

SIGNSGD: Compressed Optimisation for Non-Convex Problems

Jeremy Bernstein^{1,2} Yu-Xiang Wang^{2,3} Kamyar Azizzadenesheli⁴ Anima Anandkumar^{1,2}

Abstract

Training large neural networks requires distributing learning across multiple workers, where the **cost of communicating gradients can be a significant bottleneck**. SIGNSGD alleviates this problem by transmitting just the sign of each minibatch stochastic gradient. We prove that it can get the best of both worlds: compressed gradients *and* SGD-level convergence rate. The relative ℓ_1/ℓ_2 geometry of gradients, noise and curvature informs whether SIGNSGD or SGD is theoretically better suited to a particular problem. On the practical side we find that the momentum counterpart of SIGNSGD is able to match the accuracy and convergence speed of ADAM on deep Imagenet models. We extend our theory to the distributed setting, where the parameter server uses majority vote to aggregate gradient signs from each worker enabling 1-bit compression of worker-server communication *in both directions*. Using a theorem by Gauss (1823) we prove that majority vote can achieve the same reduction in variance as full precision distributed SGD. Thus, there is great promise for sign-based optimisation schemes to achieve fast communication *and* fast convergence. Code to reproduce experiments is to be found at <https://github.com/jxbz/signSGD>.

1. Introduction

Deep neural networks have learnt to solve numerous natural human tasks (LeCun et al., 2015; Schmidhuber, 2015). Training these large-scale models can take days or even weeks. The learning process can be accelerated by distributing training over multiple processors—either GPUs linked within a single machine, or even multiple machines linked together. Communication between workers is typically handled using a parameter-server framework (Li et al.,

¹Caltech ²Amazon AI ³UC Santa Barbara ⁴UC Irvine. Correspondence to: Jeremy Bernstein <bernstein@caltech.edu>, Yu-Xiang Wang <yuxiangw@amazon.edu>.

Algorithm 1 SIGNSGD

Input: learning rate δ , current point x_k
 $\tilde{g}_k \leftarrow \text{stochasticGradient}(x_k)$
 $x_{k+1} \leftarrow x_k - \delta \text{sign}(\tilde{g}_k)$

Algorithm 2 SIGNUM

Input: learning rate δ , momentum constant $\beta \in (0, 1)$, current point x_k , current momentum m_k
 $\tilde{g}_k \leftarrow \text{stochasticGradient}(x_k)$
 $m_{k+1} \leftarrow \beta m_k + (1 - \beta) \tilde{g}_k$
 $x_{k+1} \leftarrow x_k - \delta \text{sign}(m_{k+1})$

2014), which involves repeatedly communicating the gradients of every parameter in the model. This can still be time-intensive for large-scale neural networks. The communication cost can be reduced if gradients are compressed before being transmitted. In this paper, we analyse the theory of robust schemes for gradient compression.

An elegant form of gradient compression is just to take the sign of each coordinate of the stochastic gradient vector, which we call SIGNSGD. The algorithm is as simple as throwing away the exponent and mantissa of a 32-bit floating point number. Sign-based methods have been studied at least since the days of RPROP (Riedmiller & Braun, 1993). This algorithm inspired many popular optimisers—like RMSPROP (Tieleman & Hinton, 2012) and ADAM (Kingma & Ba, 2015). But researchers were interested in RPROP and variants because of their robust and fast convergence, and not their potential for gradient compression.

Until now there has been no rigorous theoretical explanation for the empirical success of sign-based stochastic gradient

Algorithm 3 Distributed training by majority vote

Input: learning rate δ , current point x_k , # workers M each with an independent gradient estimate $\tilde{g}_m(x_k)$

on server

pull $\text{sign}(\tilde{g}_m)$ **from** each worker

push $\text{sign}\left[\sum_{m=1}^M \text{sign}(\tilde{g}_m)\right]$ **to** each worker

on each worker

$x_{k+1} \leftarrow x_k - \delta \text{sign}\left[\sum_{m=1}^M \text{sign}(\tilde{g}_m)\right]$

methods. The sign of the stochastic gradient is a biased approximation to the true gradient, making it more challenging to analyse compared to standard SGD. In this paper, we provide extensive theoretical analysis of sign-based methods for non-convex optimisation under transparent assumptions. We show that SIGNSGD is especially efficient in problems with a particular ℓ_1 geometry: when gradients are as dense or denser than stochasticity and curvature, then SIGNSGD can converge with a theoretical rate that has similar or even better dimension dependence than SGD. We find empirically that both gradients and noise are dense in deep learning problems, consistent with the observation that SIGNSGD converges at a similar rate to SGD in practice.

We then analyse SIGNSGD in the distributed setting where the parameter server aggregates gradient signs of the workers by a majority vote. Thus we allow worker-server communication to be 1-bit compressed in both directions. We prove that the theoretical speedup matches that of distributed SGD, under natural assumptions that are validated by experiments.

We also extend our theoretical framework to the SIGNUM optimiser—which takes the sign of the momentum. Our theory suggests that momentum may be useful for controlling a tradeoff between bias and variance in the estimate of the stochastic gradient. On the practical side, we show that SIGNUM easily scales to large Imagenet models, and provided the learning rate and weight decay are tuned, all other hyperparameter settings—such as momentum, weight initialiser, learning rate schedules and data augmentation—may be lifted from an SGD implementation.

2. Related Work

Distributed machine learning: From the information theoretic angle, Suresh et al. (2017) study the communication limits of estimating the mean of a general quantity known about only through samples collected from M workers. In contrast, we focus exclusively on communication of gradients for optimisation, which allows us to exploit the fact that we do not care about incorrectly communicating small gradients in our theory. Still our work has connections with information theory. When the parameter server aggregates gradients by majority vote, it is effectively performing maximum likelihood decoding of a repetition encoding of the true gradient sign that is supplied by the M workers.

As for existing gradient compression schemes, Seide et al. (2014) and Strom (2015) demonstrated empirically that 1-bit quantisation can still give good performance whilst dramatically reducing gradient communication costs in distributed systems. Alistarh et al. (2017) and Wen et al. (2017) provide schemes with theoretical guarantees by using random number generators to ensure that the compressed gradient is

Table 1. The communication cost of different gradient compression schemes, when training a d -dimensional model with M workers.

ALGORITHM	# BITS PER ITERATION
SGD (Robbins & Monroe, 1951)	$64Md$
QSGD (Alistarh et al., 2017)	$(2 + \log(2M + 1))Md$
TERNGRAD (Wen et al., 2017)	$(2 + \log(2M + 1))Md$
SIGNSGD with majority vote	$2Md$

still an unbiased approximation to the true gradient. Whilst unbiasedness allows these schemes to bootstrap SGD theory, it unfortunately comes at the cost of hugely inflated variance, and this variance explosion¹ basically renders the SGD-style bounds vacuous in the face of the empirical success of these algorithms. The situation only gets worse when the parameter server must aggregate and send back the received gradients, since merely summing up quantised updates reduces the quantisation efficiency. We compare the schemes in Table 1—notice how the existing schemes pick up log factors in the transmission from parameter-server back to workers. Our proposed approach is different, in that we directly employ the sign gradient which is *biased*. This avoids the randomisation needed for constructing an unbiased quantised estimate, avoids the problem of variance exploding in the theoretical bounds, and even enables 1-bit compression in both directions between parameter-server and workers, at no theoretical loss compared to distributed SGD.

Deep learning: stochastic gradient descent (Robbins & Monroe, 1951) is a simple and extremely effective optimiser for training neural networks. Still Riedmiller & Braun (1993) noted the good practical performance of sign-based methods like RPROP for training deep nets, and since then variants such as RMSPROP (Tieleman & Hinton, 2012) and ADAM (Kingma & Ba, 2015) have become increasingly popular. ADAM updates the weights according to the mean divided by the root mean square of recent gradients. Let $\langle \cdot \rangle_\beta$ denote an exponential moving average with timescale β , and \tilde{g} the stochastic gradient. Then

$$\text{ADAM step} \sim \frac{\langle \tilde{g} \rangle_{\beta_1}}{\sqrt{\langle \tilde{g}^2 \rangle_{\beta_2}}}$$

Therefore taking the time scale of the exponential moving averages to zero, $\beta_1, \beta_2 \rightarrow 0$, yields SIGNSGD

$$\text{SIGNSGD step} = \text{sign}(\tilde{g}) = \frac{\tilde{g}}{\sqrt{\tilde{g}^2}}.$$

To date there has been no convincing theory of the {RPROP, RMSPROP, ADAM} family of algorithms, known as ‘adaptive gradient methods’. Indeed Reddi et al. (2018) point out

¹For the version of QSGD with 1-bit compression, the variance explosion is by a factor of \sqrt{d} , where d is the number of weights. It is common to have $d > 10^8$ in modern deep networks.

problems in the original convergence proof of ADAM, even in the convex setting. Since SIGNSGD belongs to this same family of algorithms, we expect that our theoretical analysis should be relevant for all algorithms in the family. In a parallel work, Balles & Hennig (2017) explore the connection between SIGNSGD and ADAM in greater detail, though their theory is more restricted and lives in the convex world, and they do not analyse SIGNUM as we do but employ it on heuristic grounds.

Optimisation: much of classic optimisation theory focuses on convex problems, where *local information* in the gradient tells you *global information* about the direction to the minimum. Whilst elegant, this theory is less relevant for modern problems in deep learning which are non-convex. In non-convex optimisation, finding the global minimum is generally intractable. Theorists usually settle for measuring some restricted notion of success, such as rate of convergence to stationary points (Ghadimi & Lan, 2013; Allen-Zhu, 2017a) or local minima (Nesterov & Polyak, 2006). Though Dauphin et al. (2014) suggest saddle points should abound in neural network error landscapes, practitioners report not finding this a problem in practice (Goodfellow et al., 2015) and therefore a theory of convergence to stationary points is useful and informative.

On the algorithmic level, the *non-stochastic* version of SIGNSGD can be viewed as the classical steepest descent algorithm with ℓ_∞ -norm (see, e.g., Boyd & Vandenberghe, 2004, Section 9.4). The convergence of steepest descent is well-known (see Karimi et al., 2016, Appendix C, for an analysis of signed gradient updates under the Polyak-Łojasiewicz condition). Carlson et al. (2016) study a stochastic version of the algorithm, but again under an ℓ_∞ majorisation. To the best of our knowledge, we are the first to study the convergence of signed gradient updates under an (often more natural) ℓ_2 majorisation (Assumption 2).

Experimental benchmarks: throughout the paper we will make use of the CIFAR-10 (Krizhevsky, 2009) and Imagenet (Russakovsky et al., 2015) datasets. As for neural network architectures, we train Resnet-20 (He et al., 2016a) on CIFAR-10, and Resnet-50 v2 (He et al., 2016b) on Imagenet.

3. Convergence Analysis of SIGNSGD

We begin our analysis of sign stochastic gradient descent in the non-convex setting. The standard assumptions of the stochastic optimisation literature are nicely summarised by Allen-Zhu (2017b). We will use more fine-grained assumptions. SIGNSGD can exploit this additional structure, much as ADAGRAD (Duchi et al., 2011) exploits sparsity. We emphasise that these fine-grained assumptions do not lose anything over typical SGD assumptions, since *our as-*

sumptions can be obtained from SGD assumptions and vice versa.

Assumption 1 (Lower bound). *For all x and some constant f^* , we have objective value $f(x) \geq f^*$.*

This assumption is standard and necessary for guaranteed convergence to a stationary point.

The next two assumptions will naturally encode notions of heterogeneous curvature and gradient noise.

Assumption 2 (Smooth). *Let $g(x)$ denote the gradient of the objective $f(\cdot)$ evaluated at point x . Then $\forall x, y$ we require that for some non-negative constant $\vec{L} := [L_1, \dots, L_d]$*

$$\left| f(y) - [f(x) + g(x)^T(y - x)] \right| \leq \frac{1}{2} \sum_i L_i (y_i - x_i)^2.$$

For twice differentiable f , this implies that $-\text{diag}(\vec{L}) \prec H \prec \text{diag}(\vec{L})$. This is related to the slightly weaker coordinate-wise Lipschitz condition used in the block coordinate descent literature (Richtárik & Takáč, 2014).

Lastly, we assume that we have access to the following stochastic gradient oracle:

Assumption 3 (Variance bound). *Upon receiving query $x \in \mathbb{R}^d$, the stochastic gradient oracle gives us an independent unbiased estimate \tilde{g} that has coordinate bounded variance:*

$$\mathbb{E}[\tilde{g}(x)] = g(x), \quad \mathbb{E}[(\tilde{g}(x)_i - g(x)_i)^2] \leq \sigma_i^2$$

for a vector of non-negative constants $\vec{\sigma} := [\sigma_1, \dots, \sigma_d]$.

Bounded variance may be unrealistic in practice, since as $x \rightarrow \infty$ the variance might well diverge. Still this assumption is useful for understanding key properties of stochastic optimisation algorithms. In our theorem, we will be working with a mini-batch of size n_k in the k^{th} iteration, and the corresponding mini-batch stochastic gradient is modeled as the average of n_k calls to the above oracle at x_k . This squashes the variance bound on $\tilde{g}(x)_i$ to σ_i^2/n_k .

Assumptions 2 and 3 are different from the assumptions typically used for analysing the convergence properties of SGD (Nesterov, 2013; Ghadimi & Lan, 2013), but they are natural to the geometry induced by algorithms with signed updates such as SIGNSGD and SIGNUM.

Assumption 2 is more fine-grained than the standard assumption, which is recovered by defining ℓ_2 Lipschitz constant $L := \|\vec{L}\|_\infty = \max_i L_i$. Then Assumption 2 implies that

$$\left| f(y) - [f(x) + g(x)^T(y - x)] \right| \leq \frac{L}{2} \|y - x\|_2^2.$$

which is the standard assumption of Lipschitz smoothness.

Assumption 3 is more fined-grained than the standard stochastic gradient oracle assumption used for SGD analysis. But again, the standard variance bound is recovered by defining $\sigma^2 := \|\bar{\sigma}\|_2^2$. Then Assumption 3 implies that

$$\mathbb{E}\|\tilde{g}(x) - g(x)\|^2 \leq \sigma^2$$

which is the standard assumption of bounded total variance.

Under these assumptions, we have the following result:

Theorem 1 (Non-convex convergence rate of SIGNSGD). *Run algorithm 1 for K iterations under Assumptions 1 to 3. Set the learning rate and mini-batch size (independently of step k) as*

$$\delta_k = \frac{1}{\sqrt{\|\bar{L}\|_1 K}}, \quad n_k = K$$

Let N be the cumulative number of stochastic gradient calls up to step K , i.e. $N = O(K^2)$. Then we have

$$\begin{aligned} \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right]^2 \\ \leq \frac{1}{\sqrt{N}} \left[\sqrt{\|\bar{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + 2\|\bar{\sigma}\|_1 \right]^2 \end{aligned}$$

The proof is given in Section B of the supplementary material. It follows the well known strategy of relating the norm of the gradient to the expected improvement made in a single algorithmic step, and comparing this with the total possible improvement under Assumption 1. A key technical challenge we overcome is in showing how to directly deal with a biased approximation to the true gradient. Here we will provide some intuition about the proof.

To pass the stochasticity through the non-linear sign operation in a controlled fashion, we need to prove the key statement that at the k^{th} step for the i^{th} gradient component

$$\mathbb{P}[\text{sign}(\tilde{g}_{k,i}) \neq \text{sign}(g_{k,i})] \leq \frac{\sigma_{k,i}}{|g_{k,i}|}$$

This formalises the intuition that the probability of the sign of a component of the stochastic gradient being incorrect should be controlled by the signal-to-noise ratio of that component. When a component's gradient is large, the probability of making a mistake is low, and one expects to make good progress. When the gradient is small compared to the noise, the probability of making mistakes can be high, but due to the large batch size this only happens when we are already close to a stationary point.

The large batch size in the theorem may seem unusual, but large batch training actually presents a systems advantage (Goyal et al., 2017) since it can be parallelised. The number

of gradient calls N is the important quantity to measure convergence, but large batch training achieves N gradient calls in only $O(\sqrt{N})$ iterations whereas small batch training needs $O(N)$ iterations. Fewer iterations also means fewer rounds of communication in the distributed setting. Convergence guarantees *can* be extended to the small batch case under the additional assumption of unimodal symmetric gradient noise using Lemma D.1 in the supplementary, but we leave this for future work. Experiments in this paper were indeed conducted in the small batch regime.

Another unusual feature requiring discussion is the ℓ_1 geometry of SIGNSGD. The convergence rate strikingly depends on the ℓ_1 -norm of the gradient, the stochasticity and the curvature. To understand this better, let's define a notion of density of a high-dimensional vector $\vec{v} \in \mathbb{R}^d$ as follows:

$$\phi(\vec{v}) := \frac{\|\vec{v}\|_1^2}{d\|\vec{v}\|_2^2} \quad (1)$$

To see that this is a natural definition of density, notice that for a fully dense vector, $\phi(\vec{v}) = 1$ and for a fully sparse vector, $\phi(\vec{v}) = 1/d \approx 0$. We trivially have that $\|\vec{v}\|_1^2 \leq \phi(\vec{v})d^2\|\vec{v}\|_\infty^2$ so this notion of density provides an easy way to translate from norms in ℓ_1 to both ℓ_2 and ℓ_∞ .

Remember that under our assumptions, SGD-style assumptions hold with Lipschitz constant $L := \|\bar{L}\|_\infty$ and total variance bound $\sigma^2 := \|\bar{\sigma}\|_2^2$. Using our notion of density we can translate our constants into the language of SGD:

$$\begin{aligned} \|g_k\|_1^2 &= \phi(g_k)d\|g_k\|_2^2 && \geq \phi(g)d\|g_k\|_2^2 \\ \|\bar{L}\|_1^2 &\leq \phi(\bar{L})d^2\|\bar{L}\|_\infty^2 && = \phi(\bar{L})d^2L^2 \\ \|\bar{\sigma}\|_1^2 &= \phi(\bar{\sigma})d\|\bar{\sigma}\|_2^2 && = \phi(\bar{\sigma})d\sigma^2 \end{aligned}$$

where we have assumed $\phi(g)$ to be a lower bound on the gradient density over the entire space. Using that $(x+y)^2 \leq 2(x^2 + y^2)$ and changing variables in the bound, we reach the following result for SIGNSGD

$$\begin{aligned} \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_2 \right]^2 \\ \leq \frac{2}{\sqrt{N}} \left[\frac{\sqrt{\phi(\bar{L})}}{\phi(g)} L \left(f_0 - f_* + \frac{1}{2} \right)^2 + 4 \frac{\phi(\bar{\sigma})}{\phi(g)} \sigma^2 \right] \end{aligned}$$

whereas, for comparison, a typical SGD bound (proved in Supplementary C) is

$$\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_2^2 \right] \leq \frac{1}{\sqrt{N}} [2L(f_0 - f_*) + \sigma^2].$$

The bounds are very similar, except for most notably the appearance of ratios of densities R_1 and R_2 , defined as

$$R_1 := \frac{\sqrt{\phi(\bar{L})}}{\phi(g)} \quad R_2 := \frac{\phi(\bar{\sigma})}{\phi(g)}$$

Naïvely comparing the bounds suggests breaking into cases:

- (I) $R_1 \gg 1$ and $R_2 \gg 1$. This means that both the curvature and the stochasticity are much denser than the typical gradient and the comparison suggests SGD is better suited than SIGNSGD.
- (II) $\text{NOT}[R_1 \gg 1]$ and $\text{NOT}[R_2 \gg 1]$. This means that neither curvature nor stochasticity are much denser than the gradient, and the comparison suggests that SIGNSGD may converge as fast or faster than SGD, and also get the benefits of gradient compression.
- (III) neither of the above holds, for example $R_1 \ll 1$ and $R_2 \gg 1$. Then the comparison is indeterminate about whether SIGNSGD or SGD is more suitable.

Let’s briefly provide some intuition to understand how it’s possible that SIGNSGD could outperform SGD. Imagine a scenario where the gradients are dense but there is a sparse set of extremely noisy components. Then the dynamics of SGD will be dominated by this noise, and (unless the learning rate is reduced a lot) SGD will effectively perform a random walk along these noisy components, paying less attention to the gradient signal. SIGNSGD however will treat all components equally, so it will scale down the sparse noise and scale up the dense gradients comparatively, and thus make good progress. See Figure A.1 in the supplementary for a simple example of this.

Still we must be careful when comparing upper bounds, and interpreting the dependence on curvature density is more subtle than noise density. This is because the SGD bound proved in Supplementary C is slacker under situations of sparse curvature than dense curvature. That is to say that SGD, like SIGNSGD, benefits under situations of sparse curvature but this is not reflected in the SGD bound. The potentially slack step in SGD’s analysis is in switching from L_i to $\|\tilde{L}\|_\infty$. Because of this it is safer to interpret the curvature comparison as telling us a regime where SIGNSGD is expected to lose out to SGD (rather than vice versa). This happens when $R_1 \gg 1$ and gradients are sparser than curvature. Intuitively, in this case SIGNSGD will push many components in highly curved directions even though these components had small gradient, and this can be undesirable.

To summarise, our theory suggests that when gradients are dense, SIGNSGD should be more robust to large stochasticity on a sparse set of coordinates. When gradients are sparse, SGD should be more robust to dense curvature and noise. In practice for deep networks, we find that SIGNSGD converges about as fast as SGD. That would suggest that we are either in regime (II) or (III) above. But what is the real situation for the error landscape of deep neural networks?

To measure gradient and noise densities in practice, we use Welford’s algorithm (Welford, 1962; Knuth, 1997) to

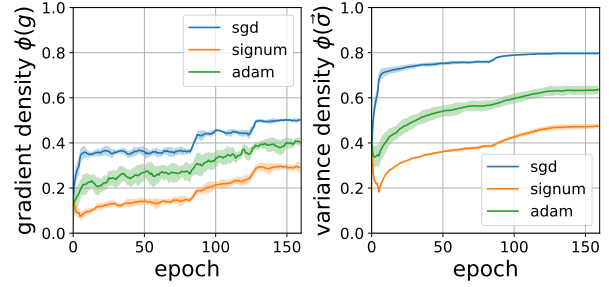


Figure 1. Gradient and noise density during an entire training run of a Resnet-20 model on the CIFAR-10 dataset. Results are averaged over 3 repeats for each of 3 different training algorithms, and corresponding error bars are plotted. At the beginning of every epoch, at that fixed point in parameter space, we do a full pass over the data to compute the exact mean of the stochastic gradient, g , and its exact standard deviation vector $\vec{\sigma}$ (square root of diagonal of covariance matrix). The density measure $\phi(\vec{v}) := \frac{\|\vec{v}\|_2^2}{d\|\vec{v}\|_2^2}$ is 1 for a fully dense vector and ≈ 0 for a fully sparse vector. Notice that both gradient and noise are dense, and moreover the densities appear to be coupled during training. Noticeable jumps occur at epoch 80 and 120 when the learning rate is decimated. Our stochastic gradient oracle (Assumption 3) is fine-grained enough to encode such dense geometries of noise.

compute the true gradient g and its stochasticity vector $\vec{\sigma}$ at every epoch of training for a Resnet-20 model on CIFAR-10. Welford’s algorithm is numerically stable and only takes a single pass through the data to compute the vectorial mean and variance. Therefore if we train a network for 160 epochs, we make an additional 160 passes through the data to evaluate these gradient statistics. Results are plotted in Figure 1. Notice that the gradient density and noise density are of the same order throughout training, and this indeed puts us in regime (II) or (III) as predicted by our theory.

In Figure A.2 of the supplementary, we present preliminary evidence that this finding generalises, by showing that gradients are dense across a range of datasets and network architectures. We have not devised an efficient means to measure curvature densities, which we leave for future work.

4. Majority Rule: the Power of Democracy in the Multi-Worker Setting

In the most common form of distributed training, workers (such as GPUs) each evaluate gradients on their own split of the data, and send the results up to a parameter-server. The parameter server aggregates the results and transmits them back to each worker (Li et al., 2014).

Up until this point in the paper, we have only analysed

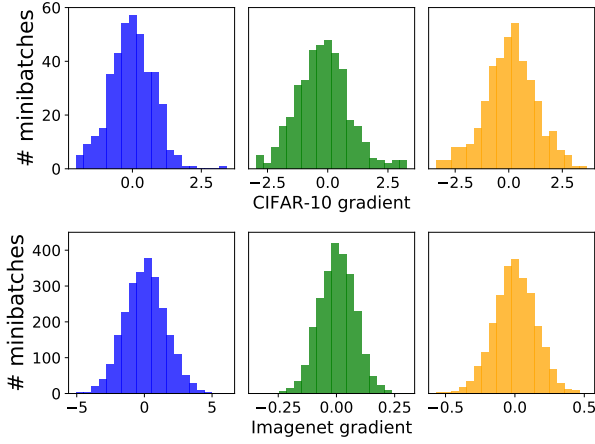


Figure 2. Histograms of the noise in the stochastic gradient, each plot for a different randomly chosen parameter (not cherry-picked). Top row: Resnet-20 architecture trained to epoch 50 on CIFAR-10 with a batch size of 128. Bottom row: Resnet-50 architecture trained to epoch 50 on Imagenet with a batch size of 256. From left to right: model trained with SGD, SIGNUM, ADAM. All noise distributions appear to be unimodal and approximately symmetric. For a batch size of 256 Imagenet images, the central limit theorem has visibly kicked in and the distributions look Gaussian.

SIGNSGD where the update is of the form

$$x_{k+1} = x_k - \delta \text{sign}(\tilde{g})$$

To get the benefits of compression we want the m^{th} worker to send the sign of the gradient evaluated only on its portion of the data. This suggests an update of the form

$$x_{k+1} = x_k - \delta \sum_{m=1}^M \text{sign}(\tilde{g}_m) \quad (\text{good})$$

This scheme is good since what gets sent to the parameter will be 1-bit compressed. But what gets sent back almost certainly will not. Could we hope for a scheme where all communication is 1-bit compressed?

What about the following scheme:

$$x_{k+1} = x_k - \delta \text{sign} \left[\sum_{m=1}^M \text{sign}(\tilde{g}_m) \right] \quad (\text{best})$$

This is called majority vote, since each worker is essentially voting with its belief about the sign of the true gradient. The parameter server counts the votes, and sends its 1-bit decision back to every worker.

The machinery of Theorem 1 is enough to establish convergence for the (good) scheme, but majority vote is more

elegant and more communication efficient, therefore we focus on this scheme from here on.

In Theorem 2 we first establish the general convergence rate of majority vote, followed by a regime where majority vote enjoys a variance reduction from $\|\tilde{\sigma}\|_1$ to $\|\tilde{\sigma}\|_1/\sqrt{M}$.

Theorem 2 (Non-convex convergence rate of distributed SIGNSGD with majority vote). *Run algorithm 3 for K iterations under Assumptions 1 to 3. Set the learning rate and mini-batch size for each worker (independently of step k) as*

$$\delta_k = \frac{1}{\sqrt{\|\tilde{L}\|_1 K}} \quad n_k = K$$

Then (a) majority vote with M workers converges at least as fast as SIGNSGD in Theorem 1.

And (b) further assuming that the noise in each component of the stochastic gradient is unimodal and symmetric about the mean (e.g. Gaussian), majority vote converges at improved rate:

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right]^2 \\ & \leq \frac{1}{\sqrt{N}} \left[\sqrt{\|\tilde{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + \frac{2}{\sqrt{M}} \|\tilde{\sigma}\|_1 \right]^2 \end{aligned}$$

where N is the cumulative number of stochastic gradient calls per worker up to step K .

The proof is given in the supplementary material, but here we sketch some details. Consider the signal-to-noise ratio of a single component of the stochastic gradient, defined as $S := \frac{|g_i|}{\sigma_i}$. For $S < 1$ the gradient is small and it doesn't matter if we get the sign wrong. For $S > 1$, we can show using a one-sided version of Chebyshev's inequality (Cantelli, 1928) that the failure probability, q , of that sign bit on an individual worker satisfies $q < \frac{1}{2}$. This means that the parameter server is essentially receiving a repetition code R_M and the majority vote decoder is known to drive down the failure probability of a repetition code exponentially in the number of repeats (MacKay, 2002).

Remark: Part (a) of the theorem does not describe a speedup over just using a single machine, and that might hint that all those extra $M - 1$ workers are a waste in this setting. **This is not the case.** From the proof sketch above, it should be clear that part (a) is an extremely conservative statement. In particular, we expect all regions of training where the signal-to-noise ratio of the stochastic gradient satisfies $S > 1$ to enjoy a significant speedup due to variance reduction. It's just that since we don't get the speedup when

$S < 1$, it's hard to express this in a compact bound.

To sketch a proof for part (b), note that a sign bit from each worker is a Bernoulli trial—call its failure probability q . We can get a tight control of q by a convenient tail bound owing to Gauss (1823) that holds under conditions of unimodal symmetry. Then the sum of bits received by the parameter server is a binomial random variable, and we can use Cantelli's inequality to bound its tail. This turns out to be enough to get tight enough control on the error probability of the majority vote decoder to prove the theorem.

Remark 1: assuming that the stochastic gradient of each worker is approximately symmetric and unimodal is very reasonable. In particular for increasing mini-batch size it will be an ever-better approximation by the central limit theorem. Figure 2 plots histograms of real stochastic gradients for neural networks. Even at batch-size 256 the stochastic gradient for an Imagenet model already looks Gaussian.

Remark 2: if you delve into the proof of Theorem 2 and graph all of the inequalities, you will notice that some of them are uniformly slack. This suggests that the assumptions of symmetry and unimodality can actually be relaxed to only hold approximately. This raises the possibility of proving a relaxed form of Gauss' inequality and using a third moment bound in the Berry-Esseen theorem to derive a minimal batch size for which the majority vote scheme is guaranteed to work by the central limit theorem. We leave this for future work.

Remark 3: why does this theorem have anything to do with unimodality or symmetry at all? It's because there exist very skewed or bimodal random variables X with mean μ such that $\mathbb{P}[\text{sign}(X) = \text{sign}(\mu)]$ is arbitrarily small. This can either be seen by applying Cantelli's inequality which is known to be tight, or by playing with distributions like

$$\mathbb{P}[X = x] = \begin{cases} 0.1 & \text{if } x = 50 \\ 0.9 & \text{if } x = -1 \end{cases}$$

Distributions like these are a problem because it means that adding more workers will actually drive up the error probability rather than driving it down. The beauty of the central limit theorem is that even for such a skewed and bimodal distribution, the mean of just a few tens of samples will already start to look Gaussian.

5. Extending the Theory to SIGNUM

Momentum is a popular trick used by neural network practitioners that can, in our experience, speed up the training of deep neural networks and improve the robustness of algorithms to other hyperparameter settings. Instead of taking steps according to the gradient, momentum algorithms take steps according to a running average of recent gradients.

Existing theoretical analyses of momentum often rely on the absence of gradient stochasticity (e.g. Jin et al. (2017)) or convexity (e.g. Goh (2017)) to show that momentum's asymptotic convergence rate can beat gradient descent.

It is easy to incorporate momentum into SIGNSGD, merely by taking the sign of the momentum. We call the resulting algorithm SIGNUM and present the algorithmic step formally in Algorithm 2. SIGNUM fits into our theoretical framework, and we prove its convergence rate in Theorem 3.

Theorem 3 (Convergence rate of SIGNUM). *In Algorithm 2, set the learning rate, mini-batch size and momentum parameter respectively as*

$$\delta_k = \frac{\delta}{\sqrt{k+1}} \quad n_k = k+1 \quad \beta$$

Our analysis also requires a warmup period to let the bias in the momentum settle down. The warmup should last for $C(\beta)$ iterations, where C is a constant that depends on the momentum parameter β as follows:

$$C(\beta) = \min_{C \in \mathbb{Z}^+} C \quad \text{s.t.} \quad \frac{C}{2} \beta^C \leq \frac{1}{1 - \beta^2} \frac{1}{C+1} \\ \& \quad \beta^{C+1} \leq \frac{1}{2}$$

Note that for $\beta = 0.9$, we have $C = 54$ which is negligible. For the first $C(\beta)$ iterations, accumulate the momentum as normal, but use the sign of the stochastic gradient to make updates instead of the sign of the momentum.

Let N be the cumulative number of stochastic gradient calls up to step K , i.e. $N = O(K^2)$. Then for $K \gg C$ we have

$$\mathbb{E} \left[\frac{1}{K-C} \sum_{k=C}^{K-1} \|g_k\|_1 \right]^2 = O \left(\frac{1}{\sqrt{N}} \left[\frac{f_C - f_*}{\delta} \right. \right. \\ \left. \left. + (1 + \log N) \left(\frac{\delta \|\tilde{L}\|_1}{1 - \beta} + \|\tilde{\sigma}\|_1 \sqrt{1 - \beta} \right) \right]^2 \right)$$

where we have used $O(\cdot)$ to hide numerical constants and the β -dependent constant C .

The proof is the greatest technical challenge of the paper, and is given in the supplementary material. We focus on presenting the proof in a modular form, anticipating that parts may be useful in future theoretical work. It involves a very general master lemma, Lemma E.1, which can be used to help prove all the theorems in this paper.

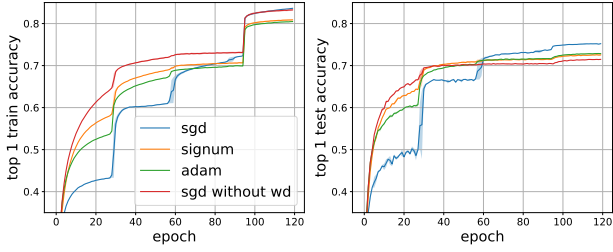


Figure 3. Imagenet train and test accuracies using the momentum version of SIGNSGD, called SIGNUM, to train Resnet-50 v2. We based our implementation on an open source implementation by github.com/tornadomeet. Initial learning rate and weight decay were tuned on a separate validation set split off from the training set and all other hyperparameters were chosen to be those found favourable for SGD by the community. There is a big jump at epoch 95 when we switch off data augmentation. SIGNUM gets test set performance approximately the same as ADAM, better than SGD with out weight decay, but about 2% worse than SGD with a well-tuned weight decay.

Remark 1: switching optimisers after a warmup period is in fact commonly done by practitioners (Akiba et al., 2017).

Remark 2: the theory suggests that momentum can be used to control a bias-variance tradeoff in the quality of stochastic gradient estimates. Sending $\beta \rightarrow 1$ kills the variance term in $\|\bar{\sigma}\|_1$ due to averaging gradients over a longer time horizon. But averaging in stale gradients induces bias due to curvature of $f(x)$, and this blows up the $\delta\|\tilde{Z}\|_1$ term.

Remark 3: for generality, we state this theorem with a tunable learning rate δ . For variety, we give this theorem in any-time form with a growing batch size and decaying learning rate. This comes at the cost of log factors appearing.

We benchmark SIGNUM on Imagenet (Figure 3) and CIFAR-10 (Figure A.3 of supplementary). The full results of a giant hyperparameter grid search for the CIFAR-10 experiments are also given in the supplementary. SIGNUM’s performance rivals ADAM’s in all experiments.

6. Discussion

Gradient compression schemes like TERNGRAD (Wen et al., 2017) quantise gradients into three levels $\{0, \pm 1\}$. This is desirable when the ternary quantisation is sparse, since it can allow further compression. Our scheme of majority vote should easily be compatible with a ternary quantisation—in both directions of communication. This can be cast as “majority vote with abstention”. The scheme is as follows: workers send their vote to the parameter server, unless they are very unsure about the sign of the true gradient in which case they send zero. The parameter-server counts the votes,

and if quorum is not reached (i.e. too many workers disagreed or abstained) the parameter-server sends back zero. This extended algorithm should readily fit into our theory.

In Section 2 we pointed out that SIGNSGD and SIGNUM are closely related to ADAM. In all our experiments we find that SIGNUM and ADAM have very similar performance, although both lose out to SGD by about 2% test accuracy on Imagenet. Wilson et al. (2017) observed that ADAM tends to generalise slightly worse than SGD. Though it is still unclear why this is the case, perhaps it could be because we don’t know how to properly regularise such methods. Whilst we found that neither standard weight decay nor the suggestion of Loshchilov & Hutter (2017) completely closed our Imagenet test set gap with SGD, it is possible that some other regularisation scheme might. One idea, suggested by our theory, is that SIGNSGD could be squashing down noise levels. There is some evidence (Smith & Le, 2018) that a certain level of noise can be good for generalisation, biasing the optimiser towards wider valleys in the objective function. Perhaps, then, adding Gaussian noise to the SIGNUM update might help it generalise better. This can be achieved in a communication efficient manner in the distributed setting by sharing a random seed with each worker, and then generating the same noise on each worker.

Finally, in Section 3 we discuss some geometric implications of our theory, and provide an efficient and robust experimental means of measuring one aspect—the ratio between noise and gradient density—through the Welford algorithm. We believe that since this density ratio is easy to measure, it may be useful to help guide those doing architecture search, to find network architectures which are amenable to fast training through gradient compression schemes.

7. Conclusion

We have presented a general framework for studying sign-based methods in stochastic non-convex optimisation. We present non-vacuous bounds for gradient compression schemes, and elucidate the special ℓ_1 geometries under which these schemes can be expected to succeed. Our theoretical framework is broad enough to handle signed-momentum schemes—like SIGNUM—and also multi-worker distributed schemes—like majority vote.

Our work touches upon interesting aspects of the geometry of high-dimensional error surfaces, which we wish to explore in future work. But the next step for us will be to reach out to members of the distributed systems community to help benchmark the majority vote algorithm which shows such great theoretical promise for 1-bit compression in both directions between parameter-server and workers.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments, as well as Jiawei Zhao, Michael Tschannen, Julian Salazar, Tan Nguyen, Fanny Yang, Mu Li, Aston Zhang and Zack Lipton for useful discussions. Thanks to Ryan Tibshirani for pointing out the connection to steepest descent.

KA is supported in part by NSF Career Award CCF-1254106 and Air Force FA9550-15-1-0221. AA is supported in part by Microsoft Faculty Fellowship, Google Faculty Research Award, Adobe Grant, NSF Career Award CCF-1254106, and AFOSR YIP FA9550-15-1-0221.

References

- Akiba, T., Suzuki, S., and Fukuda, K. Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes. *arXiv:1711.04325*, 2017.
- 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-17)*, 2017.
- Allen-Zhu, Z. Natasha: Faster Non-Convex Stochastic Optimization via Strongly Non-Convex Parameter. In *International Conference on Machine Learning (ICML-17)*, 2017a.
- Allen-Zhu, Z. Natasha 2: Faster Non-Convex Optimization Than SGD. *arXiv:1708.08694*, 2017b.
- Balles, L. and Hennig, P. Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients. *arXiv:1705.07774*, 2017.
- Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.
- Cantelli, F. P. Sui confini della probabilit. *Atti del Congresso Internazionale dei Matematici*, 1928.
- Carlson, D., Hsieh, Y., Collins, E., Carin, L., and Cevher, V. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):296–311, 2016.
- Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization. In *Advances in neural information processing systems (NIPS-14)*, 2014.
- Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.
- Gauss, C. F. Theoria combinationis observationum erroribus minimis obnoxiae, pars prior. *Commentationes Societatis Regiae Scientiarum Gottingensis Recentiores*, 1823.
- Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Glorot, X. and Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Artificial intelligence and statistics (AISTATS-10)*, 2010.
- Goh, G. Why Momentum Really Works. *Distill*, 2017.
- Goodfellow, I. J., Vinyals, O., and Saxe, A. M. Qualitatively Characterizing Neural Network Optimization Problems. In *International Conference on Learning Representations (ICLR-15)*, 2015.
- Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv:1706.02677*, 2017.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-16)*, 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-16)*, pp. 630–645. Springer, 2016b.
- Jin, C., Netrapalli, P., and Jordan, M. I. Accelerated Gradient Descent Escapes Saddle Points Faster than Gradient Descent. *arXiv:1711.10456*, 2017.
- Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD-16)*, pp. 795–811. Springer, 2016.
- Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR-15)*, 2015.
- Knuth, D. E. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1997.
- Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.
- LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436, 2015.

- 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 *Symposium on Operating Systems Design and Implementation (OSDI-14)*, pp. 583–598, 2014.
- Loshchilov, I. and Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv:1711.05101*, 2017.
- MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.
- Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2013.
- Nesterov, Y. and Polyak, B. Cubic Regularization of Newton Method and its Global Performance. *Mathematical Programming*, 2006.
- Pukelsheim, F. The Three Sigma Rule. *The American Statistician*, 1994.
- Reddi, S. J., Kale, S., and Kumar, S. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations (ICLR-18)*, 2018.
- Richtárik, P. and Takáč, M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (1-2):1–38, 2014.
- Riedmiller, M. and Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm. In *International Conference on Neural Networks (ICNN-93)*, pp. 586–591. IEEE, 1993.
- Robbins, H. and Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 1951.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.
- Schmidhuber, J. Deep Learning in Neural Networks: an Overview. *Neural Networks*, 2015.
- 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 *Conference of the International Speech Communication Association (INTERSPEECH-14)*, 2014.
- Smith, S. L. and Le, Q. V. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. In *International Conference on Learning Representations (ICLR-18)*, 2018.
- Strom, N. Scalable distributed DNN training using commodity GPU cloud computing. In *Conference of the International Speech Communication Association (INTERSPEECH-15)*, 2015.
- Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed Mean Estimation with Limited Communication. In *International Conference on Machine Learning (ICML-17)*, 2017.
- Tieleman, T. and Hinton, G. RMSprop. *Coursera: Neural Networks for Machine Learning*, Lecture 6.5, 2012.
- Welford, B. P. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 1962.
- 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-17)*, 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-17)*, 2017.

A. Further experimental results

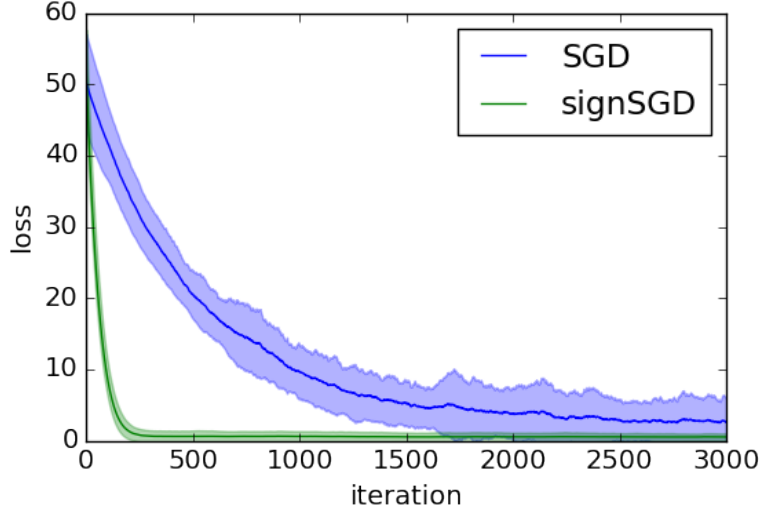


Figure A.1. A simple toy problem where SIGNSGD converges faster than SGD. The objective function is just a quadratic $f(x) = \frac{1}{2}x^2$ for $x \in \mathbb{R}^{100}$. The gradient of this function is just $g(x) = x$. We construct an artificial stochastic gradient by adding Gaussian noise $\mathcal{N}(0, 100^2)$ to only the first component of the gradient. Therefore the noise is extremely sparse. The initial point is sampled from a unit variance spherical Gaussian. For each algorithm we tune a separate, constant learning rate finding 0.001 best for SGD and 0.01 best for SIGNSGD. SIGNSGD appears more robust to the sparse noise in this problem. Results are averaged over 50 repeats with ± 1 standard deviation shaded.

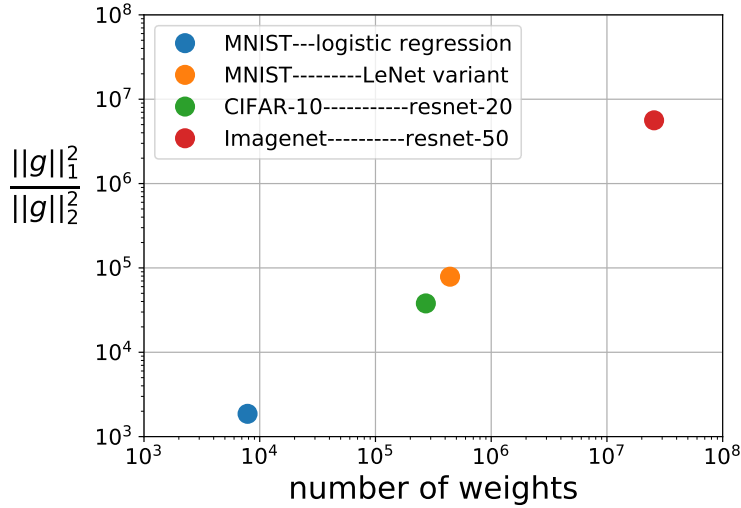


Figure A.2. Measuring gradient density via ratio of norms, over a range of datasets and architectures. For each network, we take a point in parameter space provided by the Xavier initialiser (Glorot & Bengio, 2010). We do a full pass over the data to compute the full gradient at this point. It is remarkably dense in all cases.

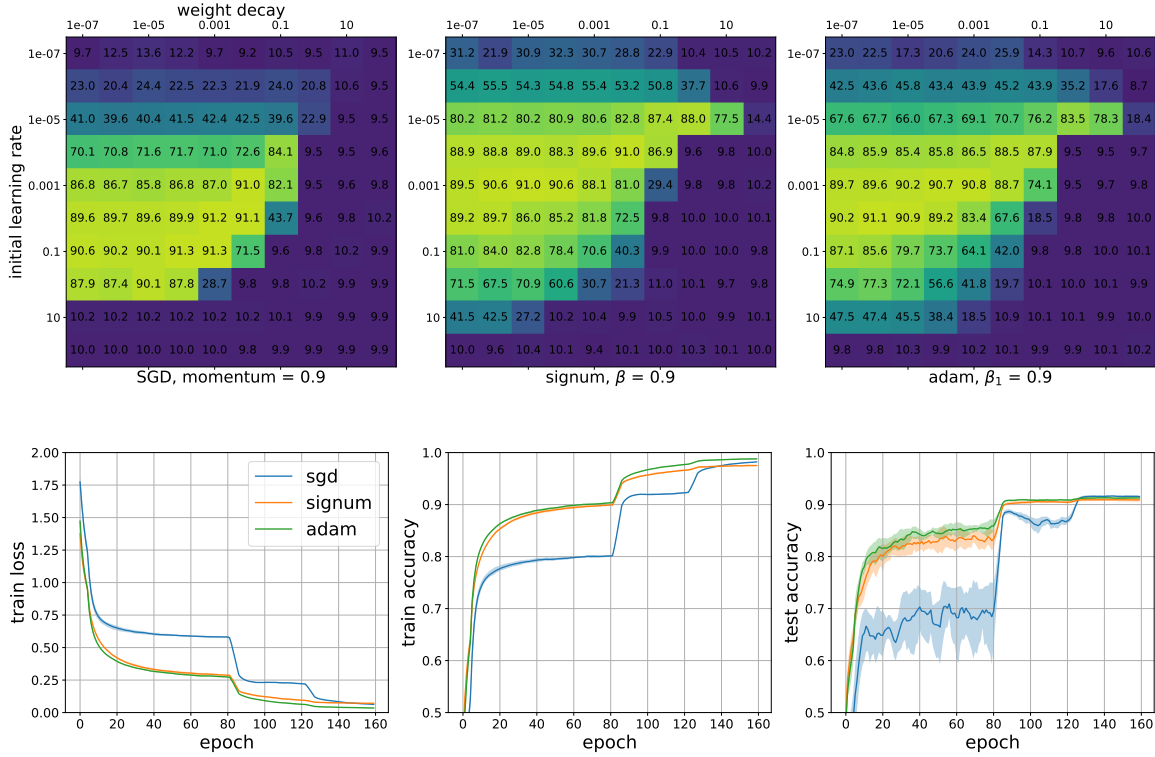


Figure A.3. CIFAR-10 results using SIGNUM to train a Resnet-20 model. Top: validation accuracies from a hyperparameter sweep on a separate validation set carved out from the training set. We used this to tune initial learning rate, weight decay and momentum for all algorithms. All other hyperparameter settings were chosen as in (He et al., 2016a) as found favourable for SGD. The hyperparameter sweep for other values of momentum is plotted in Figure A.4 of the supplementary. Bottom: there is little difference between the final test set performance of the algorithms. SIGNUM closely resembles ADAM in all of these plots.

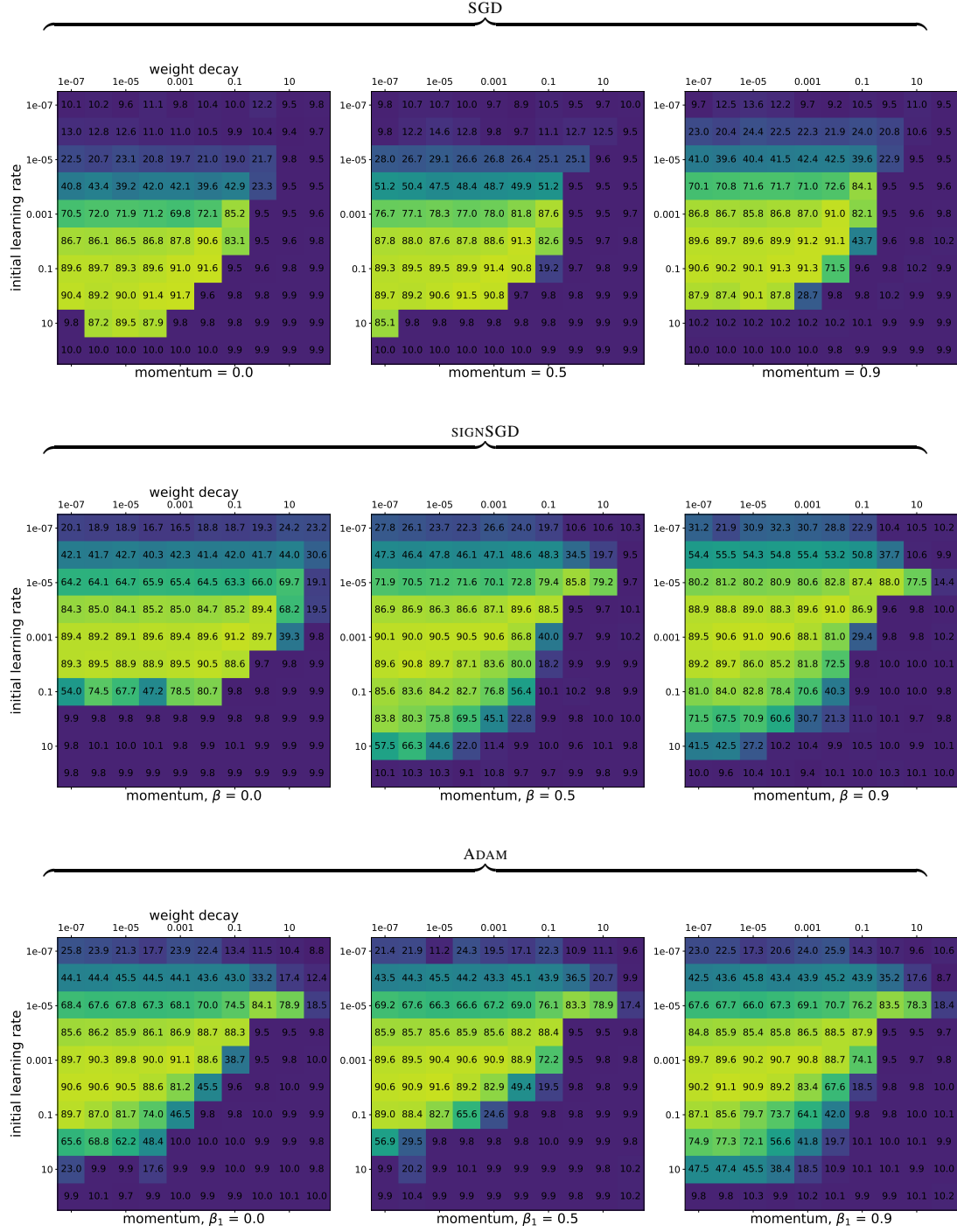


Figure A.4. Results of a massive grid search over hyperparameters for training Resnet-20 (He et al., 2016a) on CIFAR-10 (Krizhevsky, 2009). All non-algorithm specific hyperparameters (such as learning rate schedules) were set as in (He et al., 2016a). In ADAM, β_2 and ϵ were chosen as recommended in (Kingma & Ba, 2015). Data was divided according to a {45k/5k/10k} {train/val/test} split. Validation accuracies are plotted above, and the best performer on the validation set was chosen for the final test run (shown in Figure A.3). All algorithms at the least get close to the baseline reported in (He et al., 2016a) of 91.25%. Note the broad similarity in general shape of the heatmap between ADAM and SIGNSGD, supporting a notion of algorithmic similarity. Also note that whilst SGD has a larger region of very high-scoring hyperparameter configurations, SIGNSGD and ADAM appear stable over a larger range of learning rates. Notice that the SGD heatmap shifts up for increasing momentum, since the implementation of SGD in the mxnet deep learning framework actually couples the momentum and learning rate parameters.

B. Proving the convergence rate of SIGNSGD

Theorem 1 (Non-convex convergence rate of SIGNSGD). *Run algorithm 1 for K iterations under Assumptions 1 to 3. Set the learning rate and mini-batch size (independently of step k) as*

$$\delta_k = \frac{1}{\sqrt{\|\vec{L}\|_1 K}}, \quad n_k = K$$

Let N be the cumulative number of stochastic gradient calls up to step K , i.e. $N = O(K^2)$. Then we have

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right]^2 \\ & \leq \frac{1}{\sqrt{N}} \left[\sqrt{\|\vec{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + 2\|\vec{\sigma}\|_1 \right]^2 \end{aligned}$$

Proof. First let's bound the improvement of the objective during a single step of the algorithm for one instantiation of the noise. $\mathbb{I}[\cdot]$ is the indicator function, $g_{k,i}$ denotes the i^{th} component of the true gradient $g(x_k)$ and \tilde{g}_k is a stochastic sample obeying Assumption 3.

First take Assumption 2, plug in the step from Algorithm 1, and decompose the improvement to expose the stochasticity-induced error:

$$\begin{aligned} f_{k+1} - f_k & \leq g_k^T (x_{k+1} - x_k) + \sum_{i=1}^d \frac{L_i}{2} (x_{k+1} - x_k)_i^2 \\ & = -\delta_k g_k^T \text{sign}(\tilde{g}_k) + \delta_k^2 \sum_{i=1}^d \frac{L_i}{2} \\ & = -\delta_k \|g_k\|_1 + \frac{\delta_k^2}{2} \|\vec{L}\|_1 \\ & \quad + 2\delta_k \sum_{i=1}^d |g_{k,i}| \mathbb{I}[\text{sign}(\tilde{g}_{k,i}) \neq \text{sign}(g_{k,i})] \end{aligned}$$

Next we find the expected improvement at time $k+1$ conditioned on the previous iterate.

$$\begin{aligned} \mathbb{E}[f_{k+1} - f_k | x_k] & \leq -\delta_k \|g_k\|_1 + \frac{\delta_k^2}{2} \|\vec{L}\|_1 \\ & \quad + 2\delta_k \sum_{i=1}^d |g_{k,i}| \mathbb{P}[\text{sign}(\tilde{g}_{k,i}) \neq \text{sign}(g_{k,i})] \end{aligned}$$

So the expected improvement crucially depends on the probability that each component of the sign vector is correct, which is intuitively controlled by the relative scale of the gradient to the noise. To make this rigorous, first relax the probability, then use Markov's inequality followed by Jensen's inequality:

$$\begin{aligned} \mathbb{P}[\text{sign}(\tilde{g}_{k,i}) \neq \text{sign}(g_{k,i})] & \leq \mathbb{P}[|\tilde{g}_{k,i} - g_{k,i}| \geq |g_{k,i}|] \\ & \leq \frac{\mathbb{E}[|\tilde{g}_{k,i} - g_{k,i}|]}{|g_{k,i}|} \\ & \leq \frac{\sqrt{\mathbb{E}[(\tilde{g}_{k,i} - g_{k,i})^2]}}{|g_{k,i}|} \\ & = \frac{\sigma_{k,i}}{|g_{k,i}|} \end{aligned}$$

$\sigma_{k,i}$ refers to the variance of the k^{th} stochastic gradient estimate, computed over a mini-batch of size n_k . Therefore, by Assumption 3, we have that $\sigma_{k,i} \leq \sigma_i / \sqrt{n_k}$.

We now substitute these results and our learning rate and mini-batch settings into the expected improvement:

$$\begin{aligned} \mathbb{E}[f_{k+1} - f_k | x_k] &\leq -\delta_k \|g_k\|_1 + 2 \frac{\delta_k}{\sqrt{n_k}} \|\vec{\sigma}\|_1 + \frac{\delta_k^2}{2} \|\vec{L}\|_1 \\ &= -\frac{1}{\sqrt{\|\vec{L}\|_1 K}} \|g_k\|_1 + \frac{2}{\sqrt{\|\vec{L}\|_1 K}} \|\vec{\sigma}\|_1 + \frac{1}{2K} \end{aligned}$$

Now extend the expectation over randomness in the trajectory, and perform a telescoping sum over the iterations:

$$\begin{aligned} f_0 - f^* &\geq f_0 - \mathbb{E}[f_K] \\ &= \mathbb{E} \left[\sum_{k=0}^{K-1} f_k - f_{k+1} \right] \\ &\geq \mathbb{E} \sum_{k=0}^{K-1} \left[\frac{1}{\sqrt{\|\vec{L}\|_1 K}} \|g_k\|_1 - \frac{1}{2\sqrt{\|\vec{L}\|_1 K}} \left(4\|\vec{\sigma}\|_1 + \sqrt{\|\vec{L}\|_1} \right) \right] \\ &= \sqrt{\frac{K}{\|\vec{L}\|_1}} \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right] - \frac{1}{2\sqrt{\|\vec{L}\|_1}} \left(4\|\vec{\sigma}\|_1 + \sqrt{\|\vec{L}\|_1} \right) \end{aligned}$$

We can rearrange this inequality to yield the rate:

$$\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right] \leq \frac{1}{\sqrt{K}} \left[\sqrt{\|\vec{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + 2\|\vec{\sigma}\|_1 \right]$$

Since we are growing our mini-batch size, it will take $N = O(K^2)$ gradient calls to reach step K . Substitute this in, square the result, and we are done. \square

C. Large and small batch SGD

Algorithm C.1 SGD

Input: learning rate δ , current point x_k
 $\tilde{g}_k \leftarrow \text{stochasticGradient}(x_k)$
 $x_{k+1} \leftarrow x_k - \delta \tilde{g}_k$

For comparison with SIGNSGD theory, here we present non-convex convergence rates for SGD. These are classical results and we are not sure of the earliest reference.

We noticeably get exactly the same rate for large and small batch SGD when measuring convergence in terms of number of stochastic gradient calls. Although the rates are the same for a given number of gradient calls N , the large batch setting is preferred (in theory) since it achieves these N gradient calls in only \sqrt{N} iterations, whereas the small batch setting requires N iterations. Fewer iterations in the large batch case implies a smaller wall-clock time to reach a given accuracy (assuming the large batch can be parallelised) as well as fewer rounds of communication in the distributed setting. These systems benefits of large batch learning have been observed by practitioners (Goyal et al., 2017).

Theorem C.1 (Non-convex convergence rate of SGD). *Run algorithm C.1 for K iterations under Assumptions 1 to 3. Define $L := \|L\|_\infty$ and $\sigma^2 := \|\bar{\sigma}\|_2^2$. Set the learning rate and mini-batch size (independently of step k) as either*

$$\begin{aligned} (\text{large batch}) \quad \delta_k &= \frac{1}{L} & n_k &= K \\ (\text{small batch}) \quad \delta_k &= \frac{1}{L\sqrt{K}} & n_k &= 1 \end{aligned}$$

Let N be the cumulative number of stochastic gradient calls up to step K , i.e. $N = O(K^2)$ for large batch and $N = O(K)$ for small batch. Then, in either case, we have

$$\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_2^2 \right] \leq \frac{1}{\sqrt{N}} [2L(f_0 - f_*) + \sigma^2]$$

Proof. The proof begins the same for the large and small batch case.

First we bound the improvement of the objective during a single step of the algorithm for one instantiation of the noise. g_k denotes the true gradient at step k and \tilde{g}_k is a stochastic sample obeying Assumption 3.

Take Assumption 2 and plug in the algorithmic step.

$$\begin{aligned} f_{k+1} - f_k &\leq g_k^T (x_{k+1} - x_k) + \sum_{i=1}^d \frac{L_i}{2} (x_{k+1} - x_k)_i^2 \\ &\leq g_k^T (x_{k+1} - x_k) + \frac{\|L\|_\infty}{2} \|x_{k+1} - x_k\|_2^2 \\ &= -\delta_k g_k^T \tilde{g}_k + \delta_k^2 \frac{L}{2} \|\tilde{g}_k\|_2^2 \end{aligned}$$

Next we find the expected improvement at time $k+1$ conditioned on the previous iterate.

$$\mathbb{E}[f_{k+1} - f_k | x_k] \leq -\delta_k \|g_k\|_2^2 + \delta_k^2 \frac{L}{2} (\sigma_k^2 + \|g_k\|_2^2).$$

σ_k^2 refers to the variance of the k^{th} stochastic gradient estimate, computed over a mini-batch of size n_k . Therefore, by Assumption 3, we have that $\sigma_k^2 \leq \sigma^2/n_k$.

First let's substitute in the **(large batch)** hyperparameters.

$$\begin{aligned} \mathbb{E}[f_{k+1} - f_k | x_k] &\leq -\frac{1}{L} \|g_k\|_2^2 + \frac{1}{2L} \left(\frac{\sigma^2}{K} + \|g_k\|_2^2 \right) \\ &= -\frac{1}{2L} \|g_k\|_2^2 + \frac{1}{2L} \frac{\sigma^2}{K}. \end{aligned}$$

Now extend the expectation over randomness in the trajectory, and perform a telescoping sum over the iterations:

$$\begin{aligned} f_0 - f^* &\geq f_0 - \mathbb{E}[f_K] \\ &= \mathbb{E} \left[\sum_{k=0}^{K-1} f_k - f_{k+1} \right] \\ &\geq \frac{1}{2L} \mathbb{E} \sum_{k=0}^{K-1} [\|g_k\|_2^2 - \sigma^2] \end{aligned}$$

We can rearrange this inequality to yield the rate:

$$\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_2^2 \right] \leq \frac{1}{K} [2L(f_0 - f_*) + \sigma^2].$$

Since we are growing our mini-batch size, it will take $N = O(K^2)$ gradient calls to reach step K . Substitute this in and we are done for the **(large batch)** case.

Now we need to show that the same result holds for the **(small batch)** case. Following the initial steps of the large batch proof, we get

$$\mathbb{E}[f_{k+1} - f_k | x_k] \leq -\delta_k \|g_k\|_2^2 + \delta_k^2 \frac{L}{2} (\sigma_k^2 + \|g_k\|_2^2).$$

This time $\sigma_k^2 = \sigma^2$. Substituting this and our learning rate and mini-batch settings into the expected improvement:

$$\begin{aligned} \mathbb{E}[f_{k+1} - f_k | x_k] &\leq -\frac{1}{L\sqrt{K}} \|g_k\|_2^2 + \frac{1}{2LK} (\sigma^2 + \|g_k\|_2^2) \\ &\leq -\frac{1}{2L\sqrt{K}} \|g_k\|_2^2 + \frac{1}{2L} \frac{\sigma^2}{K}. \end{aligned}$$

Now extend the expectation over randomness in the trajectory, and perform a telescoping sum over the iterations:

$$\begin{aligned} f_0 - f^* &\geq f_0 - \mathbb{E}[f_K] \\ &= \mathbb{E} \left[\sum_{k=0}^{K-1} f_k - f_{k+1} \right] \\ &\geq \frac{1}{2L} \mathbb{E} \sum_{k=0}^{K-1} \left[\frac{\|g_k\|_2^2}{\sqrt{K}} - \frac{\sigma^2}{K} \right]. \end{aligned}$$

We can rearrange this inequality to yield the rate:

$$\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_2^2 \right] \leq \frac{1}{\sqrt{K}} [2L(f_0 - f_*) + \sigma^2]$$

It will take $N = O(K)$ gradient calls to reach step K . Substitute this in and we are done. \square

D. Proving the convergence rate of distributed SIGNSGD with majority vote

Theorem 2 (Non-convex convergence rate of distributed SIGNSGD with majority vote). *Run algorithm 3 for K iterations under Assumptions 1 to 3. Set the learning rate and mini-batch size for each worker (independently of step k) as*

$$\delta_k = \frac{1}{\sqrt{\|\vec{L}\|_1 K}} \quad n_k = K$$

Then (a) majority vote with M workers converges at least as fast as SIGNSGD in Theorem 1.

And (b) further assuming that the noise in each component of the stochastic gradient is unimodal and symmetric about the mean (e.g. Gaussian), majority vote converges at improved rate:

$$\begin{aligned} &\mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \|g_k\|_1 \right]^2 \\ &\leq \frac{1}{\sqrt{N}} \left[\sqrt{\|\vec{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + \frac{2}{\sqrt{M}} \|\sigma\|_1 \right]^2 \end{aligned}$$

where N is the cumulative number of stochastic gradient calls per worker up to step K .

Before we introduce the unimodal symmetric assumption, let's first address the claim that M -worker majority vote is at least as good as single-worker SIGNSGD as in Theorem 1 only using Assumptions 1 to 3.

Proof of (a). Recall that a crucial step in Theorem 1 is showing that

$$|g_i| \mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] \leq \sigma_i$$

for component i of the stochastic gradient with variance bound σ_i .

The only difference in majority vote is that instead of using $\text{sign}(\tilde{g}_i)$ to approximate $\text{sign}(g_i)$, we are instead using $\text{sign}\left[\sum_{m=1}^M \text{sign}(\tilde{g}_{m,i})\right]$. If we can show that the same bound in terms of σ_i holds instead for

$$|g_i| \mathbb{P}\left[\text{sign}\left[\sum_{m=1}^M \text{sign}(\tilde{g}_{m,i})\right] \neq \text{sign}(g_i)\right] \quad (\star)$$

then we are done, since the machinery of Theorem 1 can then be directly applied.

Define the signal-to-noise ratio of a component of the stochastic gradient as $S := |g_i|/\sigma_i$. Note that when $S \leq 1$ then (\star) is trivially satisfied, so we need only consider the case that $S > 1$. S should really be labeled S_i but we abuse notation.

Without loss of generality, assume that g_i is negative, and thus using Assumption 3 and Cantelli's inequality (Cantelli, 1928) we get that for the failure probability q of a single worker

$$q := \mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] = \mathbb{P}[\tilde{g}_i - g_i \geq |g_i|] \leq \frac{1}{1 + \frac{g_i^2}{\sigma_i^2}}$$

For $S > 1$ then we have failure probability $q < \frac{1}{2}$. If the failure probability of a single worker is smaller than $\frac{1}{2}$ then the server is essentially receiving a repetition code \tilde{R}_M of the true gradient sign. Majority vote is the maximum likelihood decoder of the repetition code, and of course decreases the probability of error—see e.g. (MacKay, 2002). Therefore in all regimes of S we have that

$$(\star) \leq |g_i| \mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] \leq \sigma_i$$

and we are done. \square

That's all well and good, but what we'd really like to show is that using M workers provides a speedup by reducing the variance. Is

$$(\star) \stackrel{?}{\leq} \frac{\sigma_i}{\sqrt{M}} \quad (\dagger)$$

too much to hope for?

Well in the regime where $S \gg 1$ such a speedup is very reasonable since $q \ll \frac{1}{2}$ by Cantelli, and the repetition code actually supplies exponential reduction in failure rate. But we need to exclude very skewed or bimodal distributions where $q > \frac{1}{2}$ and adding more voting workers will not help. That brings us naturally to the following lemma:

Lemma D.1 (Failure probability of a sign bit under conditions of unimodal symmetric gradient noise). *Let \tilde{g}_i be an unbiased stochastic approximation to gradient component g_i , with variance bounded by σ_i^2 . Further assume that the noise distribution is unimodal and symmetric. Define signal-to-noise ratio $S := \frac{|g_i|}{\sigma_i}$. Then we have that*

$$\mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] \leq \begin{cases} \frac{2}{9} \frac{1}{S^2} & \text{if } S > \frac{2}{\sqrt{3}}, \\ \frac{1}{2} - \frac{S}{2\sqrt{3}} & \text{otherwise} \end{cases}$$

which is in all cases less than $\frac{1}{2}$.

Proof. Recall Gauss' inequality for unimodal random variable X with mode ν and expected squared deviation from the mode τ^2 (Gauss, 1823; Pukelsheim, 1994):

$$\mathbb{P}[|X - \nu| > k] \leq \begin{cases} \frac{4}{9} \frac{\tau^2}{k^2} & \text{if } \frac{k}{\tau} > \frac{2}{\sqrt{3}}, \\ 1 - \frac{k}{\sqrt{3}\tau} & \text{otherwise} \end{cases}$$

By the symmetry assumption, the mode is equal to the mean, so we replace mean $\mu = \nu$ and variance $\sigma^2 = \tau^2$.

$$\mathbb{P}[|X - \mu| > k] \leq \begin{cases} \frac{4}{9} \frac{\sigma^2}{k^2} & \text{if } \frac{k}{\sigma} > \frac{2}{\sqrt{3}}, \\ 1 - \frac{k}{\sqrt{3}\sigma} & \text{otherwise} \end{cases}$$

Without loss of generality assume that g_i is negative. Then applying symmetry followed by Gauss, the failure probability for the sign bit satisfies:

$$\begin{aligned} \mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] &= \mathbb{P}[\tilde{g}_i - g_i \geq |g_i|] \\ &= \frac{1}{2} \mathbb{P}[|\tilde{g}_i - g_i| \geq |g_i|] \\ &\leq \begin{cases} \frac{2}{9} \frac{\sigma_i^2}{g_i^2} & \text{if } \frac{|g_i|}{\sigma} > \frac{2}{\sqrt{3}}, \\ \frac{1}{2} - \frac{|g_i|}{2\sqrt{3}\sigma_i} & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{2}{9} \frac{1}{S^2} & \text{if } S > \frac{2}{\sqrt{3}}, \\ \frac{1}{2} - \frac{S}{2\sqrt{3}} & \text{otherwise} \end{cases} \end{aligned}$$

□

We now have everything we need to prove part (b) of Theorem 2.

Proof of (b). If we can show (†) we'll be done, since the machinery of Theorem 1 follows through with σ replaced everywhere by $\frac{\sigma}{\sqrt{M}}$. Note that the important quantity appearing in (★) is

$$\sum_{m=1}^M \text{sign}(\tilde{g}_{m,i}).$$

Let Z count the number of workers with correct sign bit. To ensure that

$$\text{sign} \left[\sum_{m=1}^M \text{sign}(\tilde{g}_{m,i}) \right] = \text{sign}(g_i)$$

Z must be larger than $\frac{M}{2}$. But Z is the sum of M independent Bernoulli trials, and is therefore binomial with success probability p and failure probability q to be determined. Therefore we have reduced proving (†) to showing that

$$\mathbb{P} \left[Z \leq \frac{M}{2} \right] \leq \frac{1}{\sqrt{MS}} \quad (\S)$$

where Z is the number of successes of a binomial random variable $b(M, p)$ and S is our signal-to-noise ratio $S := \frac{|g_i|}{\sigma_i}$.

Let's start by getting a bound on the success probability p (or equivalently failure probability q) of a single Bernoulli trial.

By Lemma D.1, which critically relies on unimodal symmetric gradient noise, the failure probability for the sign bit of a single worker satisfies:

$$\begin{aligned} q &:= \mathbb{P}[\text{sign}(\tilde{g}_i) \neq \text{sign}(g_i)] \\ &\leq \begin{cases} \frac{2}{9} \frac{1}{S^2} & \text{if } S > \frac{2}{\sqrt{3}}, \\ \frac{1}{2} - \frac{S}{2\sqrt{3}} & \text{otherwise} \end{cases} \\ &:= \tilde{q}(S) \end{aligned}$$

Where we have defined $\tilde{q}(S)$ to be our S -dependent bound on q . Since $q \leq \tilde{q}(S) < \frac{1}{2}$, there is hope to show (\dagger) . Define ϵ to be the defect of q from one half, and let $\tilde{\epsilon}(S)$ be its S -dependent bound.

$$\epsilon := \frac{1}{2} - q = p - \frac{1}{2} \geq \frac{1}{2} - \tilde{q}(S) := \tilde{\epsilon}(S)$$

Now we have an analytical handle on random variable Z , we may proceed to show (\S) . There are a number of different inequalities that we can use to bound the tail of a binomial random variable, but Cantelli's inequality will be good enough for our purposes.

Let $\bar{Z} := M - Z$ denote the number of failures. \bar{Z} is binomial with mean $\mu_{\bar{Z}} = Mq$ and variance $\sigma_{\bar{Z}}^2 = Mpq$. Then using Cantelli we get

$$\begin{aligned} \mathbb{P}\left[Z \leq \frac{M}{2}\right] &= \mathbb{P}\left[\bar{Z} \geq \frac{M}{2}\right] \\ &= \mathbb{P}\left[\bar{Z} - \mu_{\bar{Z}} \geq \frac{M}{2} - \mu_{\bar{Z}}\right] \\ &= \mathbb{P}\left[\bar{Z} - \mu_{\bar{Z}} \geq M\epsilon\right] \\ &\leq \frac{1}{1 + \frac{M^2\epsilon^2}{Mpq}} \\ &\leq \frac{1}{1 + \frac{M}{\frac{1}{4\epsilon^2} - 1}} \end{aligned}$$

Now using the fact that $\frac{1}{1+x^2} \leq \frac{1}{2x}$ we get

$$\mathbb{P}\left[Z \leq \frac{M}{2}\right] \leq \frac{\sqrt{\frac{1}{4\epsilon^2} - 1}}{2\sqrt{M}}$$

To finish, we need only show that $\sqrt{\frac{1}{4\epsilon^2} - 1}$ is smaller than $\frac{2}{S}$, or equivalently that its square is smaller than $\frac{4}{S^2}$. Well plugging in our bound on ϵ we get that

$$\frac{1}{4\epsilon^2} - 1 \leq \frac{1}{4\tilde{\epsilon}(S)^2} - 1$$

where

$$\tilde{\epsilon}(S) = \begin{cases} \frac{1}{2} - \frac{2}{9} \frac{1}{S^2} & \text{if } S > \frac{2}{\sqrt{3}}, \\ \frac{S}{2\sqrt{3}} & \text{otherwise} \end{cases}$$

First take the case $S \leq \frac{2}{\sqrt{3}}$. Then $\tilde{\epsilon}^2 = \frac{S^2}{12}$ and $\frac{1}{4\tilde{\epsilon}^2} - 1 = \frac{3}{S^2} - 1 < \frac{4}{S^2}$. Now take the case $S > \frac{2}{\sqrt{3}}$. Then $\tilde{\epsilon} = \frac{1}{2} - \frac{2}{9} \frac{1}{S^2}$ and we have $\frac{1}{4\tilde{\epsilon}^2} - 1 = \frac{1}{S^2} \frac{\frac{8}{9} - \frac{16}{81} \frac{1}{S^2}}{1 - \frac{8}{9} \frac{1}{S^2} + \frac{16}{81} \frac{1}{S^4}} < \frac{1}{S^2} \frac{\frac{8}{9}}{1 - \frac{8}{9} \frac{1}{S^2}} < \frac{4}{S^2}$ by the condition on S .

So we have shown both cases, which proves (\S) from which we get (\dagger) and we are done. \square

E. General recipes for the convergence of approximate sign gradient methods

Now we generalize the arguments in the proof of SIGNSGD and prove a master lemma that provides a general recipe for analyzing the approximate sign gradient method. This allows us to handle momentum and the majority voting schemes, hence proving Theorem 3 and Theorem 2.

Lemma E.1 (Convergence rate for a class of approximate sign gradient method). *Let C and K be integers satisfying $0 < C \ll K$. Consider the algorithm given by $x_{k+1} = x_k - \delta_k \text{sign}(v_k)$, for a fixed positive sequence of δ_k and where $v_k \in \mathbb{R}^d$ is a measurable and square integrable function of the entire history up to time k , including $x_1, \dots, x_k, v_1, \dots, v_{k-1}$*

and all N_k stochastic gradient oracle calls up to time k . Let $g_k = \nabla f(x_k)$. If Assumption 1 and Assumption 2 are true and in addition for $k = C, C+1, C+2, \dots, K$

$$\mathbb{E} \left[\sum_{i=1}^d |g_{k,i}| \mathbb{P}[\text{sign}(v_{k,i}) \neq \text{sign}(g_{k,i}) | x_k] \right] \leq \xi(k) \quad (2)$$

where the expectation is taken over the all random variables, and the rate $\xi(k)$ obeys that $\xi(k) \rightarrow 0$ as $k \rightarrow \infty$ and then we have

$$\frac{1}{K-C} \sum_{k=C}^{K-1} \mathbb{E} \|g_k\|_1 \leq \frac{f_C - f_* + 2 \sum_{k=C}^{K-1} \delta_k \xi(k) + \sum_{k=C}^{K-1} \frac{\delta_k^2 \|\vec{L}\|_1}{2}}{(K-C) \min_{C \leq k \leq K-1} \delta_k}.$$

In particular, if $\delta_k = \delta/\sqrt{k}$ and $\xi(k) = \kappa/\sqrt{k}$, for some problem dependent constant κ , then we have

$$\frac{1}{K-C} \sum_{k=C}^{K-1} \mathbb{E} \|g_k\|_1 \leq \frac{\frac{f_C - f_*}{\delta} + (2\kappa + \|\vec{L}\|_1 \delta/2)(\log K + 1)}{\sqrt{K} - \frac{C}{\sqrt{K}}}.$$

Proof. Our general strategy will be to show that the expected objective improvement at each step will be good enough to guarantee a convergence rate in expectation. First let's bound the improvement of the objective during a single step of the algorithm for $k \geq C$, and then take expectation. Note that $\mathbb{I}[\cdot]$ is the indicator function, and $w_k[i]$ denotes the i^{th} component of the vector w_k .

By Assumption 2

$$\begin{aligned} f_{k+1} - f_k &\leq g_k^T (x_{k+1} - x_k) + \sum_{i=1}^d \frac{L_i}{2} (x_{k+1,i} - x_{k,i})^2 && \text{Assumption 2} \\ &= -\delta_k g_k^T \text{sign}(v_k) + \delta_k^2 \frac{\|\vec{L}\|_1}{2} && \text{by the update rule} \\ &= -\delta_k \|g_k\|_1 + 2\delta_k \sum_{i=1}^d |g_k[i]| \mathbb{I}[\text{sign}(v_k[i]) \neq \text{sign}(g_k[i])] + \delta_k^2 \frac{\|\vec{L}\|_1}{2} && \text{by identity} \end{aligned}$$

Now, for $k \geq C$ we need to find the expected improvement at time $k+1$ conditioned on x_k , where the expectation is over the randomness of the stochastic gradient oracle. Note that $\mathbb{P}[E]$ denotes the probability of event E .

$$\mathbb{E}[f_{k+1} - f_k | x_k] \leq -\delta_k \|g_k\|_1 + 2\delta_k \sum_{i=1}^d |g_k[i]| \mathbb{P}[\text{sign}(v_k[i]) \neq \text{sign}(g_k[i]) | x_k] + \delta_k^2 \frac{\|\vec{L}\|_1}{2}.$$

Note that g_k becomes fixed when we condition on x_k . Further take expectation over x_k , and apply (2). We get:

$$\mathbb{E}[f_{k+1} - f_k] \leq -\delta_k \mathbb{E}[\|g_k\|_1] + 2\delta_k \xi(k) + \frac{\delta_k^2 \|\vec{L}\|_1}{2}. \quad (3)$$

Rearrange the terms and sum over (3) for $k = C, C+1, \dots, K-1$.

$$\sum_{k=C}^{K-1} \delta_k \mathbb{E}[\|g_k\|_1] \leq \sum_{k=C}^{K-1} (\mathbb{E} f_k - \mathbb{E} f_{k+1}) + 2 \sum_{k=C}^{K-1} \delta_k \xi(k) + \sum_{k=C}^{K-1} \frac{\delta_k^2 \|\vec{L}\|_1}{2}$$

Dividing both sides by $[(K-C) \min_{C \leq k \leq K-1} \delta_k]$, using a telescoping sum over $\mathbb{E} f_k$ and using that $f(x) \geq f_*$ for all x , we get

$$\frac{1}{K-C} \sum_{k=C}^{K-1} \mathbb{E} \|g_k\|_1 \leq \frac{f_C - f_* + 2 \sum_{k=C}^{K-1} \delta_k \xi(k) + \sum_{k=C}^{K-1} \frac{\delta_k^2 \|\vec{L}\|_1}{2}}{(K-C) \min_{C \leq k \leq K-1} \delta_k}$$

and the proof is complete by noting that the minimum is smaller than the average in the LHS. \square

To use the above Lemma for analyzing SIGNUM and the Majority Voting scheme, it suffices to check condition (2) for each algorithm.

One possible way to establish (2) is show that v_k is a good approximation of the gradient g_k in expected absolute value.

Lemma E.2 (Estimation to testing reduction). *Equation (2) is true, if for every k*

$$\sum_{i=1}^d \mathbb{E} |v_k[i] - g_k[i]| \leq \xi(k). \quad (4)$$

Proof. First note that for any two random variables $a, b \in \mathbb{R}$.

$$\mathbb{P}[\text{sign}(a) \neq \text{sign}(b)] \leq \mathbb{P}[|a - b| > |b|].$$

Condition on x_k and apply the above inequality to every $i = 1, \dots, d$ to what is inside the expectation of (2), we have

$$\sum_{i=1}^d |g_k[i]| \mathbb{P}[\text{sign}(v_k[i]) \neq \text{sign}(g_k[i]) | x_k] \leq \sum_{i=1}^d |g_k[i]| \mathbb{P}[|v_k[i] - g_k[i]| > |g_k[i]| | x_k] \leq \sum_{i=1}^d \mathbb{E}[|v_k[i] - g_k[i]| | x_k].$$

Note that the final \leq uses Markov's inequality and constant $|g_k[i]|$ cancels out.

The proof is complete by taking expectation on both sides and apply (4). \square

Note that the proof of this lemma uses Markov's inequality in the same way information-theoretical lower bounds are often proved in statistics — reducing estimation to testing.

Another handy feature of the result is that we do not require the approximation to hold for every possible x_k . It is okay that for some x_k , the approximation is much worse as long as those x_k appears with small probability according to the algorithm. This feature enables us to analyze momentum and hence proving the convergence for SIGNUM.

F. Analysis for SIGNUM

Recall our definition of the key random variables used in SIGNUM.

$$\begin{aligned} g_k &:= \nabla f(x_k) \\ \tilde{g}_k &:= \frac{1}{n_k} \sum_{j=1}^{n_k} \tilde{g}^{(j)}(x_k) \\ m_k &:= \frac{1 - \beta}{1 - \beta^{k+1}} \sum_{t=0}^k [\beta^t g_{k-t}] \\ \tilde{m}_k &:= \frac{1 - \beta}{1 - \beta^{k+1}} \sum_{t=0}^k [\beta^t \tilde{g}_{k-t}] \end{aligned}$$

SIGNUM effectively uses $v_k = \tilde{m}_k$ and also $\delta_k = O(1/\sqrt{k})$.

Before we prove the convergence of SIGNUM, we first prove a utility lemma about the random variable $Z_k := \tilde{g}_k - g_k$. Note that in this lemma quantities like $Z_k, Y_k, |Z_k|$ and Z_k^2 are considered vectors—so this lemma is a statement about each component of the vectors separately and all operations, such as $(\cdot)^2$ are pointwise operations.

Lemma F.1 (Cumulative error of stochastic gradient). *For any $k < \infty$ and fixed weight $-\infty < \alpha_1, \dots, \alpha_k < \infty$, $\sum_{l=1}^k \alpha_l Z_l$ is a Martingale. In particular,*

$$\mathbb{E} \left[\left(\sum_{l=1}^k \alpha_l Z_l \right)^2 \right] \leq \sum_{l=1}^k \alpha_l^2 \bar{\sigma}^2.$$

Proof. We simply check the definition of a Martingale. Denote $Y_k := \sum_{l=1}^k \alpha_l Z_l$. First, we have that

$$\begin{aligned}
 \mathbb{E}[Y_k] &= \mathbb{E}\left[\sum_{l=1}^k \alpha_l Z_l\right] \\
 &\leq \sum_l |\alpha_l| \mathbb{E}[|Z_l|] && \text{triangle inequality} \\
 &= \sum_l |\alpha_l| \mathbb{E}[\mathbb{E}[|Z_l| | x_l]] && \text{law of total probability} \\
 &\leq \sum_l |\alpha_l| \mathbb{E}\left[\sqrt{\mathbb{E}[Z_l^2 | x_l]}\right] && \text{Jensen's inequality} \\
 &\leq \sum_l |\alpha_l| \bar{\sigma} < \infty
 \end{aligned}$$

Second, again using the law of total probability,

$$\begin{aligned}
 \mathbb{E}[Y_{k+1} | Y_1, \dots, Y_k] &= \mathbb{E}\left[\sum_{l=1}^{k+1} \alpha_l Z_l \mid \alpha_1 Z_1, \dots, \alpha_k Z_k\right] \\
 &= Y_k + \alpha_{k+1} \mathbb{E}[Z_{k+1} | \alpha_1 Z_1, \dots, \alpha_k Z_k] \\
 &= Y_k + \alpha_{k+1} \mathbb{E}[\mathbb{E}[Z_{k+1} | x_{k+1}, \alpha_1 Z_1, \dots, \alpha_k Z_k] | \alpha_1 Z_1, \dots, \alpha_k Z_k] \\
 &= Y_k + \alpha_{k+1} \mathbb{E}[\mathbb{E}[Z_{k+1} | x_{k+1}] | \alpha_1 Z_1, \dots, \alpha_k Z_k] \\
 &= Y_k
 \end{aligned}$$

This completes the proof that it is indeed a Martingale. We now make use of the properties of Martingale difference sequences to establish a variance bound on the Martingale.

$$\begin{aligned}
 \mathbb{E}\left[\left(\sum_{l=1}^k \alpha_l Z_l\right)^2\right] &= \sum_{l=1}^k \mathbb{E}[\alpha_l^2 Z_l^2] + 2 \sum_{l < j} \mathbb{E}[\alpha_l \alpha_j Z_l Z_j] \\
 &= \sum_{l=1}^k \alpha_l^2 \mathbb{E}[\mathbb{E}[Z_l^2 | Z_1, \dots, Z_{l-1}]] + 2 \sum_{l < j} \alpha_l \alpha_j \mathbb{E}\left[Z_l \mathbb{E}[\mathbb{E}[Z_j | Z_1, \dots, Z_{j-1}] | Z_l]\right] \\
 &= \sum_{l=1}^k \alpha_l^2 \mathbb{E}[\mathbb{E}[\mathbb{E}[Z_l^2 | x_l, Z_1, \dots, Z_{l-1}] | Z_1, \dots, Z_{l-1}]] + 0 \\
 &= \sum_{l=1}^k \alpha_l^2 \bar{\sigma}^2.
 \end{aligned}$$

□

The consequence of this lemma is that we are able to treat Z_1, \dots, Z_k as if they are independent, even though they are not—clearly Z_l is dependent on Z_1, \dots, Z_{l-1} through x_l .

Lemma F.2 (Gradient approximation in SIGNUM). *The version of the SIGNUM algorithm that takes $v_k = \tilde{m}_k$, and all parameters according to Theorem 3, obeys that for all integer $C \leq k \leq K$*

$$\|\mathbb{E}[\tilde{m}_k - g_k]\|_1 \leq \frac{2}{\sqrt{k+1}} \left(8\|\tilde{L}\|_1 \delta \frac{\beta}{1-\beta} + \sqrt{3}\|\tilde{\sigma}\|_1 \sqrt{1-\beta} \right).$$

Proof. For each $i \in [d]$ we will use the following non-standard “bias-variance” decomposition.

$$\mathbb{E}[|\tilde{m}_k[i] - g_k[i]|] \leq \underbrace{\mathbb{E}[|m_k[i] - g_k[i]|]}_{(*)} + \underbrace{\mathbb{E}[|\tilde{m}_k[i] - m_k[i]|]}_{(**)} \quad (5)$$

We will first bound $(**)$ and then deal with $(*)$.

Note that $(**) = \frac{1-\beta}{1-\beta^{k+1}} \mathbb{E} \left[\left| \sum_{t=0}^k \beta^{k-t} Z_t \right|^2 \right]$. Using Jensen's inequality and applying Lemma F.1 with our choice of $\alpha_1, \dots, \alpha_l$ (including the effect of the increasing batch size) we get that for $k \geq C$

$$(**) \leq \frac{1-\beta}{1-\beta^{k+1}} \sqrt{\mathbb{E} \left[\left| \sum_{t=0}^k \beta^t Z_{k-t} \right|^2 \right]} \leq \frac{1-\beta}{1-\beta^{k+1}} \sqrt{\sum_{t=0}^k \left[\beta^{2t} \frac{\bar{\sigma}^2}{k-t+1} \right]},$$

where

$$\begin{aligned} \sum_{t=0}^k \left[\beta^{2t} \frac{\bar{\sigma}^2}{k-t+1} \right] &= \sum_{t=0}^{\frac{k}{2}} \left[\beta^{2t} \frac{\bar{\sigma}^2}{k-t+1} \right] + \sum_{t=\frac{k}{2}+1}^k \left[\beta^{2t} \frac{\bar{\sigma}^2}{k-t+1} \right] && \text{break up sum} \\ &\leq \sum_{t=0}^{\frac{k}{2}} \left[\beta^{2t} \right] \frac{\bar{\sigma}^2}{\frac{k}{2}+1} + \sum_{t=\frac{k}{2}+1}^k \left[\beta^k \bar{\sigma}^2 \right] && \text{bound summands} \\ &\leq \frac{1}{1-\beta^2} \frac{\bar{\sigma}^2}{\frac{k}{2}+1} + \frac{k}{2} \beta^k \bar{\sigma}^2 && \text{geometric series} \\ &\leq \frac{3}{1-\beta^2} \frac{\bar{\sigma}^2}{k+1} && \text{since } k \geq C \end{aligned}$$

Combining, and again using our condition that $k \geq C$, we get

$$(**) \leq \frac{1-\beta}{1-\beta^{k+1}} \sqrt{\frac{3}{1-\beta^2}} \frac{\bar{\sigma}}{\sqrt{k+1}} \leq 2\sqrt{3} \sqrt{1-\beta} \frac{\bar{\sigma}}{\sqrt{k+1}} \quad (6)$$

We now turn to bounding $(*)$ — the “bias” term.

$$\begin{aligned} \mathbb{E}[|m_k - g_k|] &= \mathbb{E} \left[\left| \frac{1-\beta}{1-\beta^{k+1}} \sum_{t=0}^k [\beta^t g_{k-t}] - \frac{1-\beta}{1-\beta^{k+1}} \sum_{t=0}^k [\beta^t g_k] \right| \right] && \text{since } \frac{1-\beta}{1-\beta^{k+1}} \sum_{t=0}^k \beta^t = 1 \\ &\leq \frac{1-\beta}{1-\beta^{k+1}} \sum_{t=0}^k [\beta^t \mathbb{E}[|g_{k-t} - g_k|]] \\ &\leq 2(1-\beta) \sum_{t=1}^k \beta^t \mathbb{E}[|g_{k-t} - g_k|] && \text{since } k \geq C \end{aligned} \quad (7)$$

To proceed, we need the following lemma.

Lemma F.3. *Under Assumption 2, for any sign vector $s \in \{-1, 1\}^d$, any $x \in \mathbb{R}^d$ and any $\epsilon \leq \delta$*

$$\|g(x + \epsilon s) - g(x)\|_1 \leq 2\epsilon \|\vec{L}\|_1.$$

Proof. By Taylor's theorem,

$$g(x + \epsilon s) - g(x) = \left[\int_{t=0}^1 H(x + t\epsilon s) dt \right] \epsilon s.$$

Let $v := \text{sign}(g(x + \epsilon s) - g(x))$, $H := \left[\int_{t=0}^1 H(x + t\epsilon s) dt \right]$ and moreover, use H_+ to denote the psd part of H and H_- to denote the nsd part of H . Namely, $H = H_+ - H_-$.

We can write

$$\begin{aligned} \|g(x + \epsilon s) - g(x)\|_1 &= v^T (g(x + \epsilon s) - g(x)) = v^T H(\epsilon s) = \epsilon v^T H_+ s - \epsilon v^T H_- s \\ &= \epsilon \langle H_+^{1/2} v, H_+^{1/2} s \rangle - \epsilon \langle H_-^{1/2} v, H_-^{1/2} s \rangle \leq \epsilon \|H_+^{1/2} v\| \|H_+^{1/2} s\| + \epsilon \|H_-^{1/2} v\| \|H_-^{1/2} s\|. \end{aligned} \quad (8)$$

Note that assumption 2 implies the semidefinite ordering

$$H_+ \prec \text{diag}(\vec{L}) \text{ and } H_- \prec \text{diag}(\vec{L})$$

and thus $\max\{s^T H_+ s, s^T H_- s\} \leq \sum_{i=1}^d L_i = \|\vec{L}\|_1$ for all $s \in \{-1, 1\}^d$.

The proof is complete by observing that both v and s are sign vectors in (8). \square

Using the above lemma and the fact that our update rules are always following some sign vectors with learning rate smaller than δ , we have

$$\begin{aligned} \|g_{k-t} - g_k\|_1 &\leq \sum_{l=0}^{t-1} \|g_{k-l} - g_{k-l-1}\|_1 \\ &\leq 2\|\vec{L}\|_1 \sum_{l=0}^{t-1} \delta_{k-l-1} \leq \sum_{l=0}^{t-1} \frac{2\|\vec{L}\|_1 \delta}{\sqrt{k-l}} \\ &\leq 2\|\vec{L}\|_1 \delta \int_{k-t}^k \frac{dx}{\sqrt{x}} = 4\|\vec{L}\|_1 \delta (\sqrt{k} - \sqrt{k-t}) \\ &\leq 4\|\vec{L}\|_1 \delta \sqrt{k} \left(1 - \sqrt{1 - \frac{t}{k}}\right) \\ &\leq 4\|\vec{L}\|_1 \delta \frac{t}{\sqrt{k}} \quad \text{for } x \geq 0, 1 - x \leq \sqrt{1 - x} \end{aligned} \quad (9)$$

It follows that

$$\begin{aligned} \sum_i \mathbb{E}[|m_k[i] - g_k[i]|] &= \mathbb{E}[\|m_k - g_k\|_1] \\ &\leq 2(1 - \beta) \sum_{t=1}^k \beta^t \mathbb{E}\|g_{k-t} - g_k\|_1 \quad \text{Apply (7)} \\ &\leq \frac{8(1 - \beta)\|\vec{L}\|_1 \delta}{\sqrt{k}} \sum_{t=1}^{\infty} t \beta^t \quad \text{Apply (9) and extend sum to } \infty \\ &\leq \frac{8(1 - \beta)\|\vec{L}\|_1 \delta}{\sqrt{k}} \frac{\beta}{(1 - \beta)^2} \quad \text{derivative of geometric progression} \\ &\leq \frac{16\|\vec{L}\|_1 \delta}{\sqrt{k+1}} \frac{\beta}{1 - \beta} \quad \text{for } k \geq 1, \sqrt{\frac{k+1}{k}} \leq 2 \end{aligned} \quad (10)$$

Substitute (6) into (5), sum both sides over i and then further plug in (10) we get the statement in the lemma. \square

The proof of Theorem 3 now follows in a straightforward manner. Note that Lemma F.2 only kicks in after a warmup period of C iterations, with C as specified in Theorem 3. In theory it does not matter what you do during this warmup period, provided you accumulate the momentum as normal and take steps according to the prescribed learning rate and mini-batch schedules. One option is to just stay put and not update the parameter vector for the first C iterations. This is wasteful since no progress will be made on the objective. A better option in practice is to take steps using the sign of the stochastic gradient (i.e. do SIGNSGD) instead of SIGNUM during the warmup period.

Proof of Theorem 3. Substitute Lemma F.2 as $\xi(k)$ into Lemma E.1 and check that $\xi(k) = O(1/\sqrt{k})$, $\delta_k = O(1/\sqrt{k})$, $\min \delta_k = \delta/\sqrt{K}$, $C \ll K$, and in addition, we note that by the increasing minibatch size $N_K = O(K^2)$. Substitute $K = O(\sqrt{N})$ and take the square on both sides of the inequality. (We can take the \min_k out of the square since all the arguments are nonnegative and $(\cdot)^2$ is monotonic on \mathbb{R}_+). \square