Approximate Gradient Coding for Privacy-Flexible Federated Learning with Non-IID Data

Okko Makkonen, Sampo Niemelä, Camilla Hollanti, Serge Kas Hanna
Department of Mathematics and Systems Analysis, School of Science, Aalto University, Finland
Emails: {okko.makkonen, sampo.niemela, camilla.hollanti, serge.kashanna}@aalto.fi

Abstract

This work focuses on the challenges of non-IID data and stragglers/dropouts in federated learning. We introduce and explore a privacy-flexible paradigm that models parts of the clients' local data as non-private, offering a more versatile and business-oriented perspective on privacy. Within this framework, we propose a data-driven strategy for mitigating the effects of label heterogeneity and client straggling on federated learning. Our solution combines both offline data sharing and approximate gradient coding techniques. Through numerical simulations using the MNIST dataset, we demonstrate that our approach enables achieving a deliberate trade-off between privacy and utility, leading to improved model convergence and accuracy while using an adaptable portion of non-private data.

I. Introduction

In the ever-evolving landscape of machine learning (ML), one paradigm has emerged as a promising solution to balance the demand for data privacy and the need for robust distributed model training: federated learning (FL) [1], [2]. This innovative approach allows multiple devices or entities, often called *clients*, to collaboratively train a shared model under the orchestration of a central server while keeping their data decentralized. The result is a model that benefits from the collective intelligence of diverse data sources without compromising individual data privacy. Traditional distributed learning [3], [4], on the other hand, involves a central server that owns all the data and has complete control over how it distributes it to edge devices to parallelize the learning process, making it more suitable for scenarios where the data is naturally centralized.

While various assumptions can be made when designing algorithms for distributed learning and analyzing their performance, the most prominent split is between assuming independent, identically distributed (IID) data and non-IID data. The IID data assumption refers to the case where the data samples across different clients are independently drawn from the same underlying distribution, i.e., share similar statistical properties. This assumption simplifies the training process and algorithm design and enables deriving theoretical convergence guarantees such as the ones in [5]–[8]. While this assumption may be valid in traditional distributed learning settings where the data is centralized, it almost never holds in practical FL settings since the clients independently collect their own training data, which typically vary in both size and distribution.

In the context of FL, the term non-IID data refers to a scenario where the data available across clients is not statistically similar. Previous studies have shown that dealing with non-IID data is one of the most significant hurdles encountered by FL. More specifically, training on non-IID data introduces several drawbacks that can hinder the effectiveness of collaborative model training, such as slower convergence and lower model accuracy [9]–[12]. The existing solutions in the literature for dealing with non-IID data in FL can be divided into two main categories that propose orthogonal approaches: *algorithm-based* methods and *data-driven* methods. Algorithm-based methods focus on adjusting the local loss function to align the local model with the global one [11]–[16], training of personalized models for individual participants instead of employing a uniform global model [17]–[22], and developing new aggregation schemes to enhance the model aggregation process [9], [13], [14], [23]. Data-driven approaches predominantly focus on data augmentation methods that mitigate statistical imbalances by artificially expanding the local dataset of each client through generating synthetic data [24]–[29]. Other data-driven approaches include *hybrid* federated learning schemes where a statistically diverse share of the training data is presumed to be present at the central server. In these hybrid schemes, the central server is expected to actively engage in the training process to compensate for potential statistical imbalances arising from client-side distributed computations [10], [30].

In addition to the statistical challenge of dealing with non-IID data, FL also suffers from the straggler problem. The straggler or dropout problem refers to the issue where some participating clients or devices temporarily drop out or fail to complete their participation in a FL round. This can happen because clients may join or leave the FL system dynamically or due to other practical considerations such as poor network connections and resource constraints. Previous studies have demonstrated that the presence of stragglers aggravates the problem of learning on non-IID data, resulting in diminished model quality and slower convergence [31], [32]. Coding for straggler mitigation is a popular solution that has been extensively studied in the traditional distributed learning setting [33]–[42], and also more recently in FL [43], [44]. The key idea in these works is to first introduce training data redundancy in the distributed system and then apply coding techniques that exploit this redundancy to either approximate or recover the central model update, even when some clients drop out.

While previous works in FL treat all the clients' local data as equally private, our work introduces and investigates a *privacy-flexible* FL paradigm that allows for a deliberate trade-off between privacy and utility, offering more adaptable privacy measures. Our motivation for studying this setting stems from practical and commercial situations where striking the right balance between privacy and utility is essential for integrating FL in business models. More precisely, in the proposed paradigm, we model a portion of each client's data as being non-private. We argue that parts of the local data in FL can be treated as non-private for many reasons, including: *(i)* Participants often have diverse datasets of varying sensitivity, and some data may not contain anything private or sensitive, allowing for differentiation in privacy treatment. *(ii)* Privacy can be *selective*, i.e., some participants may be incentivized to sacrifice their privacy by revealing or sharing part of their raw data for enhancing the model's performance and/or for potential financial rewards.

In this work, we introduce a novel data-driven approach that leverages the concept of privacy flexibility to improve the utility of FL while addressing the challenges posed by non-IID data and stragglers. As data can be non-IID in different ways, we focus in this paper on *label heterogeneity*, which refers to the unequal representation of certain classes or labels across clients. Our proposed scheme consists of two key components. First, we introduce an offline *data sharing* mechanism, executed just once before the training phase. In this process, participants share some of their non-private data with each other to reduce the statistical imbalances resulting from label heterogeneity. Moreover, the data sharing also creates redundancy in the training datasets as some of the data becomes available to multiple clients. The second component of our scheme capitalizes on this redundancy through *approximate gradient coding*. This technique optimizes the training process by providing an unbiased estimate of the central model update rule of gradient descent in the presence of dropouts. We present simulation results using the MNIST dataset [45], demonstrating that our scheme enables achieving a convergence behavior and model accuracy that closely mirrors the IID case, contingent on the amount of shared non-private data.

II. Preliminaries

A. Notation

We use bold letters for vectors and sans-serif letters for random variables, e.g., $\boldsymbol{x}$, X, and $\mathbf{X}$ represent a vector, a random variable, and a random vector, respectively. Uppercase italic letters are used for sets. Let $[n] \triangleq \{1,2,\ldots,n\}$ be the set of integers from 1 to n (inclusive), and let $\Delta^n \triangleq \left\{(\delta_0,\delta_1,\ldots,\delta_n) \in \mathbb{R}^{n+1} \middle| \sum_{i=0}^{n} \delta_i = 1 \text{ and } \delta_i \geq 0\right\}$ be the standard n-simplex. We represent the Euclidean norm of a vector $\boldsymbol{x}$ by $\|\boldsymbol{x}\|$, and the scalar product between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

B. System Model

Consider a FL setting with $N > 1$ clients who want to use their local data to collectively train a machine learning model with the help of a central server. Suppose that the clients want to run a supervised classification task, i.e., each client has a set of training examples $(\boldsymbol{x}, y)$ with features $\boldsymbol{x}$ and labels $y \in [L]$, where L represents the total number of classes. Let $\mathcal{D}_i$ be the local dataset of client $i \in [N]$, and let $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$ denote the global training dataset of size M given by the union of the local datasets across all the clients. The goal of FL is to find an optimal global model $\boldsymbol{\beta}^*$ that minimizes a given loss function $\mathcal{L}$, i.e.,

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \mathcal{L}(\mathcal{D}, \boldsymbol{\beta}).$$

The global loss can be often expressed as the sum of the individual losses evaluated at each example $(\boldsymbol{x}, y)$, i.e.,

$$\mathcal{L}(\mathcal{D},\boldsymbol{\beta}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\beta}).$$

A common approach for solving such optimization problems is using gradient-based methods. Namely, the central server applies or approximates the following gradient descent update rule at each iteration $t \in \{0, 1, \ldots\}$ of the algorithm

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \frac{\eta}{M} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \nabla\mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\beta}^{(t)}),$$

where $\nabla\mathcal{L}$ is the gradient of the loss function and η represents the learning rate. In distributed gradient descent, the central server first sends the current model $\boldsymbol{\beta}^{(t)}$ to all clients. Each client i then locally computes the individual gradients of the loss function over the examples in its local dataset $\mathcal{D}_i$ and sends a linear combination of these gradients to the central server. The central server then aggregates these local computations and either recovers the full gradient $\nabla\mathcal{L}(\mathcal{D},\boldsymbol{\beta}^{(t)})$ or obtains an estimate of it. The previous steps are repeated iteratively until convergence. If some of the clients are unresponsive (straggler/dropout) during certain iterations, the central server will not be able to recover the full gradient. In this case, the master's aggregate gives an estimate of the full gradient. In this work, we consider a straggling model where each client is unresponsive with probability p in any given iteration, where this probability is independent across clients and iterations.

Suppose that the global training dataset $\mathcal{D}$ consists of a total of M training examples such that there are K_ℓ examples corresponding to class $\ell \in [L]$. Let us assume that $K_\ell = K$ for all $\ell \in [L]$, such that $M = KL$. Let X_i^ℓ denote the proportion of instances of label $\ell \in [L]$ that are owned by client $i \in [N]$. We focus on a label-heterogeneous setting where the initial proportions $\mathbf{X}^\ell = (\mathsf{X}_1^\ell, \ldots, \mathsf{X}_N^\ell) \in \Delta^{N-1}$ are typically far from $\boldsymbol{\Theta}^\star \triangleq \left(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}\right) \in \Delta^{N-1}$, where $\boldsymbol{\Theta}^\star$ corresponds to the perfectly label-homogeneous setting. We adopt the squared Euclidean distances between the label proportions and $\boldsymbol{\Theta}^\star$ as a measure for label heterogeneity. Some examples for modeling label-heterogeneity are the following:

1) The single-class label-heterogeneous setting, where each client has data belonging only to a single class [10]. For instance, for every $\ell \in [L]$, the initial label proportions could be $\mathbf{X}^\ell = (\mathsf{X}_1^\ell, \mathsf{X}_2^\ell, \ldots, \mathsf{X}_N^\ell) = \mathbf{e}^\ell$, where $\mathbf{e}^\ell$ is the ℓ^{th} standard basis vector of $\mathbb{R}^N$, with $N = L$. In this case, $\mathbf{X}^\ell$ is deterministic, with $\|\mathbf{X}^\ell - \boldsymbol{\Theta}^\star\|^2 = \frac{N-1}{N}$ for all $\ell \in [L]$.
2) The initial label distributions are random and follow a Dirichlet distribution with a small concentration parameter [23]. Namely, for every $\ell \in [L]$, $\mathbf{X}^\ell = (\mathsf{X}_1^\ell, \mathsf{X}_2^\ell, \ldots, \mathsf{X}_N^\ell) \sim \mathrm{Dir}_N(\alpha)$, where $\mathrm{Dir}_N(\alpha)$ is the Dirichlet distribution with N categories and $\alpha > 0$ is the concentration parameter. Smaller values of α (close to zero) correspond to strongly heterogeneous settings; while $\alpha = \infty$ corresponds to a perfectly homogeneous one.

Furthermore, we propose and study a privacy-flexible setting where a proportion $c \in [0,1]$ of each client's data is considered to be non-private. This proportion is assumed to be evenly distributed across all classes, i.e., each client $i \in [N]$ has $\mathsf{C}_i^\ell \triangleq c\mathsf{X}_i^\ell K$ non-private training examples from class $\ell \in [L]$[1].

III. Proposed Scheme

To mitigate the effects of label heterogeneity and client straggling, we propose using an offline data sharing scheme that generates redundancy across clients by replicating non-private data. We will focus on the proportions of some fixed label $\ell \in [L]$ and drop the superscript ℓ. Recall that $\mathbf{X} = (\mathsf{X}_1, \ldots, \mathsf{X}_N)$ denotes the initial label proportions prior to any data sharing, and let $\mathbf{Y} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_N)$ denote the corresponding final label proportions after data sharing. The replication-based data sharing scheme is described by $\mathbf{S} = (\mathsf{S}_1, \ldots, \mathsf{S}_N) \in \mathbb{Z}^N$, where $\mathsf{S}_i \geq 0$ denotes the number of non-private training examples that client $i \in [N]$ receives from other clients[2]. Then,

$$\mathbf{Y} = \frac{\mathbf{X}K + \mathbf{S}}{K + B}, \tag{1}$$

where $B = \sum_{i=1}^N \mathsf{S}_i$ denotes the total amount of shared data.

The goal of data sharing is twofold: *(i)* "Break" label heterogeneity by generating final label proportions $\mathbf{Y}$ that are as close as possible to $\boldsymbol{\Theta}^\star$ in order to reduce the effects of "non-IIDness". *(ii)* Create data redundancy across clients which we will use to ensure resilience against straggling clients.

As previously mentioned, we use the squared Euclidean distances between the label proportions and $\boldsymbol{\Theta}^\star$ as a measure for label heterogeneity. Prior to any data sharing the squared distance is $\|\mathbf{X} - \boldsymbol{\Theta}^\star\|^2$, and after data sharing, the squared distance becomes $\|\mathbf{Y} - \boldsymbol{\Theta}^\star\|^2$, where $\mathbf{Y}$ is given by (1). We aim to devise a data sharing scheme, i.e., determine $\mathbf{S}$, that minimizes $\|\mathbf{Y} - \boldsymbol{\Theta}^\star\|^2$ for a given realization $\boldsymbol{X}$ of $\mathbf{X}$. If the label counts of each client's local data $\mathcal{D}_i$ (private and non-private data) are assumed to be public, one can obtain an optimal deterministic data sharing scheme that minimizes $\|\mathbf{Y} - \boldsymbol{\Theta}^\star\|^2$ for any realization $\boldsymbol{X}$ of $\mathbf{X}$ by solving a constrained optimization problem. However, knowing the label counts of each client's data can potentially leak some information about the private data of the clients. Therefore, we propose a randomized data sharing scheme in Section III-A that does not require the knowledge of the label counts. We provide theoretical guarantees on the performance of the randomized data sharing scheme by evaluating $\mathbb{E}\left[\|\mathbf{Y} - \boldsymbol{\Theta}^\star\|^2 \mid \mathbf{X} = \boldsymbol{X}\right]$ in terms of the initial squared distance $\|\boldsymbol{X} - \boldsymbol{\Theta}^\star\|^2$, where the expectation is taken over the randomness of the data sharing process. Furthermore, in addition to minimizing the effects of label heterogeneity, an equally important consequence of the proposed data sharing scheme is generating redundancy across clients. In Section III-B, we present an approximate gradient coding scheme that leverages this redundancy to provide resilience against straggling clients.

A. Randomized Data Sharing

To reduce the effects of label heterogeneity without the knowledge of the label counts of each client's local data, we propose a randomized offline data sharing scheme where each client shares its non-private data with $d \in \{0, 1, \ldots, N-1\}$ other clients chosen uniformly at random. Let $\widetilde{\mathsf{C}}_i = \sum_{i' \neq i} \mathsf{C}_{i'} = c(1 - \mathsf{X}_i)K$ denote the total number of non-private examples in $\mathcal{D}$ that are not owned by client $i \in [N]$. The randomized data sharing scheme places each of the $\widetilde{\mathsf{C}}_i$ non-private examples independently with probability $\frac{d}{N-1}$ at client $i \in [N]$. Thus, client $i \in [N]$ receives S_i new examples, where $\mathsf{S}_i \sim \mathrm{Binomial}(\widetilde{\mathsf{C}}_i, \frac{d}{N-1})$ is a random variable that follows a binomial distribution with parameters $\widetilde{\mathsf{C}}_i = c(1 - \mathsf{X}_i)K$ and $\frac{d}{N-1}$. Hence, the new label proportions after the randomized data sharing follow from (1), with $B = dcK$.

The next theorem expresses the expected value of $\|\mathbf{Y} - \boldsymbol{\Theta}^\star\|^2$ in terms of the replication factor $d \in \{0, 1, \ldots, N-1\}$, privacy parameter $c \in [0,1]$, number of clients N, and the initial distance $\|\boldsymbol{X} - \boldsymbol{\Theta}^\star\|^2$ for any realization $\boldsymbol{X} \in \Delta^{N-1}$ of $\mathbf{X}$.

[1] To simplify notation, we assume that $c\mathsf{X}_i^\ell K$ are integers for all i and ℓ.

[2] Similar to [44], client-to-client offline communication can be assumed to be routed through the central server to ensure network connectivity.

Theorem 1. *For any realization $\boldsymbol{X} \in \Delta^{N-1}$ of $\mathbf{X}$, the randomized data sharing generates label proportions satisfying*

$$\mathbb{E}\left[\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2 \mid \mathbf{X} = \boldsymbol{X}\right] = \frac{dc(N-1-d)}{(1+dc)^2(N-1)K} + \frac{(N-1-dc)^2}{(1+dc)^2(N-1)^2}\|\boldsymbol{X} - \mathbf{\Theta}^{\star}\|^2,$$

where the expectation is taken over the randomness of the data sharing scheme. Furthermore, for $K \gg 1$, we have

$$\mathbb{E}\left[\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2 \mid \mathbf{X} = \boldsymbol{X}\right] \approx \frac{(N-1-dc)^2}{(1+dc)^2(N-1)^2}\|\boldsymbol{X} - \mathbf{\Theta}^{\star}\|^2.$$

Proof. We have

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2 \mid \mathbf{X} = \boldsymbol{X}\right] &= \mathbb{E}\left[\sum_{i=1}^{N}\left(\mathsf{Y}_i - \frac{1}{N}\right)^2 \;\middle|\; \mathsf{X}_i = X_i\right] &(2)\\
&= \sum_{i=1}^{N}\mathbb{E}\left[\mathsf{Y}_i^2 | \mathsf{X}_{\mathsf{i}} = X_i\right] - \frac{2}{N}\mathbb{E}\left[\mathsf{Y}_i | \mathsf{X}_{\mathsf{i}} = X_i\right] + \frac{1}{N^2}. &(3)
\end{aligned}$$

Since $B = dcK$ for the randomized data sharing scheme, then it follows from (1) that

$$\mathbb{E}\left[\mathsf{Y}_i \mid \mathsf{X}_{\mathsf{i}} = X_i\right] = \frac{1}{1+dc}\left(X_i + \frac{\mathbb{E}[\mathsf{S}_i]}{K}\right) = \frac{1}{1+dc}\left(X_i + \frac{dc}{N-1}(1-X_i)\right). \tag{4}$$

Furthermore,

$$\begin{aligned}
\mathbb{E}\left[\mathsf{Y}_i^2 \mid \mathsf{X}_{\mathsf{i}} = X_i\right] &= \frac{1}{(1+dc)^2}\left(X_i^2 + 2\frac{\mathbb{E}[\mathsf{S}_i]}{K}X_i + \frac{\mathbb{E}[\mathsf{S}_i^2]}{K^2}\right) &(5)\\
&= \frac{1}{(1+dc)^2}\left(X_i^2 + 2\frac{dc}{N-1}(1-X_i)X_i + \frac{dc}{(N-1)K}\left(1 - \frac{d}{N-1}\right)(1-X_i) + \frac{d^2c^2}{(N-1)^2}(1-X_i)^2\right). &(6)
\end{aligned}$$

By substituting (4) and (6) in (3) we get

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2 \mid \mathbf{X} = \boldsymbol{X}\right] = &\frac{dc(N-1-d)}{(1+dc)^2(N-1)^2K}\sum_{i=1}^{N} 1 - X_i\\
&+ \frac{1}{(1+dc)^2}\underbrace{\left(1 - 2\frac{dc}{N-1} + \frac{d^2c^2}{(N-1)^2}\right)}_{T_1}\sum_{i=1}^{N}X_i^2\\
&- \frac{2}{N}\frac{1}{(1+dc)^2}\underbrace{\left(1 + dc + \frac{d^2c^2N}{(N-1)^2} - \frac{dcN}{N-1} - \frac{dc(1+dc)}{N-1}\right)}_{T_2}\sum_{i=1}^{N}X_i\\
&+ \frac{1}{N^2}\frac{1}{(1+dc)^2}\underbrace{\left((1+dc)^2 - 2\frac{(1+dc)dcN}{N-1} + \frac{d^2c^2N^2}{(N-1)^2}\right)}_{T_3}. \qquad (7)
\end{aligned}$$

One can easily verify that

$$T_1 = T_2 = T_3 = \frac{(N-1-dc)^2}{(N-1)^2}. \tag{8}$$

Therefore,

$$\mathbb{E}\left[\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2 \mid \mathbf{X} = \boldsymbol{X}\right] = \frac{dc(N-1-d)}{(1+dc)^2(N-1)^2K}\sum_{i=1}^{N} 1 - X_i + \frac{(N-1-dc)^2}{(1+dc)^2(N-1)^2}\sum_{i=1}^{N}\left(X_i - \frac{1}{N}\right)^2. \tag{9}$$

Since $\boldsymbol{X} \in \Delta^{N-1}$, we have $\sum_{i=1}^{N} 1 - X_i = N - 1$. The proof is concluded by substituting the latter equality in the first term in (9), and by expressing the summation in the second term as $\|\boldsymbol{X} - \mathbf{\Theta}^{*}\|^2$. □

The significance of this theorem is that for any given set of input label proportions $\boldsymbol{X}$, the expected value of the output distance $\|\mathbf{Y} - \mathbf{\Theta}^{\star}\|^2$ is reduced by at least a factor of $(1+dc)^2$ with respect to the initial distance $\|\boldsymbol{X} - \mathbf{\Theta}^{\star}\|^2$ for large enough L. This shows that the randomized data sharing scheme can effectively reduce heterogeneity without knowing the label counts of each client's local data.

B. Approximate Gradient Coding

As previously mentioned, at each iteration $t \in \{0, 1, \ldots\}$, the central server sends the current model $\boldsymbol{\beta}^{(t)}$ to all N clients. Each client participates in any given iteration independently with probability $1-p$, i.e., the client is a straggler with probability p. Let $\mathcal{S}_i \subseteq [M]$, $i \in [N]$, be the set of indices of the training examples $(\boldsymbol{x}, y) \in \mathcal{D}_i$ that are owned by client i. The clients participating in a given iteration t send a linear combination of the partial gradients computed over their local data which is given by

$$f_i(\boldsymbol{\beta}^{(t)}) = \sum_{j \in \mathcal{S}_i} W_j \mathbf{g}_j^{(t)}, \tag{10}$$

where $\mathbf{g}_j^{(t)} \triangleq \nabla \mathcal{L}(\boldsymbol{x}_j, y_j, \boldsymbol{\beta}^{(t)})$ denotes the partial gradient evaluated at example $(\boldsymbol{x}_j, y_j)$, and W_j is the weighting factor of example $(\boldsymbol{x}_j, y_j)$. The weighting factor of $(\boldsymbol{x}_j, y_j)$ is $W_j = \frac{1}{(1-p)d_j}$, where $d_j \in \mathbb{Z}^+$ is the total number of times the example $(\boldsymbol{x}_j, y_j)$ is replicated across all N clients. Note that $d_j = 1$ for all private data, and $d_j = d+1$ for all non-private data, where d is the replication factor of the randomized data sharing scheme. The central server then aggregates the local computations of the clients to obtain an estimate of the full gradient given by

$$\hat{\mathbf{g}}^{(t)} = \sum_{i=1}^{N} \mathsf{I}_i^{(t)} f_i(\boldsymbol{\beta}^{(t)}), \tag{11}$$

where $\mathsf{I}_i^{(t)} = 1$ if client i is participating in iteration t, and $\mathsf{I}_i^{(t)} = 0$ otherwise. One can easily show that $\hat{\mathbf{g}}^{(t)}$ is an unbiased estimator of the full gradient $\mathbf{g}^{(t)} = \sum_{j=1}^{M} \mathbf{g}_j^{(t)}$, i.e., $\mathbb{E}[\hat{\mathbf{g}}^{(t)}] = \mathbf{g}^{(t)}$, where the expectation is taken over the randomness of the straggling process. The variance of $\hat{\mathbf{g}}^{(t)}$ depends on the data sharing scheme being used and has a direct effect on the rate of convergence of the algorithm. More specifically, it has been shown in [46] that for an unbiased estimator, the value of $\mathbb{E}[\|\hat{\mathbf{g}}^{(t)}\|^2]$ is inversely proportional to the rate of convergence under certain assumptions on the loss function. An important trait of our proposed scheme is that it reduces the variance of the estimator and thus speeds up the convergence of the algorithm. We highlight this phenomenon through numerical simulations in the next section.

IV. Simulation Results

A. Setup

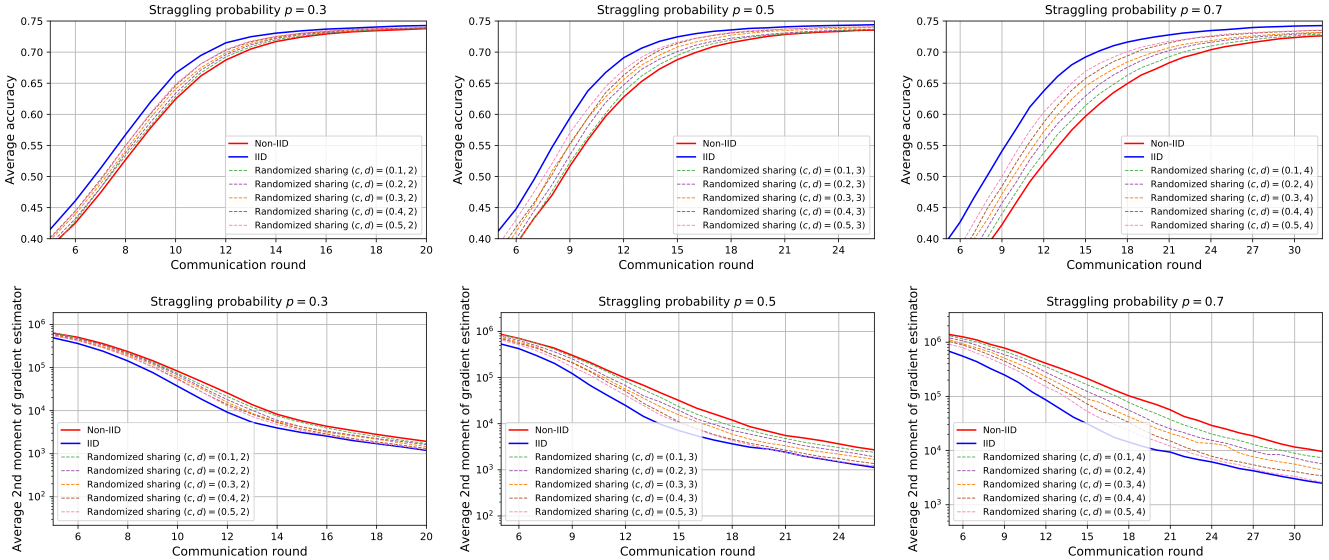
We simulate a federated learning setup with $N = 10$ clients. The goal is to train a multinomial logistic regression model on the MNIST dataset [45], which is a supervised image classification task consisting of $L = 10$ different classes. We use a global training dataset $\mathcal{D}$ of size $M = 300$, consisting of $K_\ell = K = 30$ images from each class $\ell \in [L]$ drawn uniformly at random from the 60,000 training images in MNIST. To model IID/non-IID settings, we consider multiple ways for partitioning the $M = 300$ training examples over the $N = 10$ clients: *(i)* IID setting, where the data is randomly shuffled, and then partitioned into 10 clients each receiving 30 examples. *(ii)* Two non-IID settings that follow from the single-class label-heterogeneous setting and the random Dirichlet distribution with $\alpha = 0.1$, as described in Section II.

We apply the randomized data sharing and approximate gradient coding to the two non-IID settings, and compare the performance of the trained model to the IID and non-IID cases with no data sharing. For each scenario, we run a total of 1000 simulations for 50 communication rounds (iterations) of the federated learning process and compute the average test accuracy and the second moment of the gradient estimator (defined in (11)) as a function of the communication round. In each round, the identities of the stragglers/dropouts are determined according to the straggling model parameterized by p, as explained in Section II. We train the model using the SGD (stochastic gradient descent) optimizer with a learning rate of $\eta = 0.1$ and decay $\gamma = 0.97$. The source code for these simulations can be found in [47].

B. Results

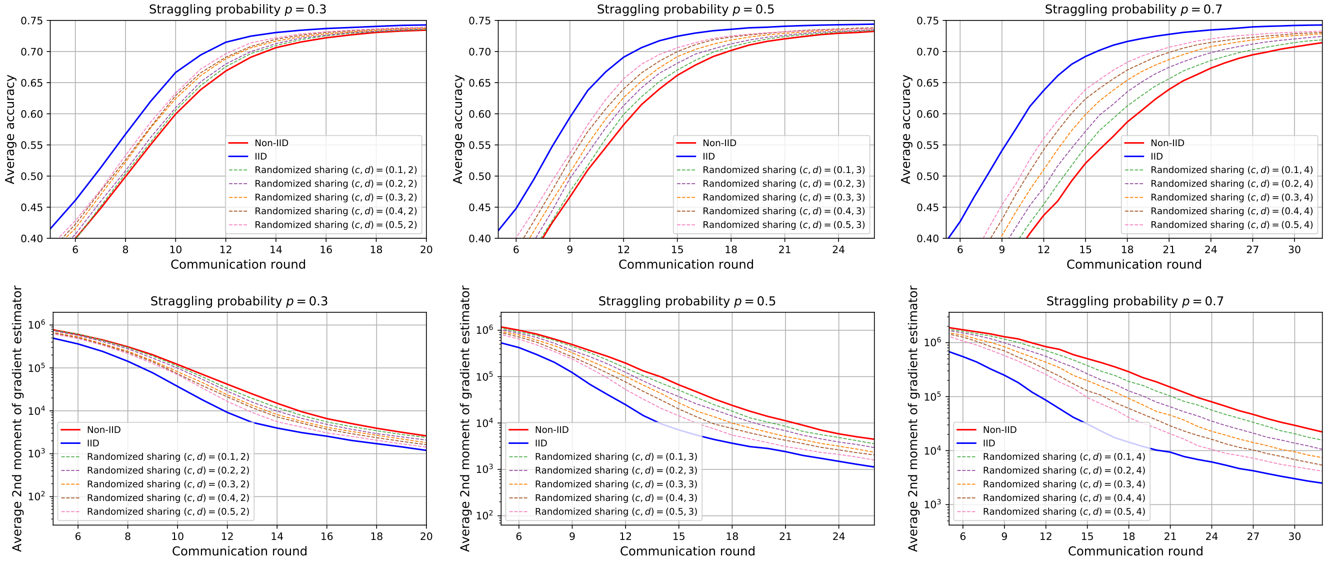
The average testing accuracy over 1000 simulations is plotted in Figures 1 and 2 for different data sharing parameters compared to the baseline non-IID (no data sharing) and IID settings. These simulations show that there is a clear increase in the convergence rate as the parameter c is increased from 0 to 0.5. We also observe a slight difference in the final model accuracy achieved. We expect this difference to be more significant for more challenging datasets and more complex models. Using our proposed scheme and by manipulating the parameters c and d, we may interpolate between the non-IID and the IID settings. Furthermore, the plots of the average second moment of the gradient estimator confirm our intuition that the increase in convergence rate is correlated with the decrease in the variance of the gradient estimator caused by our approximate gradient coding scheme. By comparing Figures 1 and 2, we see that the single-class heterogeneous setting leads to worse model convergence and accuracy than the Dirichlet setting with $\alpha = 0.1$. This suggests that given the single-class heterogeneous setting, one would need more data sharing to achieve the same performance as the Dirichlet case.

In conclusion, our simulation results demonstrate that under the privacy-flexible paradigm that we introduce, combining data sharing with gradient coding allows us to deliberately achieve a trade-off between privacy (characterized by the parameter c)



(a) Dirichlet setting $\alpha = 0.1$, $p = 0.3$ (b) Dirichlet setting $\alpha = 0.1$, $p = 0.5$ (c) Dirichlet setting $\alpha = 0.1$, $p = 0.7$

Fig. 1: Average testing accuracy and second moment of the gradient estimator (defined in (11)) as a function of the communication round (iteration) in the Dirichlet setting with $\alpha = 0.1$.



(a) Single-class setting, $p = 0.3$ (b) Single-class setting, $p = 0.5$ (c) Single-class setting, $p = 0.7$

Fig. 2: Average testing accuracy and second moment of the gradient estimator (defined in (11)) as a function of the communication round (iteration) in the single-class setting.

and utility (characterized by model convergence and accuracy). In future work, we intend to analyze this trade-off for different models and datasets, and also other algorithms such as federated averaging, where multiple local model updates are performed in a single communication round.

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 *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] D. Jakovetic, "Distributed optimization: Algorithms and convergence rates," *PhD Thesis, Carnegie Mellon University, Pittsburgh PA, USA*, 2013.

[4] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*, 2014, pp. 583–598.

[5] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," *arXiv preprint arXiv:1708.01012*, 2017.

[6] S. U. Stich, "Local SGD converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[7] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 9709–9758, 2021.

[8] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro, "Graph oracle models, lower bounds, and gaps for parallel stochastic optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[9] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[10] 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.

[11] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[12] Y. Shi, J. Liang, W. Zhang, V. Y. Tan, and S. Bai, "Towards understanding and mitigating dimensional collapse in heterogeneous federated learning," *arXiv preprint arXiv:2210.00226*, 2022.

[13] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[14] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[15] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 713–10 722.

[16] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.

[17] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[18] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[19] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[20] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[21] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[22] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[23] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 7252–7261.

[24] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[25] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[26] J. Goetz and A. Tewari, "Federated learning via synthetic data," *arXiv preprint arXiv:2008.04489*, 2020.

[27] M. Rasouli, T. Sun, and R. Rajagopal, "Fedgan: Federated generative adversarial networks for distributed data," *arXiv preprint arXiv:2006.07228*, 2020.

[28] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke, "Towards fair federated learning with zero-shot data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3310–3319.

[29] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "Fedmix: Approximation of mixup under mean augmented federated learning," *arXiv preprint arXiv:2107.00233*, 2021.

[30] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-fl for wireless networks: Cooperative learning mechanism using non-IID data," in *ICC 2020-2020 IEEE International Conference On Communications (ICC)*. IEEE, 2020, pp. 1–7.

[31] Z. Charles and J. Konečnỳ, "On the outsized importance of learning rates in local update methods," *arXiv preprint arXiv:2007.00878*, 2020.

[32] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 606–14 619, 2021.

[33] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*, 2017, pp. 3368–3376.

[34] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5610–5619.

[35] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[36] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 5434–5442.

[37] E. Ozfatura, S. Ulukus, and D. Gündüz, "Straggler-aware distributed learning: Communication–computation latency trade-off," *Entropy*, vol. 22, no. 5, 2020.

[38] Z. Charles, D. Papailiopoulos, and J. Ellenberg, "Approximate gradient coding via sparse random graphs," *arXiv preprint arXiv:1711.06771*, 2017.

[39] R. Bitar, M. Wootters, and S. El Rouayheb, "Stochastic gradient coding for straggler mitigation in distributed learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 277–291, 2020.

[40] H. Wang, Z. Charles, and D. Papailiopoulos, "Erasurehead: Distributed gradient descent without delays using approximate gradient coding," *arXiv preprint arXiv:1901.09671*, 2019.

[41] S. Wang, J. Liu, and N. Shroff, "Fundamental limits of approximate gradient coding," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 3, pp. 1–22, 2019.

[42] M. Glasgow and M. Wootters, "Approximate gradient coding with optimal decoding," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 855–866, 2021.

[43] S. Prakash, S. Dhakal, M. R. Akdeniz, Y. Yona, S. Talwar, S. Avestimehr, and N. Himayat, "Coded computing for low-latency federated learning over wireless edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 233–250, 2020.

[44] R. Schlegel, S. Kumar, E. Rosnes, and A. G. i Amat, "CodedPaddedFL and CodedSecAgg: straggler mitigation and secure aggregation in federated learning," *IEEE Transactions on Communications*, 2023.

[45] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[46] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," *arXiv preprint arXiv:1109.5647*, 2011.

[47] O. Makkonen, S. Niemelä, C. Hollanti, and S. Kas Hanna, "Approximate gradient coding for privacy-flexible federated learning with non-iid data (extended version)," 2023, https://github.com/okkomakkonen/label-heterogeneity.