On the Convergence of Federated Optimization in Heterogeneous Networks

Preprint · December 2018

CITATIONS
0

READS
307

6 authors, including:

Anit Sahu
35 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Maziar Sanjabi
University of Minnesota Twin Cities
25 PUBLICATIONS **575** CITATIONS
SEE PROFILE

Manzil Zaheer
Carnegie Mellon University
41 PUBLICATIONS **426** CITATIONS
SEE PROFILE

Virginia Smith
University of California, Berkeley
17 PUBLICATIONS **476** CITATIONS
SEE PROFILE

On the Convergence of Federated Optimization in Heterogeneous Networks

Anit Kumar Sahu*, Tian Li*, Maziar Sanjabi, Manzil Zaheer,
Ameet Talwalkar, Virginia Smith

Abstract

The burgeoning field of federated learning involves training machine learning models in massively distributed networks, and requires the development of novel distributed optimization techniques. Federated averaging (`FedAvg`) is the leading optimization method for training non-convex models in this setting, exhibiting impressive empirical performance. However, the behavior of `FedAvg` is not well understood, particularly when considering data heterogeneity across devices in terms of sample sizes and underlying data distributions. In this work, we ask the following two questions: (1) Can we gain a principled understanding of `FedAvg` in realistic federated settings? (2) Given our improved understanding, can we devise an improved federated optimization algorithm? To this end, we propose and introduce `FedProx`, which is similar in spirit to `FedAvg`, but more amenable to theoretical analysis. We characterize the convergence of `FedProx` under a novel *device similarity* assumption.

1 Introduction

Modern networks of remote devices, such as mobile phones, wearable devices, and autonomous vehicles, generate massive amounts of data each day. Federated learning involves training statistical models directly on these devices, and introduces novel statistical and systems challenges that require a fundamental departure from standard methods designed for distributed optimization in data center environments.

From a statistical perspective, each device collects data in a non-identical and heterogeneous fashion, and the number of data points on each device may also vary significantly. Federated optimization methods must therefore be designed in a robust fashion in order to provably converge when dealing with heterogeneous statistical data. From a systems perspective, the size of the network and high cost of communication impose two additional constraints on federated optimization methods: (i) limited network participation, and (ii) high communication costs. In terms of participation, at each communication round, proposed methods should only require a small number of devices to be active. As most devices have only short windows of availability, communicating with the entire network at once can be prohibitively expensive. In terms of communication, proposed methods should allow for

Preprint. Work in progress.
*Authors contributed equally.

local updates to be computed and applied on each device, as local updating has been demonstrated to reduce the total number of communication rounds and enable flexible communication patterns, e.g., compared to traditional gradient descent or stochastic gradient descent (SGD) (Smith et al., 2016, 2017; McMahan et al., 2016).

`FedAvg` has been recently introduced for the federated setting (McMahan et al., 2016). `FedAvg` works simply by running some number of epochs, E, of a local solver on a subset $K \ll N$ of the total devices N at each round, and then averaging the resulting model updates. While the effectiveness of `FedAvg` has been explored empirically (albeit on limited benchmarking settings), it lacks theoretical convergence guarantees, and recent works exploring convergence guarantees are limited to unrealistic scenarios, e.g., where (i) the data is either shared across devices or distributed in an IID (identically and independently distributed) manner, and (ii) all devices are involved in communication at each round (Stich, 2018; Wang & Joshi, 2018; Woodworth et al., 2018). While these assumptions simplify the analyses, they also violate key properties of realistic federated environments.

Contributions. In this work, we ask the following two questions: (1) Can we gain a principled understanding of `FedAvg` in realistic federated settings? (2) Can we devise an improved federated optimization algorithm, both theoretically and empirically? To this end we propose a novel federated optimization method. `FedProx` is very similar in spirit to `FedAvg`, but more amenable to theoretical analysis. We provide the first convergence guarantees for `FedProx` in practical heterogeneous data settings, and use our analysis to highlight the relative merits of these methods along with `FedAvg`.

2 Related Work

Large scale distributed machine learning, particularly in data center settings, has motivated the development of numerous distributed optimization methods in the past decade (Li et al., 2014; Dean et al., 2012; Jaggi et al., 2014; Smith et al., 2016; Ma et al., 2017; Reddi et al., 2016; Zhang et al., 2015; Shamir et al., 2014). However, as computing substrates such as phones, sensors, or wearable devices grow both in power and in popularity, it is increasingly attractive to learn statistical models directly over networks of distributed devices, as opposed to moving the data to the data center. This problem, known as federated learning, requires tackling novel challenges with privacy, heterogeneous data and nodes, and massively distributed computational networks.

Several recent methods have been proposed that are tailored to the specific challenges in the federated setting (Smith et al., 2017; McMahan et al., 2016). For example, Smith et al. (2017) proposes to learn the models in the dual domain where it is easier to decouple the global objective into distributed subproblems. Despite the theoretical guarantees and practical efficiency of the proposed method, such an approach is not generalizable to non-convex problems, e.g. deep learning, where strong duality is no longer guaranteed. In the non-convex setting, Federated Averaging (`FedAvg`), a heuristic method based on averaging local Stochastic Gradient Descent (SGD) updates in the primal, has instead been shown to work well empirically (McMahan et al., 2016).

Unfortunately, `FedAvg` is quite challenging to analyze due to its local updating scheme, the fact that few clients are active at each round, and the issue that data is frequently distributed in a heterogeneous nature in federated settings. Heterogeneity in the data gets reflected in the number of samples and also through a very few representative classes of all the classes being present in each client. Recent works have made steps towards analyzing `FedAvg` in simpler, non-federated settings. For instance, parallel SGD (Stich, 2018; Wang & Joshi, 2018), which makes local updates similar to `FedAvg`, has been studied in the IID setting. However, the main ingredient of the proof is the observation that each local SGD is a copy of the same stochastic process (due to the IID assumption), and this line of reasoning does not apply to the heterogeneous setting.

In this work, inspired by `FedAvg`, we take a different approach and propose a broader framework, `FedProx`. We can analyze the convergence behavior of the framework under a novel local similarity assumption between local functions. Our similarity assumption is inspired by the Kaczmarz method for solving linear system of equations (Kaczmarz, 1993). A similar assumption has been previously used to analyze variants of SGD for strongly convex problems (Schmidt & Roux, 2013). To the best of our knowledge, this is the first convergence analysis of any methods for federated learning with heterogeneous data. While the derived rates do not show improved theoretical behavior over traditional distributed optimization techniques, they are valuable tool for gaining insight into the strengths and weaknesses of various federated learning methods.

3 Federated Optimization: Algorithms

In this section, we introduce a few algorithms that conform to the requirements of being deployed in a federating learning framework which typically involves multiple clients[1] collecting data and a central server coordinating the learning objective. The common theme among these methods is that in all of them at each client a local objective function based on the local data is used as a surrogate for the global objective. The global model update involves selecting a few online clients of all the available clients and then using local solvers to optimize the local objective functions across the selected clients. The global model update is then obtained by incorporating the local model updates from the selected online clients. Technically speaking, we aim to minimize the following global objective function:

$$\min_w \ f(w) = \sum_{k=1}^{N} p_k F_k(w) = \mathbb{E}_k[F_k(w)], \tag{1}$$

where N is the number of nodes, $p_k \geq 0$ and $\sum_k p_k = 1$. In general, the local objectives $F_k(\cdot)$'s are given by local empirical risks, i.e., they are average risks over local n_k samples. Hence, we can set $p_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ is the total number of data points.

Before presenting different federated algorithms, we introduce the definition of an inexact solution to a subproblem which is utilized throughout the paper.

[1] We use nodes, clients and devices interchangeably throughout the paper.

Definition 3.0.1. *For a smooth convex function* $h(w; w_0) = F(w) + \frac{\mu}{2}\|w - w_0\|^2$, *and* $\gamma \in [0, 1]$, *we say* w^* *is a* γ*-inexact solution of* $\min_w h(w; w_0)$, *if* $\|\nabla h(w^*; w_0)\| \leq \gamma \|\nabla h(w_0; w_0)\|$, *where* $\nabla h(w; w_0) = \nabla F(w) + \mu(w - w_0)$. *Note that a smaller* γ *corresponds to higher accuracy.*

3.1 Federated Averaging (`FedAvg`) and Federated Proximal (`FedProx`) methods

In Federated Averaging (`FedAvg`) (McMahan et al., 2016), which is the state of the art in federated learning, the local surrogate of the global objective function at client k is taken to be $F_k(\cdot)$ and the local solver is chosen to be an SGD method which is homogeneous across clients in terms of the different optimization parameters, i.e., learning rate and the number of local epochs. The details of `FedAvg` are summarized in Algorithm 1. `FedAvg` is shown to have impressive empirical

Algorithm 1: Federated Averaging (`FedAvg`)

INPUT: K, T, η, E, w^0, N, p_k, $k = 1, \cdots, N$;
forall $t = 0, \cdots, T-1$ **do**
  Server chooses K users at random (each user k is chosen with probability p_k);
  Server sends w^t to all chosen users.;
  Each user k updates w^t for E epochs of SGD on F_k with step-size η to obtain w_k^t.;
  Each chosen user k sends w_k^{t+1} back to the central node.;
  Server aggregates the w's as
  $w^{t+1} = \frac{1}{K}\sum_k w_k^{t+1}$;

performance even when the data is heterogeneous (McMahan et al., 2016). The heterogeneity of the data is a crucial property of federated set up which stems from the disparity between the local data distributions. MacMahan et al. (McMahan et al., 2016) empirically show that it is crucial to tune the optimization parameters such as the learning rate and especially the number of local epochs to get `FedAvg` to work in heterogeneous settings. This is very intuitive as the more local epochs means the local models move further away from the initial global model in the direction to optimize local functions F_k—this could be problematic when F_k's are very different, i.e., when local distributions are heterogeneous. While a carefully chosen learning rate is imperative for any optimization problem, the choice of the number of local epochs controls the amount of local changes in the models. To be specific, a large number of local epochs is more likely to move the local models to far away from the current model at the server. Thus, by limiting the number of local SGD epochs, we limit the amount of local changes in the models.

Thus in a heterogeneous setting it would be beneficial to limit the amount of local updates through a more flexible tool beyond heuristically limiting the number of local updates. A natural way to enforce limited local model updates is to incorporate a constraint which penalizes big changes from the current model at the server. This observation paves the way for our proposed method, `FedProx`.

3.2 Federated Proximal (FedProx) Method

FedProx is similar to FedAvg in choosing the users to update in each iteration and the way it aggregates the local updates to form a global update. But in each node k instead of just minimizing the local function F_k, node k uses its local solver of choice to approximately minimize

$$\min_w h_k(w;\ w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2. \tag{2}$$

Technically speaking, the above algorithm can be re-stated as follows:

$$\min_w F_k(w) \ \text{ such that } \ \left\|w - w^t\right\| \leq \epsilon, \tag{3}$$

where the constraint $\|w - w^t\| \leq \epsilon$ is then translated into the objective function as a ℓ_2 regularizer for the optimization variable w. Such a constraint makes the handling of enforcing limited local model updates more explicit than FedAvg, where the enforcement is instead through carefully tuning the number of local epochs. Note that the proximal term in the above expression effectively limits the impact of local updates (by restricting them to be close to the initial point) without any need to manually tune the number of local epochs. Algorithm 2 summarizes the steps of FedProx. The usage of the proximal term in FedProx to limit the amount of local changes also makes it more amenable for theoretical analysis. As we will show later, FedProx behaves similarly to FedAvg empirically.

Algorithm 2: FedProx

INPUT: K, T, μ, γ, w^0, N, p_k, $k = 1, \cdots, N$;
forall $t = 0, \cdots, T-1$ **do**
  Server choose K users at random (each user k is chosen with probability p_k).;
  Server sends w^t to all chosen users ;
  Each chosen user k finds a w_k^{t+1} which is a γ-inexact minimizer of:
  $w_k^{t+1} \approx \arg\min_w\ h_k(w;\ w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|$;
  Each chosen user k sends w_k^{t+1} back to the Server.;
  Server aggregates the w's as $w^{t+1} = \frac{1}{K}\sum_k w_k^{t+1}$;

4 Federated Optimization: Convergence Analysis

4.1 Convergence Analysis of FedProx

FedAvg and FedProx are stochastic algorithms by nature; in that, each step of these algorithms, only a fraction of the users are sampled to perform the update. Moreover, the local optimizer could be stochastic, e.g. SGD. It is well known that for stochastic methods in order to converge to a stationary point, a decreasing step-size is required. This is in contrast to the non-stochastic methods, e.g. gradient descent, that can find a stationary point with constant step-size. In order to

analyze the convergence behavior of such methods with constant step-size, which is what is usually deployed in practice, we need to somehow control the dissimilarity among the local functions. This could be achieved through assuming IID, i.e. homogeneous, data among users. But we aim to generalize this idea to heterogeneous setting, which would better represent the federated learning. Thus, we propose a metric for dissimilarity among local functions and analyze `FedProx` under bounded dissimilarity assumption.We propose a metric to measure the dissimilarity between the local functions. And then analyze the convergence behavior of `FedProx` under a bounded dissimilarity assumption.

4.1.1 Local dissimilarity

Definition 4.0.1. *The local functions $F_k(\cdot)$ at w are said to be B-locally dissimilar at w if $\mathbb{E}_k[\|\nabla F_k(w)\|^2] \leq \|\nabla f(w)\|^2 B^2$. We further define $B(w) = \sqrt{\frac{\mathbb{E}_k[\|\nabla F_k(w)\|^2]}{\|\nabla f(w)\|^2}}$, when[2] $\|\nabla f(w)\| \neq 0$.*

Note that in this definition, a smaller value of $B(w)$ at any point w implies that the local functions are more locally similar. Also, $B(w) \geq 1$ by definition. In the extreme case when all the local functions are the same, then $B(w) = 1$, for all w. Let us also consider the case where $F_k(\cdot)$'s are associated with empirical risk objectives. If the samples on all the nodes are homogeneous, i.e. they are sampled in an IID fashion, then as $\min_k\ n_k \to \infty$, it follows that $B(w) \to 1$ for every w as all the local functions are converging to become the same expected risk function. However, in the federated setting the data distributions are heterogeneous. And even if the samples are IID on each device, in the finite sample case, $B > 1$ due to the sampling discrepancies. Thus, it is natural to think of the case where $B > 1$ and our definition of dissimilarity as a generalization of the IID assumption for the local distributions are heterogeneous but not very dissimilar. The hope is that although the data points are not IID, but still the dissimilarity B would not be that large throughout the training process.

Assumption 4.1. *For some $\epsilon > 0$, there exists a B_ϵ such that for all the points $w \in \mathcal{S}_\epsilon^c = \{w \mid \|\nabla f(w)\| > \epsilon\}$, $B(w) \leq B_\epsilon$.*

Remark 4.0.1. *In the case of bounded variance, i.e. $E_k[\|\nabla F_k(w) - \nabla f(w)\|^2] \leq \sigma^2$, which is essential for analyzing SGD, then for any $\epsilon > 0$, it follows that $B_\epsilon \leq \sqrt{1 + \frac{\sigma^2}{\epsilon^2}}$.*

In most practical deep learning settings, especially in the federated setup, there is no need to solve the problem to arbitrarily accurate stationary solutions to get good generalization, i.e. ϵ is typically not very small. In fact, empirical data shows that solving the problem beyond some threshold might even hurt the generalization performance due to overfitting. Although in practical federated learning problems the samples are not IID, but they are still sampled from distributions that are not drastically different. Thus, the dissimilarity between local functions would potentially stay bounded throughout most of the training process.

[2] As an exception we define $B(w) = 1$, when $\mathbb{E}_k[\|\nabla F_k(w)\|^2]$, i.e. w is a stationary solution that all the local functions F_k agree on.

4.1.2 `FedProx` Analysis

With the bounded dissimilarity assumption 4.1 in place, we can now analyze the amount of expected decrease in the objective when one step of `FedProx` is performed under the bounded dissimilarity assumption 4.1.

Lemma 4.1. *Assume the functions F_k are L-Lipschitz smooth and also there exists L_-, such that $\nabla^2 F_k \succeq -L_- \mathbf{I}$ and define $\bar{\mu} = \mu - L_- > 0$. Suppose that w^t is not a stationary solution and local functions F_k are B-dissimilar, i.e. $B(w^t) \leq B$, then if μ, K and γ in Algorithm 2 are chosen such that*

$$\rho = \left(\frac{1}{\mu} - \frac{\gamma B}{\mu} - \frac{B(1+\gamma)}{\bar{\mu}\sqrt{K}} - \frac{LB(1+\gamma)}{\bar{\mu}\mu} - \frac{L(1+\gamma)^2 B^2}{2\bar{\mu}^2} - \frac{LB^2(1+\gamma)^2}{\bar{\mu}^2 K}\left(2\sqrt{K}+1\right)\right) > 0, \quad (4)$$

then at iteration t of the `FedProx` Algorithm 2, we have the following expected decrease in the global objective $\mathbb{E}_{S_t} f(w^{t+1}) \leq f(w^t) - \rho \|\nabla f(w^t)\|^2$, where the expectation is over the set S_t of K users chosen at iteration t.

Proof. First of all note that based on our inexactness assumption, we can define e_k^{t+1} such that

$$\nabla F_k(w_k^{t+1}) + \mu(w_k^{t+1} - w^t) - e_k^{t+1} = 0, \quad \& \quad \|e_k^{t+1}\| \leq \gamma \|\nabla F_k(w^t)\| \quad (5)$$

Now let us define $\bar{w}^{t+1} = \mathbb{E}_k w_k^{t+1}$. Based on this definition, we know

$$\bar{w}^{t+1} - w^t = \frac{-1}{\mu} \mathbb{E}\, \nabla F_k(w_k^{t+1}) + \frac{1}{\mu} \mathbb{E} e_k^{t+1}. \quad (6)$$

Let us define $\bar{\mu} = \mu - L_- > 0$ and $\hat{w}_k^{t+1} = \arg\min_w \; h_k(w; w^t)$. Then, due to the $\bar{\mu}$-strong convexity of h_k, we have

$$\|\hat{w}_k^{t+1} - w_k^{t+1}\| \leq \frac{\gamma}{\bar{\mu}} \|\nabla F_k(w^t)\|. \quad (7)$$

Note that once again, due to the $\bar{\mu}$-strong convexity of h_k, we know that $\|\hat{w}_k^{t+1} - w^t\| \leq \frac{1}{\bar{\mu}} \|\nabla F_k(w^t)\|$. Now we can use the triangle inequality to get

$$\|w_k^{t+1} - w^t\| \leq \frac{1+\gamma}{\bar{\mu}} \|\nabla F_k(w^t)\|. \quad (8)$$

Therefore,

$$\|\bar{w}^{t+1} - w^t\| \leq \mathbb{E}_k \|w_k^{t+1} - w^t\| \leq \frac{1+\gamma}{\bar{\mu}} \mathbb{E}_k \|\nabla F_k(w^t)\| \leq \frac{1+\gamma}{\bar{\mu}} \sqrt{\mathbb{E}_k \left[\|\nabla F_k(w^t)\|^2\right]} \leq \frac{B(1+\gamma)}{\bar{\mu}} \|\nabla f(w^t)\|, \quad (9)$$

where the last inequality is due to the bounded dissimilarity assumption.

Now let us define E_{t+1} such that $\bar{w}^{t+1} - w^t = \frac{-1}{\mu}\big(\nabla f(w^t) + E_{t+1}\big)$, i.e.

$$E_{t+1} = \mathbb{E}_k[\nabla F_k(w_k^{t+1}) - \nabla F_k(w^t) - e_k^{t+1}] \quad (10)$$

Now let us also bound $\|E_{t+1}\|$:

$$\begin{aligned}\|E_{t+1}\| &\leq \mathbb{E}_k\left[L\|w_k^{t+1}-w_k^t\|+\|e_k^{t+1}\|\right]\\ &\leq \left(\frac{L(1+\gamma)}{\bar{\mu}}+\gamma\right)\mathbb{E}_k\|\nabla F_k(w^t)\| \leq \left(\frac{L(1+\gamma)}{\bar{\mu}}+\gamma\right)B\|\nabla f(w^t)\|, \end{aligned} \tag{11}$$

where the last inequality is also due to bounded dissimilarity assumption. Now based on the L-Lipchitz smoothness of f and Taylor expansion we have

$$\begin{aligned} f(\bar{w}^{t+1}) &\leq f(w^t)+\langle\nabla f(w^t),\bar{w}^{t+1}-w^t\rangle+\frac{L}{2}\|\bar{w}^{t+1}-w^t\|^2\\ &\leq f(w^t)-\frac{1}{\mu}\|\nabla f(w^t)\|^2-\frac{1}{\mu}\langle\nabla f(w^t),E_{t+1}\rangle+\frac{L(1+\gamma)^2B^2}{2\bar{\mu}^2}\|\nabla f(w^t)\|^2\\ &\leq f(w^t)-\left(\frac{1}{\mu}-\frac{LB(1+\gamma)}{\bar{\mu}\mu}-\frac{\gamma B}{\mu}-\frac{L(1+\gamma)^2B^2}{2\bar{\mu}^2}\right)\|\nabla f(w^t)\|^2. \end{aligned} \tag{12}$$

From the above inequality it follows that if we set the penalty parameter μ large enough, we can get a decrease in the objective value of $f(\bar{w}^{t+1})-f(w^t)$ which is proportional to $\|\nabla f(w^t)\|^2$. But this is not the way that the algorithm works. In the algorithm, we only use K users that are chosen randomly to approximate $\bar{w}^t$. So, in order to find the $\mathbb{E}f(w^{t+1})$, we use local Lipchitz continuity of the function f.

$$f(w^{t+1}) \leq f(\bar{w}^{t+1})+L_0\|w^{t+1}-\bar{w}^{t+1}\|, \tag{13}$$

where L_0 is the local Lipchitz continuity constant of function f and we have

$$L_0 \leq \|\nabla f(w^t)\|+L\max(\|\bar{w}^{t+1}-w^t\|,\|w^{t+1}-w^t\|) \leq \|\nabla f(w^t)\|+L(\|\bar{w}^{t+1}-w^t\|+\|w^{t+1}-w^t\|) \tag{14}$$

Therefore, if we take expectation with respect to the choice of users in round t we need to bound

$$\mathbb{E}_{S_t}f(w^{t+1}) \leq f(\bar{w}^{t+1})+Q_t, \tag{15}$$

where $Q_t=\mathbb{E}_{S_t}\left[L_0\|w^{t+1}-\bar{w}^{t+1}\|\right]$. Note that the expectation is taken over the random choice of users to update.

$$\begin{aligned} Q_t &\leq \mathbb{E}_{S_t}\left[\left(\|\nabla f(w^t)\|+L(\|\bar{w}^{t+1}-w^t\|+\|w^{t+1}-w^t\|)\right)\cdot\|w^{t+1}-\bar{w}^{t+1}\|\right]\\ &\leq \left(\|\nabla f(w^t)\|+L\|\bar{w}^{t+1}-w^t\|\right)\mathbb{E}_{S_t}\|w^{t+1}-\bar{w}^{t+1}\|+L\mathbb{E}_{S_t}\left[\|w^{t+1}-w^t\|\cdot\|w^{t+1}-\bar{w}^{t+1}\|\right]\\ &\leq \left(\|\nabla f(w^t)\|+2L\|\bar{w}^{t+1}-w^t\|\right)\mathbb{E}_{S_t}\|w^{t+1}-\bar{w}^{t+1}\|+L\mathbb{E}_{S_t}\left[\|w^{t+1}-\bar{w}^{t+1}\|^2\right] \end{aligned} \tag{16}$$

From (9) we know that $\|\bar{w}^{t+1}-w^t\| \leq \frac{B(1+\gamma)}{\bar{\mu}}\|\nabla f(w^t)\|$. Moreover,

$$\mathbb{E}_{S_t}\|w^{t+1}-\bar{w}^{t+1}\| \leq \sqrt{\mathbb{E}_{S_t}\left[\|w^{t+1}-\bar{w}^{t+1}\|^2\right]} \tag{17}$$

and

$$\begin{aligned}\mathbb{E}_{S_t}\left[\|w^{t+1}-\bar{w}^{t+1}\|^2\right] &\leq \frac{1}{K}\mathbb{E}_k[\|w_k^{t+1}-\bar{w}^{t+1}\|^2], \quad \text{(as we choose } K \text{ users randomly to get } w^t\text{)}\\ &\leq \frac{1}{K}\mathbb{E}_k[\|w_k^{t+1}-w^t\|^2], \quad (\text{as } \bar{w}^{t+1}=\mathbb{E}_k\ w_k^{t+1})\\ &\leq \frac{1}{K}\frac{(1+\gamma)^2}{\bar{\mu}^2}\mathbb{E}_k\|\nabla F_k(w^t)\|^2 \quad \text{(from (8))}\\ &\leq \frac{B^2}{K}\frac{(1+\gamma)^2}{\bar{\mu}^2}\|\nabla f(w^t)\|^2, \end{aligned} \tag{18}$$

where the last inequality is due to bounded dissimilarity assumption. If we replace these bounds in (16) we get

$$Q_t \leq \left(\frac{B(1+\gamma)}{\bar{\mu}\sqrt{K}}+\frac{LB^2(1+\gamma)^2}{\bar{\mu}^2K}\left(2\sqrt{K}+1\right)\right)\|\nabla f(w^t)\|^2 \tag{19}$$

Combining (12), (15), (13) and (19) and using the notation $\alpha=\frac{1}{\mu}$ we get

$$\mathbb{E}_{S_t}f(w^{t+1}) \leq f(w^t)-\left(\frac{1}{\mu}-\frac{\gamma B}{\mu}-\frac{B(1+\gamma)}{\bar{\mu}\sqrt{K}}-\frac{LB(1+\gamma)}{\bar{\mu}\mu}-\frac{L(1+\gamma)^2B^2}{2\bar{\mu}^2}-\frac{LB^2(1+\gamma)^2}{\bar{\mu}^2K}\left(2\sqrt{K}+1\right)\right)\|\nabla f(w^t)\|^2. \tag{20}$$

In the convex case, where $L_-=0$ and $\bar{\mu}=\mu$, if $\gamma=0$, i.e. all subproblems are solved accurately, can get a decrease proportional to $\|\nabla f(w^t)\|^2$ if $B<\sqrt{K}$. In such a case if we assume $1<<B\leq 0.5\sqrt{K}$, then we can write

$$\mathbb{E}_{S_t}f(w^{t+1}) \lessapprox f(w^t)-\frac{1}{2\mu}\|\nabla f(w^t)\|^2+\frac{3LB^2}{2\mu^2}\|\nabla f(w^t)\|^2 \tag{21}$$

In this case, if we choose $\mu\approx 6LB^2$ we get

$$\mathbb{E}_{S_t}f(w^{t+1}) \lessapprox f(w^t)-\frac{1}{24LB^2}\|\nabla f(w^t)\|^2 \tag{22}$$

Therefore, the number of iterations to at least generate one solution with squared norm of gradient less than ϵ is $O(\frac{LB^2\Delta}{\epsilon})$. □

Remark 4.1.1. *Let the assertions of Lemma 4.1 hold. In addition, let $F_k(\cdot)$'s be convex and $\gamma=0$, i.e. all the local problems are solved accurately, if $1\ll B\leq 0.5\sqrt{K}$, then we can choose $\mu\approx 6LB^2$ from which it follows that $\rho\approx\frac{1}{24LB^2}$.*

Remark 4.1.2. *In order for ρ in Lemma 4.1 to be positive, we need $\gamma B<1$. Moreover, we also need $\frac{B}{\sqrt{K}}<1$. Note that these conditions might be restrictive due to the worst case nature of our analysis. Nevertheless, they quantify the trade-off between dissimilarity bound and the method's parameter.*

We can use the above sufficient decrease to obtain a convergence to the set of approximate stationary solutions $\mathcal{S}_\epsilon=\{w\mid\mathbb{E}\|\nabla f(w)\|^2\leq\epsilon\}$ under the bounded dissimilarity assumption 4.1.

Corollary 4.1.1. *Given some $\epsilon > 0$, assume that for $B \geq B_\epsilon$, μ, γ and K the assumptions of Lemma 4.1 hold at each iteration of* **FedProx**. *Moreover, $f(w^0) - f^* = \Delta$. Then, after $T = O(\frac{\Delta}{\rho\epsilon})$ iterations of* **FedProx** *we have $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(w^t)\|^2 \leq \epsilon$.*

Remark 4.1.3. *Note that* **FedProx** *gets the same asymptotic convergence guarantee as SGD. In other words, under bounded variance assumption, for very small ϵ, if we replace B_ϵ with its upper-bound in Remark 4.0.1 and choose μ large enough, then the iteration complexity of* **FedProx** *when the subproblems are solved exactly and $F_k(\cdot)$'s are convex would be $O(\frac{L\Delta}{\epsilon} + \frac{L\Delta\sigma^2}{\epsilon^2})$ which is the same as SGD Ghadimi & Lan (2013).*

Small ϵ in Assumption 4.1 translates to larger B_ϵ. Corollary 4.1.1 suggests that, in order to solve the problem with increasingly higher accuracies using `FedProx` one needs to increase μ appropriately. Moreover, in corollary 4.1.1, if we plug in the upper bound for B_ϵ, under bounded variance assumption (see Corollary 4.0.1), we get the number of required steps to achieve accuracy ϵ as $O(\frac{L\Delta}{\epsilon} + \frac{L\Delta\sigma^2}{\epsilon^2})$. Our analysis captures the weaknesses of `FedProx` and similar methods when the local functions are dissimilar. As a future direction, it would be interesting to quantify lower bounds for the convergence of the methods such as `FedProx`/`FedAvg` in settings involving heterogeneous data and clients.

Remark 4.1.4. *The main difference between Elastic SGD (see (Zhang et al., 2015)) and* **FedProx** *is that in* **FedProx** *only a fraction of clients are used in each step (which is an important assumption for the federated setting). Thus, the obtained results for* **FedProx** *could be specialized to obtain a more general convergence guarantee for Elastic SGD method in non-convex setting, which is of independent interest.*

References

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pp. 1223–1231, 2012.

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

Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in neural information processing systems*, pp. 3068–3076, 2014.

Stefan Kaczmarz. Approximate solution of systems of linear equations. *International Journal of Control*, 57(6):1269–1271, 1993.

Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pp. 19–27, 2014.

Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

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

Sashank J Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

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

Virginia Smith, Simone Forte, Chenxin Ma, Martin Takac, Michael I Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *arXiv preprint arXiv:1611.02189*, 2016.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 4424–4434, 2017.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

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

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

Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. In *Advances in Neural Information Processing Systems*, pp. 685–693, 2015.