Communication-Efficient Distributed SGD With Compressed Sensing

Yujie Tang, *Member, IEEE*, Vikram Ramanathan, Junshan Zhang, *Fellow, IEEE*, and Na Li, *Member, IEEE*

***Abstract*—We consider large scale distributed optimization over a set of edge devices connected to a central server, where the limited communication bandwidth between the server and edge devices imposes a significant bottleneck for the optimization procedure. Inspired by recent advances in federated learning, we propose a distributed stochastic gradient descent (SGD) type algorithm that exploits the sparsity of the gradient, when possible, to reduce communication burden. At the heart of the algorithm is to use compressed sensing techniques for the compression of the local stochastic gradients at the device side; and at the server side, a sparse approximation of the global stochastic gradient is recovered from the noisy aggregated compressed local gradients. We conduct theoretical analysis on the convergence of our algorithm in the presence of noise perturbation incurred by the communication channels, and also conduct numerical experiments to corroborate its effectiveness.**

***Index Terms*—Optimization algorithms, large-scale systems, distributed optimization, compressed sensing.**

I. Introduction

LARGE-SCALE distributed stochastic optimization plays a fundamental role in the recent advances of machine learning, allowing models with vast sizes to be trained on massive datasets by multiple machines. In the meantime, the past few years have witnessed an explosive growth of networks of IoT devices such as smart phones, self-driving cars, robots, unmanned aerial vehicles (UAVs), etc., which are capable of data collection and processing for many learning tasks. In many of these applications, due to privacy concerns, it is preferable that the local edge devices learn the model by cooperating with the central server but without sending their own data to the server. Moreover, the communication between the edge devices and the server is often through wireless channels, which are lossy and unreliable in nature and have limited bandwidth, imposing significant challenges, especially for high-dimensional problems.

To address the communication bottlenecks, researchers have investigated communication-efficient distributed optimization methods for large-scale problems, for both the device-server setting [1], [2] and the peer-to-peer setting [3], [4]. In this letter, we consider the device-server setting where a group of edge devices are coordinated by a central server.

Most existing techniques for the device-server setting can be classified into two categories. The first category aims to reduce the number of communication rounds, based on the idea that each edge device runs multiple local SGD steps in parallel before sending the local updates to the server for aggregation. This approach has also been called `FedAvg` [1] in federated learning and convergence has been studied in [5]–[7]. Another line of work investigates lazy/adaptive upload of information, i.e., local gradients are uploaded only when found to be informative enough [8].

The second category focuses on efficient compression of gradient information transmitted from edge devices to the server. Commonly adopted compression techniques include *quantization* [9]–[11] and *sparsification* [12]–[14]. These techniques can be further classified according to whether the gradient compression yields biased [9], [14] or unbiased [10], [13] gradient estimators. To handle the bias and boost convergence, [12], [15] introduced the error feedback method that accumulates and corrects the error caused by gradient compression at each step.

Two recent papers [16], [17] employ sketching methods for gradient compression. Specifically, each device compresses its local stochastic gradient by count sketch [18] via a common sketching operator; and the server recovers the indices and the values of large entries of the aggregated stochastic gradient from the gradient sketches. However, theoretical guarantees of count sketch were developed for recovering one *fixed* signal by randomly generating a sketching operator from a given probability distribution. During SGD, gradient signals are constantly changing, making it impractical to generate a new sketching operator for every SGD iteration. Thus the papers apply a single sketching operator to all the gradients through the optimization procedure, while sacrificing theoretical guarantees. Further, there is a limited understanding of

Manuscript received September 14, 2021; revised November 22, 2021; accepted December 6, 2021. Date of publication December 23, 2021; date of current version January 6, 2022. This work was supported in part by NSF under Grant CNS: 2003111; in part by NSF under Grant AI Institute: 2112085; and in part by ONR under Grant YIP: N00014-19-1-2217. Recommended by Senior Editor F. Dabbene. *(Corresponding author: Yujie Tang.)*

Yujie Tang, Vikram Ramanathan, and Na Li are with the School of Engineering and Applied Sciences, Harvard University, Allston, MA 02134 USA (e-mail: yujietang@seas.harvard.edu; vramanathan@g.harvard.edu; nali@seas.harvard.edu).

Junshan Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: junshan.ahang@asu.edu).

Digital Object Identifier 10.1109/LCSYS.2021.3137859

the performance when there is transmission error/noise of the uploading links.

Our Contributions: We propose a distributed SGD-type algorithm that employs compressed sensing for gradient compression. Specifically, we adopt compressed sensing techniques for the compression of local stochastic gradients at the device side, and the reconstruction of the aggregated stochastic gradients at the server side. The use of compressed sensing enables the server to approximately identify the top entries of the *aggregated* gradient without querying directly each local gradient. Our algorithm also integrates error feedback strategies *at the server side* to handle the bias introduced by compression, while keeping the edge devices to be *stateless*. We provide convergence analysis of our algorithm in the presence of additive noise incurred by the uploading communication channels, and conduct numerical experiments that justify the effectiveness of our algorithm.

Besides the related work discussed above, it is worth noting that a recent paper [19] uses compressed sensing for zeroth-order optimization, which exhibits a mathematical structure similar to this letter. However, [19] considers the centralized setting and only establishes convergence to a neighborhood of the minimizer.

Notations: For $x \in \mathbb{R}^d$, $\|x\|_p$ denotes its ℓ_p-norm, and $x^{[K]} \in \mathbb{R}^d$ denotes its best-K approximation, i.e., the vector that keeps the top K entries of x in magnitude with other entries set to 0.

II. Problem Setup

Consider a group of n edge devices and a server. Each device i is associated with a differentiable local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$, and is able to query a stochastic gradient $\mathsf{g}_i(x)$ such that $\mathbb{E}[\mathsf{g}_i(x)] = \nabla f_i(x)$. Between each device and the server are an uploading communication link and a broadcasting communication link. The goal is to solve

$$\min_{x\in\mathbb{R}^d} \quad f(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1}$$

through queries of stochastic gradients at each device and exchange of information between the server and each device.

One common approach for our problem setup is the stochastic gradient descent (SGD) method: For each time step t, the server first broadcasts the current iterate $x(t)$ to all devices, and then each device produces a stochastic gradient $g_i(t) = \mathsf{g}_i(x(t))$ and uploads it to the server, after which the server updates $x(t+1) = x(t) - \eta \cdot \frac{1}{n}\sum_i g_i(t)$. However, as the server needs to collect local stochastic gradients from each device at every iteration, the vanilla SGD may encounter significant bottleneck imposed by the uploading links if d is very large. This issue may be further exacerbated if the server and the devices are connected via lossy wireless networks of limited bandwidth, which is the case for many IoT applications.

In this letter, we investigate the situation where the communication links, particularly the uploading links from each edge device to the server, have limited bandwidth that can significantly slow down the whole optimization procedure; the data transmitted through each uploading link may also be corrupted by noise. Our goal is to develop an SGD-type algorithm for solving (1) that achieves better communication efficiency over the uploading links.

Algorithm 1: SGD With Compressed Sensing

Input: sparsity level K, size of sensing matrix $Q \times d$, step size η, number of iterations T, initial point x_0
Initialize: $x(1) = x_0$, $\varepsilon(1) = 0$
The server generates the sensing matrix $\Phi \in \mathbb{R}^{Q\times d}$ and sends it to every edge device
for $t = 1, 2 \ldots, T$ **do**
  The server sends $x(t)$ to every edge device
  foreach *device* $i = 1, \ldots, n$ **do**
    Device i samples a stochastic gradient $g_i(t) = \mathsf{g}_i(x(t))$
    Device i constructs $y_i(t) = \Phi g_i(t) \in \mathbb{R}^Q$
    Device i sends $y_i(t)$ back to the server
  end
  The server receives $\tilde{y}(t) = \frac{1}{n}\sum_{i=1}^{n} y_i(t) + w(t)$, where $w(t)$ denotes additive noise incurred by the communication channels
  The server computes $z(t) = \eta\tilde{y}(t) + \varepsilon(t)$
  The server reconstructs $\Delta(t) = \mathcal{A}(z(t); \Phi)$, where $\mathcal{A}(z(t); \Phi)$ denotes the output of the compressed sensing algorithm of choice
  The server updates $x(t+1) = x(t) - \Delta(t)$
  The server updates $\varepsilon(t+1) = z(t) - \Phi\Delta(t)$
end

III. Algorithm

Our algorithm is outlined in Algorithm 1, which is based on the SGD method with the following major ingredients.

1) Compression of local stochastic gradients using compressed sensing techniques. Here each edge device compresses its local gradient by $y_i(t) = \Phi g_i(t)$ before uploading it to the server. The matrix $\Phi \in \mathbb{R}^{Q\times d}$ is called the *sensing matrix*, and its number of rows Q is strictly less than the number of columns d. As a result, the communication burden of uploading the local gradient information can be reduced.

We emphasize that Algorithm 1 employs the *for-all* scheme of compressed sensing, which allows one Φ to be used for the compression of all local stochastic gradients (see Section III-A for more details on the *for-each* and the *for-all* schemes).

After collecting the compressed local gradients and obtaining $\frac{1}{n}\sum_{i=1}^{n} y_i(t)$ (corrupted by communication channel noise), the server recovers a vector $\Delta(t)$ by a compressed sensing algorithm, which will be used for updating $x(t)$.

2) Error feedback of compressed gradients. In general, the compressed sensing reconstruction will introduce a nonzero bias in the SGD iterations that hinders convergence. To handle this bias, we adopt the error feedback method in [12], [15] and modify it similarly as FetchSGD [17]. The resulting error feedback procedure is done purely at the server side without knowing the true aggregated stochastic gradients.

Note that the aggregated vector $\tilde{y}(t)$ is corrupted by additive noise $w(t)$ from the uploading links. This noise model incorporates a variety of communication schemes, including digital transmission with quantization, and over-the-air transmission for wireless multi-access networks [14].

We now provide more details on our algorithm design.

A. Preliminaries on Compressed Sensing

Compressed sensing [20] is a technique that allows efficient sensing and reconstruction of an approximately sparse signal. Mathematically, in the sensing step, a signal $x \in \mathbb{R}^d$ is observed through linear measurement $y = \Phi x + w$, where $\Phi \in \mathbb{R}^{Q\times d}$ is a pre-specified sensing matrix with $Q < d$, and $w \in \mathbb{R}^Q$ is additive noise. Then in the reconstruction step, one recovers the original signal x by approximately solving

$$\hat{x} = \arg\min_z \ \|z\|_0 \quad \text{s.t.} \ \ y = \Phi z, \quad (w = 0) \tag{2}$$

$$\hat{x} = \arg\min_z \ \frac{1}{2}\left\|y - \Phi z\right\|_2^2 \quad \text{s.t.} \ \ \|z\|_0 \le K, \quad (w \neq 0) \tag{3}$$

where K restricts the number of nonzero entries in $\hat{x}$.

Both (2) and (3) are NP-hard nonconvex problems, and researchers have proposed various compressed sensing algorithms for obtaining approximate solutions. As discussed below, the reconstruction error $\|\hat{x} - x\|$ will heavily depend on i) the design of the sensing matrix Φ, and ii) whether the signal x can be well approximated by a sparse vector.

Design of the sensing matrix Φ: Compressed sensing algorithms can be categorized into two schemes [21]: i) the *for-each* scheme, in which a probability distribution over sensing matrices is designed to provide desired reconstruction for a fixed signal, and every time a new signal is to be measured and reconstructed, one needs to randomly generate a new Φ; ii) the *for-all* scheme, in which a single Φ is used for the sensing and reconstruction of all possible signals. We mention that count sketch is an example of a for-each scheme algorithm. In this letter, we choose the for-all scheme so that the server doesn't need to send a new matrix to each device per iteration.

To ensure that the linear measurement $y = \Phi x$ can discriminate approximately sparse signals, researchers have proposed the *restricted isometry property* (RIP) [20] as a condition on Φ.

Definition 1: We say that $\Phi \in \mathbb{R}^{Q\times d}$ satisfies the (K, δ)-restricted isometry property, if $(1-\delta)\|x\|_2^2 \le \|\Phi x\|_2^2 \le (1+\delta)\|x\|_2^2$ for any $x \in \mathbb{R}^d$ that has at most K nonzero entries.

The restricted isometry property on Φ is fundamental for analyzing the reconstruction error of many compressed sensing algorithms under the for-all scheme [22].

Metric of sparsity: The classical metric of sparsity is the ℓ_0 norm defined as the number of nonzero entries. However, for our setup, the vectors to be compressed can only be approximately sparse in general, which cannot be handled by the ℓ_0 norm as it is not stable under small perturbations. Here, we adopt the following sparsity metric from [23]:

$$\mathrm{sp}(x) := \|x\|_1^2/(\|x\|_2^2 \cdot d), \qquad x \in \mathbb{R}^d \backslash \{0\}. \tag{4}$$

The continuity of $\mathrm{sp}(x)$ indicates that $\mathrm{sp}(x)$ is robust to small perturbations on x, and it can be shown that $\mathrm{sp}(x)$ is *Schur-concave*, meaning that it can characterize *approximate sparsity* of a signal. $\mathrm{sp}(x)$ has also been used in [23] for performance analysis of compressed sensing algorithms.

B. Details of Algorithm Design

Generation of Φ: As mentioned before, we choose compressed sensing under the *for-all* scheme for gradient compression and reconstruction. We require that the sensing matrix $\Phi \in \mathbb{R}^{Q\times d}$ have a low storage cost, since it will be transmitted to and stored at each device; Φ should also satisfy RIP so that the compressed sensing algorithm $\mathcal{A}$ has good reconstruction performance. The following proposition suggests a storage-friendly approach for generating matrices satisfying RIP.

Proposition 1 [24]: Let $B \in \mathbb{R}^{d\times d}$ be an orthogonal matrix with entries of absolute values $O(1/\sqrt{d})$, and let $\delta > 0$ be sufficiently small. For some $Q = \tilde{O}(\delta^{-2}K\log^2 K \log d)$,[1] let $\Phi \in \mathbb{R}^{Q\times d}$ be a matrix whose Q rows are chosen uniformly and independently from the rows of B, multiplied by $\sqrt{d/Q}$. Then, with high probability, Φ satisfies the (K, δ)-RIP.

This proposition indicates that, we can choose a "base matrix" B satisfying the condition in Proposition 1, and then randomly choose Q rows to form Φ. In this way, Φ can be stored or transmitted by merely the corresponding row indices in B. Note that Proposition 1 only requires Q to have logarithm dependence on d. Candidates of the base matrix B include the discrete cosine transform (DCT) matrix and the Walsh-Hadamard transform (WHT) matrix, as both DCT and WHT and their inverses have fast algorithms of time complexity $O(d\log d)$, implying that multiplication of Φ or $\Phi^\top$ with any vector can be finished within $O(d\log d)$ time.

Choice of the compressed sensing algorithm: We let $\mathcal{A}$ be the *Fast Iterative Hard Thresholding* (FIHT) algorithm [25]. Our experiments suggest that FIHT achieves a good balance between computation efficiency and empirical reconstruction error compared to other algorithms we have tried.

We note that FIHT has a tunable parameter K that controls the number of nonzero entries of $\Delta(t)$. This parameter should accord with the sparsity of the vector to be recovered (see Section III-C for theoretical results). In addition, the server can broadcast the sparse vector $\Delta(t)$ instead of the whole $x(t)$ for the edge devices to update their local copies of $x(t)$, which saves communication over the broadcasting links.

Error feedback: We adopt error feedback to facilitate convergence of Algorithm 1. The following lemma verifies that Algorithm 1 indeed incorporates the error feedback steps in [15]; the proof is straightforward which we omit here.

Lemma 1: Consider Algorithm 1, and suppose Φ is generated according to Proposition 1. Then for each t, there exist unique $p(t) \in \mathbb{R}^d$ and $e(t) \in \mathbb{R}^d$ satisfying $z(t) = \Phi p(t) + \eta w(t)$, $\varepsilon(t) = \Phi e(t)$, $e(1) = 0$ such that

$$\begin{aligned} p(t) &= \eta g(t) + e(t), \quad x(t+1) = x(t) - \Delta(t), \\ \Delta(t) &= \mathcal{A}(\Phi p(t) + \eta w(t); \Phi), \\ e(t+1) &= p(t) - \Delta(t) + \frac{\eta Q}{d}\Phi^\top w(t). \end{aligned} \tag{5}$$

where $g(t) := \frac{1}{n}\sum_{i=1}^n g_i(t)$.

By comparing Lemma 1 with [15, Algorithm 2], we see that the only difference lies in the presence of communication channel noise $w(t)$ in our setting. In addition, since error feedback is implemented purely at the server side, the edge devices will be *stateless* during the whole optimization procedure.

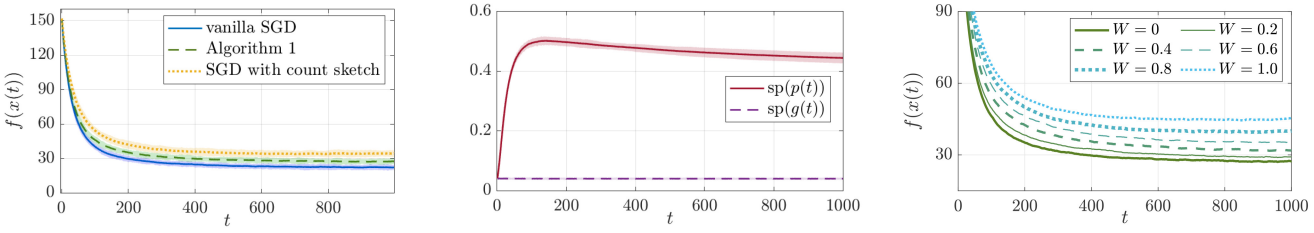
C. Theoretical Analysis

First, we make the following technical assumptions:

Assumption 1: $f(x)$ is convex and has a minimizer $x^* \in \mathbb{R}^d$. Furthermore, $f(x)$ is L-smooth for some $L > 0$, i.e., $\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x - y\|_2$ for all $x, y \in \mathbb{R}^n$.

Assumption 2: There exists $G > 0$ such that $\mathbb{E}[\|g_i(t) - \nabla f_i(x(t))\|_2^2] \le G^2$ for all t for any i.

[1]The $\tilde{O}$ notation hides logarithm dependence on $1/\delta$.



(a) Convergence of three algorithms ($w(t) = 0$).

(b) Evolution of $\mathrm{sp}(p(t))$ and $\mathrm{sp}(g(t))$ for Algorithm 1 ($w(t) = 0$).

(c) Convergence of Algorithm 1 with different amplitudes of $w(t)$.

Fig. 1. Curves represent the average of 50 random trials, and light-colored shades represent 3-sigma confidence intervals.

Assumption 3: The communication channel noise $w(t)$ satisfies $\mathbb{E}[\|w(t)\|_2^2] \le \sigma^2$ for each t.

Our theoretical analysis will be based on the following result on the reconstruction error of FIHT:

Lemma 2 [25, Corollary I.3]: Let K be the maximum number of nonzero entries of the output of FIHT. Suppose the sensing matrix $\Phi \in \mathbb{R}^{Q\times d}$ satisfies $(4K, \delta_{4K})$-RIP for sufficiently small δ_{4K}. Then, for any $x \in \mathbb{R}^d$ and $w \in \mathbb{R}^Q$,

$$\begin{aligned}&\|\mathcal{A}(\Phi x + w; \Phi) - x\|_2 \\ &\quad \le (C_{\mathcal{A},\mathrm{s}} + 1)\left\|x - x^{[K]}\right\|_2 + \frac{C_{\mathcal{A},\mathrm{s}}}{\sqrt{K}}\left\|x - x^{[K]}\right\|_1 + C_{\mathcal{A},\mathrm{n}}\|w\|_2 \quad (6)\end{aligned}$$

where $C_{\mathcal{A},\mathrm{s}}$ and $C_{\mathcal{A},\mathrm{n}}$ are constants that depend on δ_{4K}.

We are now ready to establish convergence of Algorithm 1.

Theorem 1: Let K be the maximum number of nonzero entries of the output of FIHT. Suppose the sensing matrix $\Phi \in \mathbb{R}^{Q\times d}$ satisfies $(4K, \delta_{4K})$-RIP for sufficiently small δ_{4K}. Furthermore, assume that

$$\mathrm{sp}(p(t)) \le \gamma \cdot \frac{2K/d}{[1 + C_{\mathcal{A},\mathrm{s}}(3 - 2K/d)]^2} \quad (7)$$

for all $t \ge 1$ for some $\gamma \in (0, 1)$, where $p(t)$ is defined in Lemma 1. Then for sufficiently large T, by choosing $\eta = L^{-1}\sqrt{n/T}$, we have that

$$\begin{aligned}\mathbb{E}\big[f(\bar{x}(t)) - f^*\big] \le{}& \frac{L\|x(1) - x^*\|_2^2 + G^2/L}{\sqrt{nT}} \\ &+ \frac{6}{T}\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)^2 L} + \frac{2n\big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\big)^2\sigma^2}{(1-\gamma)L}\right],\end{aligned}$$

where $\bar{x}(t) \coloneqq \frac{1}{T}\sum_{t=1}^{T} x(t)$ and $f^* \coloneqq f(x^*)$.

Remark 1: Theorem 1 requires $\mathrm{sp}(p(t))$ to remain sufficiently low. This condition is hard to check and can be violated in practice (see Section IV). However, our numerical experiments seem to suggest that even if the condition (7) is violated, Algorithm 1 may still exhibit relatively good convergence behavior when the gradient $g(t)$ itself has a relatively low sparsity level. Theoretical investigation on these observations will be interesting future directions.

IV. Numerical Results

A. Test Case With Synthetic Data

We conduct numerical experiments on a synthetic test case. We set the dimension to be $d = 2^{14}$ and the number of edge devices to be $n = 20$. The local objectives are of the form $f_i(x) = \frac{1}{2}(x - x_0)^\top A_i(x - x_0)$ where each $A_i \in \mathbb{R}^{d\times d}$ is diagonal. We generate A_i such that the diagonal entries of $A \coloneqq \frac{1}{n}\sum_i A_i$ is $A_{jj} = e^{-j/300} + 0.001$ for each j while the diagonals of each A_i are dense. We let $\mathsf{g}_i(x)$ give approximately sparse stochastic gradients for every $x \in \mathbb{R}^d$.

We test three algorithms: the uncompressed vanilla SGD, Algorithm 1, and SGD with count sketch. The SGD with count sketch just replaces the gradient compression and reconstruction of Algorithm 1 by the count sketch method [18]. We set $K = 500$ for both Algorithm 1 and SGD with count sketch. For Algorithm 1, we generate Φ from the WHT matrix and uses the FFHT library [26] for fast WHT. We set $T = 1000$, $\eta = 1/\sqrt{T}$ and $x_0 = 0$ for all three algorithms.

Fig. 1(a) illustrates the convergence of the three algorithms with no communication channel error. For Algorithm 1, we set $Q = 5000$ (the compression rate d/Q is 3.28), and for SGD with count sketch we set the sketch size to be 16×500 (the compression rate is $d/(16 \times 500) = 2.05$). We see that Algorithm 1 has better convergence behavior while also achieves higher compression rate compared to SGD with count sketch. Our experiments suggest that for approximately sparse signals, FIHT can achieve higher reconstruction accuracy and more aggressive compression than count sketch, and for signals that are not very sparse, FIHT also seems more robust.

Fig. 1(b) shows the evolution of $\mathrm{sp}(p(t))$ and $\mathrm{sp}(g(t))$ for Algorithm 1. We see that $\mathrm{sp}(p(t))$ is small for the first few iterations, and then increases and stabilizes around 0.5, which suggests that the condition (7) is likely to have been violated for large t. On the other hand, Fig. 1(a) shows that Algorithm 1 can still achieve relatively good convergence behavior. This indicates a gap between the theoretical results in Section III-C and the empirical results, and suggests our analysis could be improved. We leave relevant investigation as future work.

Fig. 1(c) illustrates the convergence of Algorithm 1 with different levels of communication channel noise. Here the entries of $w(t)$ are i.i.d. sampled from $\mathcal{N}(0, W^2)$ with $W \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. We see that the convergence of Algorithm 1 gradually deteriorates as W increases, suggesting its robustness against communication channel noise.

B. Federated Learning With CIFAR-10 Dataset

We test our algorithm on training a residual network with 668426 trainable parameters on the CIFAR-10 dateset. We set $n = 100$ and split the training dataset such that all local datasets are i.i.d.[2] The results are shown in Fig. 2. We see that Algorithm 1 is able to achieve $2\times$ upload compression with marginal effect on the training and testing accuracy over

[2]The detailed setup is provided in [27], together with analysis on the nonconvex case and more detailed discussions on the simulation results.

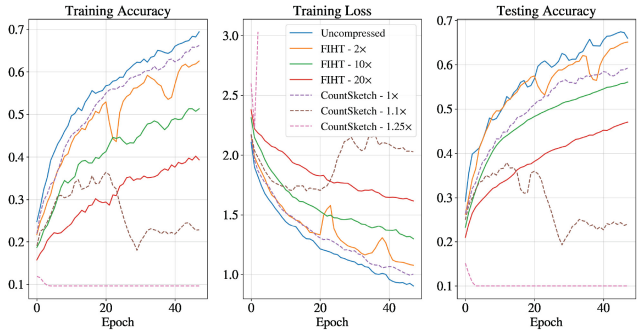


Fig. 2. Results for training on CIFAR10 with i.i.d. datasets.

50 epochs. As the compression rate increases, the convergence of Algorithm 1 deteriorates gradually. As a comparison, SGD with count sketch in our simulation setup diverges when the compression rate is set to be greater than 1.

V. Conclusion

We develop a communication efficient SGD algorithm based on compressed sensing. This algorithm has several direct variants. For example, momentum method can be directly incorporated. Also, when the number of devices n is very large, the server can choose to query compressed stochastic gradients from a random subset of devices.

Our convergence guarantees require $\mathrm{sp}(p(t))$ to be persistently low, which is hard to check in practice. The numerical experiments also show that our algorithm can work even if $\mathrm{sp}(p(t))$ grows to a relatively high level. They suggest that our theoretical analysis can be further improved, which will be an interesting future direction.

Appendix

We first derive an alternative form of the reconstruction error from the condition (7) and the guarantee (6).

Lemma 3: Suppose the conditions in Lemma 2 are satisfied. Let $w \in \mathbb{R}^Q$ be arbitrary, and let $x \in \mathbb{R}^d$ satisfy that $\mathrm{sp}(x)$ is upper bounded by the right-hand side of (7). Then

$$\|\mathcal{A}(\Phi x + w; \Phi) - x\|_2 \le \sqrt{\gamma/2}\|x\|_2 + C_{\mathcal{A},\mathrm{n}}\|w\|_2.$$

Proof: By [28, Lemma 7], we have $\|x - x^{[K]}\|_2 \le \|x\|_1/(2\sqrt{K})$. Therefore by Lemma 2,

$$\begin{aligned}
&\|\mathcal{A}(\Phi x + w; \Phi) - x\|_2 \\
&\le \frac{C_{\mathcal{A},\mathrm{s}} + 1}{2\sqrt{K}}\|x\|_1 + \frac{C_{\mathcal{A},\mathrm{s}}}{\sqrt{K}}\left\|x - x^{[K]}\right\|_1 + C_{\mathcal{A},\mathrm{n}}\|w\|_2 \\
&\le \left[\frac{C_{\mathcal{A},\mathrm{s}} + 1}{2} + C_{\mathcal{A},\mathrm{s}}\left(1 - \frac{K}{d}\right)\right]\frac{\|x\|_1}{\sqrt{K}} + C_{\mathcal{A},\mathrm{n}}\|w\|_2.
\end{aligned}$$

One finishes the proof by (7) and the definition of $\mathrm{sp}(x)$. ∎

Next, we derive a bound on the second moment of $e(t)$.

Lemma 4: We have $\mathbb{E}[\|g(t) - \nabla f(x(t))\|_2^2] \le G^2/n$.

Proof: This follows from Assumption 2 by noting $\mathbb{E}[g_i(t)|x(t)] = \nabla f_i(x(t))$ and that $g_i(t)$ and $g_j(t)$ are independent for $i \ne j$ conditioned on $x(t)$. ∎

Lemma 5: We have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\|e(t)\|_2^2\Big] \le \frac{2\eta^2}{1-\gamma}\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)n} + 2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\sigma^2\right] + \frac{2\eta^2\gamma(1+\gamma)}{(1-\gamma)^2}\cdot\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big].$$

Proof: By definition, we have

$$\begin{aligned}
&\mathbb{E}\Big[\|e(t+1)\|_2^2\Big] \\
&\le \mathbb{E}\left[\Big(\|\Delta(t) - p(t)\|_2 + \frac{\eta Q}{d}\big\|\Phi^\top w(t)\big\|_2\Big)^2\right] \\
&= \mathbb{E}\left[\Big(\|\mathcal{A}(\Phi p(t) + \eta w(t)) - p(t)\|_2 + \eta\sqrt{Q/d}\|w(t)\|_2\Big)^2\right] \\
&\le \mathbb{E}\left[\Big(\sqrt{\gamma/2}\|p(t)\|_2 + \eta\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)\|w(t)\|_2\Big)^2\right] \\
&\le \gamma\mathbb{E}\Big[\|p(t)\|_2^2\Big] + 2\eta^2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\mathbb{E}\Big[\|w(t)\|_2^2\Big] \\
&\le \gamma\mathbb{E}\Big[\|\eta g(t) + e(t)\|_2^2\Big] + 2\eta^2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\sigma^2,
\end{aligned}$$

where the second inequality follows from Lemma 3, and the last inequality follows from the definition of $p(t)$ and the assumption that $\mathbb{E}[\|w(t)\|_2^2] \le \sigma^2$. Notice that

$$\begin{aligned}
&\mathbb{E}\Big[\|\eta g(t) + e(t)\|_2^2\Big] \\
&\le \left(1 + \frac{2\gamma}{1-\gamma}\right)\mathbb{E}\Big[\|\eta g(t)\|_2^2\Big] + \left(1 + \frac{1-\gamma}{2\gamma}\right)\mathbb{E}\Big[\|e(t)\|_2^2\Big],
\end{aligned}$$

which leads to

$$\begin{aligned}
\mathbb{E}\Big[\|e(t+1)\|_2^2\Big] \le \frac{1+\gamma}{2}\mathbb{E}\Big[\|e(t)\|_2^2\Big] + \eta^2\frac{\gamma(1+\gamma)}{1-\gamma}\mathbb{E}\Big[\|g(t)\|_2^2\Big] \\
+ 2\eta^2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\sigma^2.
\end{aligned}$$

By $\mathbb{E}[g(t)|x(t)] = \nabla f(x(t))$ and Lemma 4, we have

$$\begin{aligned}
\mathbb{E}\Big[\|g(t)\|_2^2\Big] &= \mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big] + \mathbb{E}\Big[\|g(t) - \nabla f(x(t))\|_2^2\Big] \\
&\le \mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big] + G^2/n.
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathbb{E}\Big[\|e(t+1)\|_2^2\Big] \le \frac{1+\gamma}{2}\mathbb{E}\Big[\|e(t)\|_2^2\Big] + \eta^2\frac{\gamma(1+\gamma)}{1-\gamma}\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big] \\
+ \eta^2\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)n} + 2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\sigma^2\right].
\end{aligned}$$

By summing over $t = 1, \ldots, T$ and noting that $e(1) = 0$ and $\mathbb{E}[\|e(T+1)\|_2^2] \ge 0$, we get

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\|e(t)\|_2^2\Big] \\
&\le \frac{1+\gamma}{2T}\sum_{t=1}^{T}\mathbb{E}\Big[\|e(t)\|_2^2\Big] + \frac{\eta^2\gamma(1+\gamma)}{(1-\gamma)T}\sum_{t=1}^{T}\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big] \\
&\quad + \eta^2\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)n} + 2\Big(C_{\mathcal{A},\mathrm{n}} + \sqrt{Q/d}\Big)^2\sigma^2\right],
\end{aligned}$$

which then leads to the desired result. ∎

The final step is then to establish the convergence of Algorithm 1. Denote $\tilde{x}(t) = x(t) - e(t)$, and it can be checked that $\tilde{x}(t+1) = \tilde{x}(t) - \eta g(t)$. We then have

$$\begin{aligned}
\|\tilde{x}(t+1) - x^*\|_2^2 = \|\tilde{x}(t) - x^*\|_2^2 + \eta^2\|g(t)\|_2^2 \\
- 2\eta\langle g(t), \tilde{x}(t) - x^*\rangle.
\end{aligned}$$

By taking the expectation and noting $\mathbb{E}[g(t)|x(t)] = \nabla f(x(t))$ and Lemma 4, we get

$$\begin{aligned}&\mathbb{E}\Big[\|\tilde{x}(t+1)-x^*\|_2^2\Big]\\&\quad\le \mathbb{E}\Big[\|\tilde{x}(t)-x^*\|_2^2\Big]+\eta^2\Big(\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big]+G^2/n\Big)\\&\qquad-2\eta\,\mathbb{E}\Big[\langle\nabla f(x(t)),x(t)-x^*\rangle\Big]+2\eta\,\mathbb{E}\Big[\langle\nabla f(x(t)),e(t)\rangle\Big],\end{aligned}$$

and by using $\langle\nabla f(x(t)),e(t)\rangle \le \frac{1}{6L}\|\nabla f(x(t))\|_2^2+\frac{3L}{2}\|e(t)\|_2^2$, we can show that

$$\begin{aligned}&\mathbb{E}\Big[\langle\nabla f(x(t)),x(t)-x^*\rangle\Big]\\&\quad\le\frac{1}{2\eta}\Big(\mathbb{E}\Big[\|\tilde{x}(t)-x^*\|_2^2\Big]-\mathbb{E}\Big[\|\tilde{x}(t+1)-x^*\|_2^2\Big]\Big)+\frac{\eta G^2}{2n}\\&\qquad+\frac{\eta+(3L)^{-1}}{2}\,\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big]+\frac{3L}{2}\mathbb{E}\Big[\|e(t)\|_2^2\Big],\end{aligned}$$

Now, we take the average of both sides over $t=1,\dots,T$ and plug in the bound in Lemma 5 to get

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\langle\nabla f(x(t)),x(t)-x^*\rangle\Big]\\&\quad\le\frac{1}{2\eta T}\|x(1)-x^*\|_2^2+\frac{\eta G^2}{2n}\\&\qquad+\frac{3\eta^2L}{1-\gamma}\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)n}+2\Big(C_{\mathcal{A},\mathrm{n}}+\sqrt{Q/d}\Big)^2\sigma^2\right]\\&\qquad+\left(\frac{\eta+(3L)^{-1}}{2}+\frac{3\eta^2L\gamma(1+\gamma)}{(1-\gamma)^2}\right)\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\|\nabla f(x(t))\|_2^2\Big].\end{aligned}$$

With $\eta = L^{-1}\sqrt{n/T}$, we have that for sufficiently large T,

$$\frac{\eta+(3L)^{-1}}{2}+\frac{3\eta^2L\gamma(1+\gamma)}{(1-\gamma)^2}\le\frac{1}{4L}.$$

Furthermore, by the convexity of f, we have $f(x(t))-f(x^*)\le\langle\nabla f(x(t)),x(t)-x^*\rangle$, and since f is L-smooth, we have $\|\nabla f(x(t))\|_2^2\le 2L(f(x(t))-f(x^*))$. We then get

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[f(x(t))-f(x^*)\Big]\\&\quad\le\frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}\Big[f(x(t))-f(x^*)\Big]+\frac{1}{2\eta T}\|x(1)-x^*\|_2^2+\frac{\eta G^2}{2n}\\&\qquad+\frac{3\eta^2L}{1-\gamma}\left[\frac{\gamma(1+\gamma)G^2}{(1-\gamma)n}+2\Big(C_{\mathcal{A},\mathrm{n}}+\sqrt{Q/d}\Big)^2\sigma^2\right].\end{aligned}$$

By subtracting $\frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}[f(x(t))-f(x^*)]$ from both sides of the inequality, and using $f(\bar{x}(t))\le\frac{1}{T}\sum_{t=1}^{T}f(x(t))$ that follows from the convexity of f, we get the final bound.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th AISTATS*, 2017, pp. 1273–1282.

[2] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Communication complexity of dual decomposition methods for distributed resource allocation optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 717–732, Aug. 2018.

[3] C.-S. Lee, N. Michelusi, and G. Scutari, "Finite rate quantized distributed optimization with geometric convergence," in *Proc. 52nd Asilomar Conf. Signals Syst. Comput.*, 2018, pp. 1876–1880.

[4] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934–4947, Oct. 2019.

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

[6] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," 2018, *arXiv:1808.07576*.

[7] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5693–5700.

[8] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 23, 2020, doi: 10.1109/TPAMI.2020.3033286.

[9] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. 31st NIPS*, 2017, pp. 1707–1718.

[10] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 560–569.

[11] S. Magnússon, H. Shokri-Ghadikolaei, and N. Li, "On maintaining linear convergence of distributed learning and optimization under limited communication," *IEEE Trans. Signal Process.*, vol. 68, pp. 6101–6116, 2020.

[12] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. 32nd NIPS*, 2018, pp. 4452–4463.

[13] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc. 32nd NIPS*, 2018, pp. 5977–5987.

[14] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Vancouver, BC, Canada, 2021, pp. 1–10.

[15] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signSGD and other gradient compression schemes," in *Proc. 36th ICML*, 2019, pp. 3252–3261.

[16] N. Ivkin, D. Rothchild, E. Ullah, V. Braverman, I. Stoica, and R. Arora, "Communication-efficient distributed SGD with sketching," in *Proc. 33rd NeurIPS*, 2019, pp. 13142–13152.

[17] D. Rothchild *et al.*, "FetchSGD: Communication-efficient federated learning with sketching," in *Proc. 37th ICML*, 2020, pp. 8253–8265.

[18] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Proc. 29th ICALP*, Málaga, Spain, 2002, pp. 693–703.

[19] H. Cai, D. Mckenzie, W. Yin, and Z. Zhang, "Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling," 2020, *arXiv:2003.13001*.

[20] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[21] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proc. IEEE*, vol. 98, no. 6, pp. 937–947, Jun. 2010.

[22] S. Foucart, "Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants," in *Approximation Theory XIII: San Antonio 2010*. New York, NY, USA: Springer, 2012, pp. 65–77.

[23] M. E. Lopes, "Unknown sparsity in compressed sensing: Denoising and inference," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5145–5166, Sep. 2016.

[24] I. Haviv and O. Regev, "The restricted isometry property of subsampled Fourier matrices," in *Geometric Aspects of Functional Analysis*. Cham, Switzerland: Springer, 2017, pp. 163–179.

[25] K. Wei, "Fast iterative hard thresholding for compressed sensing," *IEEE Signal Process. Lett.*, vol. 22, no. 5, pp. 593–597, May 2015.

[26] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt, "Practical and optimal LSH for angular distance," in *Proc. 28th NIPS*, vol. 1, 2015, pp. 1225–1233.

[27] Y. Tang, V. Ramanathan, J. Zhang, and N. Li, "Communication-efficient distributed SGD with compressed sensing," 2021, *arXiv:2112.07836*.

[28] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: Fast algorithms for compressed sensing," in *Proc. 39th Annu. ACM Symp. Theory Comput.*, 2007, pp. 237–246.