Beyond Lipschitz Smoothness: A Tighter Analysis for Nonconvex Optimization

Zhengmian Hu [1 2] **Xidong Wu** [2] **Heng Huang** [1]

Abstract

Negative and positive curvatures affect optimization in different ways. However, a lot of existing optimization theories are based on the Lipschitz smoothness assumption, which cannot differentiate between the two. In this paper, we propose to use two separate assumptions for positive and negative curvatures, so that we can study the different implications of the two. We analyze the Lookahead and Local SGD methods as concrete examples. Both of them require multiple copies of model parameters and communication among them for every certain period of time in order to prevent divergence. We show that the minimum communication frequency is inversely proportional to the negative curvature, and when the negative curvature becomes zero, we recover the existing theory results for convex optimization. Finally, both experimentally and theoretically, we demonstrate that modern neural networks have highly unbalanced positive/negative curvatures. Thus, an analysis based on separate positive and negative curvatures is more pertinent.

1. Introduction

Lipschitz smoothness, which provides both upper and lower bounds for the Hessian matrix, is a common assumption for analyzing the convergence of optimization methods when the convexity is not guaranteed by the problem formulation. While this assumption is enough to establish convergence for many problems, it blurs certain structure of the underlying optimization problem by imposing symmetric upper bound and lower bounds for Hessian, which could lead to overly conservative convergence condition.

[1]Department of Computer Science, University of Maryland, College Park, MD, USA. [2]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA.. Correspondence to: Zhengmian Hu <huzhengmian@gmail.com>, Heng Huang <henghuanghh@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

In this paper, we propose to use separate upper and lower smoothness assumptions to guide the analysis of nonconvex optimization problems. New proof techniques are also developed under these new assumptions. We show that positive and negative eigenvalues of the Hessian matrix play different roles in the convergence analysis: positive curvature determines the maximum step size for gradient descent, while negative curvature controls the synchronization error when multiple parameters are optimized independently, thus controlling the minimum averaging frequency.

Based on this intuition, we derive tighter analysis via incorporating both upper and lower smoothness conditions for two recently popular optimization algorithms: Lookahead (Zhang et al., 2019) and Local SGD (Zinkevich et al., 2010; Konečný et al., 2016; McMahan et al., 2017; Stich, 2019). Compared to regular gradient descent, they both involve multiple copies of model parameters and require to average them from time to time to prevent exponential divergence. For Lookahead, we show that look-ahead too far could prevent convergence and the maximum horizon is bounded by the inverse negative curvature. Moreover, our analysis also captures the very rarely considered increasing Lookahead steps. We show that, when decreasing step size is used, Lookahead steps could simultaneously increase without hindering convergence. For Local SGD, we derive a better convergence condition than the previously known results and show that the minimum communication frequency is determined by the negative negative curvature. However, the requirement for linear-speed up is still the same.

Apart from using negative curvature upper bound as a precondition, we also explain why we can expect practical deep learning to enjoy milder negative curvature than positive curvature. Our experiments show that modern neural networks have very unbalanced positive and negative curvatures, making the Lipschitz smoothness assumption not tight. Thus, analysis based on separate positive and negative curvatures is more realistic. We further analyze theoretically why the negative curvature of neural networks is unbalanced, which is affected by network structure and loss function. We also establish an upper bound for the negative curvature based on first-order and zeroth-order information, which explains the change of negative curvature during training.

2. Related Works

Weakly convex. Our Assumption 4.3 is essentially the same as weakly convexity, which means the perturbed function $F(x) + \frac{L^-}{2}\|x\|^2$ is convex. Previous research works on weakly convexity (Davis et al., 2018; Davis & Drusvyatskiy, 2019; Zhu et al., 2019) mainly focus on non-smooth optimization, and the implication on communication complexity is still not clear. Chen et al. (2021) explores the distributed sub-gradient method for weakly convex problem, however, they don't show the connection between communication frequency and L^- because they use bounded gradient norm assumption to bypass the exponential divergence. More explanations can be found in Appendix A.

Lookahead. The Lookahead optimizer (Zhang et al., 2019), although is typically implemented as a serial algorithm, can be regarded as a two-agent optimization method (Wang et al., 2020b), where one agent minimizes the original objective function and the other agent optimizes a null objective $F(x) = 0$. Its generalization property has been studied via uniform stability in (Zhou et al., 2021).

Local SGD (Zinkevich et al., 2010; Konečný et al., 2016; McMahan et al., 2017; Stich, 2019) is a popular method to improve communication efficiency of parallel mini-batch SGD. The convergence of local SGD has been studied under convex (Khaled et al., 2020; Glasgow et al., 2022) and nonconvex (Yu et al., 2019b; Wang & Joshi, 2021; Jiang & Agrawal, 2018; Glasgow et al., 2022) settings. This type of algorithm has been generalized to various setup, including heterogeneous data (Khaled et al., 2020; Gorbunov et al., 2021; Woodworth et al., 2020), client sampling (McMahan et al., 2017; Yang et al., 2021), control variates (Karimireddy et al., 2020; Khanduri et al., 2021), momentum (Yu et al., 2019a; Wang et al., 2020a), quantization (Reisizadeh et al., 2020; Basu et al., 2019), adaptive step size (Xie et al., 2019; Reddi et al., 2021). However, all these works rely on Lipschitz smoothness. Our result is complimentary to these works by providing tighter analysis for vanilla method and draw connection between negative curvature and minimum communication frequency.

Second-order stationary point. Escaping from saddle points and finding local minima is widely considered as a central problem in nonconvex optimization. Various perturbed gradient methods (Ge et al., 2015; Jin et al., 2017; Li, 2019) and negative curvature descent methods (Xu et al., 2018; Allen-Zhu & Li, 2018; Fang et al., 2018; Zhou et al., 2018) have been developed to achieve second-order stationary point. However, these methods don't scale well into deep learning. Moreover, practical deep learning has been successful with SGD, which is only guaranteed to find the first-order stationary point. Our Theorem 7.2 helps explain this seemingly contradiction, by showing that for stochastic compositional optimization, where outer function is convex, and inner function is smooth, minimum Hessian eigenvalue is controlled by first-order and zeroth-order information, thus optimizing them naturally leads to second-order stationary point.

Eigenvalues of the Hessian. Imbalanced negative and positive curvatures have been observed in many empirical studies (Ghorbani et al., 2019; Sankar et al., 2020; Sagun et al., 2016). An open-source framework to compute Hessian information for DNNs by power iteration and stochastic Lanczos method was developed in (Yao et al., 2020).

Proximal point methods and Gauss-Newton methods. Proximal point algorithms have been developed to leverage the weak convexity or convexity of outer function in compositional optimization (Burke, 1985; Nesterov, 2007; Duchi & Ruan, 2018; Tran-Dinh et al., 2020; Gargiani et al., 2020). Li et al. (2020) study a multi-agent proximal method called FedProx. More discussions can be found in Appendix A.

3. Preliminary

In this paper, we consider the following optimization problem:

$$\min_{x\in\mathbb{R}^d} F(x) = \mathbb{E}_{\xi\sim\mathcal{D}}[F_\xi(x)]$$

Throughout the paper, we assume $F(x)$ is differentiable and the minimum exists. In some theorems, we also assume the variance of stochastic gradient to be bounded:

Assumption 3.1 (Bounded Variance)**.** There exist a $\sigma \geq 0$, such that for any $x \in \mathbb{R}^d$, we have

$$\mathbb{E}_{\xi\sim\mathcal{D}}[\|\nabla F_\xi(x) - \nabla F(x)\|^2] \leq \sigma^2.$$

4. Beyond Lipschitz Smoothness

The following Lipschitz smoothness condition is standard for analyzing smooth optimization:

Assumption 4.1 (Lipschitz smoothness)**.** $\forall x, y \in \mathbb{R}^d$, $\|\nabla F(y) - \nabla F(x)\| \leq L\|y - x\|$.

If $F(x)$ is twice differentiable, Lipschitz smoothness implies $-LI \preccurlyeq \nabla\nabla F \preccurlyeq LI$. Due to the symmetric upper and lower bounds, there is no way to distinguish the negative and positive curvatures. To derive tighter analysis, we decompose the smoothness condition into two parts:

Assumption 4.2 (Upper smoothness)**.** $\forall x, y \in \mathbb{R}^d$, $\langle \nabla F(y) - \nabla F(x), y - x\rangle \leq L^+\|y - x\|^2$.

Assumption 4.3 (Lower smoothness)**.** $\forall x, y \in \mathbb{R}^d$, $\langle \nabla F(y) - \nabla F(x), y - x\rangle \geq -L^-\|y - x\|^2$.

Clearly, Lipschitz smoothness implies upper and lower smoothness with $L^+ = L^- = L$ by definition. The reverse is also true:

Lemma 4.4. *Assumptions 4.2 and 4.3 implies Assumption 4.1 with* $L = \max(L^-, L^+)$.

These separate smoothness conditions allow us to derive tighter analysis when L^+ is different from L^-. We will show in Section 7 that for deep neural network, L^+ could be much larger than L^-, thus a symmetric bound by Lipschitz smoothness is not tight.

Moreover, each one of upper and lower smoothness is strictly weaker than the Lipschitz smoothness. This enforces us to differentiate clearly which curvature we are considering, and only use its own consequence but not the consequence of their both in analysis.

The main consequence of bounded positive curvature is the widely-known gradient descent lemma. Lemmas 5.2 and F.1 in our following analysis fall into this category.

On the contrary, the effects of negative curvature were less used explicitly in existing optimization literature. We show that there are two main ways to utilize the negative curvature. First, it gives an upper bound on the function value at average point minus the average of function values at a group of points. An illustration is shown in Figure B.1 (in Appendix). This effect is particularly relevant in the averaging among multiple copies of model parameter, as shown in Lemmas 5.1 and 6.6. Second, it induces exponential divergence for infinitesimally close trajectories. In order to see that, we can consider gradient flow $\frac{\mathrm{d}x}{\mathrm{d}t} = -\nabla F(x)$, and for two trajectories x and x', we have

$$\begin{aligned}\frac{\mathrm{d}\|x - x'\|^2}{\mathrm{d}t} =& -2\langle \nabla F(x) - \nabla F(x'), x - x' \rangle \\ \leq& 2L^- \|x - x'\|^2.\end{aligned}$$

An illustrative example is shown in Figure B.2 (in Appendix). For gradient descent, we derive a discrete version of the above inequality in Lemma 6.1.

5. Lookahead

Algorithm 1 Lookahead Algorithm

Input: Initial point z_0, outer/inner iteration number $T, \tau > 0$, outer/inner step size $\alpha, \gamma > 0$.
for $t = 0$ **to** $T - 1$ **do**
  $x_{t,0} = z_t$
  for $l = 0$ **to** $\tau - 1$ **do**
    Sample $\xi_{t,l} \sim \mathcal{D}$.
    $x_{t,l+1} = x_{t,l} - \gamma \nabla F_{\xi_{t,l}}(x_{t,l})$
  end for
  $z_{t+1} = z_t + \alpha(x_{t,\tau} - z_t)$
end for
Output: z_T.

The Lookahead algorithm with fixed step size γ and Lookahead steps τ is summarized in Algorithm 1. Two copies of weights, namely fast weights $x_{t,l}$ and slow weights z_t, are maintained in Lookahead. The slow weights are only updated every τ steps. Although practical Lookahead appears to be not sensitive to large τ, we show that there exists some situations where the convergence of Lookahead with large τ is not guaranteed by constructing a counterexample in Figure 1.

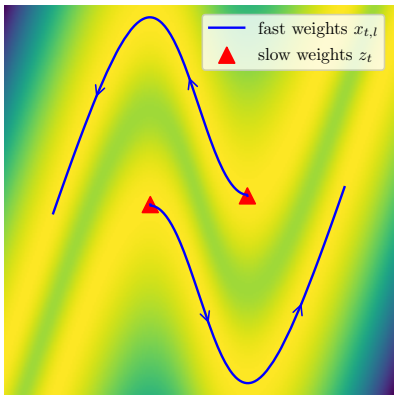


Figure 1: A counter example showing that Lookahead with too large horizon doesn't converge.

Thus, a natural question is: how far can we look ahead without breaking the convergence guarantee? Wang et al. (2020b) provided an upper bound for τ based on Lipschitz smoothness assumption, but their result is not tight. More specifically, their maximum horizon is controlled by Lipschitz smoothness constant $\gamma\tau = \mathcal{O}(1/L)$. In the following analysis, we improve that result into $\gamma\tau = \mathcal{O}(1/L^-)$. Please notice that L^- is always smaller than or equal to L, thus our result is tighter.

We next explain the main idea of our proof. First, we use the lower smoothness to control the "loss increase" due to the averaging.

Lemma 5.1. *Under Assumption 4.3, we have the following inequality for* $x_{t,\tau}$ *and* z_t *generated from Algorithm 1:*

$$\begin{aligned}F(z_{t+1}) - F(z_t) \leq& \alpha \left(F(x_{t,\tau}) - F(z_t)\right) \\ &+ \frac{L^-}{2}\alpha(1-\alpha)\|x_{t,\tau} - z_t\|^2\end{aligned}$$

We then apply the well-known gradient descent lemma as a consequence of upper smoothness.

Lemma 5.2. *Under Assumptions 3.1 and 4.2, we have following inequality for any* $l \geq 0$ *and* $x_{t,l}$ *generated from Algorithm 1:*

$$\begin{aligned}\mathbb{E}[F(x_{t,l+1}) - F(x_{t,l})] \leq& -\gamma(1 - \frac{\gamma L^+}{2})\mathbb{E}\|\nabla F(x_{t,l})\|^2 \\ &+ \frac{\gamma^2 L^+}{2}\sigma^2\end{aligned}$$

Combining the above two lemmas, we obtain the final convergence result. In order to simplify the presentation, we define two kinds of average in terms of expectation: $\mathbb{E}_t[\cdot] = \frac{1}{T}\sum_{t=0}^{T-1}(\cdot)$, and $\mathbb{E}_l[\cdot] = \frac{1}{\tau}\sum_{l=0}^{\tau-1}(\cdot)$.

Theorem 5.3. *Under Assumptions 3.1, 4.2 and 4.3, if $\gamma \leq (L^+ + (1-\alpha)L^-\tau)^{-1}$, we have the following inequality for Algorithm 1:*

$$\mathbb{E}_t\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 \leq \frac{2}{\alpha T\tau\gamma}\Big(F(z_0) - \min_z F(z)\Big) + (L^+ + (1-\alpha)L^-)\gamma\sigma^2. \quad (1)$$

We note that our analysis technique is different from the existing one. We control the "loss increase" coming from the averaging directly with lower smoothness, which only involves the negative curvature in Lemma 5.1. Wang et al. (2020b) instead consider an auxiliary average sequence y_k, and apply the gradient descent lemma for it, with gradient error being controlled by Lipschitz smoothness. This different point of view leads us to a tighter result than Wang et al. (2020b):

Corollary 5.4. *Under Assumptions 3.1, 4.2 and 4.3, for any $s < \frac{1}{(1-\alpha)L^-}$, and all small enough $\gamma \leq (1 - s(1-\alpha)L^-)/L^+$, the convergence result Eq. (1) in Theorem 5.3 holds with $\tau = s/\gamma$.*

In the Corollary 5.4, we define horizon as $s = \gamma\tau$ and we show that the maximum horizon is bounded by negative curvature, and is irrelevant to positive curvature. One interesting consequence is that for a convex loss function, we can look ahead arbitrarily far without worrying about convergence.

Corollary 5.5. *Under Assumptions 3.1, 4.2 and 4.3, for any $s < \frac{1}{(1-\alpha)L^-}$, we can define $K_0 = \left(\frac{sL^+}{1-s(1-\alpha)L^-}\right)^2$, such that we can find appropriate $\gamma = \mathcal{O}\Big(s/\sqrt{K}\Big)$,$\tau = \mathcal{O}\Big(\sqrt{K}\Big)$,$T = \mathcal{O}\Big(\sqrt{K}\Big)$ for all large enough $K \geq K_0$, that satisfy $\gamma\tau = s$ and $\tau T \leq K$, and*

$$\frac{1}{T\tau}\sum_{t=0}^{T-1}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 = \mathcal{O}\Big(1/\sqrt{K}\Big). \quad (2)$$

The $\mathcal{O}\Big(1/\sqrt{K}\Big)$ convergence rate is the same as the existing result. However, our result has a milder convergence condition $\gamma\tau L^- = \mathcal{O}(1)$.

5.1. Diminishing Learning Rate with Non-diminishing Horizon

When variance reduction techniques are not used, decreasing step-size is needed to tackle the noise from stochastic gradient. In a typical Lookahead optimizer, the inner loop steps are fixed, thus the horizon decreases with step size. However, we notice that even if we increase the inner loop steps, the convergence is still guaranteed if horizon doesn't exceed an upper bound related to negative curvature. The Lookahead with variable look ahead steps and step size is described in Algorithm 3 which is similar to Algorithm 1 and is moved to Appendix due to the limit of space.

Theorem 5.6. *Under Assumptions 3.1, 4.2 and 4.3, we define horizon for each outer iteration as $s_t = \gamma_t\tau_t$. For each given t, we define $\mathbb{E}_l[\cdot] = \frac{1}{\tau_t}\sum_{l=0}^{\tau_t-1}(\cdot)$. If $\gamma_t L^+ + (1-\alpha)L^- s_t \leq 1$ for all iterations $0 \leq t \leq T-1$, we have the following inequality for Algorithm 3:*

$$\mathbb{E}_t[s_t\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2] \leq \frac{2}{\alpha T}\Big(F(z_0) - \min_z F(z)\Big) + \mathbb{E}_t[s_t\gamma_t](L^+ + (1-\alpha)L^-)\sigma^2. \quad (3)$$

The following corollary shows that look ahead steps could be tuned according to the step size to ensure that the same horizon is used during training. The convergence is empirically verified in Figure H.1 (in Appendix). Notice that total iteration is $K = \mathcal{O}(T^2)$, therefore the final convergence rate is $\widetilde{\mathcal{O}}(1/T) = \widetilde{\mathcal{O}}(1/\sqrt{K})$.

Corollary 5.7. *Under the same assumptions and definition as Theorem 5.6, we additionally assume the horizons for each iteration are the same, such that $s_t = s < \frac{1}{(1-\alpha)L^-}$. By letting $\tau_t = (1+t)\frac{sL^+}{1-s(1-\alpha)L^-}$ and $\gamma_t = \frac{s}{\tau_t}$, we have the following convergence result:*

$$\mathbb{E}_{t,l}\mathbb{E}\|\nabla F(x_{t,l})\|^2 \leq \frac{2}{\alpha T s}\Big(F(z_0) - \min_z F(z)\Big) + \frac{\ln(T+1)}{T}\gamma_0(L^+ + (1-\alpha)L^-)\sigma^2.$$

More generally, we derive the following convergence condition for arbitrary step size and horizon schedule. Even when we restrict to cases where inner loop steps are fixed, the following corollary is still tighter than the existing result.

Corollary 5.8. *Under Assumptions 3.1, 4.2 and 4.3, if for any $t > 0$, we have $\gamma_t L^+ + (1-\alpha)L^- s_t \leq 1$ and $\sum_{t=0}^{\infty} s_t = \infty$, $\sum_{t=0}^{\infty} s_t\gamma_t < \infty$, then we have the following convergence result for Algorithm 3:*

$$\liminf_{t\to\infty}\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 = 0,$$
$$\liminf_{t\to\infty}\min_{0\leq l\leq\tau_t-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 = 0.$$

5.2. Quadratic Case: Lookahead Further to Generalize Better

Previous discussion shows that we can choose a horizon as large as $s = \gamma\tau \leq \frac{1}{(1-\alpha)L^-}$ for Lookahead, without

loss of convergence guarantee. However, it is still not clear whether larger horizon is a good choice to have. To answer this question, we derive PAC-generalization bound for quadratic loss case and show that look ahead further could help generalization.

$$F_{\mathcal{S}}(x) = \frac{1}{|\mathcal{S}|}\sum_{x'\in\mathcal{S}}\frac{1}{2}(x-x')\Lambda(x-x'),$$
$$x^* = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} x, \quad \mathcal{S}\sim\mathcal{D}^n.$$

For a data distribution $\mathcal{D}$, the training dataset and quadratic loss are defined as above. We assume Λ to be positive definite. In order to choose a horizon as large as possible, we consider infinitesimal step size, such that gradient descent turns into gradient flow:

$$\frac{\mathrm{d}x_{t,l}}{\mathrm{d}l} = -\Lambda(x_{t,l}-x^*), x_{t,0} = z_t, z_{t+1} = \alpha x_{t,\tau}+(1-\alpha)z_t.$$

The dynamics of Lookahead for this quadratic loss can be derived as follows:

Lemma 5.9. *If z_0 is initialized from a standard Gaussian distribution, then z_t also follows Gaussian distribution with the following mean and covariance:*

$$\mu_t = (I-(\alpha\exp(-\Lambda\tau)+(1-\alpha)I)^t)x^*,$$
$$\Sigma_t = (\alpha\exp(-\Lambda\tau)+(1-\alpha)I)^{2t}.$$

We will follow the PAC-Bayesian approach to the generalization problem.

Theorem 5.10 (Alquier et al. (2016))**.** *Give a prior distribution π, for any positive λ and $\delta\in(0,1]$, with at least $1-\delta$ in the probability of of training samples $\mathcal{S}\sim\mathcal{D}^n$, for all distribution p_z, we have*

$$\mathbb{E}_{z\sim p_z}[\mathbb{E}_{\mathcal{S}'\sim\mathcal{D}^n}[F_{\mathcal{S}'}(z)]] \le \mathbb{E}_{z\sim p_z}[F_{\mathcal{S}}(z)] + \frac{1}{\lambda}\mathrm{KL}(p_z\|\pi) + \Psi_{\pi,\mathcal{D}}\left(\lambda, n, \ln\left(\frac{1}{\delta}\right)\right).$$

Notice that the last term $\Psi_{\pi,\mathcal{D}}\left(\lambda, n, \ln(\frac{1}{\delta})\right)$ in above bound doesn't depend on training, therefore we only need to study the KL divergence term.

Theorem 5.11. *If z_0 is initialized from a standard Gaussian distribution, we fix the total training time $K = T\tau$, then $\mathrm{KL}(\mathcal{N}(\mu_T,\Sigma_T)\|\mathcal{N}(0,I))$ is a strictly and monotonically increasing function for $T\in\mathbb{N}^+$.*

The above theorem indicates that for fixed training time, the shorter each inner loop is, the worse the generalization error we have. This result is different from existing generalization bound in (Zhou et al., 2021), because the paper (Zhou et al., 2021) builds on uniform stability with a main focus on effect of α, while our result rely on PAC-Bayesian generalization bound and mainly focus on effect of τ.

6. Local SGD

Algorithm 2 Local SGD

Input: Initial point z_0, outer/inner iteration number $T, \tau > 0$, step size $\gamma > 0$.
for $t = 0$ **to** $T-1$ **do**
  for each worker $i = 1$ **to** N **do in parallel**
    $x_{i,t,0} = z_t$
    for $l = 0$ **to** $\tau - 1$ **do**
      Sample $\xi_{i,t,l}\sim\mathcal{D}$.
      $x_{i,t,l+1} = x_{i,t,l} - \gamma\nabla F_{\xi_{i,t,l}}(x_{i,t,l})$
    end for
  end for
  $z_{t+1} = \frac{1}{N}\sum_{i=1}^N x_{i,t,\tau}$
end for
Output: z_T.

The procedure of local SGD is shown in Algorithm 2. We only consider identical data, such that the divergence between different nodes only comes from negative curvature and random fluctuation. A group of parallel workers maintain their own local weights $x_{i,t,l}$ and update them with SGD. These local weights are aggregated and averaged every τ iterations. Clearly, choosing a large τ reduces the communication frequency, and is desired in distributed machine learning problems. In this section, we provide tighter analysis for the minimum communication required for convergence. We relax the convergence condition from $\gamma\tau L = \mathcal{O}(1)$ into two separate conditions $\gamma\tau L^- = \mathcal{O}(1)$ and $\gamma L^+ = \mathcal{O}(1)$. The final convergence condition is summarized in Figure 2. The area surrounded by the red dash line represents our new convergence result.

6.1. Convergence with Linear Speedup

We first give a convergence result that is capable to demonstrate linear speedup, but comes with an artifact in the convergence condition. Due to the negative curvature, different local weights diverge exponentially fast:

Lemma 6.1. *Under Assumptions 3.1 and 4.3, for two different workers $i\neq j$ in Algorithm 2, we have:*

$$\begin{aligned}&\frac{1}{2}\mathbb{E}\|x_{i,t,l}-x_{j,t,l}\|^2\\ \le&\gamma L^-\sum_{l'=0}^{l-1}\mathbb{E}\|x_{i,t,l'}-x_{j,t,l'}\|^2+\gamma^2 l\sigma^2\\ &+\frac{1}{2}\gamma^2\sum_{l'=0}^{l-1}\mathbb{E}\|\nabla F(x_{i,t,l'})-\nabla F(x_{j,t,l'})\|^2. \quad (4)\end{aligned}$$

We define the average for all workers in the form of expecta-

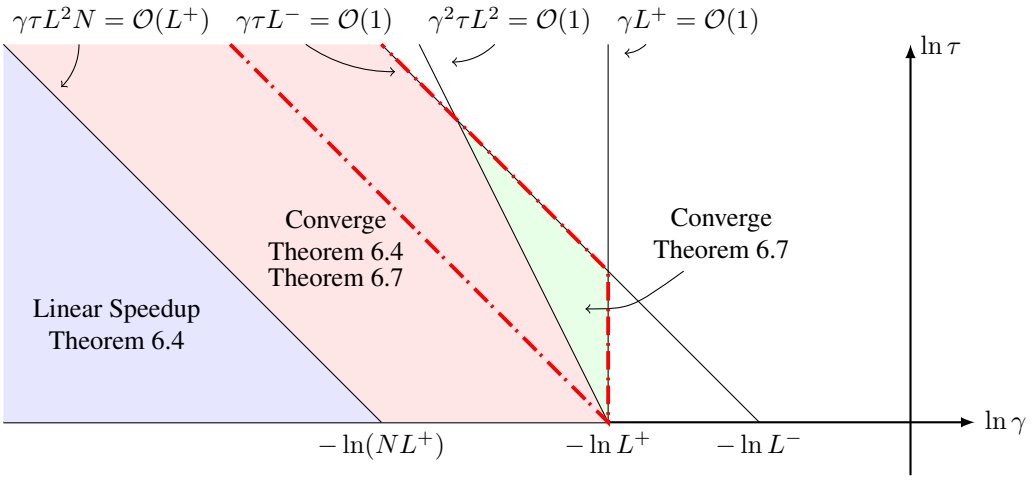


Figure 2: Different convergence conditions for local SGD.

tion: $\mathbb{E}_i[\cdot] = \frac{1}{N}\sum_{i=1}^N(\cdot)$. *Then we have:*

$$\begin{aligned}&\mathbb{E}_i\mathbb{E}\|x_{i,t,l} - \mathbb{E}_i[x_{i,t,l}]\|^2\\ \le& 2\gamma L^- \sum_{l'=0}^{l-1}\mathbb{E}_i\mathbb{E}\|x_{i,t,l'} - \mathbb{E}_i[x_{i,t,l'}]\|^2 + \gamma^2 l\frac{N-1}{N}\sigma^2\\ &+\gamma^2\sum_{l'=0}^{l-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l'}) - \mathbb{E}_i[\nabla F(x_{i,t,l'})]\|^2. \quad (5)\end{aligned}$$

The above lemma shows that exponential diverging rate is $2\gamma L^-$, thus to make sure different local weights stay close enough, we need to choose a small enough τ. Two conditions about τ are introduced in the following lemma. The first $\gamma(\tau-1) = \mathcal{O}(1/L^-)$ is only related to the negative curvature, and effectively terminates exponential divergence in its early stage. The second $\gamma^2L^2(\tau-1) = \mathcal{O}(1)$ is related to the Lipschitz smoothness constant L, which comes from the fact that we are controlling gradient error instead of parameter divergence. We introduce this technical condition to make sure a gradient variance term can be controlled. Details and proof can be found in Appendix F.1. This is an artifact, and can be avoided by a different analysis as we will show in Section 6.2.

Lemma 6.2. *Under Assumptions 3.1, 4.1 and 4.3, if* $\gamma L^-(\tau-1) \le \frac{1}{4}$ *and* $\gamma^2L^2(\tau-1)\le\frac{1}{2}$, *then*

$$\begin{aligned}&\sum_{l=0}^{\tau-1}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})] - \nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2\\ \le&\gamma^2L^2\tau(\tau-1)\frac{N-1}{N}\sigma^2. \quad (6)\end{aligned}$$

With the help of upper smoothness, we derive the following descent lemma for each outer loop.

Lemma 6.3. *Under the same assumptions as Lemma 6.2 and Assumption 4.2, we have:*

$$\begin{aligned}&\mathbb{E}[F(z_{t+1}) - F(z_t)]\\ \le& -\frac{\gamma}{2}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2\\ &+\frac{\gamma^2\tau}{2N}(L^+ + \gamma L^2(\tau-1)(N-1))\sigma^2\\ &+\frac{\gamma}{2}(-1+\gamma L^+)\sum_{l=0}^{\tau-1}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2. \quad (7)\end{aligned}$$

The final convergence result is a direct consequence of the above descent lemma.

Theorem 6.4. *Under Assumptions 3.1 and 4.1 to 4.3, if* $\gamma L^-(\tau-1)\le\frac{1}{4}$, $\gamma L^+\le 1$ *and* $\gamma^2L^2(\tau-1)\le\frac{1}{2}$, *then we have the following inequality for Algorithm 2:*

$$\begin{aligned}\mathbb{E}_{t,l}\mathbb{E}\|\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2 \le& \frac{2}{T\tau\gamma}\left(F(z_0) - \min_z F(z)\right) \quad (8)\\ &+\frac{\gamma(L^+ + \gamma L^2(\tau-1)(N-1))}{N}\sigma^2.\end{aligned}$$

Note that the coefficients for noise term contain two parts. L^+ comes from the positive curvature, and the $\gamma L^2(\tau-1)(N-1)$ term represents the extra noise due to the parameter aggregation. In order to achieve linear speed up, we need to make sure the second term doesn't overweight the first one. This essentially means $\gamma L^2(\tau-1) = \mathcal{O}(L^+/(N-1))$.

Since $L = \max(L^-, L^+)$, we know that linear speedup condition is simultaneously decided by both negative and positive curvatures. However, for modern deep learning, we typically have $L^+ \gg L^-$, thus $L = L^+$, and whether linear speedup is possible is mainly decided by positive curvature.

Corollary 6.5. *Under the same assumptions as Theorem 6.4, for large enough K, we can always find appropriate $\gamma = \mathcal{O}\left(\sqrt{\frac{N}{K}}\right)$, $\tau = \mathcal{O}\left(\frac{\sqrt{K}}{LN^{3/2}}\right)$, $T = \mathcal{O}\left(L\sqrt{K}N^{3/2}\right)$ such that*

$$\mathbb{E}_{t,l}\mathbb{E}\|\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2 = \mathcal{O}\left(\frac{F(z_0) - \min_z F(z) + L\sigma^2}{\sqrt{NK}}\right).$$

Although above linear speedup is ideal for scalability, it is difficult to obtain in practice because the required communication frequency $1/\tau$ needs to increase with the number of workers. When the communication overhead is the bottleneck, τ might be enlarged to improve the overall performance. However, it sacrifices linear speedup. In this case, our analysis gives tighter convergence conditions $\gamma L^-(\tau - 1) = \mathcal{O}(1)$ and $\gamma^2 L^2(\tau - 1) = \mathcal{O}(1)$. This compares favorably than previous results which require $\gamma L(\tau - 1) = \mathcal{O}(1)$.

6.2. Convergence without Linear Speedup

In this section, we show that $\gamma^2 L^2(\tau - 1) = \mathcal{O}(1)$ condition in the above analysis is an artifact and violating it doesn't necessarily lead to divergence. To prove that, we use lower smoothness to bound the difference in function value, instead of divergence between local parameters as in Lemma 6.1.

Lemma 6.6. *Under Assumption 4.3, we have the following inequality for $x_{t,\tau}$ and z_t generated from Algorithm 2:*

$$F(z_{t+1}) - F(z_t) \le \mathbb{E}_i[F(x_{i,t,\tau}) - F(z_t)] + \frac{L^-}{2}\mathbb{E}_i\|x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\|^2 \quad (9)$$

Based on the above lemma and gradient descent lemma F.1, which only depends on positive curvature, we obtain the following convergence result:

Theorem 6.7. *Under Assumptions 3.1, 4.2 and 4.3, if $\gamma \le (L^+ + L^-\tau)^{-1}$, we have the following inequality for Algorithm 2:*

$$\mathbb{E}_{t,l,i}[\mathbb{E}\|\nabla F(x_{i,t,l})\|^2] \le \frac{2}{T\tau\gamma}\left(F(z_0) - \min_z F(z)\right) + \gamma(L^+ + L^-)\sigma^2 \quad (10)$$

The above convergence condition effectively says $\gamma L^+ = \mathcal{O}(1)$ and $\gamma L^-\tau = \mathcal{O}(1)$. This is milder than Theorem 6.4 and completes the final picture in Figure 2.

7. Negative Curvature for Stochastic Compositional Optimization

Previous sections assume bounded negative curvature as a precondition, and show that if L^- is smaller than L^+, we have a tighter convergence condition. In this section, we instead ask how negative curvature looks like, and why we can expect the negative curvature to have smaller magnitude compared to positive curvature in practical deep learning. Answering these questions clearly requires more structure on the underlying loss function. Thus, we change our problem formulation to the following stochastic compositional optimization:

$$\min_{x\in\mathbb{R}^d} F(x) = \mathbb{E}_\xi[\phi_\xi(f_\xi(x))] \quad (11)$$

We assume that for any ξ, ϕ_ξ is convex and both $f_\xi : \mathbb{R}^d \to \mathbb{R}^{d'}$ and $\phi_\xi : \mathbb{R}^{d'} \to \mathbb{R}$ are differentiable. In a typical classification problem with deep neural network, ξ is the mini-batch including both inputs and outputs, $f_\xi(x)$ is the output logits, and ϕ_ξ is the cross-entropy loss *w.r.t.* ground truth label.

This compositional formulation is enough to explain why neural networks have imbalanced positive and negative curvature. In order to show that, we compare the Hessian spectrum for randomly initialized ResNet-18 (He et al., 2016) with cross-entropy loss and linear loss. As shown in Figure 3, convex cross-entropy loss induces a bias toward positive curvature. If linear loss is used, then the network has no preference for positive or negative curvature at initialization.

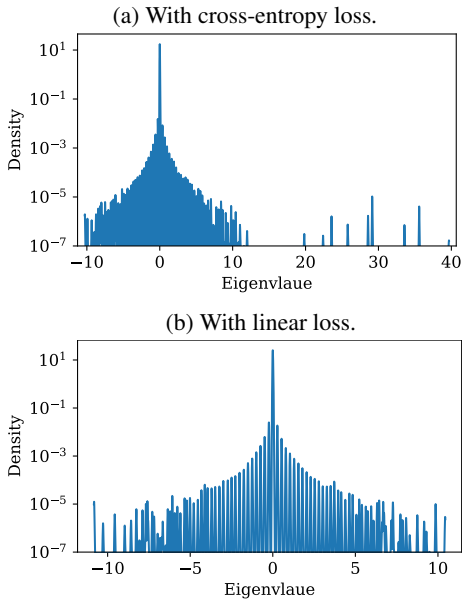


Figure 3: Estimated Hessian eigenspectrum for ResNet-18 at random initialization.

To provide an intuitive explanation, we assume that[1] both ϕ_ξ and f_ξ are deterministic, $d' = 1$ and ϕ is twice-

[1]These assumptions are solely for illustration and we do not rely on them in all theorems.

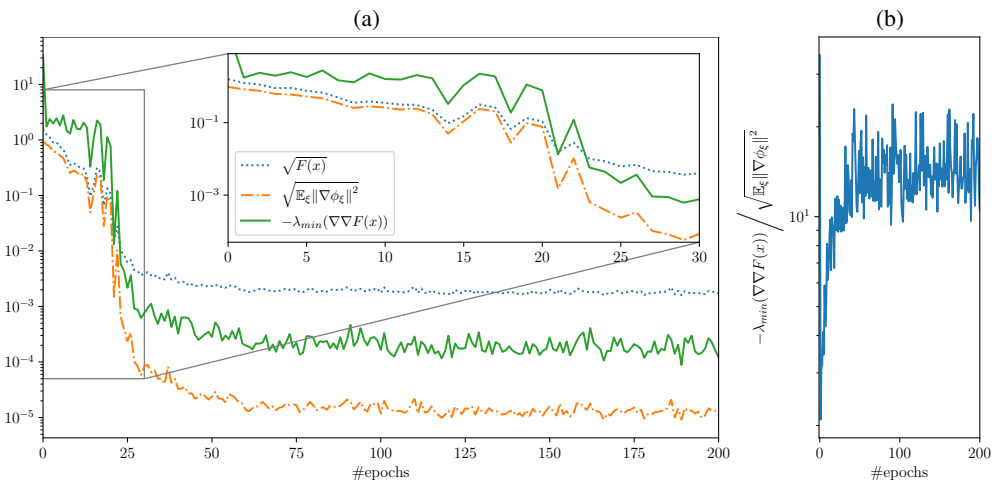


Figure 4: Dynamics of minimum Hessian eigenvalue, function value, and gradient during training.

differentiable, and the loss Hessian can be expressed as $\nabla^2 F = \nabla f^\top \phi'' \nabla f + \phi' \nabla^2 f$. Note that the first term is always positive semidefinite, as we assume ϕ to be convex. Therefore, the negative curvature can only stem from the second term.

In the following sections, we study the magnitude of negative curvature and how it depends on network definition and training. Following theorems solely focus on the bound of negative curvature, because the upper bound does not exhibit the same property and could potentially be much larger. We start with a simple global lower bound of negative curvature. We say a vector-valued function f is L_f-Lipschitz smooth under operator norm if $\forall x, x' \in \mathbb{R}^d$, $\|\nabla f(x) - \nabla f(x')\|_{\text{op}} \leq L_f \|x - x'\|$.

Theorem 7.1. *If for any ξ, the convex function $\phi_\xi(y)$ is G_ϕ-Lipschitz continuous and $f_\xi(x)$ is L_f-Lipschitz smooth under operator norm, then $F(x)$ satisfies lower smoothness as in Assumption 4.3 with $L^- = G_\phi L_f$.*

This theorem is not new (Davis et al., 2018), however, to the best of our knowledge, it is the first time it is used to explain why some networks have lighter negative curvature than others.

7.1. Control negative curvature with special networks

Theorem 7.1 tells us that the smoother network has lighter negative curvature. We verify that with two types of networks.

Wide network is known to be smoother (Allen-Zhu et al., 2019). Liu et al. (2020) show that operator norm of Hessian converges to 0 at infinite width limit. Thus, we increase the number of channels for ResNet-18 and measure the most severe negative curvature at initialization. It is shown in Figure 5a that wider network enjoys lighter negative curvature.

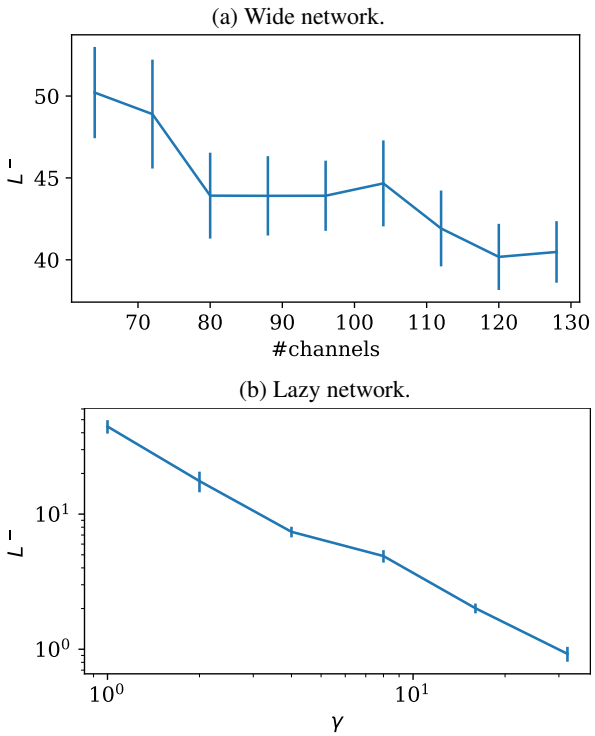


Figure 5: Estimated minimum negative curvature for different networks.

Another way to control negative curvature is related to the lazy regime (Chizat et al., 2019). If $f_\xi(x)$ is L_f-Lipschitz smooth, for any $\gamma > 0$, we can define $f_{\xi,\gamma}(x) = \gamma f_\xi(\frac{x}{\gamma})$. It is easy to verify that $f_{\xi,\gamma}(x)$ is also Lipschitz smooth with $L_{f,\gamma} = \gamma^{-1} L_f$. The numerical evaluation is shown in Figure 5b.

7.2. Control Negative Curvature with First-order or Zeroth-order Information

The unbalance of positive curvature and negative curvature not only exists in initialization but also increase during the neural network training as shown in Figure 6, indicating a global lower bound on negative curvature is not enough.

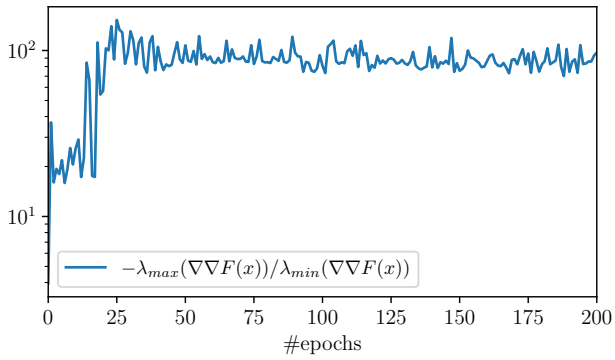


Figure 6: Dynamics of ratio of minimum and maximum Hessian eigenvalues.

In this section, we provide a novel local bound of negative curvature. We say a vector-valued function f is L_f-Lipschitz smooth under Frobenius norm if for any $x, x' \in \mathbb{R}^d$, we have $\|\nabla f(x) - \nabla f(x')\|_{\text{Fro}} \leq L_f \|x - x'\|$.

Theorem 7.2. *If for any ξ, f_ξ is L_f-Lipschitz smooth under Frobenius norm, and ϕ_ξ,f_ξ,F are all twice-differentiable, then we have inequality* (a) *shown below. Moreover, if we assume ϕ_ξ to be L_ϕ-Lipschitz smooth and $\min_y \phi_\xi(y) = 0$, then we also have inequality* (b).

$$\nabla\nabla F(x) \overset{\text{(a)}}{\succcurlyeq} -L_f \sqrt{\mathbb{E}_\xi \left\|\nabla \phi_\xi(y)|_{y=f_\xi(x)}\right\|^2} I \overset{\text{(b)}}{\succcurlyeq} -L_f \sqrt{2L_\phi F(x)} I.$$

The inequality (b) may not be tight, especially for cross-entropy, where the positive curvature is light at low loss region. Figure 4a shows that inequality (a) largely captures the dynamics of the negative curvature. The change of ratio shown in Figure 4b comes from changing smoothness of neural network f_ξ during training.

8. Conclusion

We propose to use separate smoothness assumptions for negative and positive curvatures in non-convex optimization theory to highlight their different implications. Minimum communication frequency is shown to only depend on negative curvature. This leads us to tighter convergence condition for Lookahead and Local SGD methods when negative curvature and positive curvature are imbalanced. We also show that for practical deep learning, due to the compositional loss with convex outer function, negative and positive curvatures are indeed imbalanced.

Acknowledgement

This work was partially supported by NSF IIS 1838627, 1837956, 1956002, 2211492, CNS 2213701, CCF 2217003, DBI 2225775.

References

Allen-Zhu, Z. and Li, Y. NEON2: finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, pp. 3720–3730, 2018.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Alquier, P., Ridgway, J., and Chopin, N. On the properties of variational approximations of gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.

Basu, D., Data, D., Karakus, C., and Diggavi, S. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32, 2019.

Burke, J. V. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, 1985.

Chen, S., Garcia, A., and Shahrampour, S. On distributed non-convex optimization: Projected subgradient method for weakly convex problems in networks. *IEEE Transactions on Automatic Control*, 2021.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Davis, D. and Drusvyatskiy, D. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Davis, D., Drusvyatskiy, D., MacPhee, K. J., and Paquette, C. Subgradient methods for sharp weakly convex functions. *Journal of Optimization Theory and Applications*, 179(3):962–982, 2018.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Duchi, J. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1):2325–5870, 2007.

Duchi, J. C. and Ruan, F. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization*, 28(4):3229–3259, 2018.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Gargiani, M., Zanelli, A., Diehl, M., and Hutter, F. On the promise of the stochastic generalized gauss-newton method for training dnns. *arXiv preprint arXiv:2006.02409*, 2020.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pp. 797–842. PMLR, 2015.

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

Glasgow, M. R., Yuan, H., and Ma, T. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pp. 9050–9090. PMLR, 2022.

Gorbunov, E., Hanzely, F., and Richtárik, P. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 3556–3564. PMLR, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Jiang, P. and Agrawal, G. A linear speedup analysis of distributed deep learning with sparse and quantized communication. *Advances in Neural Information Processing Systems*, 31, 2018.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020.

Khanduri, P., Sharma, P., Yang, H., Hong, M., Liu, J., Rajawat, K., and Varshney, P. Stem: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Konečný, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Li, Z. Ssrgd: Simple stochastic recursive gradient descent for escaping saddle points. *Advances in Neural Information Processing Systems*, 32, 2019.

Liu, C., Zhu, L., and Belkin, M. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33:15954–15964, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Nesterov, Y. Modified gauss-newton scheme with worst-case guarantees for its global performance. *Optimization Methods and Software*, 22, 2007.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020.

Sagun, L., Bottou, L., and LeCun, Y. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Sankar, A. R., Khasbage, Y., Vigneswaran, R., and Balasubramanian, V. N. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. *arXiv preprint arXiv:2012.03801*, 2020.

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

Tran-Dinh, Q., Pham, N., and Nguyen, L. Stochastic gauss-newton algorithms for nonconvex compositional optimization. In *International Conference on Machine Learning*, pp. 9572–9582. PMLR, 2020.

Wang, J. and Joshi, G. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22:1–50, 2021.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*, 2020a.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. Lookahead converges to stationary points of smooth non-convex functions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8604–8608. IEEE, 2020b.

Woodworth, B. E., Patel, K. K., and Srebro, N. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33: 6281–6292, 2020.

Xie, C., Koyejo, O., Gupta, I., and Lin, H. Local adaalter: Communication-efficient stochastic gradient descent with adaptive learning rates. *arXiv preprint arXiv:1911.09030*, 2019.

Xu, Y., Jin, R., and Yang, T. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in neural information processing systems*, 31, 2018.

Yang, H., Fang, M., and Liu, J. Achieving linear speedup with partial worker participation in non-IID federated learning. In *International Conference on Learning Representations*, 2021.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. W. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pp. 581–590. IEEE, 2020.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019a.

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

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhou, D., Xu, P., and Gu, Q. Stochastic nested variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhou, P., Yan, H., Yuan, X., Feng, J., and Yan, S. Towards understanding why lookahead generalizes better than sgd and beyond. *Advances in Neural Information Processing Systems*, 34, 2021.

Zhu, Z., Ding, T., Robinson, D., Tsakiris, M., and Vidal, R. A linearly convergent method for non-smooth non-convex optimization on the grassmannian with applications to robust subspace and dictionary learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Zinkevich, M., Weimer, M., Li, L., and Smola, A. Parallelized stochastic gradient descent. *Advances in neural information processing systems*, 23, 2010.

A. Comparison to related works

Chen et al. (2021) generalize convergence of the stochastic subgradient method for the non-smooth weakly convex problem (Davis et al., 2018; Davis & Drusvyatskiy, 2019) into distributed setting. They assume the norm of subgradient to be bounded as $\|\partial F\| \leq G$, which leads to following significant differences when compared to our analysis.

First, they use bounded gradient norm to bypass exponential divergence. Due to this assumption, even without communication, two local copies only diverge in a constant speed $\frac{\mathrm{d}}{\mathrm{d}t}\|\Delta_t\| = \frac{\mathrm{d}}{\mathrm{d}t}\|x_t - x_t'\| \leq 2G$, instead of exponentially $\frac{\mathrm{d}}{\mathrm{d}t}\|\Delta_t\| \leq L^-\|\Delta_t\|$. This makes their proof easier to establish, as no matter how frequent the communication is, the divergence between local copies are always bounded linearly by the step size, as shown in Lemma II.6 of Chen et al. (2021).

More importantly, this assumption blurs the effect of negative curvature. The main take away for this paper is that minimum communication complexity is controlled by negative curvature. This claim comes from a delicate balance between the convergence of local optimization and divergence among multiple parameter copies. However, they simplify the divergence between different parameter copies dramatically using bounded gradient norm assumption, thus cannot show the connection between negative curvature and communication requirement.

Li et al. (2020) consider federated learning under a proximal operator formulation. For each client, the following auxiliary optimization problem is approximated locally:

$$\min F_k(z; z_t) = F_k(z) + \frac{\mu}{2}\|z - z_t\|^2.$$

The distance penalty term μ restricts the local updates to be closer to the global weights. A separate Hessian lower bound is also introduced in their paper but due to a different reason than our paper, i.e. to ensure the above optimization is strongly convex such that the solution exists and is unique.

More importantly, the analysis of their paper is not tight and failed to show the distinct effects of positive and negative curvature. Specifically, their convergence condition in Theorem 4 (Li et al., 2020) is dominated by L instead of L^-. One consequence is that, even in convex case, $\mu \approx 0$ is not acceptable, as shown in Corollary 7 (Li et al., 2020). Instead, our analysis highlights the dependence on L^-, and gives tighter convergence condition. We show that the smaller L^- is, the less communication is required. In convex case, our theory holds for arbitrarily large horizon $s = \gamma\tau$.

B. Additional illustration for Section 4

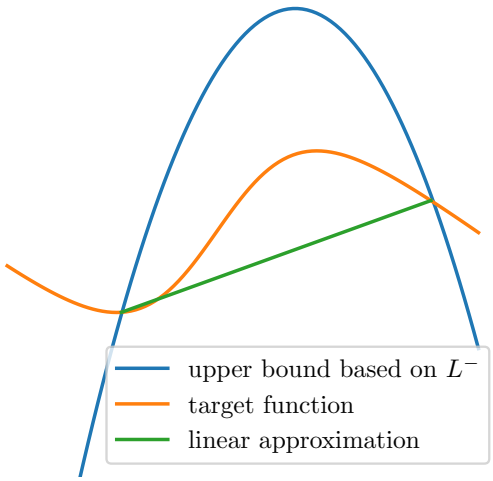
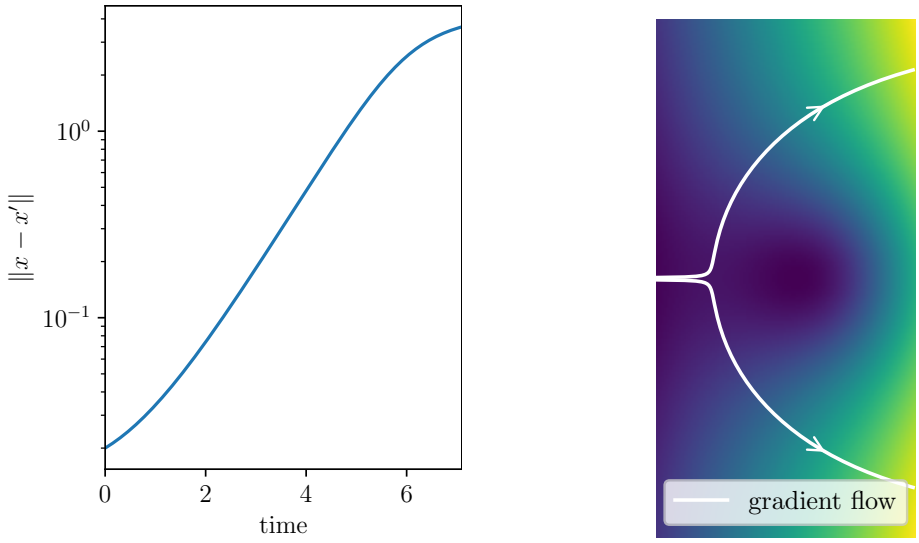
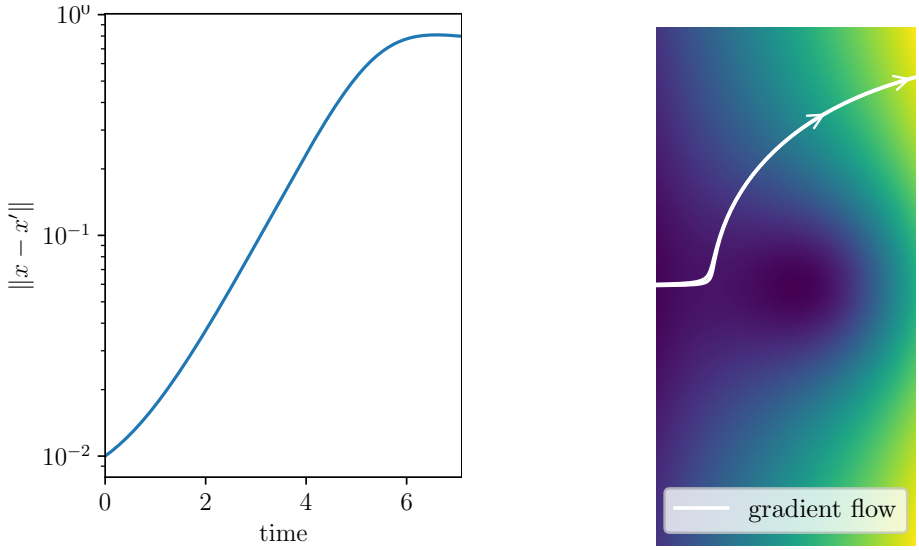


Figure B.1: Upper bound based on lower smoothness.



(a) (b)

Figure B.2: Exponential divergence for gradient flow on nonconvex objective function.



(a) (b)

Figure B.3: Similar to Figure B.2 but with different initialization.

C. Additional algorithm

Algorithm 3 Lookahead Method with Variable Step Size and Synchronization Period

Input: Initial point z_0, outer iteration number $T > 0$, outer step size $\alpha > 0$. For each iteration $0 \le t \le T-1$, the inner iteration number $\tau_t > 0$ and inner step size $\gamma_t > 0$.
for $t = 0$ **to** $T-1$ **do**
  $x_{t,0} = z_t$
  for $l = 0$ **to** $\tau_t - 1$ **do**
    Sample $\xi_{t,l} \sim \mathcal{D}$.
    $x_{t,l+1} = x_{t,l} - \gamma_t \nabla F_{\xi_{t,l}}(x_{t,l})$
  end for
  $z_{t+1} = z_t + \alpha(x_{t,\tau_t} - z_t)$
end for
Output: z_T.

D. Basic lemmas

Lemma D.1. *Under Assumptions 4.2 and 4.3, for any $x, y \in \mathbb{R}^n$, we have*

$$-\frac{L^-}{2}\|y-x\|^2 \le F(y) - F(x) - \langle \nabla F(x), y-x\rangle \le \frac{L^+}{2}\|y-x\|^2. \tag{12}$$

Proof of Lemma D.1. Since $F(x)$ is differentiable, we have

$$F(y) - F(x) = \int_0^1 \langle \nabla F(x + s(y-x)), y-x\rangle ds. \tag{13}$$

We can apply the upper smoothness as follows,

$$\begin{aligned} F(y) - F(x) - \langle \nabla F(x), y-x\rangle &= \int_0^1 \frac{1}{s}\langle \nabla F(x+s(y-x)) - \nabla F(x), s(y-x)\rangle ds && (14)\\ &\le \int_0^1 \frac{1}{s} L^+ \|s(y-x)\|^2 ds && (15)\\ &= \frac{1}{2} L^+ \|y-x\|^2. && (16) \end{aligned}$$

The lower smoothness condition can be used similarly.

$$\begin{aligned} F(y) - F(x) - \langle \nabla F(x), y-x\rangle &= \int_0^1 \frac{1}{s}\langle \nabla F(x+s(y-x)) - \nabla F(x), s(y-x)\rangle ds && (17)\\ &\ge -\int_0^1 \frac{1}{s} L^- \|s(y-x)\|^2 ds && (18)\\ &= -\frac{1}{2} L^- \|y-x\|^2. && (19) \end{aligned}$$

□

Proof of Lemma 4.4. We first relax Lemma D.1 with $L^+, L^- \le L$, such that for all $x, y \in \mathbb{R}^d$, we have

$$-\frac{L}{2}\|y-x\|^2 \le F(y) - F(x) - \langle \nabla F(x), y-x\rangle \le \frac{L}{2}\|y-x\|^2. \tag{20}$$

We now define $\Delta g = \nabla F(y) - \nabla F(x)$, and $\Delta x = y - x$. It is sufficient to show that $\|\Delta g\| \le L\|\Delta x\|$.

With $\bar{x} = \frac{x+y}{2}$, we apply Eq. (20) for $\bar{x} + \varepsilon\Delta g$ and $\bar{x} - \varepsilon\Delta g$ based on linear model at x:

$$\begin{aligned} F(\bar{x} + \Delta g) - F(x) - \langle \nabla F(x), \frac{\Delta x}{2} + \varepsilon\Delta g\rangle &\leq \frac{L}{2}\left\|\frac{\Delta x}{2} + \varepsilon\Delta g\right\|^2, \\ -\big(F(\bar{x} - \Delta g) - F(x) - \langle \nabla F(x), \frac{\Delta x}{2} - \varepsilon\Delta g\rangle\big) &\leq \frac{L}{2}\left\|\frac{\Delta x}{2} - \varepsilon\Delta g\right\|^2. \end{aligned}$$

Adding above two inequality gives

$$\begin{aligned} F(\bar{x} + \Delta g) - F(\bar{x} - \Delta g) - 2\varepsilon\langle \nabla F(x), \Delta g\rangle \leq& \frac{L}{2}\left\|\frac{\Delta x}{2} + \varepsilon\Delta g\right\|^2 + \frac{L}{2}\left\|\frac{\Delta x}{2} - \varepsilon\Delta g\right\|^2 \\ =& L\left(\left\|\frac{\Delta x}{2}\right\|^2 + \varepsilon^2\|\Delta g\|^2\right) \end{aligned}$$

Similarly, we apply Eq. (20) for $\bar{x} + \varepsilon\Delta g$ and $\bar{x} - \varepsilon\Delta g$ based on linear model at y:

$$\begin{aligned} F(\bar{x} - \Delta g) - F(y) - \langle \nabla F(y), -\frac{\Delta x}{2} - \varepsilon\Delta g\rangle &\leq \frac{L}{2}\left\|\frac{\Delta x}{2} + \varepsilon\Delta g\right\|^2, \\ -\big(F(\bar{x} + \Delta g) - F(y) - \langle \nabla F(y), -\frac{\Delta x}{2} + \varepsilon\Delta g\rangle\big) &\leq \frac{L}{2}\left\|\frac{\Delta x}{2} - \varepsilon\Delta g\right\|^2. \end{aligned}$$

Adding above two inequality gives

$$F(\bar{x} - \Delta g) - F(\bar{x} + \Delta g) + 2\varepsilon\langle \nabla F(y), \Delta g\rangle \leq L\left(\left\|\frac{\Delta x}{2}\right\|^2 + \varepsilon^2\|\Delta g\|^2\right)$$

Adding with inequality from linear model at x, we obtain

$$2\varepsilon\|\Delta g\|^2 = 2\varepsilon\langle \nabla F(y) - \nabla F(x), \Delta g\rangle \leq 2L\left(\left\|\frac{\Delta x}{2}\right\|^2 + \varepsilon^2\|\Delta g\|^2\right)$$

Letting $\varepsilon = \frac{1}{2L}$, we obtain

$$\frac{1}{L}\|\Delta g\|^2 \leq \frac{L}{2}\|\Delta x\|^2 + \frac{1}{2L}\|\Delta g\|^2$$

This implies $\|\Delta g\| \leq L\|\Delta x\|$. □

Lemma D.2. *For a differentiable function $F(x)$, if for any $x, y \in \mathbb{R}^d$, we have*

$$-\frac{L^-}{2}\|y - x\|^2 \leq F(y) - F(x) - \langle \nabla F(x), y - x\rangle \leq \frac{L^+}{2}\|y - x\|^2,$$

then we have

$$\begin{aligned} \langle \nabla F(y) - \nabla F(x), y - x\rangle \leq& \ L^+\|y - x\|^2 \\ \langle \nabla F(y) - \nabla F(x), y - x\rangle \geq& -L^-\|y - x\|^2. \end{aligned}$$

Proof of Lemma D.2. We first define $F^-(x) = F(x) + \frac{L^-}{2}\|x\|^2$. Then we have

$$\begin{aligned} & F^-(y) - F^-(x) - \langle \nabla F^-(x), y - x\rangle \\ =& F(y) - F(x) - \langle \nabla F(x), y - x\rangle + \frac{L^-}{2}\|y\|^2 - \frac{L^-}{2}\|x\|^2 - L^-\langle x, y - x\rangle \\ =& F(y) - F(x) - \langle \nabla F(x), y - x\rangle + \frac{L^-}{2}\|y - x\|^2 \\ \geq& 0. \end{aligned}$$

Thus $F^-(x)$ is a convex function and we have

$$\langle \nabla F^-(y) - \nabla F^-(x), y - x \rangle \geq 0. \tag{21}$$

By substituting the definition of $F^-(x)$, we obtain

$$\langle \nabla F(y) - \nabla F(x), y - x \rangle \geq -L^- \|y - x\|^2.$$

The upper smoothness can be proved similarly with a slightly different convex auxiliary function $F^+(x) = \frac{L}{2}\|x\|^2 - F(x)$. □

Lemma D.3. *If a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is differentiable and L-Lipschitz smooth under operator norm, then we have*

$$\|f(x') - f(x) - \langle \nabla f(x), x' - x \rangle\| \leq \frac{L}{2}\|x' - x\|^2.$$

Proof of Lemma D.3. Since f is differentiable, we have

$$\begin{aligned}
f(x') - f(x) &= \int_0^1 \langle \nabla f(x + s(x' - x)), x' - x \rangle ds, \\
f(x') - f(x) - \langle \nabla f(x), x' - x \rangle &= \int_0^1 \langle \nabla f(x + s(x' - x)) - \nabla f(x), x' - x \rangle ds, \\
\|f(x') - f(x) - \langle \nabla f(x), x' - x \rangle\| &= \left\| \int_0^1 \langle \nabla f(x + s(x' - x)) - \nabla f(x), x' - x \rangle ds \right\| \\
&\leq \int_0^1 \|\langle \nabla f(x + s(x' - x)) - \nabla f(x), x' - x \rangle\| ds \\
&\leq \int_0^1 \|\nabla f(x + s(x' - x)) - \nabla f(x)\|_{\text{op}} \|x' - x\| ds \\
&\leq \int_0^1 Ls\|x' - x\|^2 ds = \frac{L}{2}\|x' - x\|^2.
\end{aligned}$$

□

Lemma D.4. *For a random variable ξ, a vector $v_\xi \in \mathbb{R}^{d'}$ and a series of symmetric matrices $A_{\xi,i}$, we have*

$$\left(\mathbb{E}_\xi \sum_{i=1}^{d'} v_{\xi,i} A_{\xi,i} \right)^2 \preccurlyeq \left(\mathbb{E}_\xi \sum_{i=1}^{d'} A_{\xi,i}^2 \right) \left(\mathbb{E}_\xi \sum_{i=1}^{d'} v_{\xi,i}^2 \right).$$

Proof of Lemma D.4. We first define $\hat{v}_{\xi,i} = \frac{v_{\xi,i}}{\sqrt{\mathbb{E}_\xi \sum_{i=1}^{d'} v_{\xi,i}^2}}$. Then we have

$$0 \preccurlyeq \mathbb{E}_\xi \sum_{i=1}^{d'} \left(A_{\xi,i} - \hat{v}_{\xi,i} \mathbb{E}_\theta \sum_{j=1}^{d'} A_{\theta,j} \hat{v}_{\theta,j} \right)^2 = \mathbb{E}_\xi \sum_{i=1}^{d'} A_{\xi,i}^2 - \left(\mathbb{E}_\xi \sum_{i=1}^{d'} \hat{v}_{\xi,i} A_{\xi,i} \right)^2$$

We can multiply both side with $\mathbb{E}_\xi \sum_{i=1}^{d'} v_{\xi,i}^2$ and reorder the terms to obtain the final Cauchy–Schwarz inequality. □

Lemma D.5. *If a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is twice-differentiable and L-Lipschitz smooth under Frobenius norm, then we have*

$$\sum_{i=1}^{d'} (\nabla \nabla f_i)^2 \preccurlyeq L^2 I.$$

Proof of Lemma D.5. We first define $A = \sum_{k=1}^{d'} (\nabla\nabla f_k)^2$ and it follows

$$A_{i,i'} = \left(\sum_{k=1}^{d'} (\nabla\nabla f_k)^2 \right)_{i,i'} = \sum_{k=1}^{d'} \sum_{j=1}^{d} \nabla_{x_i} \nabla_{x_j} f_k \nabla_{x_j} \nabla_{x_{i'}} f_k$$

In order to prove $A \preccurlyeq L^2 I$, we just need to show that for every $v \in \mathbb{R}^d$, $v^T A v = \sum_{i,i'=1}^{d} v_i v_{i'} A_{i,i'} \leq L^2$. We will prove that by the fact that f is L-Lipschitz smooth under Frobenius norm:

$$\frac{\|\nabla f(x + sv) - \nabla f(x)\|_{\text{Fro}}}{s} \leq L.$$

By taking limit at $s \to 0$, we obtain

$$\left\| \sum_i v_i \nabla_{x_i} \nabla f \right\|_{\text{Fro}} \leq L.$$

We expand the definition of Frobenius norm to obtain

$$\begin{aligned} \left\| \sum_i v_i \nabla_{x_i} \nabla f \right\|_{\text{Fro}}^2 &= \sum_{k=1}^{d'} \sum_{j=1}^{d} \left(\sum_i v_i \nabla_{x_i} \nabla_{x_j} f_k \right)^2 \\ &= \sum_{i,i'=1}^{d} v_i v_{i'} A_{i,i'} \leq L^2. \end{aligned}$$

Notice the last equality use the fact that Hessian is symmetric: $\nabla_{x_i'} \nabla_{x_j} f = \nabla_{x_j} \nabla_{x_i'} f$. □

E. Lookahead

Proof of Lemma 5.1. We define a linear approximation of original function $F(y; x) = F(x) + \langle \nabla F(x), y - x \rangle$. According to Lemma D.1, we have

$$F(y) - F(y; x) \geq -\frac{L^-}{2} \|y - x\|^2. \tag{22}$$

We use the combine linearity of $F(y; x)$ and above inequality as follows,

$$\begin{aligned} F(z_{t+1}) =& F(z_{t+1}; z_{t+1}) \\ =& \alpha F(x_{t,\tau}; z_{t+1}) + (1 - \alpha) F(z_t; z_{t+1}) \\ \leq& \alpha \left(F(x_{t,\tau}) + \frac{L^-}{2} \|x_{t,\tau} - z_{t+1}\|^2 \right) + (1 - \alpha) \left(F(z_t) + + \frac{L^-}{2} \|z_t - z_{t-1}\|^2 \right) \\ =& \alpha \left(F(x_{t,\tau}) - F(z_t) \right) + \frac{L^-}{2} \left[\alpha(1 - \alpha)^2 + (1 - \alpha)\alpha^2 \right] \|x_{t,\tau} - z_t\|^2 + F(z_t) \\ =& \alpha \left(F(x_{t,\tau}) - F(z_t) \right) + \frac{L^-}{2} \alpha(1 - \alpha) \|x_{t,\tau} - z_t\|^2 + F(z_t) \end{aligned}$$

□

Proof of Lemma 5.2. According to Lemma D.1, we have

$$\begin{aligned} F(x_{t,l+1}) - F(x_{t,l}) \leq& \langle \nabla F(x_{t,l}), x_{t,l+1} - x_{t,l} \rangle + \frac{L^+}{2} \|x_{t,l+1} - x_{t,l}\|^2 \\ =& -\gamma \langle \nabla F(x_{t,l}), \nabla F_{\xi_{t,l}}(x_{t,l}) \rangle + \frac{\gamma^2 L^+}{2} \left\| \nabla F_{\xi_{t,l}}(x_{t,l}) \right\|^2 \end{aligned}$$

$$\begin{aligned}\mathbb{E}[F(x_{t,l+1}) - F(x_{t,l})] \leq & -\gamma\mathbb{E}\langle\nabla F(x_{t,l}), \nabla F_{\xi_{t,l}}(x_{t,l})\rangle + \frac{\gamma^2 L^+}{2}\mathbb{E}\left\|\nabla F_{\xi_{t,l}}(x_{t,l})\right\|^2 \\ = & -\gamma\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\gamma^2 L^+}{2}\mathbb{E}\left\|\nabla F_{\xi_{t,l}}(x_{t,l})\right\|^2 \\ \leq & -\gamma\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\gamma^2 L^+}{2}\left(\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \sigma^2\right)\end{aligned}$$

□

Proof of Theorem 5.3. By Lemma 5.2, we have

$$\mathbb{E}[F(x_{t,\tau}) - F(z_t)] \leq -\gamma(1 - \frac{\gamma L^+}{2})\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\gamma^2 L^+\tau}{2}\sigma^2.$$

In order to apply Lemma 5.1, we first control $\mathbb{E}\|x_{t,\tau} - z_t\|^2$ as follows

$$\begin{aligned}\mathbb{E}\|x_{t,\tau} - z_t\|^2 = & \mathbb{E}\left\|-\sum_{l=0}^{\tau-1}\gamma\nabla F_{\xi_{t,l}}(x_{t,l})\right\|^2 \leq \gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}\left\|\nabla F_{\xi_{t,l}}(x_{t,l})\right\|^2 \\ \leq & \gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \gamma^2\tau\sigma^2\end{aligned}$$

By Lemma 5.1, we have

$$\begin{aligned}\mathbb{E}[F(z_{t+1}) - F(z_t)] \leq & \alpha\mathbb{E}[F(x_{t,\tau}) - F(z_t)] + \frac{L^-}{2}\alpha(1-\alpha)\mathbb{E}\|x_{t,\tau} - z_t\|^2 \\ \leq & -\alpha\gamma(1 - \frac{\gamma L^+}{2})\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma^2 L^+\tau}{2}\sigma^2 \\ & + \frac{L^-}{2}\alpha(1-\alpha)\gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{L^-}{2}\alpha(1-\alpha)\gamma^2\tau\sigma^2 \\ = & -\alpha\gamma(1 - \frac{\gamma L^+}{2} - \frac{\gamma(1-\alpha)L^-\tau}{2})\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 \\ & + \frac{\alpha\gamma^2\tau}{2}(L^+ + (1-\alpha)L^-)\sigma^2\end{aligned}$$

If $\gamma(L^+ + (1-\alpha)L^-\tau) \leq 1$, we have

$$\mathbb{E}[F(z_{t+1}) - F(z_t)] \leq -\frac{\alpha\gamma}{2}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma^2\tau}{2}(L^+ + (1-\alpha)L^-)\sigma^2 \tag{23}$$

$$\mathbb{E}[F(z_T) - F(z_0)] \leq -\frac{\alpha\gamma}{2}\sum_{t=0}^{T-1}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma^2 T\tau}{2}(L^+ + (1-\alpha)L^-)\sigma^2$$

$$\frac{1}{T\tau}\sum_{t=0}^{T-1}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 \leq \frac{2}{\alpha T\tau\gamma}\left(F(z_0) - \min_z F(z)\right) + (L^+ + (1-\alpha)L^-)\gamma\sigma^2$$

□

Proof of Corollary 5.5. We may choose $\tau = \lceil\sqrt{K}\rceil$, $\gamma = \frac{s}{\tau}$, and $T = \lfloor\frac{K}{\tau}\rfloor$. In order to apply Theorem 5.3, we only need to ensure $\gamma L^+ + (1-\alpha)L^- s \leq 1$. This can be ensured by letting $K \geq K_0 = \left(\frac{sL^+}{1 - s(1-\alpha)L^-}\right)^2$ □

E.1. Diminishing learning rate with non-diminishing horizon

Proof of Theorem 5.6. Based on a similar argument as Theorem 5.3, we can reach following inequality:

$$\begin{aligned}\mathbb{E}[F(z_{t+1}) - F(z_t)] \leq & -\frac{\alpha\gamma_t}{2}\sum_{l=0}^{\tau_t-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma_t^2\tau_t}{2}(L^+ + (1-\alpha)L^-)\sigma^2 \\ = & -\frac{\alpha\gamma_t\tau_t}{2}\frac{1}{\tau_t}\sum_{l=0}^{\tau_t-1}\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma_t s_t}{2}(L^+ + (1-\alpha)L^-)\sigma^2 \\ = & -\frac{\alpha s_t}{2}\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\gamma_t s_t}{2}(L^+ + (1-\alpha)L^-)\sigma^2\end{aligned}$$

The first line in above inequality can simply obtained from Eq. (23) by changing γ and τ into γ_t and τ_t.

Then we compute the telescoping sum:

$$\mathbb{E}[F(z_T) - F(z_0)] \leq -\sum_{t=0}^{T-1}\frac{\alpha s_t}{2}\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 + \frac{\alpha\sum_{t=0}^{T-1}\gamma_t s_t}{2}(L^+ + (1-\alpha)L^-)\sigma^2$$

Recall that $F(z_T) - F(z_0) \geq \min_z F(z) - F(z_0)$. After reorder the terms, we have

$$\frac{\alpha T}{2}\mathbb{E}_t[s_t\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2] \leq \Big(F(z_0) - \min_z F(z)\Big) + \frac{\alpha T}{2}\mathbb{E}_t[s_t\gamma_t](L^+ + (1-\alpha)L^-)\sigma^2.$$

Dividing $\frac{\alpha T}{2}$ from both side of the above inequality gives the final result. □

Proof of Corollary 5.7. It is easy to check that $\gamma_t L^+ + (1-\alpha)L^- s_t = \frac{1+ts(1-\alpha)L^-}{1+t}$. Since $s(1-\alpha)L^- < 1$, we have $\gamma_t L^+ + (1-\alpha)L^- s_t \leq 1$ for all t. Therefore Theorem 5.6 applies. We just need to give to control $\mathbb{E}_t[\gamma_t]$ based on upper bound for harmonic series

$$\mathbb{E}_t[\gamma_t] = \frac{1}{T}\sum_{t=0}^{T-1}\gamma_t = \frac{1-s(1-\alpha)L^-}{L^+}\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{1+t} \leq \frac{1-s(1-\alpha)L^-}{L^+}\frac{\ln(1+T)}{T}.$$

□

Proof of Corollary 5.8. First, we define the total horizon as $S_T = \sum_{t=0}^{T-1} s_t$. Then, we have

$$\frac{1}{S_T}\sum_{t=0}^{T-1} s_t\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 \geq \min_t \mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 \geq \min_{t,l}\mathbb{E}\|\nabla F(x_{t,l})\|^2.$$

Due to Theorem 5.6, we have

$$\frac{1}{S_T}\sum_{t=0}^{T-1} s_t\mathbb{E}_l\mathbb{E}\|\nabla F(x_{t,l})\|^2 \leq \frac{2}{\alpha S_T}\Big(F(z_0) - \min_z F(z)\Big) + \frac{\sum_{t=0}^{T-1} s_t\gamma_t}{S_T}(L^+ + (1-\alpha)L^-)\sigma^2$$

According to our assumptions, the RHS of above inequality converge to 0 at limit of $T \to \infty$. □

E.2. Quadratic case: Lookahead further to generalize better

Proof of Lemma 5.9. First, we note that the gradient descent for quadratic loss can be solve exactly:

$$\begin{aligned}\frac{\mathrm{d}(x_{t,l} - x^*)}{\mathrm{d}l} &= -\Lambda(x_{t,l} - x^*), \\ x_{t,l} - x^* &= \exp(-\Lambda l)(x_{t,0} - x^*).\end{aligned}$$

Let $C = \exp(-\Lambda\tau)$, we then obtain the update rule for z_t:

$$z_{t+1} = (\alpha C + (1-\alpha)I)z_t + \alpha(1-C)x^*.$$

For simplicity, we define $A = \alpha C + (1-\alpha)I$, $B = \alpha(1-C)x^*$.

If $z_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, then we have

$$\mu_{t+1} = \mathbb{E}[z_{t+1}] = A\mathbb{E}[z_t] + B = A\mu_t + B,$$
$$\Sigma_{t+1} = \mathbb{E}[(z_{t+1} - \mu_{t+1})(z_{t+1} - \mu_{t+1})^T] = A\Sigma_t A.$$

Based on $\mu_0 = 0$, $\Sigma_0 = I$, we now derive the formula for μ_t and Σ_t:

$$\mu_t = A^t\mu_0 + \sum_{t'=0}^{t-1} A^{t'}B = (I - A^t)(I-A)^{-1}B = (I - A^t)x^*$$
$$\Sigma_t = A^t\Sigma_0 A^t = A^{2t}$$

Notice in the first line, we use the fact $(I-A)^{-1} = (\alpha(I-C))^{-1}$. □

Proof of Theorem 5.11. According to Duchi (2007), we can compute KL divergence as follows:

$$\mathrm{KL}(\mathcal{N}(\mu_T, \Sigma_T)\|\mathcal{N}(0, I)) = \frac{1}{2}\left(\|\mu_T\|^2 + \operatorname{Tr}\Sigma_T - \ln|\Sigma_T| - d\right)$$

Let λ_i be eigenvalues of Λ and x_i^* be the projection of x^* along eigenvectors of Λ, we have

$$\|\mu_T\|^2 = \sum_{i=1}^{d} x_i^{*2}(1 - ((1-\alpha) + \alpha\exp(-\lambda_i\tau))^T),$$
$$\operatorname{Tr}\Sigma_T - \ln|\Sigma_T| = \sum_{i=1}^{d}\left(((1-\alpha) + \alpha\exp(-\lambda_i\tau))^{2T} - 2T\ln(((1-\alpha) + \alpha\exp(-\lambda_i\tau)))\right).$$

Notice both values above are functions of $((1-\alpha) + \alpha\exp(-\lambda_i\tau))^T$, and $1-x$, $x^2 - \ln x^2$ are strictly monotonically decreasing with $0 < x < 1$, we only need to prove that $((1-\alpha) + \alpha\exp(-\lambda_i\tau))^T$ is strictly monotonically decreasing with respect to T.

$$((1-\alpha) + \alpha\exp(-\lambda_i\tau))^T = ((1-\alpha)1^{\frac{1}{T}} + \alpha(\exp(-\lambda_i K))^{\frac{1}{T}})^T$$

Since $\exp(-\lambda_i K) < 1$, the above Hölder mean strictly monotonically decrease with respect to T. □

F. Local SGD

F.1. Convergence with linear speedup

Proof of Lemma 6.1. We define $u_{i,j,t,l} = x_{i,t,l} - x_{j,t,l}$. Then we have

$$\begin{aligned}\langle u_{i,j,t,l+1} - u_{i,j,t,l}, u_{i,j,t,l}\rangle =& \frac{1}{2}\left[\|u_{i,j,t,l+1}\|^2 - \|u_{i,j,t,l}\|^2 - \|u_{i,j,t,l+1} - u_{i,j,t,l}\|^2\right] \\ =& \frac{1}{2}\left[\|u_{i,j,t,l+1}\|^2 - \|u_{i,j,t,l}\|^2\right] \\ & - \frac{1}{2}\gamma^2\left\|\nabla F_{\xi_{i,t,l}}(x_{i,t,l}) - \nabla F_{\xi_{j,t,l}}(x_{j,t,l})\right\|^2\end{aligned}$$

Due to Assumption 3.1, we have

$$\begin{aligned}\mathbb{E}\langle u_{i,j,t,l+1} - u_{i,j,t,l}, u_{i,j,t,l}\rangle \geq& \frac{1}{2}\left[\mathbb{E}\|u_{i,j,t,l+1}\|^2 - \mathbb{E}\|u_{i,j,t,l}\|^2\right] \\ & - \frac{1}{2}\gamma^2\mathbb{E}\|\nabla F(x_{i,t,l}) - \nabla F(x_{j,t,l})\|^2 - \gamma^2\sigma^2(1-\delta_{i,j})\end{aligned}$$

The term $\delta_{i,j}$ above equals 1 if and only if $i = j$, otherwise is 0.

By noting $u_{i,j,t,0} = 0$, we have following telescoping sum

$$\sum_{l'=0}^{l-1} \mathbb{E}\langle u_{i,j,t,l'+1} - u_{i,j,t,l'}, u_{i,j,t,l'}\rangle \geq \frac{1}{2}\mathbb{E}\|u_{i,j,t,l}\|^2 - \frac{1}{2}\gamma^2 \sum_{l'=0}^{l-1} \mathbb{E}\|\nabla F(x_{i,t,l'}) - \nabla F(x_{j,t,l'})\|^2 - \gamma^2 l\sigma^2(1-\delta_{i,j}) \tag{24}$$

On the other hand, based on Assumption 4.3, we have

$$\begin{aligned}\mathbb{E}\langle u_{i,j,t,l+1} - u_{i,j,t,l}, u_{i,j,t,l}\rangle =& -\gamma\mathbb{E}\langle \nabla F_{\xi_{i,t,l}}(x_{i,t,l}) - \nabla F_{\xi_{j,t,l}}(x_{j,t,l}), x_{i,t,l} - x_{j,t,l}\rangle \\ =& -\gamma\mathbb{E}\langle \nabla F(x_{i,t,l}) - \nabla F(x_{j,t,l}), x_{i,t,l} - x_{j,t,l}\rangle \\ \leq& \gamma L^- \mathbb{E}\|x_{i,t,l} - x_{j,t,l}\|^2 = \gamma L^- \mathbb{E}\|u_{i,j,t,l}\|^2\end{aligned} \tag{25}$$

Substituting Eq. (25) into Eq. (24) gives

$$\begin{aligned}\frac{1}{2}\mathbb{E}\|u_{i,j,t,l}\|^2 \leq& \gamma L^- \sum_{l'=0}^{l-1} \mathbb{E}\|u_{i,j,t,l'}\|^2 \\ &+ \frac{1}{2}\gamma^2 \sum_{l'=0}^{l-1} \mathbb{E}\|\nabla F(x_{i,t,l'}) - \nabla F(x_{j,t,l'})\|^2 + \gamma^2 l\sigma^2(1-\delta_{i,j})\end{aligned}$$

This proves Eq. (4). By taking expectation $\mathbb{E}_{i,j}[\cdot]$ for above inequality and substituting following terms, we obtain Eq. (5).

$$\begin{aligned}\frac{1}{2}\mathbb{E}_{i,j}\|u_{i,j,t,l}\|^2 = \frac{1}{2}\mathbb{E}_{i,j}\|x_{i,t,l} - x_{j,t,l}\|^2 = \mathbb{E}_i\|x_{i,t,l} - \mathbb{E}_i[x_{i,t,l}]\|^2, \\ \frac{1}{2}\mathbb{E}_{i,j}\|\nabla F(x_{i,t,l}) - \nabla F(x_{j,t,l})\|^2 = \mathbb{E}_i\|\nabla F(x_{i,t,l}) - \mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2, \\ \mathbb{E}_{i,j}[\delta_{i,j}] = \frac{1}{N}.\end{aligned}$$

□

Proof of Lemma 6.2.

$$\begin{aligned}&\sum_{l=0}^{\tau-1} \mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})] - \nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2 \\ =& \sum_{l=0}^{\tau-1} \mathbb{E}\mathbb{E}_i\|\nabla F(x_{i,t,l}) - \nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2 - \sum_{l=0}^{\tau-1} \mathbb{E}\mathbb{E}_i\|\nabla F(x_{i,t,l}) - \mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2 \\ \leq& L^2 \sum_{l=0}^{\tau-1} \mathbb{E}\mathbb{E}_i\|x_{i,t,l} - \mathbb{E}_i[x_{i,t,l}]\|^2 - \sum_{l=0}^{\tau-1} \mathbb{E}\mathbb{E}_i\|\nabla F(x_{i,t,l}) - \mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2\end{aligned} \tag{26}$$

We now control the first term based on Lemma 6.1.

$$\begin{aligned}
&\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|x_{i,t,l}-\mathbb{E}_i[x_{i,t,l}]\|^2\\
\leq& 2\gamma L^-\sum_{l=0}^{\tau-1}\sum_{l'=0}^{l-1}\mathbb{E}_i\mathbb{E}\|x_{i,t,l'}-\mathbb{E}_i[x_{i,t,l'}]\|^2\\
&+\gamma^2\sum_{l=0}^{\tau-1}\sum_{l'=0}^{l-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l'})-\mathbb{E}_i[\nabla F(x_{i,t,l'})]\|^2+\gamma^2\sum_{l=0}^{\tau-1}l\frac{N-1}{N}\sigma^2\\
\leq& 2\gamma L^-(\tau-1)\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|x_{i,t,l}-\mathbb{E}_i[x_{i,t,l}]\|^2\\
&+\gamma^2(\tau-1)\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})-\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2+\gamma^2\frac{\tau(\tau-1)}{2}\frac{N-1}{N}\sigma^2
\end{aligned}$$

If $2\gamma L^-(\tau-1)\leq\frac{1}{2}$, we have

$$\begin{aligned}
\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|x_{i,t,l}-\mathbb{E}_i[x_{i,t,l}]\|^2 \leq& 2\gamma^2(\tau-1)\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})-\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2\\
&+\gamma^2\tau(\tau-1)\frac{N-1}{N}\sigma^2
\end{aligned}$$

Substituting it into Eq. (26) give us

$$\begin{aligned}
&\sum_{l=0}^{\tau-1}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]-\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2\\
\leq&\gamma^2L^2\tau(\tau-1)\frac{N-1}{N}\sigma^2+(-1+2\gamma^2L^2(\tau-1))\sum_{l=0}^{\tau-1}\mathbb{E}\mathbb{E}_i\|\nabla F(x_{i,t,l})-\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2
\end{aligned}$$

According to our assumption, we have $2\gamma^2L^2(\tau-1)\leq 1$. therefore, the second term in above inequality can be dropped. This proves Eq. (6). □

Proof of Lemma 6.3. Based on Assumption 4.2, we commence with the gradient descent lemma or Lemma D.1:

$$\begin{aligned}
\mathbb{E}[F(\mathbb{E}_i[x_{i,t,l+1}])-F(\mathbb{E}_i[x_{i,t,l}])] \leq&\mathbb{E}\langle\nabla F(\mathbb{E}_i[x_{i,t,l}]),\mathbb{E}_i[x_{i,t,l+1}-x_{i,t,l}]\rangle\\
&+\frac{L^+}{2}\mathbb{E}\|\mathbb{E}_i[x_{i,t,l+1}-x_{i,t,l}]\|^2\\
=&-\gamma\mathbb{E}\langle\nabla F(\mathbb{E}_i[x_{i,t,l}]),\mathbb{E}_i[\nabla F_{\xi_{i,t,l}}(x_{i,t,l})]\rangle\\
&+\frac{\gamma^2L^+}{2}\mathbb{E}\left\|\mathbb{E}_i[\nabla F_{\xi_{i,t,l}}(x_{i,t,l})]\right\|^2\\
\leq&-\gamma\mathbb{E}\langle\nabla F(\mathbb{E}_i[x_{i,t,l}]),\mathbb{E}_i[\nabla F(x_{i,t,l})]\rangle\\
&+\frac{\gamma^2L^+}{2}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2+\frac{\gamma^2L^+}{2N}\sigma^2.
\end{aligned}$$

The last inequality follows the fact that $\xi_{i,t,l}$ and $\xi_{j,t,l}$ are independent for $i\neq j$. We next apply following substitution:

$$\begin{aligned}
\mathbb{E}\langle\nabla F(\mathbb{E}_i[x_{i,t,l}]),\mathbb{E}_i[\nabla F(x_{i,t,l})]\rangle =&\frac{1}{2}\mathbb{E}\|\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2+\frac{1}{2}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2\\
&-\frac{1}{2}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]-\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2.
\end{aligned}$$

Computing the telescoping sum, we have

$$\begin{aligned}\mathbb{E}[F(z_{t+1}) - F(z_t)] =& \mathbb{E}[F(\mathbb{E}_i[x_{i,t,\tau}]) - F(\mathbb{E}_i[x_{i,t,0}])] \\ \leq& -\frac{\gamma}{2}\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2 + \frac{\gamma^2 L^+ \tau}{2N}\sigma^2 \\ &+ \frac{\gamma}{2}(-1+\gamma L^+)\sum_{l=0}^{\tau-1}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})]\|^2 \\ &+ \frac{\gamma}{2}\sum_{l=0}^{\tau-1}\mathbb{E}\|\mathbb{E}_i[\nabla F(x_{i,t,l})] - \nabla F(\mathbb{E}_i[x_{i,t,l}])\|^2\end{aligned}$$

By applying Lemma 6.2, we obtain Eq. (7). □

Proof of Theorem 6.4. Actually, Eq. (8) is an immediate consequence of Eq. (7) by telescoping sum and noting that $-1+\gamma L^+ \leq 1$, $F(z_t) > \min_z F(z)$. □

Proof of Corollary 6.5. Given that $\gamma = \mathcal{O}\Big(\sqrt{\frac{N}{K}}\Big)$, $\tau = \mathcal{O}\Big(\frac{\sqrt{K}}{LN^{3/2}}\Big)$, $T = \mathcal{O}\Big(L\sqrt{K}N^{3/2}\Big)$, it is easy to check that for $K = \Omega\left(\max\left(\frac{L^2}{N}, {L^+}^2 N\right)\right)$, we have $\gamma L^-(\tau-1) = \mathcal{O}\Big(\frac{L^-}{LN}\Big) = \mathcal{O}(1)$, $\gamma L^+ = \mathcal{O}(1)$ and $\gamma^2 L^2(\tau-1) = \mathcal{O}(1)$. Therefore, with appropriate constant, Theorem 6.4 applies. □

F.2. Convergence without linear speedup

Proof of Lemma 6.6. According to Lemma D.1, for each worker i, we have

$$F(x_{i,t,\tau}) \geq F(\mathbb{E}_i[x_{i,t,\tau}]) + \langle \nabla F(\mathbb{E}_i[x_{i,t,\tau}]), x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\rangle - \frac{L^-}{2}\|x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\|^2$$

Taking average $\mathbb{E}_i[\cdot]$ for above inequality gives

$$\mathbb{E}_i[F(x_{i,t,\tau})] \geq F(\mathbb{E}_i[x_{i,t,\tau}]) - \frac{L^-}{2}\mathbb{E}_i\|x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\|^2$$

By reordering terms and substituting $\mathbb{E}_i[x_{i,t,\tau}] = z_{t+1}$, we obtain Eq. (9). □

Lemma F.1. *Under Assumptions 3.1 and 4.2, we have following inequality for any $l \geq 0$ and $x_{t,l}$ generated from Algorithm 2:*

$$\mathbb{E}[F(x_{i,t,l+1}) - F(x_{i,t,l})] \leq -\gamma(1 - \frac{\gamma L^+}{2})\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 L^+}{2}\sigma^2 \tag{27}$$

Proof of Lemma F.1. According to Lemma D.1, we have

$$\begin{aligned}F(x_{i,t,l+1}) - F(x_{i,t,l}) \leq& \langle \nabla F(x_{i,t,l}), x_{i,t,l+1} - x_{i,t,l}\rangle + \frac{L^+}{2}\|x_{i,t,l+1} - x_{i,t,l}\|^2 \\ =& -\gamma\langle \nabla F(x_{i,t,l}), \nabla F_{\xi_{i,t,l}}(x_{i,t,l})\rangle + \frac{\gamma^2 L^+}{2}\left\|\nabla F_{\xi_{i,t,l}}(x_{i,t,l})\right\|^2\end{aligned}$$

$$\begin{aligned}\mathbb{E}[F(x_{i,t,l+1}) - F(x_{i,t,l})] \leq& -\gamma\mathbb{E}\langle \nabla F(x_{i,t,l}), \nabla F_{\xi_{i,t,l}}(x_{i,t,l})\rangle + \frac{\gamma^2 L^+}{2}\mathbb{E}\left\|\nabla F_{\xi_{i,t,l}}(x_{i,t,l})\right\|^2 \\ =& -\gamma\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 L^+}{2}\mathbb{E}\left\|\nabla F_{\xi_{i,t,l}}(x_{i,t,l})\right\|^2 \\ \leq& -\gamma\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 L^+}{2}\left(\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \sigma^2\right)\end{aligned}$$

□

Proof of Theorem 6.7. By Lemma F.1, we have

$$\mathbb{E}[F(x_{i,t,\tau}) - F(z_t)] \leq -\gamma(1 - \frac{\gamma L^+}{2}) \sum_{l=0}^{\tau-1} \mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 L^+ \tau}{2}\sigma^2.$$

In order to apply Lemma 6.6, we first control $\mathbb{E}\|x_{i,t,\tau} - z_t\|^2$ as follows

$$\begin{aligned}
\mathbb{E}_i\mathbb{E}[\|x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\|^2] \leq& \mathbb{E}_i\mathbb{E}[\|x_{i,t,\tau} - z_t\|^2] = \mathbb{E}_i\mathbb{E}\left\|-\sum_{l=0}^{\tau-1}\gamma\nabla F_{\xi_{i,t,l}}(x_{i,t,l})\right\|^2 \\
\leq& \gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\left\|\nabla F_{\xi_{i,t,l}}(x_{i,t,l})\right\|^2 \\
\leq& \gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \gamma^2\tau\sigma^2
\end{aligned}$$

By Lemma 6.6, we have

$$\begin{aligned}
\mathbb{E}[F(z_{t+1}) - F(z_t)] \leq& \mathbb{E}_i[F(x_{i,t,\tau}) - F(z_t)] + \frac{L^-}{2}\mathbb{E}_i\|x_{i,t,\tau} - \mathbb{E}_i[x_{i,t,\tau}]\|^2 \\
\leq& -\gamma(1 - \frac{\gamma L^+}{2})\sum_{l=0}^{\tau-1}\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 L^+\tau}{2}\sigma^2 \\
&+ \frac{L^-}{2}\gamma^2\tau\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{L^-}{2}\gamma^2\tau\sigma^2 \\
=& -\gamma(1 - \frac{\gamma L^+}{2} - \frac{\gamma L^-\tau}{2})\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2\tau}{2}(L^+ + L^-)\sigma^2
\end{aligned}$$

If $\gamma(L^+ + L^-\tau) \leq 1$, we have

$$\begin{aligned}
\mathbb{E}[F(z_{t+1}) - F(z_t)] \leq& -\frac{\gamma}{2}\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2\tau}{2}(L^+ + L^-)\sigma^2 \\
\mathbb{E}[F(z_T) - F(z_0)] \leq& -\frac{\gamma}{2}\sum_{t=0}^{T-1}\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 + \frac{\gamma^2 T\tau}{2}(L^+ + L^-)\sigma^2 \\
\frac{1}{T\tau}\sum_{t=0}^{T-1}\sum_{l=0}^{\tau-1}\mathbb{E}_i\mathbb{E}\|\nabla F(x_{i,t,l})\|^2 \leq& \frac{2}{T\tau\gamma}\left(F(z_0) - \min_z F(z)\right) + \gamma(L^+ + L^-)\sigma^2
\end{aligned}$$

□

G. Negative curvature for stochastic compositional optimization

Proof of Theorem 7.1. We will show that for any ξ, $\phi_\xi(f_\xi(x))$ satisfies lower smoothness with $L^- = G_\phi L_f$.

We define $e = f_\xi(x') - f_\xi(x) - \langle\nabla f_\xi(x), x' - x\rangle$ and $y = f_\xi(x)$. Based on Lemma D.3, we have $\|e\| \leq \frac{L_f}{2}\|x' - x\|^2$. Moreover, due to convexity of ϕ_ξ, we have

$$\begin{aligned}
\phi_\xi(f_\xi(x')) - \phi_\xi(f_\xi(x)) \geq& \langle\nabla_y\phi_\xi(f_\xi(x)), f(x') - f(x)\rangle \\
=& \langle\nabla_y\phi_\xi(f_\xi(x)), \langle\nabla f_\xi(x), x' - x\rangle\rangle + \langle\nabla_y\phi_\xi(f_\xi(x)), e\rangle \\
=& \langle\nabla_x\phi_\xi(f_\xi(x)), x' - x\rangle + \langle\nabla_y\phi_\xi(f_\xi(x)), e\rangle \\
\geq& \langle\nabla_x\phi_\xi(f_\xi(x)), x' - x\rangle - \|\nabla_y\phi_\xi(f_\xi(x))\|\|e\| \\
\geq& \langle\nabla_x\phi_\xi(f_\xi(x)), x' - x\rangle - \frac{G_\phi L_f}{2}\|x' - x\|^2
\end{aligned}$$

Then we can apply Lemma D.2 to obtain the lower smoothness of $\phi_\xi(f_\xi(x))$. □

Proof of Theorem 7.2. For a given ξ, we let $y = f_\xi(x)$. For conciseness, we left out parameter when there is no ambiguity in following calculation.

$$\nabla\nabla F(x) = \mathbb{E}_\xi\left[\nabla f_\xi^T \nabla_y \nabla_y \phi_\xi \nabla f_\xi + \sum_{i=1}^{d'} \nabla_{y_i}\phi_\xi \nabla\nabla f_i\right]$$

Since ϕ_ξ is always convex, we know $\nabla f_\xi^T \nabla_y \nabla_y \phi_\xi \nabla f_\xi \succcurlyeq 0$. Therefore

$$-\nabla\nabla F(x) \preccurlyeq \mathbb{E}_\xi\left[\sum_{i=1}^{d'}(-\nabla_{y_i}\phi_\xi)\nabla\nabla f_{\xi,i}\right].$$

According to Cauchy–Schwarz inequality for symmetric matrices in Lemma D.4, we have

$$\begin{aligned}-\nabla\nabla F(x) \preccurlyeq& \mathbb{E}_\xi\left[\sum_{i=1}^{d'}(-\nabla_{y_i}\phi_\xi)\nabla\nabla f_{\xi,i}\right]\\ \preccurlyeq& \sqrt{\left(\mathbb{E}_\xi\sum_{i=1}^{d'}(\nabla\nabla f_{\xi,i})^2\right)\left(\mathbb{E}_\xi\sum_{i=1}^{d'}(\nabla_{y_i}\phi_\xi)^2\right)}.\end{aligned}$$

According to Lemma D.5, we have

$$\mathbb{E}_\xi\sum_{i=1}^{d'}(\nabla\nabla f_{\xi,i})^2 \le L_f^2 I.$$

According to convexity and L_ϕ-Lipschitz smoothness of ϕ_ξ, we have

$$\begin{aligned}\mathbb{E}_\xi\sum_{i=1}^{d'}(\nabla_{y_i}\phi_\xi)^2 =& \mathbb{E}_\xi\|\nabla_y\phi_\xi(y)\|^2\\ \le& \mathbb{E}_\xi[2L_\phi(\phi_\xi(y) - \min_y \phi_\xi(y))]\\ =& 2L_\phi F(x)\end{aligned}$$

Combining above inequalities gives us

$$-\nabla\nabla F(x) \preccurlyeq L_f\sqrt{\mathbb{E}_\xi\|\nabla_y\phi_\xi(y)\|^2}I \preccurlyeq L_f\sqrt{2L_\phi F(x)}I.$$

□

H. Additional experiments

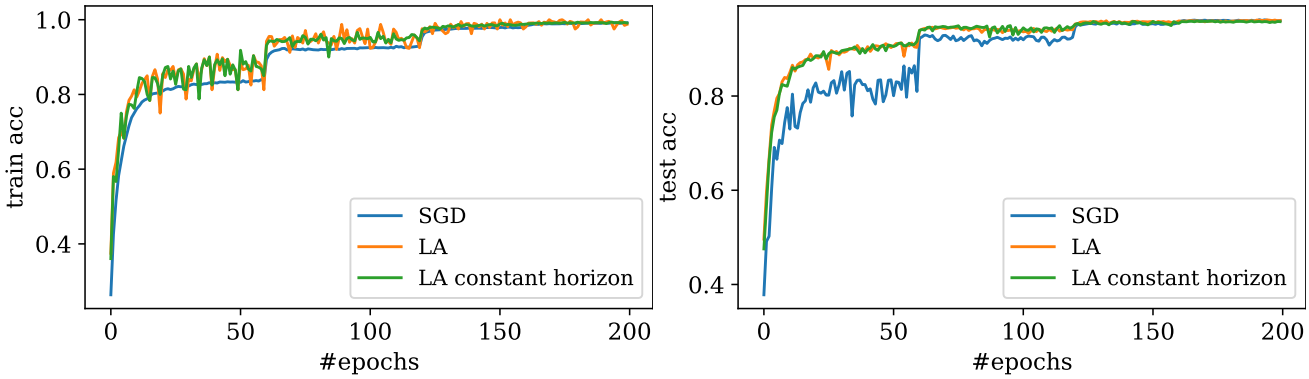


Figure H.1: Convergence of ResNet-18 on CIFAR-10.

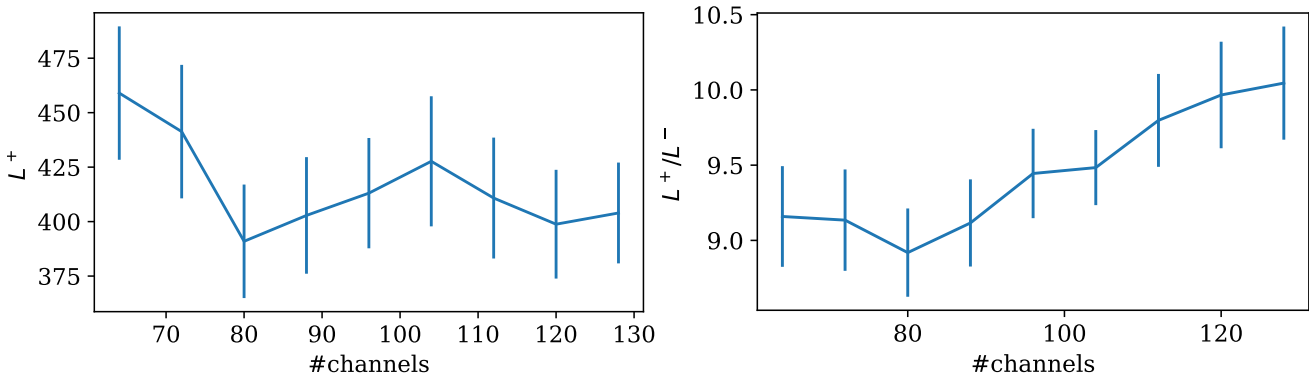


Figure H.2: Additional result for wide network.

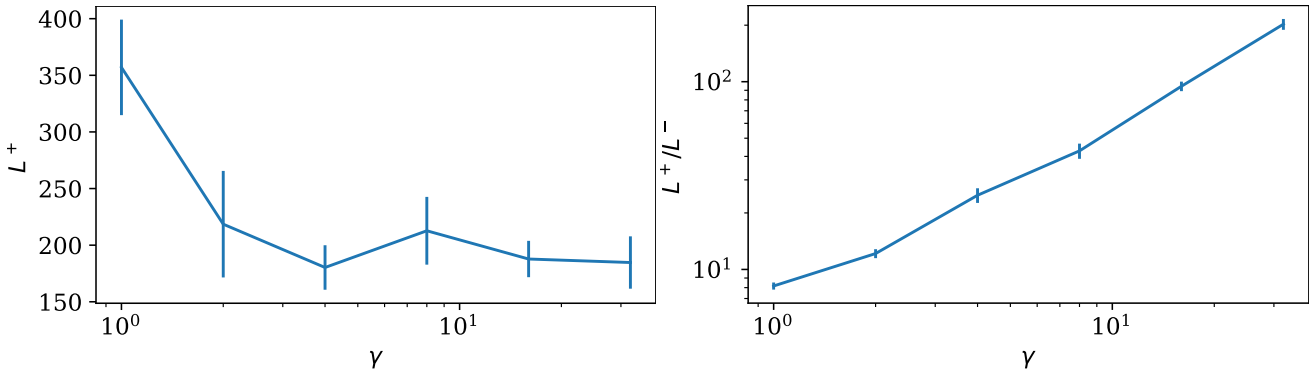


Figure H.3: Additional result for lazy network.

I. Experiment setup

We train ResNet-18 on CIFAR-10 (Krizhevsky et al., 2009) with Cutout regularization (DeVries & Taylor, 2017) for Figure H.1. We use SGD with initial step size $\gamma = 0.1$, Lookahead with $\tau = 5$ and same initial step size, and Lookahead with constant horizon with same initial τ and γ. At epochs $60, 120, 160$, step size γ for all algorithm decreases by 5 times. For Lookahead with constant horizon, the τ also increase 5 times at these epochs. Thus after 160 epochs, Lookahead with constant horizon uses $\tau = 625$. This experiment takes less than 2 hours on a NVIDIA Titan Xp graphic card.

Figure 3 is generated by stochastic Lanczos method (Yao et al., 2020) for a randomly initialized and a small batch randomly sampled from CIFAR-10 with 128 batch size. The network is initialized with Kaiming initialization. The linear loss is constructed by sampling a random weight vector from an isotropic Gaussian distribution. These two images takes less than 4 minutes to generate on a NVIDIA RTX A5000 graphic card.

We use a customized ResNet-18 in Figures 5a and H.2 for flexible channel numbers. We choose 9 different width in total, from 64 to 128 with a step as 8. For each width, we generate 50 random network. The variance of random neural network is high. In order to reduce the variance, we didn't sample independent networks for different widths. Instead, we sample 50 largest networks and then cast them into smaller networks. Besides cutting the large tensor into a smaller one, we tune the magnitude of weights to ensure the smaller network still follows Kaiming initialization. This experiment takes less than 10 hours on a NVIDIA RTX A5000 graphic card.

We use another customized ResNet-18 in Figures 5b and H.3 to implement the lazy network $f_{\xi,\gamma}(x) = \gamma f_\xi(\frac{x}{\gamma})$. During forward propagation, each parameter is multiplied by a factor $\frac{1}{\gamma}$, and output is multiplied by a factor γ. The magnitude of parameters are changed to ensure exactly same output. We choose 6 different γ in total, including $1, 2, 4, 8, 16, 32$, and sample 10 independent networks for each γ. This experiment takes around 1 hour on a NVIDIA RTX A5000 graphic card.

We train a ResNet-18 with no data-augmentation on CIFAR-10 in Figures 4 and 6. Since computing Hessian information

is time consuming, we only choose ξ from a small batch of training dataset with 128 batch size for computing $F(x) = \mathbb{E}_\xi[\phi_\xi(f_\xi(x))]$. Thus $F(x)$ is just training loss on this minibatch. At the end of each epoch, we apply power iteration to estimate the largest and smallest eigenvalue of $\nabla\nabla F(x)$ up to 0.1% relative accuracy. This experiment takes less than 12 hours on a NVIDIA Titan Xp graphic card.