A Unified Analysis of Stochastic Gradient Methods for Nonconvex Federated Optimization

Zhize Li
KAUST
zhize.li@kaust.edu.sa

Peter Richtárik
KAUST
peter.richtarik@kaust.edu.sa

Abstract

In this paper, we study the performance of a large family of SGD variants in the smooth nonconvex regime. To this end, we propose a generic and flexible assumption capable of accurate modeling of the second moment of the stochastic gradient. Our assumption is satisfied by a large number of specific variants of SGD in the literature, including SGD with arbitrary sampling, SGD with compressed gradients, and a wide variety of variance-reduced SGD methods such as SVRG and SAGA. We provide a single convergence analysis for all methods that satisfy the proposed unified assumption, thereby offering a unified understanding of SGD variants in the nonconvex regime instead of relying on dedicated analyses of each variant. Moreover, our unified analysis is accurate enough to recover or improve upon the best-known convergence results of several classical methods, and also gives new convergence results for many new methods which arise as special cases. In the more general distributed/federated nonconvex optimization setup, we propose two new general algorithmic frameworks differing in whether direct gradient compression (DC) or compression of gradient differences (DIANA) is used. We show that all methods captured by these two frameworks also satisfy our unified assumption. Thus, our unified convergence analysis also captures a large variety of distributed methods utilizing compressed communication. Finally, we also provide a unified analysis for obtaining faster linear convergence rates in this nonconvex regime under the PL condition.

1 Introduction

In this paper, we develop a general framework for studying and designing SGD-type methods for solving *nonconvex distributed/federated optimization problems* [24, 25, 18]. Given m machines/workers/devices, each having access to their own data samples, we consider the problem

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

in the heterogeneous (non-IID) data setting, i.e., we allow different workers to have access to different data distributions. We consider the case when the loss f_i in worker i is of an online/expectation form,

$$f_i(x):=\mathbb{E}_{\zeta\sim\mathcal{D}_i}[f_i(x,\zeta)], \tag{2}$$

and also the case when f_i is of a finite-sum form,

$$f_i(x):=\frac{1}{n}\sum_{j=1}^{n}f_{i,j}(x), \tag{3}$$

where $f(x), f_i(x), f_i(x,\zeta)$ and $f_{i,j}(x)$ are possibly nonconvex functions. Forms (2) and (3) capture the population (resp. empirical) risk minimization problems in distributed/federated learning.

NeurIPS Workshop on Scalability, Privacy, and Security in Federated Learning (NeurIPS-SpicyFL 2020).

Algorithm 1 Framework of stochastic gradient methods

Input: initial point x^0, stepsize η_k

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Compute stochastic gradient g^k
3: $x^{k+1} = x^k - \eta_k g^k$
4: **end for**

In particular, the single machine/node case (i.e., $m = 1$) of problem (1) reduces to the standard problem

$$\min_{x \in \mathbb{R}^d} f(x), \tag{4}$$

where $f(x)$ can be the online/expectation form

$$f(x) := \mathbb{E}_{\zeta \sim \mathcal{D}}[f(x, \zeta)] \tag{5}$$

or the finite-sum form

$$f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x), \tag{6}$$

where $f(x), f(x, \zeta)$ and $f_j(x)$ are possibly nonconvex functions. These forms capture the standard population/empirical risk minimization problems in machine learning.

There has been extensive research into solving the standard problem (4)–(6) and an enormous number of methods were proposed, e.g., [44, 43, 9, 17, 5, 45, 7, 40, 27–29, 2, 35, 10, 55, 6, 33, 46, 37]. Due to the increasing popularity of distributed/federated learning, the more general distributed/federated optimization problem (1)–(3) has attracted significant attention as well [25, 41, 39, 28, 31, 51, 42, 16, 20, 22, 54, 32, 18, 38, 23]. However, all these methods are analyzed separately, often using different approaches, intuitions, and assumptions, and separately in the $m = 1$ (single node) and $m \geq 1$ case.

1.1 Our contributions

We provide a *single and sharp analysis for a large family of SGD methods (Algorithm 1) for solving the nonconvex problem* (1). Our approach offers a *unified understanding* of many previously proposed SGD variants, which we believe helps the community making better sense of existing methods and results. More importantly, *our unified approach also motivates and facilitates the design of, and offers plug-in convergence guarantees for, many new and practically relevant SGD variants*.

While Algorithm 1 has a seemingly tame structure, the complication arises due to the fact that there is a potentially infinite number of meaningful and yet sharply distinct ways in which the gradient estimator g^k can be defined. The selection of an appropriate estimator is a very active and important area of research, as it directly impacts many aspects of the algorithm it gives rise to, including tractability, memory footprint, per iteration cost, parallelizability, iteration complexity, communication complexity, sample complexity and generalization.

The key technical idea of our approach is the design of a *flexible, tractable and accurate parametric model* capturing the behavior of the stochastic gradient. We want the model to be *flexible* in order to be able to describe many existing and have the potential to describe many variants of SGD. As we shall see, flexibility is achieved by the inclusion of a number of parameters. We want the model to be *tractable*, meaning that it needs to act as an assumption which can be used to perform a theoretical complexity analysis. Finally, we want the complexity results to be *accurate*, i.e., we want to recover best known rates for existing methods, and obtain sharp and useful rates with predictive power for new methods. Our parametric model is described in Assumption 1, and as we argue throughout the paper and its full version [36], it is indeed flexible, tractable and accurate.

Assumption 1 (Gradient estimator) *The gradient estimator g^k in Algorithm 1 is unbiased, i.e., $\mathbb{E}_k[g^k] = \nabla f(x^k)$, and there exist non-negative constants $A_1, A_2, B_1, B_2, C_1, C_2, D_1, \rho$ and a random sequence $\{\sigma_k^2\}$ such that the following two inequalities hold*

$$\mathbb{E}_k[\|g^k\|^2] \leq 2A_1(f(x^k) - f^*) + B_1\|\nabla f(x^k)\|^2 + D_1\sigma_k^2 + C_1, \tag{7}$$

$$\mathbb{E}_k[\sigma_{k+1}^2] \leq (1 - \rho)\sigma_k^2 + 2A_2(f(x^k) - f^*) + B_2\|\nabla f(x^k)\|^2 + C_2. \tag{8}$$

Table 1: Selected methods that fit our unified analysis framework for *nonconvex optimization* ($m = 1$, i.e., single node).

Problem	Assumption [1]	Algorithm	Convergence result	Recover
(4)	L-smooth	GD	Cor 1	[44]
(4) with (5) or (6)	L-smooth	SGD	Cor 2	[10, 21]
(4) with (6)	L-smooth	L-SVRG	Cor 3	[49, 3, 34, 48]
(4) with (6)	L-smooth	SAGA	Cor 4	[50]

Table 2: Selected methods that fit our unified analysis framework for *nonconvex distributed/federated optimization* ($m \geq 1$, i.e., any number of nodes).

Problem	Assumption	Algorithm	Convergence result	Recover
(1)	L-smooth	DC-GD	Cor 5	[21]
(1) with (2) or (3)	L-smooth	DC-SGD	Cor 6	[21, 15]
(1) with (3)	L-smooth	DC-LSVRG	Cor 7	**New**
(1) with (3)	L-smooth	DC-SAGA	Cor 8	**New**
(1)	L-smooth	DIANA-GD	Cor 9	**New**
(1) with (2) or (3)	L-smooth	DIANA-SGD	Cor 10	**New**
(1) with (3)	L-smooth	DIANA-LSVRG	Cor 11	**New**[†]
(1) with (3)	L-smooth	DIANA-SAGA	Cor 12	**New**[†]

[†]We want to mention that Horváth et al. [16] studied a weak version of DIANA-LSVRG and DIANA-SAGA with minibatch size $b = 1$ (non-minibatch version).

Table 3: Selected methods that fit our unified analysis framework for *nonconvex optimization under the PL condition* ($m = 1$).

Problem	Assumption	Algorithm	Convergence result	Recover
(4)	L-smooth, PL cond.	GD	Cor 13	[47, 19]
(4) with (5) or (6)	L-smooth, PL cond.	SGD	Cor 14	[21]
(4) with (6)	L-smooth, PL cond.	L-SVRG	Cor 15	[50, 34]
(4) with (6)	L-smooth, PL cond.	SAGA	Cor 16	[50]

Table 4: Selected methods that fit our unified analysis framework for *nonconvex distributed/federated optimization under PL condition* ($m \geq 1$).

Problem	Assumption	Algorithm	Convergence result	Recover
(1)	L-smooth, PL cond.	DC-GD	Cor 17	**New**
(1) with (2) or (3)	L-smooth, PL cond.	DC-SGD	Cor 18	**New**
(1) with (3)	L-smooth, PL cond.	DC-LSVRG	Cor 19	**New**
(1) with (3)	L-smooth, PL cond.	DC-SAGA	Cor 20	**New**
(1)	L-smooth, PL cond.	DIANA-GD	Cor 21	**New**
(1) with (2) or (3)	L-smooth, PL cond.	DIANA-SGD	Cor 22	**New**
(1) with (3)	L-smooth, PL cond.	DIANA-LSVRG	Cor 23	**New**
(1) with (3)	L-smooth, PL cond.	DIANA-SAGA	Cor 24	**New**

Flexibility: Our model for the behavior of the stochastic gradient for nonconvex optimization, as captured by Assumption 1, is satisfied by a large number of specific variants of SGD proposed in the literature, including SGD with arbitrary sampling [48, 12, 11, 21], SGD with compressed gradients [1, 53, 4, 24, 13, 15], and a wide variety of variance-reduced SGD methods such as SVRG [17], SAGA [5] and their variants (e.g., [14, 26, 49, 50, 3, 30, 34, 8, 42, 16]). Specific methods vary in the parameters for which recurrences (7) and (8) are satisfied. For example, SGD variants not employing variance reduction will generally have $D_1 = 0$, and recurrence (8) will not be used (i.e., we can ignore it and set $\rho = 1$, $A_2 = 0$, $B_2 = 0$ and $C_2 = 0$). This setting was considered in [21], and was an inspiration for our work. If variance reduction is applied, then $D_1 > 0$ and typically $C_1 = 0$, and recurrence (8) describes the variance reduction process, with parameter ρ describing the speed of variance reduction. If $C_2 > 0$, variance reduction is not perfect. If $C_2 = 0$ as well, then the methods will be fully variance reduced, which typically means faster convergence rate. The specific values of

[1]Due to the space limit, the detailed assumptions, algorithm descriptions, and corollaries of convergence result (note that here the indices of corollaries are corresponding to those in [36]) listed in Tables 1–4 can be found in the full version of this paper [36]. Besides, '**New**' in the last column means that we obtain the first convergence result for those cases that no previous results exist.

all the parameters depend on how the stochastic gradient g^k is constructed (e.g., via minibatching, importance sampling, variance reduction, perturbation, compression).

We design several new methods, with gradient estimators that fit Assumption 1, for solving the general nonconvex distributed/federated problem (1)–(3) using compressed (e.g., quantized or sparsified) gradient communication, which is of import when training distributed deep learning models. We adopt a direct compression (DC) framework [1, 24], and a compression of gradient differences framework (DIANA) [42, 16]. We develop several new specific methods belonging to the DC framework (Algorithm 2) and DIANA framework (Algorithm 3), show that they all satisfy Assumption 1, and thus are also captured by our unified analysis.

Tractability: We use our unified assumption to prove four complexity theorems: Theorems 1, 2, 3, and 4. Theorem 1 is the main theorem, and Theorem 2 is used to obtain sharper results under the PL condition. Theorems 3 and 4 are used in combination with the previous generic Theorems 1 and 2 to obtain specialized results for distributed/federated optimization utilizing either direct gradient compression (DC framework (Algorithm 2)), or compression of gradient differences (DIANA framework (Algorithm 3)), respectively. In Tables 1–4 we visualize how these theorems lead to corollaries which describe the detailed complexity results of various existing and new methods.

Accuracy: For all existing methods, the rates we obtain using our general analysis match the best known rates. We also obtain the first results for some cases where no previous results exist.

2 Unified Main Theorem

In this section we first provide our main unified complexity result (Theorem 1) for a large family of SGD methods (Algorithm 1) under the general parametric assumption (Assumption 1). Subsequently, we also provide a unified result (Theorem 2) in the case when the PL condition (Assumption **??**) is satisfied. Note that under the PL condition, one can obtain a faster linear convergence $O(\cdot \log \frac{1}{\epsilon})$ (see Theorem 2) rather than the sublinear convergence $O(\cdot \frac{1}{\epsilon^2})$ (see Theorem 1).

Theorem 1 (Main theorem) *Suppose that Assumptions 1 and 2 hold. If the stepsize $\eta_k \leq \frac{1}{LB_1+LD_1B_2\rho^{-1}}$, then for any $K \geq 1$, the iterates generated by Algorithm 1 satisfies*

$$\mathbb{E}[\|\nabla f(\widehat{x}^K)\|^2] \leq \frac{2\Delta_0'\beta^K}{\eta \sum_{k=0}^{K-1} \beta^{K-1-k}} + \eta L(C_1 + D_1C_2\rho^{-1}). \tag{9}$$

where $\widehat{x}^K$ randomly chosen from $\{x^k\}_{k=0}^{K-1}$ with probability $p_k = \frac{\beta^{K-1-k}}{\sum_{k=0}^{K-1}\beta^{K-1-k}}$ for x^k, $\beta := 1 + L\eta^2(A_1 + D_1A_2\rho^{-1})$ and $\Delta_0' := f(x^0) - f^* + 2^{-1}L\eta^2 D_1\rho^{-1}\sigma_0^2$.

In particular, if $A_1 + D_1A_2\rho^{-1} = 0$ (thus $\beta = 1$) and use the fixed stepsize

$$\eta_k \equiv \eta \leq \min\left\{\frac{1}{L(B_1+D_1B_2\rho^{-1})},\ \frac{\epsilon^2}{2L(C_1+D_1C_2\rho^{-1})}\right\},$$

then the number of iterations performed by Algorithm 1 to find an ϵ-solution, i.e., a point $\widehat{x}^K$ such that $\mathbb{E}[\|\nabla f(\widehat{x}^K)\|] \leq \epsilon$, can be bounded by

$$K = \frac{4\Delta_0' L}{\epsilon^2}\max\left\{B_1 + D_1B_2\rho^{-1},\ \frac{2(C_1+D_1C_2\rho^{-1})}{\epsilon^2}\right\},$$

Theorem 2 (Main theorem under PL condition) *Suppose that Assumptions 1, 2 and 8 hold. Set the stepsize as*

$$\eta_k = \begin{cases} \eta & \text{if } k \leq \frac{K}{2} \\ \frac{2\eta}{2+(k-\frac{K}{2})\mu\eta} & \text{if } k > \frac{K}{2} \end{cases}, \text{ where } \eta \leq \frac{1}{LB_1+2LD_1B_2\rho^{-1}+(LA_1+2LD_1A_2\rho^{-1})\mu^{-1}}$$

and let $\Delta_0' := f(x^0) - f^* + L\eta^2 D_1\rho^{-1}\sigma_0^2$ and $\kappa := \frac{L}{\mu}$. Then the number of iterations performed by Algorithm 1 to find an ϵ-solution, i.e., a point x^K such that $\mathbb{E}[f(x^K) - f^*] \leq \epsilon$, can be bounded by

$$K = \max\left\{2\left(B_1 + 2D_1B_2\rho^{-1} + (LA_1 + 2LD_1A_2\rho^{-1})\mu^{-1}\right)\kappa\log\frac{2\Delta_0'}{\epsilon},\ \frac{10(C_1+2D_1C_2\rho^{-1})\kappa}{\mu\epsilon}\right\}.$$

In the full version of this paper [36], we show that many specific methods, existing and new, satisfy our unified Assumption 1 and can thus be captured by our unified analysis (i.e., Theorems 1 and 2). We can thus plug their corresponding parameters (i.e., specific values for $A_1, A_2, B_1, B_2, C_1, C_2, D_1, \rho$) into our unified Theorems 1 and 2 to obtain detailed convergence rates for these methods. For example, the gradient estimator $g^k = \nabla f(x^k)$ in standard GD method, then it is easy to see that g^k satisfies the unified Assumption 1 with $A_1 = C_1 = D_1 = 0,\ B_1 = 1,\ \sigma_k^2 \equiv 0,\ \rho = 1,\ A_2 = B_2 = C_2 = 0$. See Tables 1 and 3 for an overview.

3 General Nonconvex Federated Optimization Problems

In this section, we consider the more general nonconvex distributed/federated problem (1) with online form (2) or finite-sum form (3) (i.e., any number of nodes $m \geq 1$). Here we allow different nodes/machines to have different data distributions, i.e., we consider the non-IID (heterogeneous) data setting. Note that in distributed/federated problems, the bottleneck usually is the communication cost among workers, which motivates the study of methods which employ *compressed* communication.

Definition 1 (Compression operator) *A randomized map $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an ω-compression operator if*

$$\mathbb{E}[\mathcal{C}(x)] = x, \qquad \mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq \omega\|x\|^2, \qquad \forall x \in \mathbb{R}^d. \tag{10}$$

In particular, no compression ($\mathcal{C}(x) \equiv x$) implies $\omega = 0$.

Note that (10) holds for many practical compression methods, e.g., random sparsification [52], quantization [1], natural compression [15]. We are not going to focus on any specific compression operator; instead, we will analyze our methods for any compression operator captured by Def 1.

In the following, we provide two general algorithmic frameworks differing in whether direct gradient compression (DC) or compression of gradient differences (DIANA) is used. Previous approaches mostly focus on strongly convex or convex problems for specific instances of SGD. Ours is the first unified analysis covering many variants of SGD in a single theorem, covering the nonconvex regime. In fact, many specific SGD methods arising as special cases of our general approach have not been analyzed before. See Tables 2 and 4 for an overview.

3.1 DC framework for nonconvex federated optimization

In the direct compression (DC) framework studied in this section, each machine $i \in [m]$ computes its local stochastic gradient $\widetilde{g}_i^k$, subsequently applies to it a compression operator $\mathcal{C}_i^k$, and communicates the compressed vector to a server, or to all other nodes (see Algorithm 2).

Algorithm 2 DC framework of stochastic gradient methods for nonconvex federated optimization

Input: initial point x^0, stepsizes η_k

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: **for all machines** $i = 1, 2, \ldots, m$ **do in parallel**
3: Compute local stochastic gradient $\widetilde{g}_i^k$
4: Compress local gradient $\mathcal{C}_i^k(\widetilde{g}_i^k)$ and send it to the server
5: **end for**
6: Aggregate received compressed gradient information: $g^k = \frac{1}{m}\sum_{i=1}^m \mathcal{C}_i^k(\widetilde{g}_i^k)$
7: $x^{k+1} = x^k - \eta_k g^k$
8: **end for**

Our main theoretical result describing the convergence properties of Algorithm 2 is stated below.

Theorem 3 (DC framework) *If the local stochastic gradient $\widetilde{g}_i^k$ (see Line 3 of Algorithm 2) satisfies the recursions*

$$\mathbb{E}_k[\|\widetilde{g}_i^k\|^2] \leq 2A_{1,i}(f_i(x^k) - f_i^*) + B_{1,i}\|\nabla f_i(x^k)\|^2 + D_{1,i}\sigma_{k,i}^2 + C_{1,i}, \tag{11}$$

$$\mathbb{E}_k[\sigma_{k+1,i}^2] \leq (1-\rho_i)\sigma_{k,i}^2 + 2A_{2,i}(f(x^k) - f^*) + B_{2,i}\|\nabla f(x^k)\|^2 + D_{2,i}\mathbb{E}_k[\|g^k\|^2] + C_{2,i}, \tag{12}$$

then g^k (see Line 6 of Algorithm 2) satisfies the unified Assumption 1 with

$$A_1 = \frac{(1+\omega)A}{m}, \quad B_1 = 1, \quad D_1 = \frac{1+\omega}{m}, \quad \sigma_k^2 = \frac{1}{m}\sum_{i=1}^m D_{1,i}\sigma_{k,i}^2, \quad C_1 = \frac{(1+\omega)C}{m},$$

$$\rho = \min_i \rho_i - \tau, \quad A_2 = D_A + \tau A, \quad B_2 = D_B + D_D, \quad C_2 = D_C + \tau C,$$

where $A := \max_i(A_{1,i} + B_{1,i}L_i - L_i/(1+\omega))$, $C := \frac{1}{m}\sum_{i=1}^m C_{1,i} + 2A\Delta_f^*$, $\Delta_f^* := f^* - \frac{1}{m}\sum_{i=1}^m f_i^*$, $\tau := \frac{(1+\omega)D_D}{m}$, $D_A := \frac{1}{m}\sum_{i=1}^m D_{1,i}A_{2,i}$, $D_B := \frac{1}{m}\sum_{i=1}^m D_{1,i}B_{2,i}$, $D_D := \frac{1}{m}\sum_{i=1}^m D_{1,i}D_{2,i}$, *and* $D_C := \frac{1}{m}\sum_{i=1}^m D_{1,i}C_{2,i}$.

The above result means that, as long as the local gradient estimators $\widetilde{g}_i^k$ used in the DC framework (Algorithm 2) satisfy recursions (11)–(12), the global gradient estimator g^k satisfies our unified Assumption 1, and thus we can plug their corresponding parameters (i.e., specific values for $A_1, A_2, B_1, B_2, C_1, C_2, D_1, \rho$) into our unified Theorems 1 and 2 to obtain the detailed convergence rates for DC-type methods in the DC framework. To showcase the generality and expressive power of our DC framework, we indeed describe several particular ways (e.g., GD, SGD, L-SVRG, SAGA) in which such local gradient estimators can be generated, each leading to a particular instance DC-type method. See Tables 2 and 4 for an overview.

3.2 DIANA framework for nonconvex federated optimization

We now highlight an inherent issue of the DC framework (Algorithm 2), which will serve as a motivation for the proposed DIANA framework described here. Considering any stationary point $\widehat{x}$ such that $\nabla f(\widehat{x}) = \sum_{i=1}^m \nabla f_i(\widehat{x}) = 0$, the aggregated compressed gradient (even if the full gradient is used locally, i.e., $\widetilde{g}_i^k = \nabla f_i(x^k)$), is *not* equal to 0, i.e., $g(\widehat{x}) = \frac{1}{m}\sum_{i=1}^m \mathcal{C}_i(\nabla f_i(\widehat{x})) \neq 0$. This effect slows down convergence of the methods in DC framework. To address this issue, we use the DIANA framework to compress the *gradient differences* instead (see Line 4 of Algorithm 3).

Algorithm 3 DIANA framework of stochastic gradient methods for nonconvex federated optimization

Input: initial point x^0, $\{h_i^0\}_{i=1}^m$, $h^0 = \frac{1}{m}\sum_{i=1}^m h_i^0$, stepsize parameters η_k, α_k

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: **for all machines** $i = 1, 2, \ldots, m$ **do in parallel**
3: Compute local stochastic gradient $\widetilde{g}_i^k$
4: Compress shifted local gradient $\widehat{\Delta}_i^k = \mathcal{C}_i^k(\widetilde{g}_i^k - h_i^k)$ and send $\widehat{\Delta}_i^k$ to the server
5: Update local shift $h_i^{k+1} = h_i^k + \alpha_k \mathcal{C}_i^k(\widetilde{g}_i^k - h_i^k)$
6: **end for**
7: Aggregate received compressed gradient information: $g^k = h^k + \frac{1}{m}\sum_{i=1}^m \widehat{\Delta}_i^k$
8: $x^{k+1} = x^k - \eta_k g^k$
9: $h^{k+1} = h^k + \alpha_k \frac{1}{m}\sum_{i=1}^m \widehat{\Delta}_i^k$
10: **end for**

Similarly, we show that the gradient estimator g^k also satisfies our unified Assumption 1 and thus can also be captured by our unified analysis.

Theorem 4 (DIANA framework) *Suppose that the local stochastic gradient $\widetilde{g}_i^k$ (see Line 3 of Algorithm 3) satisfies* (11)–(12)*, same as in the DC framework. Then g^k (see Line 7 of Algorithm 3) satisfies the unified Assumption 1 with*

$$A_1 = \frac{(1+\omega)A}{m}, \; B_1 = 1, \; D_1 = \frac{1+\omega}{m}, \; \sigma_k^2 = \frac{1}{m}\sum_{i=1}^m D_{1,i}\sigma_{k,i}^2 + \frac{\omega}{(1+\omega)m}\sum_{i=1}^m \|\nabla f_i(x^k) - h_i^k\|^2,$$

$$C_1 = \frac{(1+\omega)C}{m}, \quad \rho = \min\left\{\min_i \rho_i - \tau, \; 2\alpha - (1-\alpha)\beta^{-1} - \alpha^2 - \tau\right\},$$

$$A_2 = D_A + \tau A, \quad B_2 = D_B + B, \quad C_2 = D_C + \tau C,$$

where $A := \max_i(A_{1,i} + (B_{1,i}-1)L_i)$, $B := \frac{\omega(1+\beta)L^2\eta^2}{1+\omega} + D_D$, $C := \frac{1}{m}\sum_{i=1}^m C_{1,i} + 2A\Delta_f^*$, $\Delta_f^* := f^* - \frac{1}{m}\sum_{i=1}^m f_i^*$, $\tau := \alpha^2\omega + \frac{(1+\omega)B}{m}$, $D_A := \frac{1}{m}\sum_{i=1}^m D_{1,i}A_{2,i}$, $D_B := \frac{1}{m}\sum_{i=1}^m D_{1,i}B_{2,i}$, $D_D := \frac{1}{m}\sum_{i=1}^m D_{1,i}D_{2,i}$, $D_C := \frac{1}{m}\sum_{i=1}^m D_{1,i}C_{2,i}$, *and* $\forall \beta > 0$.

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*, pages 1709–1720, 2017.

[2] Zeyuan Allen-Zhu. Katyusha: the first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.

[3] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707, 2016.

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

[5] 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.

[6] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 687–697, 2018.

[7] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[8] 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, 2019.

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

[10] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155 (1-2):267–305, 2016.

[11] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. *arXiv preprint arXiv:1905.11261*, 2019.

[12] 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, 2019.

[13] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: variance reduction via gradient sketching. In *Advances in Neural Information Processing Systems 31*, pages 2082–2093, 2018.

[14] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313. 2015.

[15] Samuel Horváth, Chen-Yu Ho, Ludovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

[16] 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.

[17] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[18] 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.

[19] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[20] 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.

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

[22] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local GD on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019.

[23] 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*, 2020.

[24] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

[25] 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.

[26] 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. *arXiv preprint arXiv:1901.08689*, 2019.

[27] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *arXiv preprint arXiv:1507.02000*, 2015.

[28] Guanghui Lan and Yi Zhou. Random gradient extrapolation for distributed and stochastic optimization. *SIAM Journal on Optimization*, 28(4):2753–2782, 2018.

[29] Guanghui Lan, Zhize Li, and Yi Zhou. A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, pages 10462–10472, 2019.

[30] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2345–2355, 2017.

[31] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[32] Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication efficient decentralized training with multiple local updates. *arXiv preprint arXiv:1910.09126*, 2019.

[33] Zhize Li. SSRGD: Simple stochastic recursive gradient descent for escaping saddle points. In *Advances in Neural Information Processing Systems*, pages 1521–1531, 2019.

[34] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 5569–5579, 2018.

[35] Zhize Li and Jian Li. A fast Anderson-Chebyshev acceleration for nonlinear optimization. In *International Conference on Artificial Intelligence and Statistics*, 2020.

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

[37] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. *arXiv preprint arXiv:2008.10898*, 2020.

[38] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. In *International Conference on Machine Learning*, 2020.

[39] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

[40] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

[41] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

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

[43] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.

[44] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, 2004.

[45] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621, 2017.

[46] Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1902.05679*, 2019.

[47] Boris T Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[48] Xun Qian, Zheng Qu, and Peter Richtárik. L-SVRG and L-Katyusha with arbitrary sampling. *arXiv preprint arXiv:1906.01481*, 2019.

[49] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pages 314–323, 2016.

[50] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems*, pages 1145–1153, 2016.

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

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

[53] 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*, pages 1509–1519, 2017.

[54] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[55] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3925–3936, 2018.