# SIGNSGD: Compressed Optimisation for Non-Convex Problems

**Jeremy Bernstein** [1 2] **Yu-Xiang Wang** [2 3] **Kamyar Azizzadenesheli** [4] **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 $\ell_1/\ell_2$ 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., 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 1** SIGNSGD

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

**Algorithm 3** Distributed training by majority vote

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

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

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. 

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 $\ell_1$ 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 initisialiser, 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 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 explosion[1] 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.

*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 & Monro, 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 | $2Md$ |

**Deep learning:** stochastic gradient descent (Robbins & Monro, 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 $\langle . \rangle_\beta$ denote an exponential moving average with timescale $\beta$, 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, $\beta_1, \beta_2 \to 0$, 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

[1] For 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 $\ell_\infty$-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 $\ell_\infty$ majorisation. To the best of our knowledge, we are the first to study the convergence of signed gradient updates under an (often more natural) $\ell_2$ 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 assumptions 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 $f(.)$ evaluated at point $x$. Then $\forall x, y$ we require that for some non-negative constant $\vec{L} := [L_1, ..., L_d]$*

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

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:*

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

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

Bounded variance may be unrealistic in practice, since as $x \to \infty$ 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 $\sigma_i^2/n_k$.

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 $\ell_2$ Lipschitz constant $L := \|\vec{L}\|_\infty = \max_i L_i$. Then Assumption 2 implies that

$$\Big| f(y) - \big[f(x) + g(x)^T(y-x)\big] \Big| \leq \frac{L}{2}\|y_i - x_i\|_2^2.$$

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 $\sigma^2 := ||\vec{\sigma}||_2^2$. 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*

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

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

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

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 $\mathrm{O}(\sqrt{N})$ 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 $\ell_1$ geometry of SIGNSGD. The convergence rate strikingly depends on the $\ell_1$-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:

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

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 $\ell_1$ to both $\ell_2$ and $\ell_\infty$.

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

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

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

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

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

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

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

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

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) NOT$[R_1 \gg 1]$ and 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 $\|\vec{L}\|_\infty$. 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 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.

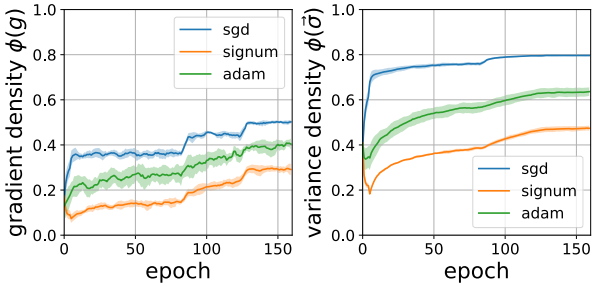


*Figure 1.* Gradient and noise density during an entire training run of a Resnet-20 model on the CIFAR-10 dataset. Results are averaged over 3 repeats for each of 3 different training algorithms, and corresponding error bars are plotted. At the beginning of every epoch, at that fixed point in parameter 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 $\phi(\vec{v}) := \frac{\|v\|_1^2}{d\|v\|_2^2}$ is 1 for a fully dense vector and $\approx 0$ for a fully sparse vector. Notice that both gradient and noise are dense, and moreover the densities appear to be coupled during training. Noticable 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.

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

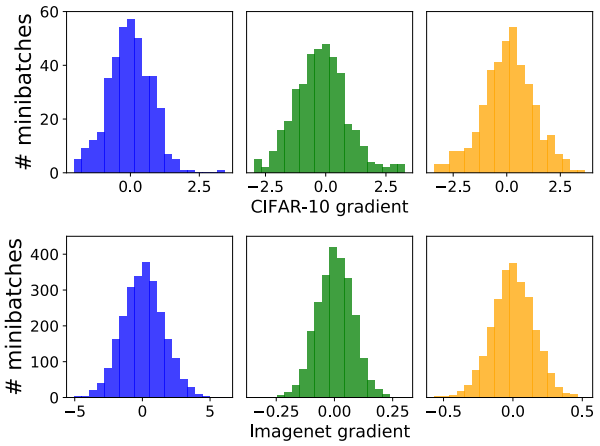


*Figure 2.* Histograms of the noise in the stochastic gradient, each plot for a different randomly chosen parameter (not cherry-picked). Top row: Resnet-20 architecture trained to epoch 50 on CIFAR-10 with a batch size of 128. Bottom row: Resnet-50 architecture trained to epoch 50 on Imagenet with a 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

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

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

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

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:

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

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 $\|\vec{\sigma}\|_1$ to $\|\vec{\sigma}\|_1/\sqrt{M}$.

**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*

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

*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:*

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

*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 $\mu$ 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*

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

*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 $\beta$ as follows:*

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

*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*

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

*where we have used $O(.)$ to hide numerical constants and the $\beta$-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.

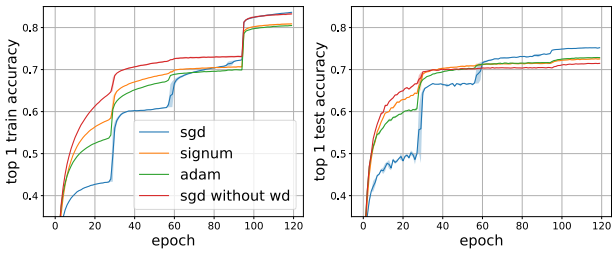


*Figure 3.* Imagenet train and test accuracies using the momentum version of SIGNSGD, called SIGNUM, to train Resnet-50 v2. We based our implementation on an open source implementation by `github.com/tornadomeet`. Initial learning rate and weight decay were tuned on a separate validation set split off 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 $\beta \to 1$ kills the variance term in $\|\vec{\sigma}\|_1$ 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 $\delta\|\vec{L}\|_1$ term.

**Remark 3:** for generality, we state this theorem with a tunable learning rate $\delta$. 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 $\ell_1$ 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.