Recent Advances in Non-convex Private Communication-Efficient Federated Optimization

Igor Sokolov
KAUST

December 11, 2022

Abstract

In various applications of machine learning, non-convex optimization problems appear. These problems have been the subject of considerable attention in recent years due to their practical importance and have led to the rapid development of new efficient stochastic gradient methods. In order to improve the generalization performance of modern deep learning models, practitioners resort to the use of ever larger data sets in training processes, which are naturally distributed to many edge devices. However, with the increase in training data, the computational cost of gradient-type methods increases considerably. Furthermore, distributed methods are almost always prone to such a bottleneck of communication: the cost of communicating the information needed for workers to solve the problem jointly is often very high, and the cost of calculation can be several orders of its magnitude. On the other hand, the protection of privacy, particularly at the client level, is another important requirement to be dealt with simultaneously in the presence of advanced communications compression techniques. In this project, we provide a comprehensive overview of the current state of development of differentially private communication-efficient methods for non-convex federated optimization problems. In particular, we provide a comprehensive characterization of their performance-comparativeitiesin terms of privacy, utility, and communication complexity.

Contents

1 Introduction **1**

2 Preliminaries **4**
2.1 Privacy 4

3 Main part **6**
3.1 CDP-SGD 6
3.2 SoteriaFL 7
3.3 Some Algorithms within SoteriaFL Framework 8

1 Introduction

Importance of stochastic first-order methods for non-convex objectives In this section, we introduce the basic concepts of supervised federated machine learning from the optimization point of view. Over the past few years, deep neural networks [48] have been used to solve many practical problems [26] such as adaptive testing, biological image classification, computer vision, cancer detection, natural language processing, object detection, facial recognition, writing and pronunciation recognition, stock market analysis, smart cities, etc. Training a deep learning model usually relates to finding a stationary point of the non-convex optimization problem

$$\min_{x\in\mathbb{R}^d} f(x), \tag{1}$$

where d is the number of trainable parameters.

Currently, the most efficient training algorithms [15, 76, 30, 25] are based on an iterative first-order gradient-type update. Specifically, these methods can be written in the form

$$x^{t+1} = x^t - \gamma_t g^t, \tag{2}$$

where $x^t \in \mathbb{R}^d$ is the model parameter vector, and g^t is possibly biased gradient estimator [1] at step t, and γ_t is the appropriately chosen stepsize.

Since in the non-convex setting, finding the global minima of (1) is intractable [105, 135], we limit ourselves to the search of the ε-stationary point, i.e., we wish to find a random vector $\hat{x} \in \mathbb{R}^d$ for which $\mathbb{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \varepsilon^2$, where the expectation $\mathbb{E}[\cdot]$ is taken over all randomness of the algorithm.

Importance of distributed training In recent years, practitioners have used ever-larger data sets [134] distributed across multiple devices in training processes and are focusing on improving the practical generalization performance of modern deep learning models. This is the case for Federated Learning [79, 78, 98, 97] (see Fig. 1 for network topology clarification).

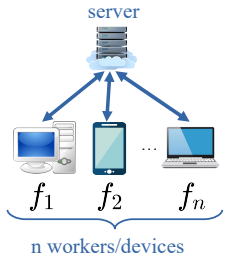

Figure 1: Topology of the data transfers between devices and server. Source [49].

This setting stores training data and uses it distributedly. This approach is the opposite of transferring all data to a central place (such as a data center) and processing them afterward. In fact, FL is a paradigm of machine learning, with extensive collections of highly heterogeneous customers (such as mobile devices, smart devices, or companies) attempting to train models together using different data stored on these devices without jeopardizing the privacy of their customers' data. For that reason , optimization community [9, 81, 118] considers a more refined version of (1) for distributed federated optimization defined as

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}, \tag{3}$$

where n is the number of devices/clients/nodes/workers, and the possibly non-convex function $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ is usually [86, 53, 52, 85, 50] represented either as **a) an expectation**, i.e.,

$$f_i(x) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[f_{\xi_i}(x)\right], \tag{4}$$

where $\mathcal{D}_i$ describes the distribution of data on device i, or **b) as a finite sum**, i.e.,

$$f_i(x) = \frac{1}{m} \sum_{j=1}^{m} f_{ij}(x), \tag{5}$$

where m corresponds to the number of training examples stored on machine i. For notation simplicity, we assume that each client possesses the same amount of data (dataset size m is shared among workers). In this review, we deal only with the setting **b)**.

[1] For which we do not explicitly assume $\mathbb{E}\left[g^t\right] = \nabla f(x^t)$ for any $x^t \in \mathbb{R}^d$.

Distributed methods are often very efficiently solves the problem (3) [55, 150]. In particular, numerical methods that can handle non-convex objectives are currently of a particular practical interest since they can tackle modern machine learning tasks [25]. In addition, all contemporary numerical methods [15, 76, 30, 131, 29, 56, 152, 94, 93, 57] specified for modern Deep Learning tasks always rely on stochastic approximation [105, 104, 114, 119, 103]. There is a large class of different SGD-type methods[54, 51, 74, 75, 88, 53, 52] that are suitable for many different problem settings. In this review we will deal with minibatch SGD and variance-reduced SGD estimators.

Variance reduction. Well-known practical property of vanilla SGD is that it can reach some crude accuracy much faster than GD in runtime; on the other hand, the accumulated variance from stochastic estimations does not allow to converge to the stationary point on asymptotic [74]. Indeed, SGD converges only to its neighborhood of size $C\gamma$ with the rate $\mathcal{O}\left(1/\sqrt{T}\right)$, where C is a problem defined constant (see Theorem 1 and Corollary 1 in [74]). However, with a carefully designed variance-reduced stochastic estimator, it is possible to convergence to the stationary point with optimal GD-like rate $\mathcal{O}\left(1/T\right)$ getting rid of the noise term. Nonetheless, to the best of our knowledge, every existing VR estimator either requires periodic full gradient computations [64, 80, 85, 112, 62, 5, 82, 87, 42, 156, 59, 38, 145, 107, 84], or it maintains m additional d dimensional vector states [27, 13, 86], which pose a significant memory requirement.

Communication compression. Training of machine learning models in the federated setting requires periodic workers-to-server and server-to-workers communication of large-scale vectors. While the server can broadcast model state x^t comparatively fast, gathering vectors from workers on the server side forms a critical bottleneck in practice [55, 32] . Several approaches are addressing this issue that has been proposed in the literature as of now: (i) reduce the frequency of communication rounds by performing multiple local updates on workers side[96, 102, 153, 109, 128, 99, 18, 123, 141, 91, 124, 142, 151, 83, 69, 72, 17, 16, 147]; (ii) reduce communication volume via certain compression techniques [146, 139, 11, 136, 115, 130, 79, 149, 4, 125, 70, 8, 117, 127, 92, 148, 70, 126, 77, 132, 110, 111, 53, 60, 122, 138]; (iii) reduce communication traffic at server by limiting the participating clients per round [33, 20, 39, 50]; (iv) skip communication if current local gradient estimate does not differ considerably from the previous one [21, 129, 44, 113] (*lazy aggregation*). These three approaches complement each other (and also can be combines). In this review, we focus on methods that apply gradient compression techniques.

The compression mechanism is mathematically formulated as a possibly random map $\mathcal{C}:\mathbb{R}^d\mapsto\mathbb{R}^d$ that takes dense vectors for input and outputs easy-to-communicate (e.g., sparse) vectors. The operator is typically applied either on models [73] or on gradients [3, 12], both of which are vectors in $\mathbb{R}^d$. Later in this work, we consider a class of so-called *unbiased* compression operators $\mathcal{Q}$ (denoted as $\mathbb{U}(\omega)$), for which there exists $\omega\geq 0$ such that

$$\mathbb{E}\left[\mathcal{Q}(x)\right]=x,\quad \mathbb{E}\left[\|\mathcal{Q}(x)-x\|^2\right]\leq\omega\|x\|^2,\quad \forall x\in\mathbb{R}^d; \tag{6}$$

An important example often used for benchmarking optimization methods is Rand-k, which we rigorously define below, state its properties, and give an example of the application.

Definition 1 (Rand-k, [12])**.** *For $k\in[d]\overset{def}{=}\{1,\ldots,d\}$, the unbiased random (aka Rand-k) sparsification operator is defined via*

$$\mathcal{Q}(x)\overset{def}{=}\frac{d}{k}\sum_{i\in S}x_ie_i, \tag{7}$$

where $S\subseteq\{1,\ldots,d\}$ is the subset of cardinality k chosen without replacement uniformly at random, and $e_1,\ldots,e_d$ are the standard unit basis vectors in $\mathbb{R}^d$.

Lemma 1 ([12])**.** *The Rand-k sparsifier* (7) *satisfies* (6) *with $\omega=d/k-1$.*

Example 1. *Let $x=(1,-15{,}2,-7{,}10{,}25)^\top$. Assume that $\mathcal{Q}$ is the Rand-2 compressor. Then one of the possible realization of $\mathcal{Q}$ is $\mathcal{Q}(x)=\frac{6}{2}(1{,}0{,}2{,}0{,}0{,}0)^\top$ (we assume that we sampled S to be $S=\{1{,}3\}$).*

2 Preliminaries

2.1 Privacy

In this section, we introduce the privacy aspects of the federated learning we focus on in this project. While federated learning does not allow servers to directly access client data, federated optimization methods use client data to calculate and send aggregate model updates to servers. Although these updates do not directly contain client data, they must contain downstream information about client data sets. [157] show that, in some cases, it is possible to reconstruct client data to some extent from model gradients. Therefore, developing a federated optimization methodology that preserves privacy in a guaranteed manner is essential [43, 67, 116, 140]. To prevent private information from being leaked during the training process, advanced privacy-preserving tools such as *differential privacy* (DP) [35] have been widely interspersed into training algorithms [34, 19, 36, 1, 137, 61, 24, 40]. We describe two general privacy technologies that are often used for federated optimization.

Data minimization. The first technique often used to improve privacy by reducing the surface of attacks is usually called **data minimization**. In federated learning, the default data minimization is achieved by ensuring that raw data is never shared with the server and that only a tiny update focused on an immediate aggregate is sent to the server. To achieve higher levels of privacy, we often use additional data minimization techniques. One such technique is secure calculation. Secure computation provides cryptography primitives that enable federated aggregation without revealing raw inputs from unreliable parties. [10, 14] propose a secure aggregation(SECAGG) that calculates secure federated sums on client inputs. This allows the calculation of the(weighted) average of the client update without revealing the individual update to the server. SECAGG has a high level of privacy, but also imposes limitations and constraints that must be taken into account when developing optimization methods compatible with SECAGG. A notable limitationis that SECAGG operates at federated sums(i.e. sums of client updates). Consequently, algorithms that require aggregates that federated sums cannot carry out are incompatible. Although it may be restrictive, this restriction opens the door to new research on developing SECAGG-compatible versions of federated optimization methods. For example, Pilutlaet al. [108] shows that the median-based aggregation can be well approximated of several federated sums (in the same client group). Also, Karimireddy et al. [68] and Vogels et al. [136] designed new communication compression and robust aggregation protocols specifically compatible with SECAGG. The question of how to approximate other aggregation strategies (such as model distillation and fusion [90]) using only federated sums remains open, as well as the implementation of a reliable, practical calculation method beyond the aggregation summary of complex aggregation methods.

Data anonymization. Another technique aimed at strengthening privacy is **data anonymization**. One of them is differential privacy. Informally, differentially private mechanisms are random algorithms for database queries, and given the output of the query and any side information, the objective is difficult to distinguish between a specific element and the data. As it was mentioned earlier, in cross-device FL, federated optimization provides a natural way to ensure this type of "client-level" differential privacy. Client\user\worker contributes to training by sending update during the communication round. Moreover, one can enforce user-level DP guarantees by applying privatizing transformations to these updates. For example, in the case of FEDAvG, the client updates can be clipped, and an adjusted amount of Gaussian noise can be added to the average, which is often sufficient to conceal the influence of any single client [100].

Central vs. local privacy Central privacy refers to the settings in which the trusted server collects client model updates, clips and aggregates the model updates, adds noise to the aggregate, updates the server model, and shares the updated model with (possibly) untrustworthy parties (or later federated clients). As a result, this differential privacy set protects clients' data privacy from external parties relying on trusted servers.

The second setting called local privacy, ensures that each customer protects data privacy from (possibly) compromised entities, including non-reliable central servers. Here, privacy protections apply to servers and other customers participating in distributed optimization. This is done by locally (on a device) clipping and noizing model updates and then sending noisy and clipped model updates to the server.

Whereas the local privacy setting provides a more robust notion against untrusted servers, it suffers from low learning performance compared to the central privacy because of the significant amount of noise to be added to obscure individual updates [31, 65, 71]. Fortunately, local privacy can be boosted when combined with the use of either a secure shuffler [6, 37, 46, 47] or a secure aggregator [66].

Finally,we give a rigorous definition of local differential privacy (LDP) property

Definition 2 (Local differential privacy (LDP) [89])**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ϵ, δ)-locally differentially private for client i if for all neighboring datasets $D_i, D_i' \in \mathcal{D}$ on client i and for all events $S \in \mathcal{R}$ in the output space of $\mathcal{M}$, we have*

$$\Pr\{\mathcal{M}(D_i) \in S\} \leq e^{\epsilon} \Pr\{\mathcal{M}(D_i') \in S\} + \delta.$$

More on Differentially Private optimization methods To the best of our knowledge, Abadi et al[1] was the first who developed a differentially-private version of SGD algorithm called DP-SGD. However, it was done in the single-node setting. Later on, several DP algorithms [63, 143, 120, 95] were proposed for Federated setting.

However, there have been only a few results simultaneously addressing communication compression and differential privacy. Unfortunately, they provide only partial answers to the above-mentioned questions. Most existing work only considers basic compression schemes that apply directly to stochastic gradient estimations in DP-SGD [2, 144, 47, 158, 155, 28].

There are two ways to combine privacy and compression appeared in the literature: (1) first perturb and then compress, and (2) first compress and then perturb. The first method is straightforward and general because compression retains differential privacy and works seamlessly with existing privacy mechanisms. The second method, however, requires a carefully designed perturbation mechanism (in other words, the perturbation can reduce the compression communication saving), for example, a binomial perturbation [2] or discrete Gaussian perturbation [66]. In addition, the first method was more useful in some settings than the second method [28]. Therefore, we are considering the first approach in this paper project. An alternative line of development [130, 23, 22, 66, 41, 133] tackling communication bottleneck approached this topic from the practical perspective without providing concrete trade-offs in terms of privacy, utility, and communication.

Furthermore, there is a lack of theoretical analyses in the non-convex setting. Existing one is usually either limited to convex problems [47] or provides pessimistic guarantees on utility and/or communication under strong assumptions [155, 28, 158].

Finally, most of existing results only studied the DP methods where compression if applied directly to SGD estimator. However, in the non-private supervised federated setting, it has been show that introducing an additional control variate [101, 58, 88, 121] for each worker may reduce the compression variance and, therefore, reach much better convergence guarantees. Finally, most existing results only studied the DP methods where compression is applied directly to the SGD estimator. However, in the non-private supervised federated setting, it has been shown that introducing an additional control variate [101, 58, 88, 121] for each worker may reduce the compression variance and, therefore, reach much better convergence guarantees. This refers to so-called DIANA framework.

Recently, [89] firstly proposed a DIANA-like framework for DP setting called SoteriaFL. They provided unified characterization regarding privacy, utility, and communication complexity for the whole family of local gradient estimators, including well-known variance-reduced gradient methods and state-of-the-art, shifted compression schemes. The proposed unified approach allowed them to obtain theoretical guarantees for several new private Federated methods, including SoteriaFL-GD, SoteriaFL-SGD, SoteriaFL-SVRG, and SoteriaFL-SAGA. In addition, they analyzed a plain compressed DP-SGD (called CDP-SGD). They showed that the achieved communication complexity and utility

bounds are worse than shifted counterparts listed above. This remaining part of the review will be primarily based on the results from [89].

3 Main part

Before we shift to the main result we need state some assumptions

Assumption 1 (Smoothness, [89])**.** *There exists some $L \geq 0$, such that for all $i \in [n], j \in [m]$, the function $f_{i,j}$ is L-smooth, i.e.,*

$$\|\nabla f_{i,j}(x_1) - \nabla f_{i,j}(x_2)\| \leq L \|x_1 - x_2\|, \qquad \forall x_1, x_2 \in \mathbb{R}^d.$$

Assumption 2 (Bounded gradient, , [89])**.** *There exists some $G \geq 0$, such that for all $i \in [n], j \in [m]$ and $x \in \mathbb{R}^d$, we have $\|\nabla f_{i,j}(x)\| \leq G$.*

The smoothness assumption is standard in the non-convex setting [106, 45]. Bounded gradient assumption is also widely used in differential privacy [7, 137, 61, 40].

3.1 CDP-SGD

[89] first introduced a simple algorithm CDP-SGD, which unified several existing algorithms [155, 158] for private FL with better theoretical guarantees.

Algorithm 1 Compressed Differentially-Private Stochastic Gradient Descent (CDP-SGD) , [89]

Require: initial point x^0, stepsize η_t, variance σ_p^2, minibatch size b

1: **for** $t = 0,1,2,\ldots,T$ **do**
2: **for each node** $i \in [n]$ **do in parallel**
3: Sample a random minibatch $\mathcal{I}_b$ from local dataset D_i
4: Compute local stochastic gradient $\tilde{g}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b} \nabla f_{i,j}(x^t)$
5: *Privacy*: $g_i^t = \tilde{g}_i^t + \xi_i^t$, where $\xi_t^i \sim \mathcal{N}(0,\sigma_p^2\mathbf{I})$
6: *Compression*: let $v_i^t = \mathcal{C}_i^t(g_i^t)$ and send to the server
7: **end each node**
8: Server aggregates compressed information $v^t = \frac{1}{n}\sum_{i=1}^n v_i^t$
9: $x^{t+1} = x^t - \eta_t v^t$
10: **end for**

Theorem 1 (Privacy, utility and communication for CDP-SGD)**.** *theorem Suppose that Assumptions 1 and 2 hold, and the compression operators $\mathcal{C}_i^t$ (cf. Line 6 of Algorithm 1) are drawn independently satisfying Definition 6. By choosing the algorithm parameters properly and letting the total number of communication rounds*

$$T = \mathcal{O}\left(\frac{\sqrt{nL}m\epsilon}{G\sqrt{(1+\omega)d\log(1/\delta)}} + \frac{m^2\epsilon^2}{d\log(1/\delta)}\right),$$

CDP-SGD *(Algorithm 1) satisfies (ϵ,δ)-LDP and the utility*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \mathcal{O}\left(\frac{G\sqrt{(1+\omega)Ld\log(1/\delta)}}{\sqrt{n}m\epsilon}\right).$$

When $\omega = 0$ and $n = 1$, utility guarantee of CDP-SGD reduces to $\mathcal{O}\left(\frac{\sqrt{d\log(1/\delta)}}{m\epsilon}\right)$ achieving the best-known utility for standard DP-SGD in the single-node case. However, the utility and communication complexity of CDP-SGD outperforms [28] by a factor of $T^{1/6}$ and [155] respectively. However, the communication complexity of CDP-SGD is far from being SOTA. In particular, if m is dominating, then CDP-SGD (even in the exact gradient scenario) requires $\mathcal{O}(m^2)$ communication rounds This is a crucial concern since both Distributed DP-SRM [143] and ISRL-DP FedProx-SPIDER [95] are having $\mathcal{O}(m)$ communication complexity, but they do not have compression.

3.2 SoteriaFL

Because of the limitations of plain CDP-SGD, [89] proposed a unified private FL framework (called SoteriaFL) by adopting shifted compression technique. That helps reduce the communication complexity of CDP-SGD from $\mathcal{O}(m^2)$ to $\mathcal{O}(m)$, which matches previous uncompressed DP algorithms.

Algorithm 2 SoteriaFL (a unified framework for compressed private FL)

Require: initial point x^0, stepsize η_t, shift stepsize γ_t, variance σ_p^2, initial reference $s_i^0 = 0$

1: **for** $t = 0,1,2,\ldots,T$ **do**
2: **for each node** $i \in [n]$ **do in parallel**
3: Compute local gradient estimator $\tilde{g}_i^t$ (it allows many methods, e.g., SGD, SVRG, and SAGA)
4: *Privacy*: $g_i^t = \tilde{g}_i^t + \xi_i^t$, where $\xi_i^t \sim \mathcal{N}(0,\sigma_p^2\mathbf{I})$
5: *Compression*: let $v_i^t = \mathcal{C}_i^t(g_i^t - s_i^t)$ and send to the server // shifted compression
6: Update shift $s_i^{t+1} = s_i^t + \gamma_t \mathcal{C}_i^t(g_i^t - s_i^t)$
7: **end each node**
8: Server aggregates compressed information $v^t = s^t + \frac{1}{n}\sum_{i=1}^n v_i^t$
9: $x^{t+1} = x^t - \eta_t v^t$
10: $s^{t+1} = s^t + \gamma_t \frac{1}{n}\sum_{i=1}^n v_i^t$
11: **end for**

The key novelty of [89] is that this is the first work that proposes a unified view of DP algorithms. Indeed, it introduces a generic assumption 3 and derives a parameters-defined utility guarantee (Theorem 2). Later we will see that the choice of the particular compressor determines each parameter set.

Assumption 3 (Generic assumption of local gradient estimator for SoteriaFL [89]). *The gradient estimator $\tilde{g}_i^t$ (Line 3 of Algorithm 2) is unbiased $\mathbb{E}_t[\tilde{g}_i^t] = \nabla f_i(x^t)$ for $i \in [n]$, where $\mathbb{E}_t$ takes the expectation conditioned on all history before round t. Moreover, it can be decomposed into two terms $\tilde{g}_i^t := \mathcal{A}_i^t + \mathcal{B}_i^t$ and there exist constants $G_A, G_B, C_1, C_2, C_3, C_4, \theta$ and a random sequence $\{\Delta^t\}$ such that*

$$\mathcal{A}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b} \varphi_{i,j}^t, \qquad \mathcal{B}_i^t = \frac{1}{m}\sum_{j=1}^m \psi_{i,j}^t, \tag{8a}$$

$$\mathbb{E}_t\Big[\frac{1}{n}\sum_{i=1}^n \|\tilde{g}_i^t - \nabla f_i(x^t)\|^2\Big] \le C_1\Delta^t + C_2, \tag{8b}$$

$$\mathbb{E}_t\left[\Delta^{t+1}\right] \le (1-\theta)\Delta^t + C_3\|\nabla f(x^t)\|^2 + C_4\mathbb{E}_t\|x^{t+1} - x^t\|^2, \tag{8c}$$

where $\varphi_{i,j}^t$ and $\psi_{i,j}^t$ are bounded by G_A and G_B respectively, and $\mathcal{I}_b$ usually denotes a random minibatch with size b. Here, $\varphi_{i,j}^t$ and $\psi_{i,j}^t$ should be viewed as functions related to the j-th sample $d_{i,j}$ stored on client i.

Theorem 2 (Privacy for SoteriaFL [89]). *Suppose that Assumption 3 holds. There exist constants c and c', for any $\epsilon < c'b^2T/m^2$ and $\delta \in (0,1)$, SoteriaFL (Algorithm 2) is (ϵ,δ)-LDP if we choose*

$$\sigma_p^2 = c\frac{(G_A^2/4 + G_B^2)T\log(1/\delta)}{m^2\epsilon^2}. \tag{9}$$

Theorem 3 (Utility and communication for SoteriaFL [89]). *Suppose that Assumptions 1 and 3 hold, and the compression operators $\mathcal{C}_i^t$ (cf. Line 5 of Algorithm 2) are drawn independently satisfying Definition* (6)*. Set the stepsize as*

$$\eta_t \equiv \eta \le \min\left\{\frac{1}{(1 + 2\alpha C_4 + 4\beta(1+\omega) + 2\alpha C_3/\eta^2)L}, \frac{\sqrt{\beta n}}{\sqrt{1 + 2\alpha C_4 + 4\beta(1+\omega)}(1+\omega)L}\right\},$$

where $\alpha = \frac{3\beta C_1}{2(1+\omega)\theta L^2}$, $\forall \beta > 0$, *the shift stepsize as* $\gamma_t \equiv \sqrt{\frac{1+2\omega}{2(1+\omega)^3}}$, *and the privacy variance* σ_p^2 *according to Theorem 2. Then,* SoteriaFL *(Algorithm 2) satisfies* (ϵ,δ)*-LDP and the following*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x^t)\|^2 \leq \frac{2\Phi_0}{\eta T} + \frac{3\beta}{(1+\omega)L\eta}\left(C_2 + \frac{c(G_A^2/4+G_B^2)dT\log(1/\delta)}{m^2\epsilon^2}\right),$$

where $\Phi_0 := f(x^0) - f^* + \alpha L\Delta^0 + \frac{\beta}{Ln}\sum_{i=1}^n \|\nabla f_i(x^0) - s_i^0\|^2$. *By further choosing the total number of communication rounds* T *as*

$$T = \max\left\{\frac{m\epsilon\sqrt{2(1+\omega)L\Phi_0}}{\sqrt{3\beta cd(G_A^2/4+G_B^2)\log(1/\delta)}}, \frac{C_2 m^2\epsilon^2}{cd(G_A^2/4+G_B^2)\log(1/\delta)}\right\}, \tag{10}$$

SoteriaFL *has the following utility (accuracy) guarantee:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x^t)\|^2 \leq \mathcal{O}\left(\max\left\{\frac{\sqrt{\beta d(G_A^2/4+G_B^2)\log(1/\delta)}}{\eta m\epsilon\sqrt{(1+\omega)L}}, \frac{\beta C_2}{(1+\omega)L\eta}\right\}\right). \tag{11}$$

Theorem 3 is a generic theorem the SoteriaFL framework, which covers a large family of methods under the unified Assumption 3.

3.3 Some Algorithms within SoteriaFL Framework

In this section, we list several newly proposed private FL algorithms (SoteriaFL-GD, SoteriaFL-SGD, SoteriaFL-SVRG and SoteriaFL-SAGA) that are all special cases of the generic Algorithm 2.

Algorithm 3 SoteriaFL-SGD, SoteriaFL-SVRG, and SoteriaFL-SAGA

Require: initial point x^0, stepsize η_t, shift stepsize γ_t, variance σ_p^2, minibatch size b, initial reference $s_i^0 = 0$, initial $w^0 = x^0$ for SVRG or $w_{i,j}^0 = x^0$ for SAGA, probability p

for $t = 0,1,2,\ldots,T$ **do**
  for each node $i \in [n]$ **do in parallel**
    Option 1: SGD
      Compute local SGD estimator $\tilde{g}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b}\nabla f_{i,j}(x^t)$ // GD if choose $b = m$
    Option 2: SVRG
      Compute local SVRG estimator $\tilde{g}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b}(\nabla f_{i,j}(x^t) - \nabla f_{i,j}(w^t)) + \nabla f_i(w^t)$
      Update SVRG snapshot point $w^{t+1} = \begin{cases} x^t, & \text{with probability } p \\ w^t, & \text{with probability } 1-p \end{cases}$
    Option 3: SAGA
      Compute local SAGA estimator:
        $\tilde{g}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b}(\nabla f_{i,j}(x^t) - \nabla f_{i,j}(w_{i,j}^t)) + \frac{1}{m}\sum_{j=1}^m \nabla f_{i,j}(w_{i,j}^t)$
      Update SAGA variables $w_{i,j}^{t+1} = \begin{cases} x^t, & \text{for } j \in \mathcal{I}_b \\ w_{i,j}^t, & \text{for } j \notin \mathcal{I}_b \end{cases}$
    End Options
    Privacy: $g_i^t = \tilde{g}_i^t + \xi_i^t$, where $\xi_i^t \sim \mathcal{N}(0,\sigma_p^2\mathbf{I})$
    Compression: let $v_i^t = \mathcal{C}_i^t(g_i^t - s_i^t)$ and send to the server
    Update shift $s_i^{t+1} = s_i^t + \gamma_t\mathcal{C}_i^t(g_i^t - s_i^t)$
  end each node
  Server aggregates compressed information $v^t = s^t + \frac{1}{n}\sum_{i=1}^n v_i^t$
  $x^{t+1} = x^t - \eta_t v^t$
  $s^{t+1} = s^t + \gamma_t\frac{1}{n}\sum_{i=1}^n v_i^t$
end for

The following Lemma 3 derives parameters of the Assumption 3 for methods GD, SGD, SVRG, and SAGA

Lemma 2 (SGD/SVRG/SAGA estimators satisfy Assumption 3 [89]). *Suppose that Assumptions 1 and 2 hold. The local SGD estimator $\tilde{g}_i^t$ (Option 1 in Algorithm 3) satisfies Assumption 3 with*

$$G_A = G, G_B = C_1 = C_3 = C_4 = 0, C_2 = \frac{(m-b)G^2}{mb}, \theta = 1, \Delta^t \equiv 0.$$

The local SVRG estimator $\tilde{g}_i^t$ (Option 2 in Algorithm 3) satisfies Assumption 3 with

$$G_A = 2G, G_B = G, C_1 = \frac{L^2}{b}, C_2 = 0, C_3 = \frac{2(1-p)\eta^2}{p}, C_4 = 1, \theta = \frac{p}{2}, \Delta^t = \|x^t - w^t\|^2.$$

The local SAGA estimator $\tilde{g}_i^t$ (Option 3 in Algorithm 3) satisfies Assumption 3 with

$$G_A = 2G, G_B = G, C_1 = \frac{L^2}{b}, C_2 = 0, C_3 = \frac{2(m-b)\eta^2}{b}, C_4 = 1,$$
$$\theta = \frac{b}{2m}, \Delta^t = \frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m \|x^t - w_{i,j}^t\|^2.$$

Armed with Lemma 2, one can plug corresponding parameters into the unified Theorem 3 and communication complexity and utility bounds.

Corollary 1 (SoteriaFL-SGD/SoteriaFL-GD [89]). *Suppose that Assumptions 1 and 2 hold and we combine Theorem 3 and Lemma 2, i.e., choosing stepsize $\eta_t \equiv \eta \leq \frac{1}{(1+2\sqrt{(1+\omega)^3/n})L}$, where we set $\beta = \frac{\tau}{2(1+\omega)}$ and $\tau := \frac{(1+\omega)^{3/2}}{n^{1/2}}$, shift stepsize $\gamma_t \equiv \sqrt{\frac{1+2\omega}{2(1+\omega)^3}}$, and privacy variance $\sigma_p^2 = \mathcal{O}\big(\frac{G^2T\log(1/\delta)}{m^2\epsilon^2}\big)$. If we further set the minibatch size $b = \min\Big\{\frac{m\epsilon G\sqrt{\beta}}{\sqrt{(1+\omega)Ld\log(1/\delta)}}, m\Big\}$ and the total number of communication rounds $T = \mathcal{O}\Big(\frac{\sqrt{nL}m\epsilon}{G\sqrt{(1+\omega)d\log(1/\delta)}}(1+\sqrt{\tau})\Big)$, then SoteriaFL-SGD satisfies (ϵ,δ)-LDP and the following utility guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{E}\|\nabla f(x_t)\|^2 \leq \mathcal{O}\Big(\frac{G\sqrt{(1+\omega)Ld\log(1/\delta)}}{\sqrt{n}m\epsilon}(1+\sqrt{\tau})\Big)$. If we choose a minibatch size $b = m$ (local full gradient) in SoteriaFL-SGD, the result of SoteriaFL-SGD leads to that of SoteriaFL-GD.*

Corollary 2 (SoteriaFL-SVRG [89]). *Suppose that Assumptions 1 and 2 hold and we combine Theorem 3 and Lemma 2, i.e., choosing stepsize $\eta_t \equiv \eta \leq \frac{p^{2/3}b^{1/3}\min\{1,\sqrt{n/(1+\omega)^3}\}}{2L}$, where we set $\beta = \frac{p^{4/3}b^{2/3}(1+\omega)^2\min\{1,n/(1+\omega)^3\}}{n}$, $p^{2/3}b^{1/3} \leq 1/4$ and $p \leq 1/4$, shift stepsize $\gamma_t \equiv \sqrt{\frac{1+2\omega}{2(1+\omega)^3}}$, and privacy variance $\sigma_p^2 = \mathcal{O}\big(\frac{G^2T\log(1/\delta)}{m^2\epsilon^2}\big)$. If we further let the minibatch size $b = \frac{m^{2/3}}{4}$, the probability $p = b/m$, and the total number of communication rounds $T = \mathcal{O}\Big(\frac{\sqrt{nL}m\epsilon}{G\sqrt{(1+\omega)d\log(1/\delta)}}\max\{1,\tau\}\Big)$, where $\tau := \frac{(1+\omega)^{3/2}}{n^{1/2}}$, then SoteriaFL-SVRG satisfies (ϵ,δ)-LDP and the following utility guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x^t)\|^2 \leq \mathcal{O}\Big(\frac{G\sqrt{(1+\omega)Ld\log(1/\delta)}}{\sqrt{n}m\epsilon}\Big)$.*

Corollary 3 (SoteriaFL-SAGA [89]). *Suppose that Assumptions 1 and 2 hold and we combine Theorem 3 and Lemma 2, i.e., choosing stepsize $\eta_t \equiv \eta \leq \frac{\min\{1,\sqrt{n/(1+\omega)^3}\}}{3L}$, where we set $\beta = \frac{(1+\omega)^2\min\{1,n/(1+\omega)^3\}}{3n}$, minibatch size $b = 3m^{2/3}$, shift stepsize $\gamma_t \equiv \sqrt{\frac{1+2\omega}{2(1+\omega)^3}}$, and privacy variance $\sigma_p^2 = \mathcal{O}\big(\frac{G^2T\log(1/\delta)}{m^2\epsilon^2}\big)$. If we further let the communication rounds $T = \mathcal{O}\Big(\frac{\sqrt{nL}m\epsilon}{G\sqrt{(1+\omega)d\log(1/\delta)}}\max\{1,\tau\}\Big)$, where $\tau := \frac{(1+\omega)^{3/2}}{n^{1/2}}$, then SoteriaFL-SAGA satisfies (ϵ,δ)-LDP and the following utility guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x^t)\|^2 \leq \mathcal{O}\Big(\frac{G\sqrt{(1+\omega)Ld\log(1/\delta)}}{\sqrt{n}m\epsilon}\Big)$.*

Thus, when the local dataset size m is large, one can conclude that SoteriaFL-style algorithms have better communication complexity than CDP-SGD ($\mathcal{O}(m)$ as opposed to $\mathcal{O}(m^2)$). However, this

Algorithm	Privacy	Utility/Accuracy	Communication Complexity	Remark
RPPSGD [154]	(ϵ,δ)-DP	$\frac{\sqrt{d\log(m/\delta)\log(1/\delta)}}{m\epsilon}$	—	single node
DP-GD/SGD [1, 137]	(ϵ,δ)-DP	$\frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$	—	single node
DP-SRM [143]	(ϵ,δ)-DP	$\frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$	—	single node
Distributed DP-SRM [143] (1)	(ϵ,δ)-DP	$\frac{\sqrt{d\log(1/\delta)}}{nm\epsilon}$	$\frac{n^2m\epsilon\sqrt{d}}{\sqrt{\log(1/\delta)}}$	n nodes, no compression
ISRL-DP FedProx-SPIDER [95]	(ϵ,δ)-LDP	$\left(\frac{\sqrt{GLd\ln(1/\delta)}}{\epsilon m\sqrt{n}}\right)^{4/3}+\frac{G^2d\ln(1/\delta)}{\epsilon^2m^2n}$	$\frac{n^{3/2}m\epsilon\sqrt{d}}{\sqrt{\log(1/\delta)}}$	n nodes, no compression
Q-DPSGD-1 [28] (2)	(ϵ,δ)-LDP	$\frac{(\tilde{\sigma}^2/n+1/m)^{2/3}(d\log(1/\delta))^{1/3}}{m^{2/3}\epsilon^{2/3}}$	$\frac{(1+n/(m\tilde{\sigma}^2))m^2\epsilon^2}{d\log(1/\delta)}$	n nodes, direct compression
SDM-DSGD [155] (3)	(ϵ,δ)-LDP	$\tilde{O}\left(\frac{\sqrt{d\log(1/\delta)}}{\sqrt{n}m\epsilon}\right)$	$\frac{n^{7/2}m\epsilon\sqrt{d}}{(1+\omega)^{3/2}\sqrt{\log(1/\delta)}}+\frac{nm^2\epsilon^2}{(1+\omega)\log(1/\delta)}$	n nodes, direct compression
CDP-SGD-SGD (Theorem 1)	(ϵ,δ)-LDP	$\frac{\sqrt{(1+\omega)d\log(1/\delta)}}{\sqrt{n}m\epsilon}$	$\frac{n^{3/2}m\epsilon\sqrt{d}}{(1+\omega)^{3/2}\sqrt{\log(1/\delta)}}+\frac{nm^2\epsilon^2}{(1+\omega)\log(1/\delta)}$	n nodes, direct compression
SoteriaFL-SGD SoteriaFL-GD (4) (Corollary 1)	(ϵ,δ)-LDP	$\frac{\sqrt{(1+\omega)d\log(1/\delta)}}{\sqrt{n}m\epsilon}(1+\sqrt{\tau})$	$\frac{n^{3/2}m\epsilon\sqrt{d}}{(1+\omega)^{3/2}\sqrt{\log(1/\delta)}}(1+\sqrt{\tau})$	n nodes, shifted compression
SoteriaFL-SVRG SoteriaFL-SAGA (4) (Corollary 2, 3)	(ϵ,δ)-LDP	$\frac{\sqrt{(1+\omega)d\log(1/\delta)}}{\sqrt{n}m\epsilon}$	$\frac{n^{3/2}m\epsilon\sqrt{d}}{(1+\omega)^{3/2}\sqrt{\log(1/\delta)}}(1+\tau)$	n nodes, shifted compression

Table 1: Comparisons among (local) DP algorithms for the non-convex problem in both central (single-node) and distributed (n-node) settings. Here, m denotes the number of data stored on a single client, n is the number of clients, d is the dimension, and ω is the parameter for the compression operator. The communication complexity is computed by $ndT/(1+\omega)$, where T is the total number of communication rounds, and $nd/(1+\omega)$ is the communication cost per round. The utility / accuracy measures the average squared gradient norm of the objective function after T rounds. Note that the algorithm is better when the utility/accuracy and the communication complexity are small under the same privacy guarantee. Table and caption were taken from [89] with some modifications

improvement is made with the price of utility. Indeed, both SoteriaFL-SVRG and SoteriaFL-SAGA have additional factor $1+\sqrt{\tau}$. Thus, the best utility they can hope for is the same one as for the CDP-SGD, which might be the case when the number of clients n is large.

Finally, none of the proposed communication efficient methods in the paper [89] is optimal since current utility lower bound for non convex objectives is $\Omega\left(L^2\min\left(1,\frac{d\log(1/\delta)}{m^2\epsilon^2}\right)\right)$ and the best known has ISRL-DP FedProx-SPIDER with $\mathcal{O}\left(\left(\frac{\sqrt{GLd\ln(1/\delta)}}{\epsilon m\sqrt{n}}\right)^{4/3}+\frac{G^2d\ln(1/\delta)}{\epsilon^2m^2n}\right)$. Providing an optimal communication-efficient distributed method for non-convex objectives with a local privacy guarantee is still an open problem in the DP community.

All of the complexities upper bounds are summarized in the Table 1.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[3] 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, 2017.

[4] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 5973–5983, 2018.

[5] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pages 699–707. PMLR, 2016.

[6] Borja Balle, Peter Kairouz, Brendan McMahan, Om Dipakbhai Thakkar, and Abhradeep Thakurta. Privacy amplification via random check-ins. In *Advances in Neural Information Processing Systems*, 2020.

[7] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

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

[9] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.

[10] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. In *CCS*, pages 1253–1269. ACM, 2020.

[11] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning (ICML)*, pages 560–569. PMLR, 2018.

[12] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

[13] Adel Bibi, Alibek Sailanbayev, Bernard Ghanem, Robert Mansel Gower, and Peter Richtárik. Improving SAGA via a probabilistic interpolation with gradient descent. *arXiv preprint arXiv:1806.05633*, 2018.

[14] 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*, pages 1175–1191, 2017.

[15] Leon Bottou. *Stochastic Gradient Descent Tricks*. Springer, 2012.

[16] Zachary Charles and Jakub Konečnỳ. Convergence and accuracy trade-offs in federated learning and meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2575–2583. PMLR, 2021.

[17] Zachary Charles and Jakub Konečný. On the outsized importance of learning rates in local update methods. *arXiv preprint arXiv:2007.00878*, 2020.

[18] Pratik Chaudhari, Carlo Baldassi, Riccardo Zecchina, Stefano Soatto, Ameet Talwalkar, and Adam Oberman. Parle: parallelizing stochastic gradient descent. *arXiv preprint arXiv:1707.00424*, 2017.

[19] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[20] Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD. *arXiv preprint arXiv:1604.00981*, 2016.

[21] Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. LAG: Lazily aggregated gradient for communication-efficient distributed learning. *Advances in Neural Information Processing Systems*, 2018.

[22] Wei-Ning Chen, Christopher A Choquette-Choo, and Peter Kairouz. Communication efficient federated learning with secure aggregation and differential privacy. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[23] Wei-Ning Chen, Peter Kairouz, and Ayfer Özgür. Breaking the communication-privacy-accuracy trilemma. *arXiv preprint arXiv:2007.11707*, 2020.

[24] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

[25] Marina Danilova, Pavel Dvurechensky, Alexander Gasnikov, Eduard Gorbunov, Sergey Guminov, Dmitry Kamzolov, and Innokentiy Shibaev. Recent theoretical advances in non-convex optimization. *arXiv preprint arXiv:2012.06188*, 2020.

[26] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092, 2020.

[27] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[28] Jiahao Ding, Guannan Liang, Jinbo Bi, and Miao Pan. Differentially private and communication efficient collaborative learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7219–7227, 2021.

[29] Timothy Dozat. Incorporating nesterov momentum into Adam. 2016.

[30] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[31] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[32] Aritra Dutta, El Houcine Bergou, Ahmed M Abdelmoniem, Chen-Yu Ho, Atal Narayan Sahu, Marco Canini, and Panos Kalnis. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3817–3824, 2020.

[33] Sanghamitra Dutta, Gauri Joshi, Soumyadip Ghosh, Parijat Dube, and Priya Nagpurkar. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. In *International Conference on Artificial Intelligence and Statistics*, pages 803–812. PMLR, 2018.

[34] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[35] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[36] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[37] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

[38] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

[39] Ilyas Fatkhullin, Igor Sokolov, Eduard Gorbunov, Zhize Li, and Peter Richtárik. EF21 with bells & whistles: practical algorithmic extensions of modern error feedback. *arXiv preprint arXiv:2110.03294*, 2021.

[40] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

[41] Vitaly Feldman and Kunal Talwar. Lossless compression of efficient private local randomizers. In *International Conference on Machine Learning*, pages 3208–3219. PMLR, 2021.

[42] Rong Ge, Zhize Li, Weiyao Wang, and Xiang Wang. Stabilized SVRG: Simple variance reduction for nonconvex optimization. In *Conference on learning theory*, pages 1394–1448. PMLR, 2019.

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

[44] Hossein Shokri Ghadikolaei, Sebastian Stich, and Martin Jaggi. LENA: Communication-efficient distributed learning with self-triggered gradient uploads. In *International Conference on Artificial Intelligence and Statistics*, pages 3943–3951. PMLR, 2021.

[45] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[46] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *arXiv preprint arXiv:1906.08320*, 2019.

[47] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2021.

[48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[49] Eduard Gorbunov. Linearly converging error compensated SGD. In *Federated Learning One World Seminar*. 2020.

[50] Eduard Gorbunov, Konstantin P. Burlachenko, Zhize Li, and Peter Richtárik. MARINA: Faster non-convex distributed learning with compression. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3788–3798. PMLR, 18–24 Jul 2021.

[51] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR, 2020.

[52] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local SGD: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3556–3564. PMLR, 2021.

[53] Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly converging error compensated SGD. *Advances in Neural Information Processing Systems*, 33:20889–20900, 2020.

[54] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209. PMLR, 2019.

[55] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[56] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[57] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Overview of mini-batch gradient descent. *Neural Networks for Machine Learning*, 575(8), 2012.

[58] 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.

[59] Samuel Horváth, Lihua Lei, Peter Richtárik, and Michael I Jordan. Adaptivity of stochastic gradient methods for nonconvex optimization. *arXiv preprint arXiv:2002.05359*, 2020.

[60] Samuel Horváth and Peter Richtárik. A better alternative to error feedback for communication-efficient distributed learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[61] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

[62] Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29, 2016.

[63] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[64] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

[65] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. *arXiv preprint arXiv:1602.07387*, 2016.

[66] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. *International Conference on Machine Learning (ICML)*, 2021.

[67] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[68] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319. PMLR, 2021.

[69] 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. *International Conference on Machine Learning (ICML)*, 2020.

[70] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *International Conference on Machine Learning (ICML)*, volume 97, pages 3252–3261, 2019.

[71] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[72] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4519–4529, 2020.

[73] Ahmed Khaled and Peter Richtárik. Gradient descent with compressed iterates. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[74] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

[75] Ahmed Khaled, Othmane Sebbouh, Nicolas Loizou, Robert M. Gower, and Peter Richtárik. Unified analysis of stochastic gradient methods for composite convex and smooth optimization. *arXiv preprint arXiv:2006.11573*, 2020.

[76] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *The 3rd International Conference on Learning Representations*, 2014.

[77] Anastasia Koloskova, Sebastian U Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, 2019.

[78] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: distributed machine learning for on-device intelligence. *arXiv:1610.02527*, 2016.

[79] Jakub Konečný, H. Brendan McMahan, Felix Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[80] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020.

[81] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*. Springer, 2020.

[82] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via SCSG methods. *Advances in Neural Information Processing Systems*, 30, 2017.

[83] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[84] Zhize Li. SSRGD: Simple stochastic recursive gradient descent for escaping saddle points. *Advances in Neural Information Processing Systems*, 32, 2019.

[85] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, pages 6286–6295. PMLR, 2021.

[86] Zhize Li, Slavomír Hanzely, and Peter Richtárik. ZeroSARAH: Efficient nonconvex finite-sum optimization with zero full gradient computation. *arXiv preprint arXiv:2103.01447*, 2021.

[87] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[88] Zhize Li and Peter Richtárik. A unified analysis of stochastic gradient methods for nonconvex federated optimization. *arXiv preprint arXiv:2006.07013*, 2020.

[89] Zhize Li, Haoyu Zhao, Boyue Li, and Yuejie Chi. Soteriafl: A unified framework for private federated learning with communication compression. *arXiv preprint arXiv:2206.09888*, 2022.

[90] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[91] Tao Lin, Sebastian U. Stich, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *International Conference on Learning Representations (ICLR)*, 2018.

[92] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations (ICLR)*, 2018.

[93] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[94] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[95] Andrew Lowy, Ali Ghafelebashi, and Meisam Razaviyayn. Private non-convex federated learning without a trusted server. *arXiv preprint arXiv:2203.06735*, 2022.

[96] Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464. Association for Computational Linguistics, 2010.

[97] Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[98] H 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)*, 2017.

[99] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017. Initial version posted on arXiv in February 2016.

[100] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[101] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[102] Philipp Moritz, Robert Nishihara, Ion Stoica, and Michael I Jordan. SparkNet: Training deep networks in spark. *arXiv preprint arXiv:1511.06051*, 2015.

[103] A Nemirovski, A Juditsky, G Lan, and A Shapiro. Stochastic approximation approach to stochastic programming. In *SIAM J. Optim*. Citeseer.

[104] Arkadi Nemirovski and D Yudin. On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions. In *Soviet Mathematics. Doklady*, volume 19, pages 258–269, 1978.

[105] Arkadi Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.

[106] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course (Applied Optimization)*. Kluwer Academic Publishers, 2004.

[107] Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *J. Mach. Learn. Res.*, 21(110):1–48, 2020.

[108] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

[109] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur. Parallel training of dnns with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, 2014.

[110] Xun Qian, Hanze Dong, Peter Richtárik, and Tong Zhang. Error compensated loopless SVRG for distributed optimization. *OPT2020: 12th Annual Workshop on Optimization for Machine Learning (NeurIPS 2020 Workshop)*, 2020.

[111] Xun Qian, Peter Richtárik, and Tong Zhang. Error compensated distributed SGD can be accelerated. *Advances in Neural Information Processing Systems*, 34, 2021.

[112] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

[113] Peter Richtárik, Igor Sokolov, Ilyas Fatkhullin, Elnur Gasanov, Zhize Li, and Eduard Gorbunov. 3PC: Three point compressors for communication-efficient distributed training and a better theory for lazy aggregation. *arXiv preprint arXiv:2202.00998*, 2022.

[114] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[115] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. FetchSGD: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning (ICML)*, pages 8253–8265. PMLR, 2020.

[116] César Sabater, Aurélien Bellet, and Jan Ramon. Distributed differentially private averaging with improved utility and robustness to malicious parties. 2020.

[117] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[118] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[119] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[120] Fanhua Shang, Tao Xu, Yuanyuan Liu, Hongying Liu, Longjie Shen, and Maoguo Gong. Differentially private admm algorithms for machine learning. *IEEE Transactions on Information Forensics and Security*, 16:4733–4745, 2021.

[121] Egor Shulgin and Peter Richtárik. Shifted compression framework: Generalizations and improvements. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[122] Navjot Singh, Deepesh Data, Jemin George, and Suhas N. Diggavi. SQuARM-SGD: Communication-efficient momentum SGD for decentralized optimization. In *IEEE International Symposium on Information Theory (ISIT)*. arXiv preprint arXiv:2005.07041.

[123] Virginia Smith, Simone Forte, Ma Chenxin, Martin Takáč, Michael I Jordan, and Martin Jaggi. CoCoA: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18:230, 2018.

[124] Sebastian U Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019.

[125] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 4447–4458, 2018.

[126] 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.

[127] Nikko Strom. Scalable distributed DNN training using commodity GPU cloud computing. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[128] Hang Su and Haoyu Chen. Experiments on parallel training of deep neural network using model averaging. *arXiv preprint arXiv:1507.01239*, 2015.

[129] Jun Sun, Tianyi Chen, Georgios Giannakis, and Zaiyue Yang. Communication-efficient distributed learning via lazily aggregated quantized gradients. *Advances in Neural Information Processing Systems*, 32:3370–3380, 2019.

[130] Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International Conference on Machine Learning*, pages 3329–3337. PMLR, 2017.

[131] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1139–1147. PMLR, 2013.

[132] Hanlin Tang, Chen Yu, Xiangru Lian, Tong Zhang, and Ji Liu. DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning (ICML)*, 2019.

[133] Aleksei Triastcyn, Matthias Reisser, and Christos Louizos. Dp-rec: Private & communication-efficient federated learning. *arXiv preprint arXiv:2111.05454*, 2021.

[134] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1195–1204. PMLR, 2019.

[135] Stephen A. Vavasis. Complexity issues in global optimization: A survey. In Horst R. and Pardalos P. M., editors, *Handbook of Global Optimization. Nonconvex Optimization and Its Applications*, volume 2. Springer, Boston, Massachusets, 1995.

[136] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[137] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[138] Hongyi Wang, Saurabh Agarwal, and Dimitris Papailiopoulos. Pufferfish: Communication-efficient models at no extra cost. *Proceedings of Machine Learning and Systems*, 3, 2021.

[139] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. ATOMO: Communication-efficient learning via atomic sparsification. *Advances in Neural Information Processing Systems (NeurIPS)*, 31:9850–9861, 2018.

[140] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[141] Jianyu Wang and Gauri Joshi. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *arXiv preprint arXiv:1810.08313*, 2018.

[142] 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.

[143] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving stochastic nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[144] Lun Wang, Ruoxi Jia, and Dawn Song. D2p-fed: Differentially private federated learning with efficient communication. *arXiv preprint arXiv:2006.13039*, 2020.

[145] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv*, 2018, 2018.

[146] 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. *arXiv preprint arXiv:1705.07878*, 2017.

[147] Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local SGD for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*, 2020.

[148] Cong Xie, Shuai Zheng, Oluwasanmi O Koyejo, Indranil Gupta, Mu Li, and Haibin Lin. CSER: Communication-efficient SGD with error reset. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[149] Hang Xu, Chen-Yu Ho, Ahmed M Abdelmoniem, Aritra Dutta, El Houcine Bergou, Konstantinos Karatsenidis, Marco Canini, and Panos Kalnis. Compressed communication for distributed deep learning: Survey and quantitative evaluation. Technical report, 2020.

[150] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[151] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

[152] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[153] Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016.

[154] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*, 2017.

[155] Xin Zhang, Minghong Fang, Jia Liu, and Zhengyuan Zhu. Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed sgd approach. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 261–270, 2020.

[156] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[157] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[158] Huixuan Zong, Qing Wang, Xiaofeng Liu, Yinchuan Li, and Yunfeng Shao. Communication reducing quantization for federated learning with local differential privacy mechanism. In *2021 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 75–80. IEEE, 2021.