instead of updating the local model one time at each round. Nishio *et al.* [3] proposed `FedCS` algorithm to do node selection intentionally rather than randomly, based on the resource conditions of local nodes. The work in [4] exposed an analytical model to dynamically adapt the frequency of global aggregation in real-time to minimize the learning loss under a fixed resource budget of the edge computing system.

Another fundamental challenge for FL is strongly non-independent-and-identically-distributed (non-IID) and highly skewed data across local nodes. The presence of non-IID data significantly degrades the performance of federated learning, which makes model training take more rounds to converge and the variance caused by non-IID data brings instability to the training process [5] [6]. Zhao *et al.* [5] quantified the weight divergence by earth movers distance between data distribution on nodes and population distribution. However, the strategy of pushing a small set of uniform distributed data to participating nodes in [5] violates the privacy concern of FL and imposes extra communication cost. It was proposed in [6] that communication rounds can be reduced by selecting nodes based on their uploaded model weight, which profiles the data distribution on that node. A deep reinforcement learning agent is trained at the central server-side for node selection. The communication rounds are reduced effectively. In contrast, Wang *et al.* [7] proposed to identify the irrelevant update caused by different data distribution at the node side. The communication cost is accordingly reduced by precluding participated nodes before updates transmission. However, local node checks the relevance in each round using the global model kept in the previous round, which is in contravention of FL and brings computational burdens to local nodes.

Another related work is [8], which utilized gradient information to do node selection. In particular, the nodes whose inner product between its gradient vector and global gradient vector is negative will be excluded from FL training. Chen *et al.* [9] proposed to assign different weights for global model aggregation adaptively by considering the time difference when the model update is done in a layerwise asynchronous manner. Differently, we propose to use the angle between the local gradient of participating node and global gradient as a metric to measure the node contribution quantitatively. By which, the weight for aggregating global model can be devised discriminatively across the node and adaptively in each round

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

according to node contribution.

In this paper, to surmount the slow convergence of `FedAvg` under the presence of non-IID dataset, we propose `Fed`erated `Ad`a`p`tive Weighting (`FedAdp`) algorithm that aims to improve the performance of federated learning through assigning distinct weight for participating node to update the global model. We observe that nodes with heterogeneous datasets make different contributions to the global model aggregation. Therefore, our main intuition is to measure the contribution of participating node based on the gradient information from local nodes then assign different weight accordingly and adaptively at each communication round for global model aggregation. The proposed adaptive weighting strategy according to node contribution is capable to reduce the expected training loss of FL in each communication round under the presence of non-IID nodes, which accelerates the model convergence.

We have implemented `FedAdp` in a federated learning simulator developed based on Pytorch and PySyft, and evaluated it under a variety of federated learning tasks. Our experimental results on the MNIST and FashionMNIST datasets have shown that FL training with `FedAdp` can reduce the communication rounds by up to 54.1% and 45.4%, respectively, as compared with the commonly adopted `FedAvg` algorithm.

II. Preliminaries

In this section, we briefly introduce key ingredients behind the recent method for federated learning, `FedAvg`, and show how non-IID data gives an impact on model convergence.

A. Federated Learning

In general, federated learning methods [1] [2] are designed to handle the consensus learning task in a decentralized manner, where a central server coordinates the global learning objective and multiple devices training local model with locally collected data. In particular, assume that we have N local nodes with dataset $\mathcal{D}_1, ..., \mathcal{D}_i, ..., \mathcal{D}_N$ and we define $D_i \triangleq |\mathcal{D}_i|$, where $|\cdot|$ denotes the size of the dataset, and $D \triangleq \sum_{i=1}^{N} D_i$, FL methods aim to minimize:

$$\min_{\mathbf{w}} \quad F(\mathbf{w}) \triangleq \sum_{i=1}^{N} \psi_i F_i(\mathbf{w}) = \mathbb{E}_i[F_i(\mathbf{w})], \tag{1}$$

where $\mathbf{w}$ is global model weight, $\psi_i = D_i/D$ is the weight for aggregation in FL training, and global objective function $F(\mathbf{w})$ is surrogated by using local objective function $F_i(\mathbf{w})$, which is defined, as an example, in the context of C-class classification problem thereinafter. In particular, C-class classification problem is defined over a feature space $\mathcal{X}$ and a label space $\mathcal{Y} = [C]$, where $[C] = \{1, \cdots, C\}$. For each labeled data sample $\{\mathbf{x}, y\}$, predicted probability vector $\widetilde{\mathbf{y}}$ is achieved by using mapping function $f : \mathcal{X} \rightarrow \widetilde{\mathcal{Y}}$, where $\widetilde{\mathcal{Y}} = \{\widetilde{\mathbf{y}} | \sum_{j=1}^{C} \widetilde{y}_j = 1, \widetilde{y}_j \geq 0, \forall j \in [C]\}$. As such, $F_i(\mathbf{w})$ commonly measures the local empirical risk over possibly differing data distribution $p^{(i)}$ of node i, which is defined by using cross entropy for C-class classification as follow,

$$\begin{aligned}\min_{\mathbf{w}} \quad F_i(\mathbf{w}) &\triangleq \mathbb{E}_{\mathbf{x},y\backsim p^{(i)}}\left[-\sum_{j=1}^{C} \mathbb{1}_{y=j} \log f_j(\mathbf{x}, \mathbf{w})\right] \\ &= -\sum_{j=1}^{C} p^{(i)}(y=j) \mathbb{E}_{\mathbf{x}|y=j}\left[\log f_j(\mathbf{x}, \mathbf{w})\right], \end{aligned} \tag{2}$$

where $f_j(\mathbf{x}, \mathbf{w})$ denotes the probability that the data sample $\mathbf{x}$ is classified as the j-th class given model $\mathbf{w}$, and $p^{(i)}(y=j)$ denotes the data distribution on node i over class $j \in [C]$.

In `FedAvg` [1], the participating nodes perform local training with the same training configuration (e.g. optimizer, learning rate, etc). At each communication round t, a subset of the nodes $\mathcal{S}_t, |\mathcal{S}_t| = K \ll N$ are selected and global model $\mathbf{w}(t-1)$ in previous iteration is sent to the selected nodes. Each of the participating nodes i performs stochastic gradient descent (SGD) training to optimize its local objective $F_i(\mathbf{w})$:

$$\mathbf{w}_i(t) = \mathbf{w}(t-1) - \eta \mathbf{g}_i(\mathbf{w}(t-1)), \tag{3}$$

where η is the learning rate and $\mathbf{g}_i(\cdot)$ is the gradient at node i. $\mathbf{w}_i(t)$ refers to the result after τ number of local updates.

The nodes then communicate their local model updates $\Delta_i(t) = \mathbf{w}_i(t) - \mathbf{w}(t-1)$ to the central server, which aggregates them and updates the global model accordingly,

$$\begin{aligned} \Delta(t) &= \sum_{i=1}^{K} \psi_i \Delta_i(t) \\ \mathbf{w}(t) &= \mathbf{w}(t-1) + \Delta(t). \end{aligned} \tag{4}$$

B. `FedAvg` *for non-IID data*

The independent and identically distributed (IID) sampling condition of training data is important that the stochastic gradient is an unbiased estimate of the full gradient [4]. `FedAvg` is shown to be effective given that the data distribution across different nodes is the same as centrally collected data. However, the data distribution determined by usage patterns across local nodes is typically non-IID, i.e., $p^{(i)}$ is different across participating nodes.

Since local objective $F_i(\mathbf{w})$ is closely related with data distribution $p^{(i)}$, a large number of local updates lead the model towards optima of its local objective $F_i(\mathbf{w})$ as opposed to the global objective $F(\mathbf{w})$. The inconsistency between local models $\mathbf{w}_i$ and global model $\mathbf{w}$ is accumulated along with local training, leading to more communication rounds before training converges. As such, local training with multiple local updates potentially hurts convergence and even leads to divergence with the presence of non-IID data [1] [5].

We conduct an experiment to demonstrate the impact of non-IID data on model convergence. We train a two-layer CNN model with the same neural network architecture in [1] using Pytorch on the MNIST dataset (containing 60,000 samples with 10 classes) until the model achieves 95% test accuracy. 10 nodes are selected, each with 600 samples that are selected based on their label criteria. If a node is at *IID setting*, 600 samples are randomly selected over the whole training set. If a node is at *x-class non-IID setting*, 600 samples are

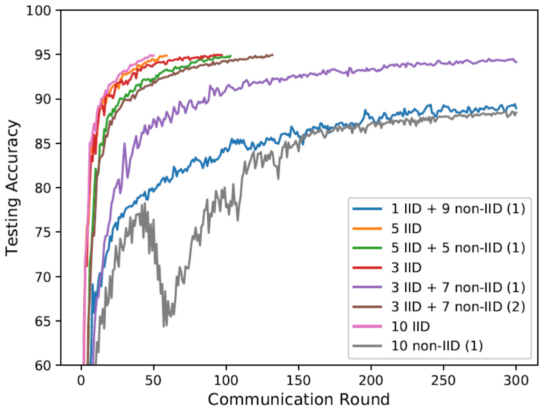

Fig. 1. Test accuracy over communication rounds of `FedAvg` with heterogeneous data distribution over participating nodes. X IID + Y non-IID (1) (or (2)) represents X nodes are at *IID setting* and Y nodes are at *1-class (or 2-class) non-IID setting*

randomly selected over a subset, which is composed of x class data samples. Each class of the x-class is selected at random and can be overlapped. The skewness of datasets is measured and reflected by the value of x.

We use the same notations for `FedAvg` algorithm as [1]: B, the batch size, and E, the number of local epochs. In this experiment, $B = 32$, $E = 1$, $\eta = 0.01$ and learning rate decay of 0.995 per communication round. We can conclude from Fig. 1:

- Model convergence highly depends on IID nodes. The presence of non-IID nodes imposes variance to model training, which slows the convergence of FL (e.g., 5 IID case converges faster than 5 IID + 5 non-IID (1) case).
- The skewness of data affects model convergence. With the participation of the non-IID node, the model converges much slower when the skewness of the dataset increases (e.g., 3 IID + 7 non-IID (2) case converges much faster than 3 IID + 7 non-IID (1) case).

III. Federated Adaptive Weighting

In this section, we first analyze the weight divergence of FL caused by non-IID distribution across different local nodes. The upper bound of the expected decrease in each training round is related to the difference in data distribution and weighting strategy. Since the gradient of different nodes is not always aligned with the global gradient due to non-IID datasets, the contribution of each node can be quantified using the angle between the local gradient of the participating node and the global gradient. We propose `Federated Adaptive` Weighting (`FedAdp`) algorithm aiming to reinforce positive or weaken negative contribution of participating nodes, which accelerates model to convergence.

A. Weight Divergence

In FL, the weight after multiple local updates will diverge from the global weight since optimization direction on local node is respect to its own objective, which might be different due to different data distribution.

Let $\mathbf{v}(t)$ denote the auxiliary parameter vector that is optimized in the centralized setting. The update of centralized SGD is as follow:

$$\mathbf{v}(t) = \mathbf{v}(t-1) - \eta \sum_{j=1}^{C} p(y=j) \nabla_{\mathbf{v}} \mathbb{E}_{\mathbf{x}|y=j} \left[\log f_j(\mathbf{x}, \mathbf{v}(t-1)) \right], \tag{5}$$

where $p(y = j)$ is the population distribution over class j.

The above rule is based on the global loss function $F(\mathbf{w})$, which is only observable when all data samples are available at a central place. We define that $\mathbf{v}(t)$ is "synchronized" with $\mathbf{w}(t)$ at the beginning of local updates between two consecutive global aggregations, i.e., $\mathbf{v}(t) \triangleq \mathbf{w}(t), t = m\tau$. For the purpose of analysis, we make the following assumption to the loss function:

Assumption 1. *For each of the participating nodes,*

- $F_i(\mathbf{w})$ *is* ρ*-Lipschitz continuous,*
  i.e., $\|F_i(\mathbf{w}) - F_i(\mathbf{w}')\| \leq \rho \|\mathbf{w} - \mathbf{w}'\|$ *for any* $\mathbf{w}$, $\mathbf{w}'$;
- $F_i(\mathbf{w})$ *is* β*-smooth,*
  i.e., $\|\nabla F_i(\mathbf{w}) - \nabla F_i(\mathbf{w}')\| \leq \beta \|\mathbf{w} - \mathbf{w}'\|$ *for any* $\mathbf{w}$, $\mathbf{w}'$.

Based on Assumption 1, the definition of $F(\mathbf{w})$, and triangle inequality, we can easily get the following lemma.

Lemma 1. $F(\mathbf{w})$ *is* ρ*-Lipschitz continuous and* β*-smooth.*

Following the similar bounding technique adopted in [5], the upper bound of weight divergence is derived as follows (the detailed derivation steps are skipped due to the space limit):

If global aggregation is conducted every τ local updates, given $|\mathbb{S}_t|$ nodes, we have the following inequality for the weight divergence between FL model and the centralized model after the m-th global aggregation,

$$\begin{aligned} &\|\mathbf{w}(m\tau) - \mathbf{v}(m\tau)\| \leq \\ &\eta \mathbb{E}_{i \in \mathbb{S}_t} \left[\widetilde{\psi}_i p_{dif}^{(i)} (\sum_{k=1}^{\tau-1} {\alpha^{(i)}}^k g_{max}(\mathbf{v}(m\tau - 1 - k))) \right], \end{aligned} \tag{6}$$

where $\alpha^{(i)} = 1 + \eta\beta \sum_{j=1}^{C} p^{(i)}(y = j))$, $g_{max}(\mathbf{v}) = max_{i=1}^{C} \|\nabla_{\mathbf{v}} \mathbb{E}_{\mathbf{x}|y=j} \left[\log f_j(\mathbf{x}, \mathbf{v}) \right] \|$, $\widetilde{\psi}_i$ is the weight for global model aggregation, $p_{dif}^{(i)} = \sum_{j=1}^{C} (p^{(i)}(y = j) - p(y = j))$ is the difference between data distribution on nodes i and population distribution, and $\| \cdot \|$ denotes the $\ell 2$ norm of a vector.

Furthermore, as $F(\mathbf{w})$ is ρ-Lipschitz continuous, we have the difference between the expected loss in FL after τ local updates and that in the centralized model as follows:

$$\begin{aligned} &\|F(\mathbf{w}(m\tau)) - F(\mathbf{v})(m\tau))\| \leq \\ &\eta \rho \mathbb{E}_{i \in \mathbb{S}_t} \left[\widetilde{\psi}_i p_{dif}^{(i)} (\sum_{k=1}^{\tau-1} {\alpha^{(i)}}^k g_{max}(\mathbf{v}(m\tau - 1 - k))) \right]. \end{aligned} \tag{7}$$

From (7), it is concluded that: *After the* m*-th global aggregation, the deviation of the global loss of FL model compared with the centralized model is affected by learning rate* η*, the number of local updates* τ*, the difference between data* *distribution on nodes i and population distribution $p_{dif}^{(i)}$, and the weight $\widetilde{\psi}_i$ assembled with node i for global aggregation.*

To accelerate model convergence and minimize communication costs, one can intuitively start with reducing the expected loss of FL in each round. From the above analysis, we note that the weight $\widetilde{\psi}_i$ impacts the expected loss of FL. One question naturally arises: Can we devise one weighting strategy, that is capable to help *reduce the expected loss of FL and accelerate the model convergence*?

B. Aggregation with Gradient Information

In FL, the direction of minimizing local objective $F_i(\mathbf{w})$ might not align with the direction of minimizing $F(\mathbf{w})$. In particular, model update from a node is closely related to its gradient $\mathbf{g}_i$. It can be deduced from (3) that the gradient on different nodes may be tremendously diverse, especially for heterogeneous datasets across participating nodes. As such, the contribution from participating node for global aggregation is different. From our experiment, we note that if the data distribution on a node is highly skewed, the gradient of which may highly deviate from or even in the opposite direction to the global gradient, causing a negative effect on the global aggregation.

Instead of assigning weight for participating nodes based on the size of datasets as in `FedAvg` [1], we measure the contribution of participating nodes based on the correlation between local gradient and global gradient. Particularly, we quantify the contribution of each node at each communication round based on *angle* $\theta_i(t)$, that is defined as:

$$\theta_i(t) = arccos\left(\frac{\langle\, \mathbf{g}_i(\mathbf{w}(t)), \mathbf{g}(\mathbf{w}(t))\rangle}{\|\mathbf{g}_i(\mathbf{w}(t))\|\|\mathbf{g}(\mathbf{w}(t))\|}\right), \tag{8}$$

where $\langle\cdot\rangle$ is the inner product operation, $\mathbf{g}(\mathbf{w}(t)) = \sum_{i=1}^{|\mathbb{S}|}\frac{D_i}{D}\mathbf{g}_i(\mathbf{w}(t))$ is the global gradient that can be calculated at the central server side. From (8), we can see that when the angle $\theta_i(t)$ is small, it means the local gradient $\mathbf{g}_i(\mathbf{w}(t))$ has a similar direction to the global gradient, thereby positively contributing to the global aggregation. In contrast, when $\theta_i(t)$ is large, e.g., larger than $\pi/2$, the local gradient $\mathbf{g}_i(\mathbf{w}(t))$ has an opposite direction to the global gradient, thereby negatively contributing to the global aggregation.

To restrain the instability caused by randomness presented in instantaneous angle $\theta_i(t)$ at each round, we use so-called *smoothed angle* $\widetilde{\theta}_i(t)$ as a substitution, which is the averaged angle over previous training rounds and is defined as:

$$\widetilde{\theta}_i(t) = \frac{1}{m}\sum_{a=1}^{m}\theta_i(a). \tag{9}$$

By using *smoothed angle* $\widetilde{\theta}_i(t)$, the angle difference across nodes uniquely depends on the data distribution. Intuitively, the angle $\widetilde{\theta}_i(t)$ will be larger as the dissimilarity between data distribution on node i and population distribution grows. Also, the smoothed angle is capable to quantify the degree of data dissimilarity among the local nodes.

We conduct an experiment to illustrate how data distribution can be reflected by angle. Under the same training model in II-B, we randomly assign i) 3 nodes with 1-*class non-IID setting*, ii) 2 nodes with 2-*class non-IID setting*, and iii) the rest of 5 nodes with *IID setting*.

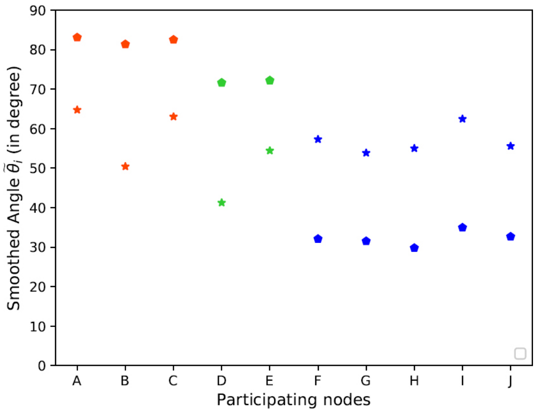

Fig. 2. The smoothed angle $\widetilde{\theta}_i$ of participating node at different training round, where star and pentagon sign denote the angle at commmunication round 1 amd communication round 15, respectively. Nodes with different data distribution are marked with different colors.

As shown in Fig. 2, the smoothed angle between the local gradient and the global gradient is full of randomness at the beginning of FL training. Along with the training, smoothed angle $\widetilde{\theta}_i$ shows diversity across the participating nodes due to the impact of data heterogeneity on local training. The gradient from the node with extremely skewed data (e.g., nodes $\mathrm{A, B, C}$) is nearly orthogonal with the global gradient after 15 communication rounds, which barely brings a contribution to the global model. If we ignore the discrepancy of node contribution and average local update according to the size of datasets, as in `FedAvg`, it slows model convergence.

C. *Federated Adaptive Weighting* (`FedAdp`)

From sections III-A and III-B, we observe that there is an implicit connection between data distribution on node and contribution of that node, which can be measured by using the smoothed angle between the local gradient vector and the global gradient vector. To accelerate model convergence and reduce the expected loss at each round, we aim to assign different weights to different nodes at each round adaptively based on the smoothed angle. Assigning adaptive weight for updating the global model in the proposed `F`ederated `Ad`a`p`tive Weighting (`FedAdp`) algorithm includes two steps:

1) Non-linear mapping function: We design a non-linear mapping function to first quantify the contribution of each node based on angle information. Inspired by sigmoid function, we use a variant of *Gompertz function* [10], which is a non-linear decreasing function defined as

$$f(\widetilde{\theta}_i(t)) = \alpha(1 - e^{-e^{-\alpha(\widetilde{\theta}_i(t)-1)}}), \tag{10}$$

where $\widetilde{\theta}_i(t)$ is the *smoothed angle* in *radian*, e denotes the exponential constant and α is a constant as explained in the following.

The designed mapping function has several properties that are important for the subsequent weight calculation:

- $\lim_{\widetilde{\theta}_i(t)\to\pi/2} f(\widetilde{\theta}_i(t)) = \epsilon$, where $\epsilon \propto \frac{1}{\alpha}$ is constant;
- $\lim_{0\to\widetilde{\theta}_i(t)\to\upsilon} f(\widetilde{\theta}_i(t)) = \alpha$, where $\upsilon \propto \alpha$ is a constant;
- α controls the decreasing rate from α to ϵ as $\widetilde{\theta}_i(t)$ increases from υ to $\pi/2$. As α increases, the gap between small angle and large angle is amplified, as well as the difference of contribution from those nodes.

2) Weighting: After getting the contribution mapped using smoothed angle from each node, we use *Softmax function* to finally calculate the weight of participating nodes for global model aggregation as follows:

$$\widetilde{\psi}_i(t) = \psi_i(t)\frac{e^{f(\widetilde{\theta}_i(t))}}{\sum_{x=1}^{|\mathcal{S}|} e^{f(\widetilde{\theta}_x(t))}} \quad i = 1,\cdots,\mathcal{S}, \tag{11}$$

where $\psi_i = D_i/D$ is the weight used in FedAvg.

The reason for adopting the Softmax function is twofold: i) The output of the Softmax function is a *normalized value* with a larger angle corresponding to a larger weight. ii) By using the Softmax function, the contribution of each node can be reinforced or suppressed, depending on the smoothed angle between its gradient and the global gradient.

The complete procedures of the proposed FedAdp algorithm are presented in Algorithm 1. Compared to FedAvg, FedAdp adopts a simple yet effective strategy that takes angle between local gradient and global gradient into consideration. Consequently, weight for the global model updates can be adaptively assigned based on node contribution rather than evenly averaging, which can accelerate model convergence drastically, as confirmed by our experimental results.

It is worth mentioning that Nguyen *et al.* [8] proposed to delete the nodes whose inner product between the local gradient and the global gradient is negative. This method is equivalent to our idea when $\widetilde{\theta}_i(t) > \pi/2$. However, this method can only identify the node whose gradient is contradicting with the global gradient. Also, simply deleting some nodes may impose variance on FL convergence.

IV. Evaluation and analysis

We implemented FedAdp with PyTorch framework and PySyft library, and studied the image classification using CNN[1] model on two datasets: MNIST, FashionMNIST. We evaluated the accuracy of the trained models using the testing set from each dataset. Similar to the experiment in section II-B, we investigated how FedAdp outperforms FedAvg [1] while different degree of skewness of non-IID dataset is presented.

In the following part, we use the number of communication rounds for the FL model to reach a target testing accuracy as a performance metric. The target accuracy is set to 95% for training on MNIST, and 80% for training on FashionMNIST.

Algorithm 1 Federated Adaptive Weighting (FedAdp)

```
procedure FEDERATED OPTIMIZATION
Input: node set $\mathcal{S}, E, B, T, \eta$,
Server initializes global model $\mathbf{w}(0)$, global update $\Delta(0)$,
Set for keeping smoothed angle $\widetilde{\Theta}(t)$
    for $t = 0,\cdots,T-1$ do
        for node $i \in \mathcal{S}$ in parallel do
            $\Delta_i(t) \leftarrow$ LOCAL UPDATE ( $i, \mathbf{w}_i(t-1)$)
        $\mathbf{w}(t) \leftarrow$ GLOBAL UPDATE
                    ($\Delta_1(t)\ \Delta_2(t),\cdots,\Delta_{|\mathcal{S}|}(t)$ )
procedure LOCAL UPDATE
Input: node index $i$, model $\mathbf{w}_i(t-1)$
Calculate local updates for $\tau = D_i\frac{E}{B}$ times of SGD with
step-size $\eta$ on $F_i(\mathbf{w})$ using (3)
Calculate the model difference $\Delta_i(t) = \mathbf{w}_i(t) - \mathbf{w}(t-1)$
return $\Delta_i(t)$
procedure GLOBAL UPDATE
Input: local update $\Delta_1(t), \Delta_2(t),\cdots,\Delta_{|\mathcal{S}|}(t)$
Calculate node gradient $\mathbf{g}_i$ and the global gradient using
(3) and $\mathbf{g}(\mathbf{w}(t)) = \sum_{i=1}^{|\mathcal{S}|}\frac{D_i}{D}\mathbf{g}_i(\mathbf{w}(t))$, respectively
Calculate instantaneous angle $\theta_i(t)$ by (8)
Get smoothed angle $\widetilde{\theta}_i(t)$ by (9)
Update smoothed angle set $\widetilde{\Theta}(t)$ using $\widetilde{\Theta}(t-1)$ and $\widetilde{\theta}_i(t)$
Calculate weight for model aggregation by (10), (11)
Update global model $\mathbb{E}_{i\in\mathcal{S}}\left[\widetilde{\psi}_i(t)\mathbf{w}_i(t-1)\right]$
return $\mathbf{w}(t)$
```

The number of participating nodes $|\mathcal{S}| = 10$, $D_i = 600$, $B = 32$, $E = 1$, $T = 300$, $\eta = 0.01$, decay rate $= 0.995$, the constant in non-linear mapping function $\alpha = 5$. The skewness of the dataset is measured by *x-class non-IID*. The dataset for nodes is generated in the same way as in section II-B.

We investigate different number of non-IID nodes with different skewness levels of non-IID data to testify the efficiency of FedAdp. For non-IID data, two skewness cases that $x = 1, 2$ are considered. We plot the test accuracy v.s. the communication rounds of federated learning in Fig. 3. From Fig. 3, we can tell FedAdp always outperforms FedAvg when the nodes with non-IID dataset are present. In particular, FedAdp converges very fast in the early training stage since the weight divergence is more obvious in the initial rounds, which makes the effect of assigning adaptive weight for updating the global model even more significant.

Each entry in Table I shows the number of communication rounds necessary to achieve a test accuracy of 95% for CNN on MNIST and 80% for FashionMNIST. The bold number indicates the better result achieved by FedAdp, as compared to FedAvg. FedAdp decreases the number of communication rounds by up to 54.1% and 43.2% for the MNIST task when non-IID nodes are at 1-class and 2-class non-IID setting, respectively. For the FashionMNIST task, the corresponding decreases are up to 43.7% and 45.4%, respectively. In the cases when the target accuracy is not reachable before 300 rounds, FedAdp always terminates with higher testing accuracy.

[1] The CNN has 7 layers with the following structure: $5 \times 5 \times 32$ Convolutional → 2×2 MaxPool → $5 \times 5 \times 64$ Convolutional → 2×2 MaxPool → 1024×512 Fully connected → 512×10 Fully connected → Softmax (1,663,370 total parameters). All Convolutional and Fully connected layers are mapped by ReLu activation. The configuration is similar to [1].

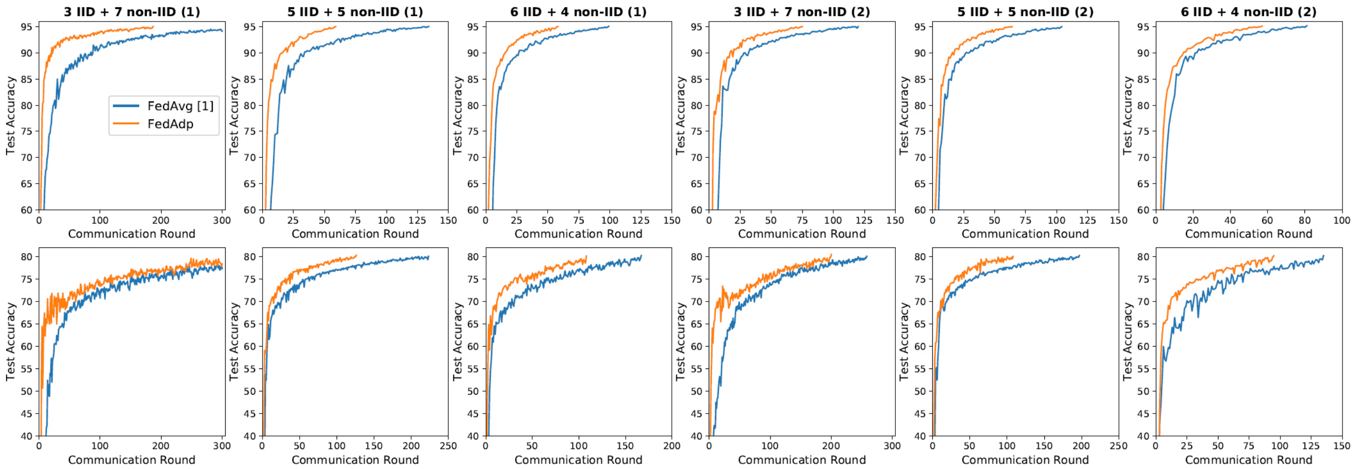

Fig. 3. Test accuracy over communication rounds of `FedAdp` and `FedAvg` with heterogeneous data distribution over participating nodes. Upper and lower subplots correspond to training performance on MNIST and FashionMNIST dataset, respectively.

TABLE I
Number of communication rounds to reach a target accuracy for `FedAdp`, versus `FedAvg` [1], within 300 rounds. N/A refers that algorithms can not reach target accuracy before termination where the highest test accuracy is shown

		MNIST 95% Accuracy	
		1-class non-IID	
	3 IID + 7 non-IID	5 IID + 5 non-IID	6 IID + 4 non-IID
`FedAvg`	N/A (94.48%)	133	99
`FedAdp`	**187**	**61**	**58**
		2-class non-IID	
`FedAvg`	120	104	81
`FedAdp`	**75**	**59**	**52**
		Fashion MNIST 80% Accuracy	
		1-class non-IID	
	3 IID + 7 non-IID	5 IID + 5 non-IID	6 IID + 4 non-IID
`FedAvg`	N/A (77.31%)	222	167
`FedAdp`	N/A (**79.5**%)	**125**	**107**
		2-class non-IID	
`FedAvg`	258	196	134
`FedAdp`	**207**	**107**	**94**

V. Conclusion

In this paper, we have presented our design of `FedAdp` algorithm that assigns nodes with different weights for updating the global model in each round adaptively to reduce the communication rounds of FL training in the presence of non-IID data. We argue that non-IID data exacerbates the model divergence and observe the nodes with non-IID data make smaller (or even negative) contribution to the global model aggregation than the nodes with IID data. We have proposed to measure the node contribution based on the angle between local gradient and global gradient and designed a non-linear mapping function to quantify node contribution. We have designed an adaptive weighting strategy that assigns weight proportional to node contribution instead of according to the size of local datasets. The simple yet effective strategy is able to reinforce positive (suppress negative) node contribution dynamically, leading to a significant communication round reduction. Experimental results have shown that FL training with `FedAdp` has reduced the communication rounds by up to 54.1% on MNIST dataset and up to 45.4% on FashionMNIST dataset, as compared to `FedAvg`.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. the Artificial Intelligence and StatisticsConference (AISTATS)*, 2017.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. the IEEE International Conference on Communications (ICC)*, 2019.

[4] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[6] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. the IEEE Conference on Computer Communications (INFOCOM)*, 2020.

[7] L. Wang, W. Wang, and B. Li, "Cmfl: Mitigating communication overhead for federated learning," in *Proc. the IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 954–964, 2019.

[8] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2021.

[9] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] M. N. Gibbs and D. J. MacKay, "Variational gaussian process classifiers," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1458–1464, 2000.