Is Local SGD Better than Minibatch SGD?

Blake Woodworth [1] **Kumar Kshitij Patel** [1] **Sebastian U. Stich** [2] **Zhen Dai** [3] **Brian Bullins** [1]
H. Brendan McMahan [4] **Ohad Shamir** [5] **Nathan Srebro** [1]

Abstract

We study local SGD (also known as parallel SGD and federated averaging), a natural and frequently used stochastic distributed optimization method. Its theoretical foundations are currently lacking and we highlight how all existing error guarantees in the convex setting are dominated by a simple baseline, minibatch SGD. (1) For quadratic objectives we prove that local SGD strictly dominates minibatch SGD and that accelerated local SGD is minimax optimal for quadratics; (2) For general convex objectives we provide the first guarantee that at least *sometimes* improves over minibatch SGD; (3) We show that indeed local SGD does *not* dominate minibatch SGD by presenting a lower bound on the performance of local SGD that is worse than the minibatch SGD guarantee.

1. Introduction

It is often important to leverage parallelism in order to tackle large scale stochastic optimization problems. A prime example is the task of minimizing the loss of machine learning models with millions or billions of parameters over enormous training sets.

One popular distributed approach is local stochastic gradient descent (SGD) (Coppola, 2015; Stich, 2018; Zhou and Cong, 2018; Zinkevich et al., 2010), also known as "parallel SGD" or "Federated Averaging"[1] (McMahan et al., 2016), which is commonly applied to large scale convex and non-convex stochastic optimization problems, including in data center and "Federated Learning" settings (Kairouz et al., 2019). Local SGD uses M parallel workers which, in each of R rounds, independently execute K steps of SGD starting from a common iterate, and then communicate and average their iterates to obtain the common iterate from which the next round begins. Overall, each machine computes $T = KR$ stochastic gradients and executes KR SGD steps locally, for a total of $N = KRM$ overall stochastic gradients computed (and so $N = KRM$ samples used), with R rounds of communication (every K steps of computation).

Given the appeal and usage of local SGD, there is significant value in understanding its performance and limitations theoretically, and in comparing it to other alternatives and baselines *that have the same computation and communication structure*. That is, other methods that are distributed across M machines and compute K gradients per round of communication for R rounds, for a total of $T = KR$ gradients per machine and R communication steps. This structure can also be formalized through the graph oracle model of Woodworth et al. (2018, see also Section 2).

So, how does local SGD compare to other algorithms with the same computation and communication structure? Is local SGD (or perhaps an accelerated variant) optimal in the same way that (accelerated) SGD is optimal in the sequential setting? Is it better than baselines?

A natural alternative and baseline is minibatch SGD (Cotter et al., 2011; Dekel et al., 2012; Shamir and Srebro) – a simple method for which we have a complete and tight theoretical understanding. Within the same computation and communication structure, minibatch SGD can be implemented as follows: Each round, calculate the K stochastic gradient estimates (at the current iterate) on each machine, and then average all KM estimates to obtain a single gradient estimate. That is, we can implement minibatch SGD that takes R stochastic gradient steps, with each step using a minibatch of size KM—this is the fair and correct minibatch SGD to compare to, and when we refer to "minibatch SGD" we refer to this implementation (R steps with minibatch size KM).

Local SGD seems intuitively better than minibatch SGD, since even when the workers are not communicating, they

[1]Toyota Technological Institute at Chicago [2]EPFL [3]University of Chicago [4]Google [5]Weizmann Institute of Science. Correspondence to: Blake Woodworth <blake@ttic.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

[1]Federated Averaging is a specialization of local SGD to the federated setting, where (a) data is assumed to be heterogenous (not i.i.d.) across workers, (b) only a handful of clients are used in each round, and (c) updates are combined with a weighted average to accommodate unbalanced datasets.

are making progress towards the optimum. In particular, local SGD performs K times more updates over the course of optimization, and can be thought of as computing gradients at less "stale" and more "updated" iterates. For this reason, it has been argued that local SGD is at least as good as minibatch SGD, especially in convex settings where averaging iterates cannot hurt you. But can we capture this advantage theoretically to understand how and when local SGD is better than minibatch SGD? Or even just establish that local SGD is at least as good?

A string of recent papers have attempted to analyze local SGD for convex objectives, (e.g. Dieuleveut and Patel, 2019; Khaled et al., 2019; Stich, 2018; Stich and Karimireddy, 2019). However, a satisfying analysis has so far proven elusive. In fact, every analysis that we are aware of for local SGD in the general convex (or strongly convex) case with a typical noise scaling (e.g. as arising from supervised learning) not only does not improve over minibatch SGD, but is actually strictly dominated by minibatch SGD! But is this just a deficiency of these analyses, or is local SGD actually not better, and perhaps worse, than minibatch SGD? In this paper, we show that the answer to this question is "sometimes." There is a regime in which local SGD indeed matches or improves upon minibatch SGD, but perhaps surprisingly, there is also a regime in which local SGD really is strictly worse than minibatch SGD.

OUR CONTRIBUTIONS

In Section 3, we start with the special case of **quadratic** objectives and show that, at least in this case, **local SGD is strictly better than minibatch SGD** in the worst case, and that an accelerated variant is even **minimax optimal**.

We then turn to general **convex objectives**. In Section 4 we prove the **first error upper bound on the performance of local SGD which is *not* dominated by minibatch SGD's** upper bound with a typical noise scaling. In doing so, we identify a regime (where M is large and $K \gtrsim R$) in which local SGD performs strictly better than minibatch in the worst case. However, our upper bound does not show that local SGD is *always* as good or better than minibatch SGD. In Section 5, we show that this is not just a failure of our analysis. We prove a **lower bound on the worst-case error of local SGD that is *higher* than the worst-case error of minibatch SGD in a certain regime!** We demonstrate this behaviour empirically, using a logistic regression problem where local SGD indeed behaves much worse than mini-batch SGD in the theoretically-predicted problematic regime.

Thus, while local SGD is frequently better than minibatch SGD—and we can now see this both in theory and in practice (see experiments by e.g. Lin et al., 2018; Zhang et al., 2016; Zhou and Cong, 2018)—our work identifies regimes in which users should be wary of using local SGD without considering alternatives like minibatch SGD, and might want to seek alternative methods that combine the best of both, and attain optimal performance in all regimes.

2. Preliminaries

We consider the stochastic convex optimization problem:

$$\min_{x\in\mathbb{R}^d} F(x) := \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[f(x;z)]\,. \tag{1}$$

We will study distributed first-order algorithms that compute stochastic gradient estimates at a point $x \in \mathbb{R}^d$ via $\nabla f(x;z)$ based on indpendent samples $z \sim \mathcal{D}$. Our focus is on objectives F that are H-smooth, either (general) convex or λ-strongly convex[2], with a minimizer $x^* \in \arg\min_x F(x)$ with $\|x^*\| \le B$. We consider ∇f which has uniformly bounded variance, i.e. $\sup_x \mathbb{E}_{z\sim\mathcal{D}}\|\nabla f(x;z) - \nabla F(x)\|^2 \le \sigma^2$. We use $\mathcal{F}(H,\lambda,B,\sigma^2)$ to refer to the set of all pairs $(f,\mathcal{D})$ which satisfy these properties. All of the analysis in this paper can be done either for general convex or strongly convex functions, and we prove all of our results for both cases. For conciseness and clarity, when discussing the results in the main text, we will focus on the general convex case. However, the picture in the strongly convex case is mostly the same.

An important instance of (1) is a supervised learning problem where $f(x;z) = \ell(\langle x, \phi(z)\rangle, \mathit{label}(z))$ is the loss on a single sample. When $|\ell'|, |\ell''| \le 1$ (referring to derivatives w.r.t. the first argument), then $H \le |\ell''|\|\phi(z)\|^2 \le \|\phi(z)\|^2$ and also $\sigma^2 \le \|\nabla f\|^2 \le |\ell'|^2\|\phi(z)\|^2 \le \|\phi(z)\|^2$. Thus, assuming that the upper bounds on ℓ', ℓ'' are comparable, the relative scaling of parameters we consider as most "natural" is $H \approx \sigma^2$.

For simplicity, we consider initializing all algorithms at zero. Then, Local SGD with M machines, K stochastic gradients per round, and R rounds of communication calculates its tth iterate on the mth machine for $t \in [KR]$ via

$$x_t^m = \begin{cases} x_{t-1}^m - \eta\nabla f(x_{t-1}^m; z_{t-1}^m) & K \nmid t \\ \frac{1}{M}\sum_{m'=1}^M x_{t-1}^{m'} - \eta\nabla f(x_{t-1}^{m'}; z_{t-1}^{m'}) & K \mid t \end{cases} \tag{2}$$

where $z_t^m \sim \mathcal{D}$ i.i.d., and $K \mid t$ refers to K dividing t. For each $r \in [R]$, minibatch SGD calculates its rth iterate via

$$x_r = x_{r-1} - \frac{\eta}{MK}\sum_{i=1}^{MK}\nabla f(x_{r-1}; z_{r-1}^i) \tag{3}$$

We also introduce another strawman baseline, which we will refer to as "thumb-twiddling" SGD. In thumb-twiddling SGD, each machine computes just one (rather

[2] An H-smooth and λ-strongly convex function satisfies $\frac{\lambda}{2}\|x-y\|^2 \le F(y) - F(x) - \langle\nabla F(x), y-x\rangle \le \frac{H}{2}\|x-y\|^2$. We allow $\lambda = 0$ in which case F is general convex.

than K) stochastic gradients per round of communication and "twiddles its thumbs" for the remaining $K-1$ computational steps, resulting in R minibatch SGD steps, but with a minibatch size of only M (instead of KM, i.e. as if we used $K=1$). This is a silly algorithm that is clearly strictly worse than minibatch SGD, and we would certainly expect any reasonable algorithm to beat it. But as we shall see, previous work has actually struggled to show that local SGD even matches, let alone beats, thumb-twiddling SGD. In fact, we will show in Section 5 that, in certain regimes, local SGD truly is *worse* than thumb-twiddling.

For a particular algorithm A, we define its worst-case performance with respect to $\mathcal{F}(H,\lambda,B,\sigma^2)$ as:

$$\epsilon_{\mathsf{A}} = \max_{(f,\mathcal{D})\in\mathcal{F}(H,\lambda,B,\sigma^2)} F(\hat{x}_{\mathsf{A}}) - F(x^*) \tag{4}$$

The worst-case performance of minibatch SGD for general convex objectives is tightly understood (Dekel et al., 2012; Nemirovsky and Yudin, 1983):

$$\epsilon_{\text{MB-SGD}} = \Theta\left(\frac{HB^2}{R} + \frac{\sigma B}{\sqrt{MKR}}\right). \tag{5}$$

In order to know if an algorithm like local or minibatch SGD is "optimal" in the worst case requires understanding the minimax error, i.e. the best error that any algorithm with the requisite computation and communication structure can guarantee in the worst case. This requires formalizing the set of allowable algorithms. One possible formalization is the graph oracle model of Woodworth et al. (2018) which focuses on the dependence structure between different stochastic gradient computations resulting from the communication pattern. Using this method, Woodworth et al. prove lower bounds which are applicable to our setting. Minibatch SGD does not match these lower bounds (nor does accelerated minibatch SGD, see Cotter et al. (2011)), but these lower bounds are not known to be tight, so the minimax complexity and minimax optimal algorithm are not yet known.

Existing analysis of local SGD Table 1 summarizes the best existing analyses of local SGD that we are aware of that can be applied to our setting. We present the upper bounds as they would apply in our setting, and after optimizing over the stepsize and other parameters. A detailed derivation of these upper bounds from the explicitly-stated theorems in other papers is provided in Appendix A. As we can see from the table, in the natural scaling $H=\sigma^2$, every previous upper bound is strictly dominated by minibatch SGD. Worse, these upper bounds can even be worse than even thumb-twiddling SGD when $M \gg R$ (although they are sometimes better). In particular, the first term of each previous upper bound (in terms of M, K, R) is never better than R^{-1} (the optimization term of minibatch and thumb-twiddling SGD), and can be much worse.

Table 1. Comparison of existing analyses of Local SGD for general convex functions, with constant factors and low-order terms (in the natural scaling $H \approx \sigma^2$) omitted. We applied existing upper bounds as optimistically as possible, e.g. making additional assumptions where necessary to apply the guarantee to our setting, and our derivations are explained in Appendix A. The bolded term is the one which compares least favorably against minibatch SGD. Analogous rates for strongly convex functions are given in Appendix A.

Minibatch SGD	$\frac{HB^2}{R} + \frac{\sigma B}{\sqrt{MKR}}$
Thumb-twiddling SGD	$\frac{HB^2}{R} + \frac{\sigma B}{\sqrt{MR}}$
Stich (2018)	$\mathbf{\frac{HB^2}{R^{2/3}}} + \frac{HB^2}{(KR)^{3/5}} + \frac{\sigma B}{\sqrt{MKR}}$
Stich and Karimireddy (2019)	$\mathbf{\frac{HB^2M}{R}} + \frac{\sigma B}{\sqrt{MKR}}$
Khaled et al. (2019)[a]	$\mathbf{\frac{\sigma^2M}{HR}} + \frac{H^2B^2+\sigma^2}{H\sqrt{MKR}}$
Our upper bound (Section 4)	$\frac{\left(\mathbf{H\sigma^2B^4}\right)^{1/3}}{(\sqrt{\mathbf{KR}})^{2/3}} + \frac{HB^2}{KR} + \frac{\sigma B}{\sqrt{MKR}}$
Our lower bound (Section 5)	$\frac{\left(\mathbf{H\sigma^2B^4}\right)^{1/3}}{(\mathbf{KR})^{2/3}} + \frac{\sigma B}{\sqrt{MKR}}$

[a] This upper bound applies only when $M \le KR$. It also requires smoothness of each $f(x;z)$ individually, i.e. not just F.

We should note that in an extremely low noise regime $\sigma^2 \le H^2B^2\min\{\frac{1}{M}, \frac{K}{R}\}$, the bound of Khaled et al. (2019) can sometimes improve over minibatch SGD. However, this only happens when KR steps of sequential SGD is better than minibatch SGD—i.e. when you are better off ignoring $M-1$ of the machines and just doing serial SGD on a single machine (such an approach would have error $\frac{HB^2}{KR} + \frac{\sigma B}{\sqrt{KR}}$). This is a trivial regime in which every update for any of these algorithms is essentially an exact gradient descent step, thus there is no need for parallelism in the first place. See Appendix A.3 for further details. The upper bound we develop in Section 4, in contrast, dominates their guarantee and shows an improvement over minibatch that *cannot* be achieved on a single machine (i.e. without leveraging any parallelism). Furthermore, this improvement can occur even in the natural scaling $H=\sigma^2$ and even when minibatch SGD is better than serial SGD on one machine.

We emphasize that Table 1 lists the guarantees specialized to our setting—some of the bounds are presented under slightly weaker assumptions, or with a more detailed dependence on the noise: Haddadpour et al. (2019a); Stich

and Karimireddy (2019) analyze local SGD assuming not-quite-convexity; and Dieuleveut and Patel (2019); Wang and Joshi (2018) derive guarantees under both multiplicative and additive bounds on the noise. Dieuleveut and Patel (2019) analyze local SGD with the additional assumption of a bounded third derivative, but even with this assumption do not improve over mini-batch SGD. Numerous works study local SGD in the non-convex setting (see e.g. Haddadpour et al., 2019b; Stich and Karimireddy, 2019; Wang et al., 2017; Yu et al., 2019; Zhou and Cong, 2018). Although their bounds would apply in our convex setting, due to the much weaker assumptions they are understandably much worse than minibatch SGD. There is also a large body of work studying the special case $R = 1$, i.e. where the iterates are averaged just one time at the end (Godichon-Baggioni and Saadane, 2017; Jain et al., 2017; Li et al., 2014; Rosenblatt and Nadler, 2016; Zhang et al., 2012; Zinkevich et al., 2010). However, these analyses do not easily extend to multiple rounds, and the $R = 1$ constraint can provably harm performance (see Shamir et al., 2014). Finally, local SGD has been studied with heterogeneous data, i.e. where each machine receives stochastic gradients from different distributions—see Kairouz et al. (2019, Sec. 3.2) a recent survey.

An Alternative Viewpoint: Reducing Communication In this work, we focus on understanding the best achievable error for a given M, K, and R. However, one might also want to know to what extent it is possible to reduce communication without paying for it. Concretely, fix $T = KR$, and consider as a baseline an algorithm which computes T stochastic gradients on each machine sequentially, but is allowed to communicate after every step. We can then ask to what extent we can compete against this baseline while using less communication. One way to do this is to use Local SGD, which reduces communcation by a factor of K. However, the amount by which we can reduce communcation using Local SGD is easily determined once we know the error of Local SGD for each fixed K. Therefore, this viewpoint of reducing communcation is essentially equivalent to the one we take.

3. Good News: Quadratic Objectives

As we have seen, existing analyses of local SGD are no better than that of minibatch SGD. In the special case where F is quadratic, we will now show that not only is local SGD *sometimes* as good as minibatch SGD, but it is *always* as good as minibatch SGD, and sometimes better. In fact, an accelerated variant of local SGD is minimax optimal for quadratic objectives. More generally, we show that the local SGD anologue for a large family of serial first-order optimization algorithms enjoys an error guarantee which depends only on the product KR and not on K or R individually. In particular, we consider the following family of linear update algorithms:

Definition 1 (Linear update algorithm)**.** *We say that a first-order optimization algorithm is a linear update algorithm if, for fixed linear functions $\mathcal{L}_1^{(t)}, \mathcal{L}_2^{(t)}$, the algorithm generates its $t+1$st iterate according to*

$$x_{t+1} = \mathcal{L}_2^{(t)}\Big(x_1, \dots, x_t, \nabla f\Big(\mathcal{L}_1^{(t)}(x_1, \dots, x_t); z_t\Big)\Big) \quad (6)$$

This family captures many standard first-order methods including SGD, which corresponds to the linear mappings $\mathcal{L}_1^{(t)}(x_1, \dots, x_t) = x_t$ and $x_{t+1} = x_t - \eta_t \nabla f(x_t; z_t)$. Another notable algorithm in this class is AC-SA (Ghadimi and Lan, 2013), an accelerated variant of SGD which also has linear updates. Some important non-examples, however, are adaptive gradient methods like AdaGrad (Duchi et al., 2011; McMahan and Streeter, 2010)—these have linear updates, but the linear functions are data-dependent.

For a linear update algorithm $\mathcal{A}$, we will use local-$\mathcal{A}$ to denote the local SGD analogue with $\mathcal{A}$ replacing SGD. That is, during each round of communication, each machine independently executes K iterations of $\mathcal{A}$ and then the M resulting iterates are averaged. For quadratic objectives, we show that this approach inherits the guarantee of $\mathcal{A}$ with the benefit of variance reduction:

Theorem 1. *Let $\mathcal{A}$ be a linear update algorithm which, when executed for T iterations on any quadratic $(f, \mathcal{D}) \in \mathcal{F}(H, \lambda, B, \sigma^2)$, guarantees $\mathbb{E}F(x_T) - F^* \le \epsilon(T, \sigma^2)$. Then, local-$\mathcal{A}$'s averaged final iterate $\bar{x}_{KR} = \frac{1}{M}\sum_{m=1}^M x_{KR}^m$ will satisfy $\mathbb{E}F(\bar{x}_{KR}) - F^* \le \epsilon(KR, \frac{\sigma^2}{M})$.*

We prove this in Appendix B by showing that the average iterate $\bar{x}_t$ is updated according to $\mathcal{A}$—even in the middle of rounds of communication when $\bar{x}_t$ is not explicitly computed. In particular, we first show that

$$\bar{x}_{t+1} = \mathcal{L}_2^{(t)}\Bigg(\bar{x}_1, \dots, \bar{x}_t, \frac{1}{M}\sum_{m'=1}^M \nabla f\Big(\mathcal{L}_1^{(t)}\Big(x_1^{m'}, \dots, x_t^{m'}\Big); z_t^{m'}\Big)\Bigg) \quad (7)$$

Then, by the linearity of ∇F and $\mathcal{L}_1^{(t)}$, we prove

$$\mathbb{E}\left[\frac{1}{M}\sum_{m'=1}^M \nabla f\Big(\mathcal{L}_1^{(t)}\Big(x_1^{m'}, \dots, x_t^{m'}\Big); z_t^{m'}\Big)\right] = \nabla F\Big(\mathcal{L}_1^{(t)}(\bar{x}_1, \dots, \bar{x}_t)\Big) \quad (8)$$

and its variance is reduced to $\frac{\sigma^2}{M}$. Therefore, $\mathcal{A}$'s guarantee carries over while still benefitting from the lower variance.

To rephrase Theorem 1, on quadratic objectives, local-$\mathcal{A}$ is in some sense equivalent to KR iterations of $\mathcal{A}$ with the

gradient variance reduced by a factor of M. Furthermore, this guarantee depends only on the product KR, and not on K or R individually. Thus, averaging the Tth iterate of M independent executions of $\mathcal{A}$, sometimes called "one-shot averaging," enjoys the same error upper bound as T iterations of size-M minibatch-$\mathcal{A}$.

Nevertheless, it is important to highlight the boundaries of Theorem 1. Firstly, $\mathcal{A}$'s error guarantee $\epsilon(T, \sigma^2)$ must not rely on any particular structure of the stochastic gradients themselves, as this structure might not hold for the implicit updates of local-$\mathcal{A}$. Furthermore, even if some structure of the stochastic gradients *is* maintained for local-$\mathcal{A}$, the particular iterates generated by local-$\mathcal{A}$ will generally vary with K and R (even holding KR constant). Thus, Theorem 1 does *not* guarantee that local-$\mathcal{A}$ with two different values of K and R would perform the same on any particular instance. We have merely proven matching upper bounds on their worst-case performance.

We apply Theorem 1 to yield error upper bounds for local-SGD and local-AC-SA (based on the AC-SA algorithm of Ghadimi and Lan (2013)) which is minimax optimal:

Corollary 1. *For any quadratic $(f, \mathcal{D}) \in \mathcal{F}(H, \lambda = 0, B, \sigma^2)$, there are constants c_1 and c_2 such that local-SGD returns a point $\hat{x}$ such that*

$$\mathbb{E}F(\hat{x}) - F^* \leq c_1 \left(\frac{HB^2}{KR} + \frac{\sigma B}{\sqrt{MKR}} \right),$$

and local-AC-SA returns a point $\tilde{x}$ such that

$$\mathbb{E}F(\tilde{x}) - F^* \leq c_2 \left(\frac{HB^2}{K^2R^2} + \frac{\sigma B}{\sqrt{MKR}} \right).$$

In particular, local-AC-SA is minimax optimal for quadratic objectives.

Comparing the bound above for local SGD with the bound for minibatch SGD (5), we see that the local SGD bound is strictly better, due to the first term scaling as $(KR)^{-1}$ as opposed to R^{-1}. We note that minibatch SGD can also be accelerated (Cotter et al., 2011), leading to a bound with better dependence on R, but this is again outmatched by the bound for the (accelerated) local-AC-SA algorithm above. A similar, improved bound can also be proven when the objective is a strongly convex quadratic.

Prior Work in the Quadratic Setting Local SGD and related methods have been previously analyzed for quadratic objectives, but in slightly different settings. Jain et al. (2017) study a similar setting and analyze our "minibatch SGD" for $M = 1$ and fixed KR, but varying K and R. They show that when K is sufficiently small relative to R, then minibatch SGD can compete with KR steps of serial SGD. They also show that for fixed $M > 1$ and bT, when b is sufficiently small then the average of M independent runs of minibatch SGD with T steps and minibatch size b can compete with T steps of minibatch SGD with minibatch size Mb. These results are qualitatively similar to ours, but they analyze a specific algorithm while we are able to provide a guarantee for a broader class of algorithms. Dieuleveut and Patel (2019) analyze local SGD on quadratic objectives and show a result analogous to our Theorem 1. However, their result only holds when M is sufficiently small relative to K and R. Finally, there is a literature on "one-shot-averaging" for quadratic objectives, which corresponds to an extreme where the outputs of an algorithm applied to several different training sets are averaged, (e.g. Zhang et al., 2013a;b). These results also highlight similar phenomena, but they do not apply as broadly as Theorem 1 and they do not provide as much insight into local SGD specifically.

4. More Good News: General Convex Objectives

In this section, we present the first analysis of local SGD for general convex objectives that is not dominated by minibatch SGD. For the first time, we can identify a regime of M, K, and R in which local SGD provably performs better than minibatch SGD in the worst case. Furthermore, our analysis dominates all previous upper bounds.

Theorem 2. *Let $(f, \mathcal{D}) \in \mathcal{F}(H, \lambda, B, \sigma^2)$. When $\lambda = 0$, an appropriate average of the iterates of Local SGD with an optimally tuned constant stepsize satisfies for a universal constant c*

$$\mathbb{E}[F(\hat{x}) - F(x^*)] \leq c \cdot \min \left\{ \frac{HB^2}{KR} + \frac{\sigma B}{\sqrt{MKR}} + \frac{\left(H\sigma^2 B^4\right)^{\frac{1}{3}}}{K^{1/3}R^{2/3}}, \frac{HB^2}{KR} + \frac{\sigma B}{\sqrt{KR}} \right\}$$

If $\lambda > 0$, then an appropriate average of the iterates of Local SGD with decaying stepsizes satisfies for a universal constant c

$$\mathbb{E}[F(\hat{x}) - F(x^*)] \leq c \cdot \min \left\{ HB^2 \exp\left(-\frac{\lambda KR}{4H}\right) + \frac{\sigma^2}{\lambda MKR} + \frac{H\sigma^2 \log\left(9 + \frac{\lambda KR}{H}\right)}{\lambda^2 KR^2}, HB^2 \exp\left(-\frac{\lambda KR}{4H}\right) + \frac{\sigma^2}{\lambda KR} \right\}.$$

This is proven in Appendix C. We use a similar approach as Stich (2018), who analyzes the behavior of the averaged iterate $\bar{x}_t = \frac{1}{M}\sum_{m=1}^{M} x_t^m$, even when it is not ex-

plicitly computed. They show, in particular, that the averaged iterate evolves almost according to size-M-minibatch SGD updates, up to a term proportional to the dispersion of the individual machines' iterates $\frac{1}{M}\sum_{m=1}^{M}\|\bar{x}_t - x_t^m\|^2$. Stich bounds this with $O(\eta_t^2 K^2\sigma^2)$, but this bound is too pessimistic—in particular, it holds even if the gradients are replaced by arbitrary vectors of norm σ. In Lemma 5, we improve this bound to $O(\eta_t^2 K\sigma^2)$ which allows for our improved guarantee.[3] Our approach resembles that of Khaled et al. (2019), which we became aware of in the process of preparing this manuscript, however our analysis is more refined. In particular, we optimize more carefully over the stepsize so that our analysis applies for any M, K, and R (rather than just $M \leq KR$) and shows an improvement over minibatch SGD in a significantly broader regime, including when $\sigma^2 \gg 0$ (see Appendix A.3 for additional details).

Comparison of our bound with minibatch SGD We now compare the upper bound from Theorem 2 with the guarantee of minibatch SGD. For clarity, and in order to highlight the role of M, K, and R in the convergence rate, we will compare rates for general convex objectives when $H = B = \sigma^2 = 1$, and we will also ignore numerical constants and the logarithmic factor in Theorem 2. In this setting, the worst-case error of minibatch SGD is:

$$\epsilon_{\text{MB-SGD}} = \Theta\left(\frac{1}{R} + \frac{1}{\sqrt{MKR}}\right) \tag{9}$$

Our guarantee for local SGD from Theorem 2 reduces to:

$$\epsilon_{\text{L-SGD}} \leq O\left(\frac{1}{K^{\frac{1}{3}}R^{\frac{2}{3}}} + \frac{1}{\sqrt{MKR}}\right) \tag{10}$$

These guarantees have matching statistical terms of $\frac{1}{\sqrt{MKR}}$, which cannot be improved by any first-order algorithm (Nemirovsky and Yudin, 1983). Therefore, in the regime where the statistical term dominates both rates, i.e. $M^3K \lesssim R$ and $MK \lesssim R$, both algorithms will have similar worst-case performance. When we leave this noise-dominated regime, we see that local SGD's guarantee $K^{-\frac{1}{3}}R^{-\frac{2}{3}}$ is better than minibatch SGD's R^{-1} when $K \gtrsim R$ and is worse when $K \lesssim R$. This makes sense intuitively: minibatch SGD benefits from computing very precise gradient estimates, but pays for it by taking fewer gradient steps; conversely, each local SGD update is much noisier, but local SGD is able to make K times more updates.

[3] In recent work, Stich and Karimireddy (2019) present a new analysis of local-SGD which, in the general convex case is of the form $\frac{MHB^2}{R} + \frac{\sigma B}{\sqrt{MKR}}$. As stated, this is strictly worse than minibatch SGD. However, we suspect that this bound should hold *for any* $1 \leq M' \leq M$ because, intuitively, having more machines should not hurt you. If this is true, then optimizing their bound over M' yields a similar result as Theorem 2.

This establishes that for general convex objectives in the large-M and large-K regime, local SGD will strictly outperform minibatch SGD. However, in the large-M and small-K regime, we are only comparing upper bounds, so it is not clear that local SGD will in fact perform worse than minibatch SGD. Nevertheless, it raises the question of whether this is the best we can hope for from local SGD. Is local SGD truly better than minibatch SGD in some regimes but worse in others? Or, should we believe the intuitive argument suggesting that local SGD is always at least as good as minibatch SGD?

5. Bad News: Minibatch SGD Can Outperform Local SGD

In Section 3, we saw that when the objective is quadratic, local SGD is strictly better than minibatch SGD, and enjoys an error guarantee that depends only on KR and not K or R individually. In Section 4, we analyzed local SGD for general convex objectives and showed that local SGD *sometimes* outperforms minibatch SGD. However, we did not show that it *always* does, nor that it is always even competitive with minibatch SGD. We will now show that this is not simply a failure of our analysis—in a certain regime, local SGD really is inferior (in the worst-case) to minibatch SGD, and even to thumb-twiddling SGD. We show this by constructing a simple, smooth piecewise-quadratic objective in three dimensions, on which local SGD performs poorly. We define this hard instance $(f, \mathcal{D}) \in \mathcal{F}(H, \lambda, B, \sigma^2)$ as

$$\begin{aligned} f(x;z) = &\frac{\lambda}{2}\left(x_1 - \frac{B}{\sqrt{3}}\right)^2 + \frac{H}{2}\left(x_2 - \frac{B}{\sqrt{3}}\right)^2 \\ &+ \frac{H}{8}\left(\left(x_3 - \frac{B}{\sqrt{3}}\right)^2 + \left[x_3 - \frac{B}{\sqrt{3}}\right]_+^2\right) + zx_3 \end{aligned} \tag{11}$$

where $\mathbb{P}[z = \sigma] = \mathbb{P}[z = -\sigma] = \frac{1}{2}$ and $[y]_+ \equiv \max\{y, 0\}$.

Theorem 3. *For $0 \leq \lambda \leq \frac{H}{16}$, there exists $(f, \mathcal{D}) \in \mathcal{F}(H, \lambda, B, \sigma^2)$ such that for any $K \geq 2$ and $M, R \geq 1$, local SGD initialized at 0 with any fixed stepsize, will output a point $\hat{x}$ such that for a universal constant c*

$$\begin{aligned} &\mathbb{E}F(\hat{x}) - \min_x F(x) \\ &\quad\geq c \cdot \min\left\{\frac{H^{1/3}\sigma^{2/3}B^{4/3}}{K^{2/3}R^{2/3}}, \frac{H\sigma^2}{\lambda^2 K^2 R^2}, HB^2\right\} \\ &\quad\quad + c \cdot \min\left\{\frac{\sigma B}{\sqrt{MKR}}, \frac{\sigma^2}{\lambda MKR}\right\}. \end{aligned}$$

We defer a detailed proof of the Theorem to Appendix D. Intuitively, it relies on the fact that for non-quadratic functions, the SGD updates are no longer linear as in Section 3, and the local SGD dynamics introduce an additional bias

term which does not depend[4] on M, and scales poorly with K, R. In fact, this phenomenon is not unique to our construction, and can be expected to exist for any "sufficiently" non-quadratic function. With our construction, the proof proceeds by showing that the suboptimality is large unless $x_3 \approx \frac{B}{\sqrt{3}}$ but local SGD introduces a bias which causes x_3 to "drift" in the negative direction by an amount proportional to the stepsize. On the other hand, optimizing the first term of the objective requires the stepsize to be relatively large. Combining these yields the first term of the lower bound. The second term is classical and holds even for first-order algorithms that compute MKR stochastic gradients sequentially (Nemirovsky and Yudin, 1983).

In order to compare this lower bound with Theorem 2 and with minibatch SGD, we again consider the general convex setting with $H = B = \sigma^2 = 1$. Then, the lower bound reduces to $K^{-\frac{2}{3}}R^{-\frac{2}{3}} + (MKR)^{-\frac{1}{2}}$. Comparing this to Theorem 2, we see that our upper bound is tight up to a factor of $K^{-\frac{1}{3}}$ in the optimization term. Furthermore, comparing this to the worst-case error of minibatch SGD (9), we see that local SGD is indeed worse than minibatch SGD in the worst case when K is small enough relative to R. The cross-over point is somewhere between $K \leq \sqrt{R}$ and $K \leq R$; for smaller K, minibatch SGD is better than local SGD in the worst case, for larger K, local SGD is better in the worst case. Since the optimization terms of minibatch SGD and thumb-twiddling SGD are identical, this further indicates that local SGD is even outperformed by thumb-twiddling SGD in the small K and large M regime.

Finally, it is interesting to note that in the *strongly convex* case (where $\lambda > 0$), the gap between local GD and minibatch SGD can be even more dramatic: In that case, the optimization term of minibatch SGD scales as $\exp(-R)$ (see Stich (2019) and references therein), while our theorem implies that local SGD cannot obtain a term better than $(KR)^{-2}$. This implies an exponentially worse dependence on R in that term, and a worse bound as long as $R \gtrsim \log(K)$.

In order to prove Theorem 3 we constructed an artificial, but easily analyzable, situation where we could prove analytically that local SGD is worse than mini-batch. In Figure 1, we also demonstrate the behaviour empirically on a logistic regression task, by plotting the suboptimality of local SGD, minibatch SGD, and thumb-twiddling SGD iterates with optimally tuned stepsizes. As is predicted by Theorem 3, we see local SGD goes from performing worse than minibatch in the small $K = 5$ regime, but improving relative to the other algorithms as K increases to 40 and then 200, when local SGD is far superior to minibatch. For each fixed K, increasing M causes thumb-twiddling SGD to improve relative to minibatch SGD, but does not have a significant effect on local SGD, which is consistent with introducing a bias which depends on K but not on M. This highlights that the "problematic regime" for local SGD is where there are few iterations per round.

[4]To see this, consider for example the univariate function $f(x; z) = x^2 + [x]_+^2 + zx$ where z is some zero-mean bounded random variable. It is easy to verify that even if we have infinitely many machines ($M = \infty$), running local SGD for a few iterations starting from the global minimum $x = 0$ of $F(x) := \mathbb{E}_z[f(x; z)]$ will generally return a point bounded away from 0. In contrast, minibatch SGD under the same conditions will remain at 0.

6. Future work

In this paper, we provided the first analysis of local SGD showing improvement over minibatch SGD in a natural setting, but also demonstrated that local SGD can sometimes be worse than minibatch SGD, and is certainly not optimal.

As can be seen from Table 1, our upper and lower bounds for local SGD are still not tight. The first term depends on $K^{1/3}$ versus $K^{2/3}$—we believe the correct behaviour might be in between, namely $\sqrt{K}$, matching the bias of K-step SGD. The exact worst case behaviour of local SGD is therefore not yet resolved.

But beyond obtaining a precise analysis of local SGD, our paper highlights a more important challenge: we see that local SGD is definitely *not* optimal, and does not even always improve over minibatch SGD. Can we suggest an optimal algorithm in this setting? Or at least a method that combines the advantages of both local SGD and minibatch SGD and enjoys guarantees that dominate both? Our work motivates developing such an algorithm, which might also have benefits in regimes where local SGD is already better than minibatch SGD.

To answer this question will require new upper bounds and perhaps also new lower bounds. Looking to the analysis of local AC-SA for quadratic objectives in Corollary 1, we might hope to design an algorithm which achieves error

$$\mathbb{E}F(\hat{x}) - F(x^*) \leq O\left(\frac{HB^2}{(KR)^2} + \frac{\sigma B}{\sqrt{MKR}}\right) \qquad (12)$$

for general convex objectives. That is, an algorithm which combines the optimization term for KR steps of accelerated gradient descent with the optimal statistical term. If this were possible, it would match the lower bound of Woodworth et al. (2018) and therefore be optimal with respect to this communication structure.

Acknowledgements This work is partially supported by NSF-CCF/BSF award 1718970/2016741, NSF-DMS 1547396, and a Google Faculty Research Award. BW is supported by a Google PhD Fellowship. Part of this work was done while NS was visiting Google. Work by SS was done while visiting TTIC.

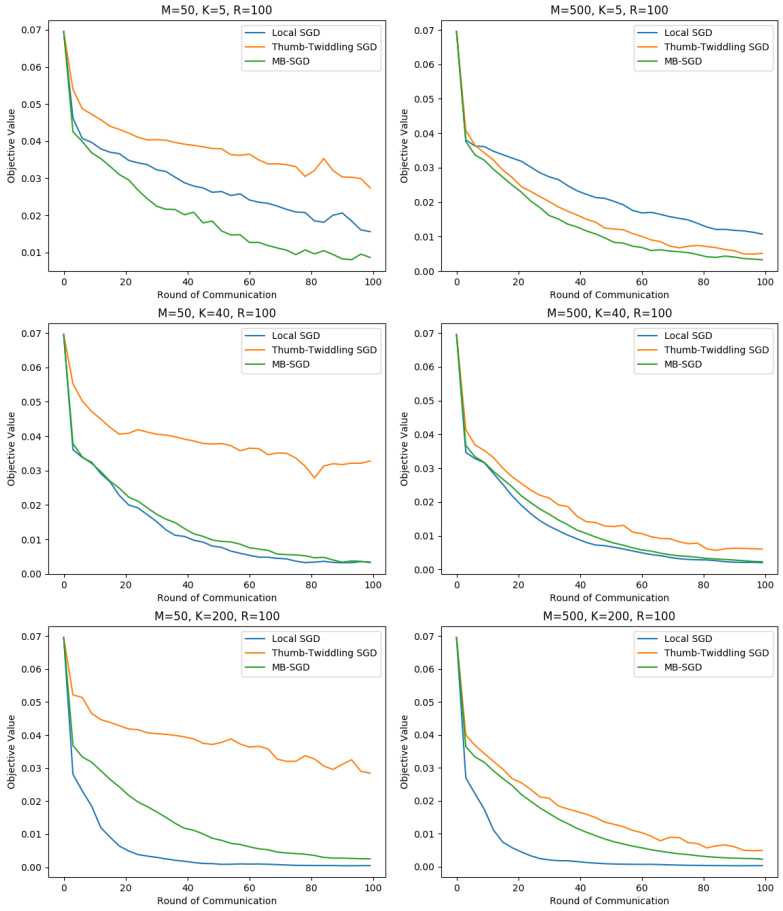

Figure 1. We constructed a dataset of 50000 points in $\mathbb{R}^{25}$ with the ith coordinate of each point distributed independently according to a Gaussian distribution $\mathcal{N}(0, \frac{10}{i^2})$. The labels are generated via $\mathbb{P}[y = 1 \,|\, x] = \sigma(\min\{\langle w_1^*, x\rangle + b_1^*, \langle w_2^*, x\rangle + b_2^*\})$ for $w_1^*, w_2^* \sim \mathcal{N}(0, I_{25\times 25})$ and $b_1^*, b_2^* \sim \mathcal{N}(0, 1)$, where $\sigma(a) = 1/(1+\exp(-a))$ is the sigmoid function, i.e. the labels correspond to an intersection of two halfspaces with label noise which increases as one approaches the decision boundary. We used each algorithm to train a linear model with a bias term to minimize the logistic loss over the 50000 points, i.e. f is the logistic loss on one sample and $\mathcal{D}$ is the empirical distribution over the 50000 samples. For each M, K, and algorithm, we tuned the constant stepsize to minimize the loss after r rounds of communication individually for each $1 \le r \le R$. Let $x_{\mathsf{A},r,\eta}$ denote algorithm A's iterate after the rth round of communication when using constant stepsize η. The plotted lines are an approximation of $g_{\mathsf{A}}(r) = \min_\eta F(x_{\mathsf{A},r,\eta}) - F(x^*)$ for each A where the minimum is calculated using grid search on a log scale.

References

Greg Coppola. *Iterative parameter mixing for distributed large-margin training of structured predictors for natural language processing*. PhD thesis, The University of Edinburgh, 2015.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1647–1655. Curran Associates, Inc., 2011.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13 (Jan):165–202, 2012.

Aymeric Dieuleveut and Kumar Kshitij Patel. Communication trade-offs for local-sgd with large step size. In *Advances in Neural Information Processing Systems*, pages 13579–13590, 2019.

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

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and op-

timal algorithms. *SIAM Journal on Optimization*, 23(4): 2061–2089, 2013.

Antoine Godichon-Baggioni and Sofiane Saadane. On the rates of convergence of parallelized averaged stochastic gradient algorithms. *arXiv preprint arXiv:1710.07926*, 2017.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems*, pages 11080–11092, 2019a.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization. In *International Conference on Machine Learning*, pages 2545–2554, 2019b.

Prateek Jain, Praneeth Netrapalli, Sham M Kakade, Rahul Kidambi, and Aaron Sidford. Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification. *The Journal of Machine Learning Research*, 18(1):8258–8299, 2017.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurlien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adri Gascn, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konen, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrde Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer zgr, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Better communication complexity for local sgd. *arXiv preprint arXiv:1909.04746*, 2019.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–670. ACM, 2014.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

H. Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 244–256, 2010. URL http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=252.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Jonathan D Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404, 2016.

Ohad Shamir and Nathan Srebro. Distributed stochastic optimization and learning. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 850–857. IEEE.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008, 2014.

Max Simchowitz. On the randomized complexity of minimizing a convex quadratic function. *arXiv preprint arXiv:1807.09386*, 2018.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018. URL https://arxiv.org/abs/1805.09767.

Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Lieven Vandenberghe. Lecture notes 1 for optimization methods for large-scale systems, 2019.

Jialei Wang, Weiran Wang, and Nathan Srebro. Memory and communication efficient distributed stochastic optimization with minibatch-prox. *arXiv preprint arXiv:1702.06269*, 2017. URL https://arxiv.org/abs/1702.06269.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

Blake Woodworth, Jialei Wang, Brendan McMahan, and Nathan Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. *arXiv preprint arXiv:1805.10222*, 2018. URL https://arxiv.org/abs/1805.10222.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Parallel sgd: When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016.

Yuchen Zhang, Martin J Wainwright, and John C Duchi. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510, 2012.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *Conference on learning theory*, pages 592–617, 2013a.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research*, 14(1):3321–3363, 2013b.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3219–3227. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/447. URL https://doi.org/10.24963/ijcai.2018/447.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pages 2595–2603, 2010.