Stochastic Sign Descent Methods: New Algorithms and Better Theory

Mher Safaryan [1] **Peter Richtárik** [1 2]

Abstract

Various gradient compression schemes have been proposed to mitigate the communication cost in distributed training of large scale machine learning models. Sign-based methods, such as signSGD (Bernstein et al., 2018), have recently been gaining popularity because of their simple compression rule and connection to adaptive gradient methods, like ADAM. In this paper, we analyze sign-based methods for non-convex optimization in three key settings: (i) standard single node, (ii) parallel with shared data and (iii) distributed with partitioned data. For single machine case, we generalize the previous analysis of signSGD relying on intuitive bounds on success probabilities and allowing even biased estimators. Furthermore, we extend the analysis to parallel setting within a parameter server framework, where exponentially fast noise reduction is guaranteed with respect to number of nodes, maintaining 1-bit compression in both directions and using small mini-batch sizes. Next, we identify a fundamental issue with signSGD to converge in distributed environment. To resolve this issue, we propose a new sign-based method, *Stochastic Sign Descent with Momentum (SSDM)*, which converges under standard bounded variance assumption with the optimal asymptotic rate. We validate several aspects of our theoretical findings with numerical experiments.

1. Introduction

One of the key factors behind the success of modern machine learning models is the availability of large amounts of training data (Bottou & Le Cun, 2003; Krizhevsky et al., 2012; Schmidhuber, 2015). However, the state-of-the-art deep learning models deployed in industry typically rely on datasets too large to fit the memory of a single computer, and hence the training data is typically split and stored across a number of compute nodes capable of working in parallel. Training such models then amounts to solving optimization problems of the form

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{M}\sum_{n=1}^{M} f_n(x), \qquad (1)$$

where $f_n : \mathbb{R}^d \to \mathbb{R}$ represents the *non-convex* loss of a deep learning model parameterized by $x \in \mathbb{R}^d$ associated with data stored on node n. Arguably, stochastic gradient descent (SGD) (Robbins & Monro, 1951; Vaswani et al., 2019; Qian et al., 2019) in of its many variants (Kingma & Ba, 2015; Duchi et al., 2011; Schmidt et al., 2017; Zeiler, 2012; Ghadimi & Lan, 2013) is the most popular algorithm for solving (1). In its basic implementation, all workers $n \in \{1, 2, \dots, M\}$ in parallel compute a random approximation $\hat{g}^n(x_k)$ of $\nabla f_n(x_k)$, known as the *stochastic gradient*. These approximations are then sent to a master node which performs the aggregation $\hat{g}(x_k) := \frac{1}{M}\sum_{n=1}^{M}\hat{g}^n(x_k)$. The aggregated vector is subsequently broadcast back to the nodes, each of which performs an update of the form

$$x_{k+1} = x_k - \gamma_k \hat{g}(x_k),$$

updating their local copies of the parameters of the model.

1.1. Gradient Compression

Typically, communication of the local gradient estimators $\hat{g}^n(x_k)$ to the master forms the bottleneck of such a system (Seide et al., 2014; Zhang et al., 2017; Lin et al., 2018). In an attempt to alleviate this communication bottleneck, a number of compression schemes for gradient updates have been proposed and analyzed (Alistarh et al., 2017; Wang et al., 2018; Wen et al., 2017; Khirirat et al., 2018; Mishchenko et al., 2019). A *compression scheme* is a (possibly randomized) mapping $Q : \mathbb{R}^d \to \mathbb{R}^d$, applied by the nodes to $\hat{g}^n(x_k)$ (and possibly also by the master to aggregated update in situations when broadcasting is expensive as well) in order to reduce the number of bits of the communicated message.

Sign-based compression. Although most of the existing theory is limited to *unbiased* compression schemes, i.e., $\mathbb{E}Q(x) = x$, *biased* schemes such as those based on communicating signs of the update entries only often perform

[1]KAUST, Saudi Arabia [2]MIPT, Russia. Correspondence to: Mher Safaryan <mher.safaryan@gmail.com>.

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

Table 1. Summary of main theoretical results obtained in this work.

	Convergence rate	Gradient norm used in theory	Weak noise assumptions	Weak dependence on smoothness	Can handle biased estimator?	Can work with small minibatch?	Can handle partitioned train data?
SGD (Ghadimi & Lan, 2013)	$\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$	l^2 norm squared	$\text{Var}[\hat{g}] \le \sigma^2$	✗ $\max_{i=1}^{d} L_i$	NO	YES	YES
signSGD (Bernstein et al., 2019)	$\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$	a mix of l^1 and l^2 squared	✗ unimodal, symmetric & $\text{Var}[\hat{g}_i] \le \sigma_i^2$	✓ $\frac{1}{d}\sum_{i=1}^{d} L_i$	NO	YES	NO
signSGD with M Maj.Vote (Bernstein et al., 2019)	$\mathcal{O}\left(\frac{1}{K^{1/4}}\right)$ (speedup$\sim \frac{1}{\sqrt{M}}$)	l^1 norm	✗ unimodal, symmetric & $\text{Var}[\hat{g}_i] \le \sigma_i^2$	✓ $\frac{1}{d}\sum_{i=1}^{d} L_i$	NO	NO	NO
Signum (Bernstein et al., 2018)	$\mathcal{O}\left(\frac{\log K}{K^{1/4}}\right)$	l^1 norm	✗ unimodal, symmetric & $\text{Var}[\hat{g}_i] \le \sigma_i^2$	✓ $\frac{1}{d}\sum_{i=1}^{d} L_i$	NO	NO	NO
signSGD **This work** (Thm. 1, 2)	$\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$	ρ-norm	✓ $\rho_i > \frac{1}{2}$	✓ $\frac{1}{d}\sum_{i=1}^{d} L_i$	YES	YES	NO
signSGD with M Maj.Vote **This work** (Thm. 3)	$\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$ (speedup$\sim e^{-M}$)	ρ_M-norm	✓ $\rho_i > \frac{1}{2}$	✓ $\frac{1}{d}\sum_{i=1}^{d} L_i$	YES	YES	NO
SSDM (Alg. 3) **This work** (Thm. 4)	$\mathcal{O}\left(\frac{1}{K^{1/4}}\right)$	l^1 norm	$\text{Var}[\hat{g}] \le \sigma^2$	✓ $\frac{1}{M}\sum_{n=1}^{M} L^n$	NO	YES	YES

much better (Seide et al., 2014; Strom, 2015; Wen et al., 2017; Carlson et al., 2015; Balles & Hennig, 2018; Bernstein et al., 2018; 2019; Zaheer et al., 2018; Liu et al., 2019). The simplest among these sign-based methods is signSGD (see Algorithm 1), whose update direction is assembled from the component-wise signs of the stochastic gradient.

Adaptive methods. While ADAM is one of the most popular *adaptive* optimization methods used in deep learning (Kingma & Ba, 2015), there are issues with its convergence (Reddi et al., 2019) and generalization (Wilson et al., 2017) properties. It was noted by Balles & Hennig (2018) that the behaviour of ADAM is similar to a momentum version of signSGD. Connection between sign-based and adaptive methods has long history, originating at least in Rprop (Riedmiller & Braun, 1993) and RMSprop (Tieleman & Hinton, 2012). Therefore, investigating the behavior of signSGD can improve our understanding on the convergence of adaptive methods such as ADAM.

2. Contributions

We now present the main contributions of this work. Our key results are summarized in Table 1.

2.1. Single Machine Setup

• **2 methods for 1-node setup.** In the $M = 1$ case, we study two general classes of sign based methods for minimizing a smooth non-convex function f. The first method has the standard form[1]

$$x_{k+1} = x_k - \gamma_k \,\text{sign}\, \hat{g}(x_k), \tag{2}$$

while the second has a new form not considered in the literature before:

$$x_{k+1} = \arg\min\{f(x_k), f(x_k - \gamma_k \,\text{sign}\, \hat{g}(x_k))\}. \tag{3}$$

• **Key novelty.** The key novelty of our methods is in a *substantial relaxation* of the requirements that need to be imposed on the gradient estimator $\hat{g}(x_k)$ of the true gradient $\nabla f(x^k)$. In sharp contrast with existing approaches, we allow $\hat{g}(x_k)$ to be *biased*. Remarkably, we only need one additional and rather weak assumption on $\hat{g}(x_k)$ for the methods to provably converge: we require the signs of the entries of $\hat{g}(x_k)$ to be equal to the signs of the entries of $g(x^k) := \nabla f(x^k)$ with a probability strictly larger than $1/2$ (see Assumption 1). Formally, we assume the following bounds on success probabilities:

$$\text{Prob}(\text{sign}\, \hat{g}_i(x_k) = \text{sign}\, g_i(x_k)) > \tfrac{1}{2} \tag{SPB}$$

for all $i \in \{1, 2, \ldots, d\}$ with $g_i(x_k) \neq 0$.

[1] sign g is applied element-wise to the entries $g_1, g_2, \ldots, g_d$ of $g \in \mathbb{R}^d$. For $t \in \mathbb{R}$ we define sign $t = 1$ if $t > 0$, sign $t = 0$ if $t = 0$, and sign $t = -1$ if $t < 0$.

We provide three necessary conditions for our assumption to hold (see Lemma 1, 2 and 3) and show through a counterexample that a slight violation of this assumption breaks the convergence.

• **Convergence theory.** While our complexity bounds have the same $\mathcal{O}(1/\sqrt{K})$ dependence on the number of iterations, they have a *better dependence on the smoothness parameters* associated with f. Theorem 1 is the first result on signSGD for non-convex functions which does not rely on mini-batching, and which allows for step sizes independent of the total number of iterations K. Finally, Theorem 1 in (Bernstein et al., 2019) can be recovered from our general Theorem 1. Our bounds are cast in terms of a *novel norm-like function, which we call the ρ-norm*, which is a weighted l^1 norm with positive variable weights.

2.2. Parallel Setting with Shared Data

• **Noise reduction at exponential speed.** Under the same SPB assumption, we extend our results to the *parallel setting* with arbitrary M nodes, where we also consider sign-based compression of the aggregated gradients. Considering the noise-free training as a baseline, we guarantee exponentially fast noise reduction with respect to M (see Theorem 3).

2.3. Distributed Training with Partition Data

• **New sign-based method for distributed training.** We describe a fundamental obstacle in distributed environment, which prevents signSGD to converge. To resolve the issue, we propose a new sign-based method–*Stochastic Sign Descent with Momentum (SSDM)*; see Algorithm 3.

• **Key novelty.** The key novelty in our SSDM method is the notion of *stochastic sign* operator $\widetilde{\mathrm{sign}}:\mathbb{R}^d\to\mathbb{R}^d$ defined as follows:

$$\left(\widetilde{\mathrm{sign}}\, g\right)_i = \begin{cases} +1, & \text{with probability } \frac{1}{2}+\frac{1}{2}\frac{g_i}{\|g\|} \\ -1, & \text{with probability } \frac{1}{2}-\frac{1}{2}\frac{g_i}{\|g\|} \end{cases}$$

for $1\le i\le d$ and $\widetilde{\mathrm{sign}}\,\mathbf{0}=\mathbf{0}$ with probability 1.

Unlike the deterministic sign operator, stochastic $\widetilde{\mathrm{sign}}$ naturally satisfies the SPB assumption and it gives an unbiased estimator with a proper scaling factor.

• **Convergence theory.** Under the standard bounded variance condition, our SSDM method guarantees the optimal asymptotic rate $\mathcal{O}(\varepsilon^{-4})$ without *error feedback* trick and communicating sign-bits only (see Theorem 4).

3. Success Probabilities and Gradient Noise

In this section we describe our key (and weak) assumption on the gradient estimator $\hat{g}(x)$, and show through a counterexample that without this assumption, signSGD can fail to converge. Then we provide several sufficient conditions for our assumption to hold and define a new norm-like function for measuring the gradients.

3.1. Success Probability Bounds

First, we state our key assumption on the stochastic gradient.

Assumption 1 (SPB: Success Probability Bounds)**.** *For any $x\in\mathbb{R}^d$, we have access to an independent (and not necessarily unbiased) estimator $\hat{g}(x)$ of the true gradient $g(x):=\nabla f(x)$ that if $g_i(x)\neq 0$, then*

$$\rho_i(x) := \mathrm{Prob}\left(\mathrm{sign}\,\hat{g}_i(x) = \mathrm{sign}\, g_i(x)\right) > \tfrac{1}{2} \qquad (4)$$

for all $x\in\mathbb{R}^d$ and all $i\in\{1,2,\dots,d\}$.

We will refer to the probabilities ρ_i as *success probabilities*. As we will see, they play a central role in the convergence of sign based methods. Moreover, we argue that it is reasonable to require from the sign of stochastic gradient to show true gradient direction more likely than the opposite one. Extreme cases of this assumption are the absence of gradient noise, in which case $\rho_i=1$, and an overly noisy stochastic gradient, in which case $\rho_i\approx\frac{1}{2}$.

Remark 1. *Assumption 1 can be relaxed by replacing bounds (4) with*

$$\mathbb{E}\left[\mathrm{sign}\left(\hat{g}_i(x)\cdot g_i(x)\right)\right] > 0, \quad \textit{if} \quad g_i(x)\neq 0.$$

However, if $\mathrm{sign}\,\hat{g}_i(x)\neq 0$ almost surely (e.g. $\hat{g}_i(x)$ is continuous), then these bounds are identical.

Extension to stochastic sign oracle. Notice that we do *not* require $\hat{g}$ to be unbiased and we do *not* assume uniform boundedness of the variance, or of the second moment. This observation allows to extend existing theory to more general sign-based methods with a stochastic sign oracle. By a stochastic sign oracle we mean an oracle that takes $x_k\in\mathbb{R}^d$ as an input, and outputs a random vector $\hat{s}_k\in\mathbb{R}^d$ with entries in ±1. However, for the sake of simplicity, in the rest of the paper we will work with the signSGD formulation, i.e., we let $\hat{s}_k=\mathrm{sign}\,\hat{g}(x_k)$.

3.2. A Counterexample to SIGNSGD

Here we analyze a counterexample to signSGD discussed in (Karimireddy et al., 2019). Consider the following least-squares problem with unique minimizer $x^*=(0,0)$:

$$\min_{x\in\mathbb{R}^2}\left\{f(x)=\tfrac{1}{2}\left[\langle a_1,x\rangle^2+\langle a_2,x\rangle^2\right]\right\}, \qquad (5)$$

$$a_1=\left[\begin{smallmatrix}1+\varepsilon\\ -1+\varepsilon\end{smallmatrix}\right], \quad a_2=\left[\begin{smallmatrix}-1+\varepsilon\\ 1+\varepsilon\end{smallmatrix}\right], \qquad (6)$$

where $\varepsilon\in(0,1)$ and stochastic gradient $\hat{g}(x)=\nabla\langle a_i,x\rangle^2=2\langle a_i,x\rangle a_i$ with probabilities $1/2$ for $i=1,2$. Let us take any point from the line $l=\{(z_1,z_2)\colon z_1+z_2=$

$2\}$ as initial point x_0 for the algorithm and notice that $\operatorname{sign}\hat{g}(x) = \pm(1,-1)$ for any $x \in l$. Hence, signSGD with any step-size sequence remains stuck along the line l, whereas the problem has a unique minimizer at the origin.

In this example, Assumption 1 is violated. Indeed, notice that $\operatorname{sign}\hat{g}(x) = (-1)^i \operatorname{sign}\langle a_i, x\rangle \left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right]$ with probabilities $1/2$ for $i=1,2$. By $S \coloneqq \{x\in\mathbb{R}^2 \colon \langle a_1,x\rangle\cdot\langle a_2,x\rangle > 0\} \neq \emptyset$ denote the open cone of points having either an acute or an obtuse angle with both a_i's. Then for any $x\in S$, the sign of the stochastic gradient is $\pm(1,-1)$ with probabilities $1/2$. Hence for any $x \in S$, we have low success probabilities:

$$\rho_i(x) = \operatorname{Prob}\left(\operatorname{sign}\hat{g}_i(x) = \operatorname{sign} g_i(x)\right) \le \tfrac{1}{2},\ i=1,2.$$

So, in this case we have an entire conic region with low success probabilities, which clearly violates (4). Furthermore, if we take a point from the complement open cone $\bar{S}^c$, then the sign of stochastic gradient equals to the sign of gradient, which is perpendicular to the axis of S (thus in the next step of the iteration we get closer to S). For example, if $\langle a_1,x\rangle < 0$ and $\langle a_2,x\rangle > 0$, then $\operatorname{sign}\hat{g}(x) = (1,-1)$ with probability 1, in which case $x - \gamma\operatorname{sign}\hat{g}(x)$ gets closer to low success probability region S.

3.3. Sufficient Conditions for SPB

To motivate our SPB assumption, we compare it with 4 different conditions commonly used in the literature and show that it holds under general assumptions on gradient noise. Below, we assume that for any point $x\in\mathbb{R}^d$, we have access to an independent and unbiased estimator $\hat{g}(x)$ of the true gradient $g(x) = \nabla f(x)$.

Lemma 1 (see C.1)**.** *If for each coordinate $\hat{g}_i$ has a unimodal and symmetric distribution with variance $\sigma_i^2 = \sigma_i^2(x)$, $1\le i\le d$ and $g_i \ne 0$, then*

$$\rho_i \ge \tfrac{1}{2} + \tfrac{1}{2}\tfrac{|g_i|}{|g_i|+\sqrt{3}\sigma_i} > \tfrac{1}{2}.$$

This is the setup used in Theorem 1 of Bernstein et al. (2019). We recover their result as a special case using Lemma 1 (see Appendix D). Next, we replace the distribution condition by coordinate-wise strong growth condition (SGC) (Schmidt & Le Roux, 2013; Vaswani et al., 2019) and fixed mini-batch size.

Lemma 2 (see C.2)**.** *Let coordinate-wise variances $\sigma_i^2(x) \le c_i\, g_i^2(x)$ are bounded for some constants c_i. Choose mini-batch size $\tau > 2\max_i c_i$. If further $g_i \ne 0$, then*

$$\rho_i \ge 1 - \tfrac{c_i}{\tau} > \tfrac{1}{2}.$$

Now we remove SGC and give an adaptive condition on mini-batch size for the SPB assumption to hold.

Lemma 3 (see C.3)**.** *Let $\sigma_i^2 = \sigma_i^2(x)$ be the variance and $\nu_i^3 = \nu_i^3(x)$ be the 3th central moment of $\hat{g}_i(x)$, $1\le i\le d$. Then SPB assumption holds if mini-batch size*

$$\tau > 2\min\left(\tfrac{\sigma_i^2}{g_i^2}, \tfrac{\nu_i^3}{|g_i|\sigma_i^2}\right).$$

Finally, we compare SPB with the standard bounded variance assumption in the sense of differential entropy.

Lemma 4 (see C.4)**.** *Differential entropy of a probability distribution under the bounded variance assumption is bounded, while under the SPB assumption it could be arbitrarily large.*

Remark 2. *Note that SPB assumption describes the convergence of sign descent methods, which is known to be problem dependent (e.g. see (Balles & Hennig, 2018), section 6.2 Results). One should view the SPB condition as a criteria to problems where sign based methods are useful.*

Remark 3. *Differential entropy argument is an attempt to bridge our new SPB assumption to one of the most basic assumptions in the literature, bounded variance assumption. Clearly, they are not comparable in the usual sense, and neither one is implied by the other. Still, we propose another viewpoint to the situation and compare such conditions through the lens of information theory. Practical meaning of such observation is that SPB handles a much broader (though not necessarily more important) class of gradient noise than bounded variance condition. In other words, this gives an intuitive measure on how much restriction we put on the noise.*

3.4. A New "Norm" for Measuring the Gradients

We introduce a norm-like function ρ-*norm*, induced from success probabilities and use it to measure gradients in our convergence rates.

Definition 1 (ρ-norm)**.** *Let $\rho \coloneqq \{\rho_i(x)\}_{i=1}^d$ be the probability functions from the SPB assumption. We define the ρ-norm of gradient $g(x)$ via*

$$\|g(x)\|_\rho \coloneqq \textstyle\sum_{i=1}^d (2\rho_i(x)-1)|g_i(x)|.$$

Although, in general, ρ-norm is not a norm in classical sense, it can be reduced to one in special cases. For example, the setup of Lemma 1 allows to lower bound ρ-norm by a mixture of l^1 and squared l^2 norms, denoted by $l^{1,2}$:

$$\|g\|_\rho = \sum_{i=1}^d (2\rho_i - 1)|g_i| \ge \sum_{i=1}^d \frac{g_i^2}{|g_i|+\sqrt{3}\sigma_i} \coloneqq \|g\|_{l^{1,2}}. \quad (7)$$

To understand the nature of the $l^{1,2}$ norm, consider the following two cases when $\sigma_i(x) \le c|g_i(x)| + \tilde{c}$ for some constants $c,\ \tilde{c} \ge 0$. If the iterations are in ε-neighbourhood of a minimizer x^* with respect to the l^∞ norm (i.e., $\max_{1\le i\le d}|g_i| \le \varepsilon$), then the $l^{1,2}$ norm is equivalent to scaled l^2 norm squared:

$$\tfrac{1}{(1+\sqrt{3}c)\varepsilon+\sqrt{3}\tilde{c}}\|g\|_2^2 \le \|g\|_{l^{1,2}} \le \tfrac{1}{\sqrt{3}\tilde{c}}\|g\|_2^2.$$

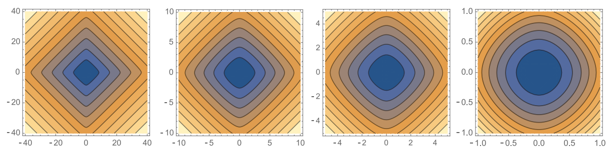

Figure 1. Contour plots of the mixed $l^{1,2}$ norm (7) at 4 different scales with fixed noise $\sigma = 1$.

On the other hand, if iterations are away from a minimizer (i.e., $\min_{1\le i\le d}|g_i| \ge L$), then the $l^{1,2}$-norm is equivalent to scaled l^1 norm:

$$\frac{1}{1+\sqrt{3}(c+\tilde{c}/L)}\|g\|_1 \le \|g\|_{l^{1,2}} \le \frac{1}{1+\sqrt{3}c}\|g\|_1.$$

These equivalences are visible in Figure 1, where we plot the level sets of $g \mapsto \|g\|_{l^{1,2}}$ at various distances from the origin. Similar mixed norm observation for signSGD was also noted by Bernstein et al. (2019) and Chen et al. (2020). Alternatively, under the setup of Lemma 2, ρ-norm reduces to weighted l^1 norm.

$$\|g\|_\rho = \sum_{i=1}^d (2\rho_i - 1)|g_i| \ge \sum_{i=1}^d (1 - \tfrac{2c_i}{\tau})|g_i|. \quad (8)$$

4. Convergence Theory

Now we turn to our theoretical results of sign based methods. First we give our general convergence rates under the SPB assumption. Afterwards, we extend the theory to parallel setting under the same SPB assumption with majority vote aggregation. Finally, we explain the convergence issue of signSGD in distributed training with partitioned data and propose a new sign based method, *SSDM*, to resolve it.

Algorithm 1 SIGNSGD

1: **Input:** step size γ_k, current point x_k
2: $\hat{g}_k \leftarrow \text{StochasticGradient}(f, x_k)$
3: $\hat{s}_k = \text{sign}\, \hat{g}_k$
4: *Option 1:* $x_{k+1} = x_k - \gamma_k \hat{s}_k$
5: *Option 2:* $x_{k+1} = \arg\min\{f(x_k), f(x_k - \gamma_k \hat{s}_k)\}$

Throughout the paper we assume that nonconvex $f\colon \mathbb{R}^d \to \mathbb{R}$ is lower bounded, i.e., $f(x) \ge f^*$ for all $x \in \mathbb{R}^d$.

4.1. Convergence Analysis for $M = 1$

We start our convergence theory with single node setting, where f is smooth with some non-negative constants $(L_1, \dots, L_d)$, i.e.,

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \sum_{i=1}^d \tfrac{L_i}{2}(y_i - x_i)^2$$

for all x, $y \in \mathbb{R}^d$. Denote $\bar{L} := \frac{1}{d}\sum_i L_i$.

Theorem 1 (see C.5)**.** *Under the SPB assumption, single node signSGD (Algorithm 1) with Option 1 and with step sizes $\gamma_k = \gamma_0/\sqrt{k+1}$ converges as follows*

$$\min_{0\le k<K} \mathbb{E}\|\nabla f(x_k)\|_\rho \le \frac{f(x_0)-f^*}{\gamma_0\sqrt{K}} + \frac{3\gamma_0 d\bar{L}}{2}\frac{\log K}{\sqrt{K}}. \quad (9)$$

If $\gamma_k \equiv \gamma > 0$, we get $1/K$ convergence to a neighbourhood:

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla f(x_k)\|_\rho \le \frac{f(x_0)-f^*}{\gamma K} + \frac{\gamma d\bar{L}}{2}. \quad (10)$$

We now comment on the above result:

• **Generalization.** Theorem 1 is the first general result on signSGD for non-convex functions without mini-batching, and with step sizes independent of the total number of iterations K. Known convergence results (Bernstein et al., 2018; 2019) on signSGD use mini-batches and/or step sizes dependent on K. Moreover, they also use unbiasedness and unimodal symmetric noise assumptions, which are stronger assumptions than our SPB assumption (see Lemma 1). Finally, Theorem 1 in (Bernstein et al., 2019) can be recovered from Theorem 1 (see Appendix D).

• **Convergence rate.** Rates (9) and (10) can be arbitrarily slow, depending on the probabilities ρ_i. This is to be expected. At one extreme, if the gradient noise was completely random, i.e., if $\rho_i \equiv 1/2$, then the ρ-norm would become identical zero for any gradient vector and rates would be trivial inequalities, leading to divergence as in the counterexample. At other extreme, if there was no gradient noise, i.e., if $\rho_i \equiv 1$, then the ρ-norm would be just the l^1 norm and we get the rate $\mathcal{O}(1/\sqrt{K})$ with respect to the l^1 norm. However, if we know that $\rho_i > 1/2$, then we can ensure that the method will eventually converge.

Theorem 1 can be further simplified under the setup of Lemma 1 (see Corollary 1) and Lemma 2 (see Corollary 2). We now state a general convergence rate for Algorithm 1 with Option 2.

Theorem 2 (see C.6)**.** *Under the SPB assumption, signSGD (Algorithm 1) with Option 2 and with step sizes $\gamma_k = \gamma_0/\sqrt{k+1}$ converges as follows:*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla f(x_k)\|_\rho \le \frac{1}{\sqrt{K}}\left[\frac{f(x_0)-f^*}{\gamma_0} + \gamma_0 d\bar{L}\right].$$

In the case of $\gamma_k \equiv \gamma > 0$, the same rate as (10) *is achieved.*

Comparing Theorem 2 with Theorem 1, notice that one can remove the log factor from (9) and bound the average of past gradient norms instead of the minimum. On the other hand, in a big data regime, function evaluations in Algorithm 1 (Option 2, line 4) are infeasible. Clearly, Option 2 is useful *only* in the setup when one can afford function evaluations and has rough estimates about the gradients (i.e., signs of stochastic gradients). This option should be considered within the framework of derivative-free optimization.

4.2. Convergence Analysis in Parallel Setting

In this part we present the convergence result of parallel signSGD (Algorithm 2) with majority vote introduced by Bernstein et al. (2018). Majority vote is considered within a parameter server framework, where for each coordinate parameter server receives one sign from each node and sends back the sign sent by the majority of nodes. In parallel setting, the training data is shared among the nodes.

Algorithm 2 PARALLEL SIGNSGD W/ MAJORITY VOTE

Input: step size γ_k, current point x_k, # of nodes M
on each node n
    $\hat{g}^n(x_k) \leftarrow \text{StochasticGradient}(f, x_k)$
on server
    get sign $\hat{g}^n(x_k)$ **from** all nodes
    send sign $\left[\sum_{n=1}^M \operatorname{sign} \hat{g}^n(x_k)\right]$ **to** all nodes
on each node n
    $x_{k+1} = x_k - \gamma_k \operatorname{sign}\left[\sum_{n=1}^M \operatorname{sign} \hat{g}^n(x_k)\right]$

Known convergence results (Bernstein et al., 2018; 2019) use $\mathcal{O}(K)$ mini-batch size as well as $\mathcal{O}(1/K)$ constant step size. In the sequel we remove this limitations extending Theorem 1 to parallel training. In this case the number of nodes M get involved in geometry introducing new ρ_M-norm, which is defined by the regularized incomplete beta function I (see Appendix C.7).

Definition 2 (ρ_M-norm)**.** *Let M be the number of nodes and denote $l := \lfloor \frac{M+1}{2} \rfloor$. Define ρ_M-norm of gradient $g(x)$ at $x \in \mathbb{R}^d$ via*

$$\|g(x)\|_{\rho_M} := \sum_{i=1}^d \left(2I(\rho_i(x); l, l) - 1\right) |g_i(x)|.$$

Clearly, ρ_1-norm coincides with ρ-norm. Now we state the convergence rate of parallel signSGD with majority vote.

Theorem 3 (see C.7)**.** *Under SPB assumption, parallel signSGD (Algorithm 2) with step sizes $\gamma_k = \gamma_0/\sqrt{k+1}$ converges as follows*

$$\min_{0 \le k < K} \mathbb{E}\|\nabla f(x_k)\|_{\rho_M} \le \frac{f(x_0) - f^*}{\gamma_0 \sqrt{K}} + \frac{3\gamma_0 d \bar{L}}{2} \frac{\log K}{\sqrt{K}}. \quad (11)$$

For constant step sizes $\gamma_k \equiv \gamma > 0$, we have convergence up to a level proportional to step size γ:

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}\|\nabla f(x_k)\|_{\rho_M} \le \frac{f(x_0) - f^*}{\gamma K} + \frac{\gamma d \bar{L}}{2}. \quad (12)$$

• **Speedup with respect to M.** Note that, in parallel setting with M nodes, the only difference in convergence rates (11) and (12) is the modified ρ_M-norm measuring the size of gradients. Using Hoeffding's inequality, we show (see Appendix C.8) that $\|g(x)\|_{\rho_M} \to \|g(x)\|_1$ exponentially fast as $M \to \infty$, namely

$$\left(1 - e^{-(2\rho(x)-1)^2 l}\right) \|g(x)\|_1 \le \|g(x)\|_{\rho_M} \le \|g(x)\|_1,$$

where $\rho(x) = \min_{1 \le i \le d} \rho_i(x) > 1/2$. To appreciate the speedup with respect to M, consider the noise-free case as a baseline, for which $\rho_i \equiv 1$ and $\|g(x)\|_{\rho_M} \equiv \|g(x)\|_1$. Then, the above inequality implies that M parallel machines reduce the variance of gradient noise exponentially fast.

• **Number of Nodes.** Theoretically there is no difference between $2l-1$ and $2l$ nodes, and this is not a limitation of the analysis. Indeed, as it is shown in the proof, expected sign vector at the master with $M = 2l-1$ nodes is the same as with $M = 2l$ nodes:

$$\mathbb{E} \operatorname{sign}(\hat{g}_i^{(2l)} \cdot g_i) = \mathbb{E} \operatorname{sign}(\hat{g}_i^{(2l-1)} \cdot g_i),$$

where $\hat{g}^{(M)}$ is the sum of stochastic sign vectors aggregated from nodes. Intuitively, majority vote with even number of nodes, e.g. $M = 2l$, fails to provide any sign with little probability (it is the probability of half nodes voting for $+1$, and half nodes voting for -1). However, if we remove one node, e.g. $M = 2l-1$, then master receives one sign-vote less but gets rid of that little probability of failing the vote (sum of odd number of ± 1 cannot vanish).

4.3. Distributed Training with Partitioned Data

First, we briefly discuss the fundamental issue of signSGD in distributed environment and then present our new sign based method which resolves that issue.

The Issue with Distributed signSGD. Consider distributed training where each machine $n \in \{1, 2, \ldots, M\}$ has its own loss function $f_n(x)$. We argue that in this setting even signGD (with full-batch gradients and no noise) can fail to converge. Indeed, let us multiply each loss function $f_n(x)$ of nth node by an arbitrary positive scalars $w_n > 0$. Then the landscape (in particular, stationary points) of the overall loss function

$$f^w(x) := \frac{1}{M} \sum_{n=1}^M w_n f_n(x)$$

can change arbitrarily while the iterates of signGD remain the same as the master server aggregates the same signs $\operatorname{sign}(w_n \nabla f_n(x)) = \operatorname{sign} \nabla f_n(x)$ regardless of the scalars $w_n > 0$. Thus, distributed signGD is unable to sense the weights $w_n > 0$ modifying total loss function f^w and cannot guarantee approximate stationary point unless loss functions f_n have some special structures.

Novel Sign-based Method for Distributed Training. The above issue of distributed signSGD stems from the biasedness of the sign operator which completely ignores the magnitudes of local gradients of all nodes. We resolve this issue by designing a novel distributed sign-based method–*Stochastic Sign Descent with Momentum (SSDM)*–including two additional layers: *stochastic sign* and *momentum*.

Motivated by SPB assumption, we introduce our new notion of *stochastic sign* to replace the usual deterministic sign.

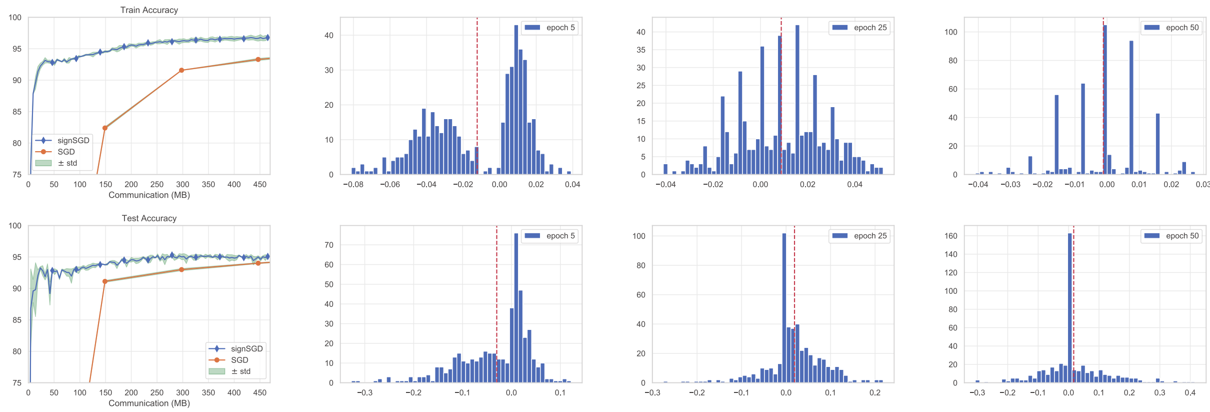

Figure 2. Convergence of signSGD and comparison with SGD on the MNIST dataset using the split batch construction strategy. The budget of gradient communication (MB) is fixed and the network is a single hidden layer FNN. We first tuned the constant step size over logarithmic scale $\{1, 0.1, 0.01, 0.001, 0.0001\}$ and then fine tuned it. First column shows train and test accuracies with mini-batch size 128 and averaged over 3 repetitions. We chose two weights (empirically, most of the network biases would work) and plotted histograms of stochastic gradients before epochs 5, 25 and 50. Dashed red lines on histograms indicate the average values.

Definition 3 (Stochastic Sign)**.** *We define the stochastic sign operator* $\widetilde{\text{sign}}: \mathbb{R}^d \to \mathbb{R}^d$ *via*

$$\left(\widetilde{\text{sign}}\, g\right)_i = \begin{cases} +1, & \textit{with probability } \frac{1}{2} + \frac{1}{2}\frac{g_i}{\|g\|} \\ -1, & \textit{with probability } \frac{1}{2} - \frac{1}{2}\frac{g_i}{\|g\|} \end{cases}$$

for $1 \le i \le d$ *and* $\widetilde{\text{sign}}\,\mathbf{0} = \mathbf{0}$ *with probability* 1.

Technical importance of stochastic $\widetilde{\text{sign}}$ is twofold. First, it satisfies the SPB assumption automatically, that is

$$\text{Prob}((\widetilde{\text{sign}}\, g)_i = \text{sign}\, g_i) = \tfrac{1}{2} + \tfrac{1}{2}\tfrac{|g_i|}{\|g\|} > \tfrac{1}{2},$$

if $g_i \neq 0$. Second, unlike the deterministic sign operator, it is unbiased with scaling factor $\|g\|$, namely $\mathbb{E}[\|g\|\, \widetilde{\text{sign}}\, g] = g$. We describe our SSDM method formally in Algorithm 3.

Algorithm 3 SSDM

Input: step size parameter γ, momentum parameter β, # of nodes M
Initialize: $x_0 \in \mathbb{R}^d$, $m_{-1}^n = \hat{g}_0^n$
for $k = 0, 1, \ldots, K-1$ **do**
  on each node n
    $\hat{g}_k^n \leftarrow \text{StochasticGradient}(f_n, x_k)$ Local sub-sampling
    $m_k^n = \beta m_{k-1}^n + (1-\beta)\hat{g}_k^n$ Update the momentum
    send $s_k^n := \widetilde{\text{sign}}\, m_k^n$ **to** the server
  on server
    send $s_k := \sum_{n=1}^M s_k^n$ **to** all nodes
  on each node n
    $x_{k+1} = x_k - \frac{\gamma}{M} s_k$ Main step: Update the global model
end for

Consider the optimization problem (1), where each node n owns only the data associated with loss function $f_n : \mathbb{R}^d \to \mathbb{R}$, which is non-convex and L^n-smooth. We model stochastic gradient oracle using the standard bounded variance condition defined below:

Assumption 2 (Bounded Variance)**.** *For any* $x \in \mathbb{R}^d$*, each node* n *has access to an unbiased estimator* $\hat{g}^n(x)$ *with bounded variance* $\sigma_n^2 \ge 0$*, namely*

$$\mathbb{E}\left[\hat{g}^n(x)\right] = \nabla f_n(x), \quad \mathbb{E}\left[\|\hat{g}^n(x) - \nabla f_n(x)\|^2\right] \le \sigma_n^2.$$

Now, we present our convergence result for SSDM method.

Theorem 4 (see C.9)**.** *Under Assumption 2,* $K \ge 1$ *iterations of SSDM (Algorithm 3) with momentum parameter* $\beta = 1 - \frac{1}{\sqrt{K}}$ *and step-size* $\gamma = \frac{1}{K^{3/4}}$ *guarantee*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla f(x^k)\| \le \frac{16}{K^{1/4}}\left[\delta_f + \tilde{\sigma} + \tilde{L}\sqrt{d} + \frac{\tilde{L}d}{\sqrt{K}}\right],$$

where $\delta_f = f(x_0) - f^*, \tilde{\sigma} = \frac{1}{M}\sum_{n=1}^M \sigma_n, \tilde{L} = \frac{1}{M}\sum_{n=1}^M L^n$.

Let us comment on the above rate of SSDM.

• **Optimal rate using sign bits only.** Note that, for non-convex distributed training, SSDM has the same optimal asymptotic rate $\mathcal{O}(\varepsilon^{-4})$ as SGD. In contrast, signSGD and its momentum version Signum (Bernstein et al., 2018; 2019) were not analyzed in distributed setting where data is partitioned between nodes and require increasingly larger mini-batches over the course of training. A general approach to handle biased compression operators, satisfying certain

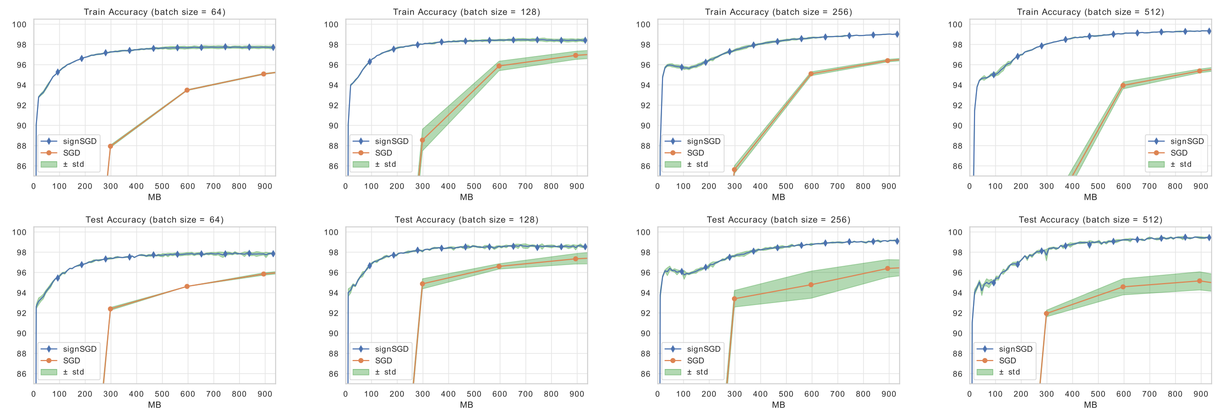

Figure 3. Comparison of signSGD and SGD on the MNIST dataset with a fixed budget of gradient communication (MB) using single hidden layer FNN and the standard batch construction strategy. For each batch size, we first tune the constant step size over logarithmic scale $\{10, 1, 0.1, 0.01, 0.001\}$ and then fine tune it. Shaded area shows the standard deviation from 3 repetition.

contraction property, is the *error feedback (EF)* mechanism proposed by Seide et al. (2014). In particular, EF-signSGD method of Karimireddy et al. (2019) fixes the convergence issues of signSGD in single node setup, overcoming SBP assumption. Furthermore, for distributed training, Tang et al. (2019) applied the error feedback trick both for the server and nodes in their DoubleSqueeze method maintaining the same asymptotic rate with bi-directional compression. However, in these methods, the contraction property of compression operator used by error feedback forces to communicate the magnitudes of local stochastic gradients together with the signs. This is not the case for sign-based methods considered in this work, where only sign bits are communicated between nodes and server.

• **Noisy signSGD.** In some sense, stochastic sign operator (see Definition 3) can be viewed as noisy version of standard deterministic sign operator and, similarly, our SSDM method can be viewed as noisy variant of signSGD with momentum. This observation reveals a connection to the noisy signSGD method of Chen et al. (2020). Despite some similarities between the two methods, there are several technical aspects that SSDM excels their noisy signSGD. First, the noise they add is *artificial* and requires a special care: too much noise blows the convergence, too little noise is unable to shrink the gap between median and mean. Moreover, as it is discussed in their paper, the variance of the noise must depend on K (total number of iterations) and tend to ∞ with K to guarantee convergence to stationary points in the limit. Meanwhile, the noise of SSDM is *natural* and does not need to be adjusted. Next, the convergence bound (17) of (Chen et al., 2020) is harder to interpret than the bound in our Theorem 4 involving l_2 norms of the gradients *only*. Besides, the convergence rate with respect to squared l_2 norm is $\mathcal{O}(d^{3/4}/K^{1/4})$, while the rate of SSDM with respect to squared l_2 norm is $\mathcal{O}(d/\sqrt{K})$, which is $\mathcal{O}(K^{1/4}/d^{1/4})$ times *faster*. Lastly, it is explicitly written before Theorem 5 that the analysis assumes *full* gradient computation for all nodes. In contrast, SSDM is analyzed under a more general stochastic gradient oracle.

• **All-reduce compatible.** In contrast to signSGD with majority vote aggregation, SSDM supports partial aggregation of compressed stochastic signs s_k^n. In other words, compressed signs s_k^n can be directly summed without additional decompression-compression steps. This allows SSDM to be implemented with efficient *all-reduce* operation instead of slower *all-gather* operation. Besides SSDM, only a few compression schemes in the literature satisfy this property and can be implemented with *all-reduce* operation, e.g., SGD with random sparsification (Wangni et al., 2018), GradiVeQ (Yu et al., 2018), PowerSGD (Vogels et al., 2019).

Finally, we show that the improved convergence theory and low communication cost of SSDM is due to the use of *both* stochastic sign operator and momentum term.

• **SSDM without stochastic sign.** If we replace stochastic sign by deterministic sign in SSDM, then the resulting method *can provably diverge* even when full gradients are computed by all nodes. In fact, the counterexample (5)-(6) in Section 3.2 can be easily extended to distributed setting and can handle momentum. Indeed, consider $M = 2$ nodes owning functions $f_n(x) = \langle a_n, x\rangle^2$, $n = 1, 2$ with a_1, a_2 as defined in (6) and initial point $x_0 \in l = \{(z_1, z_2)\colon z_1 + z_2 = 2\}$. Since $\nabla f_n(x) = 2\langle a_n, x\rangle a_n \in span(a_n)$, we imply $m_k^n \in span(a_n)$ for any value of parameter β and for all iterate $k \geq 0$ (see lines 2 and 6 of Algorithm 3). Hence, $\operatorname{sign} m_k^n = \pm \operatorname{sign} a_n = \pm \left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right]$. Since $s_k = \operatorname{sign} m_k^1 + \operatorname{sign} m_k^2 \in span(\left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right])$ (see line 9), this means that the method is again stuck along the line l as

$\frac{\gamma}{M}s_k \in span(\left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right])$ (see line 11) for any value of γ.

• **SSDM without momentum.** It is possible to obtain the same asymptotic convergence rate without the momentum term (i.e., $\beta = 0$). In this case, if all nodes also send the norms $\|\hat{g}_k^n\|$ to the server then the method can be analyzed by a standard analysis of distributed SGD with an unbiased compression. However, the drawback of this approach is the *higher communication cost*. While the overhead of worker-to-server communication is negligible (one extra float), the reverse server-to-worker communication becomes costly as the aggregated updates are dense (all entries are floats) as opposed to the original SSDM (all entries are integers).

5. Experiments

We verify several aspects of our theoretical results experimentally using the MNIST dataset with feed-forward neural network (FNN) and the well known Rosenbrock (non-convex) function with $d = 10$ variables:

$$f(x) = \sum_{i=1}^{d-1} f_i(x) = \sum_{i=1}^{d-1} 100(x_{i+1}-x_i^2)^2+(1-x_i)^2.$$

5.1. Minimizing the Rosenbrock Function

The Rosenbrock function is a classic example of non-convex function, which is used to test the performance of optimization methods. We chose this low dimensional function in order to estimate the success probabilities effectively in a reasonable time and to expose theoretical connection.

Stochastic formulation of the minimization problem for Rosenbrock function is as follows: at any point $x \in \mathbb{R}^d$ we have access to *biased* stochastic gradient $\hat{g}(x) = \nabla f_i(x) + \xi$, where index i is chosen uniformly at random from $\{1, 2, \ldots, d-1\}$ and $\xi \sim \mathcal{N}(0, \nu^2 I)$ with $\nu > 0$.

Figure 4 illustrates the effect of multiple nodes in distributed training with majority vote. As we see increasing the number of nodes improves the convergence rate. It also supports the claim that in expectation there is no improvement from $2l - 1$ nodes to $2l$ nodes. More experiments on the Rosenbrock function are moved to Appendix A.

5.2. Training FNN on the MNIST Dataset

We trained a single layer feed-forward network on the MNIST with two different batch construction strategies. The first construction is the standard way of training networks: before each epoch we shuffle the training dataset and choose batches sequentially. In the second construction, first we split the training dataset into two parts, images with labels 0, 1, 2, 3, 4 and images with labels 5, 6, 7, 8, 9. Then each batch of images were chosen from one of these parts with equal probabilities. We make the following observations based on our experiments depicted in Figure 2 and Figure 3.

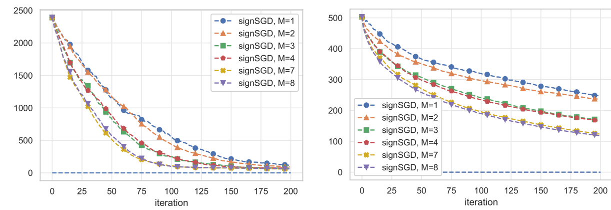

Figure 4. Experiments on distributed signSGD with majority vote using Rosenbrock function. Plots show function values with respect to iterations averaged over 10 repetitions. Left plot used constant step size $\gamma = 0.02$, right plot used variable step size with $\gamma_0 = 0.02$. We set mini-batch size 1 and used the same initial point. Dashed blue lines mark the minimum.

• **Convergence with multi-modal and skewed gradient distributions.** Due to the split batch construction strategy we unfold multi-modal and asymmetric distributions for stochastic gradients in Figure 2. With this experiment we conclude that sign based methods can converge under various gradient distributions which is allowed from our theory.

• **Effectiveness in the early stage of training.** Both experiments show that in the beginning of the training, signSGD is more efficient than SGD when we compare accuracy against communication. This observation is supported by the theory as at the start of the training success probabilities are bigger (see Lemma 1) and lower bound for mini-batch size is smaller (see Lemma 3).

• **Bigger batch size, better convergence.** Figure 3 shows that the training with larger batch size improves the convergence as backed by the theory (see Lemmas 2 and 3).

• **Generalization effect.** Another aspect of sign based methods which has been observed to be problematic, in contrast to SGD, is the generalization ability of the model (see also (Balles & Hennig, 2018), Section 6.2 Results). In the experiment with standard batch construction (see Figure 3) we notice that test accuracy is growing with training accuracy. However, in the other experiment with split batch construction (see Figure 2), we found that test accuracy does not get improved during the second half of the training while train accuracy grows consistently with slow pace.

Acknowledgments

Mher Safaryan thanks El Houcine Bergou, Konstantin Mishchenko, Samuel Horváth and anonymous reviewers for helpful discussions about the topic and their useful feedback on the paper.

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 30*, pp. 1709–1720, 2017.

Balles, L. and Hennig, P. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 404–413, 2018.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signSGD: Compressed optimisation for nonconvex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 560–569. PMLR, 2018.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. signSGD with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2019.

Bottou, L. and Le Cun, Y. Large scale online learning. In *Advances in Neural Information Processing Systems*, 2003.

Carlson, D., Cevher, V., and Carin, L. Stochastic spectral descent for restricted boltzmann machines. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 111–119, 2015.

Chen, X., Chen, T., Sun, H., Wu, Z. S., and Hong, M. Distributed training with heterogeneous data: Bridging median- and mean-based algorithms. In *34th Conference on Neural Information Processing Systems*, 2020.

Cutkosky, A. and Mehta, H. Momentum improves normalized SGD. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2260–2268, Virtual, 13–18 Jul 2020. PMLR. URL http://proceedings.mlr.press/v119/cutkosky20b.html.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research*, pp. 2121–2159, 2011.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. In *SIAM Journal on Optimization*, volume 23(4), pp. 2341–2368, 2013.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes SignSGD and other gradient compression schemes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3252–3261, 2019.

Khirirat, S., Feyzmahdavian, H. R., and Johansson, M. Distributed learning with compressed gradients. In *arXiv preprint arXiv:1806.06573*, 2018.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, W. J. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

Liu, S., Chen, P.-Y., Chen, X., and Hong, M. signSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019.

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

Qian, X., Richtárik, P., Gower, R. M., Sailanbayev, A., Loizou, N., and Shulgin, E. SGD with arbitrary sampling: General analysis and improved rates. In *International Conference on Machine Learning*, 2019.

Reddi, S., Kale, S., and Kumar, S. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2019.

Riedmiller, M. and Braun, H. A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In *IEEE International Conference on Neural Networks*, pp. 586–591, 1993.

Robbins, H. and Monro, S. A stochastic approximation method. In *The Annals of Mathematical Statistics*, volume 22(3), pp. 400–407, 1951.

Schmidhuber, J. Deep learning in neural networks: An overview. In *Neural networks*, volume 61, pp. 85–117, 2015.

Schmidt, M. and Le Roux, N. Fast convergence of stochastic gradient descent under a strong growth condition. In *arXiv preprint arXiv:1308.6370*, 2013.

Schmidt, M., Roux, N. L., and Bach, F. Minimizing finite sums with the stochastic average gradient. In *Mathematical Programming*, volume 162(1-2), pp. 83–112, 2017.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and application to data-parallel distributed training of speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Shevtsova, I. On the absolute constants in the berry–esseen type inequalities for identically distributed summands. In *arXiv preprint arXiv:1111.6554*, 2011.

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

Tang, H., Yu, C., Lian, X., Zhang, T., and Liu, J. `DoubleSqueeze`: Parallel stochastic gradient descent with double-pass error-compensated compression. In *Int. Conf. Machine Learning*, volume PMLR 97, pp. 6155–6165, 2019.

Tieleman, T. and Hinton, G. E. RMSprop. In *Coursera: Neural Networks for Machine Learning, Lecture 6.5*, 2012.

Vaswani, S., Bach, F., and Schmidt, M. Fast and faster convergence of SGD for over-parameterized models (and an accelerated perceptron). In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, PMLR*, volume 89, 2019.

Vershynin, R. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

Vogels, T., Karimireddy, S. P., and Jaggi, M. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *33th Advances in Neural Information Processing Systems*, 2019.

Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, 2018.

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In *32th Advances in Neural Information Processing Systems*, 2018.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural Information Processing Systems*, pp. 1509–1519, 2017.

Wilson, A., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

Yu, M., Lin, Z., Narra, K., Li, S., Li, Y., Kim, N. S., Schwing, A., Annavaram, M., and Avestimehr, S. GradiVeQ: Vector quantization for bandwidth-efficient gradient aggregation in distributed CNN training. In *32th Advances in Neural Information Processing Systems*, 2018.

Zaheer, M., Reddi, S., Sachan, D., Kale, S., and Kumar, S. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 9815–9825, 2018.

Zeiler, M. D. ADADELTA: An Adaptive Learning Rate Method. In *arXiv e-prints, arXiv:1212.5701*, 2012.

Zhang, H., Li, J., Kara, K., Alistarh, D., Liu, J., and Zhang, C. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 4035–4043, 2017.