MARINA: Faster Non-Convex Distributed Learning with Compression

Eduard Gorbunov [1 2 3] **Konstantin Burlachenko** [3] **Zhize Li** [3] **Peter Richtárik** [3]

Abstract

We develop and analyze MARINA: a new communication efficient method for non-convex distributed learning over heterogeneous datasets. MARINA employs a novel communication compression strategy based on the compression of gradient differences that is reminiscent of but different from the strategy employed in the DIANA method of Mishchenko et al. (2019). Unlike virtually all competing distributed first-order methods, including DIANA, ours is based on a carefully designed *biased* gradient estimator, which is the key to its superior theoretical and practical performance. The communication complexity bounds we prove for MARINA are evidently better than those of all previous first-order methods. Further, we develop and analyze two variants of MARINA: VR-MARINA and PP-MARINA. The first method is designed for the case when the local loss functions owned by clients are either of a finite sum or of an expectation form, and the second method allows for a partial participation of clients – a feature important in federated learning. All our methods are superior to previous state-of-the-art methods in terms of oracle/communication complexity. Finally, we provide a convergence analysis of all methods for problems satisfying the Polyak-Łojasiewicz condition.

1. Introduction

Non-convex optimization problems appear in various applications of machine learning, such as training deep neural networks (Goodfellow et al., 2016) and matrix completion and recovery (Ma et al., 2018; Bhojanapalli et al., 2016). Because of their practical importance, these problems gained much attention in recent years, which led to a rapid development of new efficient methods for non-convex optimization problems (Danilova et al., 2020), and especially the training of deep learning models (Sun, 2019).

Training deep neural networks is notoriously computationally challenging and time-consuming. In the quest to improve the generalization performance of modern deep learning models, practitioners resort to using increasingly larger datasets in the training process, and to support such workloads, it is imperative to use advanced parallel and distributed hardware, systems, and algorithms. Distributed computing is often necessitated by the desire to train models from data naturally distributed across several edge devices, as is the case in federated learning (Konečný et al., 2016; McMahan et al., 2017). However, even when this is not the case, distributed methods are often very efficient at reducing the training time (Goyal et al., 2017; You et al., 2020). Due to these and other reasons, distributed optimization has gained immense popularity in recent years.

However, distributed methods almost invariably suffer from the so-called *communication bottleneck*: the communication cost of information necessary for the workers to jointly solve the problem at hand is often very high, and depending on the particular compute architecture, workload, and algorithm used, it can be orders of magnitude higher than the computation cost. A popular technique for resolving this issue is *communication compression* (Seide et al., 2014; Konečný et al., 2016; Suresh et al., 2017), which is based on applying a lossy transformation/compression to the models, gradients, or tensors to be sent over the network to save on communication. Since applying a lossy compression generally decreases the utility of the exchanged messages, such an approach will typically lead to an increase in the number of communications, and the overall usefulness of this technique manifests itself in situations where the communication savings are larger compared to the increased need for the number of communication rounds (Horváth et al., 2019).

The optimization and machine learning communities have exerted considerable effort in recent years to design distributed methods supporting compressed communication. From many methods proposed, we emphasize VR-DIANA (Horváth et al., 2019), FedCOMGATE (Haddadpour et al., 2020), and FedSTEPH (Das et al., 2020) because these pa-

[1]Moscow Institute of Physics and Technology, Moscow, Russia [2]Yandex, Moscow, Russia [3]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. Correspondence to: Eduard Gorbunov <eduard.gorbunov@phystech.edu>, Peter Richtárik <peter.richtarik@kaust.edu.sa>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

pers contain the state-of-the-art results in the setup when the local loss functions can be arbitrary heterogeneous.

1.1. Contributions

We propose several new distributed optimization methods supporting compressed communication, specifically focusing on smooth but nonconvex problems of the form

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

where n workers/devices/clients/peers are connected in a centralized way with a parameter-server, and client i has an access to the local loss function f_i only. We establish strong complexity rates for them and show that they are better than previous state-of-the-art results.

• **MARINA.** The main contribution of our paper is a new distributed method supporting communication compression called MARINA (Alg 1). In this algorithm, workers apply an unbiased compression operator to the *gradient differences* at each iteration with some probability and send them to the server that performs aggregation by averaging. Unlike all known methods operating with unbiased compression operators, this procedure leads to a *biased* gradient estimator. We prove convergence guarantees for MARINA, which are strictly better than previous state-of-the-art methods (see Table 1). For example, MARINA's rate $\mathcal{O}(\frac{1+\omega/\sqrt{n}}{\varepsilon^2})$ is $\mathcal{O}(\sqrt{\omega})$ times better than that of the state-of-the-art method DIANA (Mishchenko et al., 2019), where ω is the variance parameter associated with the deployed compressor. For example, in the case of the Rand1 sparsification compressor, we have $\omega = d-1$, and hence we get an improvement by the factor $\mathcal{O}(\sqrt{d})$. Since the number d of features can be truly very large when training modern models, this is a substantial improvement that can even amount to *several orders of magnitude.*

• **Variance Reduction on Nodes.** We generalize MARINA to VR-MARINA, which can handle the situation when the local functions f_i have either a finite-sum (each f_i is an average of m functions) or an expectation form, and when it is more efficient to rely on local stochastic gradients rather than on local gradients. When compared with MARINA, VR-MARINA additionally performs *local variance reduction* on all nodes, progressively removing the variance coming from the stochastic approximation, leading to a better oracle complexity than previous state-of-the-art results (see Table 1). When no compression is used (i.e., $\omega = 0$), the rate of VR-MARINA is $\mathcal{O}(\frac{\sqrt{m}}{\sqrt{n}\varepsilon^2})$, while the rate of the state-of-the-art method VR-DIANA is $\mathcal{O}(\frac{m^{2/3}}{\varepsilon^2})$. This is an improvement by the factor $\mathcal{O}(\sqrt{n}m^{1/6})$. When much compression is applied, and ω is large, our method is faster by the factor $\mathcal{O}(\frac{m^{2/3}+\omega}{m^{1/2}+\omega^{1/2}})$. In the special case, when there is just a single node ($n = 1$), and no compression is used, VR-MARINA reduces to the PAGE method of Li et al. (2020); this is an optimal first-order algorithm for smooth non-convex finite-sum/online optimization problems.

• **Partial Participation.** We develop a modification of MARINA allowing for *partial participation* of the clients, which is a feature critical in federated learning. The resulting method, PP-MARINA, has superior communication complexity to the existing methods developed for this settings (see Table 1).

• **Convergence Under the Polyak-Łojasiewicz Condition.** We analyze all proposed methods for problems satisfying the Polyak-Łojasiewicz condition (Polyak, 1963; Łojasiewicz, 1963). Again, the obtained results are strictly better than previous ones (see Table 2). Statements and proofs of all these results are in the Appendix.

• **Simple Analysis.** The simplicity and flexibility of our analysis offer several extensions. For example, one can easily generalize our analysis to the case of different quantization operators and different batch sizes used by clients. Moreover, one can combine the ideas of VR-MARINA and PP-MARINA and obtain a single distributed algorithm with compressed communications, variance reduction on nodes, and clients' sampling. We did not do this to keep the exposition simpler.

1.2. Related Work

Non-Convex Optimization. Since finding a global minimum of a non-convex function is, in general, an NP-hard problem (Murty & Kabadi, 1987), many researchers in non-convex optimization focus on relaxed goals such as finding an ε-stationary point. The theory of stochastic first-order methods for finding ε-stationary points is well-developed: it contains lower bounds for expectation minimization without smoothness of stochastic realizations (Arjevani et al., 2019) and for finite-sum/expectation minimization (Fang et al., 2018; Li et al., 2020) as well as optimal methods matching the lower bounds (see (Danilova et al., 2020; Li et al., 2020) for the overview). Recently, distributed variants of such methods were proposed (Sun et al., 2020; Sharma et al., 2019; Khanduri et al., 2020).

Compressed Communications. Works on distributed methods supporting communication compression can be roughly split into two large groups: the first group focuses on methods using *unbiased* compression operators (which refer to as quantizations in this paper), such as RandK, and the second one studies methods using *biased* compressors such as TopK. One can find a detailed summary of the most popular compression operators in (Safaryan et al., 2020; Beznosikov et al., 2020).

Unbiased Compression. In this line of work, the first con-

Table 1: Summary of the state-of-the-art results for finding an ε**-stationary point** for the problem (1), i.e., such a point $\hat{x}$ that $\mathbf{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \varepsilon^2$. Dependences on the numerical constants, "quality" of the starting point, and smoothness constants are omitted in the complexity bounds. Abbreviations: "PP" = partial participation; "Communication complexity" = the number of communications rounds needed to find an ε-stationary point; "Oracle complexity" = the number of (stochastic) first-order oracle calls needed to find an ε-stationary point. Notation: ω = the quantization parameter (see Def. 1.1); n = the number of nodes; m = the size of the local dataset; r = (expected) number of clients sampled at each iteration; b' = the batchsize for VR-MARINA at the iterations with compressed communication. To simplify the bounds, we assume that the expected density $\zeta_{\mathcal{Q}}$ of the quantization operator $\mathcal{Q}$ (see Def. 1.1) satisfies $\omega + 1 = \Theta(d/\zeta_{\mathcal{Q}})$ (e.g., this holds for RandK and ℓ_2-quantization, see (Beznosikov et al., 2020)). We notice that (Haddadpour et al., 2020) and (Das et al., 2020) contain also better rates under different assumptions on clients' similarity.

Setup	Method	Citation	Communication Complexity	Oracle Complexity
(1)	DIANA	(Mishchenko et al., 2019) (Horváth et al., 2019) (Li & Richtárik, 2020)	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2}$	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2}$
	FedCOMGATE [(1)]	(Haddadpour et al., 2020)	$\frac{1+\omega}{\varepsilon^2}$	$\frac{1+\omega}{n\varepsilon^4}$
	FedSTEPH, $r = n$	(Das et al., 2020)	$\frac{1+\omega/n}{\varepsilon^4}$	$\frac{1+\omega/n}{\varepsilon^4}$
	MARINA (Alg. 1)	Thm. 2.1 & Cor. 2.1 (NEW)	$\frac{1+\omega/\sqrt{n}}{\varepsilon^2}$	$\frac{1+\omega/\sqrt{n}}{\varepsilon^2}$
(1)+(5)	DIANA	(Li & Richtárik, 2020)	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$
	VR-DIANA	(Horváth et al., 2019)	$\frac{\left(m^{2/3}+\omega\right)\sqrt{1+\omega/n}}{\varepsilon^2}$	$\frac{\left(m^{2/3}+\omega\right)\sqrt{1+\omega/n}}{\varepsilon^2}$
	VR-MARINA (Alg. 2), $b' = 1$ [(2)]	Thm. 3.1 & Cor. 3.1 (NEW)	$\frac{1+\max\left\{\omega,\sqrt{(1+\omega)m}\right\}/\sqrt{n}}{\varepsilon^2}$	$\frac{1+\max\left\{\omega,\sqrt{(1+\omega)m}\right\}/\sqrt{n}}{\varepsilon^2}$
(1)+(6)	DIANA [(3)]	(Mishchenko et al., 2019) (Li & Richtárik, 2020)	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$
	FedCOMGATE [(3)]	(Haddadpour et al., 2020)	$\frac{1+\omega}{\varepsilon^2}$	$\frac{1+\omega}{n\varepsilon^4}$
	VR-MARINA (Alg. 2), $b' = 1$	Thm. 3.2 & Cor. 3.2 (NEW)	$\frac{1+\omega/\sqrt{n}}{\varepsilon^2} + \frac{\sqrt{1+\omega}}{n\varepsilon^3}$	$\frac{1+\omega/\sqrt{n}}{\varepsilon^2} + \frac{\sqrt{1+\omega}}{n\varepsilon^3}$
	VR-MARINA (Alg. 2), $b' = \Theta\left(\frac{1}{n\varepsilon^2}\right)$	Thm. 3.2 & Cor. 3.2 (NEW)	$\frac{1+\omega/\sqrt{n}}{\varepsilon^2}$	$\frac{1+\omega/\sqrt{n}}{n\varepsilon^4} + \frac{1+\omega}{n\varepsilon^3}$
PP, (1)	FedSTEPH	(Das et al., 2020)	$\frac{1+\omega/n}{r\varepsilon^4} + \frac{(1+\omega)(n-r)}{r(n-1)\varepsilon^4}$	$\frac{1+\omega/n}{r\varepsilon^4} + \frac{(1+\omega)(n-r)}{r(n-1)\varepsilon^4}$
	PP-MARINA (Alg. 4)	Thm. 4.1 & Cor. 4.1 (NEW)	$\frac{1+(1+\omega)\sqrt{n/r}}{\varepsilon^2}$	$\frac{1+(1+\omega)\sqrt{n/r}}{\varepsilon^2}$

[(1)] The results for FedCOMGATE are derived under assumption that for all vectors $x_1, \ldots, x_n \in \mathbb{R}^d$ the quantization operator $\mathcal{Q}$ satisfies $\mathbf{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \mathcal{Q}(x_j)\right\|^2 - \left\|\mathcal{Q}\left(\frac{1}{n}\sum_{i=1}^n x_j\right)\right\|^2\right] \leq G$ for some constant $G \geq 0$. In fact, this assumption does not hold for classical quantization operators like RandK and ℓ_2-quantization on $\mathbb{R}^d$. The counterexample: $n = 2$ and $x_1 = -x_2 = (t, t, \ldots, t)^\top$ with arbitrary large $t > 0$.
[(2)] One can even further improve the communication complexity by increasing b'.
[(3)] No assumptions on the smoothness of the stochastic realizations $f_\xi(x)$ are used.

vergence result in the non-convex case was obtained by Alistarh et al. (2017) for QSGD, under assumptions that the local loss functions are the same for all workers, and the stochastic gradient has uniformly bounded second moment. After that, Mishchenko et al. (2019) proposed DIANA (and its momentum version) and proved its convergence rate for non-convex problems without any assumption on the boundedness of the second moment of the stochastic gradient, but under the assumption that the dissimilarity between local loss functions is bounded. This restriction was later eliminated by Horváth et al. (2019) for the variance reduced version of DIANA called VR-DIANA, and the analysis was extended to a large class of unbiased compressors. Finally, the results for QSGD and DIANA were recently generalized and tightened by Li & Richtárik (2020) in a unifying framework that included many other methods as well.

Biased Compression. Biased compression operators are less "optimization-friendly" than unbiased ones. Indeed, one can construct a simple convex quadratic problem for which distributed SGD with Top1 compression diverges exponentially fast (Beznosikov et al., 2020). However, this issue can be resolved using *error compensation* (Seide et al., 2014). The first analysis of error-compensated SGD (EC-SGD) for non-convex problems was obtained by Karimireddy et al. (2019) for homogeneous problems under the assumption that the second moment of the stochastic gradient is uniformly bounded. The last assumption was recently removed from the analysis of EC-SGD by Stich & Karimireddy (2020); Beznosikov et al. (2020), while the first results without the homogeneity assumption were obtained by Koloskova et al. (2020a) for Choco-SGD, but still under the assumption that the second moment of the stochastic gradient is uniformly bounded. This issue was resolved by Beznosikov et al. (2020). In general, the current understanding of optimization methods with biased compressors is far from complete: even in the strongly convex case, the first linearly converging (Gorbunov et al., 2020) and accelerated (Qian et al., 2020) error-compensated stochastic methods were proposed just recently.

Other Approaches. Besides communication compression, there are also different techniques aiming to reduce the overall communication cost of distributed methods. The most popular ones are based on decentralized communications and multiple local steps between communication rounds, where the second technique is very popular in federated learning (Konečný et al., 2016; Kairouz et al., 2019).

Table 2: Summary of the state-of-the-art results for finding an ε-solution for the problem (1) satifying **Polyak-Łojasiewicz condition** (see As. 2.1), i.e., such a point $\hat{x}$ that $\mathbf{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$. Dependences on the numerical constants and $\log(1/\varepsilon)$ factors are omitted and all smoothness constanst are denoted by L in the complexity bounds. Abbreviations: "PP" = partial participation; "Communication complexity" = the number of communications rounds needed to find an ε-stationary point; "Oracle complexity" = the number of (stochastic) first-order oracle calls needed to find an ε-stationary point. Notation: ω = the quantization parameter (see Def. 1.1); n = the number of nodes; m = the size of the local dataset; r = (expected) number of clients sampled at each iteration; b' = the batchsize for VR-MARINA at the iterations with compressed communication. To simplify the bounds, we assume that the expected density $\zeta_{\mathcal{Q}}$ of the quantization operator $\mathcal{Q}$ (see Def. 1.1) satisfies $\omega + 1 = \Theta(d/\zeta_{\mathcal{Q}})$ (e.g., this holds for RandK and ℓ_2-quantization, see (Beznosikov et al., 2020)). We notice that (Haddadpour et al., 2020) and (Das et al., 2020) contain also better rates under different assumptions on clients' similarity.

Setup	Method	Citation	Communication Complexity	Oracle Complexity
(1)	DIANA	(Li & Richtárik, 2020)	$\frac{L(1+(1+\omega)\sqrt{\omega/n})}{\mu}$	$\frac{L(1+(1+\omega)\sqrt{\omega/n})}{\mu}$
	FedCOMGATE [(1)]	(Haddadpour et al., 2020)	$\frac{L(1+\omega)}{\mu}$	$\frac{L(1+\omega)}{n\mu\varepsilon}$
	MARINA (Alg. 1)	Thm. 2.2 & Cor. C.2 (NEW)	$\omega + \frac{L(1+\omega/\sqrt{n})}{\mu}$	$\omega + \frac{L(1+\omega/\sqrt{n})}{\mu}$
(1)+(5)	DIANA	(Li & Richtárik, 2020)	$\frac{L(1+(1+\omega)\sqrt{\omega/n})}{\mu} + \frac{L(1+\omega)}{n\mu}\left(\frac{L}{\mu}+\frac{1}{\varepsilon}\right)$	$\frac{L(1+(1+\omega)\sqrt{\omega/n})}{\mu} + \frac{L(1+\omega)}{n\mu}\left(\frac{L}{\mu}+\frac{1}{\varepsilon}\right)$
	VR-DIANA	(Li & Richtárik, 2020)	$\frac{L\left(m^{2/3}+\omega\right)\sqrt{1+\omega/n}}{\mu}$	$\frac{L\left(m^{2/3}+\omega\right)\sqrt{1+\omega/n}}{\mu}$
	VR-MARINA (Alg. 2), $b' = 1$ [(2)]	Thm. D.2 & Cor. D.2 (NEW)	$\omega + m + \frac{L(1+\max\left\{\omega,\sqrt{(1+\omega)m}\right\}/\sqrt{n})}{\mu}$	$\omega + m + \frac{L(1+\max\left\{\omega,\sqrt{(1+\omega)m}\right\}/\sqrt{n})}{\mu}$
(1)+(6)	DIANA [(3)]	(Mishchenko et al., 2019) (Li & Richtárik, 2020)	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$	$\frac{1+(1+\omega)\sqrt{\omega/n}}{\varepsilon^2} + \frac{1+\omega}{n\varepsilon^4}$
	FedCOMGATE [(3)]	(Haddadpour et al., 2020)	$\frac{L(1+\omega)}{\mu}$	$\frac{L(1+\omega)}{n\mu\varepsilon}$
	VR-MARINA (Alg. 2), $b' = 1$	Thm. D.4 & Cor. D.4 (NEW)	$\omega + \frac{L(1+\omega/\sqrt{n})}{\mu} + \frac{L\sqrt{1+\omega}}{n\mu\varepsilon}$	$\omega + \frac{L(1+\omega/\sqrt{n})}{\mu} + \frac{L\sqrt{1+\omega}}{n\mu\varepsilon}$
	VR-MARINA (Alg. 2), $b' = \Theta\left(\frac{1}{n\mu\varepsilon}\right)$	Thm. D.4 & Cor. D.4 (NEW)	$\omega + \frac{L(1+\omega/\sqrt{n})}{\mu}$	$\frac{1+\omega}{n\mu\varepsilon} + \frac{L(1+\omega/\sqrt{n})}{n\mu^2\varepsilon} + \frac{L(1+\omega)}{n\mu^2\sqrt{\varepsilon}}$
PP, (1)	FedSTEPH [(4)]	(Das et al., 2020)	$\left(\frac{L}{\mu}\right)^{3/2}$	$\left(\frac{L}{\mu}\right)^{3/2}$
	PP-MARINA (Alg. 4)	Thm. E.2 & Cor. E.2 (NEW)	$\frac{(\omega+1)n}{r} + \frac{L(1+(1+\omega)\sqrt{n}/r)}{\mu}$	$\frac{(\omega+1)n}{r} + \frac{L(1+(1+\omega)\sqrt{n}/r)}{\mu}$

[(1)] The results for FedCOMGATE are derived under assumption that for all vectors $x_1, \ldots, x_n \in \mathbb{R}^d$ the quantization operator $\mathcal{Q}$ satisfies $\mathbf{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \mathcal{Q}(x_j)\right\|^2 - \left\|\mathcal{Q}\left(\frac{1}{n}\sum_{i=1}^n x_j\right)\right\|^2\right] \leq G$ for some constant $G \geq 0$. In fact, this assumption does not hold for classical quantization operators like RandK and ℓ_2-quantization on $\mathbb{R}^d$. The counterexample: $n = 2$ and $x_1 = -x_2 = (t, t, \ldots, t)^\top$ with arbitrary large $t > 0$.

[(2)] One can even further improve the communication complexity by increasing b'.

[(3)] No assumptions on the smoothness of the stochastic realizations $f_\xi(x)$ are used.

[(4)] The rate is derived under assumption that $r = \Omega((1+\omega)\sqrt{L/\mu}\log(1/\varepsilon))$.

One can find the state-of-the-art distributed optimization methods using these techniques and their combinations in (Lian et al., 2017; Karimireddy et al., 2020; Li et al., 2019; Koloskova et al., 2020b). Moreover, there exist results based on the combinations of communication compression with either decentralized communication, e.g., Choco-SGD (Koloskova et al., 2020a), or local updates, e.g., Qsparse-Local-SGD (Basu et al., 2019), FedCOMGATE (Haddadpour et al., 2020), FedSTEPH (Das et al., 2020), where in (Basu et al., 2019) the convergence rates were derived under an assumption that the stochastic gradient has uniformly bounded second moment and the results for Choco-SGD, FedCOMGATE, FedSTEPH were described either earlier in the text, or in Table 1.

1.3. Preliminaries

We will rely on two key assumptions thrughout the text.

Assumption 1.1 (Uniform lower bound). *There exists $f_* \in \mathbb{R}$ such that $f(x) \geq f_*$ for all $x \in \mathbb{R}^d$.*

Assumption 1.2 (L-smoothness). *We assume that f_i is L_i-smooth for all $i \in [n] = \{1, 2, \ldots, n\}$ meaning that the following inequality holds $\forall x, y \in \mathbb{R}^d$, $\forall i \in [n]$:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x - y\|. \tag{2}$$

This assumption implies that f is L_f-smooth with $L_f^2 \leq L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$.

Finally, we describe a large class of unbiased compression operators satisfying a certain variance bound, which we will refer to, in this paper, by the name *quantization*.

Definition 1.1 (Quantization). *We say that a stochastic mapping $\mathcal{Q} : \mathbb{R}^d \to \mathbb{R}^d$ is a quantization operator/quantization if there exists $\omega > 0$ such that for any $x \in \mathbb{R}^d$, we have*

$$\mathbf{E}\left[\mathcal{Q}(x)\right] = x, \quad \mathbf{E}\left[\|\mathcal{Q}(x) - x\|^2\right] \leq \omega\|x\|^2. \tag{3}$$

For the given quantization operator $\mathcal{Q}(x)$, we define the the expected density as $\zeta_{\mathcal{Q}} = \sup_{x\in\mathbb{R}^d} \mathbf{E}\left[\|\mathcal{Q}(x)\|_0\right]$, where $\|y\|_0$ is the number of non-zero components of $y \in \mathbb{R}^d$.

Notice that the expected density is well-defined for any quantization operator since $\|\mathcal{Q}(x)\|_0 \leq d$.

2. MARINA

In this section, we describe the main algorithm of this work: MARINA (see Algorithm 1). At each iteration of MARINA, each worker i either sends to the server the dense vector $\nabla f_i(x^{k+1})$ with probability p, or it sends the quantized gradient difference $\mathcal{Q}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right)$ with probability $1-p$. In the first situation, the server just averages the vectors received from workers and gets $g^{k+1} = \nabla f(x^{k+1})$, whereas in the second case, the server averages the quantized differences from all workers and then adds the result to g^k to get g^{k+1}. Moreover, if $\mathcal{Q}$ is identity quantization, i.e., $\mathcal{Q}(x) = x$, then MARINA reduces to Gradient Descent (GD).

Algorithm 1 MARINA

Input: starting point x^0, stepsize γ, probability $p \in (0,1]$, number of iterations K
Initialize $g^0 = \nabla f(x^0)$
for $k = 0, 1, \ldots, K-1$ **do**
  Sample $c_k \sim \text{Be}(p)$
  Broadcast g^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    $x^{k+1} = x^k - \gamma g^k$
    Set $g_i^{k+1} = \nabla f_i(x^{k+1})$ if $c_k = 1$, and $g_i^{k+1} = g^k + \mathcal{Q}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right)$ otherwise
  end for
  $g^{k+1} = \frac{1}{n}\sum_{i=1}^n g_i^{k+1}$
end for
Return: $\hat{x}^K$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

However, for non-trivial quantizations, we have $\mathbf{E}[g^{k+1} \mid x^{k+1}] \neq \nabla f(x^{k+1})$ unlike all other distributed methods using exclusively unbiased compressors we know of. That is, g^{k+1} is a *biased* stochastic estimator of $\nabla f(x^{k+1})$. However, MARINA is an example of a rare phenomenon in stochastic optimization when the *bias of the stochastic gradient helps to achieve better complexity.*

2.1. Convergence Results for Generally Non-Convex Problems

We start with the following result.

Theorem 2.1. *Let Assumptions 1.1 and 1.2 be satisfied. Then, after*

$$K = \mathcal{O}\left(\frac{\Delta_0 L}{\varepsilon^2}\left(1 + \sqrt{\frac{(1-p)\omega}{pn}}\right)\right)$$

iterations with $\Delta_0 = f(x^0) - f_*$, $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$ *and the stepsize* $\gamma \leq L^{-1}\left(1 + \sqrt{(1-p)\omega/(pn)}\right)^{-1}$, MARINA *produces point* $\hat{x}^K$ *for which* $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$.

One can find the full statement of the theorem together with its proof in Section C.1 of the Appendix.

The following corollary provides the bounds on the number of iterations/communication rounds and estimates the total communication cost needed to achieve an ε-stationary point in expectation. Moreover, for simplicity, throughout the paper we assume that the communication cost is proportional to the number of non-zero components of transmitted vectors from workers to the server.

Corollary 2.1. *Let the assumptions of Theorem 2.1 hold and* $p = \zeta_{\mathcal{Q}}/d$. *If* $\gamma \leq L^{-1}\left(1 + \sqrt{\omega(d-\zeta_{\mathcal{Q}})/(n\zeta_{\mathcal{Q}})}\right)^{-1}$, *then* MARINA *requires*

$$\mathcal{O}\left(\frac{\Delta_0 L}{\varepsilon^2}\left(1 + \sqrt{\frac{\omega}{n}\left(\frac{d}{\zeta_{\mathcal{Q}}} - 1\right)}\right)\right)$$

iterations/communication rounds in order to achieve $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$, *and the expected total communication cost per worker is* $\mathcal{O}(d + \zeta_{\mathcal{Q}} K)$.

Let us clarify the obtained result. First of all, if $\omega = 0$ (no quantization), then $\zeta_{\mathcal{Q}} = 0$ and the rate coincides with the rate of Gradient Descent (GD). Since GD is optimal among first-order methods in terms of reducing the norm of the gradient (Carmon et al., 2019), the dependence on ε in our bound cannot be improved in general. Next, if n is large enough, i.e., $n \geq \omega(d/\zeta_{\mathcal{Q}} - 1)$, then[1] the iteration complexity of MARINA (method with compressed communications) and GD (method with dense communications) coincide. This means that in this regime, MARINA is able to reach a provably better communication complexity than GD!

2.2. Convergence Results Under Polyak-Łojasiewicz condition

In this section, we provide a complexity bounds for MARINA under the Polyak-Łojasiewicz (PŁ) condition.

Assumption 2.1 (PŁ condition)**.** *Function* f *satisfies Polyak-Łojasiewicz (PŁ) condition with parameter* μ, *i.e.,*

$$\|\nabla f(x)\|^2 \geq 2\mu\left(f(x) - f(x^*)\right). \tag{4}$$

holds for $x^* = \arg\min_{x \in \mathbb{R}^d} f(x)$ *and for all* $x \in \mathbb{R}^d$.

Under this and previously introduced assumptions, we derive the following result.

Theorem 2.2. *Let Assumptions 1.1, 1.2 and 2.1 be satisfied. Then, after*

$$K = \mathcal{O}\left(\max\left\{\frac{1}{p}, \frac{L}{\mu}\left(1 + \sqrt{\frac{(1-p)\omega}{pn}}\right)\right\}\log\frac{\Delta_0}{\varepsilon}\right)$$

[1] For ℓ_2-quantization this requirement is satisfied when $n \geq d$.

iterations with $\Delta_0 = f(x^0) - f(x^*)$, $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$ *and the stepsize* $\gamma \leq \min\left\{L^{-1}\left(1+\sqrt{2(1-p)\omega/(pn)}\right)^{-1}, p(2\mu)^{-1}\right\}$, MARINA *produces a point* x^K *for which* $\mathbf{E}[f(x^K) - f(x^*)] \leq \varepsilon$.

One can find the full statement of the theorem together with its proof in Section C.2 of the Appendix.

3. Variance Reduction

Throughout this section, we assume that the local loss on each node has either a finite-sum form (finite sum case),

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

or an expectation form (online case),

$$f_i(x) = \mathbf{E}_{\xi_i\sim\mathcal{D}_i}[f_{\xi_i}(x)]. \tag{6}$$

3.1. Finite Sum Case

In this section, we generalize MARINA to problems of the form (1)+(5), obtaining VR-MARINA (see Algorithm 2). At

Algorithm 2 VR-MARINA: finite sum case

Input: starting point x^0, stepsize γ, minibatch size b', probability $p \in (0,1]$, number of iterations K
Initialize $g^0 = \nabla f(x^0)$
for $k = 0, 1, \ldots, K-1$ **do**
  Sample $c_k \sim \text{Be}(p)$
  Broadcast g^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    $x^{k+1} = x^k - \gamma g^k$
    Set $g_i^{k+1} = \nabla f_i(x^{k+1})$ if $c_k = 1$, and $g_i^{k+1} = g^k + \mathcal{Q}\left(\frac{1}{b'}\sum_{j\in I'_{i,k}}(\nabla f_{ij}(x^{k+1}) - \nabla f_{ij}(x^k))\right)$ otherwise, where $I'_{i,k}$ is the set of the indices in the minibatch, $|I'_{i,k}| = b'$
  end for
  $g^{k+1} = \frac{1}{n}\sum_{i=1}^n g_i^{k+1}$
end for
Return: $\hat{x}^K$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

each iteration of VR-MARINA, devices are to compute the full gradients $\nabla f_i(x^{k+1})$ and send them to the server with probability p. Typically, $p \leq 1/m$ and m is large, meaning that workers compute full gradients rarely (once per $\geq m$ iterations in expectation). At other iterations, workers compute minibatch stochastic gradients evaluated at the current and previous points, compress them using an unbiased compression operator, i.e., quantization/quantization operator, and send the resulting vectors $g_i^{k+1} - g^k$ to the server. Moreover, if $\mathcal{Q}$ is the identity quantization, i.e., $\mathcal{Q}(x) = x$, and $n = 1$, then MARINA reduces to the optimal method PAGE (Li et al., 2020).

In this part, we will rely on the following average smoothness assumption.

Assumption 3.1 (Average $\mathcal{L}$-smoothness)**.** *For all* $k \geq 0$ *and* $i \in [n]$ *the minibatch stochastic gradients difference* $\widetilde{\Delta}_i^k = \frac{1}{b'}\sum_{j\in I'_{i,k}}(\nabla f_{ij}(x^{k+1}) - \nabla f_{ij}(x^k))$ *computed on the* i*-th worker satisfies* $\mathbf{E}\left[\widetilde{\Delta}_i^k \mid x^k, x^{k+1}\right] = \Delta_i^k$ *and*

$$\mathbf{E}\left[\left\|\widetilde{\Delta}_i^k - \Delta_i^k\right\|^2 \mid x^k, x^{k+1}\right] \leq \frac{\mathcal{L}_i^2}{b'}\|x^{k+1} - x^k\|^2 \tag{7}$$

with some $\mathcal{L}_i \geq 0$*, where* $\Delta_i^k = \nabla f_i(x^{k+1}) - \nabla f_i(x^k)$.

This assumption is satisfied in many standard minibatch regimes. In particular, if $I'_{i,k} = \{1, \ldots, m\}$, then $\mathcal{L}_i = 0$, and if $I'_{i,k}$ consists of b' i.i.d. samples from the uniform distributions on $\{1, \ldots, m\}$ and f_{ij} are L_{ij}-smooth, then $\mathcal{L}_i \leq \max_{j\in[m]} L_{ij}$.

Under this and the previously introduced assumptions, we derive the following result.

Theorem 3.1. *Consider the finite sum case* (1)+(5)*. Let Assumptions 1.1, 1.2 and 3.1 be satisfied. Then, after*

$$K = \mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2}\left(L + \sqrt{\frac{1-p}{pn}\left(\omega L^2 + \frac{(1+\omega)\mathcal{L}^2}{b'}\right)}\right)\right)$$

iterations with $\Delta_0 = f(x^0) - f_*$, $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$, $\mathcal{L}^2 = \frac{1}{n}\sum_{i=1}^n \mathcal{L}_i^2$ *and the stepsize* $\gamma \leq \left(L + \sqrt{\left(\omega L^2 + (1+\omega)\mathcal{L}^2/b'\right)(1-p)/(pn)}\right)^{-1}$, VR-MARINA *produces such a point* $\hat{x}^K$ *that* $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$.

One can find the full statement of the theorem together with its proof in Section D.1.1 of the Appendix.

Corollary 3.1. *Let the assumptions of Theorem 3.1 hold and* $p = \min\{\zeta_{\mathcal{Q}}/d, b'/(m+b')\}$*, where* $b' \leq m$*. If* $\gamma \leq \left(L + \sqrt{\left(\omega L^2 + (1+\omega)\mathcal{L}^2/b'\right)\max\left\{d/\zeta_{\mathcal{Q}} - 1, m/b'\right\}/n}\right)^{-1}$ *then* VR-MARINA *requires*

$$\mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2}\left(L\left(1 + \sqrt{\frac{\omega\max\left\{d/\zeta_{\mathcal{Q}} - 1, m/b'\right\}}{n}}\right) + \mathcal{L}\sqrt{\frac{(1+\omega)\max\left\{d/\zeta_{\mathcal{Q}} - 1, m/b'\right\}}{nb'}}\right)\right)$$

iterations/communication rounds and $\mathcal{O}(m + b'K)$ *stochastic oracle calls per node in expectation in order to achieve* $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$*, and the expected total communication cost per worker is* $\mathcal{O}(d + \zeta_{\mathcal{Q}}K)$.

First of all, when workers quatize differences of the full gradients, then $I'_{i,k} = \{1, \ldots, m\}$ for all $i \in [n]$ and $k \geq 0$,

implying $\mathcal{L} = 0$. In this case, the complexity bounds for VR-MARINA recover the ones for MARINA. Next, when $\omega = 0$ (no quantization) and $n = 1$, our bounds for iteration and oracle complexities for VR-MARINA recover the bounds for PAGE (Li & Richtárik, 2020), which is optimal for finite-sum smooth non-convex optimization. This observation implies that the dependence on ε and m in the complexity bounds for VR-MARINA cannot be improved in the class of first-order stochastic methods. Next, we notice that up to the differences in smoothness constants, the iteration and oracle complexities for VR-MARINA benefit from the number of workers n. Finally, as Table 1 shows, the rates for VR-MARINA are strictly better than ones for the previous state-of-the-art method VR-DIANA (Horváth et al., 2019).

We provide the convergence results for VR-MARINA in the finite-sum case under the Polyak-Łojasiewicz condition, together with complete proofs, in Section D.1.2 of the Appendix.

3.2. Online Case

In this section, we focus on problems of type (1)+(6). For this type of problems, we consider a slightly modified version of VR-MARINA. That is, we replace line 8 in Algorithm 2 with the following update rule: $g_i^{k+1} = \frac{1}{b}\sum_{j\in I_{i,k}} \nabla f_{\xi_{ij}^k}(x^{k+1})$ if $c_k = 1$, and $g_i^{k+1} = g^k + \mathcal{Q}\left(\frac{1}{b'}\sum_{j\in I'_{i,k}} (\nabla f_{\xi_{ij}^k}(x^{k+1}) - \nabla f_{\xi_{ij}^k}(x^k))\right)$ otherwise, where $I_{i,k}, I'_{i,k}$ are the sets of the indices in the mini-batches, $|I_{i,k}| = b$, $|I'_{i,k}| = b'$, and ξ_{ij}^k is independently sampled from $\mathcal{D}_i$ for $i \in [n]$, $j \in [m]$ (see Algorithm 3 in the Appendix).

Before we provide our convergence results in this setup, we reformulate Assumption 3.1 for the online case.

Assumption 3.2 (Average $\mathcal{L}$-smoothness)**.** *For all $k \geq 0$ and $i \in [n]$ the minibatch stochastic gradients difference $\widetilde{\Delta}_i^k = \frac{1}{b'}\sum_{j\in I'_{i,k}} (\nabla f_{\xi_{ij}^k}(x^{k+1}) - \nabla f_{\xi_{ij}^k}(x^k))$ computed on the i-th worker satisfies $\mathbf{E}\left[\widetilde{\Delta}_i^k \mid x^k, x^{k+1}\right] = \Delta_i^k$ and*

$$\mathbf{E}\left[\left\|\widetilde{\Delta}_i^k - \Delta_i^k\right\|^2 \mid x^k, x^{k+1}\right] \leq \frac{\mathcal{L}_i^2}{b'}\|x^{k+1} - x^k\|^2 \quad (8)$$

with some $\mathcal{L}_i \geq 0$, where $\Delta_i^k = \nabla f_i(x^{k+1}) - \nabla f_i(x^k)$.

Moreover, we assume that the variance of the stochastic gradients on all nodes is uniformly upper bounded.

Assumption 3.3. *We assume that for all $i \in [n]$ there exists such constant $\sigma_i \in [0, +\infty)$ that for all $x \in \mathbb{R}^d$*

$$\mathbf{E}_{\xi_i\sim\mathcal{D}_i}\left[\nabla f_{\xi_i}(x)\right] = \nabla f_i(x), \quad (9)$$

$$\mathbf{E}_{\xi_i\sim\mathcal{D}_i}\left[\left\|\nabla f_{\xi_i}(x) - \nabla f_i(x)\right\|^2\right] \leq \sigma_i^2. \quad (10)$$

Under these and previously introduced assumptions, we derive the following result.

Theorem 3.2. *Consider the online case (1)+(6). Let Assumptions 1.1, 1.2, 3.2 and 3.3 be satisfied. Then, after*

$$K = \mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2}\left(L + \sqrt{\frac{1-p}{pn}\left(\omega L^2 + \frac{(1+\omega)\mathcal{L}^2}{b'}\right)}\right)\right)$$

iterations with $\Delta_0 = f(x^0) - f_*$, $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$, $\mathcal{L}^2 = \frac{1}{n}\sum_{i=1}^n \mathcal{L}_i^2$, the stepsize $\gamma \leq \left(L + \sqrt{\left(\omega L^2 + {(1+\omega)\mathcal{L}^2}/{b'}\right){(1-p)}/{(pn)}}\right)^{-1}$, and $b = \Theta\left({\sigma^2}/{(n\varepsilon^2)}\right)$, $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2$, VR-MARINA produces a point $\hat{x}^K$ for which $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$.

One can find the full statement of the theorem, together with its proof, in Section D.2.1 of the Appendix.

Corollary 3.2. *Let the assumptions of Theorem 3.2 hold and choose $p = \min\left\{{\zeta_{\mathcal{Q}}}/{d}, {b'}/{(b+b')}\right\}$, where $b' \leq b$, $b = \Theta\left({\sigma^2}/{(n\varepsilon^2)}\right)$. If $\gamma \leq \left(L + \sqrt{\left(\omega L^2 + {(1+\omega)\mathcal{L}^2}/{b'}\right)\max\left\{{d}/{\zeta_{\mathcal{Q}}} - 1, {b}/{b'}\right\}/n}\right)^{-1}$, then VR-MARINA requires*

$$\mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2}\left(L\left(1 + \sqrt{\frac{\omega}{n}\max\left\{\frac{d}{\zeta_{\mathcal{Q}}} - 1, \frac{\sigma^2}{nb'\varepsilon^2}\right\}}\right) + \mathcal{L}\sqrt{\frac{(1+\omega)}{nb'}\max\left\{\frac{d}{\zeta_{\mathcal{Q}}} - 1, \frac{\sigma^2}{nb'\varepsilon^2}\right\}}\right)\right)$$

iterations/communication rounds and $\mathcal{O}(\zeta_{\mathcal{Q}}K + {\sigma^2}/{(n\varepsilon^2)})$ stochastic oracle calls per node in expectation to achieve $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$, and the expected total communication cost per worker is $\mathcal{O}(d + \zeta_{\mathcal{Q}}K)$.

Similarly to the finite-sum case, when $\omega = 0$ (no quantization) and $n = 1$, our bounds for iteration and oracle complexities for VR-MARINA recover the bounds for PAGE (Li & Richtárik, 2020), which is optimal for online smooth non-convex optimization as well. That is, the dependence on ε in the complexity bound for VR-MARINA cannot be improved in the class of first-order stochastic methods. As previously, up to the differences in smoothness constants, the iteration and oracle complexities for VR-MARINA benefit from an increase in the number of workers n.

We provide the convergence results for VR-MARINA in the online case under the Polyak-Łojasiewicz condition, together with complete proofs, in Section D.2.2 of the Appendix.

4. Partial Participation

Finally, we propose another modification of MARINA. In particular, we prove an option for *partial participation* of

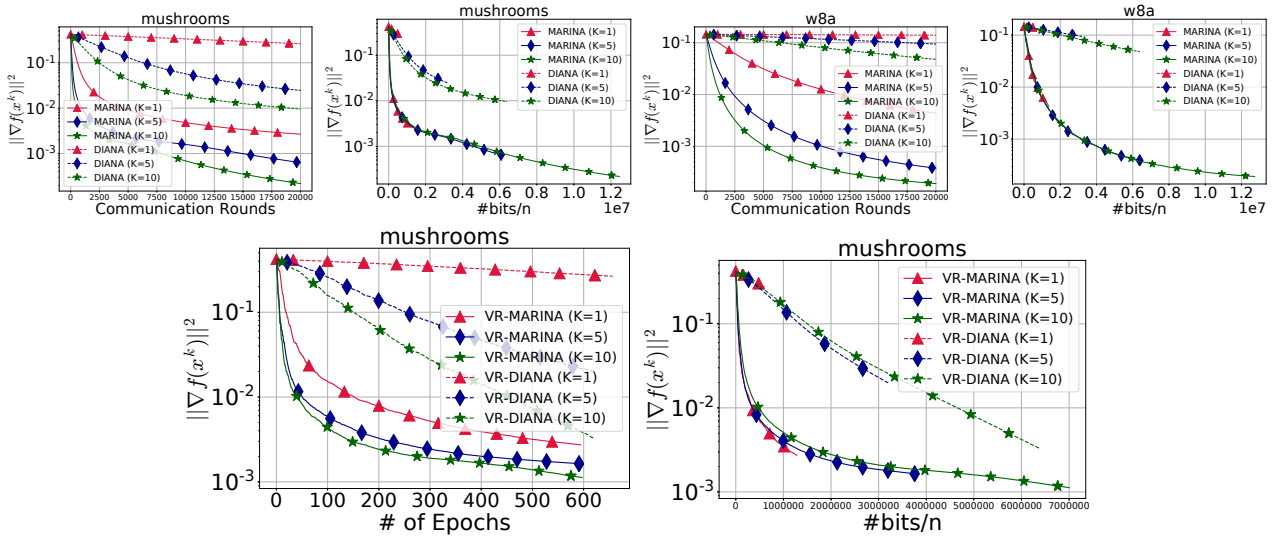

Figure 1: Comparison of MARINA with DIANA, and of VR-MARINA with VR-DIANA, on binary classification problem involving non-convex loss (11) with LibSVM data (Chang & Lin, 2011). Parameter n is chosen as per Tbl. 3 in the Appendix. Stepsizes for the methods are chosen according to the theory and the batchsizes for VR-MARINA and VR-DIANA are $\sim m/100$. In all cases, we used the RandK sparsification operator with K $\in \{1, 5, 10\}$.

the clients - a feature important in federated learning. The resulting method is called PP-MARINA (see Algorithm 4 in the Appendix). At each iteration of PP-MARINA, the server receives the quantized gradient differences from r clients with probability $1-p$, and aggregates full gradients from all clients with probability p, i.e., PP-MARINA coincides with MARINA up to the following difference: $g_i^{k+1} = \nabla f_i(x^{k+1})$, $g^{k+1} = \frac{1}{n}\sum_{i=1}^n g_i^{k+1}$ if $c_k = 1$, and $g_i^{k+1} = g^k + \mathcal{Q}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right)$, $g^{k+1} = \frac{1}{r}\sum_{i_k \in I'_k} g_{i_k}^{k+1}$ otherwise, where I'_k is the set of r i.i.d. samples from the uniform distribution over $\{1, \ldots, n\}$. That is, if the probability p is chosen to be small enough, then with high probability the server receives only quantized vectors from a subset of clients at each iteration.

Below, we provide a convergence result for PP-MARINA for smooth non-convex problems.

Theorem 4.1. *Let Assumptions 1.1 and 1.2 be satisfied. Then, after*

$$K = \mathcal{O}\left(\frac{\Delta_0 L}{\varepsilon^2}\left(1 + \sqrt{\frac{(1-p)(1+\omega)}{pr}}\right)\right)$$

iterations with $\Delta_0 = f(x^0) - f_*$, $L^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$ and the stepsize $\gamma \leq L^{-1}\left(1 + \sqrt{(1-p)(1+\omega)/(pr)}\right)^{-1}$, PP-MARINA produces a point $\hat{x}^K$ for which $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$.

One can find the full statement of the theorem together with its proof in Section E.1 of the appendix.

Corollary 4.1. *Let the assumptions of Theorem 4.1 hold and choose $p = \zeta_{\mathcal{Q}} r/(dn)$, where $r \leq n$. If $\gamma \leq L^{-1}\left(1 + \sqrt{(1+\omega)(dn - \zeta_{\mathcal{Q}} r)/(b' \zeta_{\mathcal{Q}} r)}\right)^{-1}$, then PP-MARINA requires*

$$\mathcal{O}\left(\frac{\Delta_0 L}{\varepsilon^2}\left(1 + \sqrt{\frac{1+\omega}{r}\left(\frac{dn}{\zeta_{\mathcal{Q}} r} - 1\right)}\right)\right)$$

iterations/communication rounds to achieve $\mathbf{E}[\|\nabla f(\hat{x}^K)\|^2] \leq \varepsilon^2$, and the expected total communication cost is $\mathcal{O}\left(dn + \zeta_{\mathcal{Q}} r K\right)$.

When $r = n$, i.e., all clients participate in communication with the server at each iteration, the rate for PP-MARINA recovers the rate for MARINA under the assumption that $(1+\omega)(d/\zeta_{\mathcal{Q}} - 1) = \mathcal{O}(\omega(d/\zeta_{\mathcal{Q}} - 1))$, which holds for a wide class of quantization operators, e.g., for identical quantization, RandK, and ℓ_p-quantization. In general, the derived complexity is strictly better than previous state-of-the-art one (see Table 1).

We provide the convergence results for PP-MARINA under the Polyak-Łojasiewicz condition, together with complete proofs, in Section E.2 of the Appendix.

5. Numerical Experiments

5.1. Binary Classification with Non-Convex Loss

We conduct several numerical experiments[2] on binary classification problem involving non-convex loss (Zhao et al., 2010) (used for two-layer neural networks) with LibSVM

[2]Our code is available at https://github.com/burlachenkok/marina.

data (Chang & Lin, 2011) to justify the theoretical claims of the paper. That is, we consider the following optimization problem:

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{N}\sum_{t=1}^{N}\ell(a_t^\top x,y_i)\right\}, \tag{11}$$

where $\{a_t\}\in\mathbb{R}^d$, $y_i\in\{-1,1\}$ for all $t=1,\ldots,N$, and the function $\ell:\mathbb{R}^d\to\mathbb{R}$ is defined as

$$\ell(b,c)=\left(1-\frac{1}{1+\exp(-bc)}\right)^2.$$

The distributed environment is simulated in Python 3.8 using MPI4PY and other standard libraries. Additional details about the experimental setup together with extra experiments are deferred to Section A of the Appendix.

In our experiments, we compare MARINA with the full-batch version of DIANA, and then VR-MARINA with VR-DIANA. We exclude FedCOMGATE and FedPATH from this comparison since they have significantly worse oracle complexities (see Table 1). The results are presented in Fig. 1. As our theory predicts, the first row shows the superiority of MARINA to DIANA both in terms of iteration/communication complexity and the total number of transmitted bits to achieve the given accuracy. Next, to study the oracle complexity as well, we consider non-full-batched methods – VR-MARINA and VR-DIANA – since they have better oracle complexity than the full-batched methods in the finite-sum case. Again, the results presented in the second row justify that VR-MARINA outperforms VR-DIANA in terms of oracle complexity and the total number of transmitted bits to achieve the given accuracy.

5.2. Image Classification

We also compared the performance of VR-MARINA and VR-DIANA on the training `ResNet-18` (He et al., 2016) at `CIFAR100` (Krizhevsky et al., 2009) dataset. Formally, the optimization problem is

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{N}\sum_{i=1}^{N}\ell(p(f(a_i,x)),y_i)\right\}, \tag{12}$$

where $\{(a_i,y_i)\}_{i=1}^N$ encode images and labels from `CIFAR100` dataset, $f(a_i,x)$ is the output of `ResNet-18` on image a_i with weights x, p is softmax function, and $\ell(\cdot,\cdot)$ is cross-entropy loss. The code is wrtitten in Python 3.9 using `PyTorch 1.7`, and the distributed environment is simulated.

The results are presented in Fig. 2. Again, VR-MARINA converges significantly faster than VR-DIANA both in terms of the oracle complexity and the total number of transmitted bits to achieve the given accuracy. See other details and observations in Section A of the Appendix.

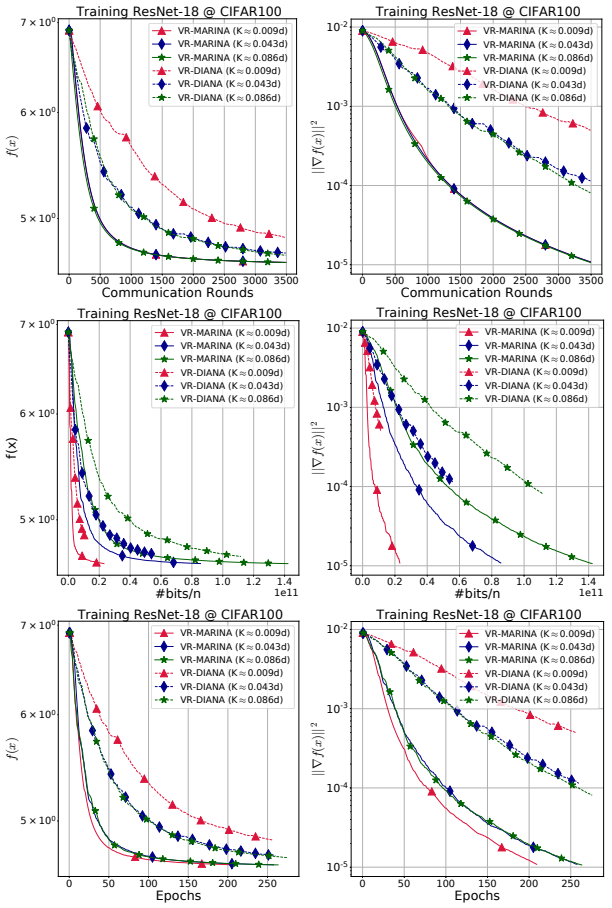

Figure 2: Comparison of VR-MARINA with VR-DIANA on training `ResNet-18` at `CIFAR100` dataset. Number of workers equals 5. Stepsizes for the methods were tuned and the batchsizes are $\sim m/50$. In all cases, we used the RandK sparsification operator, the approximate values of K are given in the legends (d is dimension of the problem).

Acknowledgements

The work of Peter Richtárik, Eduard Gorbunov, Konstantin Burlachenko and Zhize Li was supported by KAUST Baseline Research Fund. The paper was written while E. Gorbunov was a research intern at KAUST. The work of E. Gorbunov in Sections 1, 2, and C was also partially supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) 075-00337-20-03, project No. 0714-2020-0005, and in Sections 3, 4, D, E – by RFBR, project number 19-31-51001. We thank Konstantin Mishchenko (KAUST) for a suggestion related to the experiments, Elena Bazanova (MIPT) for the suggestions about improving the text, and Slavomír Hanzely (KAUST) and Egor Shulgin (KAUST) for spotting the typos.

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Basu, D., Data, D., Karakus, C., and Diggavi, S. Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, pp. 14668–14679, 2019.

Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

Bhojanapalli, S., Kyrillidis, A., and Sanghavi, S. Dropping convexity for faster semi-definite optimization. In *Conference on Learning Theory*, pp. 530–582. PMLR, 2016.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Lower bounds for finding stationary points i. *Mathematical Programming*, pp. 1–50, 2019.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Danilova, M., Dvurechensky, P., Gasnikov, A., Gorbunov, E., Guminov, S., Kamzolov, D., and Shibaev, I. Recent theoretical advances in non-convex optimization. *arXiv preprint arXiv:2012.06188*, 2020.

Das, R., Hashemi, A., Sanghavi, S., and Dhillon, I. S. Improved convergence rates for non-convex federated learning with compression. *arXiv preprint arXiv:2012.04061*, 2020.

Fang, C., Li, C., Lin, Z., and Zhang, T. Near-optimal non-convex optimization via stochastic path integrated differential estimator. *Advances in Neural Information Processing Systems*, 31:689, 2018.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Gorbunov, E., Kovalev, D., Makarenko, D., and Richtárik, P. Linearly converging error compensated sgd. *Advances in Neural Information Processing Systems*, 33, 2020.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. *arXiv preprint arXiv:2007.01154*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Horváth, S., Ho, C.-Y., Ľudovít Horváth, Sahu, A. N., Canini, M., and Richtárik, P. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

Horváth, S., Kovalev, D., Mishchenko, K., Stich, S., and Richtárik, P. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

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

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signSGD and other gradient compression schemes. In *International Conference on Machine Learning*, pp. 3252–3261, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khanduri, P., Sharma, P., Kafle, S., Bulusu, S., Rajawat, K., and Varshney, P. K. Distributed stochastic non-convex optimization: Momentum-based variance reduction. *arXiv preprint arXiv:2005.00224*, 2020.

Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. Decentralized deep learning with arbitrary communication compression. *ICLR*, pp. arXiv:1907.09356, 2020a. URL https://arxiv.org/abs/1907.09356.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020b.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Konečný, J., McMahan, H. B., Yu, F., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: strategies for

improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, X., Yang, W., Wang, S., and Zhang, Z. Communication efficient decentralized training with multiple local updates. *arXiv preprint arXiv:1910.09126*, 5, 2019.

Li, Z. and Richtárik, P. A unified analysis of stochastic gradient methods for nonconvex federated optimization. *arXiv preprint arXiv:2006.07013*, 2020.

Li, Z., Bao, H., Zhang, X., and Richtárik, P. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. *arXiv preprint arXiv:2008.10898*, 2020.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

Łojasiewicz, S. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117:87–89, 1963.

Ma, C., Wang, K., Chi, Y., and Chen, Y. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *International Conference on Machine Learning*, pp. 3345–3354. PMLR, 2018.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

Murty, K. and Kabadi, S. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

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

Qian, X., Richtárik, P., and Zhang, T. Error compensated distributed sgd can be accelerated. *arXiv preprint arXiv:2010.00091*, 2020.

Safaryan, M., Shulgin, E., and Richtárik, P. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *arXiv preprint arXiv:2002.08958*, 2020.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 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.

Sharma, P., Kafle, S., Khanduri, P., Bulusu, S., Rajawat, K., and Varshney, P. K. Parallel restarted spider–communication efficient distributed nonconvex optimization with optimal computation complexity. *arXiv preprint arXiv:1912.06036*, 2019.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates. *Journal of Machine Learning Research*, 21:1–36, 2020.

Sun, H., Lu, S., and Hong, M. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, pp. 9217–9228. PMLR, 2020.

Sun, R. Optimization for deep learning: theory and algorithms. *arXiv preprint arXiv:1912.08957*, 2019.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx4wnEtvH.

Zhao, L., Mammadov, M., and Yearwood, J. From convex to nonconvex: a loss function analysis for binary classification. In *2010 IEEE International Conference on Data Mining Workshops*, pp. 1281–1288. IEEE, 2010.