Federated Learning with Proximal Stochastic Variance Reduced Gradient Algorithms

Conference Paper · August 2020

DOI: 10.1145/3404397.3404457

6 authors, including:

Canh T.Dinh
The University of Sydney
17 PUBLICATIONS **410** CITATIONS
SEE PROFILE

Nguyen H. Tran
The University of Sydney
271 PUBLICATIONS **8,965** CITATIONS
SEE PROFILE

Tuan Dung Nguyen
Australian National University
6 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Wei Bao
Nanyang Technological University
93 PUBLICATIONS **6,866** CITATIONS
SEE PROFILE

Federated Learning with Proximal Stochastic Variance Reduced Gradient Algorithms

Canh T. Dinh
The University of Sydney
NSW, Australia
tdin6081@uni.sydney.edu.au

Nguyen H. Tran
The University of Sydney
NSW, Australia
nguyen.tran@sydney.edu.au

Tuan Dung Nguyen*
The University of Melbourne
VIC, Australia
tuandungn@unimelb.edu.au

Wei Bao
The University of Sydney
NSW, Australia
wei.bao@sydney.edu.au

Albert Y. Zomaya
The University of Sydney
NSW, Australia
albert.zomaya@sydney.edu.au

Bing B. Zhou
The University of Sydney
NSW, Australia
bing.zhou@sydney.edu.au

ABSTRACT

Federated Learning (FL) is a fast-developing distributed machine learning technique involving the participation of a massive number of user devices. While FL has benefits of data privacy and the abundance of user-generated data, its challenges of heterogeneity across users' data and devices complicate algorithm design and convergence analysis. To tackle these challenges, we propose an algorithm that exploits proximal stochastic variance reduced gradient methods for non-convex FL. The proposed algorithm consists of two nested loops, which allow user devices to update their local models approximately up to an accuracy threshold (inner loop) before sending these local models to the server for global model update (outer loop). We characterize the convergence conditions for both local and global model updates and extract various insights from these conditions via the algorithm's parameter control. We also propose how to optimize these parameters such that the training time of FL is minimized. Experimental results not only validate the theoretical convergence but also show that the proposed algorithm outperforms existing Stochastic Gradient Descent-based methods in terms of convergence speed in FL setting.

KEYWORDS

Distributed Machine Learning, Federated Learning, Stochastic Gradient Descent

ACM Reference Format:
Canh T. Dinh, Nguyen H. Tran, Tuan Dung Nguyen, Wei Bao, Albert Y. Zomaya, and Bing B. Zhou. 2020. Federated Learning with Proximal Stochastic Variance Reduced Gradient Algorithms. In *49th International Conference on Parallel Processing - ICPP (ICPP '20), August 17–20, 2020, Edmonton, AB, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404397.3404457

*Work done at The University of Sydney, Australia.

Nguyen H. Tran, Wei Bao, and Bing B. Zhou were supported by the Australian Research Council Discovery Project grant DP200103718. Albert Y. Zomaya and Wei Bao were supported by the Australian Research Council Discovery Project grant DP190103710.

ICPP '20, August 17–20, 2020, Edmonton, AB, Canada

ACM ISBN 978-1-4503-8816-0/20/08…$15.00
https://doi.org/10.1145/3404397.3404457

1 INTRODUCTION

Large-scale distributed machine learning, mainly in datacenter settings, has drastically attracted research interests on distributed optimization algorithms to promptly and efficiently train deep-learning models on large data sizes [5, 6, 10, 19, 26, 28]. In many applications, large-scale datasets are distributed over multiple machines for parallel processing in order to speed up computation. Most of the learning algorithms in these studies are designed for machines with balanced and independent and identically distributed (i.i.d.) data.

Nevertheless, maintaining the privacy of clients' data has demanded a fundamental shift from computing in data centers to doing so collectively on many devices including mobile phones and sensors, in which data are stored and processed locally. A modern smart device can be considered a decent computer with powerful processors (e.g., Hexagon DSP with Qualcomm Hexagon Vector eXtensions on Snapdragon 835 [1]), and a multitude of sensors (e.g., cameras, microphones, and GPS) for collecting a significant amount of data, which make local training feasible. Most of the users' data are privacy-sensitive and large in nature so it is risky and intensive to log data to data centers for model training, which calls for this learning-at-the-edge development. An example of a privacy-preserving learning technique is the recently proposed Federated Learning (FL) paradigm [20]. This learning technique allows the user devices to collaboratively build a global training model by only sending the local models instead of huge volumes of raw data to the central server, thus not only preserving on-device data privacy but also saving communication bandwidth. Since local data sources are naturally distributed, FL also comes with new challenges of designing learning algorithms considering statistical heterogeneity in which the devices' data are possibly non-identically distributed.

De facto optimization algorithms in machine learning such as Gradient Descent (GD), Stochastic Gradient Descent (SGD) and variance reduction SGD (e.g., SVRG [11]) have been widely employed to enable devices' local updates in FL [12, 20, 31]. One of the pioneer works on FL [20] proposed FedAvg using averaged SGD local updates, which was empirically shown to perform well in non-convex federated settings. The authors in [12] proposed FSVRG to empirically improve the performance of FedAvg using

SVRG. However, these two works lack theoretical convergence analysis. The authors in [30] provided convergence analysis of their proposed scheme similar to FedAvg using SGD, but with relaxed i.i.d assumption for all devices' data. While the proposed algorithm in [31] used GD with provable convergence, their analysis is limited to convex loss functions which might not apply to the recent strikingly successful deep learning applications that are usually non-convex [14]. Furthermore, when the number of training samples is large, GD (which works on all data samples) is much more time-consuming and computationally intensive compared to SGD or variance reduction SGD (both of which work on randomly sampled data subsets). Considering typical smart devices with limited battery capacity and computational power, faster training with SGD and its variants would be more viable options than GD for the sake of FL. On the other hand, a framework named FedProx was proposed in [16] with convergence guarantee by solving the user local problem approximately.

While variance reduction SGD methods such as SVRG [11] and SARAH [22] have been shown to have faster convergence than SGD, it comes at a cost of full batch gradient computation after a period of iterations. Remarkably, the periodic full batch evaluation policy exactly fits into the inherent global aggregation nature (after multiple local updates at each device) of FL. Thus, observing that the literature lacks a complete *non-convex FL design using variance reduction SGD with provable convergence*, we bridge the gap by the following contributions:

- We propose a proximal-stochastic-variance-reduced-gradient-based algorithm, named FedProxVR, which consists of two nested loops of global and local iterations. While the global model update is performed at every global iteration by the server, the local training models are updated using proximal SVRG or SARAH by the devices up to an accuracy threshold, which affects the number of local iterations.
- We first provide a sufficient condition for the convergence of the local models up to an accuracy threshold. This condition reflects the *relations between the learning step size, number of local iterations, and accuracy threshold*. Specifically, we qualitatively show that by *controlling a parametrized learning step size*, the convergence of local model update is guaranteed when the number of local iterations is bounded in a range determined by the step-size parameter and the local accuracy threshold.
- We next provide a sufficient condition for the global model update, which also captures the communication complexity between the devices and server. We show that the global convergence is determined by two vital control knobs: the local accuracy threshold and a proximal penalty parameter. These two parameters also characterize *the trade-off between local and global model convergence*, which provides hindsights for algorithm design.
- We then propose the method of *minimizing the total training time* (local and global model update delays at the user devices and server, respectively) based on the corresponding control variables. We also show that numerical solutions to this minimization problem match the theoretical findings of FedProxVR's convergence analysis.
- We finally validate theoretical findings by presenting empirical convergence using various real and synthetic datasets with Tensorflow. Experimental results show that FedProxVR can boost convergence speed compared to SGD-based approaches to FL.

Next, we will present the related works and system model, in Sections 2 and 3, respectively. The design and analysis of FedProxVR will be shown in Section 4. Finally we will report the experimental results in Section 5. All technical proofs will be provided in the Appendix.

2 RELATED WORKS

国内外现状

Recently, there have been increasing interests in a new machine learning technique that exploits the participation of a number of user devices, called Federated Learning [13, 16, 20, 27, 31], in which each device generates its own data and thus the data across devices has statistical heterogeneity. Especially, several works proposed algorithms for FL in the context of edge and/or wireless networks [2, 29, 31].

There have been several attempts to design algorithms for non-convex problems to tackle challenges of FL such as heterogeneity across users' devices and data. The first approach focused on running a de facto algorithm (SGD/SVRG) for a fixed number of local iterations on each device [12, 20]. This approach is practical as it uses existing core optimization packages in machine learning (e.g., SGD). However, most of these works lack the convergence analysis, mainly due to the challenges of heterogeneous nature of FL. By using additional assumptions such as convexity and L-Lipschitz function, the work in [31] provided a convergence analysis, but the core algorithm was based on GD, whose computational complexity scales linearly with respect to the number of data samples. The second approach allows the devices to solve their primal problems approximately up to a local accuracy threshold [16, 24]. Solving local model update approximately can give a flexible computation-communication trade-off on whether devices should run more rounds on their local model update or communicate more to the server for global model update. In this direction, even though the authors of [27] showed the convergence guarantee of their algorithm, the primal-dual optimization technique they adopted is only applicable to the convex task learning. We observe that there is a connection between local accuracy threshold of the approximation method and the number of local iterations, which has not been addressed before. We will show this connection by using a fixed parametrized step size, which is critical to analyzing the communication complexity and algorithm design for FL.

In constructing more sophisticated and accurate gradient estimators than SGD in non-FL settings, various methods such as SVRG [11] and SARAH [22] have been proposed to reduce the variance of the gradient estimator in SGD. Based on these well-known variance reduction techniques, proximal methods such as ProxSVRG [9, 17] and ProxSARAH [23] have been proposed to handle the non-convex, non-smooth problems.

3 SYSTEM MODEL

We consider a system consisting of one aggregation server and a set $\mathcal{N}$ of N devices. Each device n stores a local data set $\mathcal{D}_n$, with

系统模型

its size is denoted by D_n. Then, we can define the total data size by $D = \sum_{n=1}^{N} D_n$. In an example of the supervised learning setting, at device n, $\mathcal{D}_n$ defines the collection of data samples given as a set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_n}$, where $x_i \in \mathbb{R}^d$ is an input sample vector with d features, and $y_i \in \mathbb{R}$ is the labeled output value for the sample x_i. The data can be generated through the usage of device, for example, via interactions with mobile apps.

In a typical learning problem, for a sample data $\{x_i, y_i\}$ with input x_i (e.g., the response time of various apps inside the device), the task is to find the *model parameter* $w \in \mathbb{R}^l, l \geq d$, that characterizes the output y_i with the loss function $f_i(w)$. Some examples of the loss function are $f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$ for linear regression and $f_i(w) = \{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$ for support vector machine. The loss function of device n is defined as

$$F_n(w) := \frac{1}{D_n} \sum\nolimits_{i \in \mathcal{D}_n} f_i(w). \tag{1}$$

Then, the learning model is the minimizer of the following global loss function minimization problem

$$\min_{w \in \mathbb{R}^d} \bar{F}(w) := \sum\nolimits_{n=1}^{N} \frac{D_n}{D} F_n(w). \tag{2}$$

Assumption 1. *$f_i(\cdot)$ is L-smooth, and $F_n(\cdot)$ is $(-\lambda)$-strongly convex and σ_n-divergence, $\forall n$, (with $\lambda, \sigma_n > 0$), respectively, as follows*

$$\left\|\nabla f_i(w) - \nabla f_i(w')\right\| \leq L\left\|w - w'\right\| \tag{3}$$

$$F_n(w) + \left\langle \nabla F_n(w), w' - w \right\rangle \leq F_n(w') + \frac{\lambda}{2}\left\|w - w'\right\|^2 \tag{4}$$

$$\left\|\nabla F_n(w) - \nabla \bar{F}(w)\right\| \leq \sigma_n \left\|\nabla \bar{F}(w)\right\|. \tag{5}$$

Here $\langle w, w' \rangle$ is the inner product of vectors w and w'. All of the norms are Euclidean norm. While the L-smooth assumption is common in the literature [16, 31], the strong convexity parameter $-\lambda$ not only allows F_n to be non-convex (especially for deep learning applications), but also bounds the non-convexity extent that is applicable to proximal methods [4, 16]. This distinguishes our analysis from conventional approaches with convexity assumption. The σ_n-divergence assumption (5) characterizes the heterogeneity of data across devices, in which larger σ_n signifies the dissimilarity between device n to others. Different from the gradient-divergence assumption in [31] (which basically assumed that $\left\|\nabla F_n(x) - \nabla \bar{F}(x)\right\| \leq \sigma_n$), the σ_n-divergence is allowed to grow, say, quadratically in any direction if $F_n()$ is convex quadratic function [3]. This assumption (5) also includes the B-local dissimilarity in [16] as a special case. While we only specify the devices' data heterogeneity via σ_n in (5), to simplify the notation we use the same L and λ for all n. But we note that all of the results in this work are unchanged even when we allow heterogeneous value of L_n and λ_n, where L, λ can be substituted by L_n, λ_n in Lemma 1 and by $\bar{L}, \bar{\lambda}$ in Theorem 1. We also use the notation $\bar{\sigma}^2 := \sum_{n=1}^{N} \frac{D_n}{D} \sigma_n^2$, and similarly to other cases.

4 ALGORITHM DESIGN AND ANALYSIS

In this section, we present the algorithm, provide its convergence analysis, and optimize the algorithm parameters.

4.1 Algorithm Design

In this section, we propose a FL framework using proximal stochastic variance reduced gradient algorithms, named FedProxVR, as shown in Alg. 1. To solve problem (2), FedProxVR requires T number of global iterations to update the global model $\bar{w}^{(s)}$, by aggregating all local models $w_n^{(s)}$ of all devices.

Algorithm 1 FedProxVR

1: **input:** $\bar{w}_0$, $\eta = \frac{1}{\beta L}$.
2: **for** $s = 1, \ldots, T$ **do** *Global iterations*
3: **for** $n = 1, \ldots, N$ **do in parallel**
4: $w_{n,s}^{(0)} = \bar{w}^{(s-1)}$
$v_{n,s}^{(0)} = \nabla F_n(w_{n,s}^{(0)})$
$w_{n,s}^{(1)} = \text{prox}_{\eta h_s}(w_{n,s}^{(0)} - \eta v_{n,s}^{(0)})$,
5: **for** $t = 1, \ldots, \tau$ **do** *Local iterations by devices*
6: Uniformly randomly pick $(x_{i_t}, y_{i_t}) \in \mathcal{D}_n$.
7: Update $v_{n,s}^{(t)}$ according to (8a) or (8b)
8: $w_{n,s}^{(t+1)} = \text{prox}_{\eta h_s}(w_{n,s}^{(t)} - \eta v_{n,s}^{(t)})$,
9: **end for**
10: Set $w_n^{(s)} = w_{n,s}^{(t')}$ where t' is chosen uniformly at random from $\{0, \ldots, \tau\}$
11: **end for**
12: $\bar{w}^{(s)} = \sum_{n=1}^{N} \frac{D_n}{D} w_n^{(s)}$ *Global model update by server*
13: **end for**
14: **output:** $\bar{w}^{(T)}$

Local model update: In order to obtain the local model $w_n^{(s)}$, each device n will solve its following surrogate function (lines 3 to 10)

$$\min_{w \in \mathbb{R}^d} \left\{ J_n(w) := F_n(w) + h_s(w) \right\}, \tag{6}$$

$$\text{where } h_s(w) := \frac{\mu}{2}\left\|w - \bar{w}^{(s-1)}\right\|^2, \tag{7}$$

One can view the regularized h_s as a "soft" consensus constraint to penalize any local deviation from the current global model $\bar{w}^{(s-1)}$. This function was also used by elastic average SGD and FedProx in [34] and [16], respectively. Since $h_s(.)$ is μ-strongly convex, we can allow $J_n(w)$ to be a $\tilde{\mu}$-strongly convex by choosing μ satisfying $\tilde{\mu} := \mu - \lambda > 0$. FedProxVR solves (6) using the proximal update rule (line 8) with the variance reduction stochastic gradient estimator using either (8a) or (8b)

$$v_{n,s}^{(t)} = \begin{cases} \nabla f_{i_t}(w_{n,s}^{(t)}) - \nabla f_{i_t}(w_{n,s}^{(t-1)}) + v_{n,s}^{(t-1)}, \text{ (SARAH)} & \text{(8a)} \\ \nabla f_{i_t}(w_{n,s}^{(t)}) - \nabla f_{i_t}(w_{n,s}^{(0)}) + v_{n,s}^{(0)}. \text{ (SVRG)} & \text{(8b)} \end{cases}$$

If $v_{n,s}^{(t)} = \nabla f_{i_t}(w_{n,s}^{(t)})$, we have the vanilla SGD. Both SARAH [22] and SVRG [11] use the outer loop for one full gradient evaluation and "anchor" model update (lines 2 to 4). However, in inner loops (lines 5 to 9), SARAH is different from SVRG by recursively updating the stochastic direction $v_{n,s}^{(t)}$ based on stochastic components from previous local iteration $(t-1)$ and current SGD. The proximal operator is defined as follows

$$\text{prox}_{\eta h_s}(x) := \arg\min_{w \in \mathbb{R}^d} \left(h_s(w) + \frac{1}{2\eta}\left\|w - x\right\|^2 \right) \tag{9}$$

$$= \frac{\eta}{1 + \eta\mu}\left(\mu\, \bar{w}^{(s-1)} + \frac{1}{\eta} x \right). \tag{10}$$

The convergence criterion of the local problem (6) (at a given s) is defined as follows

$$\mathbb{E}\left[||\nabla J_n(w_n^{(s)})|| \mid \bar{w}^{(s-1)}\right] \leq \theta \left\|\nabla F_n(\bar{w}^{(s-1)})\right\|, \tag{11}$$

which is parametrized by a local accuracy $\theta \in (0, 1)$, and thus by the total number of local iterations τ. This local accuracy concept resembles the approximation and inexact factors in [28] and [16, 24] respectively. Here $\theta = 0$ means the local problem (6) is required to be solved optimally, and $\theta = 1$ means no progress for local problem, i.e., by setting $\tau = 0$. Since all devices have the same τ, local model updates are synchronous. $\mathbb{E}[\cdot]$ is the expectation with respect to all randomness in FedProxVR.

Global model update: After receiving all local models sent by devices, the server will update the global model according to line 12, which will be fed back to all devices for next global iteration update. We also use the expected squared norm of the gradient as convergence indicator (i.e., stationary gap) for non-convex problems [3], the global problem (2) achieves an ϵ-accurate solution if

$$\frac{1}{T}\sum_{s=1}^{T}\mathbb{E}\left\|\nabla \bar{F}(\bar{w}^{(s)})\right\|^2 \leq \epsilon. \tag{12}$$

4.2 FedProxVR's Convergence Analysis

In FedProxVR, we choose a fixed step size η[1], parametrized by β such that $\eta = \frac{1}{\beta L}$. The convergence of local model update is provided as follows.

Lemma 1. *Device n achieves θ-accurate solution* (11) *if β and τ satisfy the following conditions*

a) when SARAH update (8a) *is used:*

$$0 \leq \frac{3(\beta^2L^2+\mu^2)}{\theta^2\tilde{\mu}L(\beta-3)} \leq \tau \leq \frac{5\beta^2-4\beta}{8} \tag{13}$$

b) when SVRG update (8b) *is used:*

$$0 \leq \frac{3(\beta^2L^2+\mu^2)}{\theta^2\tilde{\mu}L(\beta-3)} \leq \tau \leq \frac{5\beta^2-4\beta}{8\,a} - 2 \tag{14}$$

where there exists $a > 0$ such that $a - 4 \geq 4\sqrt{a(\tau+1)}$.

The following remarks are about relations between the *local accuracy θ, number of local iterations τ, and step size parameter β.*

Remark 1.

(1) For an arbitrary $\theta \in (0, 1]$, we can always choose a sufficiently large β to satisfy both (13) and (14), where the lower and upper bounds of τ are $\Omega(\beta)$ and $O(\beta^2)$, respectively.It means that with a sufficiently small (fixed) step size η, local convergence is guaranteed.
(2) We see that $\tau = \Omega(\frac{1}{\theta^2})$. Thus if θ is smaller, τ must be larger to satisfy lower bound conditions. It is straightforward that with a smaller value of θ, we have the solution to (6) is closer-to-optimal, which requires running more local iterations.
(3) In practice, since large step size η and thus fast convergence (small τ) are preferred, we choose the smallest β_{min} satisfying Lemma 1 conditions by solving (e.g., in case of SARAH)

$$\frac{3(\beta^2L^2+\mu^2)}{\theta^2\tilde{\mu}L(\beta-3)} = \frac{5\beta^2-4\beta}{8}, \beta > 3, \tag{15}$$

and correspondingly obtaining (the smallest) τ:

$$\tau = \frac{5\beta_{min}^2 - 4\beta_{min}}{8}. \tag{16}$$

(4) Observing that the lower bound of τ is $\Omega(\mu)$, thus increasing μ (e.g., to make $\tilde{\mu} \geq 0$ when λ is large) will increase τ. This is because larger μ will enforce the local update more proximal to the "anchor" point $\bar{w}^{(s-1)}$ in each s, thus making the convergence to the θ-accurate solution more slowly.
(5) Compared to SARAH, SVRG has stricter condition for upper bound (due to $a \geq 4$). Thus, SVRG requires a larger β_{min} to satisfy condition (14), and thus larger τ (due to the lower bound). This can be explained that SARAH uses the stochastic gradient estimates that are more stable than that of SVRG, which was also validated by a sample dataset in [22]. We note that a concrete theoretical comparison between SARAH and SVRG has not been explored before.

Defining the cost gap of an arbitrary point $\bar{w}^{(0)}$ by $\Delta(\bar{w}^{(0)}) := \mathbb{E}\left[\bar{F}(\bar{w}^{(0)}) - \bar{F}(\bar{w}^*)\right]$, we next provide the convergence condition for the global model update of FedProxVR.

Theorem 1. *Consider FedProxVR with all devices satisfying conditions in Lemma 1, we have*

$$\frac{1}{T}\sum_{s=1}^{T}\mathbb{E}\left\|\nabla \bar{F}(\bar{w}^{(s)})\right\|^2 \leq \frac{\Delta(\bar{w}^{(0)})}{\Theta\, T}. \tag{17}$$

where

$$\Theta = \frac{1}{\mu}\Bigg(1 - \theta\sqrt{2(1+\bar{\sigma}^2)} - \frac{2L}{\tilde{\mu}}\sqrt{(1+\theta^2)(1+\bar{\sigma}^2)} - \frac{2L\mu}{\tilde{\mu}^2}(1+\theta^2)(1+\bar{\sigma}^2)\Bigg) > 0.$$

Corollary 1. *The number of global iterations required to achieve ϵ-accurate solution to* (2) *is*

$$T \geq \frac{\Delta(\bar{w}^{(0)})}{\Theta\,\epsilon}, \tag{18}$$

Remark 2.

(1) We see that θ and μ are vital control "knobs" for the convergence of FedProxVR. Specifically, to enable $\Theta > 0$, we have to choose sufficiently large μ and $\theta < (2(1+\bar{\sigma}^2))^{-1/2}$, which shows how data heterogeneity impacts both local and global convergence. Specifically, larger $\bar{\sigma}^2$, thus smaller θ, means devices will run more local iterations.
(2) These two parameters also characterize the *trade-off between local and global convergence*. While global convergence requires θ to be sufficiently small, devices prefer larger θ for faster local convergence (in Remark 1). On the other hand, while μ must be sufficiently large to ensure global convergence, it should not be too large to have a negative impact on local convergence (i.e., making τ large) and global convergence (i.e., making Θ small, thus T large).
(3) Large L and λ will require large μ in order to have $\Theta > 0$.
(4) Compared to $O(\frac{1}{\epsilon})$-iteration of the conventional proximal SVRG [17] or SARAH [32] (with non-convex and fixed step size but without FL setting), we see that FedProxVR with $T = O(\frac{1}{\Theta\,\epsilon})$ is scaled by a *federated factor* Θ. Next, we will

[1]Using a fixed step size is more practical than diminishing step size [3].

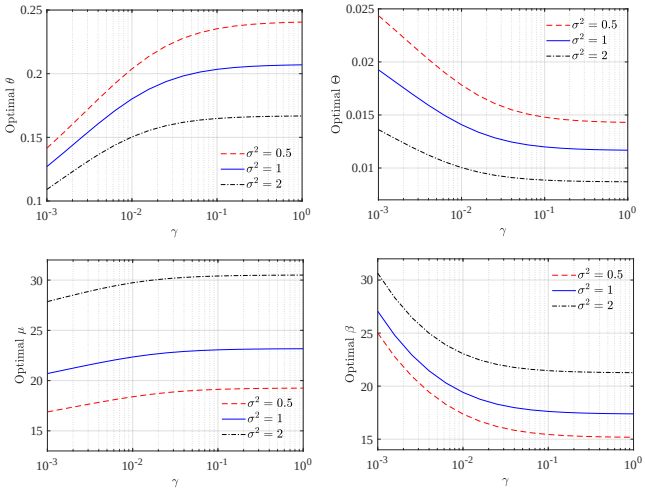

Figure 1: The effect of the weight factor γ to the solution to the problem (23), with $L = 1$, $\lambda = 0.5$ (these two values can be estimated by sampling real-world dataset.)

optimize the FedProxVR's parameters including the federated factor.

4.3 Optimizing FedProxVR's Parameters

Denoting the device's computation (i.e., steps 7 and 8 in Algorithm 1) and communication delays to send local model updates to the server by d_{cmp} and d_{com}, respectively, the total training time of FedProxVR is as follows

$$\mathcal{T} := T(d_{com} + d_{cmp}\tau). \tag{19}$$

Defining a weight factor $\gamma := \frac{d_{cmp}}{d_{com}}$ and $T = \frac{\Delta(\tilde{w}^{(0)})}{\Theta\,\epsilon}$, we minimize $\mathcal{T}$ with convergence conditions as constraints:

$$\underset{\mu,\theta,\beta,\tau}{\text{minimize}} \quad \frac{1}{\Theta}\Big(1 + \gamma\,\tau\Big) \tag{20}$$

$$\text{subject to} \quad (15), (16), \text{ and } \Theta > 0. \tag{21}$$

By removing constraint (15), (16) and substituting (with SARAH)

$$\theta^2 = \frac{24(\beta^2 L^2 + \mu^2)}{\tilde{\mu} L (5\beta^2 - 4\beta)(\beta - 3)} \tag{22}$$

into Θ, we further simplify this optimization problem as

$$\underset{\mu,\beta}{\text{minimize}} \quad \frac{1}{\Theta}\Big(1 + \gamma\,\frac{5\beta^2 - 4\beta}{8}\Big) \tag{23}$$

$$\text{subject to} \quad \beta > 3 \text{ and } \Theta > 0, \tag{24}$$

which has less variables and constraints than the original form (20). Problem (23) is unfortunate non-convex. However, since there are only two variables to optimize, we can employ numerical methods to find the global optimal solution. We numerically illustrate how the weight factor γ affects to the optimal parameters in Fig. 1. When *γ is very small, which means communication delay is much more expensive than local computation delay*, we see that optimal β (and thus τ) is very large, i.e., devices are better to have more local computation than communication rounds. When γ increases, while the optimal β decreases so that the local model update can be solved approximately with less τ, the optimal μ increases to ensure $\Omega > 0$ due to the corresponding increasing value of θ. We also observe that large $\bar{\sigma}^2$ increases the optimal μ and β, but decreases θ and Θ. All of the numerical observations in Fig. 1 exactly match the theoretical remarks of Lemma 1 and Theorem 1.

5 EXPERIMENTS

In this section, we will examine the efficacy of FedProxVR compared to the SGD-based FedAvg [20] by real-world experiments. We also show how FedProxVR's empirical convergence relates to its theoretical result by varying its control hyperparameters. All codes and data are ready to be published on GitHub [7].

Experimental settings: To evaluate the performance of FedProxVR on various tasks and learning models, we will use different types of datasets in our experiments. Besides a "Synthetic" dataset that captures the statistical heterogeneity as in [16, 26], we also consider real datasets such as "MNIST" [15] and "FASHION-MNIST" [33] for image classification tasks using both convex and non-convex models. All datasets are split randomly with 75% for training and 25% for testing.

In order to generate datasets for devices that mimic the heterogeneous nature of FL, we simulate 100 devices for convex models (Multinomial logistic regression) and 10 devices for a non-convex Convolutional Neural Network (CNN) model (since it would take drastically longer to run a CNN with 100 devices); each of the devices has a different sample size, generated according to the power law as in [16]. Furthermore, each device contains only two different labels over 10 labels. The number of data samples for each device is in the ranges of [37, 3277], [454, 3939], and [37, 1350] with respect to "Synthetic", "MNIST", and "FASHION-MNIST". We implement FedProxVR and SGD-based FedAvg using the Tensorflow framework.

To allow a fair comparison, all algorithms use the same parameters β, τ, N, T during experiments. In the final experiment, the optimal hyperparameters for each algorithm are used for performance comparison. We will deploy image classification with a multinomial logistic regression model for convex tasks and a two-layer CNN model for non-convex tasks. Regarding the CNN model, we follow the structure which is similar to that in [20] with two 5x5 convolution layers (32 and 64 channels for the first and second layer respectively, max pooling size 2x2 is used after each layer), ReLu activation, and a softmax layer at the end of the CNN. Although mini-batch is not mentioned in Alg. 1, the experiments use mini-batch to reduce tackle challenge of finding the optimal local point with a large number of data points.

Effects of step-size parameter β and local iterations τ: We first compare the convergence of FedProxVR and FedAvg in Figs. 2 and 3 in different hyperparameter settings. In both figures, we first choose the value of β, then determine τ based on its upper-bound in Lemma 1 such that the algorithms empirically converge. While the upper-bound of τ only depends on β, its lower bound is determined by parameters such as L and $\tilde{\mu}$ which are more difficult to estimate from the datasets and learning tasks. We start with small values of β and τ and then increase them to observe the convergence behavior of FedProxVR and the effect of the weight vector γ on optimal parameters β and τ.

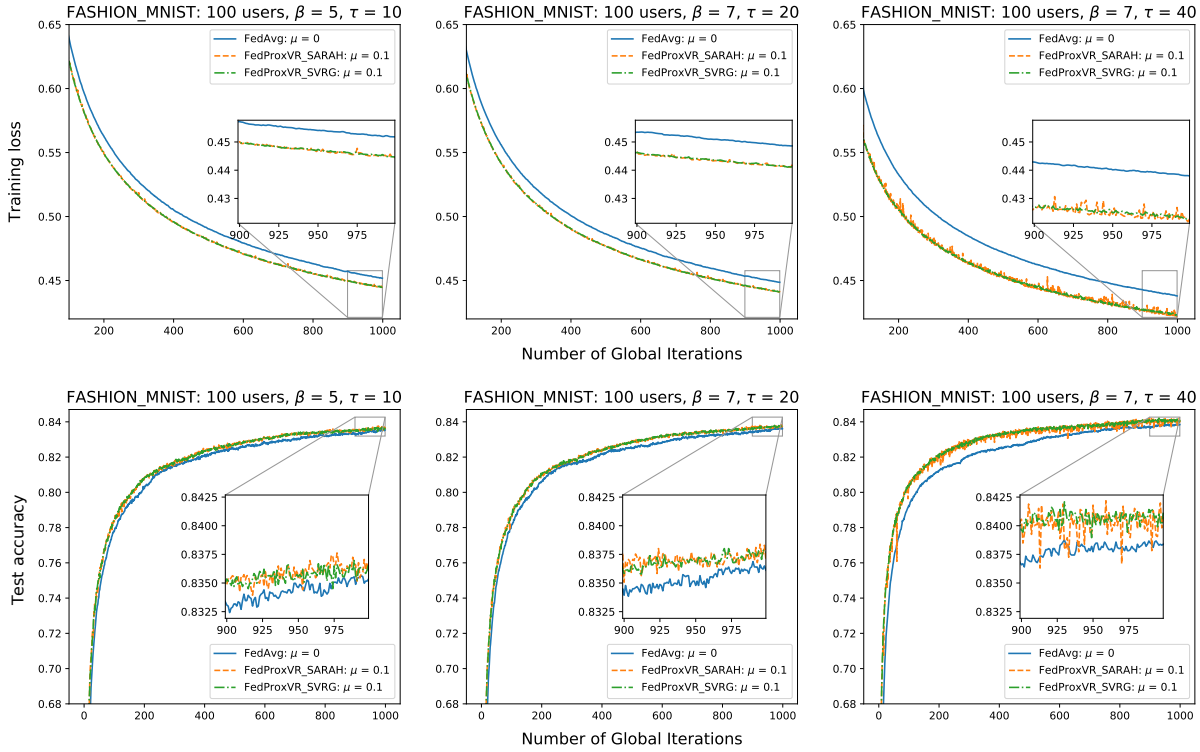

Figure 2: Convergence of FedProxVR with convex task on the Fashion-MNIST dataset. All algorithms use a batch of size $B = 32$.

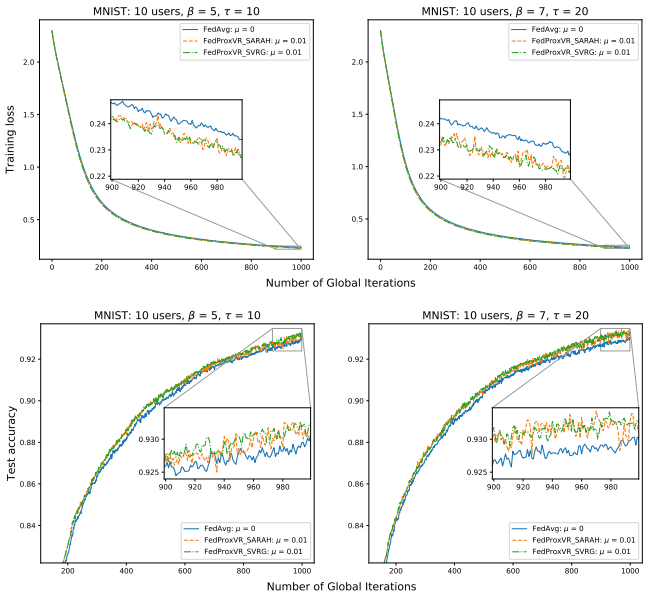

Figure 3: Convergence of FedProxVR with non-convex task (CNN) on MNIST dataset ($B = 64$).

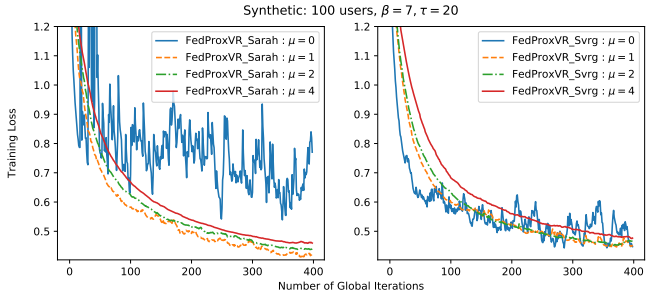

Figure 4: Effect of proximal penalty μ to FedProxVR convergence, with convex task on Synthetic dataset.

In Fig. 2, we observe similar impacts of both β and τ on the convergence of all algorithms on the same convex task. First, when β and τ are smaller (5 and 10, respectively), which is equivalent to larger learning rate η and γ, FedProxVR gains a notable performance improvement from FedAvg. Second, when β and τ are increased to 7 and 20, respectively, not only can we see an increase in the performance gap, the training losses and the test accuracies of all algorithms also converge faster. This verifies the theoretical findings in Section 4.3, and in scenarios in which device-server communication delay is more expensive local communication delay (small γ), increasing the optimal values of τ and β to certain values will reduce the communication cost while at the same time ensuring the algorithms' convergence. Finally, if the value of τ is set to be higher than its upper bound specified by β (for FedProxVR using

Table 1: Comparing the models' test accuracies using their best hyperparameters on a convex task.

Algorithm	τ	β	μ	B	T	Accuracy
FedAvg	10	10	0	16	983	**84.02%**
FedProxVR (SVRG)	20	10	0.1	32	895	**84.12%**
FedProxVR (SARAH)	20	5	0.1	32	965	**84.21%**

Table 2: Comparing the models' test accuracies using their best hyperparameters on a nonconvex task.

Algorithm	τ	β	μ	B	T	Accuracy
FedAvg	20	10	0	16	995	**93.52%**
FedProxVR (SVRG)	20	10	0.01	16	970	**94.06%**
FedProxVR (SARAH)	20	9	0.01	32	958	**93.75%**

SVRG and SARAH) and a (for FedProxVR using SVRG), thus violating Lemma 1, the learning curves of FedProxVR fluctuate much more noticeably, although the performances of FedProxVR and FedAvg are still improved and distinguishable. Therefore, with the choice of τ such that its lower- and upper-bound conditions are satisfied, FedProxVR is expected to converge better than FedAvg.

The performances of FedProxVR and FedAvg on the non-convex task are highlighted in Fig. 3. Here, we observe a similar outcome to our experiment in convex settings, and the performance gap between FedProxVR and FedAvg is slightly larger.

Effects of proximal penalty μ to global iterations T: We evaluate the effect of proximal penalty μ to the convergence of FedProxVR in Fig. 4. Using FedProxVR on the Synthetic dataset, we observe that the training loss of FedProxVR diverges when $\mu = 0$, and increasing $\mu > 0$ stabilizes the loss, allowing it to converge. However, it is also noticeable that larger values of μ will make the convergence of FedProxVR slower. Therefore, μ also reflects the trade-off between the smoothness of the learning curve and convergence speed of FedProxVR.

Performance comparison using optimized parameters: As algorithms behave differently on the same hyperparameters (e.g., μ, τ and η in our experiment), we conduct a random search on carefully chosen ranges of hyperparameters to determine which combination of them would yield the highest test accuracy with respect to each algorithm. The result is captured in Tables 1 and 2. It can be seen that when using their optimized hyperparameters, FedProxVR manages to improve its accuracies from FedAvg on both convex and non-convex tasks. Also, while FedAvg performs better on smaller batch sizes on the convex task, FedProxVR benefits from larger batch sizes. Finally, on both tasks, FedProxVR starts to converge earlier than FedAvg.

6 CONCLUSIONS

In this paper, we propose an algorithm for FL using proximal stochastic variance reduced gradient methods, which can address the heterogeneity challenges of FL due to massively participating devices with non-identically distributed data sources. In the proposed algorithm, each user device is allowed to independently solve its learning problem approximately for a number of local iterations for local model update, which will be sent to the server for global model update. We characterize the convergence analysis for both local and global model updates, which provided several fruitful insights for algorithm design. We also propose how to find the optimal algorithm parameters to minimize the FL training time. Using Tensorflow, we validate theoretical findings by presenting empirical convergence of the proposed algorithm on various real and synthetic data sets to show that our algorithm can boost the convergence speed compared to the SGD-based approaches for FL.

REFERENCES

[1] 2017. We Are Making On-Device AI Ubiquitous. https://www.qualcomm.com/news/onq/2017/08/16/we-are-making-device-ai-ubiquitous.
[2] Mohammad Mohammadi Amiri and Deniz Gunduz. 2019. Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air. *arXiv:1901.00844 [cs, math]* (Jan. 2019). arXiv:1901.00844 [cs, math]
[3] L. Bottou, F. Curtis, and J. Nocedal. 2018. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* 60, 2 (Jan. 2018), 223–311. https://doi.org/10.1137/16M1080173
[4] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. 2018. Accelerated Methods for NonConvex Optimization. *SIAM Journal on Optimization* 28, 2 (Jan. 2018), 1751–1772. https://doi.org/10.1137/17M1114296
[5] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large Scale Distributed Deep Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc., Lake Tahoe, Nevada, 1223–1231.
[6] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. 2012. Optimal Distributed Online Prediction Using Mini-Batches. *J. Mach. Learn. Res.* 13 (Jan. 2012), 165–202.
[7] Charlie Dinh. 2020. CharlieDinh/FederatedLearningWithSVRG. https://github.com/CharlieDinh/FederatedLearningWithSVRG
[8] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. 2016. Mini-Batch Stochastic Approximation Methods for Nonconvex Stochastic Composite Optimization. *Mathematical Programming* 155, 1-2 (Jan. 2016), 267–305. https://doi.org/10.1007/s10107-014-0846-1
[9] Sashank J. Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. 2016. Proximal Stochastic Methods for Nonsmooth Nonconvex Finite-Sum Optimization. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 1145–1153.
[10] Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. 2014. Communication-Efficient Distributed Dual Coordinate Ascent. *arXiv:1409.1458 [cs, math, stat]* (Sept. 2014). arXiv:1409.1458 [cs, math, stat]
[11] Rie Johnson and Tong Zhang. 2013. Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'13)*. Curran Associates Inc., Lake Tahoe, Nevada, 315–323.
[12] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv:1610.02527 [cs]* (Oct. 2016). arXiv:1610.02527 [cs]
[13] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *http://arxiv.org/abs/1610.05492* (Oct. 2016). arXiv:1610.05492
[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521, 7553 (May 2015), 436–444. https://doi.org/10.1038/nature14539
[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11 (Nov. 1998), 2278–2324. https://doi.org/10.1109/5.726791
[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2019. Federated Optimization for Heterogeneous Networks. In *Proceedings of the 1st Adaptive & Multitask Learning, ICML Workshop, 2019*. Long Beach, CA, 16.
[17] Zhize Li and Jian Li. 2018. A Simple Proximal Stochastic Gradient Method for Nonsmooth Nonconvex Optimization. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Montréal, Canada, 5569–5579.
[18] Zhize Li and Jian Li. 2018. A Simple Proximal Stochastic Gradient Method for Nonsmooth Nonconvex Optimization. *arXiv:1802.04477 [cs, math, stat]* (Feb. 2018). arXiv:1802.04477 [cs, math, stat]
[19] Chenxin Ma, Jakub Konečný, Martin Jaggi, Virginia Smith, Michael I. Jordan, Peter Richtárik, and Martin Takáč. 2017. Distributed Optimization with Arbitrary

Local Solvers. *Optimization Methods and Software* 32, 4 (July 2017), 813–848. https://doi.org/10.1080/10556788.2016.1278445
[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*. 1273–1282.
[21] Yurii Nesterov. 2018. *Lectures on Convex Optimization*. Vol. 137. Springer International Publishing. https://doi.org/10.1007/978-3-319-91578-4
[22] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. 2017. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, International Convention Centre, Sydney, Australia, 2613–2621.
[23] Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. [n.d.]. ProxSARAH: An Efficient Algorithmic Framework for Stochastic Composite Nonconvex Optimization. ([n. d.]), 45.
[24] Sashank J. Reddi, Jakub Konečný, Peter Richtárik, Barnabás Póczós, and Alex Smola. 2016. AIDE: Fast and Communication Efficient Distributed Optimization. *arXiv:1608.06879 [cs, math, stat]* (Aug. 2016). arXiv:1608.06879 [cs, math, stat]
[25] Sashank J. Reddi, Suvrit Sra, Barnabas Poczos, and Alex Smola. 2016. Fast Stochastic Methods for Nonsmooth Nonconvex Optimization. *arXiv:1605.06900 [cs, math, stat]* (May 2016). arXiv:1605.06900 [cs, math, stat]
[26] Ohad Shamir, Nathan Srebro, and Tong Zhang. 2014. Communication-Efficient Distributed Optimization Using an Approximate Newton-Type Method. In *ICML*. Beijing, China, II–1000–II–1008.
[27] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated Multi-Task Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4424–4434.
[28] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takáč, Michael I. Jordan, and Martin Jaggi. 2018. CoCoA: A General Framework for Communication-Efficient Distributed Optimization. *Journal of Machine Learning Research* 18, 230 (2018), 1–49.
[29] Nguyen H. Tran, Wei Bao, Albert Zomaya, Minh N.H. Nguyen, and Choong Seon Hong. 2019. Federated Learning over Wireless Networks: Optimization Model Design and Analysis. In *IEEE INFOCOM 2019*. Paris, France.
[30] Jianyu Wang and Gauri Joshi. 2018. Cooperative SGD: A Unified Framework for the Design and Analysis of Communication-Efficient SGD Algorithms. *arXiv:1808.07576 [cs, stat]* (Aug. 2018). arXiv:1808.07576 [cs, stat]
[31] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (June 2019), 1205–1221. https://doi.org/10.1109/JSAC.2019.2904348
[32] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. 2018. SpiderBoost: A Class of Faster Variance-Reduced Algorithms for Nonconvex Optimization. *arXiv:1810.10690 [cs, math, stat]* (Oct. 2018). arXiv:1810.10690 [cs, math, stat]
[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]* (Aug. 2017). arXiv:1708.07747 [cs, stat]
[34] Sixin Zhang, Anna E Choromanska, and Yann LeCun. 2015. Deep Learning with Elastic Averaging SGD. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 685–693.

A REVIEW OF USEFUL EXISTING RESULTS

A.1 Strong convexity

Assuming that $J_n(\cdot)$ is $\tilde{\mu}$-strongly convex, $\tilde{\mu} > 0$, according to [21][Theorem 2.1.10, eqs. (2.1.24) and (2.1.26)], we have

$$2\tilde{\mu}(J_n(w) - J_n(w^*)) \leq \left\|\nabla J(w)\right\|^2, \forall w. \tag{25}$$

$$\tilde{\mu}\left\|w - w^*\right\| \leq \left\|\nabla J(w)\right\|, \forall w. \tag{26}$$

where w^* is the solution to problem (6), i.e., $\nabla J(w^*) = 0$.

A.2 Proximal stochastic gradient methods

In the context of Alg. 1, suppose the problem (6) is solved by each device's local iterations (in a fixed global iteration s)

$$w_{n,s}^{(t+1)} = \text{prox}_{\eta h_s}\left[w_{n,s}^{(t)} - \eta v_{n,s}^{(t)}\right] \tag{27}$$

where $v_{n,s}^{(t)}$ is a stochastic (variance reduction or recursive) gradient descent (i.e., SVRG or SARAH) vector. Defining

$$\hat{w}_{n,s}^{(t+1)} := \text{prox}_{\eta h_s}\left[w_{n,s}^{(t)} - \eta \nabla F_n(w_{n,s}^{(t)})\right], \tag{28}$$

which is a proximal (full) gradient vector, then according to Eq. (25) of [18], we obtain the following

$$\begin{aligned}
&\mathbb{E}\left[J_n(w_{n,s}^{(t+1)}) - J_n(w_{n,s}^{(t)})\right] \\
&\leq \mathbb{E}\Big[-\Big(\frac{1}{3\eta} - L\Big)\left\|w_{n,s}^{(t)} - \hat{w}_{n,s}^{(t+1)}\right\|^2 + \eta\left\|\nabla F_n(w_{n,s}^{(t)}) - v_{n,s}^{(t)}\right\|^2 \\
&\quad - \Big(\frac{5}{8\eta} - \frac{L}{2}\Big)\left\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\right\|^2\Big]
\end{aligned} \tag{29}$$

We next define the *gradient mapping* [8]

$$G_n(w) := \frac{1}{\eta}\left(w - \text{prox}_{\eta h_s}(w - \eta \nabla F_n(w))\right) \tag{30}$$

Then we see that

$$\left\|w_{n,s}^{(t)} - \hat{w}_{n,s}^{(t+1)}\right\|^2 = \eta^2\left\|G_n(w_{n,s}^{(t)})\right\|^2. \tag{31}$$

A.3 SARAH

We rewrite the core of SARAH algorithm [22]

$$v_{n,s}^{(t)} = \nabla f_{i_t}(w_{n,s}^{(t)}) - \nabla f_{i_t}(w_{n,s}^{(t-1)}) + v_{n,s}^{(t-1)} \tag{32}$$

According to Lemma 2 in [22], with the L-smoothness in Assumption 1 and $v_{n,s}^{(t)}$ is updated in (32), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F_n(w_{n,s}^{(t)}) - v_{n,s}^{(t)}\right\|^2\right] \leq \sum_{j=1}^{t}\mathbb{E}\left[\left\|v_{n,s}^{(j)} - v_{n,s}^{(j-1)}\right\|^2\right] \\
&= \sum_{j=1}^{t}\left\|\nabla f_{i_t}(w_{n,s}^{(j)}) - \nabla f_{i_t}(w_{n,s}^{(j-1)})\right\|^2 \leq \sum_{j=1}^{t} L^2\left\|w_{n,s}^{(j)} - w_{n,s}^{(j-1)}\right\|^2.
\end{aligned} \tag{33}$$

A.4 SVRG

We rewrite the core of SVRG algorithm [11]

$$v_{n,s}^{(t)} = \nabla f_{i_t}(w_{n,s}^{(t)}) - \nabla f_{i_t}(\bar{w}^{(s-1)}) + \nabla F_n(\bar{w}^{(s-1)}) \tag{34}$$

According to Lemma 3 of [25] and Eq. (29) in [18], we have

$$\mathbb{E}\left[\left\|\nabla F_n(w_{n,s}^{(t)}) - v_{n,s}^{(t)}\right\|^2\right] \leq L^2\,\mathbb{E}\left[\left\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\right\|^2\right]. \tag{35}$$

B PROOF OF LEMMA 1

The following result will be useful later.

Lemma 2.

$$\left\|\nabla J_n(w)\right\|^2 \leq 2(\mu^2\eta^2 + 1)\left\|G_n(w)\right\|^2, \forall w. \tag{36}$$

Proof: For simplicity, by defining $w^+ = \text{prox}_{\eta h_s}(w - \eta \nabla F_n(w))$, from (30), we have

$$G_n(w) = \frac{1}{\eta}(w - w^+) \tag{37}$$

$$= \nabla F_n(w) + \nabla h_s(w^+). \tag{38}$$

Then we obtain

$$\left\|\nabla J_n(w)\right\|^2 \leq 2\left\|\nabla J_n(w) - G_n(w)\right\|^2 + 2\left\|G_n(w)\right\|^2 \tag{39}$$

$$= 2\left\|\nabla h_s(w) - \nabla h_s(w^+)\right\|^2 + 2\left\|G_n(w)\right\|^2 \tag{40}$$

$$= 2\mu^2\left\|w - w^+\right\|^2 + 2\left\|G_n(w)\right\|^2 \tag{41}$$

$$= 2(\mu^2\eta^2 + 1)\left\|G_n(w)\right\|^2, \tag{42}$$

where (39) holds due to Young's inequality (with $\alpha = 1$)

$$\|x+y\|^2 \le (1+\alpha)\|x\|^2 + (1+1/\alpha)\|y\|^2, \forall x, y \in \mathbb{R}^d, \quad (43)$$

(40) holds due to (38) and definition of $J_n(\cdot)$, (41) uses the definition of $h(.)$, and (42) uses (37).

B.0.1 FedProxVR with SARAH. By considering one global iteration s, we see that Alg. 1 with $v_{n,s}^{(t)}$ update (32) is a proximal SARAH to solve problem (6). Thus, using existing results (29), (30), and (33), we have

$$\begin{aligned}
&\mathbb{E}\left[J_n(w_{n,s}^{(t+1)}) - J_n(w_{n,s}^{(t)})\right] \\
&\le \mathbb{E}\Big[-\Big(\frac{1}{3\eta} - L\Big)\eta^2\|G_n(w_{n,s}^{(t)})\|^2 + \eta L^2 \sum_{j=1}^{t}\|w_{n,s}^{(j)} - w_{n,s}^{(j-1)}\|^2 \\
&\quad - \Big(\frac{5}{8\eta} - \frac{L}{2}\Big)\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2\Big].
\end{aligned}$$

By setting $\eta = \frac{1}{\beta L}$, and summing the above inequality over $0 \le t \le \tau$, we have:

$$\begin{aligned}
&\mathbb{E}\left[J_n(w_{n,s}^{(\tau+1)}) - J_n(w_{n,s}^{(0)})\right] \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3L\beta^2}\Big)\sum_{t=0}^{\tau}\|G_n(w_{n,s}^{(t)})\|^2 + \frac{L}{\beta}\sum_{t=1}^{\tau}\sum_{j=1}^{t}\|w_{n,s}^{(j)} - w_{n,s}^{(j-1)}\|^2 \quad (44)\\
&\quad - L\Big(\frac{5\beta-4}{8}\Big)\sum_{t=0}^{\tau}\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2\Big] \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3L\beta^2}\Big)\sum_{t=0}^{\tau}\|G_n(w_{n,s}^{(t)})\|^2 + \frac{L}{\beta}\sum_{t=0}^{\boldsymbol{\tau-1}}\tau\,\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2 \\
&\quad -L\Big(\frac{5\beta-4}{8}\Big)\sum_{t=0}^{\boldsymbol{\tau-1}}\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2\Big], \quad (45)\\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3L\beta^2}\Big)\sum_{t=0}^{\tau}\|G_n(w_{n,s}^{(t)})\|^2\Big], \quad (46)
\end{aligned}$$

where (44) is due to $\mathbb{E}\Big[\|\nabla F_n(w_{n,s}^{(0)}) - v_{n,s}^{(0)}\|^2\Big] = 0$ and (45) holds when $\beta \ge 4/5$ and the following fact

$$\begin{aligned}
\sum_{t=1}^{\tau}\sum_{j=1}^{t}\|w_{n,s}^{(j)} - w_{n,s}^{(j-1)}\|^2 &\le \sum_{t=1}^{\tau}\sum_{j=1}^{\tau}\|w_{n,s}^{(j)} - w_{n,s}^{(j-1)}\|^2 \\
= \sum_{t=1}^{\tau}\tau\|w_{n,s}^{(t)} - w_{n,s}^{(t-1)}\|^2 &= \sum_{t=0}^{\tau-1}\tau\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2 \quad (47)
\end{aligned}$$

and (46) holds when we choose

$$0 \le \tau \le \frac{5\beta^2 - 4\beta}{8}. \quad (48)$$

From (46), we have:

$$\begin{aligned}
\frac{\beta-3}{3L\beta^2}\sum_{t=0}^{\tau}\mathbb{E}\left[\|G_n(w_{n,s}^{(t)})\|^2\right] &\le \mathbb{E}\left[J_n(w_{n,s}^{(0)}) - J_n(w_{n,s}^{(\tau+1)})\right] \\
&\le \mathbb{E}\left[J_n(w_{n,s}^{(0)}) - J_n(w_{n,s}^{*})\right] \quad (49)\\
&\le \frac{\mathbb{E}\left[\|\nabla J_n(w_{n,s}^{(0)})\|^2\right]}{2\tilde{\mu}} \quad (50)
\end{aligned}$$

where (49) uses $J_n(w_{n,s}^{*}) \le J_n(w_{n,s}^{(\tau+1)})$ and (50) uses (25).

Finally, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\nabla J_n(w_n^{(s)})\|^2 \mid \bar{w}^{(s-1)}\right] \\
&\le 2\Big(\frac{\mu^2}{\beta^2L^2} + 1\Big)\mathbb{E}\left[\|G_n(w_n^{(s)})\|^2 \mid \bar{w}^{(s-1)}\right] \quad (51)\\
&= 2\Big(\frac{\mu^2}{\beta^2L^2} + 1\Big)\sum_{t=0}^{\tau}\frac{1}{\tau+1}\mathbb{E}\left[\|G_n(w_{n,s}^{(t)})\|^2 \mid \bar{w}^{(s-1)}\right] \quad (52)\\
&\le 2\Big(\frac{\mu^2}{\beta^2L^2} + 1\Big)\frac{1}{2\tilde{\mu}}\Big(\frac{3L\beta^2}{\beta-3}\Big)\frac{1}{\tau+1}\mathbb{E}\left[\|\nabla J_n(w_{n,s}^{(0)})\|^2 \mid \bar{w}^{(s-1)}\right] \quad (53)\\
&\le \frac{3(\beta^2L^2+\mu^2)}{\tilde{\mu}L(\beta-3)}\frac{1}{\tau}\|\nabla F_n(\bar{w}^{(s-1)})\|^2 \quad (54)
\end{aligned}$$

where (51) is due to Lemma 2, (52) holds because $w_n^{(s)}$ is chosen uniformly at random from $\{w_{n,s}^{(t)}\}_{t=0,\dots,\tau}$, (53) uses (50), and (54) is due to $w_{n,s}^{(0)} = \bar{w}^{(s-1)}$. From (54), in order to have

$$\begin{aligned}
\mathbb{E}[\|\nabla J_n(w_n^{(s)})\| \mid \bar{w}^{(s-1)}] &\le \mathbb{E}\Big[\|\nabla J_n(w_n^{(s)})\|^2 \mid \bar{w}^{(s-1)}\Big]^{\frac{1}{2}} \\
&\le \theta\,\|\nabla F_n(\bar{w}^{(s-1)})\|
\end{aligned}$$

we obtain

$$\tau \ge \frac{3(\beta^2L^2+\mu^2)}{\theta^2\tilde{\mu}L(\beta-3)} \ge 0. \quad (55)$$

B.0.2 FedProxVR with SVRG. Similarly, using (29) and (35), and setting $\eta = \frac{1}{\beta L}$, we have

$$\begin{aligned}
&\mathbb{E}\left[J_n(w_{n,s}^{(t+1)}) - J_n(w_{n,s}^{(t)})\right] \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta}{3} - 1\Big)\frac{1}{L\beta^2}\|G_n(w_{n,s}^{(t)})\|^2 + \frac{L}{\beta}\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\|^2 \\
&\quad - L\Big(\frac{5\beta-4}{8}\Big)\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2\Big] \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta}{3} - 1\Big)\frac{1}{L\beta^2}\|G_n(w_{n,s}^{(t)})\|^2 \\
&\quad + L\Big(\frac{1}{\beta} + \frac{1}{\sqrt{at}}\cdot\frac{5\beta-4}{8}\Big)\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\|^2 \\
&\quad - \Big(\frac{L}{\sqrt{at}+1}\cdot\frac{5\beta-4}{8}\Big)\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\|^2\Big], \quad (56)
\end{aligned}$$

where (56) holds by using Young's inequality

$$\begin{aligned}
\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\|^2 \le (1+\frac{1}{\alpha})\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\|^2 \quad (57)\\
+ (1+\alpha)\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2,
\end{aligned}$$

especially by choosing $\alpha = \sqrt{a(t+1)}, a > 0$, we have:

$$\begin{aligned}
&-\|w_{n,s}^{(t+1)} - w_{n,s}^{(t)}\|^2 \\
&\le -\frac{1}{1+\alpha}\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\|^2 + (\frac{1+1/\alpha}{1+\alpha})\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\|^2 \\
&= -\frac{1}{1+\sqrt{a(t+1)}}\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\|^2 + \frac{1}{\sqrt{a(t+1)}}\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\|^2. \quad (58)
\end{aligned}$$

Telescoping (56) over $0 \le t \le \tau$, we have:

$$\begin{aligned}
&\mathbb{E}\left[J_n(w_{n,s}^{(\tau+1)}) - J_n(w_{n,s}^{(0)})\right] \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3L\beta^2}\Big)\sum_{t=0}^{\tau}\big\|G_n(w_{n,s}^{(t)})\big\|^2 \\
&\quad - \sum_{t=0}^{\tau}\frac{L}{1+\sqrt{a(t+1)}}\Big(\frac{5\beta-4}{8}\Big)\big\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\big\|^2 \\
&\quad + \sum_{t=1}^{\tau} L\Big(\frac{1}{\sqrt{a(t+1)}}\cdot\frac{5\beta-4}{8} + \frac{1}{\beta}\Big)\big\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\big\|^2\Big] \qquad (59)\\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3L\beta^2}\Big)\sum_{t=0}^{\tau}\big\|G_n(w_{n,s}^{(t)})\big\|^2 \\
&\quad - \sum_{t=0}^{\tau} L\Big(\delta_t\cdot\frac{5\beta-4}{8} - \frac{1}{\beta}\Big)\big\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\big\|^2\Big] \qquad (60)\\
&\quad \text{with } \delta_t := \frac{1}{1+\sqrt{a(t+1)}} - \frac{1}{\sqrt{a(t+2)}} \\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3\beta^2 L}\Big)\sum_{t=0}^{\tau}\big\|G_n(w_{n,s}^{(t)})\big\|^2 \\
&\quad - \sum_{t=0}^{\tau} L\Big(\frac{1}{a(t+2)}\cdot\frac{5\beta-4}{8} - \frac{1}{\beta}\Big)\big\|w_{n,s}^{(t+1)} - \bar{w}^{(s-1)}\big\|^2\Big] \qquad (61)\\
&\le \mathbb{E}\Big[-\Big(\frac{\beta-3}{3\beta^2 L}\Big)\sum_{t=0}^{\tau}\big\|G_n(w_{n,s}^{(t)})\big\|^2\Big], \qquad (62)
\end{aligned}$$

where (59) is due to $\big\|w_{n,s}^{(t)} - \bar{w}^{(s-1)}\big\| = 0$ when $t = 0$, (61) holds because

$$\delta_t = \frac{\sqrt{a(t+2)} - (1+\sqrt{a(t+1)})}{\sqrt{a(t+2)}(1+\sqrt{a(t+1)})} \ge \frac{1}{a(t+2)},\ \forall 1 \le t \le \tau, \qquad (63)$$

when

$$\sqrt{a(t+2)} - (1+\sqrt{a(t+1)}) \ge 1, \forall t \le \tau, \qquad (64)$$

$$\Leftrightarrow a - 4 \ge 4\sqrt{a(\tau+1)} \qquad (65)$$

and (62) holds with the condition

$$\tau \le \frac{5\beta^2 - 4\beta}{8\,a} - 2. \qquad (66)$$

Since (62) is the same to (46), the rest of the proof follows similarly to the steps from (50) to (54), thus obtaining the same lower bound of τ as in (55).

C PROOF OF THEOREM 1

We first define the distribution $p_n = D_n/D$, $\forall n$, with its expectation notation $\mathbb{E}_n$ (to avoid the confusion with $\mathbb{E}[\cdot]$ which only deals with all randomness in FedProxVR), we have

$$\mathbb{E}_n\big[w_n^{(s)}\big] = \sum_{n=1}^{N} p_n w_n^{(s)} = \bar{w}^{(s)}. \qquad (67)$$

By definitions of $J_n(.)$ and $h(.)$

$$\nabla J_n(w_n^{(s)}) = \nabla F_n(w_n^{(s)}) + \mu(w_n^{(s)} - \bar{w}^{(s-1)}). \qquad (68)$$

Thus, we have

$$\mathbb{E}_n\big[\nabla J_n(w_n^{(s)}) - \nabla F_n(w_n^{(s)})\big] = \mu\,(\bar{w}^{(s)} - \bar{w}^{(s-1)}). \qquad (69)$$

We first provide the following useful result

Lemma 3. *Conditioning on $\bar{w}^{(s-1)}$, we have*

$$\mathbb{E}\left[\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2\right] \le \frac{4(1+\theta^2)(1+\bar{\sigma}^2)}{\tilde{\mu}^2}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2.$$

Proof: Defining $w_s^{n*} = \arg\min_{w_n^s} J_n(w_n^s)$, we obtain

$$\big\|w_s^{n*} - \bar{w}^{(s-1)}\big\|^2 \overset{(26)}{\le} \frac{1}{\tilde{\mu}^2}\big\|\nabla J_n(\bar{w}^{(s-1)})\big\|^2 \overset{(68)}{=} \frac{1}{\tilde{\mu}^2}\big\|\nabla F_n(\bar{w}^{(s-1)})\big\|^2.$$

Similarly, we have

$$\mathbb{E}\left[\big\|w_s^{n*} - w_n^{(s)}\big\|^2\right] \overset{(26)}{\le} \frac{1}{\tilde{\mu}^2}\mathbb{E}\left[\big\|\nabla J_n(w_n^{(s)})\big\|^2\right] \le \frac{\theta^2}{\tilde{\mu}^2}\big\|\nabla F_n(\bar{w}^{(s-1)})\big\|^2, \qquad (70)$$

where (70) uses Lemma 1. Then, using the Young inequality (43) (with $\alpha = 1$), we have

$$\begin{aligned}
\mathbb{E}\left[\big\|w_n^{(s)} - \bar{w}^{(s-1)}\big\|^2\right] &\le \mathbb{E}\left[2\big\|w_s^{n*} - w_n^{(s)}\big\|^2 + 2\big\|w_s^{n*} - \bar{w}^{(s-1)}\big\|^2\right] \\
&\le \frac{2(1+\theta^2)}{\tilde{\mu}^2}\big\|\nabla F_n(\bar{w}^{(s-1)})\big\|^2. \qquad (71)
\end{aligned}$$

We have the next useful result

$$\begin{aligned}
&\mathbb{E}_n\big[\big\|\nabla F_n(\bar{w}^{(s-1)})\big\|^2\big] \\
&\overset{(43)}{\le} \mathbb{E}_n\big[2\big\|\nabla F_n(\bar{w}^{(s-1)}) - \nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2\big] + 2\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2 \\
&\le 2(1+\bar{\sigma}^2)\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2, \qquad (72)
\end{aligned}$$

where (72) holds due to Assumption 1. Finally, we have

$$\begin{aligned}
\mathbb{E}\left[\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2\right] &= \mathbb{E}\left[\big\|\mathbb{E}_n[w_n^{(s)} - \bar{w}^{(s-1)}]\big\|^2\right] \\
&\le \mathbb{E}\left[\mathbb{E}_n\big[\big\|w_n^{(s)} - \bar{w}^{(s-1)}\big\|\big]^2\right] \qquad (73)\\
&\le \frac{2(1+\theta^2)}{\tilde{\mu}^2}\mathbb{E}_n\big[\big\|\nabla F_n(\bar{w}^{(s-1)})\big\|^2\big] \qquad (74)\\
&\le \frac{4(1+\theta^2)(1+\bar{\sigma}^2)}{\tilde{\mu}^2}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2, \qquad (75)
\end{aligned}$$

where (73) uses Jensen's inequality, (74) holds due to exchange of expectation and using (71), and (75) uses (72).

Since $\bar{F}(.)$ is also L-Lipschitz smooth (i.e., $\big\|\nabla\bar{F}(w) - \nabla\bar{F}(w')\big\| \le \mathbb{E}_n\big\|\nabla F_n(w) - \nabla F_n(w')\big\| \le L\big\|w - w'\big\|$ by using Jensen's inequality twice and Assumption 1), we have

$$\begin{aligned}
&\bar{F}(\bar{w}^{(s)}) - \bar{F}(\bar{w}^{(s-1)}) \le \langle\nabla\bar{F}(\bar{w}^{(s-1)}), \bar{w}^{(s)} - \bar{w}^{(s-1)}\rangle + \frac{L}{2}\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2 \\
&= \frac{1}{\mu}(\bar{w}^{(s-1)}), \mathbb{E}_n\big[\nabla J_n(w_n^{(s)}) - \nabla F_n(w_n^{(s)})\big] + \frac{L}{2}\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2 \qquad (76)\\
&= \frac{1}{\mu}\big\langle\nabla\bar{F}(\bar{w}^{(s-1)}), \mathbb{E}_n\big[\nabla J_n(w_n^{(s)}) - \nabla F_n(w_n^{(s)})\big] + \nabla\bar{F}(\bar{w}^{(s-1)})\big\rangle \\
&\quad - \frac{1}{\mu}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2 + \frac{L}{2}\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2 \qquad (77)\\
&= \frac{1}{\mu}\big\langle\nabla\bar{F}(\bar{w}^{(s-1)}), \mathbb{E}_n\big[\nabla F_n(\bar{w}^{(s-1)}) - \nabla F_n(w_n^{(s)})\big]\big\rangle \\
&\quad + \frac{1}{\mu}\big\langle\nabla\bar{F}(\bar{w}^{(s-1)}), \mathbb{E}_n\big[\nabla J_n(w_n^{(s)})\big]\big\rangle \\
&\quad - \frac{1}{\mu}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2 + \frac{L}{2}\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2 \qquad (78)\\
&\le \frac{1}{\mu}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|\cdot\mathbb{E}_n\big[\big\|\nabla F_n(\bar{w}^{(s-1)}) - \nabla F_n(w_n^{(s)})\big\|\big] \\
&\quad + \frac{1}{\mu}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|\cdot\mathbb{E}_n\big[\big\|\nabla J_n(w_n^{(s)})\big\|\big] \\
&\quad - \frac{1}{\mu}\big\|\nabla\bar{F}(\bar{w}^{(s-1)})\big\|^2 + \frac{L}{2}\big\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\big\|^2, \qquad (79)
\end{aligned}$$

where (76) uses (69), (77) holds by subtracting and adding $\nabla F_n(w_n^{(s)})$, and (79) uses Cauchy-Schwarz and Jensen's inequalities.

Taking conditional expectation of (79), we have

$$\begin{aligned}
&\mathbb{E}\left[\bar{F}(\bar{w}^{(s)}) - \bar{F}(\bar{w}^{(s-1)}) \mid \bar{w}^{(s-1)}\right] \\
&\leq \frac{L}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| \cdot \mathbb{E}_n\left[\mathbb{E}\left[\left\|\bar{w}^{(s-1)} - w_n^{(s)}\right\|\right]\right] && (80)\\
&\quad + \frac{1}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| \cdot \mathbb{E}_n\left[\mathbb{E}\left[\left\|\nabla J_n(w_n^{(s)})\right\|\right]\right] && (81)\\
&\quad - \frac{1}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|^2 + \frac{L}{2}\mathbb{E}\left[\left\|\bar{w}^{(s)} - \bar{w}^{(s-1)}\right\|^2\right] && (82)\\
&\leq \frac{L}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| \cdot \frac{2}{\tilde{\mu}}\sqrt{(1+\theta^2)(1+\bar{\sigma}^2)}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| && (83)\\
&\quad + \frac{1}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| \cdot \theta\sqrt{2(1+\bar{\sigma}^2)}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\| && (84)\\
&\quad - \frac{1}{\mu}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|^2 + \frac{2L(1+\theta^2)(1+\bar{\sigma}^2)}{\tilde{\mu}^2}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|^2 && (85)\\
&\leq -\frac{1}{\mu}\left(1 - \frac{2L}{\tilde{\mu}}\sqrt{(1+\theta^2)(1+\bar{\sigma}^2)} - \theta\sqrt{2(1+\bar{\sigma}^2)}\right.\\
&\qquad\qquad \left. - \frac{2L\mu}{\tilde{\mu}^2}(1+\theta^2)(1+\bar{\sigma}^2)\right)\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|^2, && (86)\\
&= -\Theta\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|^2. && (87)
\end{aligned}$$

where (80) is due to L-smoothness of $F_n(.)$, and we interchange expectations in (80) and (81). (83) holds because condition on $\bar{w}^{(s-1)}$, we have

$$\begin{aligned}
&\mathbb{E}_n\left[\mathbb{E}\left[\left\|\bar{w}^{(s-1)} - w_n^{(s)}\right\| \mid \bar{w}^{(s-1)}\right]\right] \leq \mathbb{E}_n\left[\mathbb{E}\left[\left\|\bar{w}^{(s-1)} - w_n^{(s)}\right\|^2\right]^{\frac{1}{2}}\right]\\
&\overset{(71)}{\leq} \frac{\sqrt{2(1+\theta^2)}}{\tilde{\mu}}\mathbb{E}_n\left[\left\|\nabla F_n(\bar{w}^{(s-1)})\right\|\right] \leq \frac{\sqrt{2(1+\theta^2)}}{\tilde{\mu}}\left(\mathbb{E}_n\left[\left\|\nabla F_n(\bar{w}^{(s-1)})\right\|^2\right]\right)^{\frac{1}{2}}\\
&\overset{(72)}{\leq} \frac{2}{\tilde{\mu}}\sqrt{(1+\theta^2)(1+\bar{\sigma}^2)}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|
\end{aligned}$$

Similarly, (84) holds because

$$\begin{aligned}
\mathbb{E}_n\left[\mathbb{E}\left[\left\|\nabla J_n(w_n^{(s)})\right\|\right]\right] &\overset{\text{Lemma 1}}{\leq} \theta\,\mathbb{E}_n\left[\left\|\nabla F_n(\bar{w}^{(s-1)})\right\|\right]\\
&\overset{(72)}{\leq} \theta\sqrt{2(1+\bar{\sigma}^2)}\left\|\nabla\bar{F}(\bar{w}^{(s-1)})\right\|.
\end{aligned}$$

And (85) uses Lemma 3.

Taking expectation of (86) over entire history and telescoping s from 1 to T, we have

$$\frac{1}{T}\sum_{s=1}^{T}\mathbb{E}\left[\left\|\bar{F}(\bar{w}^{(s)})\right\|^2\right] \leq \frac{\mathbb{E}\left[\bar{F}(\bar{w}^{(0)}) - \bar{F}(\bar{w}^{(T)})\right]}{\Theta T} \leq \frac{\mathbb{E}\left[\bar{F}(\bar{w}^{(0)}) - \bar{F}(\bar{w}^{*})\right]}{\Theta T}.$$