Error Feedback Fixes SignSGD and other Gradient Compression Schemes

Sai Praneeth Karimireddy [1] **Quentin Rebjock** [1] **Sebastian U. Stich** [1] **Martin Jaggi** [1]

Abstract

Sign-based algorithms (e.g. SIGNSGD) have been proposed as a biased gradient compression technique to alleviate the communication bottleneck in training large neural networks across multiple workers. We show simple convex counter-examples where signSGD does not converge to the optimum. Further, even when it does converge, signSGD may generalize poorly when compared with SGD. These issues arise because of the biased nature of the sign compression operator.

We then show that using error-feedback, i.e. incorporating the error made by the compression operator into the next step, overcomes these issues. We prove that our algorithm (EF-SGD) with arbitrary compression operator achieves the *same rate of convergence* as SGD without any additional assumptions. Thus EF-SGDachieves gradient compression *for free*. Our experiments thoroughly substantiate the theory.

1. Introduction

Stochastic optimization algorithms (Bottou, 2010) which are amenable to large-scale parallelization, taking advantage of massive computational resources (Krizhevsky et al., 2012; Dean et al., 2012) have been at the core of significant recent progress in deep learning (Schmidhuber, 2015; LeCun et al., 2015). One such example is the SIGNSGD algorithm and its variants, c.f. (Seide et al., 2014; Bernstein et al., 2018; 2019).

To minimize a continuous (possibly) non-convex function $f\colon \mathbb{R}^d \to \mathbb{R}$, the classic stochastic gradient algorithm (SGD) (Robbins & Monro, 1951) performs iterations of the form

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma\, \mathbf{g}_t\,, \qquad \text{(SGD)}$$

where $\gamma \in \mathbb{R}$ is the step-size (or learning-rate) and $\mathbf{g}_t$ is the stochastic gradient such that $\mathbb{E}[\mathbf{g}_t] = \nabla f(\mathbf{x}_t)$.

[1]EPFL, Switzerland. Correspondence to: Sai Praneeth Karimireddy <sai.karimireddy@epfl.ch>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Algorithm 1 EF-SIGNSGD (SIGNSGD with Error-Feedb.)

Input: learning rate γ, initial iterate $\mathbf{x}_0 \in \mathbb{R}^d$, $\mathbf{e}_0 = \mathbf{0}$
for $t = 0, \dots, T-1$ **do**
  $\mathbf{g}_t :=$ stochasticGradient$(\mathbf{x}_t)$
  $\mathbf{p}_t := \gamma \mathbf{g}_t + \mathbf{e}_t$ ▷ *error correction*
  $\Delta_t := (\|\mathbf{p}_t\|_1 / d)\,\mathrm{sign}(\mathbf{p}_t)$ ▷ *compression*
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \Delta_t$ ▷ *update iterate*
  $\mathbf{e}_{t+1} := \mathbf{p}_t - \Delta_t$ ▷ *update residual error*
end for

Methods performing updates only based on the *sign* of each coordinate of the gradient have recently gaining popularity for training deep learning models (Seide et al., 2014; Carlson et al., 2015; Wen et al., 2017; Balles & Hennig, 2018; Bernstein et al., 2018; Zaheer et al., 2018; Liu et al., 2018). For example, the update step of SIGNSGD is given by:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma\, \mathrm{sign}(\mathbf{g}_t)\,. \qquad \text{(SIGNSGD)}$$

Such sign-based algorithms are particularly interesting since they can be viewed through two lenses: as i) approximations of adaptive gradient methods such as ADAM (Balles & Hennig, 2018), and also a ii) communication efficient gradient compression scheme (Seide et al., 2014). However, we show that a severe handicap of sign-based algorithms is that they *do not* converge in general. To substantiate this claim, we present in this work simple convex counter-examples where SIGNSGD cannot converge. The main reasons being that the sign operator loses information about, i.e. 'forgets', i) the magnitude, as well as ii) the direction of $\mathbf{g}_t$. We present an elegant solution that provably fixes these problems of SIGNSGD, namely algorithms with *error-feedback*.

Error-feedback. We demonstrate that the aforementioned problems of SIGNSGD can be fixed by i) scaling the signed vector by the norm of the gradient to ensure the magnitude of the gradient is not forgotten, and ii) locally storing the difference between the actual and compressed gradient, and iii) adding it back into the next step so that the correct direction is not forgotten. We call our 'fixed' method EF-SIGNSGD (Algorithm 1).

In Algorithm 1, $\mathbf{e}_t$ denotes the accumulated error from all quantization/compression steps. This residual error is added to the gradient step $\gamma \mathbf{g}_t$ to obtain the corrected direction $\mathbf{p}_t$.

When compressing $\mathbf{p}_t$, the signed vector is again scaled by $\|\mathbf{p}_t\|_1$ and hence does not lose information about the magnitude. Note that our algorithm does not introduce any additional parameters and requires only the step-size γ.

Our contributions. We show that naively using biased gradient compression schemes (such as e.g. SIGNSGD) can lead to algorithms which may not generalize or even converge. We show both theoretically and experimentally that simply adding error-feedback solves such problems and recovers the performance of full SGD, thereby saving on communication costs. We state our results for SIGNSGD to ease our exposition but our positive results are valid for general compression schemes, and our counterexamples extend to SIGNSGD with momentum, multiple worker settings, and even other biased compression schemes. More specifically our contributions are:

1. We construct a simple convex non-smooth counterexample where SIGNSGD cannot converge, even with the full batch (sub)-gradient and tuning the step-size. Another counterexample for a wide class of smooth convex functions proves that SIGNSGD with stochastic gradients cannot converge with batch-size one.
2. We prove that by incorporating error-feedback, SIGNSGD—as well as any other gradient compression schemes—always converge. Further, our analysis for non-convex smooth functions recovers the same rate as SGD, i.e. we get gradient compression *for free*.
3. We show that our algorithm EF-SIGNSGD which incorporates error-feedback approaches the linear span of the past gradients. Therefore, unlike SIGNSGD, EF-SIGNSGD converges to the max-margin solution in over-parameterized least-squares. This provides a theoretical justification for why EF-SIGNSGD can be expected to generalize much better than SIGNSGD.
4. We show extensive experiments on CIFAR10 and CIFAR100 using Resnet and VGG architectures demonstrating that EF-SIGNSGD indeed significantly outperforms SIGNSGD, and matches SGD both on test as well as train datasets while reducing communication by a factor of $\sim 64\times$.

2. Significance and Related Work

Relation to adaptive methods. Introduced in (Kingma & Ba, 2015), ADAM has gained immense popularity as the algorithm of choice for adaptive stochastic optimization for its perceived lack of need for parameter-tuning. However since, the convergence (Reddi et al., 2018) as well the generalization performance (Wilson et al., 2017) of such adaptive algorithms has been called into question. Understanding when ADAM performs poorly and providing a principled 'fix' for these cases is crucial given its importance as the algorithm of choice for many researchers. It was recently noted by Balles & Hennig (2018) that the behavior of ADAM is in fact identical to that of SIGNSGD with *momentum*: More formally, the SIGNSGDM algorithm (referred to as 'signum' by Bernstein et al. (2018; 2019)) adds momentum to the SIGNSGD update as:

$$\begin{aligned} \mathbf{m}_{t+1} &:= \mathbf{g}_t + \beta \mathbf{m}_t \\ \mathbf{x}_{t+1} &:= \mathbf{x}_t - \gamma \operatorname{sign}(\mathbf{m}_{t+1})\,. \end{aligned} \qquad \text{(SIGNSGDM)}$$

for parameter $\beta > 0$. This connection between signed methods and fast stochastic algorithms is not surprising since sign-based gradient methods were first studied as a way to speed up SGD (Riedmiller & Braun, 1993). Given their similarity, understanding the behavior of SIGNSGD and SIGNSGDM can help shed light on ADAM.

Relation to gradient compression methods. As the size of the models keeps getting bigger, the training process can often take days or even weeks (Dean et al., 2012). This process can be significantly accelerated by massive parallelization (Li et al., 2014; Goyal et al., 2017). However, at these scales communication of the gradients between the machines becomes a bottleneck hindering us from making full use of the impressive computational resources available in today's data centers (Chilimbi et al., 2014; Seide et al., 2014; Strom, 2015). A simple solution to alleviate this bottleneck is to compress the gradient and reduce the number of bits transmitted. While the analyses of such methods have largely been restricted to *unbiased* compression schemes (Alistarh et al., 2017; Wen et al., 2017; Wang et al., 2018), biased schemes which perform extreme compression practically perform much better (Seide et al., 2014; Strom, 2015; Lin et al., 2018)—often *without any loss* in convergence or accuracy. Of these, (Seide et al., 2014; Strom, 2015; Wen et al., 2017) are all sign-based compression schemes. Interestingly, all the practical works (Seide et al., 2014; Strom, 2015; Lin et al., 2018) use some form of error-feedback.

Error-feedback. The idea of error-feedback was, as far as we are aware, first introduced in 1-bit SGD (Seide et al., 2014; Strom, 2015). The algorithm 1-bit SGD is very similar to our EF-SIGNSGD algorithm, but tailored for the specific recurrent network studied there. Though not presented as such, the 'momentum correction' used in (Lin et al., 2018) is a variant of error-feedback. However the error-feedback is not on the vanilla SGD algorithm, but on SGD with momentum. Recently, Stich et al. (2018) conducted the first theoretical analysis of error-feedback for compressed gradient methods (they call it 'memory') in the strongly convex case. Our convergence results can be seen as an extension of theirs to the non-convex and weakly convex cases.

Generalization of deep learning methods. Deep networks are almost always over-parameterized and are known to be able to fit arbitrary data and always achieve zero training error (Zhang et al., 2017). This ability of deep networks to generalize well on real data, while simultaneously being able to fit arbitrary data has recently received a lot of attention (e.g. Soudry et al. (2018); Dinh et al. (2017); Zhang et al. (2018); Arpit et al. (2017); Kawaguchi et al. (2017)). SIGNSGD and ADAM are empirically known to generalize worse than SGD (Wilson et al., 2017; Balles & Hennig, 2018). A number of recent papers try close this gap for ADAM. Luo et al. (2019) show that by bounding the adaptive step-sizes in ADAM leads to closing the generalization gap. They require new hyper-parameters on top of ADAM to adaptively tune these bounds on the step-sizes. Chen & Gu (2019) interpolate between SGD and ADAM using a new hyper-parameter p and show that tuning this can recover performance of SGD. Zaheer et al. (2018) introduce a new adaptive algorithm which is closer to Adagrad (Duchi et al., 2011). Similarly, well-tuned ADAM (where all the hyper-parameters and not just the learning rate are tuned) is also known to close the generalization gap (Gugger & Howard, 2018). In all of these algorithms, new hyper-parameters are introduced which essentially control the effect of the adaptivity. Thus they require additional tuning while the improvement upon traditional SGD is questionable. We are not aware of other work bridging the generalization gap in sign-based methods.

3. Counterexamples for SignSGD

In this section we study the limitations of SIGNSGD. Under benign conditions—for example if i) the function f is smooth, and ii) the stochastic noise is gaussian or an extemely large batch-size is used (equal to the total number of iterations)—the algorithm can be shown to converge (Bernstein et al., 2018; 2019). However, we show that SIGNSGD does not converge under more standard assumptions. We demonstrate this first on a few pedagogic examples and later also for realistic and general sum-structured loss functions.

If we use a fixed step-size $\gamma \geq 0$, SIGNSGD does not converge even for simple one-dimensional linear functions.

Counterexample 1. For $x \in \mathbb{R}$ consider the constrained problem

$$\min_{x\in[-1,1]} \left[f(x) := \tfrac{1}{4}x\right] ,$$

with minimum at $x^\star = -1$. Assume stochastic gradients are given as (note that $f(x) = \frac{1}{4}(4x - x - x - x)$)

$$g = \begin{cases} 4, & \text{with prob. } \frac{1}{4} \\ -1, & \text{with prob. } \frac{3}{4} \end{cases} \quad \text{with} \quad \mathbb{E}[g] = \nabla f(x)\,.$$

For SGD with any step-size γ,

$$\mathbb{E}_t[f(x_{t+1})] = \frac{1}{4}(x_t - \gamma\,\mathbb{E}[g]) = f(x_t) - \frac{\gamma}{16}\,.$$

On the other hand, for SIGNSGD with any fixed γ,

$$\mathbb{E}_t[f(x_{t+1})] = \frac{1}{4}(x_t - \gamma\,\mathbb{E}[\text{sign}(g)]) = f(x_t) + \frac{\gamma}{8}\,,$$

i.e. the objective function increases in expectation for $\gamma \geq 0$.

Remark 1. *In the above example, we exploit that the* sign *operator loses track of the magnitude of the stochastic gradient. Also note that our noise is bimodal. The counter-examples for the convergence of ADAM (Reddi et al., 2018; Luo et al., 2019) also use similar ideas. Such examples were previouslyt noted for* SIGNSGD *by (Bernstein et al., 2019).*

In the example above the step-size γ was fixed. However increasing batch-size or tuning the step-size may still allow convergence. Next we show that even with adaptive step-sizes (e.g. decreasing, or adaptively chosen optimal step-sizes) SIGNSGD does not converge. This even holds if the full (sub)-gradient is available (non-stochastic case).

Counterexample 2. For $\mathbf{x} \in \mathbb{R}^2$ consider the following non-smooth convex problem with $\mathbf{x}^\star = (0,0)^\top$:

$$\min_{\mathbf{x}\in\mathbb{R}^2} \left[f(\mathbf{x}) := \epsilon|x_1 + x_2| + |x_1 - x_2|\right] ,$$

for parameter $0 < \epsilon < 1$ and subgradient

$$\mathbf{g}(\mathbf{x}) = \text{sign}(x_1 + x_2)\cdot\epsilon\begin{pmatrix}1\\1\end{pmatrix} + \text{sign}(x_1 - x_2)\begin{pmatrix}1\\-1\end{pmatrix}\,.$$

See Fig. 1. The iterates of SIGNSGD started at $\mathbf{x}_0 = (1,1)^\top$ lie along the line $x_1 + x_2 = 2$. Note that for any $\mathbf{x}$ s.t. $x_1 + x_2 > 0$, $\text{sign}(\mathbf{g}(\mathbf{x})) = \pm(1,-1)^\top$, and hence $x_1 + x_2$ remains constant among the iterations of SIGNSGD. Consequently, for any step-size sequence γ_t, $f(\mathbf{x}_t) \geq f(\mathbf{x}_0)$.

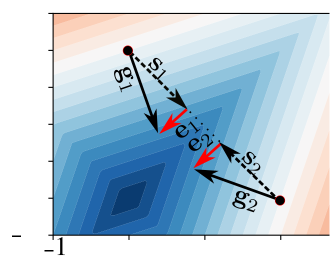

Figure 1: The gradients $\mathbf{g}$ (in solid black), singed gradient direction $\mathbf{s} = \text{sign}(\mathbf{g})$ (in dashed black), and the error $\mathbf{e}$ (in red) are plotted for $\epsilon = 0.5$. SIGNSGD moves only along $\mathbf{s} = \pm(1,-1)$ while the error $\mathbf{e}$ is ignored.

Remark 2. *In this example, we exploit the fact that the sign operator is a biased approximation of the gradient—it consistently ignores the direction* $\mathbf{e} = \epsilon(1,1)^\top$ *(see Fig 1). Tuning the step-size would not help either.*

One might wonder if the smooth-case is easier. Unfortunately, the previous example can easily be extended to show that SIGNSGD with stochastic gradients may not converge even for smooth functions.

Counterexample 3. For $\mathbf{x} \in \mathbb{R}^2$ consider the following least-squares problem with $\mathbf{x}^\star = (0, 0)^\top$:

$$\min_{\mathbf{x} \in \mathbb{R}^2} \left[f(\mathbf{x}) := (\langle \mathbf{a}_1, \mathbf{x} \rangle)^2 + (\langle \mathbf{a}_2, \mathbf{x} \rangle)^2 \right] \text{ , where}$$

$$\mathbf{a}_{1,2} := \pm(1, -1) + \epsilon(1, 1) \,,$$

for parameter $0 < \epsilon < 1$ and stochastic gradient $\mathbf{g}(\mathbf{x}) = \nabla_{\mathbf{x}}(\langle \mathbf{a}_1, \mathbf{x} \rangle)^2$ with prob. $\frac{1}{2}$ and $\mathbf{g}(\mathbf{x}) = \nabla_{\mathbf{x}}(\langle \mathbf{a}_2, \mathbf{x} \rangle)^2$ with prob. $\frac{1}{2}$. The stochastic gradient is then either $e\mathbf{a}_1$ or $e\mathbf{a}_2$ for some scalar e. Exactly as in the non-smooth case, for $\mathbf{x}_0 = (1, 1)^\top$, the sign of the gradient $\mathrm{sign}(\mathbf{g}) = \pm(1, -1)$. Hence SIGNSGD with any step-size sequence remains stuck along the line $x_1 + x_2 = 2$ and $f(\mathbf{x}_t) \geq f(\mathbf{x}_0)$ a.s.

We can generalize the above counter-example to arbitrary dimensions and loss functions.

Theorem I. *Suppose that scalar loss functions $\{l_i : \mathbb{R} \to \mathbb{R}\}_{i=1}^n$ and data-points $\{\mathbf{a}_i\}_{i=1}^n \in \mathbb{R}^d$ for $d \geq 2$ satisfy: i) $f(\mathbf{x}) := \sum_{i=1}^n l_i(\langle \mathbf{a}_i, \mathbf{x} \rangle)$ has a unique optimum at $\mathbf{x}^\star$, and ii) there exists $\mathbf{s} \in \{-1, 1\}^d$ such that $\mathrm{sign}(\mathbf{a}_i) = \pm\mathbf{s}$ for all i. Then SIGNSGD with batch-size 1 and stochastic gradients $g(\mathbf{x}) = \nabla_{\mathbf{x}} l_i(\langle \mathbf{a}_i, \mathbf{x} \rangle)$ for i chosen uniformly at random does not converge to $\mathbf{x}^\star$ a.s. for any adaptive sequence of step-sizes, even with random initialization.*

4. Convergence of Compressed Methods

We show the rather surprising result that incorporating error-feedback is sufficient to ensure that the algorithm converges at a rate which matches that of SGD. In this section we consider a general gradient compression scheme.

Algorithm 2 EF-SGD (Compr. SGD with Error-Feedback)

Input: learning rate γ, compressor $\mathcal{C}(\cdot)$, $\mathbf{x}_0 \in \mathbb{R}^d$
Initialize: $\mathbf{e}_0 = \mathbf{0} \in \mathbb{R}^d$
for $t = 0, \dots, T-1$ **do**
  $\mathbf{g}_t :=$ stochasticGradient$(\mathbf{x}_t)$
  $\mathbf{p}_t := \gamma \mathbf{g}_t + \mathbf{e}_t$ ▷ *error correction*
  $\Delta_t := \mathcal{C}(\mathbf{p}_t)$ ▷ *compression*
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \Delta_t$ ▷ *update iterate*
  $\mathbf{e}_{t+1} := \mathbf{p}_t - \Delta_t$ ▷ *update residual error*
end for

We generalize the notion of a compressor from (Stich et al., 2018).

Assumption A (Compressor)**.** *An operator $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a δ-approximate compressor over $\mathcal{Q}$ for $\delta \in (0, 1]$ if*

$$\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1 - \delta)\|\mathbf{x}\|_2^2, \ \forall \mathbf{x} \in \mathcal{Q} \,.$$

Note that $\delta = 1$ implies that $\mathcal{C}(\mathbf{x}) = \mathbf{x}$. Examples of compressors include: i) the sign operator, ii) top-k which selects k coordinates with the largest absolute value while zero-ing out the rest (Lin et al., 2018; Stich et al., 2018), iii) k-PCA which approximates a matrix $\mathbf{X}$ with its top k eigenvectors (Wang et al., 2018). Randomized compressors satisfying the assumption in expectation are also allowed.

We now state a key lemma that shows that the residual errors maintained in Algorithm 2 do not accumulate too much.

Lemma 3 (Error is bounded)**.** *Assume that $\mathbb{E}[\|\mathbf{g}_t\|^2] \leq \sigma^2$ for all $t \geq 0$. Then at any iteration t of EF-SGD, the norm of the error $\mathbf{e}_t$ in Algorithm 2 is bounded:*

$$\mathbb{E}\|\mathbf{e}_t\|_2^2 \leq \frac{4(1-\delta)\gamma^2\sigma^2}{\delta^2} \,, \qquad \forall t \geq 0 \,.$$

If $\delta = 1$, then $\|\mathbf{e}_t\| = 0$ and the error is zero as expected.

We employ standard assumptions of smoothness of the loss function and the variance of the stochastic gradient.

Assumption B (Smoothness)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if for all $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^d$ the following holds:*

$$|f(\mathbf{y}) - (f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle)| \leq \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \,.$$

Assumption C (Moment bound)**.** *For any $\mathbf{x}$, our query for a stochastic gradient returns $\mathbf{g}$ such that*

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}) \quad \textit{and} \quad \mathbb{E}\|\mathbf{g}\|_2^2 \leq \sigma^2 \,.$$

Given these assumptions, we can formally state our theorem followed by a sketch of the proof.

Theorem II (Non-convex convergence of EF-SGD)**.** *Let $\{\mathbf{x}_t\}_{t \geq 0}$ denote the iterates of Algorithm 2 for any step-size $\gamma > 0$. Under Assumptions A, B, and C,*

$$\min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{2 f_0}{\gamma(T+1)} + \frac{\gamma L \sigma^2}{2} + \frac{4\gamma^2 L^2 \sigma^2 (1-\delta)}{\delta^2} \,,$$

with $f_0 := f(\mathbf{x}_0) - f^\star$.

Proof Sketch. Intuitively, the condition that $\mathcal{C}(\cdot)$ is a δ-approximate compressor implies that at each iteration a δ-fraction of the gradient information is sent. The rest is added to $\mathbf{e}_t$ to be transmitted later. Eventually, all the gradient information is transmitted—albeit with a delay which depends on δ. Thus EF-SGD can intuitively be viewed as a delayed gradient method. If the function is smooth, the gradient does not change quickly and so the delay does not significantly matter.

More formally, consider the error-corrected sequence $\tilde{\mathbf{x}}_t$ which represents $\mathbf{x}_t$ with the 'delayed' information added:

$$\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{e}_t \,.$$

It satisfies the recurrence

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{x}_t - \mathbf{e}_{t+1} - \mathcal{C}(\mathbf{p}_t) = \mathbf{x}_t - \mathbf{p}_t = \tilde{\mathbf{x}}_t - \gamma \mathbf{g}_t \,.$$

If $\mathbf{x}_t$ was exactly equal to $\tilde{\mathbf{x}}_t$ (i.e. there was zero 'delay'), then we could proceed with the standard proof of SGD. We instead rely on Lemma 3 which shows $\tilde{\mathbf{x}}_t \approx \mathbf{x}_t$ and on the smoothness of f which shows $\nabla f(\mathbf{x}_t) \approx \nabla f(\tilde{\mathbf{x}}_t)$. □

Remark 4. *If we substitute $\gamma := \frac{1}{\sqrt{T+1}}$ in Theorem II, we get*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{4(f(\mathbf{x}_0) - f^\star) + L\sigma^2}{2\sqrt{T+1}} + \frac{4L^2\sigma^2(1-\delta)}{\delta^2(T+1)}$$

In the above rate, the compression factor δ only appears in the higher order $\mathcal{O}(1/T)$ term. For comparison, SGD under the exact same assumptions achieves

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{2(f(\mathbf{x}_0) - f^\star) + L\sigma^2}{2\sqrt{T+1}} \,.$$

This means that after $T \geq \mathcal{O}(1/\delta^2)$ iterations the second term becomes negligible and the rate of convergence catches up with full SGD—this is usually true after just the first few epochs. Thus we prove that compressing the gradient does not change the asymptotic rate of SGD.[1]

Remark 5. *The use of error-feedback was motivated by our counter-examples for biased compression schemes. However our rates show that even if using an unbiased compression (e.g. QSGD (Alistarh et al., 2017)), using error-feedback gives significantly better rates. Suppose we are given an unbiased compressor $cU(\cdot)$ such that $\mathbb{E}[\mathcal{U}(\mathbf{x})] = \mathbf{x}$ and $\mathbb{E}\left[\|\mathcal{U}(\mathbf{x})\|_2^2\right] \leq k\|\mathbf{x}\|^2$. Then without feedback, using standard analysis (e.g. (Alistarh et al., 2017)) the algorithm converges k times slower:*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{(f(\mathbf{x}_0) - f^\star) + Lk\sigma^2}{2\sqrt{T+1}} \,.$$

Instead, if we use $\mathcal{C}(\mathbf{x}) = \frac{1}{k}\mathcal{U}(\mathbf{x})$ with error-feedback, we would achieve

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{4(f(\mathbf{x}_0) - f^\star) + L\sigma^2}{2\sqrt{T+1}} + \frac{2L^2\sigma^2k^2}{T+1} \,,$$

thereby pushing the dependence on k into the higher order $\mathcal{O}(1/T)$ term.

Our counter-examples showed that biased compressors may not converge for non-smooth functions. Below we prove that adding error-feedback ensures convergence under standard assumptions even for non-smooth functions.

Theorem III (Non-smooth convergence of EF-SGD)**.** *Let $\{\mathbf{x}_t\}_{t\geq 0}$ denote the iterates of Algorithm 2 for any step-size $\gamma > 0$ and define $\bar{\mathbf{x}}_t = \frac{1}{T}\sum_{t=0}^{T}\mathbf{x}_t$. Given that f is convex and Assumptions A and C hold,*

$$\mathbb{E}[f(\bar{\mathbf{x}}_t)] - f^\star \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{2\gamma(T+1)} + \gamma\sigma^2\left(\frac{1}{2} + \frac{2\sqrt{1-\delta}}{\delta}\right) .$$

Remark 6. *By picking the optimal $\gamma = \mathcal{O}(1/\sqrt{T})$, we see that*

$$\mathbb{E}[f(\bar{\mathbf{x}}_t)] - f^\star \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|\sigma}{2\sqrt{T+1}}\sqrt{1 + \frac{4\sqrt{1-\delta}}{\delta}} \,.$$

For comparison, the rate of convergence under the same assumptions for SGD is

$$\mathbb{E}[f(\bar{\mathbf{x}}_t)] - f^\star \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|\sigma}{2\sqrt{T+1}} \,.$$

For non-smooth functions, unlike in the smooth case, the compression quality δ appears directly in the leading term of the convergence rate. This is to be expected since we can no longer assume that $\nabla f(\tilde{\mathbf{x}}_t) \approx \nabla f(\mathbf{x}_t)$, which formed the crux of our argument for the smooth case.

Remark 7. *Consider the top-1 compressor which just picks the coordinate with the largest absolute value, and zeroes out everything else. It is obvious that top-1 is a $\frac{1}{d}$-approximate compressor (cf. (Stich et al., 2018, Lemma A.1)). Running* EF-SGD *with $\mathcal{C}$ as top-1 results in a greedy coordinate algorithm. This is the first result we are aware which shows the convergence of a greedy-coordinate type algorithm on non-smooth functions.*

If the function is both smooth and convex, we can fall back to the analysis of (Stich et al., 2018) and so we won't examine it in more detail here.

Convergence of EF-SIGNSGD

What do our proven rates imply for EF-SIGNSGD (Algorithm 1), the method of our interest here?

Lemma 8 (Compressed sign)**.** *The operator $\mathcal{C}(\mathbf{v}) := \frac{\|\mathbf{v}\|_1}{d}\,\mathrm{sign}(\mathbf{v})$ is a*

$$\phi(\mathbf{v}) = \frac{\|\mathbf{v}\|_1^2}{d\|\mathbf{v}\|_2^2}$$

compressor.

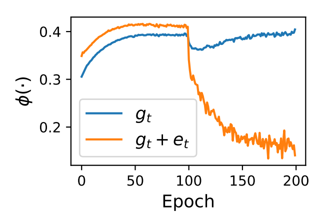

Figure 2: The density $\phi(\cdot)$ for the stochastic gradients $\mathbf{g}_t$ and the error-corrected stochastic gradients $\mathbf{g}_t + \mathbf{e}_t$ for VGG19 on CIFAR10 and batchsize 128. Minimum value is above 0.13. See Sec. 6 for setup.

We refer to the quantity $\phi(\mathbf{v})$ as the density of $\mathbf{v}$. If the vector $\mathbf{v}$ had only one non-zero element, the value of δ for EF-SIGNSGD could be as bad as $1/d$. However, in deep learning the gradients are usually dense and hence $\phi(\mathbf{v})$ is much larger (see Fig. 2). Note that for our convergence rates, it is not the density of the gradient $\mathbf{g}_t$ which matters but the density of the error-corrected gradient $\mathbf{g}_t + \mathbf{e}_t$.

[1] The astute reader would have observed that the asymptotic rate for EF-SGD is in fact 2 times slower than SGD. This is just for simplicity of presentation and can easily be fixed with a tighter analysis.

Faster convergence than SGD? (Kingma & Ba, 2015) and (Bernstein et al., 2018) note that different coordinates of the stochastic gradient $\mathbf{g}$ may have different variances. In standard SGD, the learning rate γ would be reduced to account for the *maximum* of these coordinate-wise variances since otherwise the path might be dominated by the noise in these sparse coordinates. Instead, using coordinate-wise learning-rates like ADAM does, or using only the coordinate-wise sign of $\mathbf{g}$ as SIGNSGD does, might mitigate the effect of such 'bad' coordinates by effectively scaling down the noisy coordinates. This is purported to be the reason why ADAM and SIGNSGD can be faster than SGD on train dataset.

In EF-SIGNSGD, the noise from the 'bad' coordinates gets accumulated in the error-term $\mathbf{e}_t$ and is not forgotten or scaled down. Thus, if there are 'bad' coordinates whose variance slows down convergence of SGD, EF-SIGNSGD should be similarly slow. Confirming this, in a toy experiment with sparse noise (Appendix A.1), SGD and EF-SIGNSGD converge at the same *slower* rate, whereas SIGNSGD is faster. However, our real world experiments contradict this—even with the feedback, EF-SIGNSGD is consistently *faster* than SGD, SIGNSGD, and SIGNSGDM on training data. Thus the coordinate-wise variance adaption explanation proposed by (Bernstein et al., 2018; Kingma & Ba, 2015) does not explain the faster convergence of EF-SIGNSGD, and is probably an incomplete explanation of why sign based and adaptive methods are faster than SGD!

5. Generalization of SignSGD

So far our discussion has mostly focused on the convergence of the methods i.e. their performance on training data. However for deep-learning, we actually care about their performance on test data i.e. their generalization. It has been observed that the optimization algorithm being used significantly impacts the properties of the optima reached (Im et al., 2016; Li et al., 2018). For instance, ADAM and SIGNSGD are known to generalize poorly compared with SGD (Wilson et al., 2017; Balles & Hennig, 2018).

The proposed explanation for this phenomenon is that in an over-parameterized setting, SGD reaches the 'max-margin' solution wheras ADAM and SIGNSGD do not (Zhang et al., 2017; Wilson et al., 2017), and (Balles & Hennig, 2018). As with the issues in convergence, the issues of SIGNSGD with generalization also turn out to be related to the biased nature of the sign operator. We explore how error-feedback may also alleviate the issues with generalization for any compression operator.

5.1. Distance to gradient span

Like (Zhang et al., 2017; Wilson et al., 2017), we consider an over-parameterized least-squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[f(\mathbf{x}) := \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2\right]\,,$$

where $\mathbf{A}\in\mathbb{R}^{n\times d}$ for $d>n$ is the data matrix and $\mathbf{y}\in\{-1,1\}^n$ is the set of labels. The set of solutions $X^\star := \{\mathbf{x}\colon f(\mathbf{x})=0\}$ of this problem forms a subspace in $\mathbb{R}^d$. Of particular interest is the solution with smallest norm:

$$\arg\min_{\mathbf{x}\colon\in X^\star}\|\mathbf{x}\|^2 = \mathbf{A}^\dagger\mathbf{y} = \mathbf{A}^\top\left(\mathbf{A}\mathbf{A}^\top\right)^{-1}\mathbf{y}\,,$$

as this corresponds to the *maximum margin* solution in the dual.

Maximizing margin is known to have a regularizing effect and is said to improve generalization (Valiant, 1984; Cortes & Vapnik, 1995).

The key property that SGD (with or without momentum) trivially satisfies is that the iterates always lie in the linear span of the gradients.

Lemma 9. *Given any over-parameterized least-squares problem, suppose that the iterates of the algorithm always lie in the linear span of the gradients and it converges to a 0 loss solution. This solution corresponds to the minimum norm/maximum margin solution.*

If we instead use a biased compressor (e.g. SIGNSGD), it is clear that the iterate may not lie in the span of the gradients. In fact it is easy to construct examples where this happens for SIGNSGD (Balles & Hennig, 2018), as well as top-k sparsification (Gunasekar et al., 2018), perhaps explaining the poor generalization of these schemes. Error-feedback is able to overcome this issue as well.

Theorem IV. *Suppose that we run Algorithm 2 for t iterations starting from $\mathbf{x}_0=\mathbf{0}$. Let $\mathbf{G}_t=[g_0^\top,\dots,g_{t-1}^\top]\in\mathbb{R}^{d\times t}$ denote the matrix of the stochastic gradients and let $\Pi_{\mathbf{G}_t}\colon\mathbb{R}^n\to\mathrm{Im}(\mathbf{G}_t)$ denote the projection onto the range of $\mathbf{G}_t$.*

$$\|\mathbf{x}_t-\Pi_{\mathbf{G}_t}(\mathbf{x}_t)\|_2^2\leq\|\mathbf{e}_t\|_2^2\,.$$

Here $\mathbf{e}_t$ is the error as defined in Algorithm 2. The theorem follows directly from observing that $(\mathbf{x}_{t+1}+\mathbf{e}_{t+1}) = (\mathbf{x}_0+\sum_{i=0}^t\gamma\mathbf{g}_i)$, and hence lies in the linear span of the gradients.

Remark 10. *Theorem IV along with Lemma 3 implies that the iterates of Algorithm 2 are always close to the linear span of the gradients.*

$$\|\mathbf{x}_t-\Pi_{\mathbf{G}_t}(\mathbf{x}_t)\|_2^2\leq\frac{4\gamma^2(1-\delta)}{\delta^2}\max_{i\in[t]}\|\mathbf{g}_i\|^2\,.$$

This distance further reduces as the algorithm progresses since the the step-size γ is typically reduced.

5.2. Simulations

We compare the generalization of the four algorithms with full batch gradient: i) SGD ii) SIGNSGD, iii) SIGNSGDM, and iv) EF-SIGNSGD. The data is generated as in (Wilson

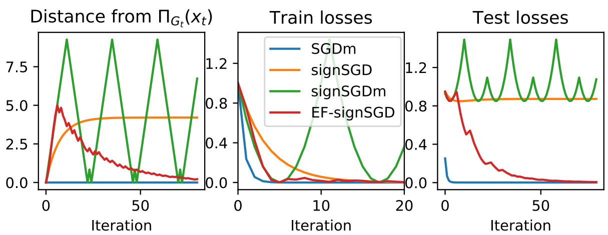

Figure 3: Left shows the distance of the iterate from the linear span of the gradients $\|\mathbf{x}_t - \Pi_{\mathbf{G}_t}(\mathbf{x}_t)\|$. The middle and the right plots show the train and test loss. SIGNSGD and SIGNSGDM have a high distance to the span, and do not generalize (test loss is higher than 0.8). Distance of EF-SIGNSGD to the linear span (and the test loss) goes to 0.

et al., 2017) and is randomly split into test and train. The step-size γ and (where applicable) the momentum parameter β are tuned to obtain fastest convergence, but the results are representative across parameter choices.

In all four cases (Fig. 3), the train loss quickly goes to 0. The distance to the linear span of gradients is quite large for SIGNSGD and SIGNSGDM. For EF-SIGNSGD, exactly as predicted by our theory, it first increases to a certain limit and then goes to 0 as the algorithm converges. The test error, almost exactly corresponding to the distance $\|\mathbf{x}_t - \Pi_{\mathbf{G}_t}(\mathbf{x}_t)\|$, goes down to 0. SIGNSGDM oscillates significantly because of the momentum term, however the conclusion remains unchanged—the best test error is higher than 0.8. This indicates that using error-feedback might result in generalization performance comparable with SGD.

6. Experiments

We run experiments on deep networks to test the validity of our insights. The results confirm that i) EF-SIGNSGD with error feedback always outperforms the standard SIGNSGD and SIGNSGDM, ii) the generalization gap of SIGNSGD and SIGNSGDM vs. SGD gets larger for smaller batch sizes, iii) the performance of EF-SIGNSGD on the other hand is much closer to SGD, and iv) SIGNSGD behaves erratically when using small batch-sizes.

6.1. Experimental setup

All our experiments used the PyTorch framework (Paszke et al., 2017) on the CIFAR-10/100 dataset (Krizhevsky & Hinton, 2009). Each experiment is repeated three times and the results are reported in Fig 4. Additional details and experiments can be found in Appendix A.

Algorithms: We experimentally compared the following four algorithms: i) SGDM which is SGD with momentum, ii) (scaled)SIGNSGD with step-size scaled by the L_1-norm of the gradient, iii) SIGNSGDM which is SIGNSGD using momentum, and iv) EF-SIGNSGD (Alg. 1). The scaled SIGNSGD performs the update

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \frac{\|\mathbf{g}_t\|_1}{d} \operatorname{sign}(\mathbf{g}_t)\,.$$

We chose to include this in our experiments since we wanted to isolate the effects of error-feedback from that of scaling. Further we drop the unscaled SIGNSGD from our discussion here since it was observed to perform worse than the scaled version. As is standard in compression schemes (Seide et al., 2014; Lin et al., 2018; Wang et al., 2018), we apply our compression layer-wise. Thus the net communication for the (scaled) SIGNSGD and EF-SIGNSGD is $\sum_{i=1}^{l}(d_i + 32)$ bits where d_i is the dimension of layer i, and l is the total number of layers. If the total number of parameters is much larger than the number of layers, then the cost of the extra $32l$ bits is negligible—usually the number of parameters is three orders of magnitude more than the number of layers.

All algorithms are run for 200 epochs. The learning-rate is decimated at 100 epochs and then again at 150 epochs. The initial learning rate is tuned manually (see Appendix A) for all algorithms using batch-size 128. For the smaller batch-sizes, the learning-rate is proportionally reduced as suggested in (Goyal et al., 2017). The momentum parameter β (where applicable) was fixed to 0.9 and weight decay was left to the default value of 5×10^{-4}.

Models: We use the VGG+BN+Dropout network on CIFAR-10 (VGG19) from (Simonyan & Zisserman, 2014) and Resnet+BN+Dropout network (Resnet18) from (He et al., 2016a). We adopt the standard data augmentation scheme and preprocessing scheme (He et al., 2016a;b). Our code builds upon on an open source library.[2]

6.2. Results

Batch-size	SGDM	SIGNSGD	SIGNSGDM	EF-SIGNSGD
128	75.35	-2.21	-3.15	**-0.92**
32	76.22	-3.04	-3.57	**-0.79**
8	74.91	-36.35	-6.6	**-0.64**

Table 1: Generalization gap on CIFAR-100 using Resnet18 for different batch-sizes. For SGDM we report the best mean test accuracy percentage, and for the other algorithms we report their difference to the SGDM accuracy (i.e. the generalization gap). EF-SIGNSGD has a much smaller gap.

The results of the experiments for Resnet18 on Cifar100 are shown in Fig. 4 and Table 1. Results for VGG19 on Cifar10 are also similar and can be found in the Appendix. We make four main observations:

EF-SIGNSGD is faster than SGDM on train. On the train dataset, both the accuracy and the losses (Fig. 6) is better for EF-SIGNSGD than for SGD for all batch-sizes

[2] github.com/kuangliu/pytorch-cifar

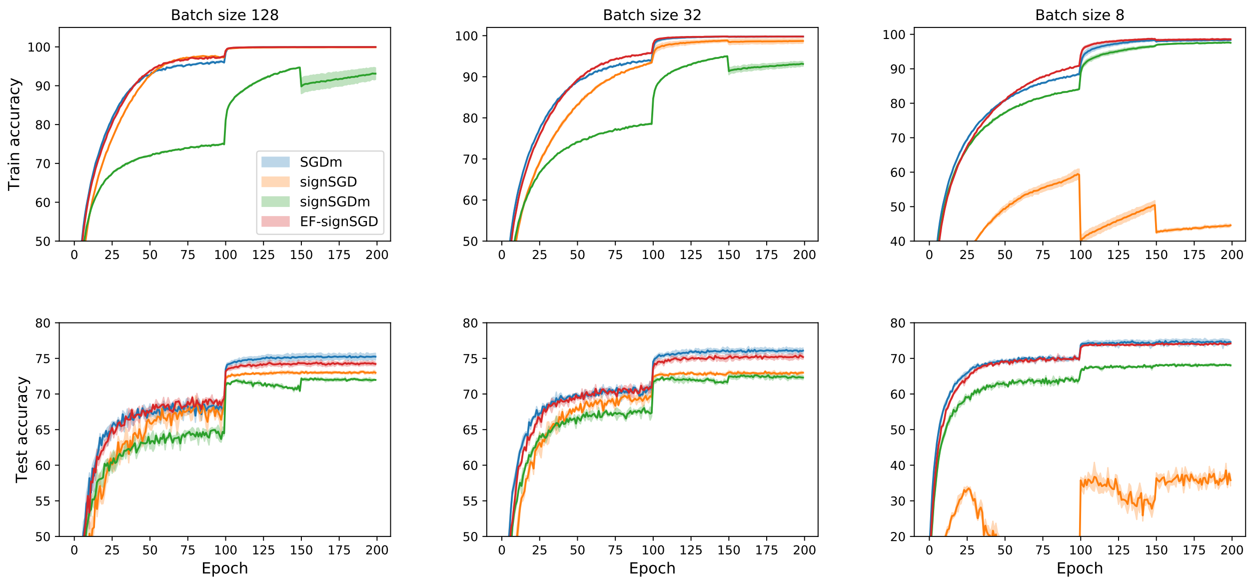

Figure 4: Experimental results showing the train and test accuracy percentages on CIFAR-100 using Resnet18 for different batch-sizes. The solid curves represent the mean value and shaded region spans one standard deviation obtained over three replications. Note that the scale of the y-axis varies across the plots. EF-SIGNSGD consistently and significantly outperforms the other sign-based methods, closely matching the performance of SGDM.

and models (Fig. 7). In fact even SIGNSGD is faster than SGDM on the train dataset on VGG19 (Fig. 7) for batch-size 128. As we note in Section 4, the result that the scaled sign methods are also faster than SGD (and in fact faster than even the without feedback algorithms) overturns previously understood intuition (cf. (Kingma & Ba, 2015; Bernstein et al., 2018)) for why SIGNSGD and other adaptive methods outperform SGD—i.e. restricting the effect of a some 'bad' coordinates with high variance may not be the main reason why sign based methods are faster than SGD on train.

EF-SIGNSGD almost matches SGDM on test. On the test dataset (Table 1), the accuracy and the loss is much closer to SGD than the other sign methods across batch-sizes and models (Tables 3, 4). The generalization gap (both in accuracy and loss) reduces with decreasing batch-size. We believe this is because the learning-rate was scaled proportional to the batch-size and hence smaller learning-rates lead to smaller generalization gap, as was theoretically noted in Remark 10.

SIGNSGD performs poorly for small batch-sizes. The performance of SIGNSGD is always worse than EF-SIGNSGD indicating that scaling is insufficient and that error-feedback is crucial for performance. Further all metrics (train and test, loss and accuracy) increasingly become worse as the batch-size decreases indicating that SIGNSGD is indeed a brittle algorithm. In fact for batch-size 8, the algorithm becomes extremely unstable.

SIGNSGDM performs poorly on some datasets and for smaller batch-sizes. We were surprised that the training performance of SIGNSGDM is significantly worse than even SIGNSGD on CIFAR-100 for batch-sizes 128 and 32. On CIFAR-10, on the other hand, SIGNSGDM manages to be faster than SGDM (though still slower than EF-SIGNSGD). We believe this may be due to SIGNSGDM being sensitive to the weight-decay parameter as was noted in (Bernstein et al., 2018). We do not tune the weight-decay parameter and leave it to its default value for all methods (including EF-SIGNSGD). Further the generalization gap of SIGNSGDM gets worse for decreasing batch-sizes with a significant 6.6% drop in accuracy for batch-size 8.

7. Conclusion

We study the effect of biased compressors on the convergence and generalization of stochastic gradient algorithms for non-convex optimization. We have shown that biased compressors if naively used can lead to bad generalization, and even non-convergence. We then show that using error-feedback all such adverse effects can be mitigated. Our theory and experiments indicate that using error-feedback, our compressed gradient algorithm EF-SGD enjoys the same rate of convergence as original SGD—thereby giving compression *for free*. We believe this should have a large impact in the design of future compressed gradient schemes for distributed and decentralized learning. Further, given the relation between sign-based methods and ADAM, we believe that our results will be relevant for better understanding the performance and limitations of adaptive methods. Finally, in this work we only consider the single worker case. Developing a practical and scalable algorithm for multiple workers is a fruitful direction for future work.

Acknowledgements. We are grateful to Tao Lin, Thijs Vogels, and Negar Foroutan for their help with the experiments. We also thank Jean-Baptiste Cordonnier, Konstantin Mishchenko, Jeremy Bernstein, and anonymous reviewers for their suggestions which helped improve our writeup.

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 (NIPS)*, 2017.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Balles, L. and Hennig, P. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *Internation Conference on Machine Learning (ICML)*, 2018.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signsgd: compressed optimisation for non-convex problems. In *Internation Conference on Machine Learning (ICML)*, 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 (ICLR)*, 2019.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In Lechevallier, Y. and Saporta, G. (eds.), *Proceedings of COMPSTAT'2010*, pp. 177–186, Heidelberg, 2010. Physica-Verlag HD. ISBN 978-3-7908-2604-3.

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, February 2015.

Chen, J. and Gu, Q. Padam: Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Chilimbi, T. M., Suzue, Y., Apacible, J., and Kalyanaraman, K. Project adam: Building an efficient and scalable deep learning training system. In *OSDI*, volume 14, pp. 571–582, 2014.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1223–1231, 2012.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Goyal, P., Dollar, 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.

Gugger, S. and Howard, J. Adamw and super-convergence is now the fastest way to train neural nets. https://www.fast.ai/2018/07/02/adam-weight-decay/, 2018. Accessed: 2019-01-17.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning (ICML)*, 2018.

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 (CVPR)*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pp. 630–645. Springer, 2016b.

Im, D. J., Tao, M., and Branson, K. An empirical analysis of the optimization of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*, 2016.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

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

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Technical Report, University of Toronto, Toronto., 2009.

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

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436, 2015.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., Long, J., Shekita, E. J., and Su, B.-Y. Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pp. 583–598, 2014.

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 (ICLR)*, 2018.

Liu, S., Chen, P.-Y., Chen, X., and Hong, M. signsgd via zeroth-order oracle. In *International Conference on Learning Representations (ICLR)*, 2018.

Luo, L., Xiong, Y., and Liu, Y. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations (ICLR)*, 2019.

Paszke, A., Gross, S., Chintala, S., and Chanan, G. Pytorch, 2017.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Robbins, H. and Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3): 400–407, September 1951.

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

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.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (70), 2018.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

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 (NeurIPS)*, 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 (NIPS)*, pp. 1509–1519, 2017.

Wilson, A. C., 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 (NIPS)*, pp. 4148–4158, 2017.

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

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Zhang, C., Vinyals, O., Munos, R., and Bengio, S. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018.

Appendix

A. Additional Experiments

In this section we give the full experimental details and results.

A.1. Convergence under sparse noise

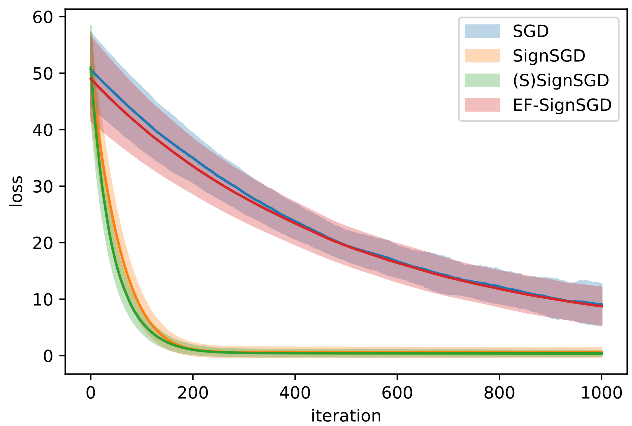

Figure 5: A simple toy problem where SIGNSGD and (scaled)SIGNSGD are faster than both SGD and EF-SIGNSGD. The experiment is repeated 100 times with mean indicated by the solid line and the shaded region spans one standard deviation. As in (Bernstein et al., 2018), the loss is $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ for $\mathbf{x} \in \mathbb{R}^{100}$, with gradient $\nabla f(\mathbf{x}) = \mathbf{x}$. The stochastic gradient is constructed by adding Gaussian noise $N(0, 100^2)$ to only the first coordinate of the gradient. The best learning-rate for SGD and EF-SIGNSGD was found to be 0.001, and for SIGNSGD and (scaled)SIGNSGD was 0.01. The conclusion of this toy experiment directly contradicts the results of our real-world experiments (Section 6) where EF-SIGNSGD is faster during training than both SGD and SIGNSGD. This shows that the sparse noisy coordinate explanation proposed by (Bernstein et al., 2018) is probably an incorrect explanation for the speed of sign based methods during training.

A.2. Description of models and datasets

The cifar dataset. The CIFAR 10 and 100 training and testing datasets was loaded using the default Pytorch torchvision api[3]. Data augmentation consisting of random 32×32 crops (padding 4) and horizontal flips was performed. Both sets were normalized over each separate channel.

VGG (on CIFAR 10). We used VGG19 architecture consisting in the following layers:
64 -> 64 -> M -> 128 -> 128 -> M -> 256 -> 256 -> 256 -> 256 -> M -> 512 -> 512 -> 512 -> 512 -> M -> 512 -> 512 -> 512 -> 512 -> M
where M denotes max pool layers (kernel 2 and stride 2), and each of the number n (either of 64/128/256/512) represents a two dimensional convolution layer with n channels a kernel of 3 and a padding of 1. All of them are followed by a batch normalization layer. Everywhere, ReLU activation is used.

Resnet (on CIFAR 100). We used a standard Resnet18 architecture with one convolution followed by four blocks and one dense layer [4].

A.3. Learning rate tuning

For all the experiments, the learning rate was divided by 10 at epochs 100 and again at 150. We tuned the initial learning rate on batchsize 128. The learning rates for batchsize 32 and 8 were scaled down by 4 and 16 respectivley. To tune the

[3] https://pytorch.org/docs/stable/torchvision/index.html
[4] https://github.com/kuangliu/pytorch-cifar/blob/master/models/resnet.py

initial learning rate, the algorithm was run with the same constant learingrate for 100 epochs. Then the learning rate which resulted in the best (i.e. smallest) test loss is chosen. The search space of possible learning rates was taken to be 9 values equally spaced in logarithmic scale over 10^{-5} to 10^{1} (inclusive).

The numbers below are rounded values (2 significant digits) of the actual learning rates:

$$1.0 \times 10^{-5}, 5.6 \times 10^{-5}, 3.2 \times 10^{-4}, 1.8 \times 10^{-3}, 1.0 \times 10^{-2}, 5.6 \times 10^{-2}, 3.2 \times 10^{-1}, 1.8 \times 10^{0}, 1.0 \times 10^{1} .$$

The best learning rate for each of the method is shown in table 2.

Algorithm	**Resnet18**	**VGG19**
SGDM	1.0×10^{-2}	1.0×10^{-2}
SIGNSGD	5.6×10^{-2}	5.6×10^{-2}
SIGNSGDM	3.2×10^{-4}	5.6×10^{-5}
EF-SIGNSGD	5.6×10^{-2}	5.6×10^{-2}

Table 2: The best initial learning rates for the four algorithms for batch size 128 on VGG19 (CIFAR 10 data) and Resnet18 (CIFAR 100 data).

A.4. Experiments with Resnet

We report the complete results (including the losses) for Resnet in Fig. 6.

		Algorithm			
		SGDM	**scaled SIGNSGD**	**SIGNSGDM**	**EF-SIGNSGD**
Batch size	128	75.35	-2.21	-3.15	**-0.92**
	32	76.22	-3.04	-3.57	**-0.79**
	8	74.91	-36.35	-6.6	**-0.64**

Table 3: Generalization gap on CIFAR-100 using Resnet18 for different batch-sizes. For SGDM we report the best mean test accuracy percentage, and for the other algorithms we report their difference to the SGDM accuracy (i.e. the generalization gap). EF-SIGNSGD has a much smaller gap which decreases with decreasing batchsize. The generalization gap of SIGNSGDM and SIGNSGD increases as the batchsize decreases.

A.5. Experiments with VGG

We report the complete results (including the losses) for VGG in Fig. 7.

		Algorithm			
		SGDM	**scaled SIGNSGD**	**SIGNSGDM**	**EF-SIGNSGD**
Batch size	128	93.38	-1.31	-0.94	**-0.68**
	32	93.42	-1.49	-1.54	**-0.71**
	8	93.09	-20.22	-2.75	**-0.27**

Table 4: Generalization gap on CIFAR-10 using VGG19 for different batch-sizes. For SGDM we report the best mean test accuracy percentage, and for the other algorithms we report their difference to the SGDM accuracy (i.e. the generalization gap). EF-SIGNSGD has a much smaller gap which decreases with decreasing batchsize. The generalization gap of SIGNSGDM and SIGNSGD increases as the batchsize decreases.

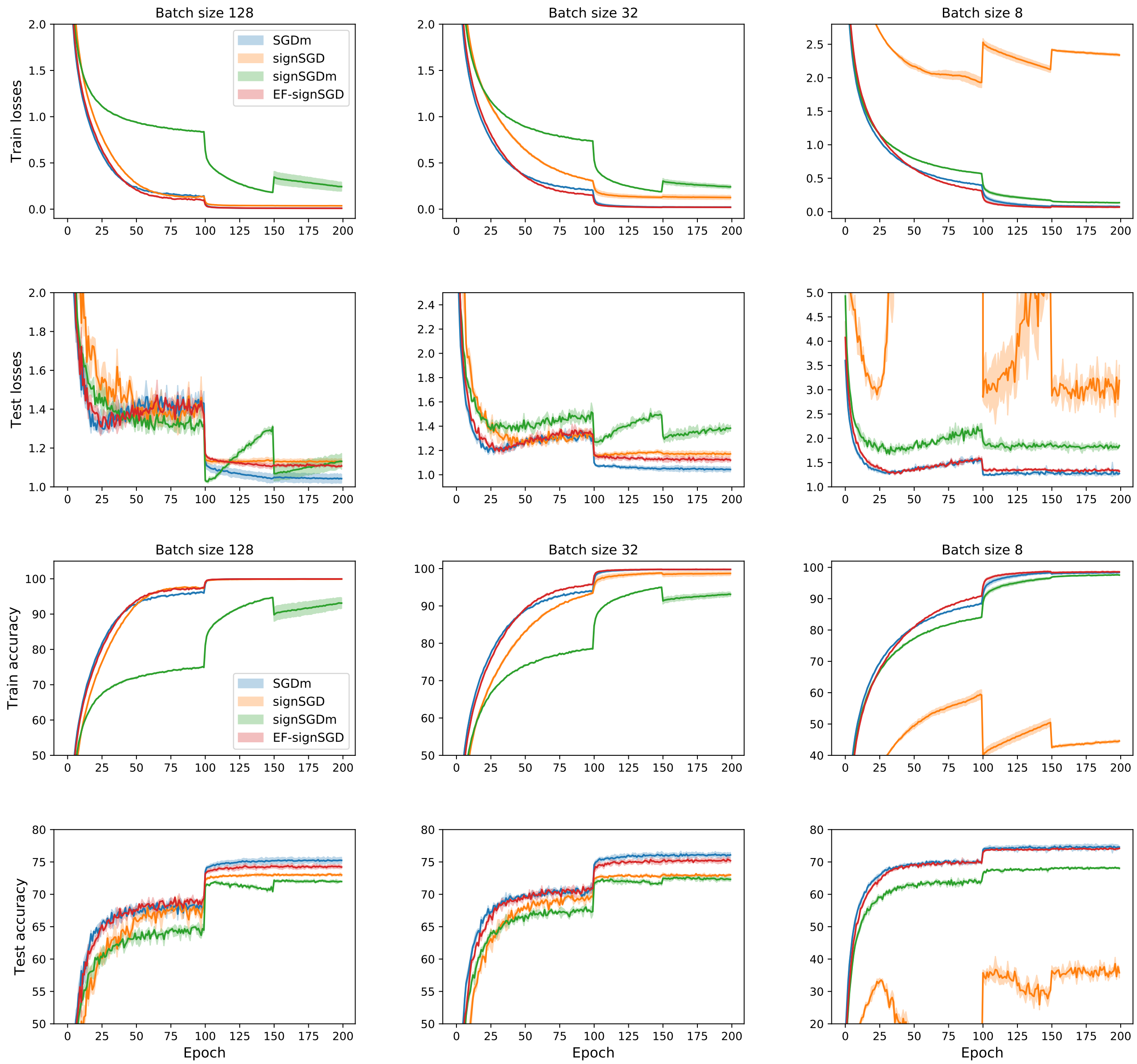

Figure 6: Experimental results showing the loss values and accuracy percentages on the train and test datasets, on CIFAR-100 using Resnet18 for different batch-sizes. The solid curves represent the mean value and shaded region spans one standard deviation obtained over three repetitions. Note that the scale of the y-axis varies across the plots. The losses behave very similar to the accuracies—EF-SIGNSGD consistently and significantly outperforms the other sign-based methods, is faster than SGDM on train, and closely matches SGDM on test.

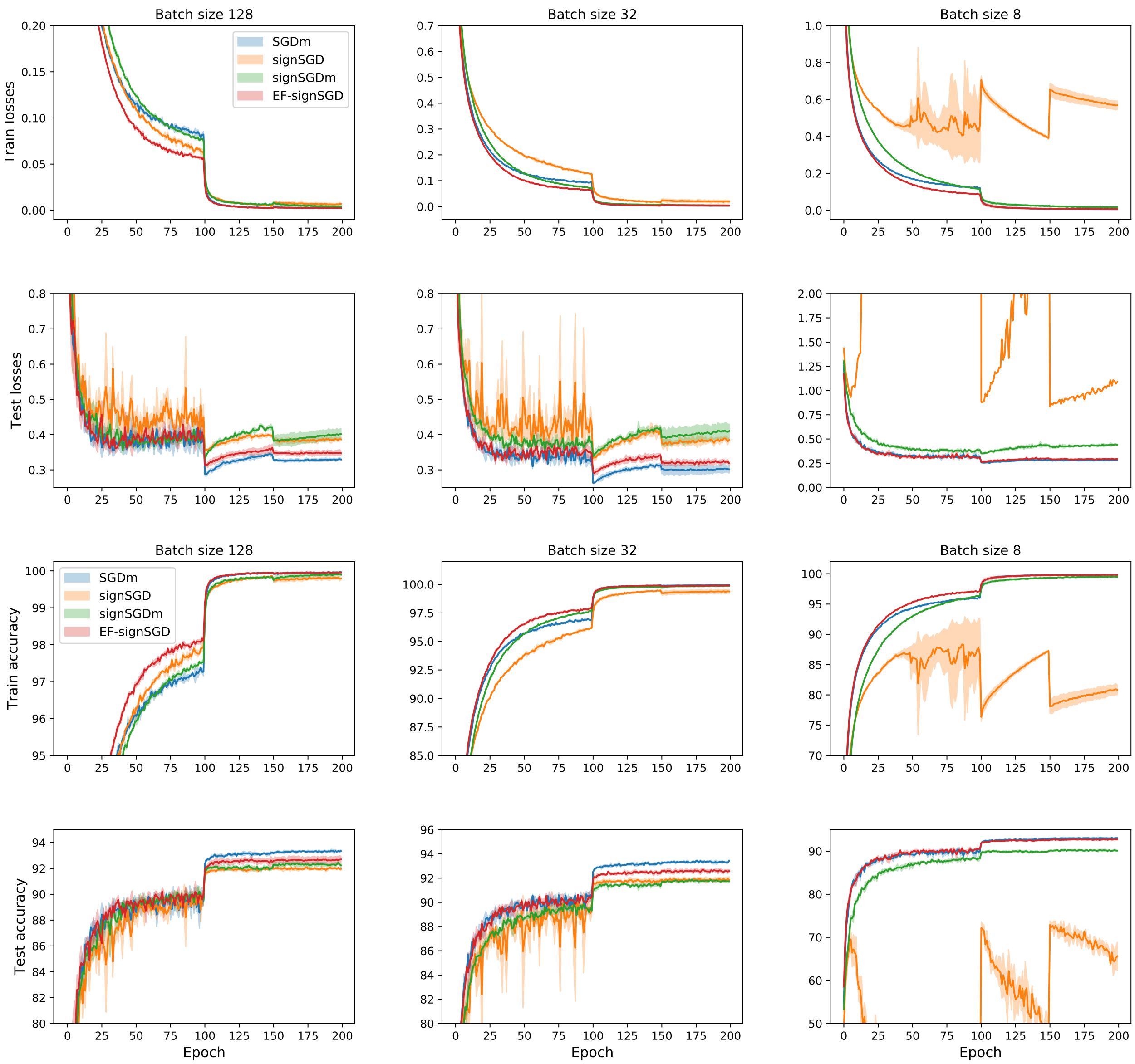

Figure 7: Experimental results showing the loss values and accuracy percentages on the train and test datasets, on CIFAR-10 using VGG19 for different batch-sizes. The solid curves represent the mean value and shaded region spans one standard deviation obtained over three repetitions. Note that the scale of the y-axis varies across the plots. The plots for VGG19 behave very similarly to that of Resnet18, except that SIGNSGDM performs better on the train dataset. On the test dataset, SIGNSGDM and the other algorithms behave exactly as in Resnet. Here too, EF-SIGNSGD consistently and significantly outperforms the other sign-based methods, is faster than SGDM on train, and also closely matches the test performance of SGDM.

A.6. Data generation process (Section 5.2)

The data is generated as in Section 3.3 of (Wilson et al., 2017). We fix $n = 200$ (the number of data points) and $d = 6n$ (dimension) in the below process. Each entry of the target label vector $\mathbf{y} \in \{-1, 1\}^n$ is uniformly set as -1 or 1. Then the jth coordinate (column) of the ith data point (row) in the data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ is filled as follows:

$$A_{i,j} = \begin{cases} y_i & j = 1\,, \\ 1 & j = 2, 3\,, \\ 1 & j = 4 + 5(i1), \dots, 4 + 5(i1) + 2(1y_i)\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Then the data matrix $\mathbf{A}$ and labels $\mathbf{y}$ are randomly (and equally) split between the train and the test dataset. Hence there are 100 data points each of dimension 1200 in the test and train.

B. Missing Proofs

In this section we fill out the proofs of claims, lemmas, and theorems made in the main paper.

B.1. Proof of counter-example (Theorem I)

The stochastic gradient at iteration t is of the form

$$\mathbf{g}_t = \mathbf{a}_{i_t} l'_{i_t}(\langle \mathbf{a}_{i_t}, \mathbf{x} \rangle)\,.$$

This means that

$$\operatorname{sign}(\mathbf{g}_t) = \operatorname{sign}(\mathbf{a}_{i_t}) \cdot \operatorname{sign}(l_{i_t}(\langle \mathbf{a}_{i_t}, \mathbf{x} \rangle)) = \pm \mathbf{s}\,.$$

Thus $\mathbf{x}_{t+1} = \mathbf{x}_t \pm \gamma \mathbf{s}$ and the iterates of SIGNSGD can only move along the direction $\mathbf{s}$. Then, SIGNSGD can converge only if there exists $\gamma^\star \in \mathbb{R}$ such that

$$\mathbf{x}_0 = \mathbf{x}^\star + \gamma^\star \mathbf{s}\,.$$

Since the measure of this set in $\mathbb{R}^d$ for $d \geq 2$ is 0, we can conclude that SIGNSGD will not converge to $\mathbf{x}^\star$ almost surely.

B.2. Proof of bounded error (Lemma 3)

By definition of the error sequence,

$$\|\mathbf{e}_{t+1}\|^2 = \|\mathcal{C}(\mathbf{p}_t) - \mathbf{p}_t\|_2^2 \leq (1-\delta)\|\mathbf{p}_t\|_2^2 = (1-\delta)\|\mathbf{e}_t + \gamma \mathbf{g}_t\|_2^2\,.$$

In the inequality above we used that $\mathcal{C}(\cdot)$ is a δ-approximate compressor. We thus have a recurrence relation on the bound of $\mathbf{e}_t$. Using Young's ineuqality, we have that for any $\eta > 0$:

$$\|\mathbf{e}_{t+1}\|^2 \leq (1-\delta)\|\mathbf{e}_t + \gamma \mathbf{g}_t\|_2^2 \leq (1-\delta)(1+\eta)\|\mathbf{e}_t\|_2^2 + \gamma^2(1-\delta)(1+1/\eta)\|\mathbf{g}_t\|_2^2\,.$$

Here on is simple algebraic computations to solve the recurrence relation above:

$$\begin{aligned} \mathbb{E}\|\mathbf{e}_{t+1}\|^2 &\leq (1-\delta)(1+\eta)\,\mathbb{E}\|\mathbf{e}_t\|_2^2 + \gamma^2(1-\delta)(1+1/\eta)\,\mathbb{E}\|\mathbf{g}_t\|_2^2 \\ &\leq \sum_{i=0}^{t} [(1-\delta)(1+\eta)]^{t-i} \gamma^2 (1-\delta)(1+1/\eta)\,\mathbb{E}\|\mathbf{g}_i\|_2^2 \\ &\leq \sum_{i=0}^{\infty} [(1-\delta)(1+\eta)]^{i} \gamma^2 (1-\delta)(1+1/\eta)\sigma^2 \\ &= \frac{(1-\delta)(1+1/\eta)}{1-(1-\delta)(1+\eta)}\gamma^2\sigma^2 \\ &= \frac{(1-\delta)(1+1/\eta)}{\delta - \eta(1-\delta)}\gamma^2\sigma^2 \\ &= \frac{2(1-\delta)(1+1/\eta)}{\delta}\gamma^2\sigma^2\,. \end{aligned}$$

Let us pick $\eta = \frac{\delta}{2(1-\delta)}$ such that $1 + 1/\eta = (2-\delta)/\delta \leq 2/\delta$. Pluggint this in the above gives

$$\mathbb{E}\|\mathbf{e}_{t+1}\|^2 \leq \frac{2(1-\delta)(1+1/\eta)}{\delta}\gamma^2\sigma^2 \leq \frac{4(1-\delta)}{\delta^2}\gamma^2\sigma^2 . \qquad \square$$

B.3. Proof of non-convex convergence of EF-SIGNSGD (Theorem II)

As outlined in the proof sketch, the analysis considers the actual sequence $\{\mathbf{x}_t\}$ as an approximation to the sequence $\{\tilde{\mathbf{x}}_t\}$, where $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{e}_t$. It satisfies the recurence

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{x}_t + \mathbf{e}_{t+1} - \mathcal{C}(\mathbf{p}_t) = \mathbf{x}_t - \mathbf{p}_t = \tilde{\mathbf{x}}_t - \gamma\mathbf{g}_t .$$

Since the function f is L-smooth,

$$\begin{aligned}
\mathbb{E}_t[f(\tilde{\mathbf{x}}_{t+1})] &\leq f(\tilde{\mathbf{x}}_t) + \langle \nabla f(\tilde{\mathbf{x}}_t), \mathbb{E}_t[\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t]\rangle + \frac{L}{2}\mathbb{E}_t\Big[\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|_2^2\Big] \\
&= f(\tilde{\mathbf{x}}_t) - \gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \mathbb{E}_t[\mathbf{g}_t]\rangle + \frac{L\gamma^2}{2}\mathbb{E}_t\Big[\|\mathbf{g}_t\|_2^2\Big] \\
&\leq f(\tilde{\mathbf{x}}_t) - \gamma\langle \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t)\rangle + \frac{L\gamma^2\sigma^2}{2} .
\end{aligned}$$

In the above we need to get rid of $\nabla f(\tilde{\mathbf{x}}_t)$ since we never encounter it in the algorithm. We can do so using an alternate definition of smoothness of f:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2 .$$

Using the above with $\mathbf{x} = \mathbf{x}_t$ and $\mathbf{y} = \tilde{\mathbf{x}}_t$ we continue as

$$\begin{aligned}
\mathbb{E}_t[f(\tilde{\mathbf{x}}_{t+1})] &\leq f(\tilde{\mathbf{x}}_t) - \gamma\langle \nabla f(\mathbf{x}_t), \nabla f(\mathbf{x}_t)\rangle + \frac{L\gamma^2\sigma^2}{2} + \gamma\langle \nabla f(\mathbf{x}_t) - \nabla f(\tilde{\mathbf{x}}_t), \nabla f(\mathbf{x}_t)\rangle \\
&\leq f(\tilde{\mathbf{x}}_t) - \gamma\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{L\gamma^2\sigma^2}{2} + \frac{\gamma\rho}{2}\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{\gamma}{2\rho}\|\nabla f(\mathbf{x}_t) - \nabla f(\tilde{\mathbf{x}}_t)\|_2^2 \\
&\leq f(\tilde{\mathbf{x}}_t) - \gamma\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{L\gamma^2\sigma^2}{2} + \frac{\gamma\rho}{2}\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{\gamma L^2}{2\rho}\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2^2 \\
&\leq f(\tilde{\mathbf{x}}_t) - \gamma\Big(1 - \frac{\rho}{2}\Big)\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{L\gamma^2\sigma^2}{2} + \frac{\gamma L^2}{2\rho}\|\mathbf{e}_t\|_2^2 .
\end{aligned}$$

In the second inequality follows from the mean-value inequality and holds for any $\rho > 0$. Lemma 3 helps us bound the norm of $\mathbf{e}_t$:

$$\mathbb{E}_t[f(\tilde{\mathbf{x}}_{t+1})] \leq f(\tilde{\mathbf{x}}_t) - \gamma\Big(1 - \frac{\rho}{2}\Big)\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{L\gamma^2\sigma^2}{2} + \frac{\gamma^3 L^2\sigma^2}{2\rho}\frac{4(1-\delta)}{\delta^2} .$$

Rearranging the terms and averaging over t gives for $\rho < 2$

$$\begin{aligned}
\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|_2^2 &\leq \frac{1}{\gamma\big(1 - \frac{\rho}{2}\big)(T+1)}\sum_{t=0}^{T}(\mathbb{E}[f(\tilde{\mathbf{x}}_t)] - \mathbb{E}[f(\tilde{\mathbf{x}}_{t+1})]) + \frac{L\gamma\sigma^2}{2-\rho} + \frac{\gamma^2 L^2\sigma^2}{\rho(2-\rho)}\frac{4(1-\delta)}{\delta^2} \\
&\leq \frac{f(\mathbf{x}_0) - f^\star}{\gamma T(1-\rho/2)} + \frac{L\gamma\sigma^2}{2-\rho} + \frac{4\gamma^2 L^2\sigma^2(1-\delta)}{\rho(2-\rho)\delta^2} . \qquad \square
\end{aligned}$$

Using $\rho = 1$ gives the result as stated in Theorem II. Note that we can choose ρ arbitarily close to 0. By choosing $\rho \overset{T=\infty}{\longrightarrow} 0$, we can show that the asymptotic convergence of EF-SIGNSGD is in fact exactly the same as SGD.

B.4. Proof of non-convex convergence of SGD (Remark 4)

The proof is exactly as above, but with some simplifications since (essentially) $\mathbf{e}_t = 0$. So using the smoothness of f as before and the fact that $\mathbb{E}_t[\mathbf{x}_{t+1}] = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)$ we get that

$$\begin{aligned}
\mathbb{E}_t[f(\mathbf{x}_{t+1})] &\leq f(\tilde{\mathbf{x}}_t) + \langle \nabla f(\mathbf{x}_t), \mathbb{E}_t[\mathbf{x}_{t+1} - \mathbf{x}_t]\rangle + \frac{L}{2}\mathbb{E}_t\Big[\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2\Big] \\
&= f(\mathbf{x}_t) - \gamma\|\nabla f(\mathbf{x}_t)\|^2 + \frac{L\sigma^2\gamma^2}{2} .
\end{aligned}$$

Now rearranging the terms and averaging over T gives Rearranging the terms and averaging over t gives

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{1}{\gamma(T+1)}\sum_{t=0}^{T}(\mathbb{E}[f(\tilde{\mathbf{x}}_t)] - \mathbb{E}[f(\tilde{\mathbf{x}}_{t+1})]) + \frac{L\gamma\sigma^2}{2}$$
$$\leq \frac{f(\mathbf{x}_0) - f^\star}{\gamma T} + \frac{L\gamma\sigma^2}{2}\,. \qquad \square$$

B.5. Proof of compression of unbiased estimators (Remark 5)

Suppose that $\mathbb{E}[\mathcal{U}(\mathbf{v})] = \mathbf{v}$ and that $\mathbb{E}[\|\mathcal{U}(\mathbf{v})\|]^2 \leq k\|\mathbf{v}\|^2$. Then

$$\mathbb{E}\left[\left\|\tfrac{1}{k}\mathcal{U}(\mathbf{v}) - \mathbf{v}\right\|^2\right] = \frac{1}{k^2}\,\mathbb{E}\left[\|\mathcal{U}(\mathbf{v})\|^2\right] - \frac{2}{k}\langle\mathbb{E}[\mathcal{U}(\mathbf{v})], \mathbf{v}\rangle + \|\mathbf{v}\|^2$$
$$\leq \frac{1}{k}\|\mathbf{v}\|^2 - \frac{2}{k}\|\mathbf{v}\|^2 + \|\mathbf{v}\|^2$$
$$= \left(1 - \frac{1}{k}\right)\|\mathbf{v}\|^2\,. \qquad \square$$

B.6. Proof of convex convergence of EF-SIGNSGD (Theorem III)

As in the proof of Theorem II, we start by considering the sequence $\{\tilde{\mathbf{x}}_t\}$ where $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{e}_t$. As we saw, $\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma\mathbf{g}_t$. Suppose that $\mathbf{x}_t^\star$ is an optimum solution. We will abuse notation here and use $\partial f(\mathbf{x})$ to mean any subgradient of f at $\mathbf{x}$.

$$\mathbb{E}_t\left[\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2\right] = \mathbb{E}_t\left[\|\tilde{\mathbf{x}}_t - \gamma\mathbf{g}_t - \mathbf{x}^\star\|^2\right]$$
$$= \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\,\mathbb{E}_t\left[\|\mathbf{g}_t\|^2\right] - \gamma\langle\mathbb{E}_t[\mathbf{g}_t], \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle$$
$$\leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \tilde{\mathbf{x}}_t - \mathbf{x}^\star\rangle\,.$$

We do not want $\tilde{\mathbf{x}}_t$ appearing in the right side of the equation and so we will replace it with $\mathbf{x}_t$ and use Lemma 3 to bound the error:

$$\mathbb{E}_t\left[\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2\right] \leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle + \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \tilde{\mathbf{x}}_t\rangle$$
$$= \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{e}_t\rangle$$
$$\leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle + \gamma\|\partial f(\mathbf{x}_t)\|\|\mathbf{e}_t\|$$
$$\leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle + \frac{2\gamma^2\sigma\sqrt{1-\delta}}{\delta}\|\partial f(\mathbf{x}_t)\|\,.$$

We use Cauchy-Shwarzch in the third step, and Lemma 3 in the last step. We will use the loose bound $\|\partial f(\mathbf{x}_t)\| \leq \sigma$. This is the key difference between the non-smooth case and the smooth (and strongly-convex) case considered in (Stich et al., 2018). In the smooth case, the term $\|\nabla f(\mathbf{x}_t)\|^2$ can be bounded by the error $f(\mathbf{x}_t) - f^\star$ i.e. the final term goes to 0 faster than γ^2, allowing the asymptotic rate to not depend on the compression quality δ. We now have

$$\mathbb{E}_t\left[\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2\right] \leq \|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2 + \gamma^2\sigma^2 - \gamma\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle + \frac{2\gamma^2\sigma^2\sqrt{1-\delta}}{\delta}\,. \tag{1}$$

Recall that $\mathbf{e}_0 = 0$ and so $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$. Rearranging the terms and averaging, we get

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\langle\partial f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle] \leq \frac{1}{\gamma(T+1)}\sum_{t=0}^{T}\left(\mathbb{E}\left[\|\tilde{\mathbf{x}}_t - \mathbf{x}^\star\|^2\right] - \mathbb{E}\left[\|\tilde{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2\right]\right) + \gamma\sigma^2 + \frac{2\gamma\sigma^2\sqrt{1-\delta}}{\delta}$$
$$\leq \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\gamma T} + \gamma\sigma^2\left(1 + \frac{2\sqrt{1-\delta}}{\delta}\right).$$

To finish the proof, we have to simply use the convexity of f twice on the left hand side of the above inequality:

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\langle\partial f(\mathbf{x}_t),\mathbf{x}_t-\mathbf{x}^\star\rangle] \geq \frac{1}{T+1}\sum_{t=0}^{T} f(\mathbf{x}_t) \geq f(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{x}_t) = f(\bar{\mathbf{x}}_T)\,.$$

For the standard rate of SGD in remark 6, we just set $\delta = 0$. □

B.7. Proof relating linear span of gradients to pseudo-inverse (Lemma 9)

Recall that $\mathbf{A} \in \mathbb{R}^{n\times d}$ for $n < d$. Assume without loss of generality that the rows of $\mathbf{A}$ are linearly independent and hence $\mathbf{A}$ is of rank n. The stochastic gradient for $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$ is of the form $\sum \alpha_i \mathbf{A}_{i,:}$ where $\mathbf{A}_{i,:}$ indicates the ith column of $\mathbf{A}$. If $\mathbf{x}_t$ is in the linear span of the stochastic gradients, then there exists a vector $\mathbf{b}_t \in \mathbb{R}^n$ such that

$$\mathbf{x}_t = \mathbf{A}^\top \mathbf{b}_t\,.$$

Suppose $\mathbf{x}^\star$ is the solution reached. Then $\mathbf{A}\mathbf{x}^\star = \mathbf{y}$ and also $\mathbf{x}^\star = \mathbf{A}^\top\mathbf{b}^\star$ for some $\mathbf{b}^\star$. Hence $\mathbf{b}^\star$ must satisfy

$$\mathbf{A}\mathbf{A}^\top\mathbf{b}^\star = \mathbf{y}\,.$$

Since the rank of $\mathbf{A}$ is n, the matrix $\mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{n\times n}$ is full-rank and invertible. This means that there exists an unique solution to $\mathbf{b}^\star$ and $\mathbf{x}^\star$:

$$\mathbf{b}^\star = \left(\mathbf{A}\mathbf{A}^\top\right)^{-1}\mathbf{y} \quad \text{and} \quad \mathbf{x}^\star = \mathbf{A}^\top\mathbf{b}^\star = \mathbf{A}^\top\left(\mathbf{A}\mathbf{A}^\top\right)^{-1}\mathbf{y}\,.$$ □