STL-SGD: Speeding Up Local SGD with Stagewise Communication Period

Shuheng Shen[1, 3*], **Yifei Cheng**[2*], **Jingchang Liu**[5], **Linli Xu**[1, 4†]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]School of Data Science, University of Science and Technology of China
[3] Ant Financial Services Group
[4] IFLYTEK Co., Ltd.
[5]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
{vaip, chengyif}@mail.ustc.edu.cn, jliude@cse.ust.hk, linlixu@ustc.edu.cn

Abstract

Distributed parallel stochastic gradient descent algorithms are workhorses for large scale machine learning tasks. Among them, local stochastic gradient descent (Local SGD) has attracted significant attention due to its low communication complexity. Previous studies prove that the communication complexity of Local SGD with a fixed or an adaptive communication period is in the order of $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ and $O(N^{\frac{3}{4}}T^{\frac{3}{4}})$ when the data distributions on clients are identical (IID) or otherwise (Non-IID), where N is the number of clients and T is the number of iterations. In this paper, to accelerate the convergence by reducing the communication complexity, we propose *ST*agewise *L*ocal *SGD* (STL-SGD), which increases the communication period gradually along with decreasing learning rate. We prove that STL-SGD can keep the same convergence rate and linear speedup as mini-batch SGD. In addition, as the benefit of increasing the communication period, when the objective is strongly convex or satisfies the Polyak-Łojasiewicz condition, the communication complexity of STL-SGD is $O(N\log T)$ and $O(N^{\frac{1}{2}}T^{\frac{1}{2}})$ for the IID case and the Non-IID case respectively, achieving significant improvements over Local SGD. Experiments on both convex and non-convex problems demonstrate the superior performance of STL-SGD.

Introduction

We consider the task of distributed stochastic optimization, which employs N clients to solve the following empirical risk minimization problem:

$$\min_{x\in R^d} f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{1}$$

where $f_i(x) := \frac{1}{|\mathcal{D}_i|}\sum_{\xi\in\mathcal{D}_i} f(x,\xi)$ is the local objective of client i. $\mathcal{D}_i$'s denote the data distributions among clients, which can be possibly different. Specifically, the scenario where $\mathcal{D}_i$'s are identical corresponds to a central problem of traditional distributed optimization. When they are not identical, (1) captures the federated learning setting (McMahan et al. 2017; Kairouz et al. 2019; Lyu, Yu, and Yang 2020), where the local data in each mobile client is independent and private, resulting in high variance of the data distributions.

As representatives of distributed stochastic optimization methods, traditional Synchronous SGD (SyncSGD) (Dekel et al. 2012; Ghadimi and Lan 2013) and Asynchronous SGD (AsyncSGD) (Agarwal and Duchi 2011; Lian et al. 2015) achieve linear speedup theoretically with respect to the number of clients. Nevertheless, for both SyncSGD and AsyncSGD, communication needs to be conducted at each iteration and $O(d)$ parameters are communicated each time, incurring significant communication cost which restricts the performance in terms of time speedup. To address this dilemma, distributed algorithms with low communication cost, either by decreasing the communication frequency (Wang and Joshi 2018b; Stich 2019; Yu, Yang, and Zhu 2019; Shen et al. 2019) or by reducing the communication bits in each round (Alistarh et al. 2017; Stich, Cordonnier, and Jaggi 2018; Tang et al. 2019), become widely applied for large scale training.

Among them, Local SGD (Stich 2019) (also called FedAvg (McMahan et al. 2017)), which conducts communication every k iterations, enjoys excellent theoretical and practical performance (Lin et al. 2018; Stich 2019). In the IID case and the Non-IID case, the communication complexity of Local SGD is respectively proved to be $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ (Wang and Joshi 2018b; Stich 2019) and $O(N^{\frac{3}{4}}T^{\frac{3}{4}})$ (Yu, Yang, and Zhu 2019; Shen et al. 2019), while the linear speedup is maintained. When the objective satisfies the Polyak-Łojasiewicz condition (Karimi, Nutini, and Schmidt 2016), (Haddadpour et al. 2019a) provides a tighter theoretical analysis which shows that the communication complexity of Local SGD is $O(N^{\frac{1}{3}}T^{\frac{1}{3}})$. In terms of the communication period k, most previous studies of Local SGD choose to fix it through the iterations. In contrast, (Wang and Joshi 2018a) suggests using an adaptively decreasing k when the learning rate is fixed, and (Haddadpour et al. 2019a) proposes an adaptively increasing k as the iterations go on. Nevertheless, none of them achieve a communication complexity lower than $O(N^{\frac{1}{3}}T^{\frac{1}{3}})$. For strongly convex objectives, if a fixed learning rate is adopted, Local SGD with fixed communication period is proved to achieve $O(N\log{(NT)})$ (Stich and

*Equal contribution.
†The corresponding author.

Karimireddy 2019; Bayoumi, Mishchenko, and Richtarik 2020) communication complexity. However, the fixed learning rate results in suboptimal convergence rate $O(\frac{\log T}{NT})$. It remains an open problem as to whether the communication complexity can be further reduced with a varying k when the optimal convergence rate $O(\frac{1}{NT})$ is maintained, to which this paper provides an affirmative solution.

Main Contributions. We propose Stagewise Local SGD (STL-SGD), which adopts a stagewisely increasing communication period , and make the following contributions:

- We first prove that Local SGD achieves $O(\frac{1}{\sqrt{NT}})$ convergence when the objective is general convex. A novel insight is that, the convergence rate $O(\frac{1}{\sqrt{NT}})$ can be attained when setting k to be $O(\frac{1}{\eta N})$ and $O(\frac{1}{\sqrt{\eta N}})$ in the IID case and the Non-IID case respectively, where η is the learning rate. This indicates that the communication period is negatively relevant to the learning rate.
- Taking Local SGD as a subalgorithm and tuning its parameters stagewisely, we propose STL-SGD^{sc} for strongly convex problems, which geometrically increases the communication period along with decreasing learning rate. We prove that STL-SGD^{sc} achieves $O(\frac{1}{NT})$ convergence rate with communication complexities $O(N \log T)$ and $O(N^{\frac{1}{2}}T^{\frac{1}{2}})$ for the IID case and the Non-IID case, respectively.
- For non-convex problems, we propose the STL-SGD^{nc} algorithm, which uses Local SGD to optimize a regularized objective $f^{\gamma}_{x_s}(\cdot)$ inexactly at each stage. When the Polyak-Łojasiewicz condition holds, the same communication complexity as in strongly convex problems is achieved. For general non-convex problems, we prove that STL-SGD^{nc} achieves the linear speedup with communication complexities $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ and $O(N^{\frac{3}{4}}T^{\frac{3}{4}})$ for the IID case and the Non-IID case, respectively.

Related Works

Local SGD. When the data distributions on clients are identical, Local SGD is proved to achieve $O(\frac{1}{NT})$ convergence for strongly convex objectives (Stich 2019) and $O(\frac{1}{\sqrt{NT}})$ convergence for non-convex objectives (Wang and Joshi 2018b) when the communication period k satisfies $k \leq O(T^{\frac{1}{2}}/N^{\frac{3}{2}})$. As demonstrated in these results, Local SGD achieves a linear speedup with the communication complexity $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ for both strongly convex and non-convex objectives in the IID case. In addition, (Haddadpour et al. 2019a) justifies that $O(N^{\frac{1}{3}}T^{\frac{1}{3}})$ rounds of communication are sufficient to achieve $O(\frac{1}{NT})$ convergence for objectives which satisfy the Polyak-Łojasiewicz condition. On the other hand, for the Non-IID case, Local SGD is proved with a $O(1/\sqrt{NT})$ convergence rate under a communication complexity of $O(N^{\frac{3}{4}}T^{\frac{3}{4}})$ for non-convex objectives (Yu, Yang, and Zhu 2019; Shen et al. 2019). Meanwhile, for strongly convex objectives, a suboptimal convergence rate of $O(\frac{k^2}{\mu NT})$ (Li et al. 2020) is obtained. Beyond that, when a small fixed learning rate is adopted, (Bayoumi, Mishchenko, and Richtarik 2020) and (Karimireddy et al. 2019) prove that the communication complexity of Local SGD is $O(N \log(NT))$ and $O(N^{\frac{1}{2}}T^{\frac{1}{2}})$ for the IID case and the Non-IID case respectively, at the cost of a suboptimal convergence rate $O(\frac{\log T}{NT})$. For general non-convex objectives, (Haddadpour and Mahdavi 2019) proves a lower communication complexity of $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ for the Non-IID case under the assumption of bounded gradient diversity. From the practical view, (Zhang et al. 2016) suggests to communicate more frequently in the beginning of the training, and (Haddadpour et al. 2019a) verifies that using a geometrically increasing period does not harm the convergence notably.

Stagewise Training. For training both strongly convex and non-convex objectives, stagewisely decreasing the learning rate is widely adopted. Epoch-SGD (Hazan and Kale 2014) and ASSG (Xu, Lin, and Yang 2017) use SGD as their subalgorithm and geometrically decrease the learning rate stage by stage. They are proved to achieve the optimal $O(1/T)$ convergence for stochastic strongly convex optimization. For training neural networks, stagewisely decreasing the learning rate (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016) is a very important trick. From a theoretical aspect, stagewise SGD is proved with $O(1/\sqrt{T})$ convergence for both general and composite non-convex objectives (Allen-Zhu 2018; Chen et al. 2019; Davis and Grimmer 2019), by adopting SGD to optimize a regularized objective at each stage and decreasing the learning rate linearly stage by stage. Stagewise training is also verified to achieve better testing error than general SGD (Yuan et al. 2019).

Large Batch SGD (LB-SGD). SyncSGD with extremely large batch is proved to achieve a linear speedup with respect to the batch size (Stich and Karimireddy 2019). Nevertheless, (Jain et al. 2016) shows that increasing the batch size does not help when the bias dominates the variance. It is also observed from practice that LB-SGD leads to a poor generalization (Keskar et al. 2016; Golmant et al. 2018; Yin et al. 2017). (Yu and Jin 2019) proposes CR-PSGD which increases the batch size geometrically step by step and proves that CR-PSGD achieves a linear speedup with $O(\log T)$ communication complexity. However, after a large number of iterations, CR-PSGD essentially becomes GD and loses the benefit of SGD.

Local SGD with Variance Reduction. Recently, several techniques are proposed to reduce the communication complexity of Local SGD in the Non-IID case. (Haddadpour et al. 2019b) shows that using redundant data among clients yields lower communication complexity. One variant of Local SGD called VRL-SGD (Liang et al. 2019) incorporates the variance reduction technique and is proved to achieve a $O(N^{\frac{3}{2}}T^{\frac{1}{2}})$ communication complexity for non-convex objectives. SCAFFOLD (Karimireddy et al. 2019) extends VRL-SGD by involving two separate learning rates, and is proved to achieve $O(\log{(NT)})$ and $O(N^{\frac{1}{2}}T^{\frac{1}{2}})$ communication complexities for strongly convex objectives and non-

Algorithms	Objectives	Convergence Rate	Communication Complexity	Data Distributions	Extra Assumptions
Local SGD (Stich 2019)	Strongly Convex	$O(\frac{1}{NT})$	$O(N^{\frac{1}{2}}T^{\frac{1}{2}})$	IID	(1)
Local SGD (Stich and Karimireddy 2019) [1]	Strongly Convex	$O(\frac{\log T}{NT})$	$O(N\log(NT))$	IID	No
STL-SGD	**Strongly Convex**	$\boldsymbol{O(\frac{1}{NT})}$	$\boldsymbol{O(N\log T)}$	**IID**	**No**
Local SGD (Li et al. 2020)	Strongly Convex	$O(\frac{k^2}{NT})$	$O(T)$	Non-IID	(1)
Local SGD (Karimireddy et al. 2019) [1]	Strongly Convex	$O(\frac{\log T}{NT})$	$O(N^{\frac{1}{2}}T^{\frac{1}{2}})$	Non-IID	No
SCAFFOLD (Karimireddy et al. 2019) [1]	Strongly Convex	$O(\frac{\log T}{NT})$	$O(\log(NT))$	Non-IID	No
STL-SGD	**Strongly Convex**	$\boldsymbol{O(\frac{1}{NT})}$	$\boldsymbol{O(N^{\frac{1}{2}}T^{\frac{1}{2}})}$	**Non-IID**	**No**
Local SGD (Haddadpour et al. 2019a) [2]	Non-Convex+PL	$O(\frac{1}{NT})$	$O(N^{\frac{1}{3}}T^{\frac{1}{3}})$	IID	No
STL-SGD	**Non-Convex+PL**	$\boldsymbol{O(\frac{1}{NT})}$	$\boldsymbol{O(N\log T)}$	**IID**	**No**
STL-SGD	**Non-Convex+PL**	$\boldsymbol{O(\frac{1}{NT})}$	$\boldsymbol{O(N^{\frac{1}{2}}T^{\frac{1}{2}})}$	**Non-IID**	**No**
Local SGD (Wang and Joshi 2018b)	Non-Convex	$O(\frac{1}{\sqrt{NT}})$	$O(N^{\frac{3}{2}}T^{\frac{1}{2}})$	IID	(1)
STL-SGD	**Non-Convex**	$\boldsymbol{O(\frac{1}{\sqrt{NT}})}$	$\boldsymbol{O(N^{\frac{3}{2}}T^{\frac{1}{2}})}$	**IID**	**No**
Local SGD (Shen et al. 2019)	Non-Convex	$O(\frac{1}{\sqrt{NT}})$	$O(N^{\frac{3}{4}}T^{\frac{3}{4}})$	Non-IID	(2)
Local SGD (Haddadpour and Mahdavi 2019)	Non-Convex	$O(\frac{1}{\sqrt{NT}})$	$O(N^{\frac{3}{2}}T^{\frac{1}{2}})$	Non-IID	(3)
SCAFFOLD (Karimireddy et al. 2019)	Non-Convex	$O(\frac{1}{\sqrt{NT}})$	$O(N^{\frac{1}{2}}T^{\frac{1}{2}})$	Non-IID	No
STL-SGD	**Non-Convex**	$\boldsymbol{O(\frac{1}{\sqrt{NT}})}$	$\boldsymbol{O(N^{\frac{3}{4}}T^{\frac{3}{4}})}$	**Non-IID**	**No**

Table 1: A comparison of the results in this paper and previous state-of-the-art results of Local SGD and its variants. Regarding orders of convergence rate and communication complexity, we highlight the dependency on T (the number of iterations), N (the number of clients) and k (communication period). Previous results may depend on some extra assumptions, which include: (1) an upper bound for gradient, (2) an upper bound for the gradient variance among clients and (3) an upper bound for the gradient diversity, which are shown in the last column.

convex objectives respectively. As SCAFFOLD adopts a small fixed learning rate, its convergence rate for strongly convex objectives is $O(\frac{\log T}{NT})$. Nevertheless, these methods are orthogonal to our study. Combining STL-SGD and variance reduction to get better performance for the Non-IID case exceeds the scope of this paper.

Table 1 summarizes the comparison of Local SGD and its state-of-the-art extensions with STL-SGD. For both strongly convex objectives and non-convex objectives which satisfy the PL condition, STL-SGD achieves the state-of-the-art communication complexity while attaining the optimal convergence rate of $O(\frac{1}{NT})$. It is worth mentioning that Local SGD with momentum (Yu, Jin, and Yang 2019) or adaptive learning rate (Reddi et al. 2020) are orthogonal to our study.

Preliminaries

Notations and Definitions

Throughout the paper, we let $\|\cdot\|$ indicate the ℓ_2 norm of a vector and $\langle\cdot,\cdot\rangle$ indicate the inner product of two vectors. The set $\{1,2,\cdots,n\}$ is denoted as $[n]$. We use x^* to represent the optimal solution of (1). ∇f represents the gradient of f. $\mathbb{E}$ indicates a full expectation with respect to all the randomness in the algorithm (the stochastic gradients sampled in all iterations and the randomness in return).

The data distributions on different clients may not be identical. To quantify the difference of distributions, we define $\zeta_f^* := \frac{1}{N}\sum_{i=1}^N \|\nabla f_i(x^*)\|^2 = \frac{1}{N}\sum_{i=1}^N \|\nabla f_i(x^*) - \nabla f(x^*)\|^2$, which represents the variance of gradients among clients at x^*. Some literatures assume that the variance of gradients among clients is bounded by a constant ζ^2 (Shen et al. 2019) or the norm of stochastic gradients is bounded by a constant G^2 (Yu, Yang, and Zhu 2019; Li et al. 2020). Note that both ζ^2 and G^2 are larger than ζ_f^*. When the data distributions are identical, we have $\|\nabla f_i(x^*)\|^2 = 0$, thus it holds that $\zeta_f^* = 0$.

Due to the space limitation, all proofs are deferred to the full version of our paper[3]. To state the convergence of algorithms for solving (1), we introduce some commonly used definitions (Chen et al. 2019; Haddadpour et al. 2019a).

Definition 1 (ρ-weakly convex)**.** *A non-convex function $f(x)$ is ρ-weakly convex ($\rho > 0$) if*

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

Definition 2 (μ-Polyak-Łojasiewicz (PL))**.** *A function $f(x)$ satisfies the μ-PL condition ($\mu > 0$) if*

$$2\mu(f(x) - f(x^*)) \leq \|\nabla f(x)\|^2, \forall x \in R^d.$$

[1]Although these studies prove lower communication complexity, a suboptimal $O(\frac{\log T}{NT})$ convergence rate is proved due to the small fixed learning rate.

[2]The adaptive variant of Local SGD proposed in (Haddadpour et al. 2019a) has the same order of communication complexity as Local SGD.

[3]https://arxiv.org/abs/2006.06377.

Assumptions

Throughout this paper, we make the following assumptions, all of which are commonly used and basic assumptions (Stich 2019; Yu, Yang, and Zhu 2019; Li et al. 2020; Chen et al. 2019; Allen-Zhu 2018).

Assumption 1. *$f_i(x)$ is L-smooth in terms of $i \in [N]$ for every $x \in R^d$:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \forall x, y \in R^d, i \in [N].$$

Assumption 2. *There exists a constant σ such that*

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}\|\nabla f(x, \xi) - \nabla f_i(x)\|^2 \leq \sigma^2, \forall x \in R^d, \forall i \in [N].$$

Assumption 3. *If the objective function is non-convex, we assume it is ρ-weakly convex.*

Remark 1. *Note that if $f(x)$ is L-smooth, it is L-weakly convex. This is because Assumption 1 implies $-\frac{L}{2}\|x-y\|^2 \leq f(x)-f(y)-\langle\nabla f(y), x-y\rangle \leq \frac{L}{2}\|x-y\|^2$ (Nesterov 2018). Therefore, for an L-smooth function, we can immediately get that the weakly-convex parameter ρ satisfies $0 < \rho \leq L$.*

Review: Synchronous SGD with Periodically Averaging (Local SGD)

To alleviate the high communication cost in SyncSGD, the periodically averaging technique is proposed (Stich 2019; Yu, Yang, and Zhu 2019). Instead of averaging models in all clients at every iteration, Local SGD lets clients update their models locally for k iterations, then one communication is conducted to average the local models to make them consistent. Specifically, the update rule of Local SGD is

$$x_t^i = \begin{cases} \frac{1}{N}\sum_{j=1}^{N}(x_{t-1}^j - \eta\nabla f(x_{t-1}^j, \xi_{t-1}^j)), & \text{if } t \,\%\, k \ = \ 0, \\ x_{t-1}^i - \eta\nabla f(x_{t-1}^i, \xi_{t-1}^i), & \text{else}, \end{cases}$$

where x_t^i is the local model in client i at iteration t. Therefore, when each client conducts T iterations, the total number of communications is T/k. The complete procedure of Local SGD is summarized in Algorithm 1. Different from previous studies (McMahan et al. 2017; Stich 2019; Yu, Yang, and Zhu 2019), Algorithm 1 returns $\tilde{x} = \frac{1}{N}\sum_{i=1}^{N} x_t^i$ for a randomly chosen $t \in \{0, 1, \cdots, T-1\}$. In practice, we can determine t at first to avoid redundant iterations.

Although several studies have analysed the convergence of Local SGD, they assume that the objective $f(x)$ is μ-strongly convex or non-convex. (Khaled, Mishchenko, and Richtárik 2019) focuses on general convex objectives while they use the full gradient descent. Besides, most of the existing analysis relies on some stronger assumptions, including bounded gradient norm (i.e., $\|\nabla f_i(x, \xi)\|^2 \leq G^2$) (Stich 2019; Li et al. 2020) or bounded variance of gradients among clients (Shen et al. 2019). Here, we give a basic convergence result of Local SGD for the general convex objectives without these assumptions.

Theorem 1. *Suppose Assumptions 1 and 2 hold, $f(x)$ is convex and $\eta \leq \frac{1}{6L}$. If we set $k \leq \min\{\frac{1}{6\eta LN}, \frac{1}{9\eta L}\}$ and $k \leq \min\{\frac{\sigma}{\sqrt{6\eta LN(\sigma^2+4\zeta_f^*)}}, \frac{1}{9\eta L}\}$ for the IID case and the Non-IID case respectively, we have*

$$\mathbb{E}f(\tilde{x}) - f(x^*) \leq \frac{3\|x_0 - x^*\|^2}{4\eta T} + \frac{\eta\sigma^2}{N}. \quad (2)$$

Algorithm 1 Local-SGD(f, x_0, η, T, k)

Initialize: $x_0^i = x_0, \forall i \in [N]$.

```
for t = 1, ..., T do
    Client C_i does:
    Uniformly sample a mini-batch ξ^i_{t-1} ∈ D_i and calculate a stochastic gradient ∇f_i(x^i_{t-1}, ξ^i_{t-1}).
    if t divides k then
        Communicate with other clients and update: x^i_t = Σ_{j=1}^N (1/N)(x^j_{t-1} − η∇f(x^j_{t-1}, ξ^j_{t-1})).
    else
        Update locally: x^i_t = x^i_{t-1} − η∇f_i(x^i_{t-1}, ξ^i_{t-1}).
    end if
end for
return x̃ = (1/N) Σ_{i=1}^N x^i_t for the randomly chosen t ∈ {0, 1, ···, T − 1}.
```

Algorithm 2 STL-SGD^{sc}(f, x_1, η_1, T_1, k_1)

```
for s = 1, 2, ..., S do
    x_{s+1} = Local-SGD(f, x_s, η_s, T_s, max{⌊k_s⌋, 1}).
    Set η_{s+1} = η_s/2, T_{s+1} = 2T_s and
```

$$k_{s+1} = \begin{cases} \sqrt{2}k_s, & \text{Non-IID case}, \\ 2k_s, & \text{IID case}. \end{cases}$$

```
end for
return x_{S+1}.
```

Remark 2. *If we set $\eta = \sqrt{\frac{N}{T}}$, we have $\mathbb{E}f(\tilde{x}) - f(x^*) \leq \frac{\|x_0-x^*\|^2+\sigma^2}{\sqrt{NT}}$, which is consistent with the result of mini-batch SGD (Dekel et al. 2012).*

Local SGD with Stagewise Communication Period

To further reduce the communication complexity, we propose *ST*agewise *L*ocal *SGD* (STL-SGD) in this section with the following features.

- At the beginning, STL-SGD employs Algorithm 1 as a subalgorithm in each stage.
- Instead of using a small fixed learning rate or a gradually decreasing learning rate (e.g. $\frac{\eta_1}{1+\alpha t}$), STL-SGD adopts a stagewisely adaptive scheme. The learning rate is fixed at first, and decreased stage by stage.
- The communication periods are increased stagewisely.

We propose two variants of STL-SGD for strongly convex and non-convex problems, respectively.

STL-SGD for Strongly Convex Problems

In this subsection, we propose the STL-SGD algorithm for strongly convex problems, which is denoted as STL-SGD^{sc} and summarized in Algorithm 2. At each stage, the learning rate is decreased exponentially. In the meantime, the number of iterations and the communication period are increased exponentially. Specifically, at the s-th stage, we set $\eta_s = \frac{\eta_{s-1}}{2}$

and $T_s = 2T_{s-1}$. The communication period k_s is set as $k_s = 2k_{s-1}$ and $k_s = \sqrt{2}k_{s-1}$ for the IID case and the Non-IID case respectively.

Below, let x_s denote the initial point of the s-th stage. Theorem 2 establishes the convergence rate of STL-SGD^{sc}.

Theorem 2. *Suppose $f(x)$ is μ-strongly convex. Let $\eta_1 \leq \frac{1}{6L}$ and $T_1\eta_1 = \frac{6}{\mu}$. We set $k_1 = \min\{\frac{1}{6\eta_1 LN}, \frac{1}{9\eta_1 L}\}$ and $k_1 = \min\{\frac{\sigma}{\sqrt{6\eta_1 LN(\sigma^2+4\zeta_f)}}, \frac{1}{9\eta_1 L}\}$ for the IID case and the Non-IID case respectively. Under Assumptions 1 and 2, when the number of stages satisfies $S \geq \log(\frac{N(f(x_0)-f(x^*))}{\eta_1\sigma^2}) + 2$, we have the following result for Algorithm 2:*

$$\mathbb{E}f(x_{S+1}) - f(x^*) \leq \frac{9\eta_1\sigma^2}{2^S N}. \tag{3}$$

Defining $T := T_1 + T_2 + \cdots + T_S$, we have

$$\mathbb{E}f(x_{S+1}) - f(x^*) \leq O\left(\frac{1}{NT}\right). \tag{4}$$

Remark 3. *Theorem 2 claims the following properties of STL-SGD^{sc}:*

- ***Linear Speedup.*** *To reach a solution x_{S+1} with $\mathbb{E}f(x_{S+1}) - f(x^*) \leq \epsilon$, the number of iterations is $O(\frac{1}{N\epsilon})$, which indicates a linear speedup.*
- ***Communication Complexity for the Non-IID Case.*** *For the Non-IID case, we set $k_{s+1} = \sqrt{2}k_s$ for Algorithm 2. Therefore, the total communication complexity is $\frac{T_1}{k_1} + \cdots + \frac{T_S}{k_S} = \frac{T_1}{k_1}(1 + 2^{\frac{1}{2}} + \cdots + 2^{\frac{S-1}{2}}) = O(\frac{T_1}{k_1}\cdot(\frac{T}{T_1})^{\frac{1}{2}}) = O(N^{\frac{1}{2}}T^{\frac{1}{2}})$, where the last equality holds because $\frac{T_1^{\frac{1}{2}}}{k_1} = O(\sqrt{T_1\eta_1 N}) = O(N^{\frac{1}{2}})$.*
- ***Communication Complexity for the IID Case.*** *If the data distributions on different clients are identical, we set $k_{s+1} = 2k_s$ for Algorithm 2. Thus, the total communication complexity is $\frac{T_1}{k_1} + \cdots + \frac{T_S}{k_S} = S\frac{T_1}{k_1} = O(N\log T)$.*

STL-SGD for Non-Convex Problems

In this subsection, we proceed to propose the variant of STL-SGD algorithm for non-convex problems (STL-SGD^{nc}). Different from Algorithm 2, which optimizes a fixed objective during all stages, STL-SGD^{nc} changes the objective once a stage is finished. Specifically, in the s-th stage, the objective is a regularized problem $f^{\gamma}_{x_s} = f(x) + \frac{1}{2\gamma}\|x - x_s\|^2$, where x_s is the initial point of the s-th stage and γ is a constant that satisfies $\gamma < \rho^{-1}$. $f^{\gamma}_{x_s}(x)$ is guaranteed to be convex due to the ρ-weak convexity of $f(x)$. In this way, the theoretical property of Algorithm 1 under convex settings still holds in each stage of STL-SGD^{nc}. Other parameters are set in two different ways (**Option 1** and **Option 2**) for non-convex objectives satisfying the PL condition and otherwise, which are detailed in Algorithm 3.

In **Option 1**, we set η_s, T_s and k_s in the same way as in Algorithm 2. Here we analyse the theoretical property of STL-SGD^{nc} with **Option 1** for non-convex objectives that satisfy the PL condition.

Algorithm 3 STL-SGD^{nc}(f, x_1, η_1, T_1, k_1)

for $s = 1, 2, ..., S$ **do**
  Let $f^{\gamma}_{x_s}(x) = f(x) + \frac{1}{2\gamma}\|x - x_s\|^2$.
  x_{s+1} = Local-SGD($f^{\gamma}_{x_s}, x_s, \eta_s, T_s, \max\{\lfloor k_s \rfloor, 1\}$).
  Option 1: Set $\eta_{s+1} = \frac{\eta_s}{2}$, $T_{s+1} = 2T_s$ and

$$k_{s+1} = \begin{cases} \sqrt{2}k_s, & \text{Non-IID case,} \\ 2k_s, & \text{IID case.} \end{cases}$$

  Option 2: Set $\eta_{s+1} = \frac{\eta_1}{s+1}$, $T_{s+1} = (s+1)T_1$ and

$$k_{s+1} = \begin{cases} \sqrt{s+1}k_1, & \text{Non-IID case,} \\ (s+1)k_1, & \text{IID case.} \end{cases}$$

end for
return x_{S+1}.

Theorem 3. *Assume $f(x)$ satisfies the PL condition defined in Definition 2 with constant μ. Suppose Assumptions 1, 2 and 3 hold and $f(x)$ is weakly convex with constant $\rho \leq \frac{\mu}{16}$. Let $\eta_1 \leq \frac{1}{12L_\gamma}$, $T_1\eta_1 = \frac{6}{\rho}$. Set $k_1 = \min\{\frac{1}{6\eta_1 L_\gamma N}, \frac{1}{9\eta_1 L_\gamma}\}$ and $k_1 = \min\{\frac{\sigma}{\sqrt{6\eta_1 L_\gamma N(\sigma^2+4\zeta_f)}}, \frac{1}{9\eta_1 L_\gamma}\}$ for the IID case and the Non-IID case respectively. When the number of stages satisfies $S \geq \log\frac{N(f(x_0)-f(x^*))}{\eta_1\sigma^2} + 2$, Algorithm 3 with **Option 1** returns a solution x_{S+1} such that*

$$\mathbb{E}f(x_{S+1}) - f(x^*) \leq O\left(\frac{1}{NT}\right), \tag{5}$$

where $T = T_1 + T_2 + \cdots + T_S$.

Remark 4. *As the result of Theorem 3 is the same as that of Theorem 2, properties stated in Remark 3 all hold here.*

Option 2 is employed for the non-convex objectives which do not satisfy the PL condition. Instead of increasing the communication period geometrically as in **Option 1** of Algorithm 3, we let it increase in a linear manner, i.e., $k_s = sk_1$. Meanwhile, we increase the stage length linearly, that is $T_s = sT_1$, while keeping $T_s\eta_s$ a constant.

Theorem 4. *Suppose Assumptions 1, 2 and 3 hold. Let $\eta_1 \leq \frac{1}{6L_\gamma}$ and $T_1\eta_1 = \frac{3}{\rho}$. Set $k_1 = \min\{\frac{1}{6\eta_1 LN}, \frac{1}{9\eta_1 L}\}$ and $k_1 = \min\{\frac{\sigma}{\sqrt{6\eta_1 LN(\sigma^2+4\zeta_f)}}, \frac{1}{9\eta_1 L}\}$ for the IID case and the Non-IID case respectively. Algorithm 3 with **Option 2** guarantees that*

$$\mathbb{E}\|\nabla f(x_s)\|^2 \leq O\left(\frac{1}{\sqrt{NT}}\right), \tag{6}$$

where s is randomly sampled from $\{1, 2, \cdots, S\}$ with probability $p_s = \frac{s}{1+2+\cdots+S}$.

Remark 5. *STL-SGD^{nc} with **Option 2** has the following properties:*

- ***Linear Speedup:*** *To achieve $\mathbb{E}\|\nabla f(x_S)\|^2 \leq \epsilon$, the total number of iterations when N clients are used is $O(\frac{1}{N\epsilon^2})$, which shows a linear speedup.*

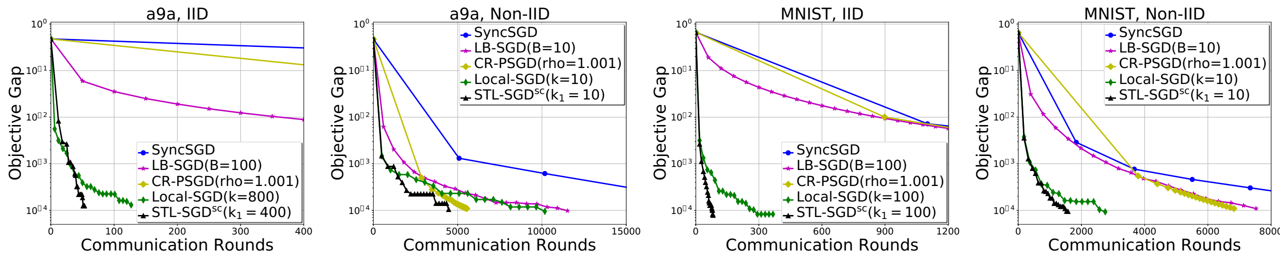

Figure 1: Training objective gap $f(x) - f(x^*)$ w.r.t the communication rounds for logistic regression on $\mathrm{a9a}$ and MNIST.

Algorithms	a9a (IID)	a9a (Non-IID)	MNIST (IID)	MNIST (Non-IID)
SyncSGD	100683 (1×)	90513 (1×)	32664 (1×)	22021 (1×)
LB-SGD	7620 (13.2×)	12221 (7.4×)	7011 (4.7×)	7740 (2.8×)
CR-PSGD	5434 (18.5×)	5772 (15.7×)	6788 (4.8×)	7029 (3.1×)
Local-SGD	184 (547.2×)	10068 (9.0×)	289 (113.0×)	2642 (8.3×)
STL-SGD^{sc}	**61 (1650.5×)**	**4417 (20.5×)**	**79 (413.5×)**	**1518 (14.5×)**

Table 2: Communication rounds to reach 10^{-4} objective gap in convex problems. We also show the speedup of these algorithms compared with SyncSGD.

- ***Communication Complexity for the Non-IID case:*** *Algorithm 3 with **Option 2** sets $k_s = \sqrt{s}k_1$. Thus, the communication complexity is $\frac{T_1}{k_1} + \frac{T_2}{k_2} + \cdots + \frac{T_S}{k_S} = \frac{T_1}{k_1}(1 + \sqrt{2} + \cdots + \sqrt{S}) = O(\frac{T_1}{k_1}(\frac{T}{T_1})^{\frac{3}{4}}) = O(N^{\frac{3}{4}}T^{\frac{3}{4}})$.*
- ***Communication Complexity for the IID case:*** *As $k_s = sk_1$, the communication complexity is $\frac{T_1}{k_1} + \frac{T_2}{k_2} + \cdots + \frac{T_S}{k_S} = \frac{T_1}{k_1}S = O(\frac{T_1}{k_1}(\frac{T}{T_1})^{\frac{1}{2}}) = O\left(N^{\frac{3}{2}}T^{\frac{1}{2}}\right)$.*

Experiments

We validate the performance of the proposed STL-SGD algorithm with experiments on both convex and non-convex problems. For each type of problems, we conduct experiments for both the IID case and the Non-IID case. Experiments are conducted on a machine with 8 Nvidia Geforce GTX 1080Ti GPUs and 2 Xeon(R) Platinum 8153 CPUs.

To simulate the Non-IID scenarios, we divide the training data and make the distributions of classes different among clients. Similar to the setting in (Karimireddy et al. 2019), at first, we randomly take $s\%$ i.i.d. data from the training set and divide them equally to each client. For the remaining data, we sort them according to their classes and then assign them to the clients in order. In our experiments, we set $s = 50$ for convex problems and $s = 0$ for non-convex problems.

We compare STL-SGD with SyncSGD, LB-SGD, CR-PSGD (Yu and Jin 2019) and Local SGD (Stich 2019). We show the comparison of these algorithms in terms of the communication rounds. The investigation regarding convergence is included in the full version of this paper[3], which validates that STL-SGD can achieve similar convergence rate as SyncSGD.

Convex Problems

We consider the binary classification problem with logistic regression, i.e.,

$$\min_{\theta \in R^d} \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T \theta)) + \frac{\lambda}{2}\|\theta\|^2, \qquad (7)$$

where $(x_i, y_i), i \in [n]$ constitute a set of training examples, and λ is the regularization parameter. It is notable that (7) is strongly convex when $\lambda > 0$, and we set $\lambda = 1/n$. We take two datasets $\mathrm{a9a}$ and MNIST from the libsvm website[4]. $\mathrm{a9a}$ has $32,561$ examples and 123 features. For MNIST, we sample a subset with $11,791$ examples and 784 features from two classes (4 and 9). Experiments are implemented on 32 clients and communication is handled with MPI[5].

SyncSGD, LB-SGD and Local SGD are implemented with the decreasing learning rate $\eta_t = \frac{\eta_1}{1+\alpha t}$ as suggested in (Stich 2019; Li et al. 2020) and we tune α in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ for the best performance. For STL-SGD^{sc}, we set $\eta_1 T_1 = \frac{1}{\lambda}$. The initial learning rate for all algorithms is tuned in $\{N, N/10, N/100\}$. The communication period k and the batch size B for LB-SGD are tuned in $\{100, 200, 400, 800, 1600\}$ for the IID case, and $\{10, 20, 40, 80, 160\}$ for the Non-IID case. The scaling factor of batch size ρ for CR-PSGD is tuned in $\{1.001, 1.01, 1.1\}$. We report the largest k, B and ρ which do not sacrifice the convergence for all algorithms.

Figure 1 shows the objective gap $f(x) - f(x^*)$ with regard to the communication rounds. We can observe that STL-SGD^{sc} converges with the fewest communication rounds for both the IID case and the Non-IID case. Although

[4] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[5] https://www.open-mpi.org/

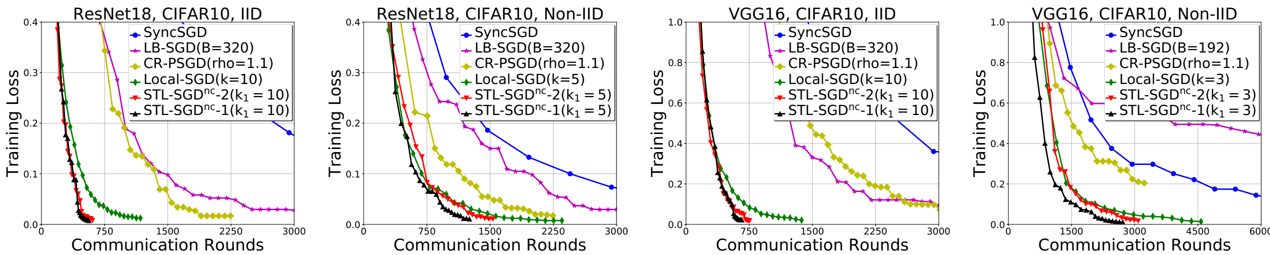

Figure 2: Training loss w.r.t the communication rounds for ResNet18 and VGG16 on CIFAR10.

Algorithms	ResNet18 (IID)	ResNet18 (Non-IID)	VGG16 (IID)	VGG16 (Non-IID)
SyncSGD	7644 (1×)	5390 (1×)	13622 (1×)	15092 (1×)
LB-SGD	3000 (2.5×)	3180 (1.7×)	– (–)	– (–)
CR-PSGD	1797 (4.3×)	1937 (2.8×)	– (–)	– (–)
Local-SGD	755 (10.1×)	1235 (4.4×)	1245 (10.9×)	3986 (3.8×)
STL-SGD^{nc}-2	**470 (16.3×)**	**1158 (4.7×)**	**696 (19.6×)**	**2732 (5.5×)**
STL-SGD^{nc}-1	**434 (17.6×)**	**954 (5.6×)**	**602 (22.6×)**	**2179 (6.9×)**

Table 3: Communication rounds to reach 99% training accuracy in non-convex problems. We run all algorithms for 200 epochs, where an epoch indicates one pass of the dataset. LB-SGD and CR-PSGD can not achieve 99% training accuracy on the VGG16 neural network until the end of training.

the initial communication period of STL-SGD^{sc} may need to be set smaller than Local SGD in the the IID case, the total number of communication rounds of STL-SGD^{sc} is still significantly lower, which validates that the communication complexity of STL-SGD^{sc} is much lower than Local SGD. As shown in Table 2, to achieve 10^{-4} objective gap, the communication rounds of STL-SGD^{sc} is almost 1.7-3 times fewer than Local SGD.

Non-Convex Problems

We train ResNet18 (He et al. 2016) and VGG16 (Simonyan and Zisserman 2014) on the CIFAR10 (Krizhevsky, Hinton et al. 2009) dataset, which includes a training set of 50,000 examples from 10 classes. 8 clients are used in total.

For our proposed algorithm, we denote STL-SGD^{nc} with **Option 1** and **Option 2** as STL-SGD^{nc}-1 and STL-SGD^{nc}-2 respectively. The learning rates of SyncSGD, LB-SGD, CR-PSGD and Local-SGD are all set fixed as suggested in their convergence theory (Ghadimi and Lan 2013; Yu and Jin 2019; Yu, Yang, and Zhu 2019). The initial learning rate for all algorithms is tuned in $\{N/10, N/100, N/1000\}$. The basic batch size at each client is 64. The first stage length of STL-SGD^{nc} is tuned in $\{20, 40, 60\}$ epochs. The parameter γ in STL-SGD^{nc} is tuned in $\{10^0, 10^2, 10^4\}$. We tune the communication period k in $\{3, 5, 10, 20\}$ and the batch size B for LB-SGD in $\{192, 320, 640, 1280\}$. For ease of implementation, we increase the batch size in CR-PSGD with $B = \rho B$ once an epoch is finished, and ρ is tuned in $\{1.1, 1.2, 1.3\}$. B stops growing when it exceeds 512 as suggested in (Yu and Jin 2019). We show the largest k and B which can maintain the same convergence rate as SyncSGD for all algorithms.

The experimental results of training loss regarding communication rounds are presented in Figure 2 and the communication rounds to achieve 99% training accuracy for all algorithms are shown in Table 3. As can be seen, STL-SGD^{nc}-1 and STL-SGD^{nc}-2 converge with much fewer communications than other algorithms. In spite of the same order of communication complexity as Local SGD, the performance of STL-SGD^{nc}-2 is better as the benefit of the negative relevance between the learning rate and the communication period. STL-SGD^{nc}-1 converges with the fewest number of communications, as it uses a geometrically increasing communication period.

Conclusion

We propose STL-SGD, which adopts a stagewisely increasing communication period to reduce the communication complexity. Two variants of STL-SGD (STL-SGD^{sc} and STL-SGD^{nc}) are provided for strongly convex objectives and non-convex objectives respectively. Theoretically, we prove that: (i) STL-SGD maintains the convergence rate and linear speedup as SyncSGD; (ii) when the objective is strongly convex or satisfies the PL condition, while attaining the optimal convergence rate $O(\frac{1}{NT})$, STL-SGD achieves the state-of-the-art communication complexity; (iii) when the objective is general non-convex, STL-SGD has the same communication complexity as Local SGD, while being more consistent with practical tricks. Experiments on both convex and non-convex problems demonstrate the effectiveness of the proposed algorithm.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61673364) and Anhui Provincial Natural Science Foundation (2008085J31). We would like to thank the Information Science Laboratory Center of USTC for the hardware and software services. We also gratefully acknowledge Xianfeng Liang from USTC for his valuable discussion.

References

Agarwal, A.; and Duchi, J. C. 2011. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, 873–881.

Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; and Vojnovic, M. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, 1709–1720.

Allen-Zhu, Z. 2018. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, 1157–1167.

Bayoumi, A. K. R.; Mishchenko, K.; and Richtarik, P. 2020. Tighter Theory for Local SGD on Identical and Heterogeneous Data. In *International Conference on Artificial Intelligence and Statistics*, 4519–4529.

Chen, Z.; Yuan, Z.; Yi, J.; Zhou, B.; Chen, E.; and Yang, T. 2019. Universal Stagewise Learning for Non-Convex Problems with Convergence on Averaged Solutions. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=Syx5V2CcFm.

Davis, D.; and Grimmer, B. 2019. Proximally guided stochastic subgradient method for nonsmooth, nonconvex problems. *SIAM Journal on Optimization* 29(3): 1908–1930.

Dekel, O.; Gilad-Bachrach, R.; Shamir, O.; and Xiao, L. 2012. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research* 13(Jan): 165–202.

Ghadimi, S.; and Lan, G. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23(4): 2341–2368.

Golmant, N.; Vemuri, N.; Yao, Z.; Feinberg, V.; Gholami, A.; Rothauge, K.; Mahoney, M. W.; and Gonzalez, J. 2018. On the Computational Inefficiency of Large Batch Sizes for Stochastic Gradient Descent. *arXiv preprint arXiv:1811.12941* .

Haddadpour, F.; Kamani, M. M.; Mahdavi, M.; and Cadambe, V. 2019a. Local SGD with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems*, 11080–11092.

Haddadpour, F.; Kamani, M. M.; Mahdavi, M.; and Cadambe, V. 2019b. Trading Redundancy for Communication: Speeding up Distributed SGD for Non-convex Optimization. In *International Conference on Machine Learning*, 2545–2554.

Haddadpour, F.; and Mahdavi, M. 2019. On the Convergence of Local Descent Methods in Federated Learning. *arXiv preprint arXiv:1910.14425* .

Hazan, E.; and Kale, S. 2014. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research* 15(1): 2489–2512.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jain, P.; Kakade, S. M.; Kidambi, R.; Netrapalli, P.; and Sidford, A. 2016. Parallelizing Stochastic Gradient Descent for Least Squares Regression: mini-batching, averaging, and model misspecification. *arXiv preprint arXiv:1610.03774* .

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977* .

Karimi, H.; Nutini, J.; and Schmidt, M. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 795–811. Springer.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2019. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378* .

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2016. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv preprint arXiv:1609.04836* .

Khaled, A.; Mishchenko, K.; and Richtárik, P. 2019. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715* .

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020. On the Convergence of FedAvg on Non-{IID} Data. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=HJxNAnVtDS.

Lian, X.; Huang, Y.; Li, Y.; and Liu, J. 2015. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, 2737–2745.

Liang, X.; Shen, S.; Liu, J.; Pan, Z.; Chen, E.; and Cheng, Y. 2019. Variance Reduced Local SGD with Lower Communication Complexity. *arXiv preprint arXiv:1912.12844* .

Lin, T.; Stich, S. U.; Patel, K. K.; and Jaggi, M. 2018. Don't Use Large Mini-Batches, Use Local SGD. *arXiv preprint arXiv:1808.07217* .

Lyu, L.; Yu, H.; and Yang, Q. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* .

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, 1273–1282.

Nesterov, Y. 2018. *Lectures on convex optimization*, volume 137. Springer.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2020. Adaptive Federated Optimization. *arXiv preprint arXiv:2003.00295* .

Shen, S.; Xu, L.; Liu, J.; Liang, X.; and Cheng, Y. 2019. Faster distributed deep net training: computation and communication decoupled stochastic gradient descent. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4582–4589. AAAI Press.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Stich, S. U. 2019. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=S1g2JnRcFX.

Stich, S. U.; Cordonnier, J.-B.; and Jaggi, M. 2018. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, 4447–4458.

Stich, S. U.; and Karimireddy, S. P. 2019. The Error-Feedback Framework: Better Rates for SGD with Delayed Gradients and Compressed Communication. *arXiv preprint arXiv:1909.05350* .

Tang, H.; Yu, C.; Lian, X.; Zhang, T.; and Liu, J. 2019. DoubleSqueeze: Parallel Stochastic Gradient Descent with Double-pass Error-Compensated Compression. In *International Conference on Machine Learning*, 6155–6165.

Wang, J.; and Joshi, G. 2018a. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *arXiv preprint arXiv:1810.08313* .

Wang, J.; and Joshi, G. 2018b. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576* .

Xu, Y.; Lin, Q.; and Yang, T. 2017. Stochastic convex optimization: Faster local growth implies faster global convergence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3821–3830. JMLR. org.

Yin, D.; Pananjady, A.; Lam, M.; Papailiopoulos, D.; Ramchandran, K.; and Bartlett, P. 2017. Gradient Diversity: a Key Ingredient for Scalable Distributed Learning. *arXiv preprint arXiv:1706.05699* .

Yu, H.; and Jin, R. 2019. On the Computation and Communication Complexity of Parallel SGD with Dynamic Batch Sizes for Stochastic Non-Convex Optimization. *arXiv preprint arXiv:1905.04346* .

Yu, H.; Jin, R.; and Yang, S. 2019. On the Linear Speedup Analysis of Communication Efficient Momentum SGD for Distributed Non-Convex Optimization. In *International Conference on Machine Learning*, 7184–7193.

Yu, H.; Yang, S.; and Zhu, S. 2019. 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, 5693–5700.

Yuan, Z.; Yan, Y.; Jin, R.; and Yang, T. 2019. Stagewise training accelerates convergence of testing error over SGD. In *Advances in Neural Information Processing Systems*, 2604–2614.

Zhang, J.; De Sa, C.; Mitliagkas, I.; and Ré, C. 2016. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365* .