GENERALIZATION OF FEDAVG UNDER CONSTRAINED POLYAK-ŁOJASIEWICZ TYPE CONDITIONS: A SINGLE HIDDEN LAYER NEURAL NETWORK ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we study the optimization and the generalization performance of the widely used FedAvg algorithm for solving Federated Learning (FL) problems. We analyze the generalization performance of FedAvg by handling the optimization error and the Rademacher complexity. Towards handling optimization error, we propose novel constrained Polyak-Łojasiewicz (PL)-type conditions on the objective function that ensure existence of a global optimal to which FedAvg converges linearly after $\mathcal{O}(\log(1/\epsilon))$ rounds of communication, where ϵ is the desired optimality gap. Importantly, we demonstrate that a class of single hidden layer neural networks satisfies the proposed constrained PL-type conditions required to establish the linear convergence of FedAvg as long as $m > nK/d$, where m is the width of the neural network, K is the number of clients, n is the number of samples at each client, and d is the feature dimension. We then bound the Rademacher complexity for this class of neural networks and establish that both Rademacher complexity and the generalization error of FedAvg decrease at an optimal rate of $\mathcal{O}(1/\sqrt{n})$. We further show that increasing the number of clients K decreases the generalization error at the rate of $\mathcal{O}(1/\sqrt{n} + 1/\sqrt{nK})$.

1 INTRODUCTION

Federated learning (FL) is a distributed learning paradigm where multiple client devices collaborate with the help of a server to solve a joint problem while keeping the data of each client private (Kairouz et al., 2021). A typical FL problem aims to solve $\min_{\boldsymbol{w}} \sum_{k=1}^{K} \Phi_k(\boldsymbol{w})$, where $\Phi_k(\boldsymbol{w})$ is the loss at the k^{th} client and $\boldsymbol{w}$ refers to the joint model the clients aim to learn. A standard and most widely adopted algorithm to solve the FL problem is the Federated Averaging (FedAvg) algorithm first proposed in (McMahan et al., 2017). Consequently, the study of the convergence performance of FedAvg has received wide attention (Konečnỳ et al., 2015; Stich, 2018; McMahan et al., 2017; Li et al., 2020; Zhou & Cong, 2017b). However, when it comes to ensuring generalization guarantees for FedAvg, the problem has not received significant attention, partially because of the challenging nature of the problem (Mohri et al., 2019; Sun et al., 2023; Hu et al., 2022). To prove the generalization guarantees for FedAvg, we need to bound (a) the **optimization error** (on empirical loss) achieved by FedAvg, and (b) the **complexity measure** such as the Rademacher complexity of the model (Arora et al., 2019; Mohri et al., 2019; 2018). The major challenge in guaranteeing good generalization performance is to bound both (a) and (b) above, which are often contradictory, i.e., proving optimization guarantees usually rely on restrictive assumptions on the loss landscape like (strong)-convexity or Polyak-Łojasiewicz (PL) inequality to be satisfied over the entire parameter space Haddadpour et al. (2019); Haddadpour & Mahdavi (2019) while the Rademacher complexity is large for an unbounded parameter space [see Theorem 5.10 (Mohri et al., 2018)]. Therefore, bounding both (a) and (b) simultaneously is challenging, thereby making it difficult to provide satisfactory generalization guarantees for FedAvg. To address these challenges in this work:

➢ We first analyze the convergence of FedAvg and establish **linear convergence** under a **new set of assumptions** that are only required to be satisfied **locally**. Importantly, to highlight the practicality of the assumptions, we establish that the proposed **assumptions are naturally satisfied by a single hidden-layer Neural Network (NN)**.

➢ We then study the generalization guarantees of FedAvg for the single hidden-layer NN and show

that the proposed local assumptions lead to a **Rademacher complexity that goes down with the number of samples** n **as** $\mathcal{O}(1/\sqrt{n})$. Specifically, our analysis captures the effects of local samples, the number of clients, and model sizes on the performance of the FedAvg algorithm.

In the following, we discuss specific challenges and the drawbacks of the current state-of-the-art with respect to challenges (a) and (b) discussed above.

Convergence of FedAvg. As discussed earlier, several works have analyzed the convergence performance of FedAvg under various settings. In the non-convex regime, multiple works have established the convergence of FedAvg to a stationary point (local optimal) (Konečnỳ et al., 2015; Stich, 2018; McMahan et al., 2017; Li et al., 2020; Zhou & Cong, 2017b). However, the local optimal does not guarantee a small empirical loss, and hence cannot be used to provide generalization guarantees. Some works have shown convergence of FedAvg to global optimal but under restrictive assumptions of (strong) convexity (Stich, 2018; Qu et al., 2020). In Haddadpour et al. (2019), the authors provide convergence of FedAvg to the global optimal by imposing the PL condition on the objective function, which is unfortunately not satisfied by several loss functions (e.g., log-logistic loss) over the whole parameter space. Importantly, assuming that the PL inequality is satisfied globally (without any restriction on the parameter space Haddadpour & Mahdavi (2019)) leads to a large Rademacher complexity, thus leading to worse generalization guarantees. This leads to the following question:

> Q1: *Can we develop conditions that are satisfied locally (on a restricted parameter space) rather than globally and provide convergence guarantees for FedAvg? Are there models that satisfy such a condition?*

To address Q1, we provide *new weaker* conditions (a constrained variant of the PL-inequality) on the global and local loss functions. Importantly, we prove that there exists a globally optimal point within a ball of radius ρ around initialization to which FedAvg converges linearly. Moreover, we also establish that there exist NN architectures that satisfy the conditions proposed in our work.

Generalization guarantees for FedAvg: The generalization performance of centralized machine learning algorithms has been extensively studied (Mohri et al., 2018; Bousquet & Elisseeff, 2002; Emami et al., 2020). However, the study of generalization guarantees of FL algorithms is rather limited (Mohri et al., 2019; Hu et al., 2022; Yuan et al., 2021a). Notably, these studies often overlook the impact of the optimization algorithm Sun et al. (2023), and often rely on assumptions like Binary loss Hu et al. (2022); Mohri et al. (2019) and the Bernstein condition (Yuan et al., 2021a). Additionally, generalization bounds for meta-learning and FL are established in Fallah et al. (2021); Chen et al. (2021) under stringent assumptions such as strong convexity and bounded loss functions. Recently, Sun et al. (2023) has investigated the generalization of FedAvg via the lens of uniform stability. We note that these analyses impose strong assumptions such as bounded gradient and heterogeneity on the data, which are usually not satisfied by many problems of practical interest. Moreover, the optimization guarantees provided in Sun et al. (2023) are weaker compared to the linear convergence established in our work. Based on the above observations, we ask the following main question:

> Q2: *Can we provide generalization guarantees for FedAvg? If so, what is the impact of* (a) *the number of samples per client,* (b) *the model size, and* (c) *the number of clients on the generalization performance?*

We address Q2 by deriving Rademacher complexity when each client employs a single hidden-layer NN for FedAvg implementation. We show that the local assumptions developed to address Q1 play an important role in bounding the Rademacher complexity for FedAvg. Importantly, our analysis captures the effect of data samples and NN size, and the number of clients on the generalization performance of FedAvg. It is worth mentioning that to address both Q1 and Q2, we *do not* make some standard assumptions that are typically used in many existing works Li et al. (2019); Stich (2018); Yu et al. (2019); Haddadpour et al. (2019); Qu et al. (2020); Woodworth et al. (2020a;b); Hu et al. (2022); Mohri et al. (2019) such as: (i) (strongly) convex loss, (ii) bounded loss, (iii) bounded gradients (iv) bounded heterogeneity, and (v) interpolation [1]. In this work, we have not assumed the existence of a global optimal point; rather, it is part of our conclusion.

[1] Interpolation refers to the existence of a $\boldsymbol{w}^*$ such that $\Phi_{k,i}(\boldsymbol{w}^*) = 0$ for all $k \in [K]$ and $i \in [n]$.

Contributions. The major contributions of our work include:
➢ Answer to Q1: For the first time, we show that FedAvg converges linearly to the optimal solution (see Corollary 3.2) if the local loss functions at each client and the global loss function satisfy a novel local PL-type assumption introduced in Assumption 2.4. It is important to note that the existence of a global optimal in our analysis is a part of our conclusion, *not* an assumption. To the best of our knowledge, both conditions introduced in Assumption 2.4 are new. It is also worth noting that these conditions do not follow from any of the existing results, even in the special case of centralized setting, i.e., for $K = 1$ (Chatterjee, 2022; An & Lu, 2023). In addition, we also establish that a single hidden-layer NN satisfies the two conditions proposed in Assumption 2.4. Specifically, we establish the conditions on the width of the NN as a function of the number of samples, number of clients, and the feature dimension, and on the eigenvalues of the Jacobian of the loss functions (or the scaling factor of the final output layer) such that the proposed conditions are satisfied. To our knowledge, these results are novel (see Theorems 4.5).
➢ Answer to Q2: To address Q2, we derive an upper bound on the Rademacher complexity for a class of single hidden layer NNs by utilizing the fact that the FedAvg iterates stay within a ρ-ball around the initialization. We point out that this is made possible by the conditions provided in Assumption 2.4. In particular, we show that the Rademacher complexity approaches zero if the radius $\rho = \mathcal{O}(\sqrt{n})$[2] and $m = \mathcal{O}(n^3)$, where n is the number of samples at each client and m is the width of the NN. We show that the generalization error regardless of the data heterogeneity diminishes as $\mathcal{O}(1/\sqrt{n})$. We finally corroborate our theoretical findings through numerical experiments.

2 FedAvg: Algorithm and Assumptions

As discussed in Section 1, FL aims to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \left\{ \Phi(\boldsymbol{w}) \coloneqq \frac{1}{K} \sum_{k=1}^{K} \Phi_k(\boldsymbol{w}) \right\}, \tag{1}$$

where $\Phi_k(\boldsymbol{w}) \coloneqq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_k} l_k(f_{\boldsymbol{w}}(\boldsymbol{x}), y)$ is the loss function at client $k \in [K]$. Here, $y \in \mathcal{Y}$ is the true label, and $f_{\boldsymbol{w}}(\boldsymbol{x})$ is the output of model $\boldsymbol{w} \in \mathbb{R}^{d'}$ for an input feature $\boldsymbol{x} \in \mathbb{R}^d$, and $l_k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is the loss function at the client $k \in [K]$. In the above, d' is the dimension of the parameter space. The following algorithm captures the main steps of FedAvg (McMahan et al., 2017). In Algorithm 1, $\Phi_{k,i}(\boldsymbol{w}_k^{r,t})$ denotes the empirical loss function at client $k \in [N]$ computed using sample $i \in [n]$.

In this and the subsequent section, we answer Q1 posed in Sec. 1. In particular, we provide a general condition for the above algorithm to converge to a global optimum and for the model parameters to stay within a closed ball of radius ρ. In the later sections, we show that this condition is, in fact, satisfied for a single hidden layer NN. Specifically, this constraint imposes a natural regularization of the NN which provides better generalization, as discussed later. To prove our claim, we make the following standard assumptions on the loss function Ji & Telgarsky (2018).

Assumption 2.1 (L- Smoothness)**.** *The loss functions Φ_k and Φ are assumed to be L_k-smooth and L-smooth, respectively, i.e., $\|\nabla\Phi_k(\boldsymbol{u}) - \nabla\Phi_k(\boldsymbol{v})\| \leq L_k\|\boldsymbol{u} - \boldsymbol{v}\|$ for all $k \in [K]$ and $\|\nabla\Phi(\boldsymbol{u}) - \nabla\Phi(\boldsymbol{v})\| \leq L\|\boldsymbol{u} - \boldsymbol{v}\|$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$.*

Assumption 2.2 (Samplewise Smoothness)**.** *The loss functions $\Phi_{k,i}(\boldsymbol{w})$ are assumed to be $l_{k,i}$-sample-wise smooth, i.e., $\|\nabla\Phi_{k,i}(\boldsymbol{v})\|^2 \leq 2l_{k,i}\Phi_{k,i}(\boldsymbol{v})$ for all $k \in [K]$ and $i \in [n]$.*

To define the major assumptions required for the convergence of FedAvg Algorithm 1, we need the following definition (Chatterjee, 2022).

Definition 2.3. *Let $f : \mathbb{R}^d \to \mathbb{R}^+$ be continuously differentiable function on closed ball $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ with center at initialization $\underline{\boldsymbol{w}}^0 \in \mathbb{R}^d$ and radius $\rho > 0$. Define*

$$\alpha(\underline{\boldsymbol{w}}^0, \rho) \coloneqq \inf_{\boldsymbol{w}\in\mathbb{B}[\underline{\boldsymbol{w}}^0,\rho]} \frac{\|\nabla f(\boldsymbol{w})\|^2}{f(\boldsymbol{w})} > 0. \tag{2}$$

Next, we state an important assumption that leads to linear convergence within a ball around initialization.

[2]This is the radius over which our new condition should be satisfied.

Algorithm 1 FedAvg McMahan et al. (2017)

Initialize: $\{\boldsymbol{w}_k^{0,0} = \underline{\boldsymbol{w}}^0\}$, $\boldsymbol{w}_k \in \mathbb{R}^d$ for $k = 1, 2, \ldots, K$
for $r = 0, 1, \ldots, R-1$ **do**
  Broadcast $\underline{\boldsymbol{w}}^r$ to all the clients $k \in [K]$
  for $\tau = 0, 1, \ldots, T-1$ **do**
    for each client $k \in [K]$ **do**
      Sample a batch $\mathcal{B}_k^{r,t}$ of size $|\mathcal{B}_k^{r,t}| = b$
      SGD step on $\boldsymbol{w}_k^{r,t}$ for $k \in [K]$:
      $\boldsymbol{w}_k^{r,t+1} = \boldsymbol{w}_k^{r,t} - \eta \widehat{\nabla \Phi}_k(\boldsymbol{w}_k^{r,t})$
      // $\widehat{\nabla \Phi}_k(\boldsymbol{w}_k^{r,t}) := \frac{1}{b} \sum_{i \in \mathcal{B}^{r,t}} \nabla \Phi_{k,i}\left(\boldsymbol{w}_k^{r,t}\right)$
    end for
  end for
  Receive $\boldsymbol{w}_k^{r,T}$ from nodes $k \in [K]$
  Aggregation step : $\underline{\boldsymbol{w}}^{r+1} = \frac{1}{K} \sum_{k \in [K]} \boldsymbol{w}_k^{r,T}$
end for

Assumption 2.4. *For some initialization $\underline{\boldsymbol{w}}^0$ and radius $\rho > 0$, we make the following assumptions on the local and global loss functions:*

1. *The loss function at each client is assumed to satisfy (see Theorem E.1)*
$$32\Phi_k(\underline{\boldsymbol{w}}^0) < \rho^2 \alpha_k(\underline{\boldsymbol{w}}^0, \rho). \tag{3}$$
Here, $\alpha_k(\underline{\boldsymbol{w}}^0, \rho)$ is as defined in equation 2 but with $f(\cdot)$ replaced by $\Phi_k(\cdot)$.

2. *The global loss function is assumed to satisfy the following condition*
$$\sqrt{128 e l'_{\max} K \Phi(\underline{\boldsymbol{w}}^0)} < (1 - \zeta_\rho)\rho \alpha_g(\underline{\boldsymbol{w}}^0, \rho), \tag{4}$$
for some $\zeta_\rho \in (0, 1)$. Here, $\alpha_g(\underline{\boldsymbol{w}}^0, \rho)$ is as defined in equation 2 but with $f(\cdot)$ replaced by $\Phi(\cdot)$.

Remark 1. *In general, two very critical assumptions are made in the literature while proving linear convergence: (i) interpolation, i.e., there exists $\boldsymbol{w}^*$ such that $\Phi_i(\boldsymbol{w}^*) = 0$ for all samples $i \in [n]$ Liu et al. (2022); Li et al. (2019), and (ii) strongly convex loss Li et al. (2019); Karimireddy et al. (2020) or loss function satisfying the PL-inequality Fan et al. (2023). Later, a relaxed version of PL-inequality called local PL or PL*-inequality was proposed where the PL-inequality needs to be satisfied over a small ball around the initialization (see Liu et al. (2022); Oymak & Soltanolkotabi (2019). Despite this relaxation, it makes a critical assumption on the existence of the optimal $\boldsymbol{w}^*$ such that the loss $\Phi_i(\boldsymbol{w}^*) = 0$ for all samples $i \in [n]$-the interpolation regime. In our work, we argue that this assumption can be relaxed with our novel condition shown in Assumption 2.4. It is important to note that our condition is fundamentally different from the PL*-inequality in the following way:*

- There is a stark difference between our proposed condition and the the PL-condition (or PL^* condition), which is defined as $\|\nabla \Phi(\boldsymbol{w})\|^2 \geq \mu(\Phi(\boldsymbol{w}) - \Phi(\boldsymbol{w}^*))$ for all $\boldsymbol{w} \in \mathbb{R}^d$ (and $\boldsymbol{w} \in \mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ for PL^* condition). In the PL-condition (and local PL), the constants do not depend on the initialization and radius as the condition is universally satisfied. Another important assumption made in the local/global PL-condition is the existence of a global optimal point $\mathbf{w}^*$. In contrast, our proposed condition does not require this assumption; instead, we prove the existence of a global optimal point under our novel condition.
- It is important to note that the PL-condition must be satisfied over the entire parameter space, which can restrict its applicability to certain loss functions such as logistic loss Karimi et al. (2016). On the other hand, our novel condition is assumed only over a small neighborhood around the initialization, making it more broadly applicable. Later we show that parameters such as initialization and the radius ρ can be chosen so that the condition is easily (compared to the PL inequality) satisfied.

In this work, we have shown that the proposed condition is satisfied for at least a single hidden layer neural network. In Chatterjee (2022), the authors have shown that the wide neural network satisfies

the constrained PL inequality for a single client setting. Therefore, we strongly believe that the proposed condition in our work will also be satisfied for wide neural networks.

3 Convergence Analysis

In this section, we establish that the FedAvg Algorithm 1 achieves linear convergence to a global optimum under the set of assumptions introduced in Sec. 2. Importantly, note that the existence of this global optimum is established as a conclusion rather than an assumption. Moreover, unlike other works, we do not explicitly assume interpolation to establish linear convergence of FedAvg (Haddadpour et al., 2019; Stich, 2018). In particular, we establish a proof that the sufficient conditions stated in equation 2.4 not only guarantee the linear convergence of Algorithm 1 but also ensure the existence of an optimal point denoted as $\boldsymbol{w}^*$ within the closed ball $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$. The following theorem is a precise statement whose proof can be found in Appendix 3.1.

Theorem 3.1. *Assuming that there exists an initialization $\underline{\boldsymbol{w}}^0 \in \mathbb{R}^d$, and a radius $\rho > 0$ such that Assumptions 2.1 and 2.4 are satisfied by loss functions Φ and Φ_k for $k \in [K]$, then FedAvg ensures that there exists a $\boldsymbol{w}^* \in \mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ such that $\lim_{R\to\infty} \Phi(\underline{\boldsymbol{w}}^R) = \Phi(\boldsymbol{w}^*) = 0$ provided the learning rate*

$$\eta \leq \min\left\{\frac{2}{\alpha_{\min}}, \frac{\alpha_{\min}}{4L_{\max}l'_{\max}}, \frac{\alpha_{\min}}{2L_{\max}l'_{\max}}, \frac{1}{T\sqrt{\Psi_0}}, \frac{8}{\alpha_g T}, \frac{\zeta_\rho \rho}{T\sqrt{\Psi_0}}, \Psi_1, \Psi_2\right\},$$

where $l'_{max} := \max_k l'_k := \max_i l_{k,i}$; $L_{\max} := \max_k L_k$; $\alpha_{min} := \min_{k\in[K]} \alpha_k$; $\Psi_0 := 2el'_{\max}K\Phi(\underline{\boldsymbol{w}}^0)$; $\Psi_1 := \sqrt{\frac{3}{L_{\max}l'_{max}}}$ and $\Psi_2 := \min\left\{\frac{\alpha_g\alpha_{\min}}{4T(4L^2_{\max}l'_{\max}+Ll'_{\max}\alpha_{\min})}, \frac{1}{3L_{\max}T}\right\}$. More precisely, after $R > 0$ communication rounds, the FedAvg Algorithm 1 satisfies

$$\Phi(\underline{\boldsymbol{w}}^R) \leq \left(1 - \frac{\eta T \alpha_g(\underline{\boldsymbol{w}}^0, \rho)}{4}\right)^R \Phi(\underline{\boldsymbol{w}}^0). \tag{5}$$

Essence of the Proof of Theorem 3.1: Assumptions 2.1 and 2.4 lead to an exponential relation, specifically $\Phi(\underline{\boldsymbol{w}}^{r+1}) \leq \gamma^r \Phi(\underline{\boldsymbol{w}}^0)$, where $\gamma \in (0, 1)$, (refer to Lemma F.4). To prove the existence of global optima $\boldsymbol{w}^*$ within the ball $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$, we have used the method of induction on two variables: global communication round r and local updates t. By doing so, we conclude that the sequence $\{\boldsymbol{w}_k^{r,\tau}\}_{r,\tau\geq 0}$ remains confined within the ball $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ (refer to Lemma F.6), which ensures that the sequence $\{\underline{\boldsymbol{w}}^r\}_{r=1}^\infty$ remains within the ball $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ for all r. Further, we have shown that the sequence $\{\underline{\boldsymbol{w}}^r\}_{r=1}^\infty$ is Cauchy sequence in the closed subset $\mathbb{B}[\underline{\boldsymbol{w}}^0, \rho]$ of complete space. Therefore, it guarantees the limit of the sequence $\{\underline{\boldsymbol{w}}^r\}_{r=1}^\infty$, denoted by $\boldsymbol{w}^*$ belongs to the ball. A complete proof is provided in Appendix F. □

Note that Chatterjee (2022) required one condition to be satisfied for the linear convergence since their work considered a centralized setting. In contrast, our work requires two conditions for both global and local loss functions as stated in Assumptions 2.4 to guarantee linear convergence of FedAvg. Later we show that as the number of clients, K, increases, the requirement becomes more stringent. The above theorem leads to the following corollary.

Corollary 3.2. *By choosing η as in Theorem 3.1, for any error $\epsilon > 0$, Algorithm 1 achieves a loss of $\Phi(\underline{\boldsymbol{w}}^R) < \epsilon$ after $R \geq \mathcal{O}\left(\left\lceil 2\log\left(\frac{\Phi(\underline{\boldsymbol{w}}^0)}{\epsilon}\right)\right\rceil\right)$ communication rounds.*

Our next goal is to show that it is possible to initialize a NN such that it satisfies the conditions provided in Assumption 2.4. However, note that this does not provide any guarantees on the generalization error. To fill this gap, in the following sections, we consider a single hidden-layer NN and show that (a) there exist an initialization and radius ρ such that it results in a linear convergence leading to zero training loss (i.e., assumptions stated in Sec. 2 are satisfied), and (b) prove that the generalization error can be made small by choosing large enough training samples and performing FedAvg for a sufficiently large number of communication rounds.

4 ASSUMPTION 2.4 FOR SINGLE HIDDEN LAYER NN WITH SQUARED ERROR LOSS

In this section, we show that there exist NNs such that Assumption 2.4 is satisfied, and hence leads to linear convergence of FedAvg (see Theorem 3.1). Towards this, we consider the following NN with a single hidden layer. In particular, we assume that the first layer has m neurons followed by a smooth activation function. The output of this NN is given by Arora et al. (2019)

$$f_{\boldsymbol{w}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}), \tag{6}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the input feature vector. With a slight abuse of notation, we have used $\boldsymbol{w} = \mathtt{vec}([\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]) \in \mathbb{R}^{dm \times 1}$ to denote the aggregated weight vectors in the first layer and $\boldsymbol{v} = (v_1, v_2, \ldots, v_m)^\top$ to denote the weight in the second layer, where $v_j \overset{\text{i.i.d.}}{\sim} \{-1, 1\}$. Now, we make the following assumption on the activation function.

Assumption 4.1. *We assume that $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth non-decreasing activation function such that $\sigma(0) = 0$. Further, first and second order derivatives of σ are bounded i.e., $|\sigma'(x)| \leq D_\sigma$ and $|\sigma''(x)| \leq \Delta_\sigma$.*

Note that the above condition is satisfied by the $\tanh$ activation function, i.e., $\sigma(x) = \tanh(x)$. The condition $\sigma(0) = 0$ is assumed for the sake of simplicity and ease of notation. It turns out that, with random initialization, this can be relaxed without changing the main result of the paper. With $\sigma(x) \neq 0$, many activation functions such as Softmax, $\tanh$ to name a few (see Xu et al. (2015)) satisfy the conditions mentioned in Assumption 4.1. It is worth noting that the well-known ReLU activation does not satisfy the smoothness condition, but it can be well approximated by a smooth proxy function (see (Xu et al., 2015)).

Assumption 4.2. *Each node $k \in [K]$ samples n i.i.d. data points denoted $\mathcal{X}_k = \{(\boldsymbol{x}_{k,1}, y_{k,1}), \ldots, (\boldsymbol{x}_{k,n}, y_{k,n})\}$ from a continuous and possibly different distributions $p_k(\boldsymbol{x}), k \in [K]$ with $y_{k,i} \leq y_{max}$ for all $i \in [n]$.*

We consider the average loss function $\Phi(\boldsymbol{w}) \coloneqq \frac{1}{K} \sum_{k=1}^{K} \Phi_k(\boldsymbol{w})$, where $\Phi_k : \mathbb{R}^{md} \to \mathbb{R}$ is the squared loss function for each client $k \in [K]$ and is defined as $\Phi_k(\boldsymbol{w}) = \sum_{i=1}^{n} \left[f_{\boldsymbol{w}}(\boldsymbol{x}_{k,i}) - y_{k,i}\right]^2 = \|\boldsymbol{e}_k\|_2^2$, where the i^{th} entry of the error vector $\boldsymbol{e}_k \coloneqq [f_{\boldsymbol{w}}(\boldsymbol{x}_{k,i}) - y_{k,i}]$. Using $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$, the global loss can be written as $\Phi(\boldsymbol{w}) \coloneqq \frac{1}{K} \|\boldsymbol{e}\|^2$. Next, we discuss the conditions under which a single hidden layer neural network satisfies Assumption 2.4. It turns out that these conditions are dependent on the following Jacobian matrix:

$$\boldsymbol{J}_k(\boldsymbol{w}) = \frac{1}{\sqrt{m}} \times \boldsymbol{H}_k(\boldsymbol{w}), \tag{7}$$

where each entry of $\boldsymbol{J}_k(\boldsymbol{w})$ is a d-dimensional row vector, and $\boldsymbol{H}_k(\boldsymbol{w})$ is defined as follows

$$\boldsymbol{H}_k(\boldsymbol{w}) \coloneqq \begin{bmatrix} v_1 \sigma'(\boldsymbol{w}_1^\top \boldsymbol{x}_{k,1}) \boldsymbol{x}_{k,1}^\top & v_2 \sigma'(\boldsymbol{w}_2^\top \boldsymbol{x}_{k,1}) \boldsymbol{x}_{k,1}^\top & \ldots & v_m \sigma'(\boldsymbol{w}_m^\top \boldsymbol{x}_{k,1}) \boldsymbol{x}_{k,1}^\top \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_1 \sigma'(\boldsymbol{w}_1^\top \boldsymbol{x}_{k,n}) \boldsymbol{x}_{k,n}^\top & v_2 \sigma'(\boldsymbol{w}_2^\top \boldsymbol{x}_{k,n}) \boldsymbol{x}_{k,n}^\top & \ldots & v_m \sigma'(\boldsymbol{w}_m^\top \boldsymbol{x}_{k,n}) \boldsymbol{x}_{k,n}^\top \end{bmatrix}, \tag{8}$$

where $k \in [K]$ and the size of the matrix $\boldsymbol{H}_k(\boldsymbol{w})$ is $n \times md$, i.e., $\boldsymbol{H}_k(\boldsymbol{w}) \in \mathbb{R}^{n \times md}$. We define a global Jacobian matrix $\boldsymbol{J}(\boldsymbol{w})$ by stacking $\boldsymbol{H}_k^\top(\boldsymbol{w})$ row-wise as $\boldsymbol{J}(\boldsymbol{w}) = \frac{1}{\sqrt{m}} \times \left[\boldsymbol{H}_1^\top(\boldsymbol{w}), \boldsymbol{H}_2^\top(\boldsymbol{w}), \ldots, \boldsymbol{H}_K^\top(\boldsymbol{w})\right] \in \mathbb{R}^{md \times Kn}$. The following lemma provides a condition under which $\boldsymbol{J}_k(\underline{\boldsymbol{w}}^0)$ and $\boldsymbol{J}(\underline{\boldsymbol{w}}^0)^\top$ are full rank matrices. Note that the full rank requirement is only at the initialization. The size of the NN scales as n/d as opposed to n in (Chatterjee, 2022). This result is similar to the results of Zhang et al. (2021) but for an FL setting.

Algorithm 2 FedAvg Algorithm for single hidden layer NN

1: **Initialization**: Initialize using $\underline{\boldsymbol{w}}^0 \sim \mathcal{N}(\mathbf{0}, \frac{1}{d} I_{md\times md})$ and $v_i \overset{\text{i.i.d.}}{\sim} \{-1, 1\}\ \forall i \in [m]$.
2: Broadcast $\underline{\boldsymbol{w}}^r$ to all the clients $k \in [K]$
3: Run the FedAvg Algorithm 1

Lemma 4.3. *At the random initialization $\underline{\boldsymbol{w}}^0 \sim \mathcal{N}(\mathbf{0}, \frac{1}{d} I_{md\times md})$, and $v_i \overset{\text{i.i.d.}}{\sim} \{-1, 1\}$ for all $i \in [m]$, the matrices $\boldsymbol{J}_k(\underline{\boldsymbol{w}}^0)$ and $\boldsymbol{J}(\underline{\boldsymbol{w}}^0)^\top$ have full column ranks* almost surely *provided $m \geq n/d$ and $m \geq nK/d$, respectively.*

Proof: The result follows by following the proof of Lemma $E.1$ of Zhang et al. (2021) for the matrices $\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)$ and $\boldsymbol{H}(\underline{\boldsymbol{w}}^0)^\top$. One main difference is that Zhang et al. (2021) uses mirrored Le-cun. However, the proof does not change for our initialization. □

Towards stating the condition for neural network, we need the following definitions

$$\lambda^-_{k,\rho}(m) \coloneqq \inf_{\boldsymbol{w}\in\mathbb{B}[\underline{\boldsymbol{w}}^0,\rho]} \frac{\boldsymbol{e}_k^\top \boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)^\top \boldsymbol{e}_k}{\|\boldsymbol{e}_k\|^2}, \tag{9}$$

where $\boldsymbol{e}_k$ and $\boldsymbol{H}_k(\boldsymbol{w})$ are as defined earlier.[3] The following is an extension of the above definition to K clients

$$\lambda^-_{\rho}(m) \coloneqq \inf_{\boldsymbol{w}\in\mathbb{B}[\underline{\boldsymbol{w}}^0,\rho]} \frac{\boldsymbol{e}^\top \boldsymbol{H}(\underline{\boldsymbol{w}}^0)^\top \boldsymbol{H}(\underline{\boldsymbol{w}}^0)\boldsymbol{e}}{\|\boldsymbol{e}\|^2} \tag{10}$$

where $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k]^\top \in \mathbb{R}^{nK}$ and $\boldsymbol{H}(\underline{\boldsymbol{w}}^0)$ is defined earlier. Similarly, $\tilde{\lambda}^-_{k,\rho}(m)$ and $\tilde{\lambda}^-_{\rho}(m)$ are defined by replacing $\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)$ by $\boldsymbol{H}_k(\boldsymbol{w})$ and $\boldsymbol{H}(\underline{\boldsymbol{w}}^0)$ by $\boldsymbol{H}(\boldsymbol{w})$ in equations 9 and equation 10, respectively. In addition, $\lambda_{\max}(\rho) \coloneqq \sup_{\boldsymbol{w}\in\mathbb{B}(\underline{\boldsymbol{w}}_0,\rho)} \lambda_{\max}\left(\boldsymbol{H}(\boldsymbol{w})\boldsymbol{H}(\boldsymbol{w})^\top\right)$. These notations will be used in Theorem 4.5. Since we know from the above Lemma that the matrices $\boldsymbol{H}(\underline{\boldsymbol{w}}^0)\boldsymbol{H}(\underline{\boldsymbol{w}}^0)^\top$ and $\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)^\top \boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)$, $k \in [K]$ are full rank, we next ask if the above terms scale with m. Recall that we are looking at the Jacobian to state the condition under which Assumption 2.4 is satisfied. Thus, the following assumption is important, whose analytical justification is provided in App. G.

Assumption 4.4. *We assume that both $\lambda^-_{k,\rho}(m)$ and $\lambda^-_{\rho}(m)$ scale linearly with m.*

Experimental Justification of Assumption 4.4: An observation similar to the above assumption was also made in (Telgarsky, 2021, page 39). We verify the above assumption via experiments in Fig. 1, where we have plotted the minimum eigenvalue of the Jacobian versus m for different numbers of clients K using the MNIST data set (LeCun & Cortes, 2010). We can observe from the figure that the variation is almost linear, and the slope increases with decreasing K.

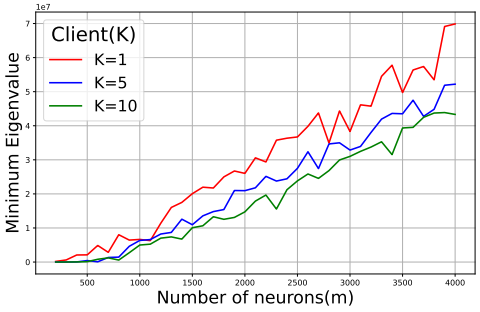


Figure 1: Plot of $\lambda_{\min}(m)$ versus m for $K = 1, 5, 10$. Here, $K = 1$ corresponds to $\lambda_{1,\min}(m)$. This shows that Assumption 4.4 is valid in the real-world setting as well, i.e., the minimum eigenvalue scales linearly with m.

4.1 Condition on Neural Network (NN)

To prove the linear convergence of Algorithm 1 for single hidden layer NN, we need the definitions stated in equations 10 and 9. The following theorem provides a condition under which the Algorithm 1 converges linearly to a global optimal point, and the proof can be found in the Appendix I.

[3]Here, $\boldsymbol{e}_k$ and $\boldsymbol{e}$ depend on $\boldsymbol{w}$.

Theorem 4.5. *Let* $\Psi_{m,K,n,\rho} \coloneqq \sqrt{bn\left(\frac{\lambda_\rho^+(m)}{m} + \frac{d\Delta_\sigma^2\rho^2}{m}\right)}$ *and* $b \coloneqq \frac{2D_\sigma^2\rho^2 d\log(2n/\delta)}{m} + 2y_{\max}^2$, *where* $\lambda_\rho^+(m) \coloneqq \sup_{\boldsymbol{w}\in\mathbb{B}[\underline{\boldsymbol{w}}^0,\rho]} \frac{\|\boldsymbol{H}(\underline{\boldsymbol{w}}^0)\boldsymbol{e}\|^2}{\|\boldsymbol{e}\|^2}$. *The loss functions for single hidden layer NN satisfy equation 3 and equation 4 of Assumption 2.4 with a probability of at least* $1-\delta/2$, *for any* $\delta > 0$ *provided the following holds:*

$$\frac{\lambda_{k,\rho}^-(m)}{m} > 2\times\left[\frac{\Delta_\sigma^2 d\rho^2}{m} + \frac{8bn}{\rho^2}\right], \text{ and } \frac{\lambda_\rho^-(m)}{m} > \frac{8K\Psi_{m,K,n,\rho}}{(1-\zeta_\rho)\rho} + \frac{2d\Delta_\sigma^2\rho}{m}, \quad (11)$$

where $\lambda_{k,\rho}^-(m)$ *and* $\lambda_\rho^-(m)$ *are as defined in equation 9 and equation 10, respectively.*

To the best of our knowledge, these conditions are the first of their kind. First, note that the terms $\lambda_{k,\rho}^-(m)/m$ and $\lambda_\rho^-(m)/m$ are less sensitive to ρ since they are sandwiched between the smallest and the largest eigenvalues of $\boldsymbol{H}(\underline{\boldsymbol{w}}^0)^\top\boldsymbol{H}(\underline{\boldsymbol{w}}^0)$ and $\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)^\top\boldsymbol{H}_k(\underline{\boldsymbol{w}}^0)$, respectively. In particular, these eigenvalues depend on the initialization $\underline{\boldsymbol{w}}^0$ while the original condition is in terms of the ball around the initialization. Hence, using the eigenvalues in place of $\lambda_{k,\rho}^-(m)$ and $\lambda_\rho^-(m)$ in the new conditions makes it easy to verify (see Fig. 1). Secondly, the larger values of ρ make the right-hand sides in the equation 11 large, and hence the conditions may not be satisfied, as expected. On the other hand, the same can be observed for smaller values of ρ as well. Thus, a critical ρ is necessary. By choosing $\rho = c\times\mathcal{O}(\sqrt{n})$ and $m = \mathcal{O}(n^3)$ in Theorem 4.5 ensures that the right hand sides scale down with c. Thus, the right-hand side is small for a large enough c. However, by Assumption 4.4, the left-hand sides, i.e., $\lambda_{k,\rho}^-(m)/m$ and $\lambda_\rho^-(m)/m$ are constants that depend only on the initialization (not on ρ), and do not scale with m or n or c. Hence, the conditions are satisfied for large enough c:

Corollary 4.6. *Choosing* $\rho = c\times\mathcal{O}(\sqrt{n})$ *and* $m = \mathcal{O}(n^3)$ *in Theorem 4.5 ensure that the conditions in equation 11 are satisfied for sufficiently large* c.

The above corollary shows that by choosing a large radius of ρ and a large number of nodes in the second layer, linear convergence can be guaranteed. This brings in several challenges while proving the generalization guarantee, especially while proving a bound on the Rademacher complexity.

5 Generalization Performance: Single Hidden Layer NN

In this section, we show that single hidden layer NN architectures exhibit impressive generalization guarantees. To state the generalization result, we need the following notion of Rademacher complexity of the single hidden layer NN.

Definition 5.1 (See Mohri et al. (2019)). *The Rademacher complexity of a class of single hidden layer NN constrained to a ball of radius* ρ *at client* $k \in [K]$ *is defined as*

$$\mathtt{Rad}_k(\underline{\boldsymbol{w}}^0,\rho) \coloneqq \mathbb{E}_{|\boldsymbol{v}\in\mathcal{G}_{\boldsymbol{v}}}\left[\sup_{\boldsymbol{w}\in\mathbb{B}[\underline{\boldsymbol{w}}^0,\rho]}\frac{1}{n}\sum_{i=1}^n \zeta_i f_{\boldsymbol{w};\boldsymbol{v}}(\boldsymbol{x}_{k,i})\right],$$

where the expectation is with respect to $\boldsymbol{\zeta} \coloneqq (\zeta_1,\zeta_2,\ldots,\zeta_n) \overset{i.i.d.}{\sim} \{-1,+1\}^n$, *conditioned on* $\boldsymbol{v} \coloneqq (v_1,v_2,\ldots,v_m) \in \mathcal{G}_{\boldsymbol{v}} \coloneqq \{\boldsymbol{v}\in\{-1,1\}^m : |\sum_{i=1}^n \zeta_i f_{\boldsymbol{w};\boldsymbol{v}}(\boldsymbol{x})| < \Delta\}$. *Here,* $\Delta \coloneqq \sqrt{2}D_\sigma d\sqrt{\frac{\rho^2+m}{m}\log 4}$ *and* $\boldsymbol{x}$ *is any data point sampled from* $p_k(\boldsymbol{x})$.

For a FL setting, the generalization guarantee is provided in Mohri et al. (2019), and the result requires the loss to be bounded. However, in our case, the loss can potentially be unbounded. We handle this by focusing on the class of "good" NNs, i.e., $\boldsymbol{v} \in \mathcal{G}_{\boldsymbol{v}}$, whose output is bounded. In Appendix H, using the fact that the weight vector lies within a ball of radius ρ around $\underline{\boldsymbol{w}}^0$, we show that there exists such NNs with bounded output. Subsequently, we show that for such NNs, the generalization is guaranteed. We use this result along with the result of Mohri et al. (2019) to show the following Theorem whose proof can be found in Appendix J.

Theorem 5.2. *Let* $\Psi \coloneqq \left((\rho^2+3m)\frac{2D_\sigma^2 d^2 \log 4}{m} + y_{max}^2\right)\sqrt{2\log(\frac{1}{\delta})}$. *For the single hidden layer NN with the initialization as in Algorithm 2 satisfying Assumptions 4.4 with* $m \geq nK/d$, *and the conditions of Theorem 4.5, with a probability of at least* $1-\delta$, *the following inequality holds*

$$\Phi(\boldsymbol{w};\boldsymbol{v}) \leq \Phi_S(\boldsymbol{w};\boldsymbol{v}) + \frac{2n}{K}\sum_{k=1}^{K} \mathtt{Rad}_k(\underline{\boldsymbol{w}}^0,\rho) + \Psi\sqrt{\frac{n}{K}}. \tag{12}$$

Recall that the loss function is defined as the sum of the loss on individual training samples. Thus, defining $\mathcal{L}(\boldsymbol{w};\boldsymbol{v}) \coloneqq \frac{\Phi(\boldsymbol{w},\boldsymbol{v})}{n}$ and $\mathcal{L}_S(\boldsymbol{w};\boldsymbol{v}) \coloneqq \frac{\Phi_S(\boldsymbol{w};\boldsymbol{v})}{n}$, and using this in the above theorem leads to the following.

Corollary 5.3. *For the single hidden layer NN with initialization as in Algorithm 2, with probability at least* $1-2\delta$ *over the draw of the samples* $X_k \sim \mathcal{D}_k^n$, *the following inequality holds*

$$\mathcal{L}(\boldsymbol{w};\boldsymbol{v}) \leq \mathcal{L}_S(\boldsymbol{w};\boldsymbol{v}) + \frac{2}{K}\sum_{k=1}^{K} \mathtt{Rad}_k(\underline{\boldsymbol{w}}^0,\rho) + \frac{\Psi}{\sqrt{nK}}. \tag{13}$$

Next, we provide an upper bound on the Rademacher complexity.

Theorem 5.4. *The Rademacher complexity of client* $k \in [K]$ *is bounded by*

$$\mathtt{Rad}_k(\underline{\boldsymbol{w}}^0,\rho) \leq \frac{1}{n\sqrt{m}} + \sqrt{\frac{\nu D_\sigma^2 d^2(\log 4)\log(N_{\theta,\rho}/\delta_1)}{n}},$$

where $\nu = (\rho^2+3m)/m$, $N_{\theta,\rho} \coloneqq 3d^{3/4}\sqrt{\rho D_\sigma nm}$ *and* $\delta_1 \coloneqq \frac{1}{2mn\sqrt{2}D_\sigma d}\sqrt{\frac{m}{\log 4(\rho^2+m)}}$.

Proof: See Appendix K. □

5.1 Discussion

To the best of our knowledge, the above is the first result of its kind for an FL setup. We make the following remarks.

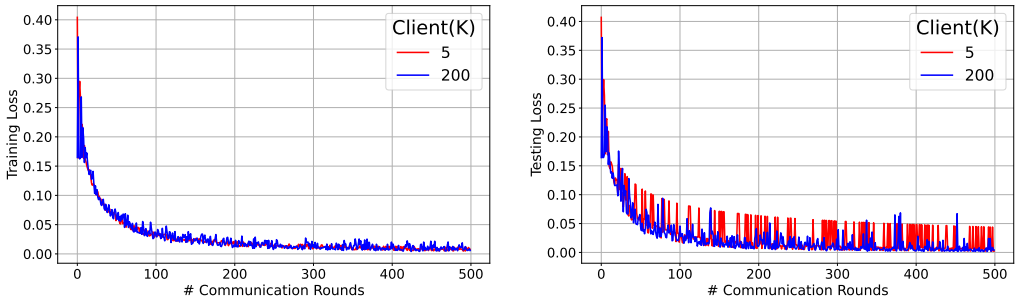
- The generalization error can be made small provided the right-hand side in the Corollary 5.3 is small. The first term, i.e., the empirical loss, depends on the communication rounds and the conditions stated in Theorem 4.5. The latter can be ensured by choosing $\rho = \mathcal{O}(\sqrt{n})$ and $m = \mathcal{O}(n^3)$, as shown in Corollary 4.6. In other words, the radius and the size of the NN scale with n which is not desired in general. However, we believe that this cannot be eliminated unless we make some structural assumptions about the data.
- Note that δ_1 and $N_{\theta,\rho}$ scale with n and m. However, it appears as a logarithmic term, and hence, the Rademacher complexity does not grow linearly with n. The above choices of ρ and m ensure that the Rademacher complexity in Theorem 5.4 goes down as $\mathcal{O}(1/\sqrt{n})$. Also, the choice of ρ cannot scale faster than $\sqrt{m}$.
- The last term in the generalization result scales down with n as $1/\sqrt{n}$. Based on these observations, it is clear that the generalization error can be made small by choosing large enough communication rounds R and the number of training samples n.
- Here, we present our theoretical insights on the effect of K. From the generalization bound in equation 5.3, it is evident that the last term decreases with K as $1/\sqrt{K}$. However, for larger values of K, the learning rate is impacted by K through $\frac{\zeta_p \rho}{T\sqrt{\Psi_0}}$, which scales as $1/\sqrt{K}$ (see Theorem 3.1). From equation 5, the loss goes down as $\exp\{-\mathcal{O}(R/\sqrt{K})\}$ leading to slower convergence. Thus, the overall effect of increasing K on the generalization is insignificant; this is also demonstrated in our experimental results as well as several existing works.

The above argument shows that the average loss can be made small by choosing sufficiently large m, n, and communication rounds, as shown next.[4]

Corollary 5.5. *With a probability of at least $1-\delta$, there exists a single hidden layer NN employing the FedAvg algorithm with sufficiently large m, n, and R that achieves a small generalization error. More specifically, the generalization error goes down as $\mathcal{O}(1/\sqrt{n})$.*

6 Experimental Results

In this section, we verify our theoretical findings with experiments performed on an NVIDIA DGX V100 machine. We have used an MNIST image data set LeCun & Cortes (2010) distributed across 5 and 200 clients. We have used the single hidden layer network model with 1000 neurons in the hidden layer and $\tanh$ activation function. In both cases, we have maintained around 50 data points at each client, which is less than the dimension of input feature vectors, i.e., around 1200, which satisfies the condition $d \geq n$ and $m \geq nK/d$. We execute FedAvg for $R = 500$ communication rounds along with $T = 5$ round of local updates at each client with i.i.d. data.



(a) (b)

Figure 2: The Figures in (a) and (b) show the effect of the number of clients K on the training and the testing losses, respectively. The experiments are done using MNIST data set.

Figure 2 shows the effect of K on the testing and training errors. As suggested by our theory (see Sec. 5.1), increasing or decreasing K has no effect on the performance (generalization and training loss).

7 Conclusions

In this work, we addressed the problem of generalization along with convergence guarantees of the widely used FedAvg algorithm for solving Federated Learning (FL) problems. We proved the generalization bound by handling the optimization error and the Rademacher complexity. The optimization error was handled by proposing a novel and new constrained Polyak-Łojasiewicz (PL) type conditions on the (local) loss functions. Under these new conditions, we showed that there exists a global optimum to which the FedAvg converges linearly after $\mathcal{O}(\log(1/\epsilon))$ rounds of communication, where ϵ is the desired optimality gap. Importantly, we demonstrated that a class of single hidden layer NNs satisfy the proposed conditions that are required to establish the linear convergence of FedAvg as long as $m > \frac{nK}{d}$, where m is the number of neurons in the hidden layer, n is the number of samples at each client, K is the number of clients, and d is the feature dimension. Finally, we showed that the generalization error of FedAvg decreases at the rate of $\mathcal{O}(1/\sqrt{n})$ by proving a bound on the Rademacher Complexity using the fact that the neural network parameters are constrained to a neighbourhood around the initialization.

[4]While stating this result, we have ignored $\log$ factors.

REFERENCES

Jing An and Jianfeng Lu. Convergence of stochastic gradient descent under a local lajasiewicz condition for deep neural networks. *arXiv preprint arXiv:2304.09221*, 2023.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Idan Attias, Aryeh Kontorovich, and Y. Mansour. Improved generalization bounds for adversarially robust learning. *J. Mach. Learn. Res.*, 23:175:1–175:31, 2018. URL https://api.semanticscholar.org/CorpusID:250244124.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pp. 2938–2948. PMLR, 2020.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. B. McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *ArXiv*, abs/1611.04482, 2016. URL https://api.semanticscholar.org/CorpusID:10933707.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Sourav Chatterjee. Convergence of gradient descent for deep neural networks. *arXiv preprint arXiv:2203.16462*, 2022.

Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J Su. A theorem of the alternative for personalized federated learning. *arXiv preprint arXiv:2103.01901*, 2021.

Melikasadat Emami, Mojtaba Sahraee-Ardakan, Parthe Pandit, Sundeep Rangan, and Alyson Fletcher. Generalization error of generalized linear models in high dimensions. In *International Conference on Machine Learning*, pp. 2892–2901. PMLR, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *Advances in Neural Information Processing Systems*, 34:5469–5480, 2021.

Chen Fan, Christos Thrampoulidis, and Mark Schmidt. Fast convergence of random reshuffling under over-parameterization and the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 301–315. Springer, 2023.

Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *ArXiv*, abs/1712.07557, 2017. URL https://api.semanticscholar.org/CorpusID:3630366.

Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *Advances in Neural Information Processing Systems*, 32, 2019.

Xiaolin Hu, Shaojie Li, and Yong Liu. Generalization bounds for federated learning: Fast rates, unparticipating clients and unbounded losses. In *The Eleventh International Conference on Learning Representations*, 2022.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pp. 795–811. Springer, 2016.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019.

Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Xiaoxiao Li, Zhao Song, Runzhou Tao, and Guangyi Zhang. A convergence theory for federated average: Beyond smoothness. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 1292–1297. IEEE, 2022.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. 2018.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pp. 4951–4960. PMLR, 2019.

Zhaonan Qu, Kaixiang Lin, Jayant Kalagnanam, Zhaojian Li, Jiayu Zhou, and Zhengyuan Zhou. Federated learning's blessing: Fedavg has linear speedup. *arXiv preprint arXiv:2007.05690*, 2020.

Zhaonan Qu, Kaixiang Lin, Zhaojian Li, Jiayu Zhou, and Zhengyuan Zhou. Federated learning's blessing: Fedavg has linear speedup, 2021. URL https://openreview.net/forum?id=yJHpncwG1B.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(90):2635–2670, 2010. URL http://jmlr.org/papers/v11/shalev-shwartz10a.html.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

Bingqing Song, Prashant Khanduri, Xinwei Zhang, Jinfeng Yi, and Mingyi Hong. Fedavg converges to zero training loss linearly for overparameterized multi-layer neural networks. In *International Conference on Machine Learning*, pp. 32304–32330. PMLR, 2023.

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

Zhenyu Sun, Xiaochun Niu, and Ermin Wei. Understanding generalization of federated learning via stability: Heterogeneity matters. *ArXiv*, abs/2306.03824, 2023. URL https://api.semanticscholar.org/CorpusID:259088815.

Matus Telgarsky. Deep learning theory lecture notes. https://mjt.cs.illinois.edu/dlt/, 2021. Version: 2021-10-27 v0.0-e7150f2d (alpha).

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *The Journal of Machine Learning Research*, 22(1):9709–9758, 2021.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020a.

Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020b.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:53092122.

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

Honglin Yuan, Warren Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? *arXiv preprint arXiv:2110.14216*, 2021a.

Honglin Yuan, Warren R. Morningstar, Lin Ning, and K. Singhal. What do we mean by generalization in federated learning? *ArXiv*, abs/2110.14216, 2021b. URL https://api.semanticscholar.org/CorpusID:239998253.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ArXiv*, abs/1611.03530, 2016. URL https://api.semanticscholar.org/CorpusID:6212000.

Jiawei Zhang, Yushun Zhang, Mingyi Hong, Ruoyu Sun, and Zhi-Quan Luo. When expressivity meets trainability: Fewer than n neurons can work. *Advances in Neural Information Processing Systems*, 34:9167–9180, 2021.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *ArXiv*, abs/1708.01012, 2017a. URL https://api.semanticscholar.org/CorpusID:3384938.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017b.