FedNS: A Fast Sketching Newton-type Algorithm for Federated Learning

Jian Li[1], **Yong Liu**[2*], **Wei Wang**[3], **Weiping Wang**[1]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]China Unicom Research Institute
lijian9026@iie.ac.cn, liuyonggsai@ruc.edu.cn, wangweiping@iie.ac.cn

Abstract

Recent Newton-type federated learning algorithms have demonstrated linear convergence with respect to the communication rounds. However, communicating Hessian matrices is often unfeasible due to their quadratic communication complexity. In this paper, we introduce a novel approach to tackle this issue while still achieving fast convergence rates. Our proposed method, named as Federated Newton Sketch methods (FedNS), approximates the centralized Newton's method by communicating the sketched square-root Hessian instead of the exact Hessian. To enhance communication efficiency, we reduce the sketch size to match the effective dimension of the Hessian matrix. We provide convergence analysis based on statistical learning for the federated Newton sketch approaches. Specifically, our approaches reach super-linear convergence rates w.r.t. the communication rounds for the first time. We validate the effectiveness of our algorithms through various experiments, which coincide with our theoretical findings.

Introduction

Due to the huge potential in terms of privacy protection and reducing computational costs, Federated Learning (FL) (Konečnỳ et al. 2016; McMahan et al. 2017; Li et al. 2020a) becomes a promising framework for handling large-scale tasks. In federated learning, a key problem is to achieve a tradeoff between the convergence rate and the communication burdens.

First-order optimization algorithms have achieved great success in federated learning, including FedAvg (LocalSGD) (McMahan et al. 2017) and FedProx (Li et al. 2020a). These methods communicate the first-order information rather than the data across local machines, which protect the privacy training data and allow the data heterogeneity to some extent. Despite recent efforts and progress on the convergence analysis (Li et al. 2020b,c; Karimireddy et al. 2020; Pathak and Wainwright 2020; Glasgow, Yuan, and Ma 2022) and the generalization analysis (Mohri, Sivek, and Suresh 2019; Yagli, Dytso, and Poor 2020; Su, Xu, and Yang 2021; Yuan et al. 2022) of FedAvg and FedProx, the convergence rate of first-order federated algorithms is still slow, i.e., a sublinear converge rate $\mathcal{O}(1/t)$, where t is the communication rounds.

In the traditional centralized learning, with some mild conditions, second-order optimal algorithms (Boyd, Boyd, and Vandenberghe 2004; Bottou, Curtis, and Nocedal 2018), for example (quasi) Newton's methods, can achieve at least a linear convergence rate. The compute of inverse Hessian is time consuming, and thus many classic approximate Newton's methods are proposed, including BFGS (Broyden 1970), L-BFGS (Liu and Nocedal 1989), inexact Newton (Dembo, Eisenstat, and Steihaug 1982), Gauss-Newton (Schraudolph 2002) and Newton sketch (Pilanci and Wainwright 2017). However, if we directly apply Newton's method to federated learning, the communication complexity of sharing local Hessian matrices is overwhelming.

Indeed, first-order algorithms characterize low communication burdens but slow convergence rates, while Newton's methods lead to fast convergence rates but with high communication complexity. To take advantage of both first-order and second-order algorithms, we propose a federated Newton sketch method, named `FedNS`, which shares both first-order and second-order information across devices, i.e., local gradients and sketched square-root Hessian. Using line search strategy and adaptive learning rates, we devise a dimension-efficient approach `FedNDES`. We then study the convergence properties for the proposed algorithms. We conclude with experiments on publicly available datasets that complement our theoretical results, exhibiting both computational and statistical benefits. We leave proofs in the appendix[1]. We summarize our contributions as below:

1) On the algorithmic front. We propose two fast second-order federated algorithms, which improve the approximation of the centralized Newton's method while the communication costs are favorable due to the small sketch size. Specifically, using line search and adaptive step-sizes in Algorithm 2, the sketch size can be reduced to the effective dimension of the Hessian matrix.

2) On the statistical front. Drawing upon the established outcomes from the centralized sketching Newton literature (Pilanci and Wainwright 2017; Lacotte, Wang, and Pilanci 2021), we furnish convergence analysis for two federated

*Corresponding author

[1]Full version: https://lijian.ac.cn/files/2024/FedNS.pdf
Code: https://github.com/superlj666/FedNS

Newton sketching algorithms: FedNS and its line-search variant FedNDES. These methodologies delineate not only super-linear convergence rates but also entail a small sketch size, corresponding to the effective dimension of the Hessian in the case of FedNDES. Note that the proposed algorithms achieve super-linear convergence rates while upholding a reasonable level of communication complexity.

Related Work

FedAvg and FedProx. FedAvg and FedProx only share local gradients of the size M and the communication complexity is $\mathcal{O}(M)$, where M is the dimension of feature space. The convergence properties of FedAvg and FedProx have been well-studied in (Li et al. 2020c,a; Su, Xu, and Yang 2021), of which the iteration complexity is $T = \mathcal{O}(1/\delta)$ to obtain some δ-accurate solution.

Newton sketch. Newton sketch was proposed in (Pilanci and Wainwright 2017) to accelerate the compute of Newton's methods. It has been further extended, for example, Newton-LESS (Derezinski et al. 2021) employed leverage scores to sparsify the Gaussian sketching matrix and (Lacotte and Pilanci 2020; Lacotte, Wang, and Pilanci 2021) proved the sketch size can be as small as the effective dimension of the Hessian matrix. With the sketching Newton's methods, the communication complexity is $\mathcal{O}(kM)$, where k is the sketch size. In some cases, it leads to a super-linear convergence $T = \mathcal{O}(\log(\log 1/\delta))$ for a δ-accurate solution.

Newton-type federated algorithms. DistributedNewton (Ghosh, Maity, and Mazumdar 2020) and LocalNewton (Gupta et al. 2021) perform Newton's method instead of SGD on local machines to accelerate the convergence of local models. FedNew (Elgabli et al. 2022) utilized one pass ADMM on local machines to calculating local directions and approximate Newton method to update the global model. FedNL (Safaryan et al. 2022) sended the compressed local Hessian updates to global server and performed Newton step globally. Based on eigendecomposition of the local Hessian matrices, SHED (Fabbro et al. 2022) incrementally updated eigenvector-eigenvalue pairs to the global server and recovered the Hessian to use Newton method.

Backgrounds

A standard federated learning system consists of a global server and m local compute nodes. On the j-th worker $\forall\ j \in [m]$, the local training data $\mathcal{D}_j = \{(\boldsymbol{x}_{ij}, y_{ij})\}_{i=1}^{n_j}$ is drawn from a local distribution ρ_j on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. We denote the disjoint union of local training data $\mathcal{D} = \bigcup_{j=1}^{m} \mathcal{D}_j$ as the entire training data that corresponds to a global distribution ρ on $\mathcal{X} \times \mathcal{Y}$. For the sake of privacy preservation and efficient distributed computation, a federated learning system aims to train a global model without sharing local data. The ideal empirical model is trained on the entire training dataset $\mathcal{D}$ w.r.t. the training objective. We denote $n_j := |\mathcal{D}_j|$ the number of local examples on the j-th machine and $N := |\mathcal{D}| = \sum_{j=1}^{m} n_j$ the number of all train examples.

Centralized Newton's Method

The objective of centralized learning on the entire train set $\mathcal{D}$ can be stated as $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle$ with

$$\boldsymbol{w}_{\mathcal{D},\lambda} = \underset{\boldsymbol{w} \in \mathcal{H}_K}{\arg\min} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \ell\left(f(\boldsymbol{x}_i), y_i\right) + \lambda \alpha(\boldsymbol{w})}_{L(\mathcal{D}, \boldsymbol{w})}, \tag{1}$$

where $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$, ℓ is the loss function, $\alpha(\boldsymbol{w})$ is the penalty term and $\lambda > 0$ is the regularity parameter. We assume that f belongs to the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ defined by a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Throughout, we denote the inner product $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_K$ and the corresponding norm by $\|\cdot\|_K$, where $\phi : \mathcal{X} \to \mathcal{H}_K$ is the implicit feature mapping. The reproducing property guarantees kernel methods $f : \mathcal{X} \to \mathcal{Y}$ admitting

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_K, \qquad \forall\ \boldsymbol{w} \in \mathcal{H}_K, \boldsymbol{x} \in \mathcal{X}. \tag{2}$$

If L is twice differentiable convex function in terms of $\boldsymbol{w}$, the centralized problem (1) on $\mathcal{D}$ can be solved by the exact Newton's method

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \mu \boldsymbol{H}_{\mathcal{D},t}^{-1} \boldsymbol{g}_{\mathcal{D},t}, \tag{3}$$

where μ is the step-size and the gradient and the Hessian matrix are computed by

$$\begin{aligned} \boldsymbol{g}_{\mathcal{D},t} &:= \nabla L(\mathcal{D}, \boldsymbol{w}_t) + \lambda \nabla \alpha(\boldsymbol{w}_t), \\ \boldsymbol{H}_{\mathcal{D},t} &:= \nabla^2 L(\mathcal{D}, \boldsymbol{w}_t) + \lambda \nabla^2 \alpha(\boldsymbol{w}_t). \end{aligned}$$

Let the feature mapping be finite dimensional $\phi : \mathcal{X} \to \mathbb{R}^M$. Then, there are finite dimensional gradients $\boldsymbol{g}_{\mathcal{D},t} \in \mathbb{R}^M$ and Hessian matrices $\boldsymbol{H}_{\mathcal{D},t} \in \mathbb{R}^{M \times M}$.

Since the federated learning system cannot visit local training data, it fails to compute the global gradient $\boldsymbol{g}_{\mathcal{D},t}$ and Hessian matrix $\boldsymbol{H}_{\mathcal{D},t}$ directly. The total time complexity for Newton's method is $\mathcal{O}(NM^2t)$ with a super-linear convergence rate $t = \log(\log(1/\delta))$ for δ-approximation guarantee (Dennis Jr and Schnabel 1996), i.e., $L(\mathcal{D}, \boldsymbol{w}_t) - L(\mathcal{D}, \boldsymbol{w}_{\mathcal{D},\lambda}) \leq \delta$.

Newton's Method with Partial Sketching

To improve the computational efficiency of Newton's method, (Pilanci and Wainwright 2017) proposed Newton sketch to construct a structured random embedding of the Hessian matrix with a sketch matrix $\boldsymbol{S} \in \mathbb{R}^{k \times M}$ where $k \ll N$. Instead of sketching the entire Hessian matrix, partial Newton sketch (Pilanci and Wainwright 2017; Lacotte, Wang, and Pilanci 2021) only sketched the loss term $\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t) \approx (\boldsymbol{S}\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t)^{1/2})^\top (\boldsymbol{S}\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t)^{1/2})$ while reserving the exact Hessian for the regularity term $\nabla^2 \alpha(\boldsymbol{w}_t)$. The partial Newton sketch can be stated as

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \mu \widetilde{\boldsymbol{H}}_{\mathcal{D},t}^{-1} \boldsymbol{g}_{\mathcal{D},t}, \qquad \text{with} \\ \widetilde{\boldsymbol{H}}_{\mathcal{D},t} &= (\boldsymbol{S}\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t)^{1/2})^\top (\boldsymbol{S}\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t)^{1/2}) \\ &\quad + \lambda \nabla^2 \alpha(\boldsymbol{w}_t). \end{aligned} \tag{4}$$

Note that, the sketch matrix $\boldsymbol{S} \in \mathbb{R}^{k\times M}$ are zero-mean and normalized $\mathbb{E}[\boldsymbol{S}^\top \boldsymbol{S}/k] = \mathbf{I}_M$. Different types of randomized sketches lead to different the sketch times (Lacotte, Wang, and Pilanci 2021), i.e., $\mathcal{O}(NMk)$ for sub-Gaussian embeddings, $\mathcal{O}(NM\log k)$ for subspace randomized Hadamard transform (SRHT), and $\mathcal{O}(\mathbf{nnz}(\widetilde{\boldsymbol{H}}_{\mathcal{D},t}))$ for the sparse Johnson-Lindenstrauss transform (SJLT). For the SJLT, only one non-zero entry exists per column, which causes much faster sketch time but requires a larger subspace. Throughout this work, we focus on the SRHT because it achieves a tradeoff between fast sketch time and small sketch dimension (Ailon and Chazelle 2006).

There are various examples for the optimization problem (1) where the regularization term is λ-strongly convex and partial sketched square-root Hessian $S\nabla^2 L(\boldsymbol{w})$ is amenable to fast computation. For example, the widely used ridge regression and regularized logistic regression. More examples are referred to Section 3.1 (Lacotte, Wang, and Pilanci 2021).

Federated Learning with Newton's Methods

To reduce the communication burdens in FedNewton, we propose a communication-efficient algorithm to approximate the global Hessian matrix. Based on the Newton Sketch (Pilanci and Wainwright 2017), we devise an efficient Newton sketch method for federated learning, which performs an approximate local Hessians using a randomly projected or sub-sampled Hessian on the local workers. Then, it summarizes the local Hessian matrices to approximate the global Hessian matrix and then performs Newton's method on the approximate global Hessian matrix.

Federated Newton's Method (FedNewton)

To approximate the global model $f_{\mathcal{D},\lambda}$ well, the federated learning algorithms usually share local information to the other clients, i.e. first-order gradient information in FedAvg and FedProx. We consider using both first-order and second-order information to characterize the exact solution of (1) on the entire dataset $\mathcal{D}$.

Noting that, both local gradients and Hessians in (3) can be summarized up to the global gradient and Hessian, respectively. This motivates us to summarize local gradients and Hessians to conduct federated Newton's method (FedNewton)

$$\begin{aligned} &\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \mu \boldsymbol{H}_{\mathcal{D},t}^{-1}\boldsymbol{g}_{\mathcal{D},t} \qquad \text{with} \\ &\boldsymbol{H}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N}\boldsymbol{H}_{\mathcal{D}_j,t}, \quad \boldsymbol{g}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N}\boldsymbol{g}_{\mathcal{D}_j,t}. \end{aligned} \tag{5}$$

Complexity Analysis. Before the iterations, the computation of feature mapping on any local machine consumes $\mathcal{O}(n_j M d)$ time. On the j-th local worker, the time complexity for iteration is at least $\mathcal{O}(n_j M^2 + n_j M)$ to compute $\boldsymbol{H}_{\mathcal{D}_j,t}$ and $\boldsymbol{g}_{\mathcal{D}_j,t}$, respectively. And the time complexity is $\mathcal{O}(mM^2 + M^3)$ to summarize local Hessians and compute the global inverse Hessian. However, the communication cost is $\mathcal{O}(M^2)$ to upload local Hessian matrices in each iteration, which is infeasible in the practical federated learning scenarios. The total computational complexity is $\mathcal{O}(\max_{j\in[m]} n_j M d + n_j M^2 t + M^3 t)$ and the communication complexity is $\mathcal{O}(M^2 t)$, where Newton's method achieves a quadratic convergence $t = \mathcal{O}(\log(\log(1/\delta)))$ for δ-approximation guarantee (Dennis Jr and Schnabel 1996), i.e., $L(\mathcal{D}, \boldsymbol{w}_t) - L(\mathcal{D}, \boldsymbol{w}_{\mathcal{D},\lambda}) \le \delta$, where $\boldsymbol{w}_{\mathcal{D},\lambda}$ is the empirical risk minimizer (1).

Algorithm 1: Federated Learning with Newton Sketch (`FedNS`)

Input: Feature mapping $\phi : \mathbb{R}^d \to \mathbb{R}^M$, start point $\boldsymbol{w}_0$, the termination iterations T and the step-size μ.

Output: The global estimator $\boldsymbol{w}_T$.

1: **Local machines:** Compute the local feature mapping data matrix $\phi(\boldsymbol{X}_j)$.

2: **for** $t = 1$ to T **do**

3: **Local machines:** Sample the sketch matrix $\boldsymbol{S}_j^t \in \mathbb{R}^{k\times n_j}$ from the SRHT. Compute local sketch Hessian matrices $\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} = \boldsymbol{S}_j^t \nabla^2 L(\mathcal{D}_j, \boldsymbol{w}_t)^{1/2}$ and local gradients $\boldsymbol{g}_{\mathcal{D}_j,t}$. Upload them to the global server ($\uparrow$).

4: **Global server:** Compute the global Hessian matrix $\widetilde{\boldsymbol{H}}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N}\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda}^\top \boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} + \lambda\nabla^2\alpha(\boldsymbol{w}_t)$ and the global gradient $\boldsymbol{g}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N}\boldsymbol{g}_{\mathcal{D}_j,t}$ and update the global estimator

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \mu \widetilde{\boldsymbol{H}}_{\mathcal{D},t}^{-1}\boldsymbol{g}_{\mathcal{D},t}$$

and communicate it to local machines ($\downarrow$).

5: **end for**

Federated Learning with Newton Sketch (`FedNS`)

Now suppose the local Hessian matrix square-root $\nabla^2 L(\mathcal{D}, \boldsymbol{w}_t)^{1/2}$ of dimensions $n_j \times M$ is available, from (4), we obtain local sketch Hessian on the empirical loss term

$$\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} = \boldsymbol{S}_j \nabla^2 L(\mathcal{D}_j, \boldsymbol{w}_t)^{1/2}.$$

Here, $\boldsymbol{S}_j \in \mathbb{R}^{k\times n_j}$ is the sketch matrix for the j-th worker with $k \ll n_j$ and thus the communicated sketch Hessian is with a small size $\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} \in \mathbb{R}^{k\times M}$. From (5), we approximate the global Hessian by summarizing local sketching Hessian matrices

$$\widetilde{\boldsymbol{H}}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N}\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda}^\top \boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} + \lambda\nabla^2\alpha(\boldsymbol{w}_t). \tag{6}$$

Here, we approximate the global Hessian with local sketch Hessian matrices $\boldsymbol{H}_{\mathcal{D},t} \approx \widetilde{\boldsymbol{H}}_{\mathcal{D},t}$. We approximate the exact Newton's method on the entire data by local Newton sketch

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \mu \widetilde{\boldsymbol{H}}_{\mathcal{D},t}^{-1}\boldsymbol{g}_{\mathcal{D},t} \tag{7}$$

where $\widetilde{\boldsymbol{H}}_{\mathcal{D},t}$ is the approximate global Hessian from (6) and $\boldsymbol{g}_{\mathcal{D},t}$ is the global gradient from (5).

We formally introduce the general framework for Federated Newton Sketch (`FedNS`) in Algorithm 1. Note that, using sketch Hessian, we communicate $\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda}$ of the dimensions $k \times M$ instead of the exact local Hessian $\boldsymbol{H}_{\mathcal{D}_j,t}$ of

the dimensions M^2. Since local gradients $\boldsymbol{g}_{\mathcal{D}_j,t}$ are M dimensional vectors, the uploads of sketch Hessian matrices dominate the communication complexity.

Complexity Analysis. Before the iterations, the computation of feature mapping on any local machine consumes $\mathcal{O}(n_j M d)$ time. On the j-th local worker, the sketching time is $\mathcal{O}(n_j M \log k)$ for SRHT, while it is $\mathcal{O}(n_j M k)$ for classic sub-Gaussian embeddings. On the global server, the time complexity of FedNS is $\mathcal{O}(mkM^2 + M^3)$ to obtain the global sketch Hessian and its inverse. Nevertheless, FedNS reduces the communication burdens from $\mathcal{O}(M^2)$ to $\mathcal{O}(kM)$. Overall, FedNS not only speedup the local computations, but more importantly reduce the communication costs. The total computational complexity is $\mathcal{O}(\max_{j\in[m]} n_j M d + n_j M t \log k + mkM^2 t + M^3 t)$. More importantly, the communication complexity is reduced from $\mathcal{O}(M^2 t)$ in FedNewton to $\mathcal{O}(kMt)$ in FedNS. The sketch Newton method leads to a super-linear convergence rate $t = \mathcal{O}(\log(1/\delta))$ for δ-approximation guarantee (Pilanci and Wainwright 2017; Lacotte, Wang, and Pilanci 2021). Note that, from Theorem 1, since $k = \Omega(M)$ for FedNS, just using sketching without line-search does not reduce communication cost. We then present line-search step based sketching Federated Newton method, which can guarantee smaller communication costs.

Federated Newton's Method with Dimension-Efficient Sketching (FedNDES)

From the theoretical results in the next section, the sketch size for FedNS is $k \simeq M$ to achieve a super-linear convergence rate, which is still too expensive when M is large to approximate the kernel. Since the communication is determined by the sketch size, we devise a dimension-efficient federated Newton sketch approach (FedNDES), shown in Algorithm 2 to reduce the sketch size, based on the Newton sketch with backtracking line (Armijo) search (Boyd, Boyd, and Vandenberghe 2004; Nocedal and Wright 2006; Pilanci and Wainwright 2017; Lacotte, Wang, and Pilanci 2021). Backtracking line search begins with an initial stepsize μ and backtracks until the adjusted linear estimate overestimates the loss function. For more information refer to (Boyd, Boyd, and Vandenberghe 2004).

Different from FedNS, Algorithm 2 applies the two phases updates with different sketch sizes $\overline{m}_1$ and $\overline{m}_2$. The following theoretical results illustrate both these two sketch sizes depend on the effective dimension that is much smaller than M and N. The approximate Newton decrement $\tilde{\lambda}(\boldsymbol{w}_t)$ is used as an exit condition and the threshold for the adaptive sketch sizes to guarantee smaller iterations and sketch sizes.

Complexity Analysis. Since the compute of approximate global Hessian and the inverse of it dominate the training time, the total computational complexity for FedNDES is similar to FedNS that is $\mathcal{O}(\max_{j\in[m]} n_j M d + n_j M t \log k + mkM^2 t + M^3 t)$. However, the communication complexity $\mathcal{O}(kMt)$ is reduced owing to a smaller sketch size in FedNDES. The sketch size is $k \simeq M$ for FedNS, while it is $k \simeq \mathrm{Tr}(\widetilde{\boldsymbol{H}}_{\mathcal{D},t}(\widetilde{\boldsymbol{H}}_{\mathcal{D},t} + \lambda I)^{-1})$ for FedNDES that is much smaller than M. Smaller sketch size makes the Newton methods practical for multiple communications in federated learning settings. Meanwhile, even with the smaller sketch size, FedNDES can still achieve the super-linear convergence rate $t = \mathcal{O}(\log\log(1/\delta))$ for δ-approximation solution.

Algorithm 2: Dimension-efficient federated Newton (FedNDES)

Input: Feature mapping $\phi : \mathbb{R}^d \to \mathbb{R}^M$, start point $\boldsymbol{w}_0$, accuracy tolerance $\delta > 0$, line-search parameters (a, b), threshold sketch sizes $\overline{m}_1$ and $\overline{m}_2$, and the decrement parameter η.

Output: The global estimator $\boldsymbol{w}_T$.

1: **Local machines:** Compute the local feature mapping data matrix $\phi(\boldsymbol{X}_j)$. Initialize and $k_t = \overline{m}_1$.

2: **for** $t = 1, \cdots, T$ **do**

3: **Local machines:** Sample the sketch matrix $\boldsymbol{S}_j^t \in \mathbb{R}^{k\times n_j}$ from the SRHT. Compute local sketch square root Hessian $\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} = \boldsymbol{S}_j^t \nabla^2 L(\mathcal{D}_j, \boldsymbol{w}_t)^{1/2}$ and local gradients $\boldsymbol{g}_{\mathcal{D}_j,t}$. Upload them to the global server ($\uparrow$).

4: **Global server:** Compute the global Hessian matrix $\widetilde{\boldsymbol{H}}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N} \boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda}^\top \boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} + \lambda \nabla^2 \alpha(\boldsymbol{w}_t)$, the global gradient $\boldsymbol{g}_{\mathcal{D},t} = \sum_{j=1}^m \frac{n_j}{N} \boldsymbol{g}_{\mathcal{D}_j,t}$ and the approximate Newton step $\Delta \boldsymbol{w}_t = -\widetilde{\boldsymbol{H}}_{\mathcal{D},t}^{-1} \boldsymbol{g}_{\mathcal{D},t}$. Compute the approximate Newton decrement

$$\tilde{\lambda}(\boldsymbol{w}_t) = \boldsymbol{g}_{\mathcal{D},t}^\top \Delta \boldsymbol{w}_t.$$

If $\tilde{\lambda}(\boldsymbol{w}_t)^2 \le \frac{3}{4}\delta$ return the model $\boldsymbol{w}_t$. Otherwise send $\Delta \boldsymbol{w}_t$ and $\tilde{\lambda}(\boldsymbol{w}_t)$ to local workers.

5: **Local machines:** Line search from $\mu_j = 1$: **while** $L(\mathcal{D}_j, \boldsymbol{w}_t + \mu_j \Delta \boldsymbol{w}_t) > L(\mathcal{D}_j, \boldsymbol{w}_t) + a\mu_j \tilde{\lambda}(\boldsymbol{w}_t)$, **then** $\mu_j \leftarrow b\mu_j$. Send μ_j to the global server.

6: **Global server:** Let $\mu = \min_{j\in m} \mu_j$. Update the global estimator

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \mu \Delta \boldsymbol{w}_t.$$

If $\tilde{\lambda}(\boldsymbol{w}_t) > \eta$, set $k = \overline{m}_1$. Otherwise, set $k = \overline{m}_2$. Communicate the global model $\boldsymbol{w}_t$ and the sketch size k to local machines ($\downarrow$).

7: **end for**

Theoretical Guarantees

Before the generalization analysis for algorithms, we start with some notations and assumptions.

Assumption 1 (Twice differentiable and convex)**.** *The loss function and regularity function $\ell, \alpha : \mathbb{R}^d \to \mathbb{R}$ are both closed and twice differentiable convex functions and $\nabla^2 \alpha(\boldsymbol{w}) \succeq I_d$.*

Assumption 2 (Strongly convextiy and smoothness)**.** *Let $\gamma = \lambda_{min}(\nabla^2 L(\boldsymbol{w}))$ be the minimum eigenvalue and $\beta = \lambda_{max}(\nabla^2 L(\boldsymbol{w}))$ be the maximum eigenvalue of the Hessian.*

In the standard analysis of Newton's method, γ and β are commonly used to measure the strong convexity and

smoothness of the objective function L (Pilanci and Wainwright 2017).

Assumption 3 (Lipschitz continuous Hessian)**.** *The Hessian map is Lipschitz continuous with modulus G, i.e., $\|\nabla^2 L(\boldsymbol{w}) - \nabla^2 L(\boldsymbol{w}')\| \leq G\|\boldsymbol{w} - \boldsymbol{w}'\|_2$.*

The above assumptions are standard conditions for both convex and non-convex optimization. Under these conditions, with the appropriate initialization $\|\boldsymbol{w} - \boldsymbol{w}_{\mathcal{D},\lambda}\| \leq \frac{\gamma}{2G}$, the Newton approach can guarantee a quadratic convergence (Boyd, Boyd, and Vandenberghe 2004). Let $\boldsymbol{w}_t$ be the federated Newton model defined in `FedNS` or `FedNDES`, and $\boldsymbol{w}^*$ be the target model. It holds the following error decomposition

$$\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2 \leq \underbrace{\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2}_{\text{federated error}} + \underbrace{\|\boldsymbol{w}_{\mathcal{D},\lambda} - \boldsymbol{w}^*\|_2}_{\text{centralized excess risk}}, \quad (8)$$

where $\boldsymbol{w}_{\mathcal{D},\lambda}$ is the centralized ERM model on the training data $\mathcal{D}$. Since the centralized excess risk $L(\boldsymbol{w}_{\mathcal{D},\lambda}) - L(\boldsymbol{w}^*)$ is standard in statistical learning (Bartlett and Mendelson 2002; Caponnetto and De Vito 2007), we focused on the federated error term.

Convergence Analysis for `FedNS`

Theorem 1 in (Pilanci and Wainwright 2017) provided the convergence analysis for *centralized* Newton sketch in (4). Based on it, we present on the generalization analysis for *federated* Newton sketch in Algorithm 1.

Theorem 1 (Convergence guarantees of FedNS)**.** *Let $\delta \in (0, 1)$. Under Assumptions 1, 2, 3, FedNS updates in Algorithm 1 based on an appropriate initialization $\|\boldsymbol{w}_0 - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2 \leq \frac{\delta\gamma}{8G}$. Using the iteration-dependent sketching accuracy $\epsilon = \frac{1}{\log(1+t)}$ and sketch size $k = \Omega(M)$, with the probability at least $1 - c_1 e^{-c_2 k\epsilon^2}$, we have*

$$\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2 \leq \frac{1}{\log(1+t)} \frac{\beta}{\gamma} \|\boldsymbol{w}_{t-1} - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2.$$

This guarantee a super-linear convergence rate, since $\lim_{t\to\infty} \frac{\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2}{\|\boldsymbol{w}_{t-1} - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2} = 0$. Here, $\boldsymbol{w}_{\mathcal{D},\lambda}$ is the centralized model on $\mathcal{D}$ and $\boldsymbol{w}_t$ is the federated model trained in Algorithm 1.

The above theorem shows that, depending on the structure of the problem γ, β, `FedSN` achieves the super-linear convergence rate using a sufficient sketch size $k \gtrsim M$. And thus, the communication complexity in each iteration is at least $\mathcal{O}(M^2)$, the same as FedNewton. There are two drawbacks in Theorem 1: 1) The analysis depends on some constants from the properties of L, i.e., the curvature constants γ, β and the Lipschitz constant G, which are usually unknown in practice. 2) Theorem 1 requires a initialization condition $\|\boldsymbol{w}_0 - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2 \leq \frac{\delta\gamma}{8G}$. However, it is hard to find a appropriate start point $\boldsymbol{w}_0$ satisfying the condition in practice. 3) The communication complexity of `FedSN` $\mathcal{O}(kMt)$ depends on the sketch size, but the communication of current sketch size $k \gtrsim M$ is overly expensive.

Convergence Analysis for `FedNDES`

To derive global convergence free from unknown problem parameters, we require a new condition.

Assumption 4 (Self-concordant function)**.** *A closed convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$ is self-concordant if $|\varphi'''(\boldsymbol{w})| \leq 2(\varphi''(\boldsymbol{w}))^{3/2}$. We assume the loss function ℓ is a convex self-concordant function.*

The condition extends to the loss function $\ell : \mathbb{R}^d \to \mathbb{R}$ by imposing the requirement on the univariate function $\varphi_{\boldsymbol{w},\boldsymbol{\omega}}(t) := \ell(\boldsymbol{w} + t\boldsymbol{\omega})$ for $\boldsymbol{w}, \boldsymbol{\omega}$ in the domain of ℓ. Examples for self-concordant functions include linear, quadratic functions, negative logarithm, and more examples can be found in (Lacotte, Wang, and Pilanci 2021).

Definition 1 (Empirical effective dimension)**.** *If the regularity term is λ-strongly convex, the empirical effective Hessian dimension is defined as $\tilde{d}_\lambda := \sup_{\boldsymbol{w}\in\mathcal{H}_K} \operatorname{Tr}\left(\nabla^2 L(\mathcal{D}, \boldsymbol{w})(\nabla^2 L(\mathcal{D}, \boldsymbol{w}) + \lambda I)^{-1}\right)$.*

The empirical effective dimension $\tilde{d}_\lambda$ is substantially smaller than the feature space M (Bach 2013; Alaoui and Mahoney 2015). The effective dimension is related to the covariance matrix $\nabla^2 L(\mathcal{D}, \boldsymbol{w}) = \frac{1}{N}\phi(\boldsymbol{X})^\top \phi(\boldsymbol{X})$, which has been well-studied for leverage scores sampling (Rudi et al. 2018; Luise et al. 2019; Chen and Yang 2021) in low-rank approximation. Meanwhile, the expected effective dimension is defined as $\mathcal{N}(\lambda) = \operatorname{Tr}(T_K(T_K + \lambda I)^{-1})$ based on the covariance operator $T_K = \int_X \langle \cdot, \phi(\boldsymbol{x})\rangle \phi(\boldsymbol{x}) d\rho_X(\boldsymbol{x})$, which has been widely used to prove the optimal learning guarantees for the squared loss (Caponnetto and De Vito 2007; Smale and Zhou 2007).

Without initialization condition, we provide the following convergence guarantee for `FedNDES`.

Theorem 2 (Convergence guarantees of FedNDES)**.** *Let $\delta \in (0, 1)$ and the sketch matrices be SRHT. Under Assumptions 1, 4, FedNDES updates in Algorithm 2, then with a high probability, the number of iterations T and the sketch size k satisfying*

$$T = \mathcal{O}\left(\log\log\left(\frac{1}{\delta}\right)\right), \quad k = \Omega\left(\tilde{d}_\lambda\right),$$

can obtain a δ-accurate solution $\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\| \leq \delta$ without any initialization condition. Here, $\boldsymbol{w}_{\mathcal{D},\lambda}$ is the centralized model on $\mathcal{D}$ and $\boldsymbol{w}_t$ is the federated model trained in Algorithm 2.

Compared to Theorem 1, the above theorem removes the initialization condition. More importantly, it reduces the sketch size from M to $\tilde{d}_\lambda$, which is much smaller than M and thus it is more practical in federated learning settings. Since the communicated local sketch square root Hessian $\boldsymbol{\Upsilon}_{\mathcal{D}_j,\lambda} \in \mathbb{R}^{k\times M}$, the communication complexity in each iteration is $\mathcal{O}(\tilde{d}_\lambda M)$. For example, supposing $\tilde{d}_\lambda \gtrsim \log\log(1/\delta)$ as used in (Lacotte, Wang, and Pilanci 2021) with a standard learning rate $\delta = \mathcal{O}(1/\sqrt{N})$, we obtain the sketch size $\tilde{d}_\lambda \gtrsim \log(0.5\log N)$, which is significantly smaller than M.

Since the generalization error δ and effective dimension $\tilde{d}_\lambda$ are relevant to the specific tasks, they are not estimated

Table 1: Summary of communication properties for related work.

Related Work	Heterogeneous setting	Sketch size k	Iterations t	Communication per round	Total communication complexity
FedAvg (Li et al. 2020c; Su, Xu, and Yang 2021)	$\surd$	–	$\mathcal{O}\left(\frac{1}{\delta}\right)$	$\mathcal{O}(M)$	$\mathcal{O}(\frac{M}{\delta})$
FedProx (Li et al. 2020a; Su, Xu, and Yang 2021)	$\surd$	–	$\mathcal{O}\left(\frac{1}{\delta}\right)$	$\mathcal{O}(M)$	$\mathcal{O}(\frac{M}{\delta})$
DistributedNewton (Ghosh, Maity, and Mazumdar 2020)	$\times$	–	$\mathcal{O}\left(\log \frac{1}{\delta}\right)$	$\mathcal{O}(M)$	$\mathcal{O}(M \log \frac{1}{\delta})$
LocalNewton (Gupta et al. 2021)	$\times$	–	$\mathcal{O}\left(\log \frac{1}{\delta}\right)$	$\mathcal{O}(M)$	$\mathcal{O}(M \log \frac{1}{\delta})$
FedNL (Safaryan et al. 2022)	$\surd$	–	$\mathcal{O}\left(\log \frac{1}{\delta}\right)$	$\mathcal{O}(M)$	$\mathcal{O}(M \log \frac{1}{\delta})$
SHED (Fabbro et al. 2022)	$\surd$	–	$\mathcal{O}(\log \frac{1}{\delta})$	–	$\mathcal{O}(M^2)$
FedNewton	$\surd$	–	$\mathcal{O}\left(\log\log \frac{1}{\delta}\right)$	$\mathcal{O}(M^2)$	$\mathcal{O}(M^2 \log\log \frac{1}{\delta})$
`FedNS` (Algorithm 1)	$\surd$	M	$\mathcal{O}\left(\log\log \frac{1}{\delta}\right)$	$\mathcal{O}(kM)$	$\mathcal{O}(kM \log\log \frac{1}{\delta})$
`FedNDES` (Algorithm 2)	$\surd$	$\tilde{d}_\lambda$	$\mathcal{O}\left(\log\log \frac{1}{\delta}\right)$	$\mathcal{O}(kM)$	$\mathcal{O}(kM \log\log \frac{1}{\delta})$

Note: The computational complexities are computed in terms of regularized least squared loss to obtain a δ-accurate solution, i.e., $L(\boldsymbol{w}_t) - L(\boldsymbol{w}_{\mathcal{D},\lambda}) \leq \delta$. The convergence analysis for FedAvg is provided in (Li et al. 2020c; Su, Xu, and Yang 2021) and that for FedProx is provided in (Li et al. 2020a; Su, Xu, and Yang 2021).

Dataset	n	M	k	m	ρ	α
phishig	$11,055$	68	17	40	0.1	0.25
cod-rna	$59,535$	8	10	60	30	1
covtype	$581,012$	54	20	200	50	1
SUSY	$5,000,000$	18	10	1000	50	1

Table 2: Summary of datasets and hyperparameters.

in Theorem 2. However, they are important to measure the communication complexity in federated learning. It allows to provide accurate estimates for the error δ and the empirical effective dimension $\tilde{d}_\lambda$.

Compared with Related Work

Table 1 reports the computational properties of related work to achieve δ-accurate solutions. In terms of the regularized least squared loss, we compare the proposed `FedNS` and FedNDES with first-order algorithms, and Newton-type FL methods. `FedNS` applies to commonly used sketch approaches, e.g. sub-Gaussian, SRHT, and SJLT, while `FedNDES` only applies to SRHT. Different sketch types leads to various sketch times on the j-th local machine, i.e., $\mathcal{O}(n_j M k)$ for sub-Gaussian, $\mathcal{O}(n_j M \log k)$ for SRHT and $\mathcal{O}(\text{nnz}(\phi(\boldsymbol{X}_j)))$ for SJLT.

Compared with first-order algorithms. Federated Newton's methods converge much faster, $\mathcal{O}(\log 1/\delta)$ v.s. $\mathcal{O}(1/\delta)$. But the communication complexities of federated Newton's methods are much higher, at least $\mathcal{O}(kMt)$, while it is $\mathcal{O}(Mt)$ for FedAvg and FedProx. The proposed `FedNDES` achieves balance between fast convergence rate and small communication complexity, of which the convergence rate is $\mathcal{O}(\log 1/\delta)$ and the communication complexity is $\mathcal{O}(\tilde{d}_\lambda Mt)$.

Compared with Newton-type FL methods. DistributedNewton (Ghosh, Maity, and Mazumdar 2020) and LocalNewton (Gupta et al. 2021) perform Newton's method instead of SGD on local machines. However, they only utilized local information and implicitly assume the local datasets cross devices are homogeneous, which limits their application in FL. In contrast, our proposed algorithms communicate local sketching Hessian matrices to approximate the global one, which are naturally applicable to heterogeneous settings. More recently, there are three Newton-type FL methods:

- **FedNL** (Safaryan et al. 2022) compressed the difference between the local Hessian and the global Hessian from the previous step, and transferred the compressed difference to the global server for merging. On the theoretical front, FedNL achieved at least the linear convergence $\mathcal{O}(\log 1/\delta)$ with the communication cost $O(M)$ per round.
- **FedNew** (Elgabli et al. 2022) used ADMM to solve an unconstrained convex optimization problem for obtaining the local update directions $\boldsymbol{H}_{\mathcal{D}_j,t}^{-1}\boldsymbol{g}_{\mathcal{D}_j,t}$ and performed Newton's method by averaging these directions in the server. However, this work only proved the algorithm can converge but without the convergence rates.
- **SHED** (Fabbro et al. 2022) first performed eigendecomposition on the local Hessian and incrementally send the eigenvalues and eigenfunctions to the server. The local Hessians were recovered on the server to perform Newton's method. The algorithm achieved sup-linear convergence with the total communications costs $\mathcal{O}(M^2)$.

These recent Newton-type FL methods usually admit linear convergence rates, while the proposed algorithms in this work reach super-linear convergences.

Experiments

In this section, we carry out experiments to corroborate our theoretical statements on several real-world federated datasets. We implemented our methods by utilizing the public code from (Elgabli et al. 2022) which includes FedNew

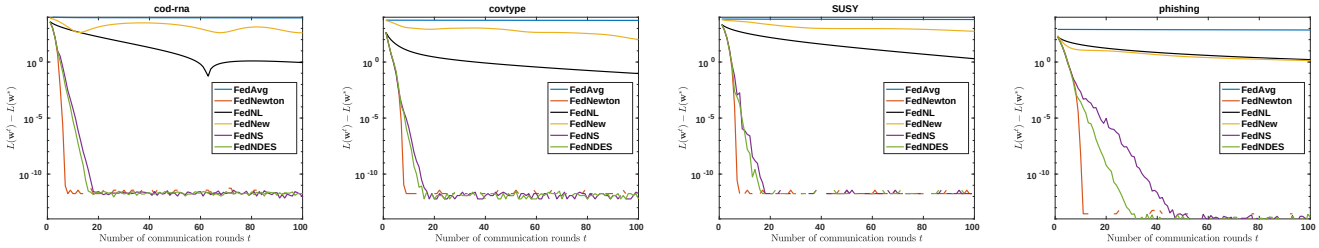

Figure 1: The loss discrepancy between the compared methods and the optimal learner in terms of the number of communication rounds t.

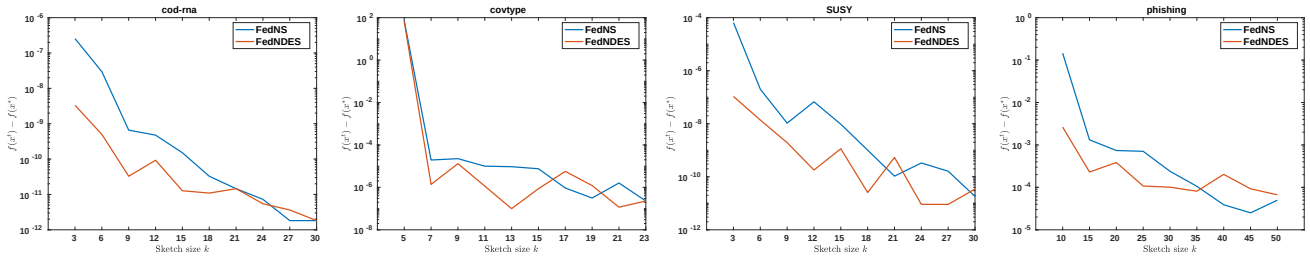

Figure 2: The loss discrepancy between the compared methods and the optimal learner in terms of the sketch size k on the datasets cod-rna, covtype, SUSY, and phishing

(Elgabli et al. 2022) and FedNL (Safaryan et al. 2022), while SHED algorithm was excluded due to the lack of public code (Fabbro et al. 2022). The base model is a logistic regression and the algorithms update the Hessian at each iteration. We first explore the impact of sketching size on the proposed `FedNS` and `FedNDES`, and then compare related algorithms w.r.t. the communication rounds.

Following FedNew (Elgabli et al. 2022), we consider the regularized logistic regression $L(\mathcal{D}, \boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp(y_i \boldsymbol{x}_i^\top \boldsymbol{w})\right) + \lambda\|\boldsymbol{w}\|_2^2$, where λ is a regularization parameter chosen to set 10^{-3}. All experiments are recorded by averaging results after 10 trials and figures report the mean value. We use the optimal gap $L(\boldsymbol{w}^t) - L(\boldsymbol{w}^*)$ as the performance indicator, where we use the global Newton's method as the optimal estimator $\boldsymbol{w}^*$. We evaluate the compared algorithms on public LIBSVM Data (Chang and Lin 2011). We report the statistics of datasets and the corresponding hyperparameters in Table 2, where ρ and α hyperparameters are used in FedNew.

Convergence comparison. Figures 1 reports the convergence of compared methods, demonstrating that: 1) There is significant gaps between the convergence speeds of our proposed methods `FedNS`, `FedNDES` and the existing Newton-type FL methods, i.e. FedNew and FedNL. This validate the super-linear convergence of `FedNS` and `FedNDES`. 2) The proposed `FedNS` and `FedNDES` converge nearly to FedNewton, while FedNew and FedNL converge slowly closed to FedAvg. 3) `FedNDES` converges faster than `FedNS` and the final predictive accuracies of `FedNDES` are higher. 4) Even with small sketch sizes, the proposed sketched Newton-type FL methods can still preserve considerable accuracy. 5) Although our communication cost is higher $\mathcal{O}(Mk)$ than FedNew and FedNL, the number of communications is much smaller, resulting in lower total communications for our methods.

Impact of sketch size on Performance Figure 2 reports the predictive accuracies versus the sketch size, illustrating that 1) A larger sketch size always leads to better generalization performance. 2) The proposed `FedNS` and `FedNDES` finally converges around the global Newton' method. 3) A small sketch size, i.e., $k \ll M$ and $k \ll N$, can still achieve good performance. 4) `FedNDES` obtains better generalization performance than `FedNS` with smaller sketch size.

Conclusion

Both convergence rate and communication costs are important to federated learning algorithms. In this paper, by sketching the square-root Hessian, we devise federated Newton sketch methods, which communicate sketched matrices instead of the exact Hessian. Theoretical guarantees show that the proposed algorithms achieve super-linear convergence rates with moderate communication costs. Specifically, the sketch size of `FedNDES` can be small as the effective dimension of Hessian matrix.

Our techniques pave the way for designing Newton-type distributed algorithms with fast convergence rates. There are some future directions, including 1) One can employ adaptive effective dimension to effectively estimate the effective dimension of Hessian in practice (Lacotte, Wang, and Pilanci 2021). 2) We consider the sparsification for the sketch Newton update in future (Derezinski et al. 2021) to further reduce the communication complexity. 3) The proposed methods and theoretical guarantees can be extened to decentralized learning (Hsieh et al. 2020).

Acknowledgments

The work of Jian Li is supported partially by National Natural Science Foundation of China (No. 62106257), and Project funded by China Postdoctoral Science Foundation (No. 2023T160680). The work of Yong Liu is supported partially by National Natural Science Foundation of China (No.62076234), Beijing Outstanding Young Scientist Program (No.BJJWZYJH012019100020098), the Unicom Innovation Ecological Cooperation Plan, and the CCF-Huawei Populus Grove Fund.

References

Ailon, N.; and Chazelle, B. 2006. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the 38th annual ACM Symposium on Theory of Computing (STOC)*, 557–563.

Alaoui, A.; and Mahoney, M. W. 2015. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 775–783.

Bach, F. 2013. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, 185–209.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research (JMLR)*, 3(Nov): 463–482.

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2018. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2): 223–311.

Boyd, S.; Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Broyden, C. G. 1970. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1): 76–90.

Caponnetto, A.; and De Vito, E. 2007. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3): 331–368.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3): 1–27.

Chen, Y.; and Yang, Y. 2021. Fast Statistical Leverage Score Approximation in Kernel Ridge Regression. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2935–2943. PMLR.

Dembo, R. S.; Eisenstat, S. C.; and Steihaug, T. 1982. Inexact newton methods. *SIAM Journal on Numerical analysis*, 19(2): 400–408.

Dennis Jr, J. E.; and Schnabel, R. B. 1996. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. SIAM.

Derezinski, M.; Lacotte, J.; Pilanci, M.; and Mahoney, M. W. 2021. Newton-LESS: Sparsification without Trade-offs for the Sketched Newton Update. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2835–2847.

Elgabli, A.; Issaid, C. B.; Bedi, A. S.; Rajawat, K.; Bennis, M.; and Aggarwal, V. 2022. FedNew: A communication-efficient and privacy-preserving Newton-type method for federated learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 5861–5877. PMLR.

Fabbro, N. D.; Dey, S.; Rossi, M.; and Schenato, L. 2022. A Newton-type algorithm for federated learning based on incremental Hessian eigenvector sharing. *arXiv preprint arXiv:2202.05800*.

Ghosh, A.; Maity, R. K.; and Mazumdar, A. 2020. Distributed newton can communicate less and resist byzantine workers. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, volume 33, 18028–18038.

Glasgow, M. R.; Yuan, H.; and Ma, T. 2022. Sharp Bounds for Federated Averaging (Local SGD) and Continuous Perspective. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9050–9090. PMLR.

Gupta, V.; Ghosh, A.; Derezinski, M.; Khanna, R.; Ramchandran, K.; and Mahoney, M. 2021. LocalNewton: Reducing communication bottleneck for distributed learning. *arXiv preprint arXiv:2105.07320*.

Hsieh, K.; Phanishayee, A.; Mutlu, O.; and Gibbons, P. 2020. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 4387–4398. PMLR.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 5132–5143. PMLR.

Konečnỳ, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Lacotte, J.; and Pilanci, M. 2020. Effective dimension adaptive sketching methods for faster regularized least-squares optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 19377–19387.

Lacotte, J.; Wang, Y.; and Pilanci, M. 2021. Adaptive Newton sketch: linear-time optimization with quadratic convergence and effective Hessian dimensionality. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 5926–5936. PMLR.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020a. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020b. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems 2020 (MLSys)*.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020c. On the Convergence of FedAvg on Non-IID Data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Lin, J.; and Cevher, V. 2020. Optimal Convergence for Distributed Learning with Stochastic Gradient Methods and Spectral Algorithms. *Journal of Machine Learning Research (JMLR)*, 21(147): 1–63.

Liu, D. C.; and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1): 503–528.

Liu, Y.; Liu, J.; and Wang, S. 2021. Effective Distributed Learning with Random Features: Improved Bounds and Algorithms. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

Luise, G.; Stamos, D.; Pontil, M.; and Ciliberto, C. 2019. Leveraging Low-Rank Relations Between Surrogate Tasks in Structured Prediction. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, 4193–4202.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1273–1282. PMLR.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 4615–4625. PMLR.

Nocedal, J.; and Wright, S. 2006. *Numerical optimization*. Springer Science & Business Media.

Pathak, R.; and Wainwright, M. J. 2020. FedSplit: an algorithmic framework for fast federated optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, volume 33, 7057–7066.

Pilanci, M.; and Wainwright, M. J. 2017. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1): 205–245.

Rudi, A.; Calandriello, D.; Carratino, L.; and Rosasco, L. 2018. On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 5672–5682.

Rudi, A.; and Rosasco, L. 2017. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 3215–3225.

Safaryan, M.; Islamov, R.; Qian, X.; and Richtarik, P. 2022. FedNL: Making Newton-Type Methods Applicable to Federated Learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 18959–19010. PMLR.

Schraudolph, N. N. 2002. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7): 1723–1738.

Smale, S.; and Zhou, D.-X. 2007. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26(2): 153–172.

Su, L.; Xu, J.; and Yang, P. 2021. A Non-parametric View of FedAvg and FedProx: Beyond Stationary Points. *arXiv preprint arXiv:2106.15216*.

Yagli, S.; Dytso, A.; and Poor, H. V. 2020. Information-theoretic bounds on the generalization error and privacy leakage in federated learning. In *Proceedings of the 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5. IEEE.

Yuan, H.; Morningstar, W. R.; Ning, L.; and Singhal, K. 2022. What Do We Mean by Generalization in Federated Learning? In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

Proofs

Convergence Analysis for `FedNS`

Proof of Theorem 1. The main difference is that `FedNS` sketches local Hessian $\boldsymbol{S}_j^t \boldsymbol{H}_{\mathcal{D}_j,t}^{1/2}$ on the loss function while `FedNS` directly sketches the global Hessian $\boldsymbol{S}^t \boldsymbol{H}_{\mathcal{D},t}^{1/2}$ on the objective.

In Algorithm 1, we generalize local sketch matrices $(\boldsymbol{S}_j^t)_{j=1}^m$ of the dimension $k \times n_j$ in an independent serialization. By concatenating local sketch matrices in the column direction and local square-root Hessian matrices in the row direction, we obtain

$$\begin{aligned}\boldsymbol{S}^t &= [\boldsymbol{S}_1^t, \cdots, \boldsymbol{S}_m^t], \\ \boldsymbol{H}_{\mathcal{D},t}^{1/2} &= [\boldsymbol{H}_{\mathcal{D}_1,t}, \cdots, \boldsymbol{H}_{\mathcal{D}_m,t}]^\top.\end{aligned}$$

The above equations lead to

$$\boldsymbol{S}^t \boldsymbol{H}_{\mathcal{D},t}^{1/2} = \sum_{j=1} \boldsymbol{S}_j^t \boldsymbol{H}_{\mathcal{D}_j,t}^{1/2}.$$

Therefore, the update of `FedNS` recovers the centralized Newton's method.

From Corollary 1 (Pilanci and Wainwright 2017), the sketch size is lower bounded by the form of the squared Gaussian width, which is at most $\min\{N, M\}$. Since $N > M$ in federated learning, we have $k \gtrsim M$. The distance $\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|$ becomes substantially less than 1 as the iteration increase. And then from Corollary 1 in (Pilanci and Wainwright 2017), considering the Newton sketch iterates using the iteration-dependent sketching accuracy $\epsilon = \frac{1}{\log(1+t)}$, it holds with the probability at least $1 - c_1 e^{-c_2 k \epsilon^2}$ that

$$\begin{aligned}&\|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2 \\ \leq &\frac{1}{\log(1+t)} \frac{\beta}{\gamma} \|\boldsymbol{w}_{t-1} - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2 + \frac{4L}{\gamma} \|\boldsymbol{w}_t - \boldsymbol{w}_{\mathcal{D},\lambda}\|_2^2.\end{aligned}$$

Note that from Lemma 1 in (Pilanci and Wainwright 2017), the sketch size satisfies $m \gtrsim \epsilon^{-2} M = \frac{1}{\log^2(1+t)} M$. □

Convergence Analysis for `FedNDES`

Proof of Theorem 2. From Theorem 2 (Pilanci and Wainwright 2017) and Lemma (Lacotte, Wang, and Pilanci 2021), based on the backtracking parameters (a, b) in Algorithm 2, we define the parameters

$$\nu := ab \frac{\eta^2}{1 + \left(\frac{1+\epsilon}{1-\epsilon}\right)\eta}, \qquad \eta := \frac{1}{8} \frac{1 - \frac{1}{2}\left(\frac{1+\epsilon}{1-\epsilon}\right)^2 - a}{\left(\frac{1+\epsilon}{1-\epsilon}\right)^3}.$$

Then, from Theorem 2 (Lacotte, Wang, and Pilanci 2021), to obtain δ-accurate solution with the probability at least $1 - p_0$, the number of total iterations T should satisfy the condition

$$T \leq \overline{T} := \frac{L(\boldsymbol{w}_0) - L(\boldsymbol{w}_{\mathcal{D},\lambda})}{\nu} + T_{\tau, \frac{3}{8}\delta} + 1,$$

where $\lim_{\tau \to 0} T_{\tau, \frac{3}{8}\delta} \leq \frac{\log(8/3\delta)}{\log(25/16)}$.

Meanwhile, two stages sketch sizes should satisfy

$$\begin{aligned}\overline{m}_1 &\gtrsim \tilde{d}_\lambda + \log\left(\frac{\overline{T}}{p_0}\right) \log\left(\frac{\tilde{d}_\lambda \overline{T}}{p_0}\right), \\ \overline{m}_2 &\gtrsim \delta^{-\tau} \left[\tilde{d}_\lambda + \log\left(\frac{\overline{T}}{p_0 \delta^{\tau/2}}\right) \log\left(\frac{\tilde{d}_\lambda \overline{T}}{p_0}\right)\right].\end{aligned}$$

We consider the linear convergence case, i.e., $\tau = 1$. Using $\tau = 0$ and ignoring the logarithm terms, we obtain $p_0 \asymp \frac{1}{\tilde{d}_\lambda}$, the sketch sizes $\overline{m}_1 \asymp \tilde{d}_\lambda$ and $\overline{m}_2 \asymp \frac{\tilde{d}_\lambda \log(\tilde{d}_\lambda/\delta)}{\delta}$, and the number of iterations $T \lesssim \log\log(\frac{1}{\delta})$. □

Generalization Analysis for FedNS with the Squared Loss

In the above sections, we present the converge analysis for FedNS and FedNDES, but the generalization analysis relies on the specific loss function. Here, we consider the squared loss with the RKHS and the ridge regularization, i.e. kernel ridge regression (KRR). Together with the classic integral operator theory, we derive the generalization error bound for KRR with the optimal learning rates.

The target of regression learning is to find a predictor to approximate the true regression in the RKHS

$$f^*(\boldsymbol{x}) = \int_{\mathcal{Y}} y d\rho(y|\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{X}. \tag{9}$$

Assumption 5 (Source condition)**.** *Define the integral operators $L : L^2(\mathbb{P}) \to L^2(\mathbb{P})$,*

$$(Lg)(\cdot) = \int_X \langle \phi(\cdot), \phi(\boldsymbol{x}) \rangle g(\boldsymbol{x}) d\rho_X(\boldsymbol{x}), \quad \forall\, g \in L^2(\mathbb{P}).$$

Assume there exists $R > 0$, $r \in [1/2, 1]$, such that

$$\|L^{-r} f^*\| \leq R.$$

where the operator L^r denotes the r-th power of L as a compact and positive operator.

Assumption 6 (Capacity condition)**.** *For $\lambda \in (0, 1)$, we define the effective dimensions as*

$$\mathcal{N}(\lambda) = \mathit{Tr}(C(C + \lambda I)^{-1}),$$

Assume there exists $Q > 0$ and $\gamma \in [0, 1]$, such that

$$\mathcal{N}(\lambda) \leq Q^2 \lambda^{-\gamma}.$$

Both source condition and capacity condition are standard assumptions in the optimal statistical learning for the KRR related literature (Caponnetto and De Vito 2007; Smale and Zhou 2007; Rudi and Rosasco 2017; Lin and Cevher 2020; Liu, Liu, and Wang 2021). The effective dimension $\mathcal{N}(\lambda)$ measure the capacity of the RKHS $\mathcal{H}_K$, and it is the expected version of $\tilde{d}_\lambda$ for KRR, depending on the distribution rather than the sample.

Theorem 3 (Excess risk bound for FedNS with the squared loss)**.** *Under Assumptions 5, 6, $r \in [1/2, 1]$ and $\gamma \in [0, 1]$, then with a high probability, the number of iterations T and the sketch size k satisfying*

$$T = \mathcal{O}\left(\frac{2r}{2r+\gamma} \log N\right),$$
$$k = \begin{cases} \Omega\left(M\right), & \text{for FedNS} \\ \Omega\left(N^{\frac{\gamma}{2r+\gamma}}\right), & \text{for FedNDES.} \end{cases}$$

can obtain the following generalization error bound

$$L(\boldsymbol{w}_t) - L(\boldsymbol{w}^*) = \mathcal{O}\left(N^{\frac{-2r}{2r+\gamma}}\right).$$

Note that, $\boldsymbol{w}^* \in \mathcal{H}_K$ is the RKHS model for the target predictor $f^*(\boldsymbol{x}) = \langle \boldsymbol{w}^*, \phi(\boldsymbol{x}) \rangle$.

Proof of Theorem 3. From the error decomposition (8), we have

$$L(\boldsymbol{w}_t) - L(\boldsymbol{w}^*) \leq L(\boldsymbol{w}_t) - L(\boldsymbol{w}_{\mathcal{D},\lambda}) + L(\boldsymbol{w}_{\mathcal{D},\lambda}) - L(\boldsymbol{w}^*), \tag{10}$$

where $\boldsymbol{w}_{\mathcal{D},\lambda} = \left[\phi(\boldsymbol{X})^\top \phi(\boldsymbol{X}) + \lambda N I\right]^{-1} \phi(\boldsymbol{X})^\top \boldsymbol{y}$ is the closed-form solution on the entire training data $\mathcal{D}$.

Under Assumptions 5, 6, and setting $\lambda = N^{\frac{-1}{2r+\gamma}}$, the excess risk bound for centralized KRR is standard (Caponnetto and De Vito 2007; Smale and Zhou 2007)

$$L(\boldsymbol{w}_{\mathcal{D},\lambda}) - L(\boldsymbol{w}^*) = \mathcal{O}\left(N^{\frac{-2r}{2r+\gamma}}\right). \tag{11}$$

We let $\delta \asymp N^{\frac{-2r}{2r+\gamma}}$, and then the federated error holds with a high probability

$$L(\boldsymbol{w}_t) - L(\boldsymbol{w}_{\mathcal{D},\lambda}) = \mathcal{O}\left(N^{\frac{-2r}{2r+\gamma}}\right) \tag{12}$$

From Theorem 1, since the square loss satisfy Assumptions 1, 2, 3, the number iterations and the sketch size for FedNS achieve

$$T = \mathcal{O}\left(\frac{2r}{2r+\gamma}\log N\right), \quad k = \Omega(M). \tag{13}$$

Similarly, from Theorem 2, since the square loss satisfy Assumptions 1, 4, the number iterations and the sketch size for FedNDES should satisfy

$$T = \mathcal{O}\left(\frac{2r}{2r+\gamma}\log N\right), \quad k = \Omega\left(\tilde{d}_\lambda\right) = \Omega\left(N^{\frac{\gamma}{2r+\gamma}}\right). \tag{14}$$

The last step is due to $(1/3)N^{\frac{\gamma}{2r+\gamma}} \leq \tilde{d}_\lambda \leq 3N^{\frac{\gamma}{2r+\gamma}}$ from Lemma 1 (Rudi et al. 2018) together with Assumption 6 and $\lambda = N^{\frac{-1}{2r+\gamma}}$.

Substituting (11), (12), (13) and (14) to (10), we prove the theorem. □

The learning rate in the above generalization error bound is $\mathcal{O}(N^{\frac{-2r}{2r+\gamma}})$, which is optimal in the minimax sense (Caponnetto and De Vito 2007). Since both the number of iterations T and the sketch size k are estimated, we can compute the total time complexity and communication complexity. For the sake of simplicity, we assume $n_1 = \cdots = n_m = N/m$. For FedNS, the total computational complexity is $\mathcal{O}(\max_{j\in[m]} n_j M d + n_j M \log M + m M^3 + M^2 \log N)$ and the communication complexity is $\mathcal{O}(M^2 \log N)$. For FedNDES, the total computational complexity is $\mathcal{O}(\max_{j\in[m]} n_j M d + n_j M \log N + m N^{\frac{\gamma}{2r+\gamma}} M^2 + M^3 + M^2 \log N)$ and the communication complexity is $\mathcal{O}(M N^{\frac{\gamma}{2r+\gamma}} \log N)$. Note that, when $M \leq N^{\frac{\gamma}{2r+\gamma}}$, FedNS obtain smaller complexities but it requires the initialization condition.

In the worst case ($r = 1/2, \gamma = 1$), without Assumptions 5, 6, the sketch size is $k = \Omega(\sqrt{N})$ to achieve the optimal learning rate. In the benign case $\gamma \to 0$, a constant number of sketch size $k = \Omega(1)$ is sufficient to guarantee the optimal rate.