FedSKETCH: Communication-Efficient Federated Learning via Sketching

Anonymous Author
Anonymous Institution

Abstract

Communication complexity and data privacy are the two key challenges in Federated Learning where the goal is to perform a distributed learning through a large volume of devices. In this work, we introduce two new algorithms, namely `FedSKETCH` and `FedSKETCHGATE`, to address jointly both challenges and which are, respectively, intended to be used for homogeneous and heterogeneous data distribution settings. Our algorithms are based on a key and novel sketching technique, called `HEAPRIX` that is unbiased, compresses the accumulation of local gradients using count sketch, and exhibits communication-efficiency properties leveraging low-dimensional sketches. We provide sharp convergence guarantees of our algorithms and validate our theoretical findings with various sets of experiments.

1 Introduction

Federated Learning (FL) is a recently emerging framework for distributed large scale machine learning problems. In FL, data is distributed across devices [33, 24] and due to privacy concerns, users are only allowed to communicate with the parameter server. Formally, the optimization problem across p distributed devices is defined as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d,\, \sum_{j=1}^{p} q_j=1} f(\boldsymbol{x}) \triangleq \sum_{j=1}^{p} q_j F_j(\boldsymbol{x})\,, \tag{1}$$

where $F_j(\boldsymbol{x}) = \mathbb{E}_{\xi\in\mathcal{D}_j}\left[L_j\left(\boldsymbol{x},\xi\right)\right]$ is the local cost function at device j, $q_j \triangleq \frac{n_j}{n}$, n_j is the number of data shards at device j and $n = \sum_{j=1}^{p} n_j$ is the total number of data samples, ξ is a random variable distributed according to probability distribution $\mathcal{D}_j$, and L_j is a loss function that measures the performance of model $\boldsymbol{x}$ at device j. We note that, while for the homogeneous setting we assume $\{\mathcal{D}_j\}_{j=1}^{p}$ have the same distribution across devices and $L_i = L_j$ for all (i, j), in the heterogeneous setting, these distributions and loss functions L_j can vary from a device to another.

There are several challenges that need to be addressed in FL in order to efficiently learn a global model that performs well in average for all devices:
– *Communication-efficiency*: There are often many devices communicating with the server, thus incurring immense communication overhead. One approach to reduce communication round is using *local SGD with periodic averaging* [47, 40, 46, 42] which periodically averages models after few local updates, contrary to baseline SGD [6] where model averaging is performed at each iteration. Local SGD has been proposed in [33, 24] under the FL setting and its convergence analysis is studied in [40, 42, 47, 46], later on improved in the follow up references [12, 13, 3, 15, 22, 39] for homogeneous setting. It is further extended to heterogeneous setting [45, 30, 37, 31, 15, 21]. Second approach to deal with communication cost aims at reducing the size of communicated message per communication round, such as local gradient quantization [1, 4, 41, 43, 44] or sparsification [2, 32, 38, 39].

–*Data heterogeneity*: Since locally generated data in each device may come from different distribution, local computations involved in Federated Learning setting can lead to poor convergence error in practice [28, 31]. To mitigate the negative impact of data heterogeneity, [14, 17, 31, 21] suggest applying variance reduction or gradient tracking techniques along local computations.

–*Privacy* [11, 16]: Data privacy of users has been widely addressed by injecting an additional layer of

Preliminary work. Under review by AISTATS 2021. Do not distribute.

random noise in order to respect differential-privacy property [34] or using cryptography-based approaches under secure multi-party computation [5].

To tackle all major aforementioned challenges in FL jointly, sketching based algorithms [7, 9, 23, 26] are promising approaches. For instance, to reduce communication cost, [19] develop a distributed SGD algorithm using sketching along providing its convergence analysis in the homogeneous setting, and establish a communication complexity of order $\mathcal{O}(\log(d))$ per round, where d is the dimension of the vector of parameters compared to $\mathcal{O}(d)$ complexity per round of baseline mini-batch SGD. Yet, the proposed sketching scheme in [19], built from a communication-efficiency perspective, is based on a deterministic procedure which requires access to the exact information of the gradients, thus not meeting the crucial privacy-preserving criteria. This systemic flaw is partially addressed in [36].

Focusing on privacy, [27] derive a single framework in order to tackle these issues jointly and introduces `DiffSketch` algorithm, based on the Count Sketch operator, yet does not provide its convergence analysis. Additionally, the estimation error of `DiffSketch` is higher than the sketching scheme in [19] which may end up in poor convergence.

In this paper, we propose new sketching algorithms to address the aforementioned challenges simultaneously. Our main contributions are summarized as:

- We provide a new algorithm – `HEAPRIX` – and theoretically show that it reduces the cost of communication between devices and server, which is based on unbiased sketching without requiring the broadcast of exact values of gradients to the server. Based on `HEAPRIX`, we develop general algorithms for communication-efficient and sketch-based FL, namely `FedSKETCH` and `FedSKETCHGATE` for both homogeneous and heterogeneous data distribution settings respectively.

- We establish non-asymptotic convergence bounds for convex, Polyak-Łojasiewicz and non-convex functions in Theorems 1 and 2 in both homogeneous and heterogeneous cases, and highlight an improvement in the number of iteration to reach a stationary point. We also provide a convergence analysis for the `PRIVIX` algorithm proposed in [27].

- We illustrate benefits of `FedSKETCH` and `FedSKETCHGATE` over baseline methods through a set of experiments. The latter shows the advantages of the `HEAPRIX` compression method achieving comparable test accuracy as Federated SGD (`FedSGD`) while compressing the information exchanged between devices and server.

Notation: We denote the number of communication rounds and bits per round and per device by R and B respectively. The count sketch of any vector $\boldsymbol{x}$ is designated by $\mathbf{S}(\boldsymbol{x})$. $[p]$ denotes the set $\{1, \ldots, p\}$.

2 Compression using Count Sketch

In this paper, we exploit commonly used `Count Sketch` [7] which is described in Algorithm 1.

Algorithm 1 Count Sketch (`CS`) [7]

1: **Inputs:** $\boldsymbol{x} \in \mathbb{R}^d, t, k, \mathbf{S}_{m\times t}, h_j(1 \leq i \leq t), \mathrm{sign}_j(1 \leq i \leq t)$
2: **Compress vector $\boldsymbol{x} \in \mathbb{R}^d$ into $\mathbf{S}(\boldsymbol{x})$:**
3: **for** $\boldsymbol{x}_i \in \boldsymbol{x}$ **do**
4:   **for** $j = 1, \cdots, t$ **do**
5:     $\mathbf{S}[j][h_j(i)] = \mathbf{S}[j-1][h_{j-1}(i)] + \mathrm{sign}_j(i).\boldsymbol{x}_i$
6:   **end for**
7: **end for**
8: **return** $\mathbf{S}_{m\times t}(\boldsymbol{x})$

Count Sketch uses two sets of functions that encode any input vector $\boldsymbol{x}$ **into a hash table** $\boldsymbol{S}_{m\times t}(\boldsymbol{x})$. Pairwise independent hash functions $\{h_{j,1\leq j\leq t} : [d] \to m\}$ are used along with another set of pairwise independent sign hash functions $\{\mathrm{sign}_{j,1\leq j\leq t} : [d] \to \{+1, -1\}\}$ to map entries of $\boldsymbol{x}$ ($\boldsymbol{x}_i$, $1 \leq i \leq d$) into t different columns of $\mathbf{S}_{m\times t}$, wherein to lower the dimension of the input vector we usually have $d \gg mt$. There are various types of sketching algorithms which are developed based on count sketching that we develop in the following subsections.

2.1 Sketching based Unbiased Compressor

We define an unbiased compressor as follows:

Definition 1 (Unbiased compressor)**.** *A randomized function, $C : \mathbb{R}^d \to \mathbb{R}^d$ is called an unbiased compression operator with $\Delta \geq 1$, if we have*

$$\mathbb{E}\left[C(\boldsymbol{x})\right] = \boldsymbol{x} \quad \text{and} \quad \mathbb{E}\left[\|C(\boldsymbol{x})\|_2^2\right] \leq \Delta \|\boldsymbol{x}\|_2^2 .$$

We denote this class of compressors by $\mathbb{U}(\Delta)$.

This definition leads to the following property

$$\mathbb{E}\left[\|\mathrm{C}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2\right] \leq (\Delta - 1) \|\boldsymbol{x}\|_2^2 .$$

Note that if $\Delta = 1$ then our algorithm reduces to the case of no compression. This property allows us to control the noise of the compression.

An instance of such unbiased compressor is `PRIVIX` which obtains an estimate of input $\boldsymbol{x}$ from a count sketch noted $\boldsymbol{S}(\boldsymbol{x})$. In this algorithm, to query the quantity x_i, the $i-th$ element of the vector $\boldsymbol{x}$, we compute the median of t approximated values specified by the indices of $h_j(i)$ for $1 \leq j \leq t$, see [27] or Algorithm 6 in the Appendix (for more details). For the purpose of our proof, we state the following crucial properties of the count sketch:

Property 1 ([27]). *For any $\mathbf{x} \in \mathbb{R}^d$, we have:*

Unbiased estimation*: As in [27], we have:*

$$\mathbb{E}_{\mathbf{S}}\left[\mathit{PRIVIX}[\mathbf{S}(\mathbf{x})]\right] = \mathbf{x}\,.$$

Bounded variance: *if $m = \mathcal{O}\left(\frac{e}{\mu^2}\right)$, $t = \mathcal{O}\left(\ln\left(\frac{d}{\delta}\right)\right)$:*

$$\mathbb{E}_{\mathbf{S}}\left[\left\|\mathit{PRIVIX}[\mathbf{S}(\mathbf{x})] - \mathbf{x}\right\|_2^2\right] \leq \mu^2 d\left\|\mathbf{x}\right\|_2^2,\ w.p.\ 1-\delta\,.$$

Thus, `PRIVIX` $\in \mathbb{U}(1+\mu^2 d)$ with probability $1-\delta$. We note that $\Delta = 1 + \mu^2 d$ implies that if $m \to d$, $\Delta \to 1 + 1 = 2$, indicating that the case of no compression is not covered.

Remark 1. *As shown in [27], if the data is normally distributed, PRIVIX is differential private [10], up to additional assumptions and algorithmic design.*

2.2 Sketching based Biased Compressor

A biased compressor is defined as follows:

Definition 2 (Biased compressor)**.** *A (randomized) function, $C : \mathbb{R}^d \to \mathbb{R}^d$ belongs to $\mathbb{C}(\Delta, \alpha)$, a class of compression operators with $\alpha > 0$ and $\Delta \geq 1$, if*

$$\mathbb{E}\left[\left\|\alpha\boldsymbol{x} - C(\boldsymbol{x})\right\|_2^2\right] \leq \left(1 - \frac{1}{\Delta}\right)\left\|\boldsymbol{x}\right\|_2^2,$$

The reference [18] proves that $\mathbb{U}(\Delta) \subset \mathbb{C}(\Delta, \alpha)$. An instance of a biased compression method via sketching and using top_m operation is given in Algorithm 2:

Algorithm 2 `HEAVYMIX`

1: **Inputs:** $\mathbf{S}(\mathbf{g})$; parameter m
2: **Query the vector $\tilde{\mathbf{g}} \in \mathbb{R}^d$ from $\mathbf{S}(\mathbf{g})$:**
3: Query $\hat{\ell}_2^2 = (1 \pm 0.5)\left\|\mathbf{g}\right\|^2$ from sketch $\mathbf{S}(\mathbf{g})$
4: $\forall j$ query $\hat{\mathbf{g}}_j^2 = \hat{\mathbf{g}}_j^2 \pm \frac{1}{2m}\left\|\mathbf{g}\right\|^2$ from sketch $\mathbf{S}_{\mathbf{g}}$
5: $H = \{j | \hat{\mathbf{g}}_j \geq \frac{\hat{\ell}_2^2}{m}\}$ and $NH = \{j | \hat{\mathbf{g}}_j < \frac{\hat{\ell}_2^2}{m}\}$
6: $\text{Top}_m = H \cup \text{rand}_\ell(NH)$, where $\ell = m - |H|$
7: Get exact values of Top_m
8: **Output:** $\tilde{\mathbf{g}} : \forall j \in \text{Top}_m : \tilde{\mathbf{g}}_i = \mathbf{g}_i$ else $\mathbf{g}_i = 0$

Following [19], `HEAVYMIX` with sketch size $\Theta\left(m \log\left(\frac{d}{\delta}\right)\right)$ is a biased compressor with $\alpha = 1$ and $\Delta = d/m$ with probability $\geq 1 - \delta$. In other words, with probability $1-\delta$, `HEAVYMIX` $\in C(\frac{d}{m}, 1)$. We note that Algorithm 2 is a variation of the sketching algorithm developed in [19] with distinction that `HEAVYMIX` does not require a second round of communication to obtain the exact values of top_m. Additionally, while a sketching algorithm implementing `HEAVYMIX` has smaller estimation error compared to `PRIVIX`, it requires having access to the exact values of top_m, therefore not benefiting from privacy properties contrary to `PRIVIX`. In the following we introduce our sketching scheme – `HEAPRIX` – as a combination of those two methods.

2.3 Sketching based Induced Compressor

Due to Theorem 3 in [18], which illustrates that we can convert the biased compressor into an unbiased one such that, for $C_1 \in \mathbb{C}(\Delta_1)$ with $\alpha = 1$, if you choose $C_2 \in \mathbb{U}(\Delta_2)$, then induced compressor $C : x \mapsto C_1(\mathbf{x}) + C_2\left(x - C_1\left(\mathbf{x}\right)\right)$ belongs to $\mathbb{U}(\Delta)$ with $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1}$. Based on this notion, Algorithm 3 proposes an induced sketching algorithm by utilizing `HEAVYMIX` and `PRIVIX` for C_1 and C_2 respectively where the reconstruction of input $\mathbf{x}$ is performed using hash table $\mathbf{S}$ and $\mathbf{x}$, similar to `PRIVIX` and `HEAVYMIX`.

Algorithm 3 `HEAPRIX`

1: **Inputs:** $\boldsymbol{x} \in \mathbb{R}^d, t, m, \mathbf{S}_{m\times t}, h_j(1 \leq i \leq t), \text{sign}_j(1 \leq i \leq t)$, parameter m
2: **Approximate $\mathbf{S}(x)$ using** `HEAVYMIX`
3: **Approximate $\mathbf{S}\left(x - \texttt{HEAVYMIX}[\mathbf{S}(x)]\right)$ using** `PRIVIX`
4: **Output:** $\texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right] + \texttt{PRIVIX}\left[\mathbf{S}\left(\mathbf{x} - \texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]\right)\right]$

We note if $m \to d$, then $C(x) \to x$, meaning that the algorithm convergence can be improved by decreasing the noise of compression m.

Corollary 1. *Based on [18, Theorem 3], HEAPRIX in Algorithm 3 satisfies $C(x) \in \mathbb{U}(\mu^2 d)$.*

Benefits of HEAPRIX: Corollary 1 states that, unlike `PRIVIX`, `HEAPRIX` compression noise can be made as small as possible using larger hash size. Contrary to `HEAVYMIX`, `HEAPRIX` does not require having access to exact top_m values of the input, thus helps preserving privacy. In other words, `HEAPRIX` leverages the best of both worlds: the *unbiasedness* of `PRIVIX` while using *heavy hitters* as in `HEAVYMIX`.

3 FedSKETCH and FedSKETCHGATE

In this section, we define two general frameworks for different sketching algorithms for homogeneous and heterogeneous settings.

3.1 Homogeneous Setting

In FedSKETCH, the number of local updates, between two consecutive communication rounds, at device j is denoted by τ. Unlike [14], server node does not store any global model, instead device j has two models, $\boldsymbol{x}^{(r)}$ and $\boldsymbol{x}_j^{(\ell,r)}$, respectively local and global models. We develop FedSKETCH in Algorithm 4.

Algorithm 4 FedSKETCH(R, τ, η, γ)

1: **Inputs:** $\boldsymbol{x}^{(0)}$: initial model shared by all local devices, global and local learning rates γ and η, respectively
2: **for** $r = 0, \ldots, R-1$ **do**
3: **parallel for device** $j \in \mathcal{K}^{(r)}$ **do**:
4: **if PRIVIX variant:**

$$\mathbf{\Phi}^{(r)} \triangleq \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)}\right]$$

5: **if HEAPRIX variant:**

$$\mathbf{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$$

6: Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \mathbf{\Phi}^{(r)}$ and $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
7: **for** $\ell = 0, \ldots, \tau - 1$ **do**
8: Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)}$
9: Update $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\, \tilde{\mathbf{g}}_j^{(\ell,r)}$
10: **end for**
11: Device j broadcasts $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$.
12: Server **computes** $\mathbf{S}^{(r)} = \frac{1}{k}\sum_{j\in\mathcal{K}} \mathbf{S}_j^{(r)}$.
13: Server **broadcasts** $\mathbf{S}^{(r)}$ to devices in randomly drawn devices $\mathcal{K}^{(r)}$.
14: **if HEAPRIX variant:**
15: Second round of communication: $\delta_j^{(r)} := \mathbf{S}_j\left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)})\right]$ and broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{k}\sum_{j\in\mathcal{K}} \delta_j^{(r)}$ to devices in set $\mathcal{K}^{(r)}$
16: **end parallel for**
17: **end**
18: **Output:** $\boldsymbol{x}^{(R-1)}$

A variant of this algorithm implementing HEAPRIX is also described in Algorithm 4. We note that for this variant, we need to have an additional communication round between server and worker j to aggregate $\delta_j^{(r)} \triangleq \mathbf{S}_j\left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)})\right]$, see Lines 5 and 12. The main difference between our FedSKETCH and the DiffSketch algorithm in [27] is that we use distinct local and global learning rates. Additionally, unlike [27], we do not add local Gaussian noise.

Comparison with [14] An important feature of our algorithm is that due to a lower dimension of the count sketch, the resulting averages ($\mathbf{S}^{(r)}$ and $\tilde{\mathbf{S}}^{(r)}$) received by the server, are also of lower dimension. Therefore, these algorithms exploit a bidirectional compression during the communication from server to device back and forth. As a result, due to this bidirectional property of communicating sketching for the case of large quantization error $\omega = \theta(\frac{d}{m})$ as shown in [14], our algorithms can outperform FedCOM and FedCOMGATE developed in [14] if sufficiently large hash tables are used and the uplink communication cost is high. Furthermore, while, in [14], server stores a global model and aggregates the partial gradients from devices which can enable the server to extract some information regarding the device's data, in contrast, in our algorithms server does not store the global model and only broadcasts the average sketches. Thus, sketching-based server-devices communication algorithms such as ours does not reveal the exact values of the inputs (to preserve privacy) as a by-product.

3.2 Heterogeneous Setting

In this section, we focus on the optimization problem of (1) in the special case of $q_1 = \ldots = q_p = \frac{1}{p}$ with full device participation ($k = p$). These results can be extended to the scenario where devices are sampled. For non i.i.d. data, the FedSKETCH algorithm, designed for homogeneous setting, may fail to perform well in practice. The main reason is that in FL, devices are using local stochastic descent direction which could be different than global descent direction when the data distribution are non-identical. Therefore, to mitigate the effect of data heterogeneity, we introduce a new algorithm called FedSKETCHGATE described in Algorithm 5.

This algorithm leverages the idea of gradient tracking applied in [14] (with compression) and a special case of $\gamma = 1$ without compression [31]. The main idea is that using an approximation of global gradient, $\mathbf{c}_j^{(r)}$ allows to correct the local gradient direction. For the FedSKETCHGATE with PRIVIX variant, the correction vector $\mathbf{c}_j^{(r)}$ at device j and communication round r is computed in Line 4. While using HEAPRIX compression method, FedSKETCHGATE also updates $\tilde{\mathbf{S}}^{(r)}$ via Line 16.

Algorithm 5 $\texttt{FedSKETCHGATE}(R,\, \tau, \eta, \gamma)$

Inputs: $\boldsymbol{x}^{(0)} = \boldsymbol{x}_j^{(0)}$ shared by all local devices, global and local learning rates γ and η.
for $r = 0, \ldots, R-1$ **do**
parallel for device $j = 1, \ldots, p$ **do**:
if PRIVIX variant:

$$\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{\texttt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \texttt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)}{\tau}$$

where $\boldsymbol{\Phi}^{(r)} \triangleq \texttt{PRIVIX}(\mathbf{S}^{(r-1)})$
if HEAPRIX variant:

$$\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\boldsymbol{\Phi}^{(r)} - \boldsymbol{\Phi}_j^{(r)}\right)$$

Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\boldsymbol{\Phi}^{(r)}$ and $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
for $\ell = 0, \ldots, \tau-1$ **do**
Sample mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)}$
$\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\left(\tilde{\mathbf{g}}_j^{(\ell,r)} - \mathbf{c}_j^{(r)}\right)$
end for
Device j broadcasts $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$.
Server **computes** $\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1}\mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
if HEAPRIX variant:
Device j computes $\boldsymbol{\Phi}_j^{(r)} \triangleq \texttt{HEAPRIX}[\mathbf{S}_j^{(r)}]$
Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left(\texttt{HEAVYMIX}[\mathbf{S}^{(r)}]\right)$
Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p}\delta_j^{(r)}$ to devices
end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

4 Convergence Analysis

We first state common assumptions needed in the following convergence analysis (reminder of our notations can be found Table 1 of the Appendix).

Assumption 1 (Smoothness and Lower Boundedness). *The local objective function $f_j(\cdot)$ of jth device is differentiable for $j \in [p]$ and L-smooth, i.e., $\|\nabla f_j(\boldsymbol{x}) - \nabla f_j(\mathbf{y})\| \leq L\|\boldsymbol{x} - \mathbf{y}\|, \forall\, \boldsymbol{x}, \mathbf{y} \in \mathbb{R}^d$. Moreover, the optimal objective function $f(\cdot)$ is bounded below by $f^* = \min_{\boldsymbol{x}} f(\boldsymbol{x}) > -\infty$.*

Assumption 2 (Polyak-Lojasiewicz). *A function $f(\boldsymbol{x})$ satisfies the Polyak-Lojasiewicz(PL) condition with constant μ if $\frac{1}{2}\|\nabla f(\boldsymbol{x})\|_2^2 \geq \mu\big(f(\boldsymbol{x}) - f(\boldsymbol{x}^*)\big)$, $\forall \boldsymbol{x} \in \mathbb{R}^d$ with $\boldsymbol{x}^*$ is an optimal solution.*

Assumption 1 is common in stochastic optimization. It is shown in [20] that PL condition implies strong convexity property with same module (PL objectives can also be non-convex, hence strong convexity does not imply PL condition necessarily).

4.1 Convergence of FEDSKETCH

We now focus on the homogeneous case where data is i.i.d. among local devices, and therefore , the stochastic local gradient of each worker is an unbiased estimator of the global gradient. We have:

Assumption 3 (Bounded Variance). *For all $j \in [m]$, we can sample an independent mini-batch ℓ_j of size $|\Xi_j^{(\ell,r)}| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w}; \Xi_j)$, $\mathbb{E}_{\xi_j}[\tilde{\mathbf{g}}_j] = \nabla f(\boldsymbol{w}) = \mathbf{g}$ with the variance bounded is bounded by a constant σ^2, i.e., $\mathbb{E}_{\Xi_j}\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}\|^2\right] \leq \sigma^2$.*

Theorem 1. *Suppose Assumptions 1-3 hold. Given $0 < m = O\left(\frac{e}{\mu^2}\right) \leq d$ and considering Algorithm 4 with sketch size $B = O\left(m\log\left(\frac{dR}{\delta}\right)\right)$ and $\gamma \geq k$, with probability $1-\delta$ we have:*

*In the **non-convex** case, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ if:*

- *FS-PRIVIX, for $\eta = \frac{1}{L\gamma}\sqrt{\frac{k}{R\tau\left(\frac{\mu^2 d}{k}+1\right)}}$:*

$$R = O\left(1/\epsilon\right) \quad \text{and} \quad \tau = O\left((\mu^2 d + 1)/(k\epsilon)\right)$$

- *FS-HEAPRIX, for $\eta = \frac{1}{L\gamma}\sqrt{\frac{k}{R\tau\left(\frac{\mu^2 d-1}{k}+1\right)}}$:*

$$R = O\left(1/\epsilon\right) \quad \text{and} \quad \tau = O\left(\mu^2 d/(k\epsilon)\right)$$

*In the **PL or strongly convex** case, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\mathbb{E}[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})] \leq \epsilon$ if we set:*

- *FS-PRIVIX, for $\eta = 1/(2L(\mu^2 d/k + 1)\tau\gamma)$:*

$$R = O\left(\left(\mu^2 d/k + 1\right)\kappa\log\left(1/\epsilon\right)\right)$$
$$\tau = O\left((\mu^2 d + 1)/k\left(\mu^2 d/k + 1\right)\epsilon\right)$$

- *FS-HEAPRIX, for $\eta = 1/(2L((\mu^2 d - 1)/k + 1)\tau\gamma)$:*

$$R = O\left(\left((\mu^2 d - 1)/k + 1\right)\kappa\log\left(1/\epsilon\right)\right)$$
$$\tau = O\left(\mu^2 d/(k\left((\mu^2 d - 1)/k + 1\right)\epsilon)\right)$$

*In the **Convex** case, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ if we set:*

- *FS-PRIVIX, for $\eta = 1/(2L(\mu^2 d/k + 1)\tau\gamma)$:*

$$R = O\left(L\left(1 + \mu^2 d/k\right)/\epsilon\log\left(1/\epsilon\right)\right)$$
$$\tau = O\left(\left(\mu^2 d + 1\right)^2/(k\left(\mu^2 d/k + 1\right)^2\epsilon^2)\right)$$

- *FS-HEAPRIX, for* $\eta = 1/(2L((\mu^2 d-1)/k+1)\tau\gamma)$*:*

$$R = O\left(L\left(1+(\mu^2 d-1)/k\right)/\epsilon\log\left(1/\epsilon\right)\right)$$
$$\tau = O\left(\left(\mu^2 d\right)^2/(k\left((\mu^2 d-1)/k+1\right)^2\epsilon^2)\right)$$

Remark 2. *Most of the existing communication-efficient algorithms with compression only consider communication-efficiency from devices to server. However, Algorithm 4 also improves the communication efficiency from server to devices since it exploits low-dimensional sketches (and averages), communicated from the server to devices.*

Comparison with [19] From privacy aspect, we note [19] requires for server to have access to exact values of top_m gradients, hence do not preserve privacy, whereas our schemes do not need those exact values. From communication cost point of view, for strongly convex objective and compared to [19], we improve the total communication per worker from $RB = O\left(\frac{\mu^2 d}{\epsilon}m\log\left(\frac{d}{\delta\sqrt{\epsilon}}\max\left(\mu^2 d, \frac{1}{\sqrt{\epsilon}}\right)\right)\right)$ to

$RB = O\left(m\kappa(\frac{\mu^2 d-1}{k}+1)\log\frac{1}{\epsilon}\log\left(\frac{\kappa d}{\delta}(\frac{\mu^2 d-1}{k}+1)\log\frac{1}{\epsilon}\right)\right)$

We note that while reducing communication cost, our scheme requires $\tau = O(\mu^2 d/(k(\frac{\mu^2 d}{k}+1)\epsilon)) > 1$. Yet, it scales down with the number of sampled devices, k. Moreover, unlike [19], we do not use bounded gradient assumption. Therefore, we obtain stronger result with weaker assumptions. Regarding general non-convex objectives, our result improves the total communication cost per worker in [19] from $RB = O\left(\max(\frac{1}{\epsilon^2}, \frac{d^2}{k^2\epsilon}\right)\log(\frac{d}{\delta}\max(\frac{1}{\epsilon^2}, \frac{d^2}{k^2\epsilon})))$ for *only one device* to $RB = O(\frac{m}{\epsilon}\log(\frac{d}{\epsilon\delta}))$. We also highlight that we can obtain similar rates for Algorithm 4 in heterogeneous environment if we make the additional assumption of uniformly bounded gradient.

Note: Such improved communication cost over prior related works is due to both the exploitation of *sketching*, to reduce the dimension of broadcast messages, and the use of *local updates*, to reduce the total number of communication rounds leading to a specific convergence error.

4.2 Convergence of `FedSKETCHGATE`

We start with bounded local variance assumption:

Assumption 4 (Bounded Local Variance). *For all $j \in [p]$, we can sample an independent mini-batch Ξ_j of size $|\xi_j| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w};\Xi_j)$ with $\mathbb{E}_\xi[\tilde{\mathbf{g}}_j] = \nabla f_j(\boldsymbol{w}) = \mathbf{g}_j$. Moreover, the variance of local stochastic gradients is bounded such that $\mathbb{E}_\Xi\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}_j\|^2\right] \leq \sigma^2$.*

Theorem 2. *Suppose Assumptions 1 and 4 hold. Given $0 < m = O\left(\frac{e}{\mu^2}\right) \leq d$, and considering* `FedSKETCHGATE` *in Algorithm 5 with sketch size $B = O\left(m\log\left(\frac{dR}{\delta}\right)\right)$ and $\gamma \geq p$ with probability $1-\delta$ we have*

In the ***non-convex*** *case, $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\mu^2 d)}}$, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ if:*

- *FS-PRIVIX:*

$$R = O((\mu^2 d+1)/\epsilon) \quad \text{and} \quad \tau = O(1/(p\epsilon))$$

- *FS-HEAPRIX:*

$$R = O(\mu^2 d/\epsilon) \quad \text{and} \quad \tau = O(1/(p\epsilon))$$

In the ***PL or Strongly convex*** *case, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ if:*

- *FS-PRIVIX, for* $\eta = 1/(2L(\mu^2 d+1)\tau\gamma)$*:*

$$R = O\left((\mu^2 d+1)\kappa\log(1/\epsilon)\right) \quad \text{and} \quad \tau = O\left(1/(p\epsilon)\right)$$

- *FS-HEAPRIX, for* $\eta = 1/(2L\mu^2 d\tau\gamma)$*:*

$$R = O\left((\mu^2 d)\kappa\log(1/\epsilon)\right) \quad \text{and} \quad \tau = O\left(1/(p\epsilon)\right)$$

In the ***convex*** *case, $\{\boldsymbol{w}^{(r)}\}_{r=>0}$ satisfies $\mathbb{E}[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})] \leq \epsilon$ if:*

- *FS-PRIVIX, for* $\eta = 1/(2L(\mu^2 d+1)\tau\gamma)$*:*

$$R = O\left(L(\mu^2 d+1)\epsilon\log(1/\epsilon)\right) \quad \text{and} \quad \tau = O\left(1/(p\epsilon^2)\right)$$

- *FS-HEAPRIX, for* $\eta = 1/(2L\mu^2 d\tau\gamma)$*:*

$$R = O\left(L(\mu^2 d)\epsilon\log(1/\epsilon)\right) \quad \text{and} \quad \tau = O\left(1/(p\epsilon^2)\right)$$

These results are summarized in Table 2-3 in Section B.1 of the Appendix.

4.3 Comparison with Prior Methods

We now compare our results with prior works:

Comparison with [27]. Note that our convergence analysis does not rely on the bounded gradient assumption. We also improve both the number of communication rounds R and the size of transmitted bits B per communication round. Additionally, we highlight that, while [27] provides a convergence analysis for convex objectives, our analysis holds for PL (thus strongly convex case), general convex and general non-convex objectives.

Comparison with [36]. Due to gradient tracking, our algorithm tackles data heterogeneity issue, while algorithms in [36] does not particularly. As a consequence, in `FedSKETCHGATE` each device has to store an additional state vector compared to [36]. Yet, unlike [36], as our method is built upon an unbiased compressor, server does not need to store any additional error correction vector. The convergence results for both of two variants of `FetchSGD` in [36] rely on the uniform bounded gradient assumption which may not be applicable with L-smoothness assumption when data distribution is highly heterogeneous, as in FL, see [22], while our bounds do not assume such boundedness. Besides, Theorem 1 [36] supposes that *Contraction Holds* for the sequence of gradients which may not hold in practice, yet based on this strong assumption their total communication cost (RB) to achieve $\epsilon-$ error is $BR = O\left(m\max(\frac{1}{\epsilon^2}, \frac{d^2-dm}{m^2\epsilon})\log\left(\frac{d}{\delta}\max(\frac{1}{\epsilon^2}, \frac{d^2-dm}{m^2\epsilon})\right)\right)$. Note that for the sake of comparison we let the compression ratio in [36] to be $\frac{m}{d}$. In contrast, without any extra assumptions, our results in Theorem 2 for `PRIVIX` and `HEAPRIX` are respectively $BR = O(\frac{m(\mu^2 d+1)}{\epsilon}\log(\frac{\mu^2 d^2+d}{\epsilon\delta}\log(\frac{1}{\epsilon})))$ and $BR = O(\frac{m(\mu^2 d)}{\epsilon}\log(\frac{\mu^2 d^2}{\epsilon\delta}\log(\frac{1}{\epsilon})))$ which improves the total communication cost of Theorem 1 in [36] under regimes such that $\frac{1}{\epsilon} \geq d$ or $d >> m$. Theorem 2 in [36] is based on the assumption of *Sliding Window Heavy Hitters*, which is similar to the gradient diversity assumption in [29, 15]. They show that, under such assumption, the total communication cost is $BR = O\left(\frac{m\max(I^{2/3}, 2-\alpha)}{\epsilon^3\alpha}\log\left(\frac{d\max(I^{2/3}, 2-\alpha)}{\epsilon^3\delta}\right)\right)$ where I is a constant linked to the window of gradients assumption. Our result improves the latter bound with weaker assumptions in a regime where $\frac{I^{2/3}}{\epsilon^2} \geq d$. We also provide bounds for PL, convex and non-convex objectives unlike [36].

5 Numerical Applications

In this section, we provide empirical results on MNIST dataset to demonstrate the effectiveness of our proposed algorithms. We train LeNet-5 Convolutional Neural Network (CNN) architecture introduced in [25], with 60 000 parameters. We compare Federated SGD (FedSGD), SketchSGD [19], FedSketch-PRIVIX (FS-PRIVIX) and FedSketch-HEAPRIX (FS-HEAPRIX). Note that in Algorithm 4, FS-PRIVIX with global learning rate $\gamma = 1$ is equivalent to the DiffSketch algorithm proposed in [29]. The number of workers is set to 50 and the number of local updates τ is varying for FL methods. For SketchedSGD, which is under synchronous distributed framework, $\tau = 1$. We tune the learning rates (both local, i.e. η and global, i.e. γ, if applicable) over log-scale and report the best results. At each round, we randomly choose half of the devices to be active, as commonly done in real-world applications. Numerical results are reported for both *homogeneous* and *heterogeneous* settings. In the former case, each device receives uniformly drawn data samples and in the latter, it receives samples from one or two classes among ten.

Homogeneous case. In Figure 1, we provide the training loss and test accuracy for the four algorithms mentioned above, with $\tau = 1$ (recall $\tau = 1$ for SketchSGD). We also test different sizes of sketches, $(t, k) = (20, 40)$ and $(50, 100)$. Note that these two choices of sketch sizes correspond to a $75\times$ and $12\times$ compression ratio, respectively. In general, higher compression ratio leads to worse learning performance. In both cases, FS-HEAPRIX performs the best in terms of both training objective and test accuracy. FS-PRIVIX is superior when sketch size is large, while SketchSGD performs better with small sketch size. Results for multiple local updates $\tau = 5$ are presented Figure 1 ($\tau = 2$ is deferred to the Appendix). FS-HEAPRIX is significantly better than FS-PRIVIX, either with small or large sketches. FS-HEAPRIX yields acceptable extra test error compared to FedSGD, especially when considering the high compression ratio (e.g. $75\times$). However, FS-PRIVIX performs poorly with small sketch size $(20, 40)$, and even diverges with $\tau = 5$. We also observe that the performances of FS-HEAPRIX improve when the number of local updates increases. *That is, the proposed method is able to further reduce the communication cost by reducing the number of rounds required for communication.* This is also consistent with our theoretical findings. For $\tau = 1, 5$, we see that a sketch size of $(50, 100)$ is sufficient to give similar test accuracy as the FedSGD.

Heterogeneous case. We plot similar results in Figure 2 for non-i.i.d. data distribution. This setting leads to more twists and turns in the training curves. From the first column ($\tau = 1$), we see that SketchSGD performs very poorly in the heterogeneous case, while both our proposed FedSketchGATE methods, see Algorithm 5, achieve similar generalization accuracy as the FedSGD algorithm, even with small sketch size (i.e. $75\times$ compression ratio). The slow convergence of federated SGD in non-i.i.d. data distribution case has also been reported in [33, 8]. In addition, FS-HEAPRIX ourperforms FS-PRIVIX in terms of training loss and test accuracy.

We also notice Figure 2 the advantage of FS-

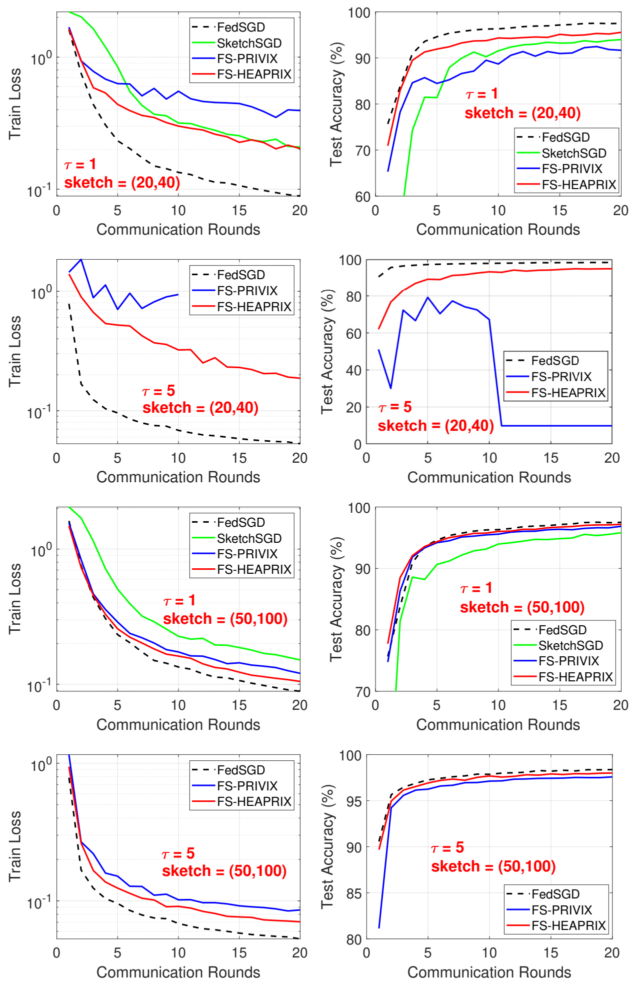

Figure 1 Homogeneous case: Comparison of compressed optimization methods on LeNet CNN.

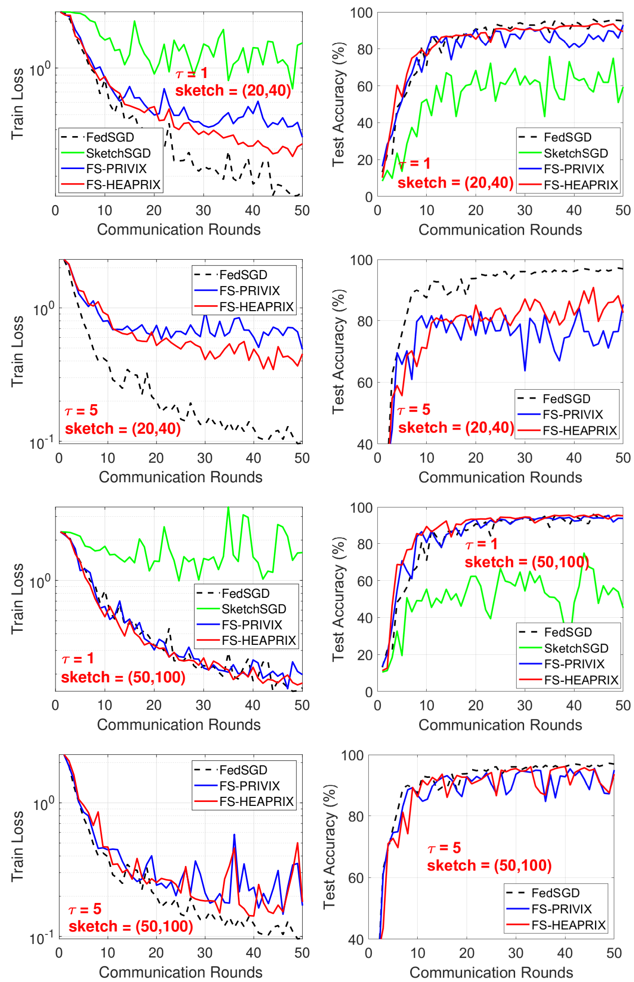

Figure 2 Heterogeneous case: Comparison of compressed optimization algorithms on LeNet CNN.

HEAPRIX over FS-PRIVIX with multiple local updates. However, empirically we see that in the heterogeneous setting, more local updates τ tend to undermine the learning performance, especially with small sketch size. When the sketch size is large, i.e. $(50, 100)$, FS-HEAPRIX can still provide comparable test accuracy as FedSGD with $\tau = 5$. Our empirical study demonstrates that our proposed FedSketch (and FedSketchGATE) frameworks are able to perform well in homogeneous (resp. heterogeneous) setting, with high compression rate. In particular, FedSketch methods are advantageous over prior SketchedSGD [19] method in both cases. FS-HEAPRIX performs the best among all the tested compressed optimization algorithms, which in many cases achieves similar generalization accuracy as Federated SGD with small sketch size.

6 Conclusion

In this paper, we introduced `FedSKETCH` and `FedSKETCHGATE` algorithms for homogeneous and heterogeneous data distribution setting respectively for Federated Learning wherein communication between server and devices is only performed using count sketch. Our algorithms, thus, provide communication-efficiency and privacy, through random hashes based sketches. We analyze the convergence error for *non-convex*, *Polyak-Lojasiewicz* and *general convex* objective functions in the scope of Federated Optimization. We provide insightful numerical experiments showcasing the advantages of our `FedSKETCH` and `FedSKETCHGATE` methods over current federated optimization algorithm. The proposed algorithms outperform competing compression method and can achieve comparable test accuracy as Federated SGD, with high compression ratio.

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1709–1720, Long Beach, 2017.

[2] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5973–5983, Montréal, Canada, 2018.

[3] Debraj Basu, Deepesh Data, Can Karakus, and Suhas N. Diggavi. Qsparse-local-sgd: Distributed SGD with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 14668–14679, Vancouver, Canada, 2019.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SIGNSGD: compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 559–568, Stockholmsmässan, Stockholm, Sweden, 2018.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191, Dallas, TX, 2017.

[6] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168, Vancouver, Canada, 2008.

[7] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[8] Xiangyi Chen, Xiaoyun Li, and Ping Li. Toward communication efficient adaptive gradient method. In *ACM-IMS Foundations of Data Science Conference (FODS)*, Seattle, WA, 2020.

[9] Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[10] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[11] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[12] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. pages 11080–11092, Vancouver, Canada, 2019.

[13] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck R. Cadambe. Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2545–2554, Long Beach, CA, 2019.

[14] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. *arXiv preprint arXiv:2007.01154*, 2020.

[15] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[16] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

[17] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[18] Samuel Horváth and Peter Richtárik. A better alternative to error feedback for communication-efficient distributed learning. *arXiv preprint arXiv:2006.11077*, 2020.

[19] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed

SGD with sketching. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13144–13154, Vancouver, Canada, 2019.

[20] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 795–811, Riva del Garda, Italy, 2016.

[21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[22] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4519–4529, Online [Palermo, Sicily, Italy], 2020.

[23] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[24] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[26] Ping Li, Kenneth Ward Church, and Trevor Hastie. One sketch for all: Theory and application of conditional random sampling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 953–960, Vancouver, Canada, 2008.

[27] Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. Privacy for free: Communication-efficient learning with differential privacy using sketches. *arXiv preprint arXiv:1911.00972*, 2019.

[28] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.*, 37(3):50–60, 2020.

[29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, Austin, TX, 2020.

[30] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

[31] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[32] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, Fort Lauderdale, FL, 2017.

[34] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

[35] Constantin Philippenko and Aymeric Dieuleveut. Artemis: tight convergence guarantees for bidirectional compression in federated learning. *arXiv preprint arXiv:2006.14591*, 2020.

[36] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. FetchSGD: Communication-efficient federated learning with sketching. *arXiv preprint arXiv:2007.07682*, 2020.

[37] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[38] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4447–4458, Montréal, Canada, 2018.

[39] Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

[40] Sebastian Urban Stich. Local sgd converges fast and communicates little. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019.

[41] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7652–7662, Montréal, Canada, 2018.

[42] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[43] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems (NIPS)*, pages 1509–1519, Long Beach, CA, 2017.

[44] Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized sgd and its applications to large-scale distributed optimization. *arXiv preprint arXiv:1806.08054*, 2018.

[45] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed nonconvex optimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7184–7193, Long Beach, CA, 2019.

[46] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 5693–5700, Honolulu, HI, 2019.

[47] Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3219–3227, Stockholm, Sweden, 2018.