CONVERGENCE RATE OF SIGN STOCHASTIC GRADIENT DESCENT FOR NON-CONVEX FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The sign stochastic gradient descent method (signSGD) utilizes only the sign of the stochastic gradient in its updates. Since signSGD carries out one-bit quantization of the gradients, it is extremely practical for distributed optimization where gradients need to be aggregated from different processors. For the first time, we establish convergence rates for signSGD on general non-convex functions under transparent conditions. We show that the rate of signSGD to reach first-order critical points matches that of SGD in terms of number of stochastic gradient calls, up to roughly a linear factor in the dimension. We carry out simple experiments to explore the behaviour of sign gradient descent (without the stochasticity) close to saddle points and show that it often helps completely avoid them without using either stochasticity or curvature information.

1 INTRODUCTION

Deep neural network training takes place in an error landscape that is high-dimensional, non-convex and stochastic. In practice, simple optimization techniques perform surprisingly well but have very limited theoretical understanding. While stochastic gradient descent (SGD) is widely used, algorithms like Adam (Kingma & Ba, 2015), RMSprop (Tieleman & Hinton, 2012) and Rprop (Riedmiller & Braun, 1993) are also popular. These latter algorithms involve component-wise rescaling of gradients, and so bear closer relation to signSGD than SGD. Currently, convergence rates have only been derived for close variants of SGD for general non-convex functions, and indeed the Adam paper gives convex theory.

Recently, another class of optimization algorithms has emerged which also pays attention to the resource requirements for training, in addition to obtaining good performance. Primarily, they focus on reducing costs for communicating gradients across different machines in a distributed training environment (Seide et al., 2014; Strom, 2015; Li et al., 2016; Alistarh et al., 2017; Wen et al., 2017). Often, the techniques involve quantizing the stochastic gradients at radically low numerical precision. Empirically, it was demonstrated that one can get away with using only one-bit per dimension without losing much accuracy (Seide et al., 2014; Strom, 2015). The theoretical properties of these approaches are however not well-understood. In particular, it was not known until now how quickly signSGD (the simplest incarnation of one-bit SGD) converges or even whether it converges at all to the neighborhood of a meaningful solution.

Our contribution: we supply the non-convex rate of convergence to first order critical points for signSGD. The algorithm updates parameter vector x_k according to

$$x_{k+1} = x_k - \delta_k \text{sign}(\bar{g}_k) \tag{1}$$

where $\bar{g}_k$ is the mini-batch stochastic gradient and δ_k is the learning rate. We show that for non-convex problems, signSGD entertains convergence rates as good as SGD, up to a linear factor in the dimension. Our statements impose a particular learning rate and mini-batch schedule.

Ours is the first work to provide non-convex convergence rates for a *biased* quantisation procedure as far as we know, and therefore does not require the randomisation that other gradient quantisation algorithms need to ensure unbiasedness. The technical challenge we overcome is in showing how to carry the stochasticity in the gradient through the sign non-linearity of the algorithm in a controlled-fashion.

Whilst our analysis is for first order critical points, we experimentally test the performance of sign gradient descent without stochasticity (signGD) around saddle points. We removed stochasticity in order to investigate whether signGD has an inherent ability to escape saddle points, which would suggest superiority over gradient descent (GD) which can take exponential time to escape saddle points if it gets too close to them (Du et al., 2017).

In our work we make three assumptions. Informally, we assume that the objective function is lower-bounded, smooth, and that the stochastic gradient has tails that vanish like a Gaussian distribution or faster. These assumptions are very general and hold for a wide class of functions, including the ones encountered in deep learning.

Outline of paper: in Section 3 we formally state our assumptions. In Section 4 we give our analysis of signSGD. In Section 5 we compare our results to SGD. In Section 6 we experimentally test the ability of the signGD (without the S) to escape saddle points.

2 Related work

Deep learning: the prototypical optimisation algorithm for neural networks is stochastic gradient descent (SGD)—see Algorithm 2. The deep learning community has discovered many practical tweaks to ease the training of large neural network models. In Rprop (Riedmiller & Braun, 1993) each weight update ignores the magnitude of the gradient and pays attention only to the sign, bringing it close to signSGD. It differs in that the learning rate for each component is modified depending on the consistency of the sign of consecutive steps. RMSprop (Tieleman & Hinton, 2012) is Rprop adapted for the minibatch setting—instead of dividing each component of the gradient by its magnitude, they estimate the rescaling factor as an average over recent iterates. Adam (Kingma & Ba, 2015) is RMSprop with momentum, meaning both gradient and gradient rescaling factors are estimated as bias-corrected averages over iterates. Indeed switching off the averaging in Adam yields signSGD. These algorithms have been applied to a breadth of interesting practical problems, e.g. (Xu et al., 2015; Gregor et al., 2015).

Some hope that having a more principled understanding of *why* deep learning works could lead to improved algorithms as well as a general feeling of satisfaction. To this end, Dauphin et al. (2014) frame the primary obstacle to neural network training as the proliferation of saddle points in high dimensional objectives. Practitioners challenge this view, suggesting that saddle points may be seldom encountered at least in retrospectively successful applications of deep learning (Goodfellow et al., 2015).

Optimisation theory: in convex optimisation there is a natural notion of success—rate of convergence to the global minimum x^*. Convex optimisation is eased by the fact that *local information* in the gradient provides *global information* about the direction towards the minimum, i.e. $\nabla f(x)$ tells you information about $x^* - x$.

In non-convex problems finding the global minimum is in general intractable, so theorists usually settle for measuring some restricted notion of success, such as rate of convergence to stationary points (e.g. Allen-Zhu (2017a)) or local minima (e.g. Nesterov & Polyak (2006)). Given the importance placed by Dauphin et al. (2014) upon evading saddle points, recent work considers the efficient use of noise (Jin et al., 2017; Levy, 2016) and curvature information (Allen-Zhu, 2017b) to escape saddle points and find local minima.

Distributed machine learning: whilst Rprop and Adam were proposed by asking how we can use gradient information to make better optimisation steps, another school asks how much information can we throw away from the gradient and still converge at all. Seide et al. (2014); Strom (2015) demonstrated empirically that one-bit quantisation can still give good performance whilst dramatically reducing gradient communication costs in distributed systems. Convergence of quantized stochastic gradient methods remains largely unknown. Alistarh et al. (2017) provide convergence rates for quantisation schemes that are unbiased estimators of the true gradient, and are thus able to rely upon vanilla SGD convergence results. Wen et al. (2017) prove asymptotic convergence of a $\{-1, 0, 1\}$ ternary quantization scheme that also retains the unbiasedness of the stochastic gradient. Our proposed approach is different, in that we directly employ the sign gradient which is *biased*. This avoids the randomization needed for constructing an unbiased quantized estimate. To

the best of our knowledge, the current work is the first to establish a convergence rate for a biased quantisation scheme, and our proof differs to that of vanilla SGD.

The fact that signSGD is related to both attempts to *improve* gradient descent like Rprop and Adam, and attempts to *damage it but not too badly* like quantised SGD is intriguing.

3 ASSUMPTIONS

Assumption 1 (The objective function is bounded below)**.** *For all x and some constant f^*, the objective function satisfies*

$$f(x) \geq f^* \tag{2}$$

Remark: this assumption applies to every practical objective function that we are aware of.

Assumption 2 (The objective function is L-Lipschitz smooth)**.** *Let $g(x)$ denote the gradient of the objective $f(.)$ evaluated at point x. Then for every y we assume that*

$$\Big| f(y) - \big[f(x) + g(x)^T (y - x) \big] \Big| \leq \frac{L}{2} \|y - x\|_2^2 \tag{3}$$

Remark: this assumption allows us to measure the error in trusting the local linearisation of our objective, which will be useful for bounding the error in a single step of the algorithm. For signSGD we can actually relax this assumption to only hold only for y within a local neighbourhood of x, since signSGD takes steps of bounded size.

Assumption 3 (The noise in the stochastic gradient has vanishing tails)**.** *In particular, we assume that the mini-batch stochastic gradient satisfies Hoeffding's inequality with variance-proxy σ. Consider a stochastic gradient estimate averaged over a mini-batch of size n.*

$$\bar{g}(x) = \frac{1}{n} \sum_{k=1}^{n} \textit{stochasticGradient}(x)$$

Let $\bar{g}_i(x)$ refer to the ith component of $\bar{g}(x)$. And let $g(x)$ refer to the true gradient at x. Then we assume that there exists a parameter σ such that the tails of the distribution of $\bar{g}_i(x)$ decay like

$$\mathbb{P}\Big[\Big| \bar{g}_i(x) - g_i(x) \Big| > t \Big] \leq 2 \exp\left(-\frac{nt^2}{2\sigma^2} \right) \tag{4}$$

Remark: this assumption encodes the natural idea that increasing mini-batch size should reduce variance in the stochastic gradient estimate. It holds for noise distributions that are Gaussian or sub-Gaussian.

4 NON-CONVEX CONVERGENCE RATE OF SIGNSGD

In this section we will state our theoretical results and provide intuition. Rigorous proof is deferred to the appendices. Before we dive into the theorems, here's a refresher on our notation:

g_k	the gradient at step k
f^*	the lower bound on the objective function
f_0	the initial value of the objective function
d	the dimension of the space
K	the total number of iterations
N_K	the cumulative number of stochastic gradient calls at step K
σ	the intrinsic variance-proxy for each component of the stochastic gradient
L	the maximum curvature (see Assumption 2)

Now we will state our primary result. Informally it says that if we run signSGD with the prescribed learning rate and mini-batch schedules, then after N stochastic gradient evaluations steps, we should

Algorithm 1 Sign stochastic gradient descent (signSGD)

1: **Inputs**: x_0, K ▷ initial point and time budget
2: **for** $k \in [0, K-1]$ **do**
3: $\delta_k \leftarrow \text{learningRate}(k)$
4: $n_k \leftarrow \text{miniBatchSize}(k)$
5: $\bar{g}_k \leftarrow \frac{1}{n_k}\sum_{i=1}^{n_k} \text{stochasticGradient}(x_k)$
6: $x_{k+1} \leftarrow x_k - \delta_k \text{sign}(\bar{g}_k)$ ▷ the sign operation is element-wise

expect that *somewhere along the optimisation trajectory* will be a place where the 1-norm of the gradient is smaller than O($N^{-0.25}$). Note that insofar as they can be compared, this matches the best non-convex rate for SGD, ignoring constants.

Theorem 1 (Non-convex convergence rate of signSGD)**.** *Apply Algorithm 1 under Assumptions 1, 2 and 3. Schedule the learning rate and mini-batch size as*

$$\delta_k = \frac{\delta}{\sqrt{k+1}} \qquad n_k = k+1 \tag{5}$$

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

$$\mathbb{E}\left[\min_{0\le k\le K-1} \|g_k\|_1\right]^2 \le \frac{1}{\sqrt{N_{K-2}}}\left[\frac{f_0 - f_*}{\delta} + 2d(2+\log(2N_{K-1}))(\sigma + \delta L)\right]^2 \tag{6}$$

The proof is deferred to Appendix A and here we sketch the intuition. Proving non-convex convergence rates to stationary points relies on the following observation: we should be able to relate the decrease in the objective function during one step of the algorithm to the size of the gradient. First consider the non-stochastic case: we know that if we take lots of steps for which the gradient is large, we will make lots of progress downhill. But since the objective function has a lower bound, it is impossible to keep taking large gradient steps downhill indefinitely, therefore increasing the number of steps requires that we must run into somewhere with small gradient.

To get a handle on this analytically, we must bound the per-step improvement in terms of the norm of the gradient. Assumption 2 allows us to do exactly this. Then we know that the sum of the per-step improvements over all steps must be smaller than the total possible improvement, and that gives us a bound on how large the minimum gradient that we see can be.

In the non-stochastic case, the obstacle to this process is curvature. Curvature means that if we take too large a step the gradient becomes unreliable, and we might move uphill instead of downhill. Since the step size in signSGD is set purely by the learning rate, this means we must anneal the learning rate if we wish to be sure to control the curvature-induced error and make good progress downhill.

Stochasticity also poses a problem in signSGD. In regions where the gradient signal is smaller than the noise, the noise is enough to flip the sign of the gradient. This is more severe than the additive noise in SGD, and so the batch size must be grown to control this effect.

You might expect that growing the batch size should lead to a worse convergence rate than SGD. This is forgetting that signSGD has an advantage in that it takes large steps even when the gradient is small. It turns out that this positive effect cancels out the fact that the batch size needs to grow, and the convergence rate ends up being the same as SGD.

For completeness, we also present the convergence rate for SGD derived under our assumptions. The proof is given in Appendix B. SGD does not require an increasing batch size since the effect of the noise is second order in the learning rate, and therefore gets squashed as the learning rate decays. The rate ends up being the same in N_K as signSGD because SGD makes slower progress when the gradient is small.

We should note finally that the learning rate and mini-batch schedules should be taken with a pinch of salt. In practice it is enough to merely get *close* to a stationary point and therefore a constant learning rate and mini-batch size may suffice without annealing/growth.

Algorithm 2 Stochastic gradient descent

1: **Inputs**: x_0, K, β ▷ initial point and time budget
2: **for** $k \in [0, K-1]$ **do**
3: $\delta_k \leftarrow \text{learningRate}(k)$
4: $n_k \leftarrow \text{miniBatchSize}(k)$
5: $\hat{g}_k \leftarrow \frac{1}{n_k}\sum_{i=1}^{n_k} \text{stochasticGradient}(x_k)$
6: $x_{k+1} \leftarrow x_k - \delta_k \bar{g}_k$

Theorem 2 (Non-convex convergence rate of stochastic gradient descent)**.** *Apply Algorithm 2 under Assumptions 1, 2 and 3. Schedule the learning rate and mini-batch size as*

$$\delta_k = \frac{\delta}{\sqrt{k+1}} \qquad n_k = 1 \tag{7}$$

Let N_K be the cumulative number of stochastic gradient calls up to step K, i.e. $N_K = K$. Then we have that

$$\mathbb{E}\left[\min_{0\le k\le K-1} \|g_k\|_2^2\right] \le \frac{1}{\sqrt{N_K}}\left[\frac{f_0 - f^*}{\delta\left(1-\frac{\delta L}{2}\right)} + d(1+\log N_K)\frac{2\delta L}{1-\frac{\delta L}{2}}\sigma^2\right] \tag{8}$$

Check Theorem 1 for the definitions of all the parameters. Note that this appears to be a classic result, although we are not sure of the earliest reference. Authors often hide the dimension dependence of the variance bound.

5 Comparing the convergence rate to SGD

To make a clean comparison, let us set $\delta = \frac{1}{L}$ (as is often recommended) and hide all numerical constants in Theorems 1 and 2. Then for signSGD, we get

$$\mathbb{E}\Big[\min\|g_k\|_1\Big]^2 \sim \frac{1}{\sqrt{N}}\Big[L(f_0 - f^*) + d(\sigma+1)\log N\Big]^2 \tag{9}$$

And for SGD we get

$$\mathbb{E}\Big[\min\|g_k\|_2^2\Big] \sim \frac{1}{\sqrt{N}}\Big[L(f_0 - f^*) + d\sigma^2\log N\Big] \tag{10}$$

What do these bounds mean? They say that after we have made a cumulative number of stochastic gradient evaluations N, that we should expect *somewhere along our trajectory* to have hit a point with gradient norm smaller than $N^{-\frac{1}{4}}$.

One important remark should be made. SignSGD more naturally deals with the one norm of the gradient vector, hence we had to square the bound to enable direct comparison with SGD[1]. This means that the constant factor in signSGD is roughly worse by a square. Paying attention only to dimension, this looks like

$$\text{signSGD: } \mathbb{E}\Big[\min\|g_k\|_1\Big]^2 \sim \frac{d^2}{\sqrt{N}} \qquad \text{SGD: } \mathbb{E}\Big[\min\|g_k\|_2^2\Big] \sim \frac{d}{\sqrt{N}} \tag{11}$$

This defect in dimensionality should be expected in the bound, since signSGD almost never takes the direction of steepest descent, and the direction only gets worse as dimensionality grows. This

[1]Technically we could use the fact that $\|g\|_1 \ge \|g\|_2$ and apply Jensen's inequality to the SGD bound, to bring them into the same form, but that loses information so we prefer not to.

raises the question, why do algorithms like Adam, which closely resemble signSGD, work well in practice? We suggest that it may be because they take advantage of additional structure in neural network landscapes that is not captured by our relatively straightforward assumptions. We will broach this subject more fully in the discussion section.

6 SWINGING BY SADDLE POINTS? AN EXPERIMENT

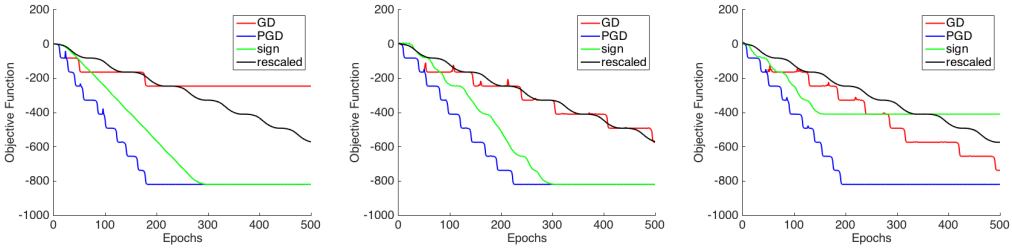

Figure 1: Descending the tube function of (Du et al., 2017). To reach the minimum, the algorithm must navigate a series of saddle points. Optimisers tested were gradient descent (GD), perturbed gradient descent (PGD) (Jin et al., 2017), sign gradient descent (signGD) and the rescaled gradient method (noiseless version of (Levy, 2016)). No learning rate tuning was attempted, so we suggest only focusing on the qualitative behaviour. Left: signGD appears not to 'see' the saddle points in the original objective function. Middle: after breaking the objective function's axis alignment by rotating it, the sign method's performance is still quantitatively different. Also the numerical error in our rotation operation appears to help unstick GD from the saddle points. Right: for some rotations, the sign method (with fixed learning rate and zero stochasticity) can get stuck in perfectly periodic orbits around saddle points.

Seeing as our theoretical analysis only deals with convergence to stationary points, it does not address how signSGD might behave around saddle points. We wanted to investigate the naïve intuition that gradient rescaling should help flee saddle points—or in the words of Zeyuan Allen-Zhu—swing by them.

For a testbed, the authors of (Du et al., 2017) kindly provided their 10-dimensional 'tube' function. The tube is a specially arranged gauntlet of saddle points, each with only one escape direction, that must be navigated in sequence before reaching the global minimum of the objective. The tube was designed to demonstrate how stochasticity can help escape saddles. Gradient descent takes much longer to navigate the tube than *perturbed gradient descent* of (Jin et al., 2017). It is interesting to ask, even empirically, whether the sign non-linearity in signSGD can also help escape saddle points efficiently. For this reason we strip out the stochasticity and pit the sign gradient descent method (signGD) against the tube function.

There are good reasons to expect that signGD might help escape saddles—for one, it takes large steps even when the gradient is small, which could drive the method away from regions of small gradient. For another, it is able to move in directions orthogonal to the gradient, which might help discover escape directions of the saddle. We phrase this as signGD's greater ability to *explore*.

Our experiments revealed that these intuitions sometimes hold out, but there are cases where they break down. In Figure 6, we compare the sign gradient method against gradient descent, perturbed gradient descent (Jin et al., 2017) and rescaled gradient descent $\left(x_{k+1} = x_k - \frac{g}{\|g\|_2}\right)$ which is a noiseless version of the algorithm considered in (Levy, 2016). No learning rate tuning was conducted, so we suggest paying attention to the qualitative behaviour rather than the ultimate convergence speed. The left hand plot pits the algorithms against the vanilla tube function. SignGD has very different qualitative behaviour to the other algorithms—it appears to make progress completely unimpeded by the saddles. We showed that this behaviour is partly due to the axis alignment of the tube function, since after randomly rotating the objective the behaviour changes (although it is still qualitatively different to the other algorithms).

One unexpected result was that for certain random rotations of the objective, signGD could get stuck at saddle points (see right panel in Figure 6). On closer inspection, we found that the algorithm was getting stuck in perfect periodic orbits around the saddle. Since the update is given by the learning rate multiplied by a binary vector, if the learning rate is constant it is perfectly possible for a sequence of updates to sum to zero. We expect that this behaviour relies on a remarkable structure in both the tube function and the algorithm. We hypothesise that for higher dimensional objectives and a non-fixed learning rate, this phenomenon might become extremely unlikely. This seems like a worthy direction of future research. Indeed we found empirically that introducing momentum into the update rule was enough to break the symmetry and avoid this periodic behaviour.

7 Discussion

First we wish to discuss the connections between signSGD and Adam (Kingma & Ba, 2015). Note that setting the Adam hyperparameters $\beta_1 = \beta_2 = \epsilon = 0$, Adam and signSGD are equivalent. Indeed the authors of the Adam paper suggest that during optimisation the Adam step will commonly look like a binary vector of ± 1 (multiplied by the learning rate) and thus resemble the sign gradient step. If this algorithmic correspondence is valid, then there seems to be a discrepancy between our theoretical results and the empirical good performance of Adam. Our convergence rates suggest that signSGD should be worse than SGD by roughly a factor of dimension d. In deep neural network applications d can easily be larger than 10^6. We suggest a resolution to this proposed discrepancy—there is structure present in deep neural network error surfaces that is not captured by our simplistic theoretical assumptions. For example it is reasonable to expect that neural network error surfaces might exhibit only weak coupling across dimensions. To provide intuition for how such an assumption can help improve the dimension scaling of signSGD, note that in the idealised case of total decoupling (the Hessian is everywhere diagonal) then the problem separates into d independent one dimensional problems, so the dimension dependence is lost.

Next, let's talk about saddle points. Though general non-convex functions are littered with local minima, recent work rather characterises successful optimisation as the evasion of a web of saddle points (Dauphin et al., 2014). Current theoretical work focuses either on using noise Levy (2016); Jin et al. (2017) or curvature information (Allen-Zhu, 2017b) to establish bounds on the amount of time needed to escape saddle points. We noted that merely passing the gradient through the sign operation introduces an algorithmic instability close to saddle points, and we wanted to empirically investigate whether this could be enough to escape them. We removed stochasticity from the algorithm to focus purely on the effect of the sign function.

We found that when the objective function was axis aligned, then sign gradient descent without stochasticity (signGD) made progress unhindered by the saddles. We suggest that this is because signGD has a greater ability to 'explore', meaning it typically takes larger steps in regions of small gradient than SGD, and it can take steps almost orthogonal to the true gradient direction. This exploration ability could potentially allow it to break out of subspaces convergent on saddle points without sacrificing its convergence rate—we hypothesise that this may contribute to the often superior practical performance of algorithms like Rprop and Adam, which bear closer relation to signSGD than SGD. For non axis-aligned objectives, signGD could sometimes get stuck in perfect periodic orbits around saddle points, though we hypothesise that this behaviour may be much less likely for higher dimensional objectives (the testbed function had dimension 10) with non-constant learning rate.

Finally we want to discuss the implications of our results for gradient quantisation schemes. Whilst we do not analyse the multi-machine case of distributed optimisation, we imagine that our results will extend naturally to that setting. In particular our results stand as a proof of concept that we can provide guarantees for *biased* gradient quantisation schemes. Existing quantisation schemes with guarantees require delicate randomisation to ensure unbiasedness. If a scheme as simple as ours can yield provable guarantees on convergence, then there is a hope that exploring further down this avenue can yield new and useful practical quantisation algorithms.

8 Conclusion

We have investigated the theoretical properties of the sign stochastic gradient method (signSGD) as an algorithm for non-convex optimisation. The study was motivated by links that the method has

both to deep learning stalwarts like Adam and Rprop, as well as to newer quantisation algorithms that intend to cheapen the cost of gradient communication in distributed machine learning.

We have proved non-convex convergence rates for signSGD to first order critical points. Insofar as the rates can directly be compared, they are of the same order as SGD in terms of number of gradient evaluations, but worse by a linear factor in dimension. SignSGD has the advantage over existing gradient quantisation schemes with provable guarantees, in that it doesn't need to employ randomisation tricks to remove bias from the quantised gradient.

We wish to propose some intriguing directions for future work. First our analysis only looks at convergence to first order critical points. Whilst we present preliminary experiments exhibiting success and failure modes of the algorithm around saddle points, a more detailed study attempting to pin down exactly when we can expect signSGD to escape saddle points efficiently would be welcome. This is an interesting direction seeing as existing work always relies on either stochasticity or second order curvature information to avoid saddles.

Second the link that signSGD has to both Adam-like algorithms and gradient quantisation schemes is enticing. In future work we intend to investigate whether this connection can be exploited to develop large scale machine learning algorithms that get the best of both worlds in terms of optimisation speed and communication efficiency.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Communication-Efficient Stochastic Gradient Descent, with Applications to Neural Networks. December 2017.

Zeyuan Allen-Zhu. Natasha: Faster Non-Convex Stochastic Optimization via Strongly Non-Convex Parameter. ICML, 2017a.

Zeyuan Allen-Zhu. Natasha 2: Faster Non-Convex Optimization Than SGD. *arXiv:1708.08694 [cs, math, stat]*, August 2017b. arXiv: 1708.08694.

Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and Attacking the Saddle Point Problem in High-dimensional Non-convex Optimization. In *NIPS*, 2014.

Simon S. Du, Chi Jin, Jason D. Lee, Michael I. Jordan, Barnabas Poczos, and Aarti Singh. Gradient Descent Can Take Exponential Time to Escape Saddle Points. *arXiv:1705.10412 [cs, math, stat]*, May 2017. arXiv: 1705.10412.

Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe. Qualitatively characterizing neural network optimization problems. *ICLR*, 2015. arXiv: 1412.6544.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. DRAW: A Recurrent Neural Network For Image Generation. ICML, 2015.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to Escape Saddle Points Efficiently. *ICML*, 2017. arXiv: 1703.00887.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ICLR*, 2015. arXiv: 1412.6980.

Kfir Y. Levy. The Power of Normalization: Faster Evasion of Saddle Points. *CoRR*, abs/1611.04831, 2016.

Mu Li, Ziqi Liu, Alexander J Smola, and Yu-Xiang Wang. Difacto: Distributed factorization machines. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 377–386. ACM, 2016.

Yurii Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, August 2006. ISSN 1436-4646. doi: 10.1007/s10107-006-0706-8.

M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *IEEE International Conference on Neural Networks*, pp. 586–591 vol.1, 1993. doi: 10.1109/ICNN.1993.298623.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-Bit Stochastic Gradient Descent and Application to Data-Parallel Distributed Training of Speech DNNs. In *Interspeech 2014*, September 2014.

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

Tijmen Tieleman and Geoffrey Hinton. RMSprop. *Coursera: Neural Networks for Machine Learning*, Lecture 6.5, 2012.

Ryota Tomioka and Milan Vojnovic. Qsgd: Communication-optimal stochastic gradient descent, with applications to training neural networks.

Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. ICML, 2015.

A PROVING THE CONVERGENCE RATE OF THE SIGN GRADIENT METHOD

Theorem 1 (Non-convex convergence rate of signSGD)**.** *Apply Algorithm 1 under Assumptions 1, 2 and 3. Schedule the learning rate and mini-batch size as*

$$\delta_k = \frac{\delta}{\sqrt{k+1}} \qquad\qquad n_k = k+1 \tag{5}$$

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

$$\mathbb{E}\left[\min_{0\leq k\leq K-1} \|g_k\|_1\right]^2 \leq \frac{1}{\sqrt{N_{K-2}}}\left[\frac{f_0 - f_*}{\delta} + 2d(2+\log(2N_{K-1}))(\sigma+\delta L)\right]^2 \tag{6}$$

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 one instantiation of the noise. Note that $\mathbb{I}[.]$ is the indicator function, and $g_{k,i}$ denotes the ith component of the vector g_k.

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

$$\begin{aligned}
f_{k+1} - f_k &\leq g_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|_2^2 \\
&= -\delta_k g_k^T \operatorname{sign}(\bar{g}_k) + \delta_k^2 \frac{L}{2} d \\
&= -\delta_k \|g_k\|_1 + 2\delta_k \sum_{i=1}^{d} |g_{k,i}|\, \mathbb{I}[\operatorname{sign}(\bar{g}_{k,i}) \neq \operatorname{sign}(g_{k,i})] + \delta_k^2 \frac{L}{2} d
\end{aligned}$$

The important observation is that when the true gradient is larger than the standard deviation of the noise, the sign will typically be correct. Mistakes will typically only be made when $|g_{k,i}|$ is smaller than σ. This allows us to bound the error by σ.

Let's make this intuition rigorous using Hoeffding's inequality (Assumption 3). Find the expected improvement at time $k+1$ conditioned on the previous iterates. Note that $\mathbb{P}[E]$ denotes the probability of event E.

$$\begin{aligned}
\mathbb{E}[f_{k+1} - f_k | x_k] &\leq -\delta_k \|g_k\|_1 + 2\delta_k \sum_{i=1}^{d} |g_{k,i}|\, \mathbb{P}[\operatorname{sign}(\bar{g}_{k,i}) \neq \operatorname{sign}(g_{k,i})] + \delta_k^2 \frac{L}{2} d \\
&\leq -\delta_k \|g_k\|_1 + 2\delta_k \sum_{i=1}^{d} |g_{k,i}|\, \mathbb{P}[|\bar{g}_{k,i} - g_{k,i}| > |g_{k,i}|] + \delta_k^2 \frac{L}{2} d \\
&\leq -\delta_k \|g_k\|_1 + 4\delta_k \sum_{i=1}^{d} |g_{k,i}| \exp\left(-\frac{n_k g_{k,i}^2}{2\sigma^2}\right) + \delta_k^2 \frac{L}{2} d
\end{aligned}$$

Now use the fact that $x\exp(-x^2) < \frac{1}{2}$, and our schedules on the learning rate and batch size to write

$$\begin{aligned}
\mathbb{E}[f_{k+1} - f_k | x_k] &\leq -\frac{\delta}{\sqrt{k+1}}\|g_k\|_1 + 2\sqrt{2}d\frac{\delta\sigma}{k+1} + \frac{\delta^2}{k+1}\frac{L}{2}d \\
&\leq -\frac{\delta}{\sqrt{K}}\|g_k\|_1 + \frac{4\delta d}{k+1}(\sigma + \delta L)
\end{aligned}$$

Now take the expectation over the noise in all previous iterates, and sum over k:

$$
\begin{aligned}
f_0 - f^* &\geq f_0 - \mathbb{E}[f_K] && \text{Assumption 1} \\
&= \mathbb{E}\left[\sum_{k=0}^{K-1} f_k - f_{k+1}\right] && \text{telescope} \\
&\geq \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{\delta}{\sqrt{K}}\|g_k\|_1 - \frac{4\delta d}{k+1}(\sigma + \delta L)\right] && \text{previous result} \\
&\geq \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{\delta d}{\sqrt{K}}\|g_k\|_1\right] - 4\delta d(1 + \log K)(\sigma + \delta L) && \text{harmonic sum}
\end{aligned}
$$

We can rearrange this inequality to yield a rate:

$$
\begin{aligned}
\mathbb{E}\left[\min_{0\leq k\leq K-1}\|g_k\|_1\right] &\leq \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{1}{K}\|g_k\|_1\right] \\
&\leq \frac{1}{\sqrt{K}}\left[\frac{f_0 - f_*}{\delta} + 4d(1 + \log K)(\sigma + \delta L)\right]
\end{aligned}
$$

Finally since we are growing our mini-batch size, it will take $N_{K-1} = \frac{K(K+1)}{2}$ gradient evaluations to reach step $K-1$. Using that $2N_{K-2} \leq K^2 \leq 2N_{K-1}$ yields the result.

□

B PROVING THE CONVERGENCE RATE OF STOCHASTIC GRADIENT DESCENT

Theorem 2 (Non-convex convergence rate of stochastic gradient descent)**.** *Apply Algorithm 2 under Assumptions 1, 2 and 3. Schedule the learning rate and mini-batch size as*

$$
\delta_k = \frac{\delta}{\sqrt{k+1}} \qquad n_k = 1 \tag{7}
$$

Let N_K be the cumulative number of stochastic gradient calls up to step K, i.e. $N_K = K$. Then we have that

$$
\mathbb{E}\left[\min_{0\leq k\leq K-1}\|g_k\|_2^2\right] \leq \frac{1}{\sqrt{N_K}}\left[\frac{f_0 - f^*}{\delta\left(1 - \frac{\delta L}{2}\right)} + d(1 + \log N_K)\frac{2\delta L}{1 - \frac{\delta L}{2}}\sigma^2\right] \tag{8}
$$

Check Theorem 1 for the definitions of all the parameters. Note that this appears to be a classic result, although we are not sure of the earliest reference. Authors often hide the dimension dependence of the variance bound.

Proof. Consider the objective improvement in a single step, under one instantiation of the noise. Use Assumption 2 followed by the definition of the algorithm.

$$
\begin{aligned}
f_{k+1} - f_k &\leq g_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|_2^2 \\
&= -\delta_k g_k^T \bar{g}_k + \delta_k^2 \frac{L}{2}\|\bar{g}_k\|_2^2
\end{aligned}
$$

Take the expectation conditioned on previous iterates, and decompose the mean squared stochastic gradient into its mean and variance. Technically since we are assuming Hoeffding's inequality, $4\sigma^2$

is actually an upper bound on the variance. Note that since $4\sigma^2$ is the variance bound for each component, the variance bound for the full vector will be $4d\sigma^2$.

$$\mathbb{E}[f_{k+1} - f_k | x_k] \leq -\delta_k \|g_k\|_2^2 + \delta_k^2 \frac{L}{2}\left(\|g_k\|_2^2 + 4d\sigma^2\right)$$

Plugging in the learning rate schedule, and using that $\frac{1}{k+1} \leq \frac{1}{\sqrt{k+1}}$, we get that

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

Sum over k and we get that

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

And rearranging yields the result.

□