2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)

CFedAvg: Achieving Efficient Communication and Fast Convergence in Non-IID Federated Learning

Haibo Yang
Department of ECE
The Ohio State University
Columbus, OH, USA
yang.5952@osu.edu

Jia Liu
Department of ECE
The Ohio State University
Columbus, OH, USA
liu@ece.osu.edu

Elizabeth S. Bentley
Information Directorate
Air Force Research Laboratory
Rome, NY, USA
elizabeth.bentley.3@us.af.mil

***Abstract*—Federated learning (FL) is a prevailing distributed learning paradigm, where a large number of workers jointly learn a model without sharing their training data. However, high communication costs could arise in FL due to large-scale (deep) learning models and bandwidth-constrained connections. In this paper, we introduce a communication-efficient algorithmic framework called CFedAvg for FL with non-i.i.d. datasets, which works with general (biased or unbiased) SNR-constrained compressors. We analyze the convergence rate of CFedAvg for non-convex functions with constant and decaying learning rates. The CFedAvg algorithm can achieve an $\mathcal{O}(1/\sqrt{mKT} + 1/T)$ convergence rate with a constant learning rate, implying a linear speedup for convergence as the number of workers increases, where K is the number of local steps, T is the number of total communication rounds, and m is the total worker number. This matches the convergence rate of distributed/federated learning without compression, thus achieving high communication efficiency while not sacrificing learning accuracy in FL. Furthermore, we extend CFedAvg to cases with heterogeneous local steps, which allows different workers to perform a different number of local steps to better adapt to their own circumstances. The interesting observation in general is that the noise/variance introduced by compressors does not affect the overall convergence rate order for non-i.i.d. FL. We verify the effectiveness of our CFedAvg algorithm on three datasets with two gradient compression schemes of different compression ratios.**

***Index Terms*—Distributed/federated learning, communication efficient, convergence analysis, Non-IID Data**

I. Introduction

In recent years, advances in machine learning (ML) have sparked many new and emerging applications that transform our society. Traditionally, the training of ML applications often relies on cloud-based large data-centers to collect and process a vast amount of data. However, with the rise of Internet of Things (IoT), data and demands for ML are increasingly being generated from mobile devices in wireless edge networks. Due to high latency, low bandwidth, and privacy/security concerns, aggregating all data to the cloud for ML training may no longer be desirable or may even be infeasible. In these circumstances, Federated Learning (FL) has emerged as a prevailing ML paradigm, thanks to the rapidly growing computation capability of modern mobile devices. Generally speaking, FL is a network-based ML architecture, under which a large number of local devices (often referred to as workers) collaboratively train a model based on their local datasets and coordinated by a central server. In FL, the server is responsible for aggregating and updating model parameters without requiring the knowledge of the data located at each worker. Workers process their computational tasks independently with decentralized data, and communicate to the server to update the global learning model. By doing so, not only can FL significantly alleviate the risk of exposing data privacy, it also fully utilizes idle computation resources at workers. This constitutes a win-win situation that have led to a variety of successful applications [1].

Despite the aforementioned advantages of FL, a number of technical challenges also arise due to the unique characteristics of FL. One key challenge in FL is the high communication cost due to the every-increasing sizes of learning models and datasets [1]. The problem of a high communication load in FL is further exacerbated by the fact that, in many wireless edge networks, the communication links are often bandwidth-constrained and their link capacities are highly dynamic due to stochastic channel fading effects. As a result, information exchanges between the server and workers could be very inefficient, rendering a major bottleneck in FL. Generally speaking, the total communication cost during the training process is determined by two factors: the number of communication rounds and the size (or amount) of the update parameters in each communication round. There are some algorithms in FL (e.g., FedAvg [2]) that take more local steps at each node and communicate infrequently with the server, thus decreasing the total number of communication rounds and in turn reducing the total communication cost. However, this does not completely solve the problem since the communicated gradient vectors could still be high-dimensional. On the other hand, there exist a variety of gradient compression techniques (e.g., signSGD [3], gradient dropping [4], TernGrad [5], etc.) that were originally proposed for centralized/distributed learning and shown to be effective in reducing the size of exchanged parameters in each communication round.

Given the above encouraging results of information compression in distributed/decentralized learning, an interesting question naturally arises: *Could we combine compression with*

This work has been supported in part by NSF grants CAREER CNS-2110259, CNS-2102233, CCF-2110252, ECCS-2140277, and a Google Faculty Research Award. Distribution A. Approved for public release: Distribution unlimited 88ABW-2020-3364 on 29 Oct 2020.

ISBN 978-3-903176-37-9 © 2021 IFIP

infrequent communication to further reduce the communication cost of FL? However, answering this question turns out to be highly non-trivial. One key challenge stems from the *heterogeneity* of the local datasets among different workers. In the traditional distributed learning literature, the dataset at each node is usually well-shuffled and hence can be assumed to be independent and identically distributed (i.i.d.). However, the dataset at each worker in FL could be generated based on the local environment and cannot be shuffled with other workers due to privacy protection. Thus, the i.i.d. assumption often fails to hold. In some circumstances, the dataset distributions at different workers could vary dramatically due to factors such as geographic location differences, time window gaps, among many others. Upon integrating compression with FL, the already-complicated non-i.i.d. dataset problem is further worsened by the significant loss of information due to the use of compression operators. In addition, under infrequent communication, the multiple local steps in each worker introduce further "model drift" to the non-i.i.d. datasets. It has been shown that this model drift effect results in extra variances that may lead to deterioration or even failures of training. Due to these complex randomness couplings between compression, local steps, and non-i.i.d. datasets, results on non-i.i.d. compressed FL remain limited. This motivates us to fill this gap and rigorously investigate the algorithmic design that integrates compression in non-i.i.d. FL.

Moreover, workers in FL system vary tremendously in terms of computation capabilities and resources limits (e.g., memory, battery capacity). Hence, using a predefined constant number of local steps for all workers (assumed in most existing work in FL) may not be a good design strategy, which may result in the faster workers idling and slower workers causing straggler problems. As a result, another important question in FL emerges: *Could we use heterogeneous local steps for workers to further improve flexibility and efficiency in FL?*

In this paper, we answer the above open questions by proposing a communication-efficient algorithm called *CFedAvg* (compressed FedAvg) with error-feedback. Our CFedAvg algorithm reduces both the communication rounds and link capacity requirement in each communication round. It also allows the use of heterogeneous local steps (i.e., different workers perform different local steps to better adapt to their own computing environments). Our main contributions and results are summarized as follows:

- We show that, under general signal-to-noise-ratio (SNR) constrained compressors, the convergence rate of our CFedAvg algorithm is $\mathcal{O}(1/\sqrt{mKT}+1/T)$ and $\tilde{\mathcal{O}}(1/\sqrt{mKT})+\mathcal{O}(1/\sqrt{T})$ with constant and decaying learning rates, respectively, for general non-convex functions and non-i.i.d. datasets in FL, where K is the number of local steps, T is the number of total communication rounds, and m is the total number of workers. For a sufficiently large T, this implies that CFedAvg achieves an $\mathcal{O}(1/\sqrt{mKT})$ convergence rate with a constant learning rate and enjoys the linear speedup effect as the number of workers increases. (To attain an ϵ accuracy for an algorithm, it takes $\mathcal{O}(1/\epsilon^2)$ steps with a convergence rate $\mathcal{O}(1/\sqrt{T})$, while it takes $\mathcal{O}(1/m\epsilon^2)$ steps if with $\mathcal{O}(1/\sqrt{mT})$ rate. In this sense, it is a *linear speedup* with respect to m.) Note that this matches the convergence rate order of uncompressed distributed/federated learning algorithm [6], [7], [8], thus achieving high communication efficiency while not sacrificing learning accuracy in FL.
- We extend CFedAvg to heterogeneous local steps among workers, which allows each worker performs different local steps based on its own computation capability and other conditions. We show that FL can still offer theoretical performance guarantee without requiring the same predefined constant number of local steps among all workers.
- We show that the use of SNR-constrained compressors in CFedAvg only slightly increases the local variance constant and does not affect overall convergence rate order for non-i.i.d. FL with infrequent communication. Moreover, we show the convergence results of CFedAvg hold for either unbiased or biased SNR-constrained compressors, which is far more flexible than previous works that require unbiased compressors.
- We verify the effectiveness of CFedAvg on MNIST, FMNIST and CIFAR-10 datasets with different SNR-constrained compressors. We find that CFedAvg can reduce up to 99% of information exchange with minimal impacts on learning accuracy, which confirms the communication-efficiency advantages of training large learning models in non-i.i.d. FL with information compression.

II. Related work

For distributed/federated learning in communication-constrained environments, a variety of communication-efficient algorithms have been proposed. We categorize the existing work into two classes: one is to use infrequent communication to reduce the communication rounds and the other is to compress the transmission from each worker to the parameter server in each communication round.

Infrequent-Communication Approaches: One notable algorithm of FL is the Federated Averaging (FedAvg) algorithm, which was first proposed by McMahan *et al.* [2] as a heuristic to improve both communication efficiency and data privacy. In FedAvg, every worker performs multiple SGD steps independently to update the model locally before communicating with the parameter server, which is different from traditional distributed learning with only one local step. Since then, this work has sparked many follow-ups that focus on FL with i.i.d./non-i.i.d. datasets [1]. These studies demonstrated the effectiveness of FedAvg and its variants on reducing communication cost. Also, researchers have theoretically shown that FedAvg and its variants can achieve the same convergence rate order as the traditional distributed learning (see, e.g., [7], [9], [8]).

Compression-Based Approaches: Although FedAvg and its variants save communication costs by utilizing multiple local steps to reduce the total number of communication rounds, it has to transmit all model parameters in each communication round at every worker. Thus, it could still induce high latency and communication overhead in networks with low connection speeds or large channel variations. To address this challenge,

ISBN 978-3-903176-37-9 © 2021 IFIP

a natural idea is to compress the parameters to reduce the amount of transmitted data from each worker to the parameter server. Compression-based approaches have attracted increasing attention in recent years in distributed and decentralized learning [10], [11], which have enabled the training of large-size models over networks with low-speed connections. Broadly speaking, compression-based approaches can be classified into the following two main categories: quantization and sparsification. The basic idea of quantization is to project a vector from a high-dimensional space to a low-dimensional subspace, so that the projected vector can be represented by a fewer number of bits, e.g., signSGD [3] and QSGD [12]. Given a high-dimensional vector, the basic idea of sparsification is to select only a part of its components to transmit based on a threshold [13], [4], [14], [15].

In fact, these two approaches are closely related, and there are works that combine them to achieve better compression results [16], [12], [5]. Qsparse-local-SGD proposed by Basu *et al.* [17] is the most related work to this paper, which combines quantization, sparsification and local steps to achieve communication-efficiency. However, our algorithm has a better convergence rate with more relaxed assumptions. We also propose an algorithm design that allows more flexible heterogeneous local steps. Please see Section IV-B for details.

So far, however, it remains unknown whether the same convergence rate (with linear speedup) could be achieved in FL with compression, particularly under the asynchrony due to non-i.i.d. dataset and heterogeneous local steps at each worker. Answering this question constitutes the rest of this paper.

III. System Model, Problem Formulation, and Preliminaries

A. System Model and Problem Formulation

Consider an FL system with m workers who collaboratively learn a model with decentralized data and under the coordination of a central parameter server. The goal of the FL system is to solve the following optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x}), \tag{1}$$

where $F_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\sim D_i}[F_i(\mathbf{x},\xi_i)]$ denotes the local (non-convex) loss function, which evaluates the average discrepancy between the learning model's output and the ground truth corresponding to a random training sample ξ_i that follows a local data distribution D_i. In (1), the parameter d represents the dimensionality of the training model. For the i.i.d. setting, each local dataset is assumed to sample from some common latent distribution, i.e., $D_i = D, \forall i \in [m]$. In practice, however, the local dataset at each worker in FL could be generated based on its local environment and thus being non-i.i.d., i.e., $D_i \neq D_j$ if $i \neq j$. Note that the i.i.d. setting can be viewed as a special case of the non-i.i.d. setting. Hence, our results for the non-i.i.d. setting are directly applicable to the i.i.d. setting.

B. General SNR-Constrained Compressors

To facilitate the discussions of our CFedAvg algorithm, we will first formally define the notion of *general SNR (signal-to-noise ratio)-constrained compressors*[18], [19]:

Definition 1. *(General SNR-Constrained Compressor) An operator $\mathcal{C}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is said to be constrained by an SNR threshold $\gamma \geq 1$ if it satisfies:*

$$\mathbb{E}_{\mathcal{C}}\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|^2 \leq (1/\gamma)\|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

It is clear from Definition 1 that, for a given compressor, γ is its lowest SNR guarantee yielded by its largest compression noise power $\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|^2$. The γ-threshold can be viewed as a proxy of compression rate of the compressor. For $\gamma = \infty$, we have $\mathcal{C}(\mathbf{x}) = \mathbf{x}$, which means no compression and zero information loss. On the other hand, $\gamma \to 1$ implies that the compression rate is arbitrarily high and the output contains no information of $\mathbf{x}$. It is worth pointing out that, in Definition 1, the SNR-constrained compressor is not assumed to be unbiased, hence the term "general". Definition 1 covers a large class of compression schemes, e.g., the Top-k compressor [4], [10] that selects k coordinates with the largest absolute values, and the random sparsifier [14] that randomly selects components.

IV. Compressed FedAvg (CFedAvg) for Non-IID Federated Learning

In this section, we will first introduce our CFedAvg (compressed FedAvg) algorithm in Section IV-A. Then, we will present the main theoretical result and their key insights/interpretations in Section IV-B. Due to space limitation, we provide proof sketches for the main results in Section IV-C and relegate the full proofs of all theoretical results in our online technical report [20].

A. The CFedAvg Algorithmic Framework

The general CFedAvg algorithmic framework is stated in Algorithm 1. We aim to not only reduce total communication rounds, but also compress the gradients transmitted in each communication round. The algorithm contains four key stages:

1. *Local Computation:* Line 6 says that, in each communication round, each worker runs K_i local updates before communicating with the server. As shown in Line 7, each local update step takes an unbiased gradient estimator (e.g., vanilla SGD). A local learning rate $\eta_{L,t}$ is adopted for each local step (Line 8).
2. *Gradient Compression:* We compress the model changes $\mathbf{g}_t^i$ instead of the last model $\mathbf{x}_{t,K_i}^i$ in each worker, where $\mathbf{g}_t^i = \mathbf{x}_{t,K}^i - \mathbf{x}_t$ for homogeneous local step ($K_i = K, \forall i \in [m]$) in Line 10 or $\mathbf{g}_t^i = \frac{1}{K_i}(\mathbf{x}_{t,K_i}^i - \mathbf{x}_t)$ for heterogeneous local step (different local steps $K_i, \forall i \in [m]$) in Line 11. Before compressing the parameters, we add the error term to compensate the parameter in each worker in Line 12, i.e., $\mathbf{p}_t^i = \mathbf{g}_t^i + \mathbf{e}_t^i, \forall i \in [m]$. Then, we compress the parameter $\mathbf{p}_t^i$ and send the result $\tilde{\Delta}_t^i = \mathcal{C}(\mathbf{p}_t^i)$ to the server, where $\mathcal{C}(\cdot)$ denotes a general SNR-constrained compressor.
3. *Error Feedback:* We update error term after gradient compression in each communication round in Line 15, representing the information loss due to compression. This would be used later to compensate the parameters in the next communication round to ensure not too much parameter information is lost.
4. *Global Update:* Upon the reception of all returned parameters, the server updates the parameters using a global learning rate η and broadcasts the new model parameters to all workers.

ISBN 978-3-903176-37-9 © 2021 IFIP

Algorithm 1 The General CFedAvg Algorithmic Framework.

Initialize $\mathbf{x}_0$.
for $t = 0, \cdots, T-1$ **do**
  Initialize $\mathbf{e}_0^i = 0, i \in [m]$ if $t = 0$.
  for each worker $i \in [m]$ in parallel **do**
    $\mathbf{x}_{t,0}^i = \mathbf{x}_t$
    for $k = 0, \cdots, K_i - 1$ **do**
      Compute an unbiased stochastic gradient estimate $\mathbf{g}_{t,k}^i = \nabla F_i(\mathbf{x}_{t,k}^i, \xi_{t,k}^i)$ of $\nabla F_i(\mathbf{x}_{t,k}^i)$.
      Local update: $\mathbf{x}_{t,k+1}^i = \mathbf{x}_{t,k}^i - \eta_{L,t}\mathbf{g}_{t,k}^i$.
    end for
    For Homogeneous Local Steps: $\mathbf{g}_t^i = \mathbf{x}_{t,K}^i - \mathbf{x}_t$.
    For Heterogeneous Local Steps: $\mathbf{g}_t^i = \frac{1}{K_i}(\mathbf{x}_{t,K_i}^i - \mathbf{x}_t)$.
    $\mathbf{p}_t^i = \mathbf{g}_t^i + \mathbf{e}_t^i$.
    $\tilde{\Delta}_t^i = \mathcal{C}(\mathbf{p}_t^i)$ ($\mathcal{C}(\cdot)$ is an SNR-constrained compressor).
    Send $\tilde{\Delta}_t^i$ to server.
    $\mathbf{e}_{t+1}^i = \mathbf{p}_t^i - \tilde{\Delta}_t^i$.
  end for
  At Parameter Server:
    Receive $\tilde{\Delta}_t^i, i \in [m]$.
    $\tilde{\Delta}_t = \frac{1}{m}\sum_{i\in[m]} \tilde{\Delta}_t^i$.
    Server Update: $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta\tilde{\Delta}_t$.
    Broadcast $\mathbf{x}_{t+1}$ to each worker.
end for

B. Main Theoretical Results

In this subsection, we will establish the convergence results of our proposed CFedAvg algorithmic framework. Our convergence results are proved under the following mild assumptions:

Assumption 1. *(Lipschitz Smoothness) There exists a constant $L > 0$, s.t. $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \forall i \in [m]$.*

Assumption 2. *(Unbiased Gradient Estimator) Let ξ_t^i be a random local sample in the t-th round at worker i. The gradient estimator is unbiased, i.e., $\mathbb{E}[\nabla F_i(\mathbf{x}, \xi^i)] = \nabla F_i(\mathbf{x})$, $\forall i \in [m]$.*

Assumption 3. *(Bounded Local/Global Variance) There exist two constants $\sigma_L \geq 0$ and $\sigma_G \geq 0$, such that the variance of each local gradient estimator is bounded by $\mathbb{E}[\|\nabla F_i(\mathbf{x}, \xi_t^i) - \nabla F_i(\mathbf{x})\|^2] \leq \sigma_L^2$, and the global variability of the local gradient is bounded by $\|\nabla F_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \sigma_G^2$, $\forall i \in [m]$.*

The first two assumptions and the bounded local variance assumption in Assumption 3 are standard assumptions in the convergence analysis of stochastic gradient-type algorithms in non-convex optimization (e.g., [1]). We use a universal bound σ_G to quantify the heterogeneity of the *non-i.i.d. datasets* among different workers. This assumption has also been used in other works for FL with non-i.i.d. datasets [21] as well as in decentralized optimization [1]. It is worth noting that we do *not* require a bounded gradient assumption, which is often used in FL optimization analysis [1]. With the above assumptions, we are now in a position to present our main theoretical results. First, we state a useful result in Lemma 1:

Lemma 1. *(Bounded Error). For any local learning rate satisfying $\eta_{L,t} \leq \frac{1}{8LK}$, the error term can be upper bounded by $\sum_{t=0}^{T-1} \|\mathbf{e}_t\|^2 \leq \sum_{t=0}^{T-1} h(\gamma, \alpha_t)\|\Delta_t\|^2$, where $\mathbf{e}_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{e}_t^i, \Delta_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{g}_t^i, h(\gamma, \alpha_t) = (1/\gamma)(1 + 1/a)b\alpha_t, \alpha_t = \frac{\mathbb{E}\|\tilde{\Delta}_t\|^2}{\mathbb{E}\|\Delta_t\|^2}$ and a and b are constants such that $\gamma\epsilon - 1 \geq a$ and $\frac{1}{1-\epsilon} \leq b$ for $\epsilon \in (0, 1)$.*

Lemma 1 implies that the error term cannot grow arbitrarily large with a proper SNR threshold γ (determined by compression rate) for which $h(\gamma, \alpha_t)$ does not go to infinity. The error is upper bounded by the accumulated parameters Δ_t. This suggests that the total information loss due to compression is only a fraction of the total of the accumulated parameters.

1) CFedAvg with Constant Learning Rates (Homogeneous Local Steps): We consider a simpler case where CFedAvg uses constant learning rates (i.e., $\eta_{L,t} \equiv \eta_L$, $\forall t$) and homogeneous local steps (i.e., $K_i \equiv K$, $\forall i$). In this case, by Lemma 1, we can establish convergence result for CFedAvg as follows:

Theorem 1. *(Convergence Rate of CFedAvg). Choose constant local and global learning rates η_L and η s.t. $\eta_L \leq \frac{1}{8LK}$, $\eta\eta_L < \frac{1}{KL}$ and $\eta\eta_L K(L^2 h(\gamma, \alpha_t) + 1 + L) \leq 1, \forall t \in [T]$. Under Assumptions 1–3, the sequence $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L KT} + \Phi, \tag{2}$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2 h(\gamma, \alpha) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$, c is a constant, $h(\gamma, \alpha)$ is defined by $h(\gamma, \alpha) = \frac{1}{T}\sum_{t=0}^{T-1} h(\gamma, \alpha_t)$, $h(\gamma, \alpha_t)$ is defined the same as that in Lemma 1, $f_0 = f(\hat{\mathbf{x}}_0)$ and f_* is the optimal.

Remark 1 (Decomposition of The Bound)**.** The bound of the convergence rate in (2) contains two terms: the first term is a vanishing term $\frac{f_0 - f_*}{c\eta\eta_L KT}$ as T increases, and the second term is a constant Φ independent of T. Note that Φ depends on three factors: local variance σ_L, global variance σ_G, and the number of local steps K. We can further decompose the constant term Φ into two parts. The first part of Φ is due to the local variance of the stochastic gradient in each local SGD step for each worker. It shrinks at the rate $\frac{1}{m}$ with respect to the number of workers m, which favors large distributed systems. This makes intuitive sense since more workers means more training samples in one communication round, thus decreasing the local variance due to stochastic gradients. It can also be viewed as having a larger batch size to decrease the variance in SGD. The cumulative variance of K local steps contributes to the second term of Φ. This term depends on the number of local steps K, local learning rate η_L, local variance σ_L^2 and global variance σ_G^2 (non-i.i.d. data), but independent of m.

Remark 2 (Comparison with FedAvg without Compression)**.** Compared to the results of generalized FedAvg without compression, i.e., $\Phi \triangleq \frac{1}{c}[\frac{L\eta\eta_L}{2m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$ (cf. [8]), we have two key observations. First, the compressor, which significantly reduces the communication cost, only slightly increases the constant Φ by $\frac{1}{c}[(\frac{1}{2}L^2\eta_L^2 h(\gamma, \alpha) + \frac{1}{2})\frac{\eta_L}{m}\sigma_L^2]$ and does not change the convergence rate $O(1/T)$. This extra variance comes from increased local variance due to more noisy stochastic gradients after compression, and is independent of the number of local steps K. This insight means that one can *safely* use more local steps K without worrying about any accumulative effect due to compression, which is a somewhat

ISBN 978-3-903176-37-9 © 2021 IFIP

surprising and counter-intuitive insight. On the other hand, this extra variance shrinks at rate $\frac{1}{m}$, which favors large FL systems with more workers. This makes intuitive sense since the server could obtain more information from the model updates with more workers, although each worker's information is noisy due to compression. In other words, as the number of worker m increases, the extra variance due to compression becomes negligible. Second, the extra variance due to compression is irrelevant to the global variance σ_G^2 from the non-i.i.d. datasets. The global variance measures the heterogeneity among the loss functions of the workers with non-i.i.d. local datasets. Intuitively speaking, the compressor only introduces extra noise to the model's information. Thus, the compression operation only increases the local variance of the stochastic gradient and is likely irrelevant to non-i.i.d. datasets and local steps in FL.

Remark 3 (Choice of Compressor)**.** Our analysis also shows that many compression methods (e.g., [3], [12], [14]) that work well in traditional distributed and decentralized learning can also be used with the CFedAvg algorithm in FL. With error feedback, CFedAvg enjoys the same benefits as traditional distributed learning even with the use of local steps in FL and under non-i.i.d. datasets. Moreover, we do not restrict the choice of compressors. Thus, CFedAvg works with both biased and unbiased compressors, as long as they satisfy Definition 1. However, care must still be taken when one chooses a compressor in CFedAvg since it does not mean any compression methods with arbitrary compression rate could work. Consider a compressor with arbitrary compression rate such that $\gamma \to 1$ and then $\epsilon \to 1$, so $h(\gamma, \alpha) \to \infty$. This compressed model is too noisy to be trained since no useful information is available for the server. This will also be empirically verified in Section V, where significant performance degradation due to an overly aggressive compression rate can be observed.

From Theorem 1, we immediately have the following convergence rate with a proper choice of learning rates:

Corollary 2. *(Linear Speedup for Convergence). If* $\eta_L = \frac{1}{\sqrt{T}KL}$ *and* $\eta = \sqrt{Km}$*, the convergence rate of Algorithm 1 is:* $\mathcal{O}(\frac{h(\gamma,\alpha)}{\sqrt{mKT}} + \frac{1}{T}) = \mathcal{O}(\frac{1}{\sqrt{mKT}} + \frac{1}{T})$.

Remark 4. With a proper SNR-threshold γ such that $h(\gamma, \alpha) = \mathcal{O}(1)$, CFedAvg achieves a linear speedup for convergence for non-i.i.d. datasets, i.e., $\mathcal{O}(\frac{1}{\sqrt{mKT}})$ convergence rate for $T \geq mK$. This matches the convergence rate in distributed learning and FL without compression [1], [7], [22], [8], indicating CFedAvg achieves high communication efficiency while not sacrificing learning accuracy in FL. When degenerating to i.i.d. case, CFedAvg still achieves the linear speedup effect, matching the results of previous work in distributed and decentralize learning [23], [11]. The most related work to this paper is Qsparse-local-SGD [17], which combines unbiased quantization, sparsification and local steps together and is able to recover or generalize other compression methods. It achieves $\mathcal{O}(\frac{1}{\sqrt{mKT}} + \frac{mK}{T})$ convergence, implying a linear speedup for $T \geq m^3K^3$. However, for large systems (m) and large local steps (K), their T will be very large. Besides a better convergence rate in this paper, we have a weak assumption (no bounded gradient assumption). Challenges from this relaxed assumption are addressed in Section IV-C.

2) CFedAvg with Decaying Learning Rates (Homogeneous Local Steps): We can see from Corollary 2 that the choice of constant learning rate requires knowledge of time horizon T before running the algorithm, which may not be available in practice. In other words, the constant-learning-rate version of CFedAvg is not an *"anytime"* algorithm. To address this limitation, we propose CFedAvg with decaying learning rate, which is an anytime algorithm.

Theorem 3. *(Convergence with Decaying Learning Rate). Choose decaying local learning rate* $\eta_{L,t}$ *and constant global learning rates* η *s.t.* $\eta_{L,t} \leq \frac{1}{8LK}$*,* $\eta\eta_{L,t} < \frac{1}{KL}$ *and* $\eta\eta_{L,t}K(L^2h(\gamma,\alpha_t)+1+L) \leq 1, \forall t \in [T]$*. Under Assumptions 1–3, the sequence* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 satisfies:*

$$\mathbb{E}\|\nabla f(\mathbf{z})\|_2^2 \leq \frac{f_0 - f_*}{c\eta K H_T} + \Phi, \tag{3}$$

where $\Phi \triangleq (\frac{1}{2}L^2h(\gamma,\alpha) + \frac{1}{2} + \frac{L}{2})\frac{\eta}{cmH_T}\sigma_L^2 \sum_{t=0}^{T-1}\eta_{L,t}^2 + \frac{5KL^2}{2cH_T}(\sigma_L^2 + 6K\sigma_G^2)\sum_{t=0}^{T-1}\eta_{L,t}^3$*,* $H_T = \sum_{t=0}^{T-1}\eta_{L,t}$ *and* $\mathbf{z}$ *is sampled from* $\{\mathbf{x}_t\}, \forall t \in [T]$ *with probability* $\mathbb{P}[\mathbf{z} = \mathbf{x}_t] = \frac{\eta_{L,t}}{H_T}$*. Other parameters are defined the same as Theorem 1.*

Corollary 4. *Let* $\eta_L = \frac{1}{\sqrt{t+a}KL}$ *(constant* $a > 0$*) and* $\eta = \sqrt{Km}$*, the convergence rate of Algorithm 1 is:* $\tilde{\mathcal{O}}(\frac{1}{\sqrt{mKT}}) + \mathcal{O}(\frac{1}{\sqrt{T}})$.

Remark 5. When $h(\gamma, \alpha)$ is a constant, our CFedAvg algorithm achieves $\mathcal{O}(\frac{1}{\sqrt{mKT}}\ln(T) + \frac{1}{\sqrt{T}})$ convergence rate, which is slower than that with constant learning rates. However, it is still better than $\mathcal{O}(1/\ln(T))$ proved in Qsparse-local-SGD [17].

3) CFedAvg with Heterogeneous Local Steps (Constant Learning Rates): In practice, FL systems are often formed by heterogeneous devices with various computing capabilities and resource limits (e.g., computation speed, memory size). Hence, fixing the same number of local steps at all workers results in: i) fast workers being idle after finishing computation in each round and thus wasting resources and ii) slow workers being stragglers in the FL system. Next, we show that it is possible to perform heterogeneous local steps among workers in CFedAvg while still offering theoretical performance guarantees.

Theorem 5. *(Convergence of Heterogeneous Local Steps). Choose constant local and global learning rates* η_L *and* η *s.t.* $\eta_L \leq \frac{1}{8LK_i}$*,* $\eta\eta_L < \frac{1}{K_iL}, \forall i \in [m]$ *and* $\eta\eta_L(L^2h(\gamma,\alpha_t)+1+L) \leq 1, \forall t \in [T]$*. Under Assumptions 1–3, sequence* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 with heterogeneous local steps satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L T} + \Phi, \tag{4}$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2h(\gamma,\alpha) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m^2}\sum_{i=1}^m \frac{1}{K_i}\sigma_L^2 + \frac{5\eta_L^2L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2 + 6K_i\sigma_G^2)]$*. Other parameters are defined the same as Theorem 1.*

Corollary 6. *(Linear Speedup with Heterogeneous Local Steps). Let* $\eta_L = \frac{1}{\sqrt{T}L}$ *and* $\eta = \sqrt{K_{min}m}$*. The convergence rate of Algorithm 1 with heterogeneous local steps and constant*

ISBN 978-3-903176-37-9 © 2021 IFIP

learning rate is: $\mathcal{O}(\frac{1}{\sqrt{mK_{min}T}}) + \mathcal{O}(\frac{K_{max}^2}{T})$, *where* $K_{min} = \min_{i\ in[m]}\{K_i\}$ *and* $K_{max} = \max_{i\in[m]}\{K_i\}$.

Remark 6. Allowing heterogeneous local steps at different workers entails efficient FL implementations in practice. Specifically, instead of waiting for all workers to finish the same number of local steps, the server can broadcast a periodic "time-out" signal to all workers to interrupt their local updates. All worker can simply submit their current computation results even if they are in different stages in their local updates. We can see from Corollary 6 that CFedAvg with heterogeneous local steps can still achieve the same linear speedup for convergence $\mathcal{O}(\frac{1}{\sqrt{mK_{min}T}})$ for $T \geq mK_{min}K_{max}^4$, while having the flexibility of choosing different number of local steps at each worker. Although this result is for CFedAvg, our theoretical analysis is general and can be applied to uncompressed FL algorithms (e.g., FedAvg) to allow heterogeneous $K_i^t, \forall i \in [m]$.

To our knowledge, our work is the first to show that performing heterogeneous local steps among workers still achieves theoretical performance guarantees for FL. We note that, in asynchronous Qsparse-local-SGD [17], the workers synchronize with the server at different times based on the workers, but it is required that each worker follows the same rate and performs the same local steps. Rizk et al. [24] proposed dynamic federated learning (without compression) to use heterogeneous local steps at each worker. But they require the local steps to be known in advance in order to scale the gradient in each local step. For our CFedAvg, K_i can be set in an ad-hoc fashion without being known in advance.

C. Proof of the Main Results

Proof Sketch for Theorem 1. Due to space limitation, we only provide a proof sketch here. For convenience, we define the following notation: $\mathbf{e}_t \triangleq \frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i$, $\hat{\mathbf{x}}_t \triangleq \mathbf{x}_t + \eta\mathbf{e}_t$ and $\Delta_t = \frac{1}{m}\sum_{i\in[m]}\Delta_t^i = \frac{1}{m}\sum_{i\in[m]}\mathbf{g}_t^i$. Note that we do not assume bounded gradients, which poses two challenges. One key difficulty is to bound the error feedback term $\mathbf{e}_t^i, \forall i \in [m]$ due to the compression. $\mathbf{e}_t^i$ cannot be bounded in this fashion if the bounded gradient assumption is relaxed. In our analysis, we manage to bound $\mathbf{e}_t$ rather than $\mathbf{e}_t^i$ individually. By using $\|\mathbf{e}_t\|^2 = \big\|\frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i\big\|^2 \leq \frac{1}{m}\sum_{i=1}^m \big\|\mathbf{e}_t^i\big\|^2 = \big\|\bar{\mathbf{e}}_t\big\|^2$ and the recursion $\big\|\bar{\mathbf{e}}_t\big\|^2 \leq (1/\gamma)\frac{1}{\epsilon}\big\|\bar{\mathbf{e}}_{t-1}\big\|^2 + (1/\gamma)\frac{1}{1-\epsilon}\big\|\bar{\Delta}_{t-1}\big\|^2$, where $\|\bar{\Delta}_t\|^2 = \frac{1}{m}\sum_{i=1}^m \big\|\Delta_t^i\big\|^2$, $\sum_{t=0}^{T-1}\|\mathbf{e}_t\|^2$ can be bounded in Lemma 1. Second, the lack of bounded gradient assumption also results in difficulty in bounding the model drift stemming from the non-i.i.d. datasets and the increase of the local steps, i.e., $\|\mathbf{x}_i - \bar{\mathbf{x}}\|, \forall i \in [m]$, where $\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^m \mathbf{x}_i$. To address this challenge, we derive the recursive relation based on communication round instead of local steps. Thanks to the virtual variable $\hat{\mathbf{x}}_t$, we have the recursive relation of $\hat{\mathbf{x}}_t$: $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \eta\Delta_t$, where t is the index of communication round.

After addressing these two challenges, Assumption 1 yields per-communication-round descent as follows:

$$\mathbb{E}_t f(\hat{\mathbf{x}}_{t+1}) - f(\hat{\mathbf{x}}_t) \leq \underbrace{\mathbb{E}_t\big\langle\nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t\big\rangle}_{A_1} + \frac{L\eta^2}{2}\underbrace{\mathbb{E}_t\|\Delta_t\|^2}_{A_2}. \quad (5)$$

For A_1, the first step is to transform variable $\hat{\mathbf{x}}_t$ to $\mathbf{x}_t$ since only $\mathbf{x}_t$ is involved in the update. Towards this end, we have:

$$A_1 \leq \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 + \eta\big\langle\nabla f(\mathbf{x}_t), \Delta_t + K\eta_L\nabla f(\mathbf{x}_t)\big\rangle\big]. \quad (6)$$

Further bounding the last term of equation (6), we have:

$$A_1 \leq \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2\big] - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 + \eta\eta_L K(\frac{1}{2} + 15K^2\eta_L^2L^2)\|\nabla f(x_t)\|^2 + \frac{5\eta K^2\eta_L^3L^2}{2}\times (\sigma_L^2 + 6K\sigma_G^2) - \frac{\eta\eta_L}{2Km^2}\mathbb{E}_t\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2. \quad (7)$$

For A_2, by using $\mathbb{E}[\|\mathbf{x}\|^2] = \mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|^2] + \|\mathbb{E}[\mathbf{x}]\|^2$ to decompose Δ_t and assumption 3, we have:

$$A_2 = \mathbb{E}_t[\|\Delta_t\|^2] \leq \frac{K\eta_L^2}{m}\sigma_L^2 + \frac{\eta_L^2}{m^2}\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2. \quad (8)$$

Combining (7) and (8), $\mathbb{E}_t\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2$ is canceled with proper learning rates. Then, we simplify (5) as:

$$\mathbb{E}f(\hat{\mathbf{x}}_{t+1}) - \nabla f(\hat{\mathbf{x}}_t) \leq (\frac{1}{2}L^2\eta^2h(\gamma,\alpha_t) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{K\eta_L^2}{m}\sigma_L^2 - c\eta\eta_L K\big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_L^3L^2}{2}(\sigma_L^2 + 6K\sigma_G^2).$$

Telescoping and rearranging yields the stated result. □

Theorems 3 and 5 can also be proved in a similar fashion.

V. Experimental Results

1) Experiment Settings: *1-a) Datasets and Models:* We use three datasets (non-i.i.d. version) in FL settings, including MNIST, Fashion-MNIST and CIFAR-10 . Each of these three datasets contains 10 different classes of items. To induce non-i.i.d. datasets, we partition the data based on the classes of items (p) contained in these datasets. By doing so, we can use the number of classes in worker's local dataset, denoted as p, to control the non-i.i.d. level of the datasets quantitatively. We set four levels of non-i.i.d. datasets for comparison: $p = 1, 2, 5$, and 10. We distribute the dataset among $m = 100$ workers randomly and evenly in a class-based manner, such that the local dataset at each worker contains only a subset of classes of items with the same number of training/test samples. We experiment two learning models: i) convolution neural network (CNN) (architecture is detailed in our online technical report) on MNIST and Fashion-MNIST, and ii) ResNet-18 on CIFAR-10. *1-b) Compressors:* We consider two compression methods: i) Top-k sparsification and ii) random dropping (RD) [4], [19]. For a given vector $\mathbf{x}$, Top-k sparsification compresses $\mathbf{x}$ by retaining k elements of this vector that have the largest absolute value and setting others to zero, while RD randomly drops each component with a fixed probability. we use a constant compression parameter $comp \in (0, 1]$ to represent the compression rate. *1-c) Hyper-parameters:* We set the default hyper-parameters as follows: the number of workers m=100, local learning rate η_L=0.1, global learning rate η=1.0, batch

ISBN 978-3-903176-37-9 © 2021 IFIP

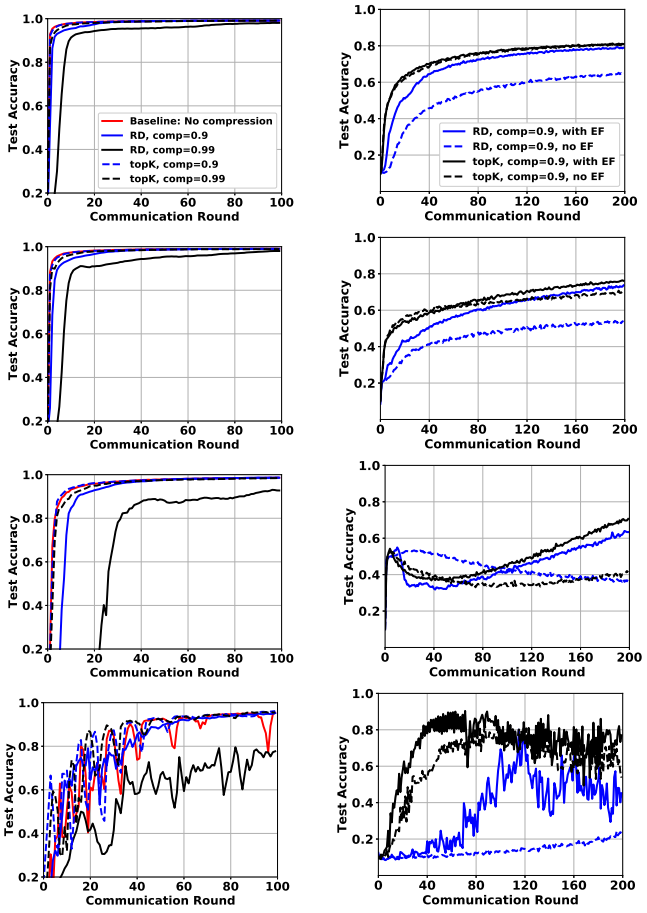
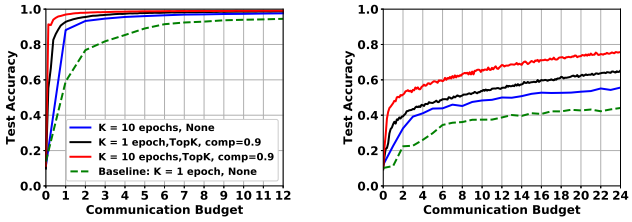


Fig. 1. Test accuracy (left: MNIST, right: CIFAR-10). The non-i.i.d. levels are $p = 10, 5, 2, 1$ from top to bottom.

size B=64, local steps K=10 epochs, total rounds T=100 for MNIST and Fashion-MNIST and T=200 for CIFAR-10.

2) Numerical Results: We now present two types of experimental results. The first is to show the effectiveness of our CFedAvg algorithm with significant communication reduction. The second is to evaluate the importance of error feedback. We only show a part of results due to space limitation.



(a) MNIST. (b) CIFAR-10.

Fig. 2. Comparison of test accuracy with same communication load for $p = 5$. The unit of the communication budget is the model size.

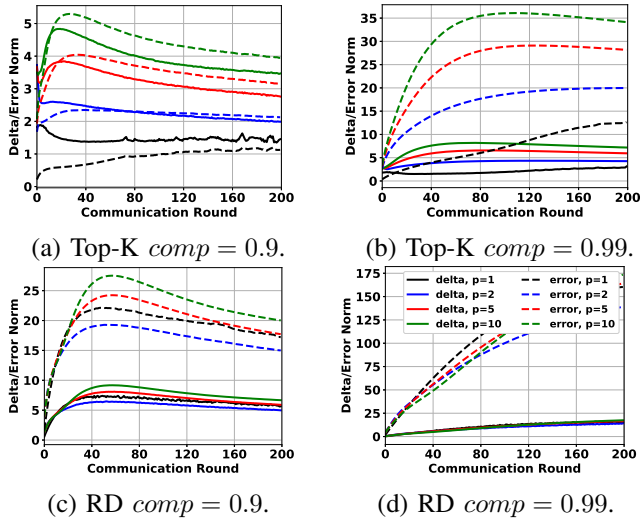
2-a) Effectiveness of Compressors: As shown in Fig. 1, the figures are for test accuracy versus communication rounds of MNIST in the left column and CIFAR-10 in the right colomn. We can see that our CFedAvg algorithm with two compressors converges for all heterogeneity levels of non-i.i.d. datasets from top ($p = 10$) to bottom ($p = 1$). For the random dropping (RD) method, it becomes worse from $comp = 0.9$ to $comp = 0.99$. While for Top-k method, both cases ($comp = 0.9$ and $comp = 0.99$) can achieve almost the same convergence speed comparable to that without any compression. This indicates that we can reduce the communication cost in each communication round by about 99% with Top-k for MNIST, i.e., we only need to transmit 1% of coordinates in the gradient vector without significantly sacrificing the convergence rate. This will greatly facilitates FL on such communication-constrained devices. Another interesting observation is that the compression methods (with error feedback) help stabilize the training process for non-i.i.d. case. Compared with i.i.d. datasets, the training curve is zigzagging for non-i.i.d. case. As the heterogeneity level of non-i.i.d. datasets increases, this zigzagging phenomena of the curves is more pronounced as shown in the figures, which we believe is an inherent feature of non-i.i.d. dataset in FL. Meanwhile, the learning curves are smoother with compression and error feedback, particularly with highly non-i.i.d. datasets, see Fig. 1 ($p = 1$). The intuition is that the compressor could filter some noises that lead to the instability of the learning curve due to model heterogeneity among workers originated from the non-i.i.d. datasets and local steps. However, this appears to require proper compression rate $comp$ based on the compression method. We do not rule out the possibility of mutual interactions that lead to poor performance between the compression and non-i.i.d. datasets.

In addition, we compare the test accuracy based on the same uplink communication budgets (the model size as the unit) among different compression methods under non-i.i.d. datasets $p = 5$ in Fig. 2. The baseline is the FedAvg with a single local step and no compression. Consistent with previous work [2], [10], both compressors and local update steps are effective to reduce communication cost, confirming our theoretical analysis.

2-b) Importance of Error Feedback: Although it is a natural idea to apply those compression methods that have been proved to be useful in traditional distributed learning to FL, there could be a significant information loss if one uses these compressors naively. It has been shown that the learning performance is poor without error feedback in compression under i.i.d. datasets [18], [25]. This conclusion is confirmed in our experiments as shown in right column of Fig. 1, where we observe the gap between cases with and without error feedback (EF) under i.i.d. case ($p = 10$). As the heterogeneity degree of non-i.i.d. datasets increases from $p = 10$ to $p = 1$, the gap becomes larger and is no longer negligible. It is obvious that both compression methods, RD and Top-k, perform better with error feedback in Fig. 1. This indicates the significant impact of error feedback. If naively applying compression, a huge amount of information could be lost, thus resulting in poor performance.

With error feedback, the error term accumulates the information that is not transmitted to the parameter server in the current communication round and then compensates the gradients in the next communication round. This is verified in Fig. 3, which shows the mean of gradient norm changes $\frac{1}{m}\sum_{i=1}^{m}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{e}_t^i\|^2$ for the total worker number $m = 100$. One key observation is that the error term is bounded under appropriate compression methods and compression rates,

ISBN 978-3-903176-37-9 © 2021 IFIP



(a) Top-K $comp = 0.9$. (b) Top-K $comp = 0.99$.

(c) RD $comp = 0.9$. (d) RD $comp = 0.99$.

Fig. 3. Mean of the norms of $\frac{1}{m}\sum_{i=1}^{100}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{100}\|\mathbf{e}_t^i\|^2$ for the ResNet-18 on CIFAR-10.

which is usually several times of the gradient change term in general. Under the same condition, the error term is much larger with aggressive compression rate. With an overly aggressive compression (e.g., Fig. 3 with $comp = 0.99$), the error term continues to grow. Thus, in general, error feedback guarantees not too much information is dropped due to compression, verifying our theoretical analysis. It has been shown that error feedback is effective in distributed/decentralized learning [18], [25], [11]. In this paper, we show that its effectiveness continues to hold in non-i.i.d. compressed FL.

VI. Conclusion

In this paper, we proposed a communication-efficient algorithmic framework called CFedAvg for FL on non-i.i.d. datasets. CFedAvg works with general (biased/unbiased) SNR-constrained compressors to reduce the communication cost and other techniques to accelerate the training. Theoretically, we analyzed the convergence rates of CFedAvg for non-convex functions with constant learning and decaying learning rates. The convergence rates of CFedAvg match that of distributed/federated learning without compression, thus achieving high communication efficiency while not significantly sacrificing learning accuracy in FL. Furthermore, we extended CFedAvg to heterogeneous local steps with convergence guarantees, which allows different workers perform different numbers of local steps to better adapt to their own circumstances. Our key observation is that the noise/variance introduced by compressors does not affect the overall convergence rate order for non-i.i.d. FL. We verified the effectiveness of our CFedAvg algorithm on three datasets with two gradient compression schemes of different compression ratios. Our results contribute to the first step toward developing advanced compression methods for communication-efficient FL.

References

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[3] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," *arXiv preprint arXiv:1802.04434*, 2018.

[4] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[5] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing systems*, 2017, pp. 1509–1519.

[6] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *The Journal of Machine Learning Research*, vol. 13, pp. 165–202, 2012.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," *arXiv preprint arXiv:1910.06378*, 2019.

[8] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *International Conference on Learning Representations*, 2021.

[9] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[10] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[11] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.

[12] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[13] N. Strom, "Scalable distributed dnn training using commodity gpu cloud computing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[14] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems*, 2018, pp. 1299–1309.

[15] N. Dryden, T. Moon, S. A. Jacobs, and B. Van Essen, "Communication quantization for data-parallel training of deep neural networks," in *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*. IEEE, 2016, pp. 1–8.

[16] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[17] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 217–226, 2020.

[18] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," *arXiv preprint arXiv:1901.09847*, 2019.

[19] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Advances in Neural Information Processing Systems*, 2018, pp. 4447–4458.

[20] H. Yang, J. Liu, and E. S. Bentley, "CFedAvg: achieving efficient communication and fast convergence in non-iid federated learning," The Ohio State Univeristy, Tech. Rep., 2020. [Online]. Available: https://kevinliu-osu-ece.github.io/publications/CFedAvg_TR.pdf

[21] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[22] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," *arXiv preprint arXiv:1905.03817*, 2019.

[23] H. Tang, X. Lian, S. Qiu, L. Yuan, C. Zhang, T. Zhang, and J. Liu, "Deepsqueeze : Decentralization meets error-compensated compression," *arXiv*, pp. arXiv–1907, 2019.

[24] E. Rizk, S. Vlaski, and A. H. Sayed, "Dynamic federated learning," *arXiv preprint arXiv:2002.08782*, 2020.

[25] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication," *arXiv preprint arXiv:1909.05350*, 2019.

ISBN 978-3-903176-37-9 © 2021 IFIP