LOCAL SGD CONVERGES FAST AND COMMUNICATES LITTLE

Anonymous authors
Paper under double-blind review

ABSTRACT

Mini-batch stochastic gradient descent (SGD) is state of the art in large scale distributed training. The scheme can reach a linear speedup with respect to the number of workers, but this is rarely seen in practice as the scheme often suffers from large network delays and bandwidth limits. To overcome this communication bottleneck recent works propose to reduce the communication frequency. An algorithm of this type is local SGD that runs SGD independently in parallel on different workers and averages the sequences only once in a while. This scheme shows promising results in practice, but eluded thorough theoretical analysis.

We prove concise convergence rates for local SGD on convex problems and show that it converges at the same rate as mini-batch SGD in terms of number of evaluated gradients, that is, the scheme achieves linear speedup in the number of workers and mini-batch size. The number of communication rounds can be reduced up to a factor of $T^{3/2}$—where $T$ denotes the number of total steps—compared to mini-batch SGD. This also holds for asynchronous implementations.

Local SGD can also be used for large scale training of deep learning models. The results shown here aim serving as a guideline to further explore the theoretical and practical aspects of local SGD in these applications.

1 INTRODUCTION

Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) consists of iterations of the form

$$x_{t+1} := x_t - \eta_t g_t,$$

for iterates (or weights) $x_t, x_{t+1} \in \mathbb{R}^d$, stepsize (or learning rate) $\eta_t > 0$, and stochastic gradient $g_t$ with the property $\mathbb{E} g_t = \nabla f(x_t)$, for a loss function $f: \mathbb{R}^d \to \mathbb{R}$. This scheme can easily be parallelized by replacing $g_t$ in (1) by an average of stochastic gradients that are independently computed in parallel on separate workers (parallel SGD). This simple scheme has a major drawback: in each iteration the results of the computations on the workers have to be shared with the other workers to compute the next iterate $x_{t+1}$. Communication has been reported to be a major bottleneck for many large scale deep learning applications, see e.g. (Seide et al., 2014; Alistarh et al., 2017; Zhang et al., 2017; Lin et al., 2018). Mini-batch parallel SGD addresses this issue by increasing the compute to communication ratio. Each worker computes a mini-batch of size $b$ before communication. This scheme is implemented in state-of-the-art distributed deep learning frameworks (Abadi et al., 2016; Paszke et al., 2017; Seide & Agarwal, 2016). Recent work in (You et al., 2017; Goyal et al., 2017) explores various limitations of this approach, as in general it is reported that performance degrades for too large mini-batch sizes (Keskar et al., 2016; Ma et al., 2018).

In this work we follow an orthogonal approach, still with the goal to increase the compute to communication ratio: Instead of increasing the mini-batch size, we reduce the communication frequency. Rather than keeping the sequences on different machines in sync, we allow them to evolve locally on each machine, independent from each other, and only average the sequences once in a while (local SGD). Such strategies have been explored widely in the literature, under various names.

An extreme instance of this concept is one shot SGD (Mcdonald et al., 2009; Zinkevich et al., 2010) where the local sequences are only exchanged once, after the local runs have converged. Zhang et al. (2013) show statistical convergence (see also Godichon-Baggioni & Saadane (2017)), but the
We consider finite-sum convex optimization problems $f: \mathbb{R}^d \rightarrow \mathbb{R}$ of the form

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad x^* := \arg \min_{x \in \mathbb{R}^d} f(x), \quad f^* := f(x^*),$$

where $f$ is $L$-smooth and $\mu$-strongly convex. We consider $K$ parallel mini-batch SGD sequences with mini-batch size $b$ that are synchronized after at most every $H$ iterations. For appropriate chosen stepsizes and an averaged iterate $x_T$ after $T$ steps (for $T$ sufficiently large, see Section 3 below) and synchronization delay $H = O(\sqrt{T/(Kb)})$ we show convergence

$$\mathbb{E} f(x_T) - f^* = O\left(\frac{G^2}{\mu KT}\right),$$

with second moment bound $G^2 \geq \mathbb{E} \|\nabla f_i(x)\|^2$. Thus, we see that compared to parallel mini-batch SGD the communication rounds can be reduced by a factor $H = O(\sqrt{T/(Kb)})$ without hampering the asymptotic convergence. Equation (3) shows perfect linear speedup in terms of computation, but with much less communication that mini-batch SGD. The resulting speedup when taking communication cost into account is illustrated in Figure 1 (see also Section D below). To improve the compute to communication ratio (denoted by $\rho$) one has thus now two strategies: (i) either to increase the mini-batch size $b$ or (ii) to increase the communication interval $H$. Both strategies give the same improvement when $b$ and $H$ are small (linear speedup). Similar to mini-batch SGD that faces some limitations for $b \gg 1$ (as discussed in e.g. [Ma et al., 2018]), the parameter $H$ cannot be chosen too large in local SGD. We give some practical guidelines in Section 4.

Our proof is simple and straightforward, and we imagine that—with slight modifications of the proof—the technique can also be used to analyze other variants of SGD that evolve sequences on different worker that are not perfectly synchronized. Although we do not yet provide convergence guarantees for the non-convex setting we feel that the positive results presented here will spark further investigation of local SGD for this important application (see e.g. [Yu et al., 2018]).

---

1 On convex functions, the average of the $K$ local solutions can of course only decrease the objective value, but convexity does not imply that the averaged point is $K$ times better.

2 $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$.

3 $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$. 

---

Figure 1: Illustration of the speedup (3) for time-to-accuracy when either increasing mini-batch size $b$ (1 $\rightarrow$ 2) or communication interval $H$ (1 $\rightarrow$ 2), for compute to communication ratio $\rho = 25$.
1.2 Related Work

A parallel line of work reduces the communication cost by compressing the stochastic gradients before communication. For instance by limiting the number of bits in the floating point representation (Gupta et al., 2013), Na et al. (2017) or random quantization (Alistarh et al., 2017, Wen et al., 2017). The ZipML framework applies this technique also to the data (Zhang et al., 2017). Sparsification methods reduce the number of non-zero entries in the stochastic gradient (Alistarh et al., 2017, Wangni et al., 2017). A very aggressive—and promising—sparsification method is to keep only very few coordinates of the stochastic gradient by considering only the coordinates with the largest magnitudes (Seide et al., 2014, Strom, 2015, Dryden et al., 2016, Aji & Heafield, 2017, Sun et al., 2017, Lin et al., 2018b, Stich et al., 2018).

Allowing asynchronous updates provides an alternative solution to disguise the communication overhead to a certain amount (Niu et al., 2011, Sa et al., 2015, Lian et al., 2015), though alternative strategies might be better when high accuracy is desired (Chen et al., 2016). The analysis of Agarwal & Duchi (2011) shows that asynchronous SGD on convex functions can tolerated delays up to $O(\sqrt{T/K})$, which is identical to the maximal length of the local sequences in local SGD. Asynchronous SGD converges also for larger delays (see also Zhou et al., 2018) but without linear speedup, a similar statement holds for local SGD (see discussion in Section 3). The current frameworks for the analysis of asynchronous SGD do not cover local SGD. A fundamental difference is that asynchronous SGD maintains a (almost) synchronized sequence and gradients are computed with respect this unique sequence (but just applied with delays), whereas each worker in local SGD evolves a different sequence and computes gradient with respect these ite- rate.

For the training of deep neural networks, Bijral et al. (2016) discuss a stochastic averaging schedule whereas Zhang et al. (2016) study local SGD with more frequent communication at the beginning of the optimization process. The elastic averaging technique (Zhang et al., 2015) is different to local SGD, as it uses the average of the iterates only to guide the local sequences, but does not perform a hard reset after averaging. Three recent preprints discuss local SGD for non-convex settings: Lin (2018a) gives an empirical analysis whereas Yu et al. (2018) and Wang & Joshi (2018) give theoretically analyses. Yu et al. (2018) show linear speedup of local SGD on non-convex functions for $H = O(T^{1/4}K^{-3/4})$, which is more restrictive than the constraint on $H$ in the convex setting.

1.3 Outline

We formally introduce local SGD in Section 2 and sketch the convergence proof in Section 3. In Section 4 show numerical results to illustrate the result. We analyze asynchronous local SGD in Section 5. The proof of the technical results, further discussion about the experimental setup and implementation guidelines are deferred to the appendix.

2 Local SGD

The algorithm local SGD (depicted in Algorithm 1) generates in parallel $K$ sequences $\{x^k_t\}_{t=0}^T$ of iterates, $k \in [K]$. Here $K$ denotes the level of parallelization, i.e. the number of distinct parallel sequences and $T$ the number of steps (i.e. the total number of stochastic gradient evaluations is $TK$).

Let $I_T \subseteq [T]$ with $T \in I_T$ denote a set of synchronization indices. Then local SGD evolves the sequences $\{x^k_t\}_{t=0}^T$ in the following way:

$$x^k_{t+1} := \begin{cases} x^k_t - \eta_t \nabla f_i^k (x^k_t) , & \text{if } t+1 \notin I_T \\ \frac{1}{K} \sum_{k=1}^K (x^k_t - \eta_t \nabla f_i^k (x^k_t)) & \text{if } t+1 \in I_T \end{cases}$$

(4)

where indices $i^k_t \sim_{u.a.r.} [n]$ and $\{\eta_t\}_{t \geq 0}$ denotes a sequence of stepsizes. If $I_T = [T]$ then the synchronization of the sequences is performed every iteration. In this case, (4) amounts to parallel or mini-batch SGD with mini-batch size $K$. On the other extreme, if $I_T = \{T\}$, the synchronization only happens at the end, which is known as one-shot averaging.

---

4For the ease of presentation, we assume here that each worker in local SGD only processes a mini-batch of size $b = 1$. This can be done without loss of generality, as we discuss later in Remark 2.3.
We were not especially careful to optimize the constants (and the lower order terms) in (5) Algorithm 1

**Theorem 2.2.** This indicates that one has to show that local SGD decreases the variance \( \sigma \). For local SGD such a simple argument is elusive. For instance, just capitalizing the convexity of the objective function \( f \) and for stepsizes \( \eta \), SGD converges at rate \( O\left(\frac{\sigma^2}{T}\right) \), i.e. achieves a linear speedup.

**Definition 2.1 (gap).** The gap of a set \( \mathcal{P} := \{p_0, \ldots, p_t\} \) of \( t + 1 \) integers, \( p_i \leq p_{i+1} \) for \( i = 0, \ldots, t - 1 \), is defined as \( \text{gap}(\mathcal{P}) := \max_{i=1,...,t}(p_i - p_{i-1}) \).

### 2.1 Variance Reduction in Local SGD

Before jumping to the convergence result, we first discuss an important observation.

**Parallel (mini-batch) SGD.** For carefully chosen stepsizes \( \eta \), SGD converges at rate \( O\left(\frac{\sigma^2}{T}\right) \) on strongly convex and smooth functions \( f \), where \( \sigma^2 \geq \mathbb{E}\left[\|\nabla f_i(x^k_i) - \nabla f(x^k)\|^2\right] \) for \( t > 0, k \in [K] \) is an upper bound on the variance, see for instance [Zhao & Zhang, 2015]. By averaging \( K \) stochastic gradients—such as in parallel SGD—the variance decreases by a factor of \( K \), and we conclude that parallel SGD converges at a rate \( O\left(\frac{\sigma^2}{TK}\right) \), i.e. achieves a linear speedup.

**Towards local SGD.** For local SGD such a simple argument is elusive. For instance, just capitalizing the convexity of the objective function \( f \) is not enough: this will show that the averaged iterate of \( K \) independent SGD sequences converges at rate \( O\left(\frac{\sigma^2}{T}\right) \), i.e. no speedup can be shown in this way.

This indicates that one has to show that local SGD decreases the variance \( \sigma^2 \) instead, similar as in parallel SGD. Suppose the different sequences \( x^k_i \) evolve close to each other. Then it is reasonable to assume that averaging the stochastic gradients \( \nabla f_i(x^k_i) \) for all \( k \in [K] \) can still yield a reduction in the variance by a factor of \( K \) — similar as in parallel SGD. Indeed, we will make this statement precise in the proof below.

### 2.2 Convergence Result and Discussion

**Theorem 2.2.** Let \( f \) be \( L \)-smooth and \( \mu \)-strongly convex, \( \mathbb{E}_i \left[\|\nabla f_i(x^k_i) - \nabla f(x^k)\|^2\right] \leq \sigma^2 \), \( \mathbb{E}_i \left[\|\nabla f_i(x^k_i)\|^2\right] \leq G^2 \), for \( t = 0, \ldots, T - 1 \), where \( \{x^k_i\}_{i=0}^T \) for \( k \in [K] \) are generated according to (4) with \( \text{gap}(I_T) \leq H \) and for stepsizes \( \eta_i = \frac{1}{\mu(\alpha + T)} \) with shift parameter \( \alpha > \max\{16\kappa, H\} \), for \( \kappa = \frac{L}{\mu} \). Then

\[
\mathbb{E} f(\mathbf{x}_T) - f^* \leq \frac{\mu_3}{2S_T} \|x_0 - x^*\|^2 + \frac{4T(T + 2\alpha)}{\mu KS_T} \sigma^2 + \frac{256T}{\mu^2 S_T} G^2 H^2 L ,
\]

where \( \mathbf{x}_T = \frac{1}{\mu \sigma^2} \sum_{k=1}^K \sum_{t=0}^{T-1} \omega_t x^k_t \) for \( \omega_t = (\alpha + t)^2 \) and \( S_T = \sum_{t=0}^{T-1} \omega_t \geq \frac{1}{3} T^3 \).

We were not especially careful to optimize the constants (and the lower order terms) in (5), so we now state the asymptotic result.

**Corollary 2.3.** Let \( \mathbf{x}_T \) be as defined as in Theorem 2.2 for parameter \( \alpha = \max\{16\kappa, H\} \). Then

\[
\mathbb{E} f(\mathbf{x}_T) - f^* = O\left(\frac{1}{\mu KT} + \frac{\kappa + H}{\mu KT^2}\right) \sigma^2 + O\left(\frac{\kappa H^2}{\mu T^2} + \frac{\kappa^3 + H^3}{\mu T^3}\right) G^2 .
\]

In order to measure the longest interval between subsequent synchronization steps, we introduce the gap of a set of integers.

**Algorithm 1 LOCAL SGD**

1: Initialize variables \( x^k_0 = x_0 \) for \( k \in [K] \)
2: for \( t \) in \( 0 \ldots T - 1 \) do
3: parallel for \( k \in [K] \) do
4: Sample \( i^k_t \) uniformly in \([n]\)
5: \( x^k_{t+1} \leftarrow x^k_t - \eta_t \nabla f_{i^k_t}(x^k_t) \) \( \triangleright \) local update
6: end parallel for
7: if \( t + 1 \in I_T \) then
8: \( x^k_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^K x^k_t \) \( \triangleright \) global synchronization
9: end if
10: end for

Under review as a conference paper at ICLR 2019
We now give the outline of the proof. The proofs of the lemmas are given in Appendix A.

We now state some consequences of equation (6). For the ease of the exposition we omit the dependency on $L$, $\mu$, $\sigma^2$ and $G^2$ below, but depict the dependency on the local mini-batch size $b$.

**Convergence rate.** For $T$ large enough, the very first term is dominating in (6) and local SGD converges at rate $O(1/(KTb))$. That is, local SGD achieves a linear speedup in both, the number of workers $K$ and the mini-batch size $b$.

**Global synchronization steps.** It needs to hold $H = O(\sqrt{T/(Kb)})$ to get the linear speedup. This yields a reduction of the number of communication rounds by a factor $O(\sqrt{T/(Kb)})$ compared to parallel mini-batch SGD without hurting the convergence rate.

**Extreme Cases.** We have not optimized the result for extreme settings of $H$, $K$, $L$ or $\sigma$. For instance, we do not recover convergence for the one shot averaging, i.e. the setting $H = T$ (though convergence for $H = o(T)$, but at a lower rate).

**Unknown Time Horizon/Adaptive Communication Frequency.** Zhang et al. (2016) empirically observe that more frequent communication at the beginning of the optimization can help to get faster time-to-accuracy. Indeed, when the number of total iterations $T$ is not known beforehand (as it e.g. depends on the target accuracy, cf. (9) and also Section 4 below), then increasing the communication frequency seems to be a good strategy to keep the communication low, why still respecting the constraint $H = O(\sqrt{T/(Kb)})$ for all $T$.

## 3 Proof Outline

We now give the outline of the proof. The proofs of the lemmas are given in Appendix A.

**Perturbed iterate analysis.** Inspired by the perturbed iterate framework of Mania et al. (2017) we first define a virtual sequence $\{\bar{x}_t\}_{t \geq 0}$ in the following way:

$$\bar{x}_0 = x_0, \quad \bar{x}_t = \frac{1}{K} \sum_{k=1}^{K} x^k_t, \quad (7)$$

where the sequences $\{x^k_t\}_{t \geq 0}$ for $k \in [K]$ are the same as in (4). Notice that this sequence never has to be computed explicitly, it is just a tool that we use in the analysis. Further notice that $\bar{x}_t = x^k_t$ for $k \in [K]$ whenever $t \in \mathcal{I}_T$. Especially, when $\mathcal{I}_T = [T]$, then $\bar{x}_t \equiv x^k_t$ for every $k \in [K], t \in [T]$. It will be useful to define

$$g_t := \frac{1}{K} \sum_{k=1}^{K} \nabla f^k_t (x^k_t), \quad \bar{g}_t := \frac{1}{K} \sum_{k=1}^{K} \nabla f(x^k_t). \quad (8)$$

Observe $\bar{x}_{t+1} = \bar{x}_t - \eta_t g_t$ and $\mathbb{E} g_t = \bar{g}_t$.

Now the proof proceeds as follows: we show (i) that the virtual sequence $\{\bar{x}_t\}_{t \geq 0}$ almost behaves like mini-batch SGD with batch size $K$ (Lemma 3.1 and 3.2), and (ii) the true iterates $\{x^k_t\}_{t \geq 0, k \in [K]}$ do not deviate much from the virtual sequence (Lemma 3.3). These are the main ingredients in the proof. To obtain the rate we exploit a technical lemma from Stich et al. (2018).

**Lemma 3.1.** Let $\{x^k_t\}_{t \geq 0}$ and $\{\bar{x}_t\}_{t \geq 0}$ for $k \in [K]$ be defined as in (4) and (7) and let $f$ be $L$-smooth and $\mu$-strongly convex and $\eta_t \leq 1/4L$. Then

$$\mathbb{E}\|\bar{x}_{t+1} - x^*\|^2 \leq (1 - \mu \eta_t) \mathbb{E}\|\bar{x}_t - x^*\|^2 + \eta_t^2 \mathbb{E}\|g_t - \bar{g}_t\|^2 - \frac{1}{2} \eta_t \mathbb{E}(f(\bar{x}_t) - f^*) + 2 \eta_t L \sum_{k=1}^{K} \mathbb{E}\|\bar{x}_t - x^k_t\|^2. \quad (9)$$

Under review as a conference paper at ICLR 2019
Bounding the variance. From equation [9] it becomes clear that we should derive an upper bound on \( \mathbb{E} \| g_t - \bar{g}_t \|^2 \). We will relate this to the variance \( \sigma^2 \).

Lemma 3.2. Let \( \sigma^2 \geq \mathbb{E} \| \nabla f_i(x^k_t) - \nabla f(x^k_t) \|^2 \) for \( k \in [K] \), \( t \in [T] \). Then \( \mathbb{E} \| g_t - \bar{g}_t \|^2 \leq \frac{\sigma^2}{K} \).

Bounding the deviation. Further, we need to bound \( \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \| \bar{x}_t - x^k_t \|^2 \). For this we impose a condition on \( I_T \) and an additional condition on the stepsize \( \eta_t \).

Lemma 3.3. If \( \operatorname{gap}(I_T) \leq H \) and sequence of decreasing positive stepsizes \( \{\eta_t\}_{t \geq 0} \) satisfying \( \eta_t \leq 2\eta_{t+H} \) for all \( t \geq 0 \), then

\[
\frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \| \bar{x}_t - x^k_t \|^2 \leq 4\eta_t^2 G^2 H^2, \tag{10}
\]

where \( G^2 \) is a constant such that \( \mathbb{E} \| \nabla f_i(x^k_t) \|^2 \leq G^2 \) for \( k \in [K], t \in [T] \).

Optimal Averaging. Similar as in [Lacoste-Julien et al., 2012; Shamir & Zhang, 2013; Rakhlin et al., 2012], we define a suitable averaging scheme for the iterates \( \{x_t\}_{t \geq 0} \) to get the optimal per iteration rate. In contrast to [Lacoste-Julien et al., 2012] that use linearly increasing weights, we use quadratically increasing weights, as for instance [Shamir & Zhang, 2013; Stich et al., 2018].

Lemma 3.4 ([Stich et al., 2018]). Let \( \{a_t\}_{t \geq 0}, a_t \geq 0, \{e_t\}_{t \geq 0}, e_t \geq 0 \) be sequences satisfying

\[
a_{t+1} \leq (1 - \mu e_t) a_t - \eta_t e_t A + \eta_t^2 B + \eta_t^3 C, \tag{11}
\]

for \( \eta_t = \frac{4}{\mu(a+t)} \) and constants \( A > 0, B, C \geq 0, \mu > 0, a > 1 \). Then

\[
\frac{A}{S_T} \sum_{t=0}^{T-1} w_t e_t \leq \frac{\mu a^3}{4 S_T} a_0 + \frac{2T(T+2a)}{\mu S_T} B + \frac{16T}{\mu^2 S_T} C, \tag{12}
\]

for \( w_t = (a+t)^2 \) and \( S_T := \sum_{t=0}^{T-1} w_t = \frac{T}{6} (2T^2 + 6aT - 3T + 6a^2 - 6a + 1) \geq \frac{1}{3} T^3 \).

Proof. This is a reformulation of Lemma 3.3 in [Stich et al., 2018]. \( \square \)

Proof of Theorem 2.2. By convexity of \( f \) we have \( \mathbb{E} f(\bar{x}_T) - f^* \leq \frac{1}{2} \sum_{t=0}^{T-1} w_t \mathbb{E} (f(x^k_t) - f^*) \).

The proof of the theorem thus follows immediately from the four lemmas that we have presented, i.e. by Lemma 3.4 with \( e_t := \mathbb{E} (f(\bar{x}_t) - f^*) \) and constants \( A = \frac{1}{2}, \) (Lemma 3.1), \( B = \frac{\sigma^2}{H} \), (Lemma 3.2) and \( C = 8G^2 H^2 \), (Lemma 3.3). Observe that the stepsizes \( \eta_t = \frac{4}{\mu(a+t)} \) satisfy both the conditions of Lemma 3.1 \( (\eta_0 = \frac{1}{\mu a}, H = \alpha) \) as \( a \geq 16\alpha \) and of Lemma 3.3 \( (\eta_t = \frac{a_t + 1 + H}{a + t}, 2, \alpha \geq H) \).

4 NUMERICAL ILLUSTRATION

In this section we show some numerical experiments to illustrate the results of Theorem 2.2.

Speedup. When Algorithm 1 is implemented in a distributed setting, there are two components that determine the wall-clock time: (i) the total number of gradient computations, \( TK \), and (ii) the total time spend for communication. In each communication round \( 2(K-1) \) vectors need to be exchanged, and there will be \( T/H \) communication rounds. Typically, the communication is more expensive than a single gradient computation. We will denote this ratio by a factor \( \rho \geq 1 \) (in practice, \( \rho \) can be 10–100, or even larger on slow networks). The parameter \( T \) depends on the desired accuracy \( \epsilon > 0 \), and according to (6) we roughly have \( T(\epsilon, H, K) \approx \frac{1}{K \rho} \left( \frac{1}{2} + \frac{1}{2} \sqrt{1 + \epsilon(1 + H + H^2 K)} \right) \).

Thus, the theoretical speedup \( S(K) \) of local SGD on \( K \) machines compared to SGD on one machine \( (H = 1, K = 1) \) is

\[
S(K) = \frac{K}{\frac{1}{2} + \frac{1}{2} \sqrt{1 + \epsilon(1 + H + H^2 K)} \left( 1 + 2\sigma^2 \frac{(K-1)}{H} \right)} \tag{13}
\]
We also see that for a small number of workers $H$ the technique from (Zhang et al., 2016), i.e. adaptively increasing $H$ over time seems to be a good strategy to get the best choice of $H$ when the time horizon is unknown.

**Experimental.** We examine the practical speedup on a logistic regression problem, $f(x) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-b_i a_i^T x)) + \frac{1}{2} \|x\|^2$, where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, +1\}$ are the data samples. The regularization parameter is set to $\lambda = 1/n$. We consider the w8a dataset (Platt, 1999) ($d = 300$, $n = 49749$). We initialize all runs with $x_0 = 0_d$ and measure the number of iterations to reach the target accuracy $\epsilon$. We consider the target accuracy reached, when either the last iterate, the uniform average, the average with linear weights, or the average with quadratic weights (such as in Theorem 2.2) reaches the target accuracy. By extensive grid search we determine for each configuration $(H, K, B)$ the best stepsize from the set $\{\min(32, \frac{c}{tH\epsilon}), 32c\}$, where $c$ can take the values $c = 2^i$ for $i \in \mathbb{Z}$. For more details on the experimental setup refer Section D in the appendix. We depict the results in Figure 3 again under the assumption $\rho = 25$.

**Conclusion.** The restriction on $H$ imposed by theory is not severe for $T \to \infty$. Thus, for training that either requires many passes over the data or that is performed only on a small cluster, large values of $H$ are advisable. However, for smaller $T$ (few passes over the data), the $O(1/\sqrt{K})$ dependency shows significantly in the experiment. This has to be taken into account when deploying the algorithm on a massively parallel system, for instance through the technique mentioned in (Zhang et al., 2016).

## 5 Asynchronous Local SGD

In this section we present asynchronous local SGD that does not require that the local sequences are synchronized at the same time step. This does not only reduce communication bottlenecks, but by using load-balancing techniques the algorithm can optimally be tuned to heterogeneous settings (slower workers do less computation between synchronization, and faster workers do more). We will discuss this in more detail in Section C.

Asynchronous local SGD generates in parallel $K$ sequences $\{x_k^t\}_{t=0}^T$ of iterates, $k \in [K]$. Similar as in Section 2 we introduce sets of synchronization indices, $I_k^t \subseteq [T]$ with $T \in I_k^T$ for $k \in [K]$. Note that the sets do not have to be equal for different workers. Each worker $k$ evolves locally a sequence...
We prove convergence of synchronous and asynchronous local SGD and are the first to show that if we cast synchronized local SGD in this notation, then it holds where $\mathcal{W}$

\begin{algorithm}[h]
\caption{Asynchronous Local SGD (Schematic)}
1: Initialize variables $x_0^k = x_0$, $r^k = 0$ for $k \in [K]$, aggregate $\bar{x} = x_0$.
2: $\text{parallel for } k \in [K]$ do
3: $\text{for } t \in 0 \ldots T - 1$ do
4: Sample $i_t^k$ uniformly in $[n]$
5: $x_{t+1}^k \leftarrow x_t^k - \gamma_t \nabla f_{i_t^k}(x_t^k)$ \hfill $\triangleright$ local update
6: if $t + 1 \notin \mathcal{I}_{t+1}^k$ then
7: $\bar{x} \leftarrow \text{add}(\bar{x}, \frac{1}{n}(x_t^k - x_{t+1}^k))$ \hfill $\triangleright$ atomic write of updates
8: $x_{t+1}^k \leftarrow \text{read}(\bar{x})$; update time of last read $r^k \leftarrow t + 1$
9: end if
10: end parallel for

end for
end parallel for

\end{algorithm}

$x_i^k$ in the following way:

$$x_{t+1}^k = \begin{cases} x_t^k - \gamma_t \nabla f_{i_t^k}^t(x_t^k) & \text{if } t + 1 \notin \mathcal{I}_{t+1}^k \\ \bar{x}_{t+1} & \text{if } t + 1 \in \mathcal{I}_{t+1}^k \end{cases}$$ \quad (14)

where $\bar{x}_{t+1}$ denotes the state of the aggregated variable at the time when worker $k$ reads the aggregated variable. To be precise, we use the notation

$$\bar{x}_t^k = x_0 - \frac{1}{K} \sum_{h=1}^K \sum_{j=0}^{t-1} I_j \in \mathcal{W}_{t}^{k, h} (\gamma_j \nabla f_{i_j^h}^t(x_j^h)),$$

where $\mathcal{W}_{t}^{k, h} \subseteq [T]$ denotes all updates that have been written at the time the read takes place. The sets $\mathcal{W}_{t}^{k, h}$ are indexed by iteration $t$, worker $k$ that initiates the read and $h \in [K]$. Thus $\mathcal{W}_{t}^{k, h}$ denotes all updates of the local sequence $\{x_j^h\}_{j \geq 0}$, that have been reported back to the server at the time worker $k$ reads (in iteration $t$). This notation is necessary, as we don’t necessarily have $\mathcal{W}_t^{k, h} = \mathcal{W}_t^{k', h}$ for $k \neq k'$. We have $\mathcal{W}_t^{k, h} \subseteq \mathcal{W}_t^{k', h}$ for $t' \geq t$, as updates are not overwritten. When we cast synchronized local SGD in this notation, then it holds $\mathcal{W}_t^{k, h} = \mathcal{W}_t^{k', h'}$ for all $k, h, k', h'$, as all the writes and reads are synchronized.

**Theorem 5.1.** Let $f, \sigma, G$ and $\kappa$ be as in Theorem 5.1 and sequences $\{x_t^j\}_{t=0}^T$ for $k \in [K]$ generated according to (14) with gap $\mathcal{I}_{t+1}^k \leq H$ for $k \in K$ and for stepsize $\eta_t = \frac{1}{\mu (a + \tau)}$ with shift parameter $a > \max \{16\kappa, H + \tau\}$. If there exists $\tau > 0$, s.t. $\Theta_t^{k, h, \tau} \geq [t - \tau]$ for all $k, h \in [K], t \in [T]$, then

$$\mathbb{E} f(\bar{x}_T) - f^* \leq \frac{\mu a^3}{2ST} \|x_0 - x^*\|^2 + \frac{4T(T + 2\tau)}{\mu KST} \sigma^2 + \frac{768T}{\mu ST} G^2 (H + \sigma^2) L,$$

where $\bar{x}_T = \frac{1}{KST} \sum_{k=1}^K \sum_{i=0}^{T-1} w_i x_i^k$, for $w_i = (a + t)^2$ and $S_T = \sum_{t=0}^{T-1} w_t \geq \frac{1}{3} T^3$.

Hence, for $T$ large enough and $(H + \tau) = O(\sqrt{T/K})$, asynchronous local SGD converges with rate $O(\frac{G^2}{K})$, the same rate as synchronous local SGD.

6 Conclusion

We prove convergence of synchronous and asynchronous local SGD and are the first to show that local SGD attains theoretically linear speedup on strongly convex functions when parallelized among $K$ workers. We show that local SGD saves up to a factor of $O(T^{1/2})$ in global communication rounds compared to mini-batch SGD, while still converging at the same rate in terms of total stochastic gradient computations. There are no apparent reasons that would limit the extension of the theory to non-convex objective functions; Lemma 3.3 does neither use the smoothness nor the strong convexity assumption, so this can be applied in the non-convex setting as well. We feel that the positive results shown here can motivate and spark further research on non-convex problems. Indeed, very recent work [Yu et al., 2018] analyzes local SGD for non-convex optimization problems and shows convergence of SGD to a stationary point, though the restrictions on $H$ are stronger than here.
REFERENCES


Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (eds.), INTERSPEECH, pp. 1058–1062. ISCA, 2014. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2014.html#SeideFLDY14


A Missing Proofs for Synchronized Local SGD

In this section we provide the proofs for the three lemmas that were introduced in Section 3.

Proof of Lemma 3.1 Using the update equation (7) we have

\[ \| \bar{x}_{t+1} - x^* \|^2 = \| \bar{x}_t - \eta_t g_t - x^* \|^2 = \| \bar{x}_t - \eta_t g_t - x^* - \eta_t g_t + \eta_t g_t \|^2 \]

\[ = \| \bar{x}_t - x^* - \eta_t g_t \|^2 + \eta_t^2 \| g_t - \bar{g}_t \|^2 + 2 \eta_t \langle \bar{x}_t - x^* - \eta_t g_t, \bar{g}_t - g_t \rangle . \]  

(17)

(18)

Observe that

\[ \| x_t - x^* - \eta_t g_t \|^2 = \| x_t - x^* \|^2 + \eta_t^2 \| g_t \|^2 - 2 \eta_t \langle x_t - x^*, g_t \rangle \]

\[ = \| x_t - x^* \|^2 + \eta_t^2 \| g_t \|^2 - 2 \eta_t \sum_{k=1}^{K} \langle x_t - x^*, \nabla f(x_t^k) \rangle \]

\[ \leq \| x_t - x^* \|^2 + \eta_t^2 \sum_{k=1}^{K} \| \nabla f(x_t^k) \|^2 \]

\[ - 2 \eta_t \sum_{k=1}^{K} \langle x_t - x_t^k + x_t^k - x^*, \nabla f(x_t^k) \rangle \]

\[ = \| x_t - x^* \|^2 + \eta_t^2 \sum_{k=1}^{K} \| \nabla f(x_t^k) - \nabla f(x^*) \|^2 \]

\[ - 2 \eta_t \sum_{k=1}^{K} \langle x_t^k - x^*, \nabla f(x_t^k) \rangle - 2 \eta_t \sum_{k=1}^{K} \langle x_t - x_t^k, \nabla f(x_t^k) \rangle, \]

(22)

where we used the inequality \( \sum_{i=1}^{K} a_i \|^2 \leq K \sum_{i=1}^{K} \| a_i \|^2 \) in (21). By \( L \)-smoothness,

\[ \| \nabla f(x_t^k) - \nabla f(x^*) \|^2 \leq 2L(f(x_t^k) - f^*) , \]

(23)

and by \( \mu \)-strong convexity

\[ - \langle x_t^k - x^*, \nabla f(x_t^k) \rangle \leq -(f(x_t^k) - f^*) - \frac{\mu}{2} \| x_t^k - x^* \|^2 . \]

(24)

To estimate the last term in (22) we use \( 2 \langle a, b \rangle \leq \gamma \| a \|^2 + \gamma^{-1} \| b \|^2 \), for \( \gamma > 0 \). This gives

\[ -2 \langle x_t - x_t^k, \nabla f(x_t^k) \rangle \leq 2L \| x_t - x_t^k \|^2 + \frac{1}{2L} \| \nabla f(x_t^k) \|^2 \]

\[ = 2L \| x_t - x_t^k \|^2 + \frac{1}{2L} \| \nabla f(x_t^k) - \nabla f(x^*) \|^2 \]

\[ \leq 2L \| x_t - x_t^k \|^2 + (f(x_t^k) - f^*) , \]

(27)

where we have again used (25) in the last inequality. By applying these three estimates to (22) we get

\[ \| \bar{x}_t - x^* - \eta_t g_t \|^2 \leq \| \bar{x}_t - x^* \|^2 + 2 \eta_t \frac{L}{K} \sum_{k=1}^{K} \| x_t - x_t^k \|^2 \]

\[ + 2 \eta_t \sum_{k=1}^{K} \left( \left( \eta_t L - \frac{1}{2} \right) \left( f(x_t^k) - f^* \right) - \frac{\mu}{2} \| x_t^k - x^* \|^2 \right) , \]

(28)

For \( \eta_t \leq \frac{1}{4L} \) it holds \( (\eta_t L - \frac{1}{2}) \leq -\frac{1}{4} \). By convexity of \( a(f(x) - f^*) + b\| x - x^* \|^2 \) for \( a, b \geq 0 \):

\[ - \frac{1}{K} \sum_{k=1}^{K} \left( a(f(x_t^k) - f^*) + b\| x_t^k - x^* \|^2 \right) \leq - \left( a(f(\bar{x}_t) - f^*) + b\| \bar{x}_t - x^* \|^2 \right) , \]

(29)
hence we can continue in (28) and obtain
\[
\|\bar{x}_t - x^* - \eta_t \bar{g}_t\|^2 \leq (1 - \mu \eta_t) \|\bar{x}_t - x^*\|^2 - \frac{1}{2} \eta_t (f(\bar{x}_t) - f^*) + 2\eta_t \frac{L}{K} \sum_{k=1}^{K} \|\bar{x}_t - x_k^t\|^2. \tag{30}
\]

Finally, we can plug (30) back into (18). By taking expectation we get
\[
\mathbb{E} \|\bar{x}_{t+1} - x^*\|^2 \leq (1 - \mu \eta_t) \mathbb{E} \|\bar{x}_t - x^*\|^2 + \eta_t^2 \mathbb{E} \|\bar{g}_t - \bar{g}_t\|^2 - \frac{1}{2} \eta_t \mathbb{E} (f(\bar{x}_t) - f^*) + 2\eta_t \frac{L}{K} \mathbb{E} \sum_{k=1}^{K} \|\bar{x}_t - x_k^t\|^2.
\]

**Proof of Lemma 3.2** By definition of $g_t$ and $\bar{g}_t$ we have
\[
\mathbb{E} \|g_t - \bar{g}_t\|^2 = E \left\| \frac{1}{K} \sum_{k=1}^{K} \left( \nabla f_{t^k}^i (x_k^t) - \nabla f (x_k^t) \right) \right\|^2 = \frac{1}{K^2} \sum_{k=1}^{K} E \left\| \nabla f_{t^k}^i (x_k^t) - \nabla f (x_k^t) \right\|^2 \leq \frac{\sigma^2}{K}, \tag{31}
\]
where we used $\text{Var}(\sum_{k=1}^{K} X_k) = \sum_{k=1}^{K} \text{Var}(X_k)$ for independent random variables.

**Proof of Lemma 3.3** As the gap($i_T$) $\leq H$, there is an index $t_0$, $t - t_0 \leq H$ such that $\bar{x}_{t_0} = x_k^t_0$ for $k \in [K]$. Observe, using $E \|X - \mathbb{E} X\|^2 = \mathbb{E} \|X\|^2 - \|\mathbb{E} X\|^2$ and $\|\sum_{i=1}^{H} a_i\|^2 \leq H \sum_{i=1}^{H} \|a_i\|^2$,
\[
\frac{1}{K} \sum_{k=1}^{K} E \|\bar{x}_t - x_k^t\|^2 = \frac{1}{K} \sum_{k=1}^{K} E \left\| x_k^t - x_{t_0} - (\bar{x}_t - x_{t_0})\right\|^2 \leq \frac{1}{K} \sum_{k=1}^{K} E \|x_k^t - x_{t_0}\|^2 \tag{32}
\]
\[
\leq \frac{1}{K} \sum_{k=1}^{K} \left[ t_0 H \eta_t \sum_{h=t_0}^{t-1} E \left\| \nabla f_{t^k}^i (x_h^k) \right\|^2 \right] \tag{33}
\]
\[
\leq \frac{1}{K} \sum_{k=1}^{K} H^2 \eta_t^2 G^2, \tag{34}
\]
where we used $\eta_t \leq \eta_0$ for $t \geq t_0$ and the assumption $E \|\nabla f_{t^k}^i (x_h^k)\|^2 \leq G^2$. Finally, the claim follows by the assumption on the stepsizes, $\frac{\eta_t}{\mu} \leq 2$.

**B Missing Proof for Asynchronous Local SGD**

In this Section we prove Theorem 5.1. The proof follows closely the proof presented in Section 3.

We again introduce the virtual sequence
\[
\bar{x}_t = x_0 - \frac{1}{K} \sum_{h=1}^{t} \sum_{j=0}^{t-1} \eta_j \nabla f_{t^j}^i (x_j^h), \tag{36}
\]
as before. By the property $T \in \mathcal{I}_k^T$ for $k \in K$ we know that all workers will written their updates then the algorithm terminates. This assumption is not very critical and could be relaxed, but it facilitates the (already quite heavy) notation in the proof.

Observe, that Lemmas 3.1 and 3.2 hold for the virtual sequence $\{\bar{x}_t\}_{t=0}^{T}$. Hence, all we need is a refined version of Lemma 3.3 that bounds how far the local sequences can deviate from the virtual average.
We decompose the error term as

\[ \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \left\| \bar{x}_t - x^{k}_t \right\|^2 \leq 12 \eta_t^2 G^2 (H + \tau)^2, \tag{37} \]

where \( G^2 \) is a constant such that \( \mathbb{E}_t \| \nabla f_i(x^t) \|^2 \leq G^2 \) for all \( k \in [K], t \in [T] \).

Here we use the notation \( [s] = \{ \} \) for \( s < 0 \), such that \( \lceil t - \tau \rceil \) is also defined for \( t < \tau \).

**Proof.** As \( \text{gap}(I^k_t) \leq H \) there exists for every \( k \in K \) a \( t_k, t - t_k \leq H \), such that \( x^k_t = \bar{x}^k_t \). Let \( t_0 := \min\{t_1, \ldots, t_K\} \) and observe \( t_0 \geq t - H \). Let \( t'_0 = \max\{t_0 - \tau, 0\} \). As \( \mathcal{W}^{k,h}_t \supseteq [t - \tau] \) for all \( k, h \in [K], t \in [T] \), it holds

\[ \bar{x}^k_{t_k} = x_{t'_0} - \frac{1}{K} \sum_{k=1}^{K} \sum_{j=t'_0}^{t_k-1} \mathbb{1}_{j \in \mathcal{W}^{k,h}_t} (\gamma_j \eta_j f_j(x^j_t)), \tag{38} \]

for each \( k \in [K] \). In other words, all updates up to iteration \( t'_0 \) have been written to the aggregated sequence.

We decompose the error term as

\[ \left\| \bar{x}_t - x^k_t \right\|^2 \leq 3 \left( \left\| x^k_t - x_t \right\|^2 + \left\| x^k_t - \bar{x}^k_t \right\|^2 + \left\| \bar{x}^k_t - \bar{x}_t \right\|^2 \right). \tag{39} \]

Now, using \( \eta_t \geq \eta_{t+1} \), and \( t - t_k \leq H \), we conclude (as in (35))

\[ \left\| x^k_t - x^k_t \right\|^2 \leq \eta^2_t H^2 G^2 \leq \eta^2_{t_0} H^2 G^2. \tag{40} \]

As \( t_k - t'_0 \leq \tau \),

\[ \left\| x^k_{t_k} - x_{t'_0} \right\|^2 \leq \eta^2_{t_0} \tau^2 G^2, \tag{41} \]

and similarly, as \( t - t'_0 \leq H + \tau \),

\[ \left\| \bar{x}_{t'_0} - \bar{x}_t \right\|^2 \leq \eta^2_{t_0} (H + \tau)^2 G^2. \tag{42} \]

Finally, as \( \frac{\eta_{t_0}}{\eta_t} \leq 2 \), we can conclude

\[ \left\| \bar{x}_t - x^k_t \right\|^2 \leq 12 \eta^2_t (H + \tau)^2 G^2. \tag{43} \]

and the lemma follows. \( \square \)

Now the proof of Theorem 5.1 follows immediately.

**Proof of Theorem 5.1** As in the proof of Theorem 2.2 we rely on Lemma 3.4 to derive the convergence rate. Again we have \( A = \frac{1}{2}, B = \frac{\tau^2}{1} \), and \( C = LG^2 (H + \tau)^2 \) (Lemma B.1). It is easy to see that the stepsizes satisfy the condition of Lemma B.1 as clearly \( \frac{\eta_{t_0}}{\eta_t} \leq \frac{\eta_{t_0}}{\eta_{t_0} + H + \tau} = \frac{a + t + H + \tau}{a + t} \leq 2 \), as \( a \geq H + \tau \). \( \square \)

**C Comments on Implementation Issues**

**C.1 Synchronous Local SGD**

In Theorem 5 we do not prove convergence of the sequences \( \{ x^k_t \}_{t \geq 0} \) of the iterates, but only convergence of a weighted average of all iterates. In practice, the last iterate might often be sufficient, but we like to remark that the weighted average of the iterates can easily be tracked on the fly with an auxiliary sequence \( \{ y_t \}_{t \geq 0}, y_0 = x_0 \), without storing all intermediate iterates, see Table 1 for some examples.
We here state the precise procedure that was used to generate the figures in this report. As briefly
mentioned, for synchronous local SGD, the weighted averages of the iterates (if needed), can be tracked on
each worker locally by a recursive formula as explained above.

As for synchronous local SGD, the weighted averages of the iterates (if needed), can be tracked on
each worker locally by a recursive formula as explained above. The precise formulas for the
load balancing, \( \lambda \leq 3H \).

### C.2 Asynchronous Local SGD

As for synchronous local SGD, the weighted averages of the iterates (if needed), can be tracked on
each worker locally by a recursive formula as explained above.

Table 1: Formulas to recursively track weighted averages.

<table>
<thead>
<tr>
<th>criteria</th>
<th>weights</th>
<th>formula</th>
<th>recursive update</th>
</tr>
</thead>
<tbody>
<tr>
<td>last iterate</td>
<td>( w_t = 1 )</td>
<td>( y_t = x_t )</td>
<td>( y_t = x_t )</td>
</tr>
<tr>
<td>uniform average</td>
<td>( w_t = 1 )</td>
<td>( y_t = \frac{1}{t+1} \sum_{i=0}^{t} x_i )</td>
<td>( y_t = \frac{1}{t+1} x_t + \frac{1}{t+2} y_{t-1} )</td>
</tr>
<tr>
<td>linear weights</td>
<td>( w_t = (t + 1) )</td>
<td>( y_t = \frac{1}{(t+1)(t+2)} \sum_{i=0}^{t} (i+1)x_i )</td>
<td>( y_t = \frac{1}{2} x_t + \frac{1}{2} y_{t-1} )</td>
</tr>
<tr>
<td>quadratic weights</td>
<td>( w_t = (t + 1)^2 )</td>
<td>( y_t = \frac{1}{(t+1)(t+2)(t+3)} \sum_{i=0}^{t} (i+1)^2x_i )</td>
<td>( y_t = \frac{1}{3} x_t + \frac{2}{3} y_{t-1} )</td>
</tr>
</tbody>
</table>

In Figures 4 and 5 we give additional results for mini-batch sizes \( b \in \{1, 16\} \).

### D Details on Experiments

We here state the precise procedure that was used to generate the figures in this report. As briefly
stated in Section 4 we examine empirically the speedup on a logistic regression problem, \( f(x) = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \exp(-b_i a_i^T x)) + \frac{1}{2} \|x\|^2 \), where \( a_i \in \mathbb{R}^d \) and \( b_i \in \{-1, +1\} \) are the data samples. The
regularization parameter is set to \( \lambda = 1/n \). We consider the small scale \( \text{MNIST} \) dataset (Platt 1999)
(\( d = 300, n = 49749 \)).

For each run, we initialize \( x_0 = 0_d \) and measure the number of iterations (and number of stochastic
gradient evaluations) to reach the target accuracy \( \epsilon \in \{0.005, 0.0001\} \). As we prove convergence
only for a special weighted sum of the iterates in Theorem 2.2 and not for standard criteria (last
iterate or uniform average), we evaluate the function value for different weighted averages \( y_t = \sum_{i=0}^{t} w_i x_i \), and consider the accuracy reached when one of the averages satisfies \( f(y_t) - f^* \leq \epsilon \), with \( f^* := 0.126433176216545 \) (numerically determined). The precise formulas for the
averages that we used are given in Table 1.

For each configuration \((K, H, b, c)\), we report the best result found with any of the following two
stepsizes: \( \eta_0 := \min(32, \frac{c}{H \tau}) \) and \( \eta_0 = 32c \). Here \( c \) is a parameter that can take the values \( c = 2^i \) for
\( i \in \mathbb{Z} \). For each stepsize we determine the best parameter \( c \) by a grid search, and consider parameter \( c \) optimal, if parameters \( \{2^{-2} c, 2^{-1} c, 2c, 2^2 c\} \) yield worse results (i.e. more iterations to reach
the target accuracy).

We conclude with the claim that, as mentioned above, the average of \( \{2^{-i} x_i\} \) is the same for all
\( i \in \mathbb{N} \). This is due to the fact that the iterates \( x_i \) are generated in a way that each weight is applied
only once to each parameter and each parameter is updated only once.

Table 2: Simple load balancing. The faster worker can advance both sequences, even when the slower
worker has not yet finished the computation. In the example each worker does \( H \) steps of local SGD
(denoted by the operator \( U : \mathbb{R}^d \rightarrow \mathbb{R}^d \)) before writing back the updates to the aggregate \( \bar{x} \). Due to
the load balancing, \( \tau \leq 3H \).

<table>
<thead>
<tr>
<th>wall clock time</th>
<th>( x^1_{H} \leftarrow U(\bar{x}) )</th>
<th>( x^2_{H} \leftarrow U(\bar{x}) )</th>
<th>( x^3_{H} \leftarrow U(\bar{x}) )</th>
<th>( x^4_{H} \leftarrow U(\bar{x}) )</th>
<th>( x^5_{H} \leftarrow U(\bar{x}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker 1</td>
<td>( x^1_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^2_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^3_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^4_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^5_{H} \leftarrow U(\bar{x}) )</td>
</tr>
<tr>
<td>worker 2</td>
<td>( x^1_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^2_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^3_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^4_{H} \leftarrow U(\bar{x}) )</td>
<td>( x^5_{H} \leftarrow U(\bar{x}) )</td>
</tr>
</tbody>
</table>
Figure 4: Measured speedup of local SGD with mini-batch $b = 1$ for different numbers of workers $K$ and parameters $H$.

(a) Measured speedup, $\epsilon = 0.005$.

(b) Measured speedup, $\epsilon = 0.0001$.

Figure 5: Measured speedup of local SGD with mini-batch $b = 16$ for different numbers of workers $K$ and parameters $H$.

(a) Measured speedup, $\epsilon = 0.005$.

(b) Measured speedup, $\epsilon = 0.0001$. 