FedNL: Making Newton-Type Methods Applicable to Federated Learning

Mher Safaryan [1] **Rustem Islamov** [1 2 3] **Xun Qian** [1] **Peter Richtárik** [1]

Abstract

Inspired by recent work of Islamov et al (2021), we propose a family of Federated Newton Learn (FedNL) methods, which we believe is a marked step in the direction of making second-order methods applicable to FL. In contrast to the aforementioned work, FedNL employs a different Hessian learning technique which i) enhances privacy as it does not rely on the training data to be revealed to the coordinating server, ii) makes it applicable beyond generalized linear models, and iii) provably works with general contractive compression operators for compressing the local Hessians, such as Top-K or Rank-R, which are vastly superior in practice. Notably, we do not need to rely on error feedback for our methods to work with contractive compressors. Moreover, we develop FedNL-PP, FedNL-CR and FedNL-LS, which are variants of FedNL that support partial participation, and globalization via cubic regularization and line search, respectively, and FedNL-BC, which is a variant that can further benefit from bidirectional compression of gradients and models, i.e., smart uplink gradient and smart downlink model compression. We prove local convergence rates that are independent of the condition number, the number of training data points, and compression variance. Our communication efficient Hessian learning technique provably learns the Hessian at the optimum. Finally, we perform a variety of numerical experiments that show that our FedNL methods have state-of-the-art communication complexity when compared to key baselines.

[1]King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia [2]Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, Russia [3]Institut Polytechnique de Paris (IP Paris), Palaiseau, France. Correspondence to: Mher Safaryan <mher.safaryan@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

1. Introduction

In this paper we consider the *federated learning* problem

$$\min_{x\in\mathbb{R}^d}\left\{f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right\}, \tag{1}$$

where d denotes dimension of the model $x\in\mathbb{R}^d$ we wish to train, n is the total number of silos/machines/devices/clients in the distributed system, $f_i(x)$ is the loss/risk associated with the data stored on machine $i\in[n]:=\{1,2,\dots,n\}$, and $f(x)$ is the empirical loss/risk.

1.1. First-order methods for FL

The prevalent paradigm for training federated learning (FL) models (Konečný et al., 2016b;a; McMahan et al., 2017) (see also the recent surveys by Kairouz et al (2019); Li et al. (2020a)) is to use *distributed first-order optimization methods* employing one or more tools for enhancing communication efficiency, which is a key bottleneck in the federated setting.

These tools include *communication compression* (Konečný et al., 2016b; Alistarh et al., 2017; Khirirat et al., 2018) and techniques for progressively reducing the variance introduced by compression (Mishchenko et al., 2019; Horváth et al., 2019; Gorbunov et al., 2020a; Li et al., 2020b; Gorbunov et al., 2021a), *local computation* (McMahan et al., 2017; Stich, 2020; Khaled et al., 2020; Mishchenko et al., 2021a) and techniques for reducing the client drift introduced by local computation (Karimireddy et al., 2020; Gorbunov et al., 2021b), and *partial participation* (McMahan et al., 2017; Gower et al., 2019) and techniques for taming the slow-down introduced by partial participation (Gorbunov et al., 2020a; Chen et al., 2020).

Other useful techniques for further reducing the communication complexity of FL methods include the use of *momentum* (Mishchenko et al., 2019; Li et al., 2020b), and *adaptive learning rates* (Malitsky & Mishchenko, 2019; Xie et al., 2019; Reddi et al., 2020; Xie et al., 2019; Mishchenko et al., 2021b). In addition, aspiring FL methods need to protect the *privacy* of the clients' data, and need to be built with *data heterogeneity* in mind (Kairouz et al, 2019).

1.2. Towards second-order methods for FL

While first-order methods are the methods of choice in the context of FL at the moment, their communication complexity necessarily depends on (a suitable notion of) the condition number of the problem, which can be very large as it depends on the structure of the model being trained, on the choice of the loss function, and most importantly, on the properties of the training data.

However, in many situations when algorithm design is not constrained by the stringent requirements characterizing FL, it is very well known that carefully designed *second-order methods* can be vastly superior. On an intuitive level, this is mainly because these methods make an extra computational effort to estimate the local curvature of the loss landscape, which is useful in generating more powerful and adaptive update direction. However, in FL, it is often communication and not computation which forms the key bottleneck, and hence the idea of "going second order" looks attractive. The theoretical benefits of using curvature information are well known. For example, the classical Newton's method, which forms the basis for most efficient second-order method in much the same way the gradient descent method forms the basis for more elaborate first-order methods, enjoys a fast *condition-number-independent* (local) convergence rate (Beck, 2014), which is beyond the reach of *all* first-order methods. However, Newton's method does not admit an efficient distributed implementation in the heterogeneous data regime as it requires repeated communication of local Hessian matrices $\nabla^2 f_i \in \mathbb{R}^{d\times d}$ to the server, which is prohibitive as this constitutes a massive burden on the communication links.

1.3. Desiderata for second-order methods applicable to FL

In this paper we take the stance that it would be highly *desirable* to develop Newton-type methods for solving the federated learning problem (1) that would

[hd] work well in the truly *h*eterogeneous *d*ata setting (i.e., we do not want to assume that the functions $f_1, \ldots, f_n$ are "similar"),

[fs] apply to the general *f*inite-*s*um problem (1), without imparting strong structural assumptions on the local functions $f_1, \ldots, f_n$ (e.g., we do not want to assume that the functions $f_1, \ldots, f_n$ are quadratics, generalized linear models, and so on),

[as] benefits from Newton-like (matrix-valued) *a*daptive *s*tepsizes,

[pe] employ at least a rudimentary *p*rivacy *e*nhancement mechanism (in particular, we do not want the devices to be sending/revealing their training data to the server),

[uc] enjoy, through *u*biased communication *c*ompression strategies applied to the Hessian, such as Rand-K, the same low $\mathcal{O}(d)$ communication cost per communication round as gradient descent,

[cc] be able to benefit from the more aggressive *c*ontractive communication *c*ompression strategies applied to the Hessian, such as Top-K and Rank-R,

[fr] have *f*ast local *r*ates unattainable by first order methods (e.g., rates independent of the condition number),

[pp] support *p*artial *p*articipation (this is important when the number n of devices is very large),

[gg] have *g*lobal convergence *g*uarantees, and superior global empirical behavior, when combined with a suitable globalization strategy (e.g., line search or cubic regularization),

[gc] optionally be able to use, for a more dramatic communication reduction, additional smart uplink (i..e, device to server) *g*radient *c*ompression,

[mc] optionally be able to use, for a more dramatic communication reduction, additional smart downlink (i.e., server to device) *m*odel *c*ompression,

[lc] perform provably useful *l*ocal *c*omputation, even in the heterogeneous data setting (it is known that local computation via gradient-type steps, which form the backbone of methods such as FedAvg and LocalSGD, provably helps under some degree of data similarity only).

However, to the best of our knowledge, existing Newton-type methods are not applicable to FL as they are not compatible with most of the aforementioned desiderata.

It is therefore natural and pertinent to ask whether it is possible to design theoretically well grounded and empirically well performing Newton-type methods that would be able to conform to the FL-specific desiderata listed above.

In this work, we address this challenge *in the affirmative.*

2. Contributions

Before detailing our contributions, it will be very useful to briefly outline the key elements of the recently proposed *Newton Learn* (NL) framework of Islamov et al. (2021), which served as the main inspiration for our work, and which is also the closest work to ours.

2.1. The Newton Learn framework of Islamov et al. (2021)

The starting point of their work is the observation that the Newton-like method

$$x^{k+1} = x^k - (\nabla^2 f(x^*))^{-1} \nabla f(x^k),$$

called Newton Star (NS), where x^* is the (unique) solution of (1), converges to x^* locally quadratically under suitable assumptions, which is a desirable property it inherits from the classical Newton method. Clearly, this method is not practical, as it relies on the knowledge of the Hessian at the

Table 1: Theoretical comparison of 2 gradient-based (Gradient Descent and ADIANA) and 3 second-order (Newton, NL and FedNL) methods. See Section A and the extended Table 5 for more details.

Method	# Communication Rounds	Communication Cost per Communication Round	Communication Complexity
Gradient Descent[1]	$\mathcal{O}(\kappa \log \frac{1}{\epsilon})$	$\mathcal{O}(d)$	$\mathcal{O}(d\kappa \log \frac{1}{\epsilon})$
ADIANA[1] (Li et al., 2020b)	$\mathcal{O}\left(\left(d + \sqrt{\kappa} + \sqrt{\left(\frac{d}{n} + \sqrt{\frac{d}{n}}\right) d\kappa}\right) \log \frac{1}{\epsilon}\right)$	$\mathcal{O}(1)$	$\mathcal{O}\left(\left(d + \sqrt{\kappa} + \sqrt{\left(\frac{d}{n} + \sqrt{\frac{d}{n}}\right) d\kappa}\right) \log \frac{1}{\epsilon}\right)$
Newton	$\mathcal{O}(\log \log \frac{1}{\epsilon})$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2 \log \log \frac{1}{\epsilon})$
NL (Islamov et al., 2021)	$\mathcal{O}\left(\sqrt{\#\text{data}}\sqrt{\log \frac{1}{\epsilon}}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d\sqrt{\#\text{data}}\sqrt{\log \frac{1}{\epsilon}}\right)$
FedNL (this work; (7))	$\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d \log \frac{1}{\epsilon}\right)$
FedNL (this work; (9))	$\mathcal{O}\left(\sqrt{d}\sqrt{\log \frac{1}{\epsilon}}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d\sqrt{d}\sqrt{\log \frac{1}{\epsilon}}\right)$

[1] These methods have global rates. κ is the condition number: $\kappa = \frac{L}{\mu}$ where L is a smoothness constant and μ is the strong convexity constant.

[2] The last column (communication complexity) is the product of the previous two columns and is the key quantity to be compared.

optimum. However, under the assumption that the matrix $\nabla^2 f(x^*)$ is known to the server, NS can be implemented with $\mathcal{O}(d)$ cost in each communication round. Indeed, NS can simply be treated as gradient descent, albeit with a matrix-valued stepsize equal to $(\nabla^2 f(x^*))^{-1}$. The first key contribution of Islamov et al. (2021) is the design of a strategy, for which they coined the term *Newton Learn*, which *learns the Hessians* $\nabla^2 f_1(x^*), \ldots, \nabla^2 f_n(x^*)$, and hence their average, $\nabla^2 f(x^*)$, progressively throughout the iterative process, and does so in a *communication efficient manner,* using unbiased compression *[uc]* of Hessian information. In particular, the compression level can be adjusted so that in each communication round, $\mathcal{O}(d)$ floats need to be communicated between each device and the server only. In each iteration, the master uses the average of the current learned local Hessian matrices in place of the Hessian at the optimum, and subsequently performs a step similar to that of NS. So, their method uses *adaptive matrix-valued stepsizes [as]*. Islamov et al. (2021) prove that their learning procedure indeed works in the sense that the sequences of the learned local matrices converge to the local optimal Hessians $\nabla^2 f_i(x^*)$. This property leads to a Newton-like *acceleration*, and as a result, their NL methods enjoy a local linear convergence rate (for a Lyapunov function that includes Hessian convergence) and local superlinear convergence rate (for distance to the optimum) that is *independent of the condition number*, which is a property beyond the reach of any first-order method *[fr]*. Moreover, all of this provably works in the heterogeneous data setting *[hd]*. Finally, they develop a practical and theoretically grounded globalization strategy *[gg]* based on cubic regularization, called *Cubic Newton Learn* (CNL).

2.2. Issues with the Newton Learn framework

While the above development is clearly very promising in the context of distributed optimization, the results suffer from several limitations which prevent the methods from being applicable to FL. First, the Newton Learn strategy of Islamov et al. (2021) critically depends on the assumption that the local functions are of the form

$$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x), \tag{2}$$

where $\varphi_{ij} : \mathbb{R} \to \mathbb{R}$ are sufficiently well behaved functions, and $a_{i1}, \ldots, a_{im} \in \mathbb{R}^d$ are the training data points owned by device i. As a result, their approach is limited to generalized linear models only, which violates *[fs]* from the aforementioned wish list. Second, their communication strategy critically relies on each device i sending a small subset of their private training data $\{a_{i1}, \ldots, a_{im}\}$ to the server in each communication round, which violates *[pe]*. Further, while their approach supports $\mathcal{O}(d)$ communication, it does not support more general contractive compressors *[cc]*, such as Top-K and Rank-R, which have been found very useful in the context of first order methods with gradient compression. Finally, the methods of Islamov et al. (2021) do not support bidirectional compression *[bc]* of gradients and models, and do not support partial participation *[pp]*.

2.3. Our FedNL framework

We propose a family of five *Federated* Newton Learn methods (Algorithms 1–5), which we believe constitutes a marked step in the direction of making second-order methods applicable to FL.

In contrast to the work of Islamov et al. (2021) (see Table 2), our vanilla method FedNL (Algorithm 1) employs a *different Hessian learning technique*, which makes it applicable beyond generalized linear models (2) to general finite-sum problems *[fs]*, enhances privacy as it does not rely on the training data to be revealed to the coordinating server *[pe]*, and provably works with general contractive compression operators for compressing the local Hessians, such as Top-K or Rank-R, which are vastly superior in practice *[cc]*. Notably, we do not need to rely on error feedback (Seide

Table 2: Comparison of the main features of our family of FedNL algorithms and results with those of Islamov et al. (2021), which we used as an inspiration. We have made numerous and significant modifications and improvements in order to obtain methods applicable to federated learning.

#	Feature	(Islamov et al., 2021)	This Work
[hd]	supports *h*eterogeneous *d*ata setting	✓	✓
[fs]	applies to general *f*inite-*s*um problems	✗	✓
[as]	uses *a*daptive *s*tepsizes	✓	✓
[pe]	*p*rivacy is *e*nhanced (training data is not sent to the server)	✗	✓
[uc]	supports *u*nbiased Hessian *c*ompression (e.g., Rand-K)	✓	✓
[cc]	supports *c*ontractive Hessian *c*ompression (e.g., Top-K)	✗	✓
[fr]	*f*ast local *r*ate: independent of the condition number	✓	✓
[fr]	*f*ast local *r*ate: independent of the # of training data points	✗	✓
[fr]	*f*ast local *r*ate: independent of the compressor variance	✗	✓
[pp]	supports *p*artial *p*articipation	✗	✓(Alg 2)
[gg]	has *g*lobal convergence *g*uarantees via line search	✗	✓(Alg 3)
[gg]	has *g*lobal convergence *g*uarantees via cubic regularization	✓	✓(Alg 4)
[gc]	supports smart uplink *g*radient *c*ompression at the devices	✗	✓(Alg 5)
[mc]	supports smart downlink *m*odel *c*ompression by the master	✗	✓(Alg 5)
[lc]	performs useful *l*ocal *c*omputation	✓	✓

et al., 2014; Stich et al., 2018; Karimireddy et al., 2019; Gorbunov et al., 2020b), which is essential to prevent divergence in first-order methods employing such compressors (Beznosikov et al., 2020), for our methods to work with contractive compressors. We prove that our communication efficient Hessian learning technique provably learns the Hessians at the optimum.

Like Islamov et al. (2021), we prove local convergence rates that are independent of the condition number *[fr]*. However, unlike their rates, some of our rates are also independent of number training data points, and of compression variance *[fr]*. All our complexity results are summarized in Table 3.

Moreover, we show that our approach works in the partial participation *[pp]* regime by developing the FedNL-PP method (Algorithm 2), and devise methods employing globalization strategies: FedNL-LS (Algorithm 3), based on backtracking line search, and FedNL-CR (Algorithm 4), based on cubic regularization *[gg]*. We show through experiments that the former is much more efficient in practice than the latter. Hence, the proposed line search globalization is superior to the cubic regularization approach employed by Islamov et al. (2021). Our approach can further benefit from smart uplink gradient compression *[gc]* and smart downlink model compression *[mc]* – see FedNL-BC (Algorithm 5). Finally, we perform a variety of numerical experiments that show that our FedNL methods have state-of-the-art communication complexity when compared to key baselines.

3. The Vanilla Federated Newton Learn

We start the presentation of our algorithms with the vanilla FedNL method, commenting on the intuitions and technical novelties. The method is formally described[1] in Alg. 1.

3.1. New Hessian learning technique

The first key technical novelty in FedNL is the new mechanism for learning the Hessian $\nabla^2 f(x^*)$ at the (unique) solution x^* in a communication efficient manner. This is achieved by maintaining and progressively updating local Hessian estimates $\mathbf{H}_i^k$ of $\nabla^2 f_i(x^*)$ for all devices $i \in [n]$ and the global Hessian estimate $\mathbf{H}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^k$ of $\nabla^2 f(x^*)$ for the central server. Thus, the goal is to induce $\mathbf{H}_i^k \to \nabla^2 f_i(x^*)$ for all $i \in [n]$, and as a consequence, $\mathbf{H}^k \to \nabla^2 f(x^*)$, throughout the training process.

A *naive* choice for the local estimates $\mathbf{H}_i^k$ would be the exact local Hessians $\nabla^2 f_i(x^k)$, and consequently the global estimate $\mathbf{H}^k$ would be the exact global Hessian $\nabla^2 f(x^k)$. While this naive approach learns the global Hessian at the optimum, it needs to communicate the entire matrices $\nabla^2 f_i(x^k)$ to the server in each iteration, which is extremely costly. Instead, in FedNL we aim to *reuse* past Hessian information and build the next estimate $\mathbf{H}_i^{k+1}$ by updating the current estimate $\mathbf{H}_i^k$. Since all devices have to be synchronized with the server, we also need to make sure the update from $\mathbf{H}_i^k$ to $\mathbf{H}_i^{k+1}$ is easy to communicate. With this intuition in mind, we propose to update the local Hessian estimates via the rule

$$\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k, \text{ where } \mathbf{S}_i^k = \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k),$$

and $\alpha > 0$ is the learning rate. Notice that we reduce the communication cost by explicitly requiring all devices $i \in [n]$ to send compressed matrices $\mathbf{S}_i^k$ to the server only.

[1]For all our methods, we describe the steps constituting a single communication round only. To get an iterative method, one simply needs to repeat provided steps in an iterative fashion.

Table 3: Summary of algorithms proposed and convergence results proved in this paper.

Method	Convergence result †	Convergence type	Convergence rate	Rate independent of the condition # (left) # training data (middle) compressor (right)	Theorem
Newton Zero N0 (Equation (10))	$r_k \leq \frac{1}{2^k} r_0$	local	linear	✓ ✓ ✓	3.6
FedNL (Algorithm 1)	$r_k \leq \frac{1}{2^k} r_0$	local	linear	✓ ✓ ✓	3.6
	$\Phi_1^k \leq \theta^k \Phi_1^0$	local	linear	✓ ✓ ✗	3.6
	$r_{k+1} \leq c\theta^k r_k$	local	superlinear	✓ ✓ ✗	3.6
Partial Participation FedNL-PP (Algorithm 2)	$\mathcal{W}^k \leq \theta^k \mathcal{W}^0$	local	linear	✓ ✓ ✓	D.1
	$\Phi_2^k \leq \theta^k \Phi_2^0$	local	linear	✓ ✓ ✗	D.1
	$r_{k+1} \leq c\theta^k \mathcal{W}_k$	local	linear	✓ ✓ ✗	D.1
Line Search FedNL-LS (Algorithm 3)	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	E.1
Cubic Regularization FedNL-CR (Algorithm 4)	$\Delta_k \leq c/k$	global	sublinear	✗ ✓ ✓	F.1
	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	F.1
	$\Phi_1^k \leq \theta^k \Phi_1^0$	local	linear	✓ ✓ ✗	F.1
	$r_{k+1} \leq c\theta^k r_k$	local	superlinear	✓ ✓ ✗	F.1
Bidirectional Compression FedNL-BC (Algorithm 5)	$\Phi_3^k \leq \theta^k \Phi_3^0$	local	linear	✓ ✓ ✗	G.4
Newton Star NS (Equation (58))	$r_{k+1} \leq c r_k^2$	local	quadratic	✓ ✓ ✓	H.1

Quantities for which we prove convergence: (i) distance to solution $r_k := \|x^k - x^*\|^2$; $\mathcal{W}^k := \frac{1}{n}\sum_{i=1}^n \|w_i^k - x^*\|^2$ (ii) Lyapunov functions $\Phi_1^k := c\|x^k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|^2_{\rm F}$; $\Phi_2^k := c\mathcal{W}^k + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|^2_{\rm F}$; $\Phi_3^k := \|z^k - x^*\|^2 + c\|w^k - x^*\|^2$. (iii) Function value suboptimality $\Delta_k := f(x^k) - f(x^*)$

† constants $c > 0$ and $\theta \in (0, 1)$ are possibly different each time they appear. Refer to the precise statements of the theorems for the exact values.

Algorithm 1 FedNL (Federated Newton Learn)

Parameters: Hessian learning rate $\alpha \geq 0$; compression operators $\{\mathcal{C}_1^k, \dots, \mathcal{C}_n^k\}$
Initialization: $x^0 \in \mathbb{R}^d$; $\mathbf{H}_1^0, \dots, \mathbf{H}_n^0 \in \mathbb{R}^{d \times d}$ and $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$
for each device $i = 1, \dots, n$ in parallel **do**
  Get x^k from the server and compute local gradient $\nabla f_i(x^k)$ and local Hessian $\nabla^2 f_i(x^k)$
  Send $\nabla f_i(x^k)$, $\mathbf{S}_i^k := \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$ and $l_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^k)\|_{\rm F}$ to the server
  Update local Hessian shift to $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$
end for
on server
  Get $\nabla f_i(x^k)$, $\mathbf{S}_i^k$ and l_i^k from each node $i \in [n]$
  $\mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k$, $l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$, $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$
  Option 1: $x^{k+1} = x^k - \left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$
  Option 2: $x^{k+1} = x^k - \left[\mathbf{H}^k + l^k \mathbf{I}\right]^{-1} \nabla f(x^k)$

The Hessian learning technique employed in the Newton Learn framework of (Islamov et al., 2021) is critically different to ours as it heavily depends on the structure (2) of the local functions. Indeed, the local optimal Hessians

$$\nabla^2 f_i(x^*) = \frac{1}{m}\sum_{j=1}^m \varphi_{ij}''(a_{ij}^\top x^*) a_{ij} a_{ij}^\top$$

are learned via the proxy of learning the optimal scalars $\varphi_{ij}''(a_{ij}^\top x^*)$ for all local data points $j \in [m]$, which also requires the transmission of the active data points a_{ij} to the server in each iteration. This makes their method inapplicable to the general finite sum problems *[fs]*, and incapable of securing even the most rudimentary privacy enhancement *[pe]* mechanism.

We do not make any structural assumption on the problem (1), and rely on the following general conditions to prove effectiveness of our Hessian learning technique:

Assumption 3.1. The average loss f is μ-strongly convex, and all local losses $f_i(x)$ have Lipschitz continuous Hessians. Let L_*, $L_{\rm F}$ and L_∞ be the Lipschitz constants with respect to three different matrix norms: spectral, Frobenius and infinity norms, respectively. Formally, we require

$$\begin{aligned} \|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| &\leq L_*\|x - y\| \\ \|\nabla^2 f_i(x) - \nabla^2 f_i(y)\|_{\rm F} &\leq L_{\rm F}\|x - y\| \\ \max_{j,l} |(\nabla^2 f_i(x) - \nabla^2 f_i(y))_{jl}| &\leq L_\infty\|x - y\| \end{aligned}$$

to hold for all $i \in [n]$ and $x, y \in \mathbb{R}^d$.

3.2. Compressing matrices

In the literature on first-order compressed methods, compression operators are typically applied to vectors (e.g., gradients, gradient differences, models). As our approach is based on second-order information, we apply compression operators to $d \times d$ matrices of the form $\nabla^2 f_i(x^k) - \mathbf{H}_i^k$ instead. For this reason, we adapt two popular classes of compression operators used in first-order methods to act on $d \times d$ matrices by treating them as vectors of dimension d^2.

Definition 3.2 (Unbiased Compressors)**.** By $\mathbb{B}(\omega)$ we denote the class of (possibly randomized) unbiased compression operators $\mathcal{C}\colon \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ with variance parameter $\omega \geq 0$ satisfying

$$\mathbb{E}\left[\mathcal{C}(\mathbf{M})\right] = \mathbf{M},\; \mathbb{E}\left[\|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_{\rm F}^2\right] \leq \omega\|\mathbf{M}\|_{\rm F}^2 \quad (3)$$

for all matrices $\mathbf{M} \in \mathbb{R}^{d\times d}$.

Common choices of unbiased compressors are random sparsification and quantization (see Appendix).

Definition 3.3 (Contractive Compressors)**.** By $\mathbb{C}(\delta)$ we denote the class of deterministic contractive compression operators $\mathcal{C}\colon \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ with contraction parameter $\delta \in [0,1]$ satisfying

$$\|\mathcal{C}(\mathbf{M})\|_{\rm F} \leq \|\mathbf{M}\|_{\rm F},\; \|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_{\rm F}^2 \leq (1-\delta)\|\mathbf{M}\|_{\rm F}^2 \quad (4)$$

for all matrices $\mathbf{M} \in \mathbb{R}^{d\times d}$.

The first condition of (4) can be easily removed by scaling the operator $\mathcal{C}$ appropriately. Indeed, if for some $\mathbf{M} \in \mathbb{R}^{d\times d}$ we have $\|\mathcal{C}(\mathbf{M})\|_{\rm F} > \|\mathbf{M}\|_{\rm F}$, then we can use the scaled compressor $\widetilde{\mathcal{C}}(\mathbf{M}) := \frac{\|\mathbf{M}\|_{\rm F}}{\|\mathcal{C}(\mathbf{M})\|_{\rm F}}\mathcal{C}(\mathbf{M})$ instead, as this satisfies (4) with the same parameter δ. Common examples of contractive compressors are Top-K and Rank-R operators (see Appendix).

From the theory of first-order methods employing compressed communication, it is known that handling contractive biased compressors is much more challenging than handling unbiased compressors. In particular, a popular mechanism for preventing first-order methods utilizing biased compressors from divergence is the *error feedback* framework. However, contractive compressors often perform much better empirically than their unbiased counterparts. To highlight the strength of our new Hessian learning technique, we develop our theory in a flexible way, and handle both families of compression operators. Surprisingly, we do not need to use error feedback for contractive compressors for our methods to work.

Compression operators are used in (Islamov et al., 2021) in a fundamentally different way. First, their theory supports unbiased compressors only, and does not cover the practically favorable contractive compressors *[cc]*. More importantly, compression is applied within the representation (2) as an operator acting on the space $\mathbb{R}^m$. In contrast to our strategy of using compression operators, this brings the necessity to reveal, in each iteration, the training data $\{a_{i1},\dots,a_{im}\}$ whose corresponding coefficients in (2) are not zeroed out after the compression step *[pe]*. Moreover, when $\mathcal{O}(d)$ communication cost per communication round is achieved, the variance of the compression noise depends on the number of data points m, which then negatively affects the local convergence rates. As the amount of training data can be huge, our convergence rates provide stronger guarantees by not depending on the size of the training dataset *[fr]*.

3.3. Two options for updating the global model

Finally, we offer two options for updating the global model at the server. The first option assumes that the server knows the strong convexity parameter $\mu > 0$ (see Assumption 3.1), and that it is powerful enough to compute the projected Hessian estimate $\left[\mathbf{H}^k\right]_\mu$, i.e., that it is able to project the current global Hessian estimate $\mathbf{H}^k$ onto the set $\left\{\mathbf{M} \in \mathbb{R}^{d\times d}\colon \mathbf{M}^\top = \mathbf{M},\; \mu\mathbf{I} \preceq \mathbf{M}\right\}$ in each iteration (see the Appendix). Alternatively, if μ is unknown, all devices send the compression errors $l_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^k)\|_{\rm F}$ (this extra communication is extremely cheap as all l_i^k variables are floats) to the server, which then computes the corrected Hessian estimate $\mathbf{H}^k + l^k\mathbf{I}$ by adding the average error $l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$ to the global Hessian estimate $\mathbf{H}^k$.

Both options require the server in each iteration to solve a linear system to invert either the projected, or the corrected, global Hessian estimate. The purpose of these options is quite simple: unlike the true Hessian, the compressed local Hessian estimates $\mathbf{H}_i^k$, and also the global Hessian estimate $\mathbf{H}^k$, might not be positive definite, or might even not be of full rank. Further importance of the errors l_i^k will be discussed when we consider extensions of FedNL to partial participation and globalization via cubic regularization.

3.4. Local convergence theory

To provide theoretical guarantees, we need one of the following two assumptions.

Assumption 3.4. $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ for all $i \in [n]$ and $k \geq 0$. Moreover, (i) $\alpha = 1 - \sqrt{1-\delta}$, or (ii) $\alpha = 1$.

Assumption 3.5. $\mathcal{C}_i^k \in \mathbb{B}(\omega)$ for all $i \in [n]$ and $k \geq 0$ and $0 < \alpha \leq \frac{1}{\omega+1}$. Moreover, for all $i \in [n]$ and $j,l \in [d]$, each entry $(\mathbf{H}_i^k)_{jl}$ is a convex combination of $\{(\nabla^2 f_i(x^t))_{jl}\}_{t=0}^k$ for any $k \geq 0$.

To present our results in a unified manner, we define some constants depending on what parameters and which option is used in FedNL. Below, constants A and B depend on the choice of the compressors $\mathcal{C}_i^k$ and the learning rate α, while C and D depend on which option is chosen for the global update.

$$(A,B) \;:=\; \begin{cases} (\alpha^2,\alpha) & \text{if Assumption 3.4(i) holds} \\ (\delta/4, 6/\delta - 7/2) & \text{if Assumption 3.4(ii) holds} \\ (\alpha,\alpha) & \text{if Assumption 3.5 holds} \end{cases} \quad (5)$$

$$(C,D) \;:=\; \begin{cases} (2, L_*^2) & \text{if } \textit{Option 1} \text{ is used} \\ (8, (L_* + 2L_{\rm F})^2) & \text{if } \textit{Option 2} \text{ is used} \end{cases} \quad (6)$$

We prove three local rates for FedNL: for the squared distance to the solution $\|x^k - x^*\|^2$, and for the Lyapunov function

$$\Phi^k := \mathcal{H}^k + 6BL_{\rm F}^2\|x^k - x^*\|^2,$$

where

$$\mathcal{H}^k := \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2.$$

Theorem 3.6. *Let Assumption 3.1 hold. Assume $\|x^0 - x^*\| \leq \frac{\mu}{\sqrt{2D}}$ and $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for all $k \geq 0$. Then, FedNL (Algorithm 1) converges linearly with the rate*

$$\|x^k - x^*\|^2 \leq \frac{1}{2^k}\|x^0 - x^*\|^2. \tag{7}$$

Moreover, depending on the choice (5) of the compressors $\mathcal{C}_i^k$ (Assumption 3.4 or 3.5), learning rate α, and which option is used for global model updates, we have the following linear and superlinear rates:

$$\mathbb{E}[\Phi^k] \leq \left(1 - \min\left\{A, \frac{1}{3}\right\}\right)^k \Phi^0, \tag{8}$$

$$\mathbb{E}\left[\frac{\|x^{k+1}-x^*\|^2}{\|x^k - x^*\|^2}\right] \leq \left(1-\min\{A,\frac{1}{3}\}\right)^k \left(C + \frac{D}{12BL_{\rm F}^2}\right)\frac{\Phi^0}{\mu^2}. \tag{9}$$

Let us comment on these rates. First, the local linear rate (7) with respect to iterates is based on a universal constant, i.e., it does not depend on the condition number of the problem, the size of the training data, or the dimension of the problem. Indeed, the squared distance to the optimum is halved in each iteration. Second, we have linear rate (8) for the Lyapunov function Φ^k, which implies the linear convergence of all local Hessian estimates $\mathbf{H}_i^k$ to the local optimal Hessians $\nabla^2 f_i(x^*)$. Thus, our initial goal to progressively learn the local optimal Hessians in a communication efficient manner is achieved, justifying the effectiveness of the new Hessian learning technique. Finally, our Hessian learning process accelerates the convergence of iterates to a superlinear rate (9). Both rates (8) and (9) are independent of the condition number of the problem, or the number of data points. However, they do depend on the compression variance (since A depends on δ or ω), which, in case of $\mathcal{O}(d)$ communication constraints, depend on the dimension d only.

For clarity of exposition, in Theorem 3.6 we assumed $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for all iterations $k \geq 0$. Below, we prove that this inequality holds, using the initial conditions only.

Lemma 3.7. *Let Assumption 3.4 hold, and assume $\|x^0 - x^*\| \leq e_1 := \min\{\frac{\mu}{2L_{\rm F}}\sqrt{\frac{A}{BC}}, \frac{\mu}{\sqrt{2D}}\}$ and $\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F} \leq \frac{\mu}{2\sqrt{C}}$. Then $\|x^k - x^*\| \leq e_1$ and $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F} \leq \frac{\mu}{2\sqrt{C}}$ for all $k \geq 0$.*

Lemma 3.8. *Let Assumption 3.5 hold, and assume $\|x^0 - x^*\| \leq e_2 := \frac{\mu}{\sqrt{D+4Cd^2L_\infty^2}}$. Then $\|x^k - x^*\| \leq e_2$ and $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for all $k \geq 0$.*

3.5. FedNL and the "Newton Triangle"

One implication of Theorem 3.6 is that the local rate $\frac{1}{2^k}$ (see (7)) holds even when we specialize FedNL to $\mathcal{C}_i^k \equiv \mathbf{0}$, $\alpha = 0$ and $\mathbf{H}_i^0 = \nabla^2 f_i(x^0)$ for all $i \in [n]$. These parameter choices give rise to the following simple method, which we call Newton Zero (N0):

$$x^{k+1} = x^k - \left[\nabla^2 f(x^0)\right]^{-1}\nabla f(x^k), \quad k \geq 0. \tag{10}$$

Interestingly, N0 only needs *initial second-order information*, i.e., Hessian at the zeroth iterate, and the same first-order information as Gradient Descent (GD), i.e., $\nabla f(x^k)$ in each iteration. Moreover, unlike GD, whose rate depends on a condition number, the local rate $\frac{1}{2^k}$ of N0 does not. Besides, FedNL includes NS (when $\mathcal{C}_i^k \equiv \mathbf{0}$, $\alpha = 0$, $\mathbf{H}_i^0 = \nabla^2 f_i(x^*)$) and classical Newton (N) (when $\mathcal{C}_i^k \equiv \mathbf{I}$, $\alpha = 1$, $\mathbf{H}_i^0 = \mathbf{0}$) as special cases.

It can be helpful to visualize the three special Newton-type methods—N, NS and N0 —as the vertices of a triangle capturing a subset of two of these three requirements: 1) $\mathcal{O}(d)$ communication cost per round, 2) implementability in practice, and 3) local quadratic rate. Indeed, each of these three methods satisfies *two* of these requirements only: N (2+3), NS (1+3) and N0 (1+2). Finally, FedNL interpolates between these requirements. See Figure 14.

4. Extensions to FedNL

Here we briefly describe four extensions to FedNL and the key technical contributions. Detailed sections for each extension are deferred to the Appendix.

4.1. Partial Participation

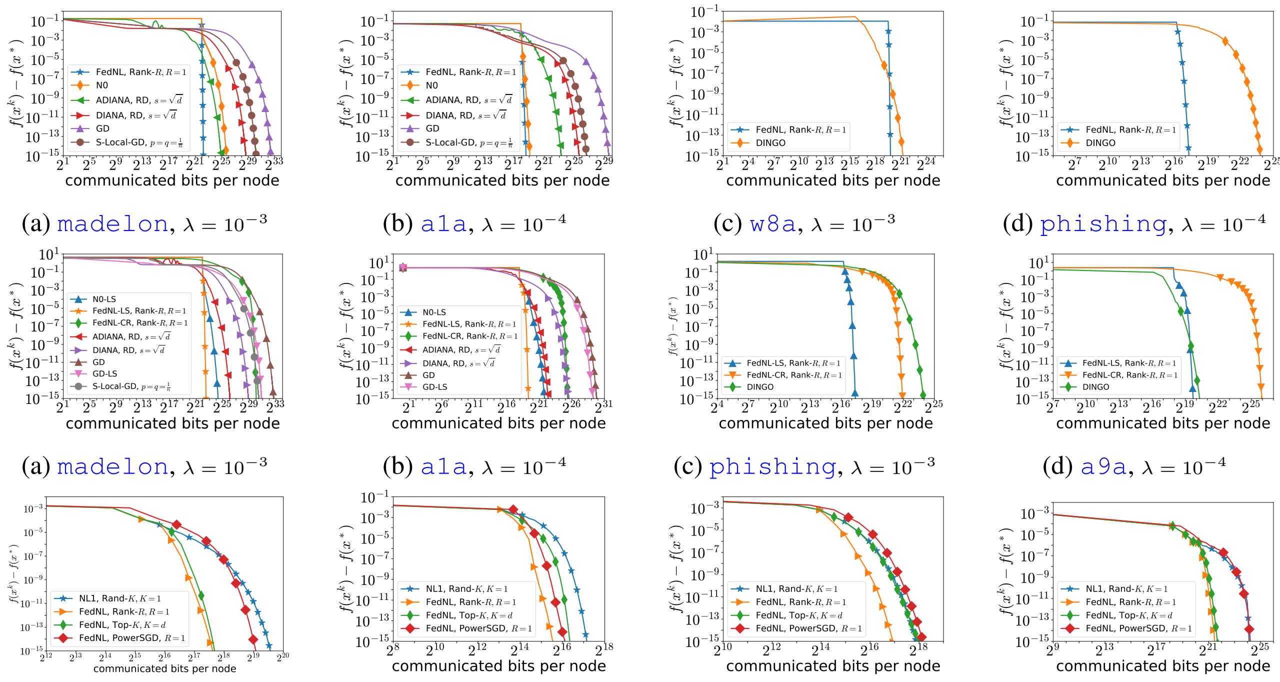
In FedNL-PP (Algorithm 2), the server selects a subset $S^k \subseteq [n]$ of τ devices, uniformly at random, to participate in each iteration. As devices might be inactive for several iterations, the same local gradient and local Hessian used in FedNL does not provide convergence in this case. To guarantee convergence, devices need to compute *Hessian corrected local gradients*

$$g_i^k = (\mathbf{H}_i^k + l_i^k\mathbf{I})w_i^k - \nabla f_i(w_i^k),$$

where w_i^k is the last global model that device i received from the server. This is an innovation which also requires a different analysis.

4.2. Globalization via Line Search

Our first globalization strategy, FedNL-LS (Algorithm 3), which performs *significantly better in practice* than FedNL-CR (described next), is based on a backtracking line search procedure. The idea is to fix the search direction



(a) madelon, $\lambda = 10^{-3}$ (b) a1a, $\lambda = 10^{-4}$ (c) w8a, $\lambda = 10^{-3}$ (d) phishing, $\lambda = 10^{-4}$

(a) madelon, $\lambda = 10^{-3}$ (b) a1a, $\lambda = 10^{-4}$ (c) phishing, $\lambda = 10^{-3}$ (d) a9a, $\lambda = 10^{-4}$

(a) w8a, $\lambda = 10^{-3}$ (b) phishing, $\lambda = 10^{-3}$ (c) a1a, $\lambda = 10^{-4}$ (d) w7a, $\lambda = 10^{-4}$

Figure 1: First row: Local comparison of FedNL and N0 with (a), (b) ADIANA, DIANA, GD; with (c), (d) DINGO in terms of communication complexity. **Second row:** Global comparison of FedNL-LS, N0-LS and FedNL-CR with (a), (b) ADIANA, DIANA, GD, and GD with line search; with (c), (d) DINGO in terms of communication complexity. **Third row:** Local comparison of FedNL with 3 types of compression operators and NL1 in terms of communication complexity.

$d^k = -\left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$ by the server and find the smallest integer $s \geq 0$ which leads to a sufficient decrease in the loss

$$f(x^k + \gamma^s d^k) \leq f(x^k) + c\gamma^s \left\langle \nabla f(x^k), d^k \right\rangle$$

with some parameters $c \in (0, 1/2]$ and $\gamma \in (0, 1)$.

4.3. Globalization via Cubic Regularization

Our next globalization strategy, FedNL-CR (Algorithm 4), is to use a cubic regularization term $\frac{L_*}{6}\|h\|^3$, where L_* is the Lipschitz constant for Hessians and h is the direction to the next iterate. However, to get a global upper bound, we had to *correct the global Hessian estimate* $\mathbf{H}^k$ via compression error l^k. Indeed, since $\nabla^2 f(x^k) \preceq \mathbf{H}^k + l^k\mathbf{I}$, we deduce

$$f(x^{k+1}) \leq f(x^k) + \left\langle \nabla f(x^k), h^k \right\rangle + \frac{1}{2}\left\langle (\mathbf{H}^k + l^k\mathbf{I})h^k, h^k \right\rangle + \frac{L_*}{6}\|h^k\|^3 \quad (11)$$

for all $k \geq 0$. This leads to theoretical challenges and necessitates a new analysis.

4.4. Bidirectional Compression

Finally, we modify FedNL to allow for an even more severe level of compression that can't be attained by compressing the Hessians only. This is achieved by compressing the gradients (uplink) and the model (downlink), in a "smart" way. In FedNL-BC (Alg. 5), the server operates its own compressors $\mathcal{C}_{\mathrm{M}}^k$ applied to the model, and uses an additional "smart" *model learning technique* similar to the proposed Hessian learning technique. Besides, all devices compress their local gradients via a Bernoulli compression scheme, which necessitates the use of another "smart" strategy using *Hessian corrected local gradients*

$$g_i^k = \mathbf{H}_i^k(z^k - w^k) + \nabla f_i(w^k),$$

where z^k is the current learned global model and w^k is the last learned global model when local gradients are sent to the server. These changes are substantial and require novel analysis.

5. Experiments

We carry out numerical experiments to study the performance of FedNL, and compare it with various state-of-the-art methods in federated learning. We consider the problem

(1) with local loss functions

$$f_i(x) = \frac{1}{m}\sum_{j=1}^{m}\log\left(1+\exp(-b_{ij}a_{ij}^\top x)\right)+\frac{\lambda}{2}\|x\|^2, \quad (12)$$

where $\{a_{ij}, b_{ij}\}_{j\in[m]}$ are data points at the i-th device and $\lambda > 0$ is a regularization parameter. The datasets were taken from LibSVM library (Chang & Lin, 2011): `a1a`, `a9a`, `w7a`, `w8a`, and `phishing`.

5.1. Parameter setting

In all experiments we use the theoretical parameters for gradient type methods (except those using line search): vanilla gradient descent GD, DIANA (Mishchenko et al., 2019), ADIANA (Li et al., 2020b), and Shifted Local gradient descent, S-Local-GD (Gorbunov et al., 2021b). For DINGO (Crane & Roosta, 2019) we use the authors' choice: $\theta = 10^{-4}, \phi = 10^{-6}, \rho = 10^{-4}$. Backtracking line search for DINGO selects the largest stepsize from $\{1, 2^{-1}, \ldots, 2^{-10}\}$. The initialization of $\mathbf{H}_i^0$ for NL1 (Islamov et al., 2021), FedNL and FedNL-LS is $\nabla^2 f_i(x^0)$, and for FedNL-CR is $\mathbf{0}$. For FedNL, FedNL-LS, and FedNL-CR we use Rank-1 compression operator and stepsize $\alpha = 1$. We use two values of the regularization parameter: $\lambda \in \{10^{-3}, 10^{-4}\}$. In the figures we plot the relation of the optimality gap $f(x^k) - f(x^*)$ and the number of communicated bits per node, or the number of communication rounds. The optimal value $f(x^*)$ is chosen as the function value at the 20-th iterate of standard Newton's method.

5.2. Local convergence

In our first experiment we compare FedNL and N0 with gradient type methods: ADIANA with random dithering (ADIANA, RD, $s = \sqrt{d}$), DIANA with random dithering (DIANA, RD, $s = \sqrt{d}$), Shifted Local gradient descent (S-Local-GD, $p = q = \frac{1}{n}$), vanilla gradient descent (GD), and DINGO. *According to the results summarized in Figure 1 (first row), we conclude that* FedNL *outperforms all gradient type methods and* DINGO*, locally, by many orders in magnitude.* We want to note that we include the communication cost of the initialization for FedNL and N0 in order to make a fair comparison (this is why there is a straight line for these methods initially).

5.3. Global convergence

We now compare FedNL-LS, N0-LS, and FedNL-CR with the first-order methods ADIANA and DIANA with random dithering, Shifted Local gradient descent S-Local-GD, gradient descent (GD), and GD with line search (GD-LS). Besides, we compare FedNL-LS and FedNL-CR with DINGO. In this experiment we choose x^0 far from the solution x^*, i.e., we test the global convergence behavior; see Figure 1 (second row). We observe that FedNL-LS is more communication efficient than all first-order methods and DINGO. However, FedNL-CR is better than GD and GD-LS only. In these experiments we again include the communication cost of initialization for FedNL-LS and N0-LS.

5.4. Comparison with NL1

Next, we compare FedNL with three type of compression operators: Rank-R ($R = 1$), Top-K ($K = d$), and PowerSGD (Vogels et al., 2019) ($R = 1$) against NL1 with the Rand-K ($K = 1$) compressor. The results, presented in Figure 1 (third row), show that FedNL with Rank-1 compressor performs the best.

References

Alimisis, F., Davies, P., and Alistarh, D. Communication-efficient distributed optimization with quantized preconditioners. In *International Conference on Machine Learning (ICML)*, 2021.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Beck, A. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. Society for Industrial and Applied Mathematics, USA, 2014. ISBN 1611973643.

Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

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

Chen, W., Horváth, S., and Richtárik, P. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

Crane, R. and Roosta, F. Dingo: Distributed newton-type method for gradient-norm optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9498–9508, 2019.

Gorbunov, E., Hanzely, F., and Richtárik, P. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *The 23rd International Conference on Artificial Intelligence and Statistics*, 2020a.

Gorbunov, E., Kovalev, D., Makarenko, D., and Richtárik, P. Linearly converging error compensated SGD. In *34th*

Conference on Neural Information Processing Systems (NeurIPS 2020), 2020b.

Gorbunov, E., Burlachenko, K., Li, Z., and Richtárik, P. MARINA: Faster non-convex distributed learning with compression. *arXiv preprint arXiv:2102.07845*, 2021a.

Gorbunov, E., Hanzely, F., and Richtárik, P. Local SGD: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021b.

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. SGD: General analysis and improved rates. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Horváth, S., Kovalev, D., Mishchenko, K., Stich, S., and Richtárik, P. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

Islamov, R., Qian, X., and Richtárik, P. Distributed second order methods with fast rates and compressed communication. *arXiv preprint arXiv:2102.07158*, 2021.

Kairouz et al, P. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes SignSGD and other gradient compression schemes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3252–3261, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. In *International Conference on Machine Learning (ICML)*, 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, 2020.

Khirirat, S., Feyzmahdavian, H. R., and Johansson, M. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

Konečný, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

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

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a. doi: 10.1109/MSP.2020.2975749.

Li, Z., Kovalev, D., Qian, X., and Richtárik, P. Acceleration for compressed gradient descent in distributed and federated optimization. In *International Conference on Machine Learning*, 2020b.

Liu, X., Li, Y., Tang, J., and Yan, M. A double residual compression algorithm for efficient distributed learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Malitsky, Y. and Mishchenko, K. Adaptive gradient descent without descent. In *International Conference on Machine Learning (ICML)*, 2019.

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

Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

Mishchenko, K., Khaled, A., and Richtárik, P. Proximal and federated random reshuffling. *arXiv preprint arXiv:2102.06704*, 2021a.

Mishchenko, K., Wang, B., Kovalev, D., and Richtárik, P. IntSGD: Floatless compression of stochastic gradients. *arXiv preprint arXiv:2102.08374*, 2021b.

Philippenko, C. and Dieuleveut, A. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees. *arXiv preprint arXiv:2006.14591*, 2021.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Reddi, S. J., Konečný, J., Richtárik, P., Póczos, B., and Smola, A. J. AIDE: Fast and communication efficient distributed optimization. *CoRR*, abs/1608.06879, 2016.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Shamir, O., Srebro, N., and Zhang, T. Communication-effcient distributed optimization using an approximate newton-type method. In *Proceedings of the 31th International Conference on Machine Learning*, volume 32, pp. 1000–1008, 2014.

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2020.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Vogels, T., Karimireddy, S. P., and Jaggi, M. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

Wang, S., abd Peng Xu, F. R., and Mahoney, M. W. GIANT: Globally improved approximate Newton method for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

Zhang, J., You, K., and Başar, T. Achieving globally superlinear convergence for distributed optimization with adaptive newton method. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2329–2334, 2020. doi: 10.1109/CDC42340.2020.9304321.

Zhang, Y. and Lin, X. Disco: Distributed optimization for self-concordant empirical loss. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 362–370, Lille, France, 07–09 Jul 2015. PMLR.

Contents

Appendix

A. Theoretical Comparisons with Related Works

In this part, we compare our results with the most relevant prior works in the literature. We start comparing our work with several recently proposed second order distributed optimization methods to the following criterias: problem structure, assumptions on the loss functions, communication complexity (the number of encoding bits sent from client to server in each communication round), theoretical convergence rate and other aspects of the method (such as local computation and privacy). Table 4 below provides the summary.

Table 4: Theoretical comparison of 7 second order methods (including ours). Advantages are written in green, while limitations are colored in red.

Method	Problem	Assumptions	Communication Cost per Communication Round	Rate	Comments
GIANT (Wang et al., 2018)	GLM[2]	LipC[1] Hessian, convex + l_2 reg., $\approx$ i.i.d. data	$\mathcal{O}(d)$	Local κ-dependent linear. Global $\mathcal{O}(\log {}^{\kappa}/_{\epsilon})$, quadratics	Big data regime (#data $\gg d$)
DINGO (Crane & Roosta, 2019)	GFS[3]	Moral Smoothness[4], $\approx$ strong convexity[5]	$\mathcal{O}(d)$	Global linear rate. No fast local rate.	Operates full gradients, Hessian-vector products, Hessian pseudo-inverse and vector products.
DAN (Zhang et al., 2020)	GFS	LipC Hessian, strong convexity	$\mathcal{O}(nd^2)$	Global quadratic rate after $\mathcal{O}({}^{L}/_{\mu^2})$ iterations.	Operates full gradients and Hessian matrices.
DAN-LA (Zhang et al., 2020)	GFS	LipC Hessian, LipC gradient, strong convexity	$\mathcal{O}(nd)$	Asymptotic and implicit global superlinear rate.	$\lim_{k\to\infty} \frac{\\|x_{k+1}-x^*\\|}{\\|x_k-x^*\\|} = 0$ Independent of κ ? Better non-asymptotic complexity over linear rate ?
NL (Islamov et al., 2021)	GLM	LipC Hessian, convex + l_2 reg.	$\mathcal{O}(d)$	Local superlinear rate independent of κ, but dependent on #data. Global linear rate.	reveals local data to server
Quantized Newton (Alimisis et al., 2021)	GFS	LipC Hessian, LipC gradient, strong convexity[5]	$\tilde{\mathcal{O}}(d^2)$	Local (fixed) linear rate. No global rate.	Operates full gradients and Hessian matrices.
FedNL (this work)	GFS	LipC Hessian, strong convexity	$\mathcal{O}(d)$	Local (fixed) linear rate. Local superlinear rate independent of κ, independent of #data. Global linear rate.	Operates full gradients and Hessian matrices. Supports contractive Hessian compression. Extensions[†]

[1] **LipC** = **Lip**schitz **C**ontinuous.
[2] GLM = Generalized Linear Model, e.g. $loss_j(x; a_j) = \phi_j(a_j^\top x) + \lambda\|x\|^2$. [3] GFS = General Finite Sum.
[4] Moral Smoothness: $\|\nabla^2 f(x)\nabla f(x) - \nabla^2 f(y)\nabla f(y)\| \leq L\|x-y\|$. [5] Applies to local loss functions for all clients.
[†] Partial Participation, Globalization (via Line Search and Cubic Regularization) and Bidirectional Compression.

As we can see from the table, in contrast to FedNL, the other methods suffer at least one of the following issues:

- Theoretical analysis does not cover general finite sum problems (GIANT and NL).
- Communication cost per client/iteration is high (DAN and Quantized Newton).
- Convergence rate either depends on condition number (GIANT and DINGO) or the number of data points (NL) or is not explicit/clear (DAN-LA).
- Privacy is broken by directly revealing local training data (NL).

We do not compare with algorithms DANE (Shamir et al., 2014) and its accelerated variant AIDE (Reddi et al., 2016) since they are first-order methods. This means that convergence rates depend on the conditioning of the problem and hence are worse than what we prove for FedNL. Moreover, DANE does not work well for heterogeneous datasets - the analysis and experimental evidence of DANE only shows benefits in a sufficiently homogeneous data regime. On the other hand, our concern is the heterogeneous data regime typical to FL. We also omit DiSCO (Zhang & Lin, 2015) from our empirical study because the problem setup is restricted to homogeneous data distribution regime, generalized linear models and convergence rates depend on the conditioning of the problem. Furthermore, the authors of DINGO experimentally showed that DINGO outperforms methods like DiSCO and GIANT, and this is why we focused on comparing to DINGO.

Next, we compare several first and second order methods based on their communication complexity, defined as the total number of bits sent from a client to the server to achieve some prescribed accuracy ϵ. For this purpose, we use sparsification as an example of a compressor in most cases. For all methods supporting sparsification, we have used the sparsification, which reduces the number of communicated floats by the factor of d compared to the non-compressed variant of the method. That is, for gradient based methods DCGD, DIANA and ADIANA, we have used the Rand-1 sparsifier, which compresses $\mathcal{O}(d)$ gradient to $\mathcal{O}(1)$.

To transform the local superlinear convergence rate (9) of FedNL into an iteration complexity, we proceed as follows. Let $r_{k+1} \leq C(1-\rho)^k r_k$, where $r_k = \|x^k - x^*\|^2$, $\rho \in (0,1)$ and $C > 0$ is some constant. Note that if $k \geq \frac{2}{\rho}\log\frac{1}{C}$, then $(1-\rho)^{k/2}C \leq 1$. Hence, after $\mathcal{O}(\frac{1}{\rho})$ iteration we have $r_{k+1} \leq (1-\rho)^{k/2} r_k$. Unraveling the recursion we get

$$r_k \leq (1-\rho)^{\frac{k-1}{2}}(1-\rho)^{\frac{k-2}{2}}\dots(1-\rho)^{\frac{1}{2}}r_0 = (1-\rho)^{\frac{k(k-1)}{2}}r_0.$$

Therefore, FedNL needs $\mathcal{O}\left(\sqrt{\frac{1}{\rho}\log\frac{1}{\epsilon}}\right)$ number of iterations to achieve ϵ-accuracy. For FedNL, we used step-size $\alpha = 1$ (see Assumption 3.4(ii) and also (5)) and matrix sparsification described in Appendix B.3.3, which compresses $\mathcal{O}(d^2)$ Hessian down to $\mathcal{O}(d)$ (i.e., $\delta = \frac{1}{d}$). With this choice we get $\frac{1}{\rho} = \mathcal{O}(d)$ and the iteration of FedNL becomes $\mathcal{O}\left(\sqrt{d\log\frac{1}{\epsilon}}\right)$. For Newton Learn (NL), $\frac{1}{\rho} = \mathcal{O}(\#\text{data})$, where #data is the number of data points in each device. DAN and Quantized Newton use their own bespoke ways of compressing communication. Table 5 provides the details, from which we make the following observations:

Table 5: Theoretical comparison of 3 gradient-based and 5 second-order methods. The last column (communication complexity) is the product of the previous two columns and is the key quantity to be compared.

Method	# Communication Rounds	Comm. Cost per Comm. Round	Communication Complexity
Gradient Descent[1]	$\mathcal{O}(\kappa\log\frac{1}{\epsilon})$	$\mathcal{O}(d)$	$\mathcal{O}(d\kappa\log\frac{1}{\epsilon})$
DCGD[1] (Khirirat et al., 2018)	$\mathcal{O}\left(\frac{d\sigma^*}{n\mu^2}\frac{1}{\epsilon}\log\frac{1}{\epsilon}\right)$	$\mathcal{O}(1)$	$\mathcal{O}\left(\frac{d\sigma^*}{n\mu^2}\frac{1}{\epsilon}\log\frac{1}{\epsilon}\right)$
DIANA[1] (Mishchenko et al., 2019)	$\mathcal{O}\left((d+\kappa+\kappa\frac{d}{n})\log\frac{1}{\epsilon}\right)$	$\mathcal{O}(1)$	$\mathcal{O}\left((d+\kappa+\kappa\frac{d}{n})\log\frac{1}{\epsilon}\right)$
ADIANA[1] (Li et al., 2020b)	$\mathcal{O}\left(\left(d+\sqrt{\kappa}+\sqrt{\left(\frac{d}{n}+\sqrt{\frac{d}{n}}\right)d\kappa}\right)\log\frac{1}{\epsilon}\right)$	$\mathcal{O}(1)$	$\mathcal{O}\left(\left(d+\sqrt{\kappa}+\sqrt{\left(\frac{d}{n}+\sqrt{\frac{d}{n}}\right)d\kappa}\right)\log\frac{1}{\epsilon}\right)$
Newton	$\mathcal{O}(\log\log\frac{1}{\epsilon})$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2\log\log\frac{1}{\epsilon})$
DAN[1] (Zhang et al., 2020)	$\mathcal{O}\left(\frac{L_*}{\mu^2}+\log\log\frac{1}{\epsilon}\right)$	$\mathcal{O}(d^2)$	$\mathcal{O}\left(d^2\left(\frac{L_*}{\mu^2}+\log\log\frac{1}{\epsilon}\right)\right)$
Quantized Newton (Alimisis et al., 2021)	$\mathcal{O}(\log\frac{1}{\epsilon})$	$\widetilde{\mathcal{O}}(d^2)$	$\widetilde{\mathcal{O}}(d^2\log\frac{1}{\epsilon})$
NL (Islamov et al., 2021)	$\mathcal{O}\left(\sqrt{\#\text{data}}\sqrt{\log\frac{1}{\epsilon}}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d\sqrt{\#\text{data}}\sqrt{\log\frac{1}{\epsilon}}\right)$
FedNL (this work; (7))	$\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d\log\frac{1}{\epsilon}\right)$
FedNL (this work; (9))	$\mathcal{O}\left(\sqrt{d}\sqrt{\log\frac{1}{\epsilon}}\right)$	$\mathcal{O}(d)$	$\mathcal{O}\left(d\sqrt{d}\sqrt{\log\frac{1}{\epsilon}}\right)$

[1] These methods have global rates. [2] DCGD, DIANA and ADIANA are first order methods.
[3] Newton, DAN, Quantized Newton, NL and FedNL are second order methods.
[4] κ is the condition number: $\kappa = \frac{L}{\mu}$ where L is a smoothness constant and μ is the strong convexity constant.

- **FedNL achieves better communication complexity than Newton whenever $d > \frac{\log\frac{1}{\epsilon}}{\left(\log\log\frac{1}{\epsilon}\right)^2}$.** For example, if we set $\epsilon = 10^{-10}$, then this requirement means $d > 10$, and hence is not restrictive. So, virtually in all situations of practical interest, FedNL is better than Newton. The improvement is more pronounced with larger d, and is approximately of the size $\mathcal{O}(\sqrt{d})$. So, for $d = 10^6$, for example, FedNL finds the solution using approximately 1000 times less communicated bits than Newton.
- **FedNL achieves better communication complexity than Gradient Descent whenever $\kappa > \frac{\sqrt{d}}{\sqrt{\log\frac{1}{\epsilon}}}$.** So, FedNL is better when the condition number κ is large enough. This is expected, since FedNL complexity does not depend on the condition number. The advantage of FedNL grows if d or ϵ are smaller.
- ADIANA is known to have the state of the art complexity (in the strongly convex regime) among all first order method, and hence we do not need to compare FedNL to DCGD and DIANA, which are both inferior to ADIANA. It is clear that FedNL can beat ADIANA as well since the complexity of ADIANA depends on κ. **So, for large enough κ, FedNL is better than ADIANA. For example, a simple sufficient condition for this to happen is to require $\kappa > d^3$** (this

can be refined, but the expression will become uglier). Likewise, FedNL has square root dependence on $\log \frac{1}{\epsilon}$, and hence it becomes better than ADIANA if ϵ is sufficiently small (and other terms are kept constant).

- Neither Quantized Newton nor DAN improve on Newton in communication complexity (but may be better in practice). We already explained that FedNL improves on Newton.

- We do not include GIANT in the table since GIANT does not work in the heterogeneous data regime, which is critical to FL and our paper. We do not include DINGO in the table since its rate depends on various iterate-dependent assumptions which make the analysis convoluted. It is not clear that such assumptions can actually be satisfied. Their rates are not explicit - it is not possible to compare to them.

It worths noting that our work is not just about communication complexity. In fact, our contributions go far beyond this, and we make it clear in the paper. Our work is the first serious attempt to make second order methods applicable to federated learning in the sense that we address many issues which previously made second order methods inapplicable to FL. We support compression of matrices, rudimentary privacy protection (by not revealing data), partial participation, compression of (Hessian corrected) gradients, compression of model (at the master), arbitrary (strongly convex) finite sum problems rather than generalized linear models only, arbitrary contractive compressors, two globalization strategies and more. The best way to judge our contribution to the literature is via comparison to the NewtonLearn work of (Islamov et al., 2021) as their work is the closest work to ours and was the SOTA second order method supporting communication compression before our work. We have made a very detailed comparison to their work, including tables.

B. Extra Experiments

We carry out numerical experiments to study the performance of FedNL, and compare it with various state-of-the-art methods in federated learning. We consider the following problem

$$\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \frac{\lambda}{2}\|x\|^2 \right\}, \quad f_i(x) = \frac{1}{m}\sum_{j=1}^{m} \log\left(1+\exp(-b_{ij}a_{ij}^\top x)\right), \tag{13}$$

where $\{a_{ij}, b_{ij}\}_{j\in[m]}$ are data points at the i-th device.

B.1. Data sets

The datasets were taken from LibSVM library (Chang & Lin, 2011): `a1a`, `a9a`, `w7a`, `w8a`, `phishing`. We partitioned each data set across several nodes to capture a variety of scenarios. See Table 6 for more detailed description of data sets settings.

Table 6: Data sets used in the experiments with the number of worker nodes n used in each case.

Data set	# workers n	# data points ($= nm$)	# features d
`a1a`	16	1600	123
`a9a`	80	32560	123
`w7a`	50	24600	300
`w8a`	142	49700	300
`phishing`	100	110	68
`madelon`	10	2000	500

B.2. Parameters setting

In all experiments we use theoretical parameters for gradient type methods (except those with line search procedure): vanilla gradient descent, DIANA (Mishchenko et al., 2019), ADIANA (Li et al., 2020b), and Shifted Local gradient descent (Gorbunov et al., 2021b). The constants for DINGO (Crane & Roosta, 2019) are set as the authors did: $\theta = 10^{-4}, \phi = 10^{-6}, \rho = 10^{-4}$. Backtracking line search for DINGO selects the largest stepsize from $\{1, 2^{-1}, \ldots, 2^{-10}\}$. The initialization of $\mathbf{H}_i^0$ for NL1 (Islamov et al., 2021), FedNL, FedNL-LS, and FedNL-PP is $\nabla^2 f_i(x^0)$, and for FedNL-CR is $\mathbf{0}$.

We conduct experiments for two values of regularization parameter $\lambda \in \{10^{-3}, 10^{-4}\}$. In the figures we plot the relation of the optimality gap $f(x^k) - f(x^*)$ and the number of communicated bits per node or the number of communication rounds. The optimal value $f(x^*)$ is chosen as the function value at the 20-th iterate of standard Newton's method.

B.3. Compression operators

Here we describe four compression operators that are used in our experiments.

B.3.1. Random dithering for vectors

For first order methods ADIANA and DIANA we use random dithering operator (Alistarh et al., 2017; Horváth et al., 2019). This compressor with s levels is defined via the following formula

$$\mathcal{C}(x) := \mathrm{sign}(x) \cdot \|x\|_q \cdot \frac{\xi_s}{s}, \tag{14}$$

where $\|x\|_q := \left(\sum_i |x_i|^q\right)^{1/q}$ and $\xi_s \in \mathbb{R}^d$ is a random vector with i-th element defind as follows

$$(\xi_s)_i = \begin{cases} l+1 & \text{with probability } \frac{|x_i|}{\|x\|_q}s - l, \\ l & \text{otherwise.} \end{cases} \tag{15}$$

Here $s \in \mathbb{N}_+$ denotes the levels of rounding, and l satisfies $\frac{|x_i|}{\|x\|_q} \in \left[\frac{l}{s}, \frac{l+1}{s}\right]$. According to (Horváth et al., 2019), this compressor has the variance parameter $\omega \le 2 + \frac{d^{1/2}+d^{1/q}}{s}$. However, for standard euclidean norm ($q=2$) one can improve the bound by $\omega \le \min\left\{\frac{d}{s^2}, \frac{\sqrt{d}}{s}\right\}$ (Alistarh et al., 2017).

B.3.2. Rank-R compression operator for matrices

Our theory supports contractive compression operators; see Definition 3.3. In the experiments for FedNL we use Rank-R compression operator. Let $\mathbf{X} \in \mathbb{R}^{d\times d}$ and $\mathbf{U}\Sigma\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{X}$:

$$\mathbf{X} = \sum_{i=1}^d \sigma_i u_i v_i^\top, \tag{16}$$

where the singular values σ_i are sorted in non-increasing order: $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_d$. Then, the Rank-$R$ compressor, for $R \le d$, is defined by

$$\mathcal{C}(\mathbf{X}) := \sum_{i=1}^R \sigma_i u_i v_i^\top. \tag{17}$$

Note that

$$\|\mathbf{X}\|_{\mathrm{F}}^2 \overset{(16)}{=} \left\|\sum_{i=1}^d \sigma_i u_i v_i^\top\right\|_{\mathrm{F}}^2 = \sum_{i=1}^d \sigma_i^2$$

and

$$\|\mathcal{C}(\mathbf{X}) - \mathbf{X}\|_{\mathrm{F}}^2 \overset{(16)+(17)}{=} \left\|\sum_{i=R+1}^d \sigma_i u_i v_i^\top\right\|_{\mathrm{F}}^2 = \sum_{i=R+1}^d \sigma_i^2.$$

Since $\frac{1}{d-R}\sum_{i=R+1}^d \sigma_i^2 \le \frac{1}{d}\sum_{i=1}^d \sigma_i^2$, we have

$$\|\mathcal{C}(\mathbf{X}) - \mathbf{X}\|_{\mathrm{F}}^2 \le \frac{d-R}{d}\|\mathbf{X}\|_{\mathrm{F}}^2 = \left(1 - \frac{R}{d}\right)\|\mathbf{X}\|_{\mathrm{F}}^2,$$

and hence the Rank-R compression operator belongs to $\mathbb{C}(\delta)$ with $\delta = \frac{R}{d}$. In case when $\mathbf{X} \in \mathbb{S}^d$, we have $u_i = v_i$ for all $i \in [d]$, and Rank-R compressor on matrix $\mathbf{X}$ transforms to $\sum_{i=1}^R \sigma_i u_i u_i^\top$, i.e., the output of Rank-R compressor is automatically a symmetric matrix, too.

B.3.3. TOP-K COMPRESSION OPERATOR FOR MATRICES

Another example of contractive compression operators is Top-K compressor for matrices. For arbitrary matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$ let sort its entires in non-increasing order by magnitude, i.e., X_{i_k,j_k} is the k-th maximal element of $\mathbf{X}$ by magnitude. Let's $\{\mathbf{E}_{ij}\}_{i,j=1}^d$ me matrices for which

$$\left(\mathbf{E}_{ij}\right)_{ps} := \begin{cases} 1, & \text{if } (p,s)=(i,j), \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Then, the Top-K compression operator can be defined via

$$\mathcal{C}(\mathbf{X}) := \sum_{k=1}^{K} X_{i_k,j_k} E_{i_k,j_k}. \tag{19}$$

This compression operator belongs to $\mathbb{C}(\delta)$ with $\delta = \frac{K}{d^2}$. If we need to keep the output of Top-K on symmetric matrix $\mathbf{X}$ to be symmetric matrix too, then we apply Top-K compressor only on lower triangular part of $\mathbf{X}$.

B.3.4. RAND-K COMPRESSION OPERATOR FOR MATRICES

Our theory also supports unbiased compression operators; see Definition 3.2. One of the examples is Rand-K. For arbitrary matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$ we choose a set $\mathcal{S}_K$ of indexes (i,j) of cardinality K uniformly at random. Then Rand-K compressor can be defined via

$$\mathcal{C}\left(\mathbf{X}\right)_{ij} := \begin{cases} \frac{d^2}{K} X_{ij} & \text{if } (i,j) \in \mathcal{S}_K, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

This compression operator belongs to $\mathbb{B}(\omega)$ with $\omega = \frac{d^2}{K} - 1$. If we need to make the output of this compressor to be symmetric matrix, then we apply this compressor only on lower triangular part of the input.

B.4. Projection onto the cone of positive definite matrices

If one uses FedNL with Option 1, then we need to project onto the cone of symmetric and positive definite matrices with constant μ:

$$\{\mathbf{M} \in \mathbb{R}^{d\times d} : \mathbf{M}^\top = \mathbf{M},\ \mathbf{M} \succeq \mu \mathbf{I}\}.$$

The projection (with respect to either Frebenious or spectral norm) of symmetric matrix $\mathbf{X}$ onto the cone of positive semidefinite matrices can by computed via

$$\left[\mathbf{X}\right]_0 := \sum_{i=1}^{d} \max\{\lambda_i, 0\} u_i u_i^\top, \tag{21}$$

where $\sum_i \lambda_i u_i u_i^\top$ is an eigenvalue decomposition of $\mathbf{X}$. Using the projection onto the cone of positive semidefinite matrices we can define the projection onto the cone of positive definite matrices with constant μ via

$$\left[\mathbf{X}\right]_\mu := \left[\mathbf{X} - \mu\mathbf{I}\right]_0 + \mu\mathbf{I}. \tag{22}$$

B.5. The effect of compression

First, we investigate how the level of compression influences the performance of FedNL; see Figure 2. Here we study the performance for three types of compression operators: Rank-R, Top-K, and PowerSGD of rank R. According to numerical experiments, the smaller parameter is, the better performance of FedNL is. This statement is true for all three types of compressors.

B.6. Comparison of Options 1 and 2

In our next experiment we investigate which Option (1 or 2) for FedNL with Rank-R and stepsize $\alpha = 1$ compressor demonstrates better results in terms of communication compexity. According to the results in Figure 3, we see that FedNL with projection (Option 1) is more communication effective than that with Option 2. However, Option 1 requires more computing resources.

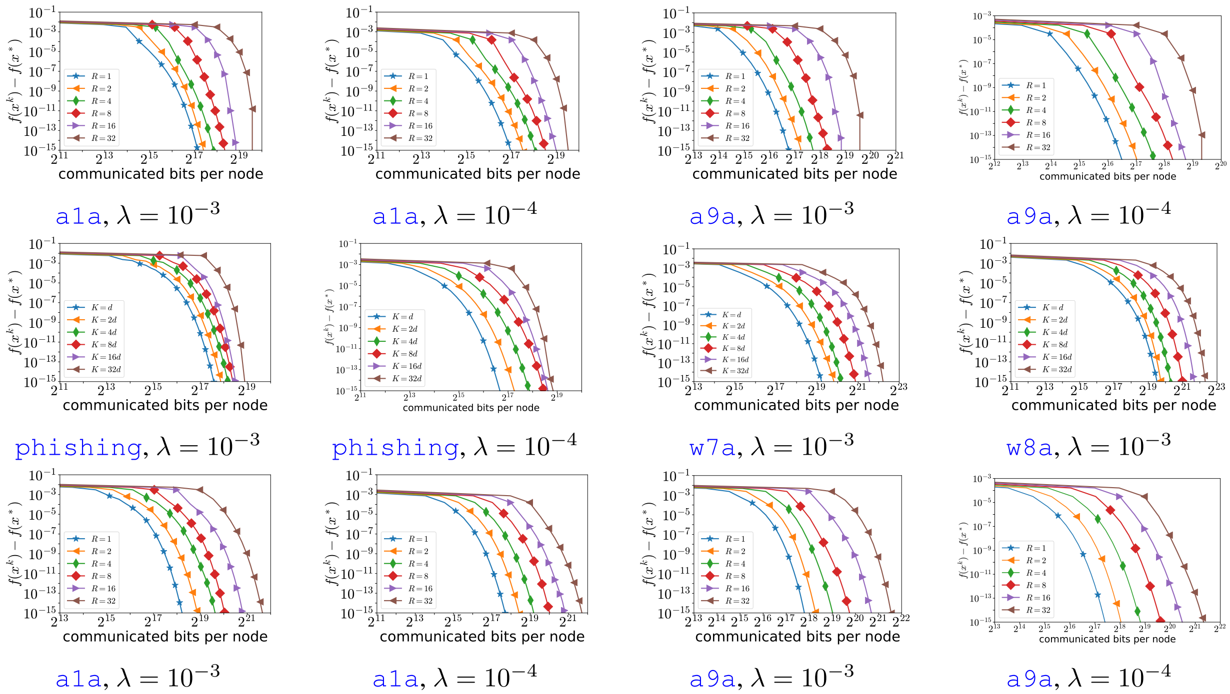


Figure 2: The performance of FedNL with different types of compression operators: Rank-R (first row); Top-K (second row); PowerSGD of rank R (third row) for several values of R and K in terms of communication complexity.

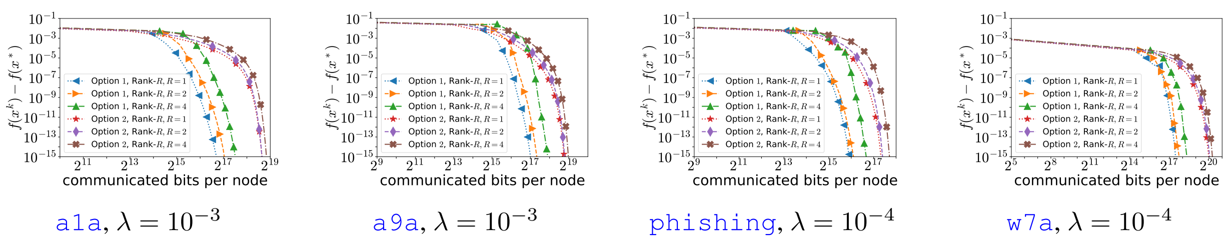


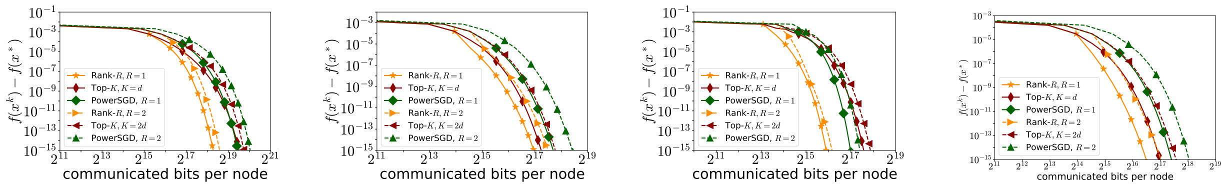
Figure 3: The performance of FedNL with Options 1 and 2 in terms of communication complexity.

B.7. Comparison of different compression operators

Next, we study which compression operator is better in terms of communication complexity. Based on the results in Figure 4, we can conclude that Rank-R is the best compression operator; Top-K and PowerSGD compressors can beat each other in different cases.

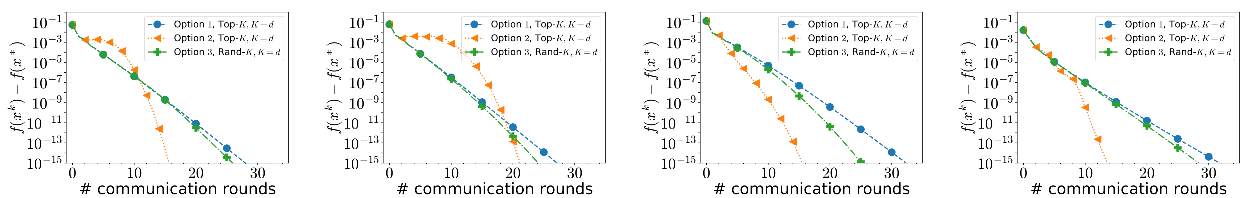
B.8. Comparison of different update rules for Hessians

On the following step we compare FedNL with three update rules for Hessians in order to find the best one. They are biased Top-K compression operator with stepsize $\alpha = 1$ (Option 1); biased Top-K compression operator with stepsize $\alpha = 1 - \sqrt{1-\delta}$; unbiased Rand-$K$ compression operator with stepsize $\alpha = \frac{1}{\omega+1}$. The results of this experiment are presented in Figure 5. Based on them, we can make a conclusion that FedNL with Top-K compressor and stepsize $\alpha = 1$ demonstrates the best performance. FedNL with Rand-K compressor and stepsize $\alpha = \frac{1}{\omega+1}$ performs a little bit better than that with Top-K compressor and stepsize $\alpha = 1 - \sqrt{1-\delta}$. As a consequence, we will use biased compression operator with stepsize $\alpha = 1$ for FedNL in further experiments.



`w8a`, $\lambda = 10^{-3}$ `a1a`, $\lambda = 10^{-4}$ `phishing`, $\lambda = 10^{-3}$ `a9a`, $\lambda = 10^{-4}$

Figure 4: Comparison of the performance of FedNL with different compression operators in terms of communication complexity.



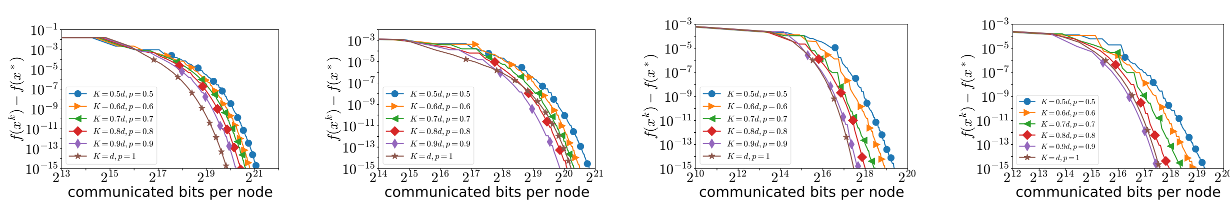
(a) `a1a`, $\lambda = 10^{-3}$ (b) `a9a`, $\lambda = 10^{-3}$ (c) `phishing`, $\lambda = 10^{-3}$ (d) `w7a`, $\lambda = 10^{-3}$

Figure 5: Comparison of FedNL with three update rules: Top-K, $\alpha = 1 - \sqrt{1-\delta}$ (Option 1); Top-K, $\alpha = 1$ (Option 2); Rand-K, $\alpha = \frac{1}{\omega+1}$ (Option 3) in terms of iteration complexity.

B.9. Bidirectional compression

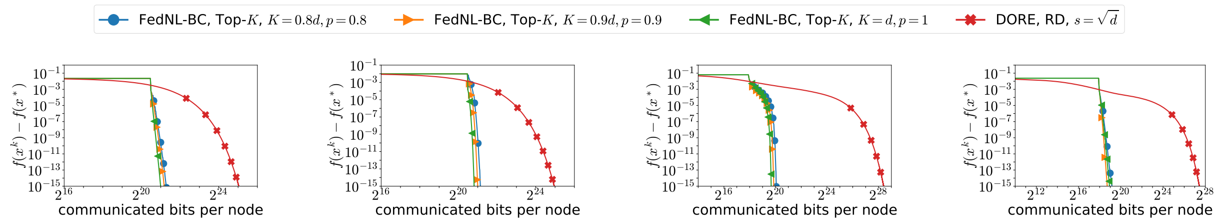
Now we study how the performance of FedNL-BC (with Option 1 and stepsize $\alpha = 1$) is affected by the level of compression in Figure 6. Here we use Top-K compressor for Hessians and models, and broadcast gradients with probability p. In order to make the results more interpretable, we set K to be pd, then we carry out experiments for several values of p. We clearly see that deep compression ($p = 0.5; 0.6$) influences negatively the performance of FedNL-BC. However, small compression ($p = 0.9$) can be beneficial in some cases (see Figure 6: (b), (d)), but this is not the case for Figure 6: (a), (c), where the best performance is demonstrated by FedNL-BC with $p = 1$. We can conclude that only weak compression (the value of p is close to 1) can improve the performance of FedNL-BC, but the improvement is relatively small.

We also compare FedNL-BC (compression was described above, Option 2 was used in the experiments) with DORE method (Liu et al., 2020). This method applies bi-directional compression on gradients (uplink compression) and models (downlink compression). All constants for this method were chosen according theoretical results in the paper. We use random dithering compressor in both directions ($s = \sqrt{d}$). Based on the numerical experiments in Figure 7, we can conclude that FedNL-BC is much more communication efficient method than DORE by many orders in magnitude.



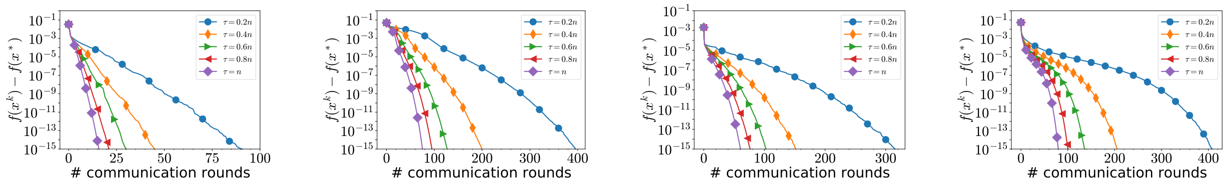
(a) `w7a`, $\lambda = 10^{-3}$ (b) `w8a`, $\lambda = 10^{-3}$ (c) `a1a`, $\lambda = 10^{-4}$ (d) `a9a`, $\lambda = 10^{-4}$

Figure 6: The performance of FedNL-BC with Top-K applied to Hessians and models ($K = pd$), and broadcasting gradients with probability p for several values of p in terms of communication complexity.



(a) w7a, $\lambda = 10^{-3}$ (b) w8a, $\lambda = 10^{-3}$ (c) a1a, $\lambda = 10^{-4}$ (d) a9a, $\lambda = 10^{-4}$

Figure 7: Comparison of FedNL-BC with Top-K applied to Hessians and models ($K = pd$), and broadcasting gradients with probability p and DORE in terms of communication complexity.



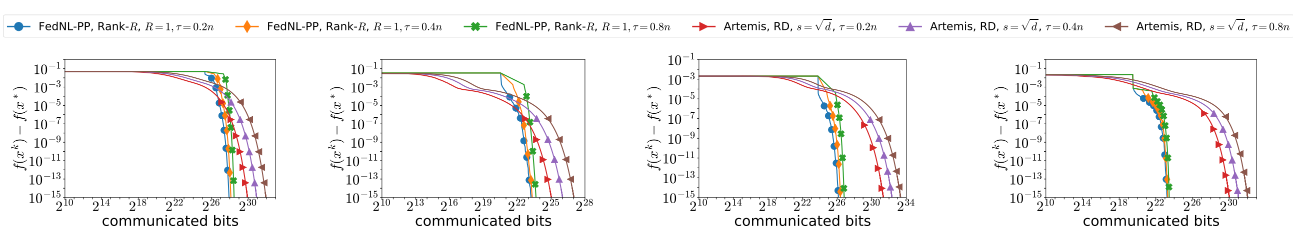
(a) phishing, $\lambda = 10^{-3}$ (b) w8a, $\lambda = 10^{-3}$ (c) w7a, $\lambda = 10^{-4}$ (d) a9a, $\lambda = 10^{-4}$

Figure 8: The performance of FedNL-PP with Rank-1 compressor in terms of iteration complexity.

B.10. The performance of FedNL-PP

Now we deploy our FedNL-PP method in order to study how the performance is inlfuenced by the value of active nodes τ. We use FedNL-PP with Rank-1 compression operator, and run the method for several values of τ; see Figure 8. As we can see, the smaller value of τ is, the worse performance of FedNL-PP is, as it expected.

Now we compare FedNL-PP with Artemis (Philippenko & Dieuleveut, 2021) which supports partial participation too. We use random sparsification compressor ($s = \sqrt{d}$) in uplink direction, and the server broadcasts descent direction to each node without compression. All contstants of the method were chosen according theory from the paper. Each node i computes full local gradient $\nabla f_i(x^k)$. We conduct experiments for several number of active nodes: $\tau \in \{0.2n, 0.4n, 0.8n\}$, then we calculate the total number of transmitted bits received by the server from all active nodes. All results are presented in Figure 9. We clearly see that FedNL-PP outperforms Artemis by several orders in magnitude in terms of communication complexity.



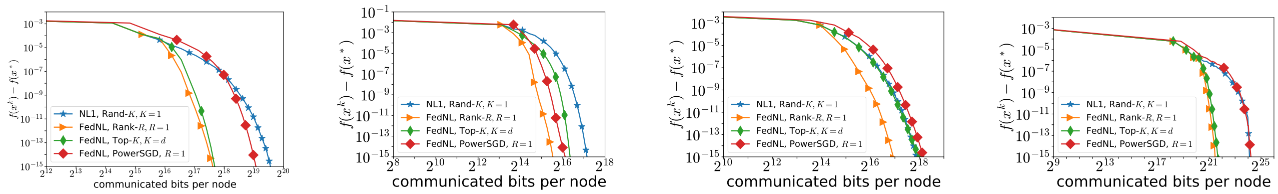
(a) w8a, $\lambda = 10^{-3}$ (b) phishing, $\lambda = 10^{-3}$ (c) w7a, $\lambda = 10^{-4}$ (d) a1a, $\lambda = 10^{-4}$

Figure 9: Comparison of FedNL-PP with Artemis in terms of communication complexity for several values of active nodes τ.

B.11. Comparison with NL1

In our next experiment we compare FedNL with three types of compression operators (Rank-R, Top-K, PowerSGD) and NL1. As we can see in Figure 10, FedNL with Rank-1 are more communication efficient method in all cases. FedNL with

Top-d and PowerSGD of rank 1 compressors performs better or the same as NL1 in almost all cases, except Figure 10: (c), where FedNL with PowerSGD demonstrates a little bit worse results than NL1. Based on these experiments, we can conclude that new compression mechanism for Hessians is more effective than that was introduced in (Islamov et al., 2021).

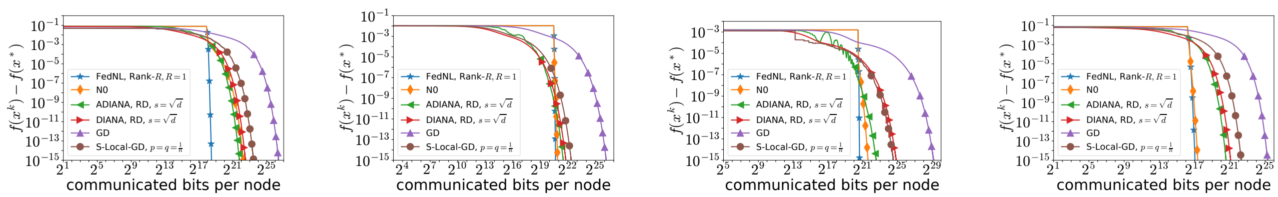


(a) `w8a`, $\lambda = 10^{-3}$ (b) `phishing`, $\lambda = 10^{-3}$ (c) `a1a`, $\lambda = 10^{-4}$ (d) `w7a`, $\lambda = 10^{-4}$

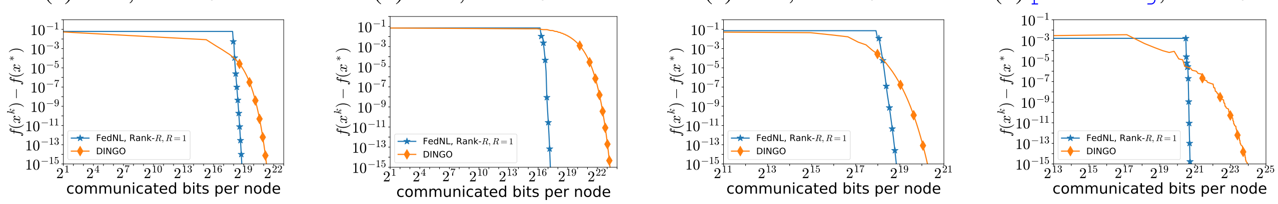
Figure 10: Comparison of FedNL with three types of compression and NL1 in terms of communication complexity.

B.12. Local comparison

Now we compare FedNL (Rank-1 compressor, $\alpha = 1$) and N0 with first order methods: ADIANA with random dithering (ADIANA, RD, $s = \sqrt{d}$), DIANA with random dithering (DIANA, RD, $s = \sqrt{d}$), Shifted Local gradient descent (S-Local-GD, $p = q = \frac{1}{n}$), and vanilla gradient descent (GD). Here we set x^0 close to the solution x^* in order to highlight fast local rates of FedNL and N0 independent of the condition number. Moreover, we compare FedNL (Rank-1 compressor, $\alpha = 1$) against DINGO. In order to make fair comparison we calculate transmitted bits in both directions, since DINGO requires several expensive communication round per one iteration of the algorithm. All results are presented in Figure 11. We clearly see that FedNL and N0 are more communication effective methods than gradient type ones. In some cases the difference is large; see Figure 11: (a), (d). In addition FedNL is more effective than DINGO in terms of communication complexity.



(a) `a1a`, $\lambda = 10^{-3}$ (b) `w8a`, $\lambda = 10^{-3}$ (c) `w7a`, $\lambda = 10^{-4}$ (d) `phishing`, $\lambda = 10^{-4}$

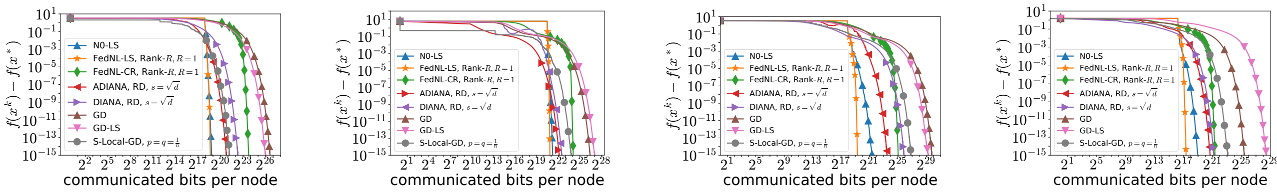


(a) `a1a`, $\lambda = 10^{-3}$ (b) `phishing`, $\lambda = 10^{-3}$ (c) `a9a`, $\lambda = 10^{-4}$ (d) `w7a`, $\lambda = 10^{-4}$

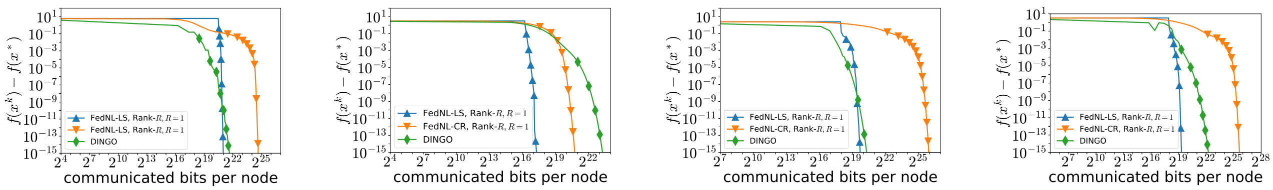
Figure 11: First row: Local comparison of FedNL and N0 with ADIANA, DIANA, S-Local-GD, and GD in terms of communication complexity. **Second row:** Local comparison of FedNL with DINGO in terms of communication complexity.

B.13. Global compersion

In our next test we compare FedNL-LS (Rank-1 compressor, $\alpha = 1$), N0-LS, and FedNL-CR (Rank-1 compressor, $\alpha = 1$) with gradient type methods such as ADIANA with random dithering (ADIANA, RD, $s = \sqrt{d}$), DIANA with random dithering (DIANA, RD, $s = \sqrt{d}$), Shifted Local gradient descent (S-Local-GD, $p = q = \frac{1}{n}$), vanilla gradient descent (GD), and gradient descent with line search (GD-LS). Besides, we compare FedNL-LS (Rank-1 compressor, $\alpha = 1$) and FedNL-CR (Rank-1 compressor, $\alpha = 1$) with DINGO. Since DINGO requires several expensive communication round per iteration, we calculate transmitted bits in both directions to make fair comparison. According to numerical experiments, we can conclude that FedNL-LS and N0-LS are more communication effective methods than gradient type ones. In some cases (see Figure 12: (c), (d)) FedNL-CR performs better or the same as DIANA.



(a) a9a, $\lambda = 10^{-3}$ (b) w7a, $\lambda = 10^{-3}$ (c) a1a, $\lambda = 10^{-4}$ (d) phishing, $\lambda = 10^{-4}$



(a) w7a, $\lambda = 10^{-3}$ (b) phishing, $\lambda = 10^{-3}$ (c) a9a, $\lambda = 10^{-4}$ (d) a1a, $\lambda = 10^{-4}$

Figure 12: First row: Global comparison of FedNL-LS, N0-LS, and FedNL-CR with ADIANA, DIANA, S-Local-GD, GD, and GD-LS in terms of communication complexity. **Second row:** Global comparison of FedNL-LS and FedNL-CR with DINGO in terms of communication complexity.

B.14. Effect of statistical heterogeneity

In this set of experiments we investigate the performance of FedNL under different level of heterogeneity of data. We generate synthethic data via rules as (Li et al., 2018) did. We set number of nodes $n = 30$, the size of local data $m = 200$, the dimension of the problem $d = 100$, and regularization parameter $\lambda = 10^{-3}$.

The generation rules for non-IID synthetic data have two positive parameteres α, β. For each node $i \in [n]$ let $B_i \sim \mathcal{N}(0, \beta)$. We use diagonal covariance matrix Σ with $\Sigma_{j,j} = j^{-1.2}$, and mean vector v_i, each element of which is generated from $\mathcal{N}(B_i, 1)$ in order to get feature vector $a_{ij} \in \mathbb{R}^d$ from $\mathcal{N}(v_i, \Sigma)$. Let $u_i \sim \mathcal{N}(0, \alpha)$, $c_i \sim \mathcal{N}(u_i, 1)$, then we generate vector $w_i \in \mathbb{R}^d$ each entire of which is sampled from $\mathcal{N}(u_i, 1)$. Let $p_{ij} = \sigma\left(w_i^\top a_{ij} + c_i\right)$, where $\sigma(\cdot)$ is a sigmoid function. Finally the label b_{ij} is equal to -1 with probability p_{ij}, and is equal to $+1$ with probability $1 - p_{ij}$. We denote the data which is generated following the rules above as Synthetic (α, β).

In addition, we generate IID data where $w \sim \mathcal{N}(0, 1)$ and $c \sim \mathcal{N}(0, 1)$ are sampled only once and used for each node i. Feature vectors a_{ij} is generated from $\mathcal{N}(v_i, \Sigma)$, where each element of v_i is equal to $B_i \sim \mathcal{N}(0, \beta)$. The label b_{ij} is equal to -1 with probability $p_{ij} = \sigma(w^\top a_{ij} + c)$, and $+1$ otherwise. We denote such data as IID.

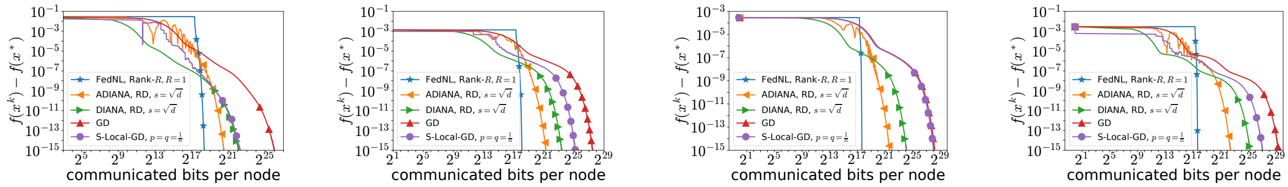
Using generated synthetic datasets we compare local performance of FedNL (Rank-1 compressor, $\alpha = 1$, Option 2), ADIANA with random dithering (ADIANA, RD, $s = \sqrt{d}$), DIANA with random dithering (DIANA, RD, $s = \sqrt{d}$), Shifted Local gradient descent (S-Local-GD, $p = q = \frac{1}{n}$), and vanilla gradient descent (GD) in terms of communication complexit; see Figure 13 (first row). Besides, we compare FedNL and DINGO; see Figure 13 (second row). According to the results, we see that the difference between FedNL and gradient type methods is getting larger, when the local data is becoming more heterogeneous; FedNL outperforms other methods by several orders in magnitude. FedNL is more stable varying data heterogeneity than DINGO. The difference between these two methods on IID data is small; when data is becoming more heterogeneous, the difference is increasing dramatically.

C. Proofs of Results from Section 3

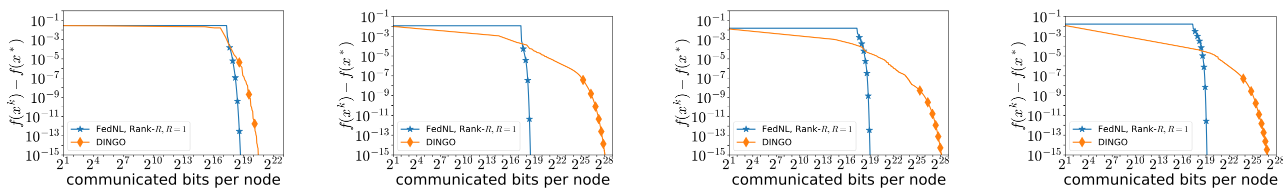
C.1. Auxiliary lemmas

Denote by $\mathbb{E}_k\left[\cdot\right]$ the conditional expectation given k^{th} iterate x^k. We first develop a lemma to handle different cases of compressors for $\mathbb{E}_k\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2$, where $\mathbb{E}_k[y] = y$ and $\mathbb{E}_k[z] = z$.

Lemma C.1. *For any y, $z \in \mathbb{R}^d$ such that $\mathbb{E}_k[y] = y$ and $\mathbb{E}_k[z] = z$, we have the following results in different cases.*



(a) IID (b) Synthetic $(0.5, 0.5)$ (c) Synthetic $(0.75, 0.75)$ (d) Syntethic $(1, 1)$



(a) IID (b) Synthetic $(0.5, 0.5)$ (c) Synthetic $(0.75, 0.75)$ (d) Syntethic $(1, 1)$

Figure 13: First row: Local comparison of FedNL, with ADIANA, DIANA, S-Local-GD, and GD in terms of communication complexity. **Second row:** Local comparison of FedNL with DINGO in terms of communication complexity.

(i) If $\mathcal{C}_i^k \in \mathbb{B}(\omega)$ and $\alpha \le \frac{1}{\omega+1}$, then

$$\mathbb{E}_k\left[\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2\right] \le (1-\alpha)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + \alpha L_{\rm F}^2\|y-z\|^2.$$

(ii) If $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and $\alpha = 1-\sqrt{1-\delta}$, then

$$\mathbb{E}_k\left[\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2\right] \le (1-\alpha^2)\left\|\mathbf{H}_i^k - \nabla^2 f_i(z)\right\|_{\rm F}^2 + \alpha L_{\rm F}^2\|y-z\|^2.$$

(iii) If $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and $\alpha = 1$, then

$$\mathbb{E}_k\left[\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2\right] \le \left(1-\frac{\delta}{4}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + \left(\frac{6}{\delta}-\frac{7}{2}\right) L_{\rm F}^2\|y-z\|^2.$$

Using the notation from (5), we can unify the above three cases into

$$\mathbb{E}_k\left[\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2\right] \le (1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + BL_{\rm F}^2\|y-z\|^2.$$

Proof. Let

$$LHS := \mathbb{E}_k\left[\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\rm F}^2\right]$$

be the left hand side appearing in these inequalities.

(i). If $\mathcal{C}_i^k \in \mathbb{B}(\omega)$, then

$$\begin{aligned} LHS &= \|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + 2\alpha\langle \mathbf{H}_i^k - \nabla^2 f_i(z), \nabla^2 f_i(y) - \mathbf{H}_i^k\rangle + \alpha^2\mathbb{E}_k\|\mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k)\|_{\rm F}^2 \\ &\le \|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + 2\alpha\langle \mathbf{H}_i^k - \nabla^2 f_i(z), \nabla^2 f_i(y) - \mathbf{H}_i^k\rangle + \alpha^2(\omega+1)\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\rm F}^2. \end{aligned}$$

Using the stepsize restriction $\alpha \le \frac{1}{\omega+1}$, we can bound $\alpha^2(\omega+1) \le \alpha$. Plugging this back to the above inequality and using the identity $2\langle \mathbf{A}, \mathbf{B}\rangle_{\rm F} + \|\mathbf{B}\|_{\rm F}^2 = -\|\mathbf{A}\|_{\rm F}^2 + \|\mathbf{A}+\mathbf{B}\|_{\rm F}^2$, we get

$$\begin{aligned} LHS &\le \|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + 2\alpha\langle \mathbf{H}_i^k - \nabla^2 f_i(z), \nabla^2 f_i(y) - \mathbf{H}_i^k\rangle + \alpha\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\rm F}^2 \\ &= (1-\alpha)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + \alpha\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\rm F}^2 \\ &\le (1-\alpha)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\rm F}^2 + \alpha L_{\rm F}^2\|y-z\|^2. \end{aligned}$$

(ii). Let $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and $\alpha = 1-\sqrt{1-\delta}$. Denote

$$\mathbf{A} := \mathbf{H}_i^k - \nabla^2 f_i(z), \quad \mathbf{B} := \nabla^2 f_i(y) - \mathbf{H}_i^k.$$

Then

$$\begin{aligned}
LHS &= \left\|\mathbf{A}+\alpha\mathcal{C}_i^k(\mathbf{B})\right\|_{\mathrm{F}}^2 \\
&= \|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathcal{C}_i^k(\mathbf{B})\right\rangle_{\mathrm{F}} + \alpha^2\|\mathcal{C}_i^k(\mathbf{B})\|_{\mathrm{F}}^2 \\
&\leq \|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + 2\alpha\left\langle \mathbf{A}, \mathcal{C}_i^k(\mathbf{B})-\mathbf{B}\right\rangle_{\mathrm{F}} + \alpha^2\|\mathbf{B}\|_{\mathrm{F}}^2 \\
&\leq \|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + 2\alpha\|\mathbf{A}\|_{\mathrm{F}}\|\mathcal{C}_i^k(\mathbf{B})-\mathbf{B}\|_{\mathrm{F}} + \alpha^2\|\mathbf{B}\|_{\mathrm{F}}^2 \\
&\leq \|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + 2\alpha\sqrt{1-\delta}\|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_{\mathrm{F}} + \alpha^2\|\mathbf{B}\|_{\mathrm{F}}^2 \\
&\leq \|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + \alpha\sqrt{1-\delta}\left(\|\mathbf{A}\|_{\mathrm{F}}^2 + \|\mathbf{B}\|_{\mathrm{F}}^2\right) + \alpha^2\|\mathbf{B}\|_{\mathrm{F}}^2 \\
&\leq (1+\alpha\sqrt{1-\delta})\,\|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + (\alpha\sqrt{1-\delta}+\alpha^2)\|\mathbf{B}\|_{\mathrm{F}}^2.
\end{aligned}$$

Since $\alpha = 1-\sqrt{1-\delta}$, we have $\alpha\sqrt{1-\delta}+\alpha^2 = \alpha$. Using the identity $2\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + \|\mathbf{B}\|_{\mathrm{F}}^2 = -\|\mathbf{A}\|_{\mathrm{F}}^2 + \|\mathbf{A}+\mathbf{B}\|_{\mathrm{F}}^2$, we get

$$\begin{aligned}
LHS &\leq (1+\alpha\sqrt{1-\delta})\,\|\mathbf{A}\|_{\mathrm{F}}^2 + 2\alpha\left\langle \mathbf{A}, \mathbf{B}\right\rangle_{\mathrm{F}} + \alpha\|\mathbf{B}\|_{\mathrm{F}}^2 \\
&= (1+\alpha\sqrt{1-\delta}-\alpha)\,\|\mathbf{A}\|_{\mathrm{F}}^2 + \alpha\|\mathbf{A}+\mathbf{B}\|_{\mathrm{F}}^2 \\
&= (1-\alpha^2)\,\|\mathbf{A}\|_{\mathrm{F}}^2 + \alpha\|\mathbf{A}+\mathbf{B}\|_{\mathrm{F}}^2 \\
&= (1-\alpha^2)\left\|\mathbf{H}_i^k - \nabla^2 f_i(z)\right\|_{\mathrm{F}}^2 + \alpha\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&\leq (1-\alpha^2)\left\|\mathbf{H}_i^k - \nabla^2 f_i(z)\right\|_{\mathrm{F}}^2 + \alpha L_{\mathrm{F}}^2\|y-z\|^2.
\end{aligned}$$

(iii). If $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and $\alpha = 1$, we have

$$\begin{aligned}
LHS &= \|\mathbf{H}_i^k + \mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&= \|\mathbf{H}_i^k - \nabla^2 f_i(y) + \mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k) + \nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&\leq (1+\beta)\|\mathbf{H}_i^k - \nabla^2 f_i(y) + \mathcal{C}_i^k(\nabla^2 f_i(y) - \mathbf{H}_i^k)\|_{\mathrm{F}}^2 + \left(1+\frac{1}{\beta}\right)\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&\leq (1+\beta)(1-\delta)\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\mathrm{F}}^2 + \left(1+\frac{1}{\beta}\right)\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2,
\end{aligned}$$

where we use Young's inequality in the first inequality for some $\beta > 0$, and use the contraction property in the last inequality. By choosing $\beta = \frac{\delta}{2(1-\delta)}$ when $0 < \delta < 1$, we can get

$$\begin{aligned}
LHS &\leq \left(1-\frac{\delta}{2}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\mathrm{F}}^2 + \left(\frac{2}{\delta}-1\right)\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&\leq \left(1-\frac{\delta}{2}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\mathrm{F}}^2 + \left(\frac{2}{\delta}-1\right)L_{\mathrm{F}}^2\|y-z\|^2.
\end{aligned}$$

When $\delta = 1$,

$$LHS = \|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \leq L_{\mathrm{F}}^2\|y-z\|^2.$$

Overall, for any $0 < \delta \leq 1$ we have

$$\begin{aligned}
LHS &\leq \left(1-\frac{\delta}{2}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(y)\|_{\mathrm{F}}^2 + \left(\frac{2}{\delta}-1\right)L_{\mathrm{F}}^2\|y-z\|^2 \\
&\leq (1+\beta)\left(1-\frac{\delta}{2}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 + \left(1+\frac{1}{\beta}\right)\left(1-\frac{\delta}{2}\right)\|\nabla^2 f_i(y) - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 \\
&\quad + \left(\frac{2}{\delta}-1\right)L_{\mathrm{F}}^2\|y-z\|^2.
\end{aligned}$$

By choosing $\beta = \frac{\delta}{4-2\delta}$, we arrive at

$$\begin{aligned}
LHS &\leq \left(1-\frac{\delta}{4}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 + \left(\frac{4}{\delta}+\frac{\delta}{2}-3+\frac{2}{\delta}-1\right)L_{\mathrm{F}}^2\|y-z\|^2 \\
&\leq \left(1-\frac{\delta}{4}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(z)\|_{\mathrm{F}}^2 + \left(\frac{6}{\delta}-\frac{7}{2}\right)L_{\mathrm{F}}^2\|y-z\|^2.
\end{aligned}$$

□

Lemma C.2. *Let* $\mathbf{H}^* \in \mathcal{X} := \{\mathbf{M} \in \mathbb{R}^{d\times d} \colon \mathbf{M}^\top = \mathbf{M},\ \mu\mathbf{I} \preceq \mathbf{M}\}$ *and* $\mathbf{H} \in \mathbb{R}^{d\times d}$ *be any matrix. Then for the projected matrix*

$$\mathbf{H}_\mu = \arg\min_{\mathbf{M}\in\mathcal{X}} \|\mathbf{M} - \mathbf{H}\|_{\rm F}$$

we have the following inequality

$$\|\mathbf{H}_\mu - \mathbf{H}^*\|_{\rm F} \le \|\mathbf{H} - \mathbf{H}^*\|_{\rm F}. \tag{23}$$

Proof. The proof follows from general priciples of functional analysis using the fact that $\mathbb{R}^{d\times d}$ is a Hilbert space with respect to Frobenious norm and the observation that the set $\mathcal{X}$ is closed and convex set.

Fix $\epsilon \in (0,1)$ and denote $\mathbf{M} = \mathbf{H}_\mu + \epsilon(\mathbf{H}^* - \mathbf{H}_\mu)$. Since both $\mathbf{H}^*$ and $\mathbf{H}_\mu$ are from $\mathcal{X}$, we have $\mathbf{M} = \epsilon\mathbf{H}^* + (1-\epsilon)\mathbf{H}_\mu \in \mathcal{X}$ from the convexity of $\mathcal{X}$. Then, from the definition of $\mathbf{H}_\mu$ we imply that

$$\begin{aligned}
\|\mathbf{H}_\mu - \mathbf{H}\|_{\rm F}^2 &\le \|\mathbf{M} - \mathbf{H}\|_{\rm F}^2 \\
&= \|\mathbf{H} - \mathbf{H}_\mu - \epsilon(\mathbf{H}^* - \mathbf{H}_\mu)\|_{\rm F}^2 \\
&= \|\mathbf{H} - \mathbf{H}_\mu\|_{\rm F}^2 - 2\epsilon \left\langle \mathbf{H} - \mathbf{H}_\mu, \mathbf{H}^* - \mathbf{H}_\mu \right\rangle_{\rm F} + \epsilon^2 \|\mathbf{H}^* - \mathbf{H}_\mu\|_{\rm F}^2,
\end{aligned}$$

from which we get

$$\left\langle \mathbf{H} - \mathbf{H}_\mu, \mathbf{H}^* - \mathbf{H}_\mu \right\rangle_{\rm F} \le \frac{\epsilon}{2} \|\mathbf{H}^* - \mathbf{H}_\mu\|_{\rm F}^2$$

for any $\epsilon \in (0,1)$. As we can choose ϵ arbitrarily small, it must hold

$$\left\langle \mathbf{H} - \mathbf{H}_\mu, \mathbf{H}^* - \mathbf{H}_\mu \right\rangle_{\rm F} \le 0.$$

It remains to notice that

$$\begin{aligned}
\|\mathbf{H} - \mathbf{H}^*\|_{\rm F}^2 &= \|\mathbf{H} - \mathbf{H}_\mu + \mathbf{H}_\mu - \mathbf{H}^*\|_{\rm F}^2 \\
&= \|\mathbf{H} - \mathbf{H}_\mu\| + \|\mathbf{H}_\mu - \mathbf{H}^*\|_{\rm F}^2 + 2\left\langle \mathbf{H} - \mathbf{H}_\mu, \mathbf{H}_\mu - \mathbf{H}^* \right\rangle_{\rm F} \\
&\ge \|\mathbf{H}_\mu - \mathbf{H}^*\|_{\rm F}^2.
\end{aligned}$$

□

C.2. Proof of Theorem 3.6

We derive recurrence relation for $\|x^k - x^*\|^2$ covering both options of updating the global model. If *Option 1.* is used in FedNL, then

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &= \left\| x^k - x^* - \left[\mathbf{H}_\mu^k\right]^{-1} \nabla f(x^k) \right\|^2 \\
&\le \left\| \left[\mathbf{H}_\mu^k\right]^{-1} \right\|^2 \left\| \mathbf{H}_\mu^k (x^k - x^*) - \nabla f(x^k)) \right\|^2 \\
&\le \frac{2}{\mu^2} \left(\left\| \left(\mathbf{H}_\mu^k - \nabla^2 f(x^*)\right)(x^k - x^*) \right\|^2 + \left\| \nabla^2 f(x^*)(x^k - x^*) - \nabla f(x^k) + \nabla f(x^*) \right\|^2 \right) \\
&= \frac{2}{\mu^2} \left(\left\| \left(\mathbf{H}_\mu^k - \nabla^2 f(x^*)\right)(x^k - x^*) \right\|^2 + \left\| \nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*) \right\|^2 \right) \\
&\le \frac{2}{\mu^2} \left(\left\| \mathbf{H}_\mu^k - \nabla^2 f(x^*) \right\|^2 \|x^k - x^*\|^2 + \frac{L_*^2}{4} \|x^k - x^*\|^4 \right) \\
&= \frac{2}{\mu^2} \|x^k - x^*\|^2 \left(\left\| \mathbf{H}_\mu^k - \nabla^2 f(x^*) \right\|^2 + \frac{L_*^2}{4} \|x^k - x^*\|^2 \right) \\
&\overset{(23)}{\le} \frac{2}{\mu^2} \|x^k - x^*\|^2 \left(\left\| \mathbf{H}^k - \nabla^2 f(x^*) \right\|^2 + \frac{L_*^2}{4} \|x^k - x^*\|^2 \right) \\
&\le \frac{2}{\mu^2} \|x^k - x^*\|^2 \left(\left\| \mathbf{H}^k - \nabla^2 f(x^*) \right\|_{\rm F}^2 + \frac{L_*^2}{4} \|x^k - x^*\|^2 \right),
\end{aligned}$$

where we use $\mathbf{H}_\mu^k \succeq \mu \mathbf{I}$ in the second inequality, and $\nabla^2 f(x^*) \succeq \mu \mathbf{I}$ in the fourth inequality. From the convexity of $\|\cdot\|_{\mathrm{F}}^2$, we have

$$\|\mathbf{H}^k - \nabla^2 f(x^*)\|_{\mathrm{F}}^2 = \left\| \frac{1}{n}\sum_{i=1}^n \left(\mathbf{H}_i^k - \nabla^2 f_i(x^*)\right)\right\|_{\mathrm{F}}^2 \leq \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 = \mathcal{H}^k.$$

Thus,

$$\|x^{k+1} - x^*\|^2 \leq \frac{2}{\mu^2}\|x^k - x^*\|^2 \mathcal{H}^k + \frac{L_*^2}{2\mu^2}\|x^k - x^*\|^4. \tag{24}$$

If *Option 2.* is used in FedNL, then as $\mathbf{H}^k + l^k \mathbf{I} \succeq \nabla^2 f(x^k) \succeq \mu \mathbf{I}$ and $\nabla f(x^*) = 0$, we have

$$\begin{aligned}
\|x^{k+1} - x^*\| &= \|x^k - x^* - [\mathbf{H}^k + l^k \mathbf{I}]^{-1}\nabla f(x^k)\| \\
&\leq \|[\mathbf{H}^k + l^k\mathbf{I}]^{-1}\| \cdot \|(\mathbf{H}^k + l^k\mathbf{I})(x^k - x^*) - \nabla f(x^k) + \nabla f(x^*)\| \\
&\leq \frac{1}{\mu}\|(\mathbf{H}^k + l^k\mathbf{I} - \nabla^2 f(x^*))(x^k - x^*)\| + \frac{1}{\mu}\|\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*)\| \\
&\leq \frac{1}{\mu}\|\mathbf{H}^k + l^k\mathbf{I} - \nabla^2 f(x^*)\|\|x^k - x^*\| + \frac{L_*}{2\mu}\|x^k - x^*\|^2 \\
&\leq \frac{1}{n\mu}\sum_{i=1}^n \|\mathbf{H}_i^k + l_i^k\mathbf{I} - \nabla^2 f_i(x^*)\|\|x^k - x^*\| + \frac{L_*}{2\mu}\|x^k - x^*\|^2 \\
&\leq \frac{1}{n\mu}\sum_{i=1}^n (\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\| + l_i^k)\|x^k - x^*\| + \frac{L_*}{2\mu}\|x^k - x^*\|^2.
\end{aligned}$$

From the definition of l_i^k, we have

$$l_i^k = \|\mathbf{H}_i^k - \nabla^2 f_i(x^k)\|_{\mathrm{F}} \leq \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + L_{\mathrm{F}}\|x^k - x^*\|.$$

Thus,

$$\|x^{k+1} - x^*\| \leq \frac{2}{n\mu}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}\|x^k - x^*\| + \frac{L_* + 2L_{\mathrm{F}}}{2\mu}\|x^k - x^*\|^2.$$

From Young's inequality, we further have

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &\leq \frac{8}{\mu^2}\left(\frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}\|x^k - x^*\|\right)^2 + \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}\|x^k - x^*\|^4 \\
&\leq \frac{8}{\mu^2}\|x^k - x^*\|^2\left(\frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2\right) + \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}\|x^k - x^*\|^4 \\
&= \frac{8}{\mu^2}\|x^k - x^*\|^2\mathcal{H}^k + \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}\|x^k - x^*\|^4,
\end{aligned} \tag{25}$$

where we use the convexity of $\|\cdot\|_{\mathrm{F}}^2$ in the second inequality.

Thus, from (24) and (25), we have the following unified bound for both *Option 1* and *Option 2*:

$$\|x^{k+1} - x^*\|^2 \leq \frac{C}{\mu^2}\|x^k - x^*\|^2\mathcal{H}^k + \frac{D}{2\mu^2}\|x^k - x^*\|^4. \tag{26}$$

Assume $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{2D}$ and $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for all $k \geq 0$. Then we show that $\|x^k - x^*\|^2 \leq \frac{\mu^2}{2D}$ for all $k \geq 0$ by induction. Assume $\|x^k - x^*\|^2 \leq \frac{\mu^2}{2D}$ for all $k \leq K$. Then from (26), we have

$$\|x^{K+1} - x^*\|^2 \leq \frac{1}{4}\|x^K - x^*\|^2 + \frac{1}{4}\|x^K - x^*\|^2 \leq \frac{\mu^2}{2D}.$$

Thus we have $\|x^k - x^*\|^2 \leq \frac{\mu^2}{2D}$ and $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for $k \geq 0$. Using (26) again, we obtain

$$\|x^{k+1} - x^*\|^2 \leq \frac{1}{2}\|x^k - x^*\|^2. \tag{27}$$

Choosing $y = x^k$ and $z = x^*$ in Lemma C.1, we get

$$\mathbb{E}_k\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k) - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq (1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2\|x^k - x^*\|^2.$$

Then by $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$, we have

$$\mathbb{E}_k[\mathcal{H}^{k+1}] \leq (1-A)\mathcal{H}^k + BL_{\rm F}^2\|x^k - x^*\|^2.$$

Using the above inequality and (27), for Lyapunov function Φ^k we deduce

$$\begin{aligned}
\mathbb{E}_k[\Phi^{k+1}] &\leq (1-A)\mathcal{H}^k + BL_{\rm F}^2\|x^k - x^*\|^2 + 3BL_{\rm F}^2\|x^k - x^*\|^2 \\
&= (1-A)\mathcal{H}^k + \left(1 - \frac{1}{3}\right)6BL_{\rm F}^2\|x^k - x^*\|^2 \\
&\leq \left(1 - \min\left\{A, \frac{1}{3}\right\}\right)\Phi^k.
\end{aligned}$$

Hence $\mathbb{E}[\Phi^k] \leq \left(1 - \min\left\{A, \frac{1}{3}\right\}\right)^k \Phi^0$. We further have $\mathbb{E}[\mathcal{H}^k] \leq \left(1 - \min\left\{A, \frac{1}{3}\right\}\right)^k \Phi^0$ and $\mathbb{E}[\|x^k - x^*\|^2] \leq \frac{1}{6BL_{\rm F}^2}\left(1 - \min\left\{A, \frac{1}{3}\right\}\right)^k \Phi^0$ for $k \geq 0$. Assume $x^k \neq x^*$ for all k. Then from (26), we have

$$\frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \leq \frac{C}{\mu^2}\mathcal{H}^k + \frac{D}{2\mu^2}\|x^k - x^*\|^2,$$

and by taking expectation, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2}\right] &\leq \frac{C}{\mu^2}\mathbb{E}[\mathcal{H}^k] + \frac{D}{2\mu^2}\mathbb{E}[\|x^k - x^*\|^2] \\
&\leq \left(1 - \min\left\{A, \frac{1}{3}\right\}\right)^k \left(C + \frac{D}{12BL_{\rm F}^2}\right)\frac{\Phi^0}{\mu^2}.
\end{aligned}$$

C.3. Proof of Lemma 3.7

We prove this by induction. Assume $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq \frac{\mu^2}{4C}$ and $\|x^k - x^*\|^2 \leq \min\{\frac{A\mu^2}{4BCL_{\rm F}^2}, \frac{\mu^2}{2D}\}$ for $k \leq K$. Then we also have $\mathcal{H}^k \leq \frac{\mu^2}{4C}$ for $k \leq K$. From (26), we can get

$$\begin{aligned}
\|x^{K+1} - x^*\|^2 &\leq \frac{C}{\mu^2}\|x^K - x^*\|^2\mathcal{H}^K + \frac{D}{2\mu^2}\|x^K - x^*\|^4 \\
&\leq \frac{1}{4}\|x^K - x^*\|^2 + \frac{1}{4}\|x^K - x^*\|^2 \\
&\leq \min\left\{\frac{A\mu^2}{4BCL_{\rm F}^2}, \frac{\mu^2}{2D}\right\}.
\end{aligned}$$

From Lemma C.1, by choosing $y = x^k$ and $z = x^*$, for all $i \in [n]$, we have

$$\begin{aligned}
\|\mathbf{H}_i^{K+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 &= \mathbb{E}_k\|\mathbf{H}_i^K + \alpha\mathcal{C}_i^k(\nabla^2 f_i(x^K) - \mathbf{H}_i^K) - \nabla^2 f_i(x^*)\|_{\rm F}^2 \\
&\leq (1-A)\|\mathbf{H}_i^K - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2\|x^K - x^*\|^2 \\
&\leq (1-A)\frac{\mu^2}{4C} + BL_{\rm F}^2 \cdot \frac{A\mu^2}{4BCL_{\rm F}^2} \\
&= \frac{\mu^2}{4C}.
\end{aligned}$$

C.4. Proof of Lemma 3.8

Notice that Assumption 3.5 implies $\mathbf{H}_i^0 = \nabla^2 f_i(x^0)$, from which we have

$$\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 = \sum_{j,l} |(\nabla^2 f_i(x^0) - \nabla^2 f_i(x^*))_{jl}|^2 \le d^2 L_\infty^2 \frac{\mu^2}{D + 4Cd^2 L_\infty^2} \le \frac{\mu^2}{4C},$$

which implies $\mathcal{H}^0 \le \frac{\mu^2}{4C}$. Next we prove $\|x^k - x^*\|^2 \le \frac{\mu^2}{D+4Cd^2L_\infty^2}$ for all $k \ge 0$ by induction. Assume $\|x^k - x^*\|^2 \le \frac{\mu^2}{D+4Cd^2L_\infty^2}$ for $k \le K$. Since $(\mathbf{H}_i^k)_{jl}$ is a convex combination of $\{(\nabla^2 f_i(x^0))_{jl}, ..., (\nabla^2 f_i(x^k))_{jl}\}$ for all $i \in [n], j, l \in [d]$, from the convexity of $|\cdot|^2$, we have

$$|(\mathbf{H}_i^k - \nabla^2 f_i(x^*))_{jl}|^2 \le L_\infty^2 \cdot \frac{\mu^2}{D + 4Cd^2 L_\infty^2} \le \frac{\mu^2}{4Cd^2},$$

for $k \le K$. Then we can get $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 \le \frac{\mu^2}{4C}$ and thus $\mathcal{H}^k \le \frac{\mu^2}{4C}$ for $k \le K$. From (26), we have

$$\begin{aligned}
\|x^{K+1} - x^*\|^2 &\le \frac{C}{\mu^2}\|x^K - x^*\|^2 \mathcal{H}^K + \frac{D}{2\mu^2}\|x^K - x^*\|^4 \\
&\le \frac{1}{4}\|x^K - x^*\|^2 + \frac{1}{2}\|x^K - x^*\|^2 \\
&\le \frac{\mu^2}{D + 4Cd^2 L_\infty^2}.
\end{aligned}$$

D. Extension: Partial Participation (FedNL-PP)

Our first extension to the vanilla FedNL is to handle partial participation: a setup when in each iteration only randomly selected clients participate. This is important when the number n of devices is very large.

Algorithm 2 FedNL-PP (Federated Newton Learn with Partial Participation)

Parameters: Hessian learning rate $\alpha > 0$; compression operators $\{\mathcal{C}_1^k, \dots, \mathcal{C}_n^k\}$; number of participating devices $\tau \in \{1, 2, \dots, n\}$
Initialization: For all $i \in [n]$: $w_i^0 = x^0 \in \mathbb{R}^d$; $\mathbf{H}_i^0 \in \mathbb{R}^{d\times d}$; $l_i^0 = \|\mathbf{H}_i^0 - \nabla^2 f_i(w_i^0)\|_{\mathrm{F}}$; $g_i^0 = (\mathbf{H}_i^0 + l_i^0 \mathbf{I}) w_i^0 - \nabla f_i(w_i^0)$; Moreover: $\mathbf{H}^0 = \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$; $l^0 = \frac{1}{n}\sum_{i=1}^n l_i^0$; $g^0 = \frac{1}{n}\sum_{i=1}^n g_i^0$
on server
  $x^{k+1} = \left(\mathbf{H}^k + l^k \mathbf{I}\right)^{-1} g^k$ Main step: Update the global model
  Choose a subset $S^k \subseteq \{1, \dots, n\}$ of devices of cardinality τ, uniformly at random
  Send x^{k+1} to the selected devices $i \in S^k$ Communicate to selected clients
for each device $i = 1, \dots, n$ in parallel **do**
  for participating devices $i \in S^k$ **do**
  $w_i^{k+1} = x^{k+1}$ Update local model
  $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathcal{C}_i^k(\nabla^2 f_i(w_i^{k+1}) - \mathbf{H}_i^k)$ Update local Hessian estimate
  $l_i^{k+1} = \|\mathbf{H}_i^{k+1} - \nabla^2 f_i(w_i^{k+1})\|_{\mathrm{F}}$ Compute local Hessian error
  $g_i^{k+1} = (\mathbf{H}_i^{k+1} + l_i^{k+1}\mathbf{I}) w_i^{k+1} - \nabla f_i(w_i^{k+1})$ Compute Hessian-corrected local gradient
  Send $\mathcal{C}_i^k(\nabla^2 f_i(w_i^{k+1}) - \mathbf{H}_i^k)$, $l_i^{k+1} - l_i^k$ and $g_i^{k+1} - g_i^k$ to server Communicate to server
  for non-participating devices $i \notin S^k$ **do**
  $w_i^{k+1} = w_i^k$, $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k$, $l_i^{k+1} = l_i^k$, $g_i^{k+1} = g_i^k$ Do nothing
end for
on server
  $g^{k+1} = g^k + \frac{1}{n}\sum_{i\in S^k}\left(g_i^{k+1} - g_i^k\right)$ Maintain the relationship $g^k = \frac{1}{n}\sum_{i=1}^n g_i^k$
  $\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{\alpha}{n}\sum_{i\in S^k}\mathcal{C}_i^k(\nabla^2 f_i(w_i^{k+1}) - \mathbf{H}_i^k)$ Update the Hessian estimate on the server
  $l^{k+1} = l^k + \frac{1}{n}\sum_{i\in S^k}\left(l_i^{k+1} - l_i^k\right)$ Maintain the relationship $l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$

D.1. Hessian corrected local gradients g_i^k

The key technical novelty in FedNL-PP is the structure of local gradients

$$g_i^k = (\mathbf{H}_i^k + l_i^k \mathbf{I}) w_i^k - \nabla f_i(w_i^k)$$

(see line 12 of Algorithm 2). The intuition behind this form is as follows. Because of the partial participation, some devices might remain inactive for several rounds. As a consequence, each device i holds a local model w_i^k, which is a stale global model (true global model of the last round client i participated) when the device is inactive. This breaks the analysis of FedNL and requires an additional trick to handle stale global models of inactive clients. The trick is to apply some form of Newton-type step locally and then update the global model at the server in communication efficient manner. In particular, clients use their corrected learned local Hessian estimates $\mathbf{H}_i^k + l_i^k \mathbf{I}$ to do Newton-type step from w_i^k to $w_i^k - \left[\mathbf{H}_i^k + l_i^k \mathbf{I}\right]^{-1} \nabla f_i(w_i^k)$, which can be transformed into

$$\left(\mathbf{H}_i^k + l_i^k \mathbf{I}\right)^{-1} \left[\left(\mathbf{H}_i^k + l_i^k \mathbf{I}\right) w_i^k - \nabla f_i(w_i^k)\right] = \left(\mathbf{H}_i^k + l_i^k \mathbf{I}\right)^{-1} g_i^k.$$

Next, all active clients communicate compressed differences $\mathcal{C}_i^k(\nabla^2 f_i(w_i^{k+1}) - \mathbf{H}_i^k)$, $l_i^{k+1} - l_i^k$ and $g_i^{k+1} - g_i^k$ to the sever, which then updates global estimates $g^{k+1}, \ \mathbf{H}^{k+1}, \ l^{k+1}$ (see lines 18, 19, 20) and the global model x^{k+1} (see line 4).

D.2. Importance of compression errors l_i^k

Notice that, unlike FedNL, here we have only one option to update the global model at the sever (this corresponds to *Option 2* of FedNL). Although, it is possible to extend the theory also for *Option 1*, it would require strong practical requirements. Indeed, in order to carry out the analysis with *Option 1*, either all active clients have to compute projected estimates $\left[\mathbf{H}_i^k\right]_\mu^{-1}$ or the central server needs to maintain this for all clients in each iteration. Although implementable, both variants seem to be too much restrictive from the practical point of view. Compression errors l_i^k mitigate the storage and computation requirements by the cost of sending an extra float per active client.

D.3. Local convergence theory

We prove three local rates for FedNL-PP: for the squared distance of the global model x^k to the solution $\|x^k - x^*\|^2$, averaged squared distance of stale (due to partial participation) local models w_i^k to the solution $\mathcal{W}^k := \frac{1}{n}\sum_{i=1}^n \|w_i^k - x^*\|^2$, and for the Lyapunov function

$$\Psi^k := \mathcal{H}^k + B L_{\rm F}^2 \mathcal{W}^k.$$

Theorem D.1. *Let Assumption 3.1 holds and further assume that all loss functions f_i are μ-convex. Suppose $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{4(L_* + 2L_{\rm F})^2}$ and $\mathcal{H}^k \leq \frac{\mu^2}{64}$ for all $k \geq 0$. Then, global model x^k and all local models w_i^k of FedNL-PP (Algorithm 2) converge linearly as follows*

$$\|x^{k+1} - x^*\|^2 \leq \mathcal{W}^k, \qquad \mathbb{E}\left[\mathcal{W}^k\right] \leq \left(1 - \frac{3\tau}{4n}\right)^k \mathcal{W}^0.$$

Moreover, depending on the choice (5) of compressors $\mathcal{C}_i^k$ and step-size α, we have linear rates

$$\mathbb{E}\left[\Psi^k\right] \leq \left(1 - \frac{\tau}{n} \min\left\{A, \frac{1}{2}\right\}\right)^k \Psi^0, \tag{28}$$

$$\mathbb{E}\left[\frac{\|x^{k+1} - x^*\|^2}{\mathcal{W}^k}\right] \leq \left(1 - \min\left\{A, \frac{1}{2}\right\}\right)^k \left(\frac{(L_* + 2L_{\rm F})^2}{2BL_{\rm F}^2} + 8\right) \frac{\Psi^0}{\mu^2}. \tag{29}$$

Similar to Theorem 3.6, we assumed $\mathcal{H}^k \leq \frac{\mu^2}{64}$ holds for all iterates $k \geq 0$. Below, we prove that this inequality holds, using the initial conditions only.

Lemma D.2. *Let Assumption 3.4 holds. Assume $\|x^0 - x^*\|^2 \leq e_3 := \min\{\frac{A\mu^2}{16BL_{\rm F}^2}, \frac{\mu^2}{4(L_* + 2L_{\rm F})^2}\}$ and $\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq \frac{\mu^2}{64}$. Then $\|x^k - x^*\|^2 \leq e_3$ and $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq \frac{\mu^2}{64}$ for all $k \geq 1$.*

Lemma D.3. *Let Assumption 3.5 holds and assume $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{(L_* + 2L_{\rm F})^2 + 64d^2L_\infty^2}$. Then $\mathcal{H}^k \leq \frac{\mu^2}{64}$ for all $k \geq 0$.*

In the upcoming three subsections we provide the proofs of Theorem D.1, Lemma D.2 and D.3.

D.4. Proof of Theorem D.1

From

$$x^{k+1} = \left(\mathbf{H}^k + l^k\mathbf{I}\right)^{-1} g^k = \left(\mathbf{H}^k + l^k\mathbf{I}\right)^{-1} \left[\frac{1}{n}\sum_{i=1}^n (\mathbf{H}_i^k + l_i^k\mathbf{I})w_i^k - \nabla f_i(w_i^k)\right],$$

and

$$x^* = \left(\mathbf{H}^k + l^k\mathbf{I}\right)^{-1} \left[(\mathbf{H}^k + l^k\mathbf{I})x^* - \nabla f(x^*)\right] = \left(\mathbf{H}^k + l^k\mathbf{I}\right)^{-1} \left[\frac{1}{n}\sum_{i=1}^n (\mathbf{H}_i^k + l_i^k\mathbf{I})x^* - \nabla f_i(x^*)\right],$$

we can obtain

$$x^{k+1} - x^* = \left(\mathbf{H}^k + l^k\mathbf{I}\right)^{-1} \left[\frac{1}{n}\sum_{i=1}^n (\mathbf{H}_i^k + l_i^k\mathbf{I})(w_i^k - x^*) - (\nabla f_i(w_i^k) - \nabla f_i(x^*))\right].$$

As all functions f_i are μ-convex, we get $\mathbf{H}^k + l^k\mathbf{I} \succeq \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(w_i^k) \succeq \mu\mathbf{I}$. Using the triangle inequality, we have

$$\begin{aligned}
\|x^{k+1} - x^*\| &\leq \frac{1}{\mu n}\sum_{i=1}^n \left\|\nabla f_i(w_i^k) - \nabla f_i(x^*) - (\mathbf{H}_i^k + l_i^k\mathbf{I})(w_i^k - x^*)\right\| \\
&\leq \frac{1}{\mu n}\sum_{i=1}^n \left\|\nabla f_i(w_i^k) - \nabla f_i(x^*) - \nabla^2 f_i(x^*)(w_i^k - x^*)\right\| \\
&\quad + \frac{1}{\mu n}\sum_{i=1}^n \left\|(\mathbf{H}_i^k + l_i^k\mathbf{I} - \nabla^2 f_i(x^*)(w_i^k - x^*))\right\| \\
&\leq \frac{L_*}{2\mu n}\sum_{i=1}^n \|w_i^k - x^*\|^2 + \frac{1}{\mu n}\sum_{i=1}^n \|\mathbf{H}_i^k + l_i^k\mathbf{I} - \nabla^2 f_i(x^*)\| \cdot \|w_i^k - x^*\| \\
&\leq \frac{L_*}{2\mu}\mathcal{W}^k + \frac{1}{\mu n}\sum_{i=1}^n \left(\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\| + l_i^k\right) \cdot \|w_i^k - x^*\|.
\end{aligned}$$

Recall that

$$\begin{aligned}
l_i^k &= \|\mathbf{H}_i^k - \nabla^2 f_i(w_i^k)\|_{\mathrm{F}} \\
&\leq \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \|\nabla^2 f_i(x^*) - \nabla^2 f_i(w_i^k)\|_{\mathrm{F}} \\
&\leq \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + L_{\mathrm{F}}\|w_i^k - x^*\|.
\end{aligned}$$

Then we arrive at

$$\begin{aligned}
&\|x^{k+1} - x^*\| \\
\leq\;& \frac{L_*}{2\mu}\mathcal{W}^k + \frac{1}{\mu n}\sum_{i=1}^n \left(\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\| + \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + L_{\mathrm{F}}\|w_i^k - x^*\|\right) \cdot \|w_i^k - x^*\| \\
\leq\;& \frac{L_* + 2L_{\mathrm{F}}}{2\mu}\mathcal{W}^k + \frac{2}{\mu n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} \cdot \|w_i^k - x^*\|.
\end{aligned}$$

We further use Young's inequality to bound $\|x^{k+1} - x^*\|^2$ as

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &\leq \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}(\mathcal{W}^k)^2 + \frac{8}{\mu^2 n^2}\left(\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} \cdot \|w_i^k - x^*\|\right)^2 \\
&\leq \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}(\mathcal{W}^k)^2 + \frac{8}{\mu^2}\left(\frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2\right)\mathcal{W}^k \\
&= \frac{(L_* + 2L_{\mathrm{F}})^2}{2\mu^2}(\mathcal{W}^k)^2 + \frac{8}{\mu^2}\mathcal{H}^k\mathcal{W}^k, \qquad (30)
\end{aligned}$$

where we use Cauchy-Schwarz inequality in the second inequality and use $\mathcal{H}^k = \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2$ in the last equality. From the update rule of w_i^k, we have

$$\begin{aligned} \mathbb{E}_k[\mathcal{W}^{k+1}] &= \frac{\tau}{n}\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2\right] + \left(1-\frac{\tau}{n}\right)\mathcal{W}^k \\ &\leq \frac{\tau}{n}\mathcal{W}^k\left(\frac{(L_* + 2L_{\rm F})^2}{2\mu^2}\mathcal{W}^k + \frac{8}{\mu^2}\mathcal{H}^k\right) + \left(1-\frac{\tau}{n}\right)\mathcal{W}^k. \end{aligned} \tag{31}$$

From the assumptions we have $\|w_i^0 - x^*\|^2 = \|x^0 - x^*\|^2 \leq \frac{\mu^2}{4(L_*+2L_{\rm F})^2}$ and $\mathcal{H}^k \leq \frac{\mu^2}{64}$ for all $k \geq 0$. Next we show that $\|x^k - x^*\|^2 \leq \frac{\mu^2}{4(L_*+2L_{\rm F})^2}$ for all $k \geq 1$ by mathematical induction. First, we have $\mathcal{W}^0 \leq \frac{\mu^2}{4(L_*+2L_{\rm F})^2}$. Then from (30) we have

$$\begin{aligned} \|x^1 - x^*\|^2 &\leq \frac{(L_* + 2L_{\rm F})^2}{2\mu^2}(\mathcal{W}^0)^2 + \frac{8}{\mu^2}\mathcal{H}^0\mathcal{W}^0 \\ &\leq \frac{1}{8}\mathcal{W}^0 + \frac{1}{8}\mathcal{W}^0 \\ &\leq \frac{\mu^2}{4(L_* + 2L_{\rm F})^2}. \end{aligned}$$

Assume $\|x^k - x^*\|^2 \leq \frac{\mu^2}{4(L_*+2L_{\rm F})^2}$ for $k \leq K$. Then $\mathcal{W}^k \leq \min\{\frac{\mu^2}{(L_*+2L_{\rm F})^2}, M\}$ for $k \leq K$, and from (30) and the assumption that $\mathcal{H}^k \leq \frac{\mu^2}{64}$ for $k \geq 0$, we have

$$\begin{aligned} \|x^{K+1} - x^*\|^2 &\leq \frac{(L_* + 2L_{\rm F})^2}{2\mu^2}(\mathcal{W}^K)^2 + \frac{8}{\mu^2}\mathcal{H}^K\mathcal{W}^K \\ &\leq \frac{1}{8}\mathcal{W}^K + \frac{1}{8}\mathcal{W}^K \\ &\leq \frac{\mu^2}{4(L_* + 2L_{\rm F})^2}. \end{aligned}$$

This indicates that $\frac{(L_*+2L_{\rm F})^2}{2\mu^2}\mathcal{W}^k + \frac{8}{\mu^2}\mathcal{H}^k \leq \frac{1}{4}$ for all $k \geq 0$. Then from (31), we can obtain

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leq \left(1 - \frac{3\tau}{4n}\right)\mathcal{W}^k. \tag{32}$$

By applying the tower property, we have $\mathbb{E}[\mathcal{W}^{k+1}] \leq \left(1 - \frac{3\tau}{4n}\right)\mathbb{E}[\mathcal{W}^k]$. Unrolling the recursion, we can get $\mathbb{E}[\mathcal{W}^k] \leq \left(1 - \frac{3\tau}{4n}\right)^k \mathcal{W}^0$. Since at each step, each worker makes update with probability $\frac{\tau}{n}$, we have

$$\begin{aligned} & \mathbb{E}_k\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 \\ = \;& \left(1-\frac{\tau}{n}\right)\mathbb{E}_k\left[\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 | i \notin S^k\right] + \frac{\tau}{n}\mathbb{E}_k\left[\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 | i \in S^k\right] \\ = \;& \left(1-\frac{\tau}{n}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{\tau}{n}\mathbb{E}_k\|\mathbf{H}_i^k + \alpha\mathcal{C}_i^k(\nabla^2 f_i(x^{k+1}) - \mathbf{H}_i^k) - \nabla^2 f_i(x^*)\|_{\rm F}^2. \end{aligned}$$

Then since $\mathbb{E}_k[x^{k+1}] = x^{k+1}$ and $\mathbb{E}_k[x^*] = x^*$, by choosing $z = x^*$ and $y = x^{k+1}$ in Lemma C.1, we have

$$\begin{aligned} & \mathbb{E}_k\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 \\ \leq \;& \left(1-\frac{\tau}{n}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{\tau}{n}(1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{\tau}{n}BL_{\rm F}^2\|x^{k+1} - x^*\|^2 \\ = \;& \left(1-\frac{A\tau}{n}\right)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{\tau BL_{\rm F}^2}{n}\|x^{k+1} - x^*\|^2. \end{aligned}$$

Summing up the above inequality from $i = 1$ to n and multiplying $\frac{1}{n}$, we can obtain

$$\mathbb{E}_k[\mathcal{H}^{k+1}] \leq \left(1 - \frac{A\tau}{n}\right)\mathcal{H}^k + \frac{\tau BL_{\rm F}^2}{n}\mathbb{E}_k\|x^{k+1} - x^*\|^2.$$

Recall that $\frac{(L_*+2L_{\rm F})^2}{2\mu^2}\mathcal{W}^k + \frac{8}{\mu^2}\mathcal{H}^k \leq \frac{1}{4}$ for all $k \geq 0$, from (30), we have

$$\|x^{k+1}-x^*\|^2 \leq \frac{1}{4}\mathcal{W}^k,$$

which implies that

$$\mathbb{E}_k[\mathcal{H}^{k+1}] \leq \left(1-\frac{A\tau}{n}\right)\mathcal{H}^k + \frac{\tau BL_{\rm F}^2}{4n}\mathcal{W}^k. \tag{33}$$

Then from (32) and (33), we have the following recurrence relation for the Lyapunov function Ψ:

$$\begin{aligned}
\mathbb{E}_k[\Psi^{k+1}] &= \mathbb{E}_k[\mathcal{H}^{k+1}] + BL_{\rm F}^2\mathbb{E}_k[\mathcal{W}^{k+1}] \\
&\leq \left(1-\frac{A\tau}{n}\right)\mathcal{H}^k + \frac{\tau BL_{\rm F}^2}{4n}\mathcal{W}^k + \left(1-\frac{3\tau}{4n}\right)BL_{\rm F}^2\mathcal{W}^k \\
&= \left(1-\frac{A\tau}{n}\right)\mathcal{H}^k + \left(1-\frac{\tau}{2n}\right)BL_{\rm F}^2\mathcal{W}^k \\
&\leq \left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)\Psi^k.
\end{aligned}$$

By applying the tower property, we have $\mathbb{E}[\Psi^{k+1}] \leq \left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)\mathbb{E}[\Psi^k]$. Unrolling the recursion, we can obtain $\mathbb{E}[\Psi^k] \leq \left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)^k\Psi^0$. We further have $\mathbb{E}[\mathcal{H}^k] \leq \left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)^k\Psi^0$ and $\mathbb{E}[\mathcal{W}^k] \leq \frac{1}{BL_{\rm F}^2}\left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)^k\Psi^0$, which applied on (30) gives

$$\begin{aligned}
\mathbb{E}\left[\frac{\|x^{k+1}-x^*\|^2}{\mathcal{W}^k}\right] &\leq \frac{(L_*+2L_{\rm F})^2}{2\mu^2}\mathbb{E}[\mathcal{W}^k] + \frac{8}{\mu^2}\mathbb{E}[\mathcal{H}^k] \\
&\leq \left(1-\frac{\tau}{n}\min\left\{A,\frac{1}{2}\right\}\right)^k\left(\frac{(L_*+2L_{\rm F})^2}{2BL_{\rm F}^2}+8\right)\frac{\Psi^0}{\mu^2}.
\end{aligned}$$

D.5. Proof of Lemma D.2

First, we have $\mathcal{W}^0 \leq \min\{\frac{A\mu^2}{16BL_{\rm F}^2}, \frac{\mu^2}{4(L_*+2L_{\rm F})^2}\}$ and $\mathcal{H}^0 \leq \frac{\mu^2}{64}$. Then from (30) we can get

$$\|x^1-x^*\|^2 \leq \frac{1}{4}\mathcal{W}^0.$$

For each i, either $\mathbf{H}_i^1 = \mathbf{H}_i^0$, or by Lemma C.1

$$\begin{aligned}
\|\mathbf{H}_i^1 - \nabla^2 f_i(x^*)\|_{\rm F}^2 &= \|\mathbf{H}_i^0 + \alpha\mathcal{C}_i^0(\nabla^2 f_i(x^1) - \mathbf{H}_i^0) - \nabla^2 f_i(x^*)\|_{\rm F}^2 \\
&\leq (1-A)\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2\|x^1-x^*\|^2 \\
&\leq (1-A)\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 + A\cdot\frac{1}{4A}BL_{\rm F}^2\mathcal{W}^0 \\
&\leq (1-A)\frac{\mu^2}{64} + A\cdot\frac{\mu^2}{64} \\
&\leq \frac{\mu^2}{64}.
\end{aligned}$$

We assume $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq \frac{\mu^2}{64}$ and $\|x^k-x^*\|^2 \leq \min\{\frac{A\mu^2}{16BL_{\rm F}^2}, \frac{\mu^2}{4(L_*+2L_{\rm F})^2}\}$ for all $k \leq K$. Then we have $\mathcal{H}^k \leq \frac{\mu^2}{64}$ and $\mathcal{W}^k \leq \min\{\frac{A\mu^2}{16BL_{\rm F}^2}, \frac{\mu^2}{4(L_*+2L_{\rm F})^2}\}$ for all $k \leq K$. Then from (30) we can get

$$\|x^{K+1}-x^*\|^2 \leq \frac{1}{4}\mathcal{W}^k \leq \min\{\frac{A\mu^2}{16BL_{\rm F}^2}, \frac{\mu^2}{4(L_*+2L_{\rm F})^2}\}.$$

For each i, either $\mathbf{H}_i^{K+1} = \mathbf{H}_i^K$, or by Lemma C.1

$$\begin{aligned}
\|\mathbf{H}_i^{K+1} - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 &= \|\mathbf{H}_i^K + \alpha \mathcal{C}_i^K(\nabla^2 f_i(x^{K+1}) - \mathbf{H}_i^K) - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 \\
&\leq (1-A)\|\mathbf{H}_i^K - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + BL_{\mathrm{F}}^2\|x^{K+1} - x^*\|^2 \\
&\leq (1-A)\|\mathbf{H}_i^K - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + A \cdot \frac{1}{4A} BL_{\mathrm{F}}^2 \mathcal{W}^K \\
&\leq (1-A)\frac{\mu^2}{64} + A \cdot \frac{\mu^2}{64} \\
&\leq \frac{\mu^2}{64}.
\end{aligned}$$

D.6. Proof of Lemma D.3

First, since $\mathbf{H}_i^0 = \nabla^2 f_i(w_i^0)$, we have

$$\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 = \sum_{j,l} |(\nabla^2 f_i(w_i^0) - \nabla^2 f_i(x^*))_{jl}|^2 \leq d^2 L_\infty^2 \frac{\mu^2}{(H + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2} \leq \frac{\mu^2}{64},$$

which implies $\mathcal{H}^0 \leq \frac{\mu^2}{64}$. Then from (30), we have

$$\|x^1 - x^*\|^2 \leq \mathcal{W}^0 \leq \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2}.$$

Next we prove $\|x^k - x^*\|^2 \leq \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2}$ for all $k \geq 1$ by induction.
Assume $\|x^k - x^*\|^2 \leq \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2}$ for $k \leq K$. Then since $(\mathbf{H}_i^k)_{jl}$ is a convex combination of $\{(\nabla^2 f_i(w_i^0))_{jl}, (\nabla^2 f_i(x^1))_{jl}, ..., (\nabla^2 f_i(x^k))_{jl}\}$, from the convexity of $|\cdot|^2$, we have

$$|(\mathbf{H}_i^k - \nabla^2 f_i(x^*))_{jl}|^2 \leq L_\infty^2 \cdot \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2} \leq \frac{\mu^2}{64 d^2},$$

for $k \leq K$. Therefore, $\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 \leq \frac{\mu^2}{64}$ and $\mathcal{H}^k \leq \frac{\mu^2}{64}$ for $k \leq K$. Furthermore, from $\mathcal{W}^k \leq \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2}$ for all $k \leq K$ and (30), we can also obtain

$$\|x^{K+1} - x^*\|^2 \leq \mathcal{W}^K \leq \frac{\mu^2}{(L_* + 2L_{\mathrm{F}})^2 + 64 d^2 L_\infty^2}.$$

E. Extension: Globalization via Line Search (FedNL-LS)

Next two extensions of FedNL is to incorporate globalization strategy. Our first globalization technique is based on backtracking line search described in FedNL-LS below.

Algorithm 3 FedNL-LS (Federated Newton Learn with Line Search)

1: **Parameters:** Hessian learning rate $\alpha \geq 0$; compression operators $\{\mathcal{C}_1^k, \ldots, \mathcal{C}_n^k\}$; line search parameters $c \in (0, 1/2]$ and $\gamma \in (0,1)$
2: **Initialization:** $x^0 \in \mathbb{R}^d$; $\mathbf{H}_1^0, \ldots, \mathbf{H}_n^0 \in \mathbb{R}^{d\times d}$ and $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$
3: **for** each device $i = 1, \ldots, n$ in parallel **do**
4: Get x^k from the server; compute $f_i(x^k)$, $\nabla f_i(x^k)$ and $\nabla^2 f_i(x^k)$
5: Send $f_i(x^k)$, $\nabla f_i(x^k)$ and $\mathbf{S}_i^k := \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$ to the server
6: Update local Hessian shifts $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$
7: **end for**
8: **on** server
9: Get $f_i(x^k)$, $\nabla f_i(x^k)$ and $\mathbf{S}_i^k$ from all devices $i \in [n]$
10: $f(x^k) = \frac{1}{n}\sum_{i=1}^n f_i(x^k)$, $\nabla f(x^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^k)$, $\mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k$
11: Compute search direction $d^k = -\left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$
12: Find the smallest integer $s \geq 0$ satisfying $f(x^k + \gamma^s d^k) \leq f(x^k) + c\gamma^s \langle \nabla f(x^k), d^k \rangle$
13: Update global model to $x^{k+1} = x^k + \gamma^s d^k$
14: Update global Hessian shift to $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$

E.1. Line search procedure

In contrast to the vanilla FedNL, here we do not follow the direction $d^k = -\left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$ with unit step size. Instead, FedNL-LS aims to select some step size which would guarantee sufficient decrease in the empirical loss. Thus, we fix the direction d_k (see line 11 of Algorihtm 3) of next iterate x^{k+1}, but want to adjust the step size along that direction. With parameters $c \in (0, 1/2]$ and $\gamma \in (0,1)$, we choose the largest step size of the form γ^s, which leads to a sufficient decrease in the loss $f(x^k + \gamma^s d^k) \leq f(x^k) + c\gamma^s \langle \nabla f(x^k), d^k \rangle$ (see line 12). Note that this procedure requires computation of local functions f_i for all devices $i \in [n]$ in order to do the step in line 12. One the other hand, communication cost of line search procedure is extremely cheap compared to communication cost of gradients and Hessians.

E.2. Local convergence theory

We provide global linear convergence analysis for FedNL-LS. Despite the fact that theoretical rate is slower than the rate of GD, it shows excellent results in experiments. By L-smoothness we assume Lipschitz continuity of gradients with Lipschitz constant L.

Theorem E.1. *Let Assumption 3.1 hold, function f be L-smooth and assume $\widetilde{L} := \sup_{k\geq 0} \|\mathbf{H}^k\|$ is finite. Then convergence of FedNL-LS is linear with the following rate*

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\min\left\{\frac{\mu}{\widetilde{L}}, 1\right\}\right)^k \left(f(x^0) - f(x^*)\right) \tag{34}$$

Next, we provide upper bounds for $\widetilde{L}$, which was assumed to be finite in Theorem E.1.

Lemma E.2. *If Assumption 3.4 holds, then $\widetilde{L} \leq \|\nabla^2 f(x^*)\| + \|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \sqrt{\frac{B}{A}} L_{\mathrm{F}} R$. If Assumption 3.5 holds, then $\widetilde{L} \leq dL_\infty R + \|\nabla^2 f(x^*)\|$.*

E.3. Proof of Theorem E.1

Denote $\kappa := \frac{L}{\mu}$. Using L-smoothness of f we get

$$\begin{aligned} f\left(x^k + \frac{1}{\kappa}d^k\right) &\leq f(x^k) + \frac{1}{\kappa}\langle \nabla f(x^k), d^k\rangle + \frac{L}{2\kappa^2}\left\|d^k\right\|^2 \\ &= f(x^k) - \frac{1}{\kappa}\langle \nabla f(x^k), \left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\rangle + \frac{L}{2\kappa^2}\left\|\left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\right\|^2 \\ &\leq f(x^k) - \frac{1}{\kappa}\langle \nabla f(x^k), \left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\rangle + \frac{L}{2\mu\kappa^2}\langle \nabla f(x^k), \left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\rangle \\ &= f(x^k) - \frac{1}{2\kappa}\langle \nabla f(x^k), \left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\rangle. \end{aligned}$$

From this we conclude that, if $c = \gamma = \frac{1}{2}$, then line search procedure needs at most $s \leq \log_2 \kappa$ steps. To continue the above chain of derivations, we need to upper bound shifts $\mathbf{H}_\mu^k$ in spectral norm.

Notice that if $\mathbf{H}^k$ has at least on eigenvalue larger than μ, then clearly $\|\mathbf{H}_\mu^k\| = \|\mathbf{H}^k\|$. Otherwise, if all eigenvalues do not exceed μ, then projection gives $\mathbf{H}_\mu^k = \mu\mathbf{I}$. Thus, in both cases we can state that $\|\mathbf{H}_\mu^k\| \leq \max\{\|\mathbf{H}^k\|, \mu\} \leq \max\{\widetilde{L}, \mu\}$. Hence

$$\begin{aligned} f\left(x^k + \frac{1}{\kappa}d^k\right) &\leq f(x^k) - \frac{1}{2\kappa}\langle \nabla f(x^k), \left[\mathbf{H}_\mu^k\right]^{-1}\nabla f(x^k)\rangle \\ &\leq f(x^k) - \frac{1}{2\kappa}\frac{1}{\max\{\widetilde{L}, \mu\}}\|\nabla f(x^k)\|^2 \\ &= f(x^k) - \frac{1}{2\kappa}\frac{1}{\max\{\widetilde{L}, \mu\}}\|\nabla f(x^k) - \nabla f(x^*)\|^2 \\ &\leq f(x^k) - \frac{1}{\kappa}\frac{\mu}{\max\{\widetilde{L}, \mu\}}\left(f(x^k) - f(x^*)\right). \end{aligned}$$

Taking $x^{k+1} = x^k + \frac{1}{\kappa}d^k$, subtracting both sides by $f(x^*)$ and unraveling the above recurrence, we get (34).

E.4. Proof of Lemma E.2

Recall that $R = \sup\{\|x - x^*\| \colon f(x) \leq f(x^0)\}$. It follows from the line search procedure that function values are non-increasing, namely $f(x^{k+1}) \leq f(x^k) \leq f(x^0)$. Hence $\|x^k - x^*\| \leq R$ for all $k \geq 0$. Denote

$$\tilde{l}^k := \frac{1}{n}\sum_{i=1}^n \tilde{l}_i^k, \quad \tilde{l}_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}.$$

Consider the case when compressors $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and the learning rate is either $\alpha = 1 - \sqrt{1-\delta}$ or $\alpha = 1$. Using Lemma C.1 with $y = x^k$ and $z = x^*$, for both cases we get

$$\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 \leq (1 - A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + BL_{\mathrm{F}}^2\|x^k - x^*\|^2. \tag{35}$$

Reusing (35) multiple times we get

$$\begin{aligned} \|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 &\leq (1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + BL_{\mathrm{F}}^2R^2 \\ &\leq (1-A)^2\|\mathbf{H}_i^{k-1} - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + \left[1 + (1-A)\right]BL_{\mathrm{F}}^2R^2 \\ &\leq (1-A)^{k+1}\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + BL_{\mathrm{F}}^2R^2\sum_{t=0}^\infty (1-A)^t \\ &\leq \|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + \frac{B}{A}L_{\mathrm{F}}^2R^2, \end{aligned}$$

which implies boundedness of $\tilde{l}^k$:

$$\tilde{l}^k = \frac{1}{n}\sum_{i=1}^n \tilde{l}_i^k \leq \frac{1}{n}\sum_{i=1}^n \sqrt{\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 + \frac{B}{A}L_{\mathrm{F}}^2R^2} \leq \|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \sqrt{\frac{B}{A}}L_{\mathrm{F}}R.$$

From this we also conclude boundedness of $\widetilde{L}$ as follows

$$\begin{aligned}
\|\mathbf{H}^k\| &\leq \|\mathbf{H}^k - \nabla^2 f(x^*)\| + \|\nabla^2 f(x^*)\| \\
&\leq \left\| \frac{1}{n}\sum_{i=1}^n (\mathbf{H}_i^k - \nabla^2 f_i(x^*)) \right\|_{\mathrm{F}} + \|\nabla^2 f(x^*)\| \\
&\leq \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \|\nabla^2 f(x^*)\| \\
&\leq \|\nabla^2 f(x^*)\| + \|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \sqrt{\frac{B}{A}} L_{\mathrm{F}} R.
\end{aligned}$$

Consider the case when compressors $\mathcal{C}_i^k \in \mathbb{B}(\omega)$ and the learning rate $\alpha \leq \frac{1}{\omega+1}$. As we additionally assume that $(\mathbf{H}_i^k)_{jl}$ is a convex combination of past Hessians $\{(\nabla^2 f_i(x^0))_{jl}, \ldots, (\nabla^2 f_i(x^k))_{jl}\}$, we get

$$|(\mathbf{H}_i^k - \nabla^2 f_i(x^*))_{jl}|^2 \leq L_\infty^2 \max_{0\leq t\leq k} \|x^t - x^*\|^2 \leq L_\infty^2 R^2.$$

Therefore

$$\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2 \leq d^2 L_\infty^2 R^2,$$

from which

$$\|\mathbf{H}^k\| \leq \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}} + \|\nabla^2 f(x^*)\| \leq d L_\infty R + \|\nabla^2 f(x^*)\|.$$

F. Extension: Globalization via Cubic Regularization (FedNL-CR)

Our next extension to FedNL providing global convergence guarantees is cubic regularization.

Algorithm 4 FedNL-CR (Federated Newton Learn with Cubic Regularization)

Parameters: Hessian learning rate $\alpha \geq 0$; compression operators $\{\mathcal{C}_1^k, \ldots, \mathcal{C}_n^k\}$; Lipschitz constant $L_* \geq 0$ for Hessians
Initialization: $x^0 \in \mathbb{R}^d$; $\mathbf{H}_1^0, \ldots, \mathbf{H}_n^0 \in \mathbb{R}^{d\times d}$ and $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$
for each device $i = 1, \ldots, n$ in parallel **do**
  Get x^k from the server and compute local gradient $\nabla f_i(x^k)$ and local Hessian $\nabla^2 f_i(x^k)$
  Send $\nabla f_i(x^k)$, $\mathbf{S}_i^k := \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$ and $l_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^k)\|_{\mathrm{F}}$ to the server
  Update local Hessian shift to $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$
end for
on server
  Get $\nabla f_i(x^k)$, $\mathbf{S}_i^k$ and l_i^k from all devices $i \in [n]$
  $\nabla f(x^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^k), \quad \mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k, \quad l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$
  $h^k = \arg\min_{h\in\mathbb{R}^d} T_k(h)$, where $T_k(h) := \left\langle \nabla f(x^k), h \right\rangle + \frac{1}{2}\left\langle (\mathbf{H}^k + l^k \mathbf{I})h, h \right\rangle + \frac{L_*}{6}\|h\|^3$
  Update global model to $x^{k+1} = x^k + h^k$
  Update global Hessian shift to $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$

F.1. Cubic regularization

Adding third order regularization term $\frac{L_*}{6}\|h\|^3$ is a well known technique to guarantee global convergence for Newton-type methods. Basically, this term provides means to upper bound the loss function globally, which ultimately leads to global convergence. Notice that, without this term FedNL-CR reduces to FedNL with *Option 2*. However, cubic regularization alone does not provide us global upper bounds as the second order information, the Hessians, are compressed, and thus upper bounds might be violated.

F.2. Solving the subproblem

In each iteration, the sever needs to solve the subproblem in line 11 in order to compute h^k. Although it does not admit a closed form solution, the server can solve it by reducing to certain one-dimensional nonlinear equation. For more details, see section C.1 of (Islamov et al., 2021).

F.3. Importance of compression errors l_i^k

Unlike FedNL and FedNL-PP, compression errors are the only option for FedNL-CR to update the global model. The reason is that to get a cubic upper bound for f we need to upper bound current true Hessians $\nabla^2 f_i(x^k)$ in the matrix order. Neither current learned Hessian $\mathbf{H}_i^k$ nor the projected matrix $\left[\mathbf{H}_i^k\right]_\mu$ does not guarantee upper bound for $\nabla^2 f_i(x^k)$. Meanwhile, from $l_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^k)\|_{\rm F}$, we have $\nabla^2 f_i(x^k) \preceq \mathbf{H}_i^k + l_i^k \mathbf{I}$.

F.4. Global and local convergence theory

We prove two global rates (covering convex and strongly convex cases) and the same three local rates of FedNL.

Theorem F.1. *Let Assumption 3.1 hold and assume $l := \sup_{k\geq 0} l^k$ is finite. Then if $f(x)$ is convex (i.e., $\mu = 0$), we have global sublinear rate*

$$f(x^k) - f(x^*) \leq \frac{9lR^2}{k} + \frac{9L_* R^3}{k^2} + \frac{3\left(f(x^0) - f(x^*)\right)}{k^3}, \tag{36}$$

where $R := \{\|x - x^*\|\colon f(x) \leq f(x^0)\}$. Moreover, if $f(x)$ is μ-convex with $\mu > 0$, then convergence becomes linear with respect to function sub-optimality, i.e., $f(x^k) - f(x^*) \leq \varepsilon$ is guaranteed after

$$\mathcal{O}\left(\left(\frac{l}{\mu} + \sqrt{\frac{L_* R}{\mu}} + 1\right) \log \frac{f(x^0) - f(x^*)}{\varepsilon}\right) \tag{37}$$

iterations. Furthermore, if $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{20(L_*^2 + 8L_{\rm F}^2)}$ and $\mathcal{H}^k \leq \frac{\mu^2}{160}$ for all $k \geq 0$, then we have the same local rates (7), (8) and (9).

Next, we provide upper bounds for l, which was assumed to be finite in the theorem.

Lemma F.2. *If Assumption 3.4 holds, then $l \leq \sqrt{\mathcal{H}^0} + \left(1 + \sqrt{\frac{B}{A}}\right) L_{\rm F} R$. If Assumption 3.5 holds, then $l \leq (dL_\infty + L_{\rm F})R$.*

F.5. Proof of Theorem F.1

Global rate for general convex case ($\mu = 0$). First, from L_*-Lipschitzness of the Hessian of f we get

$$\begin{aligned}
f(x^{k+1}) &\leq f(x^k) + \left\langle \nabla f(x^k), h^k \right\rangle + \frac{1}{2}\left\langle \nabla^2 f(x^k) h^k, h^k \right\rangle + \frac{L_*}{6}\|h^k\|^3 \\
&\leq f(x^k) + \left\langle \nabla f(x^k), h^k \right\rangle + \frac{1}{2}\left\langle (\mathbf{H}^k + l^k \mathbf{I}) h^k, h^k \right\rangle + \frac{L_*}{6}\|h^k\|^3 \\
&= f(x^k) + \min_{h\in\mathbb{R}^d} T_k(h) \qquad (38) \\
&\leq f(x^k) + T_k(y - x^k) \\
&\leq f(x^k) + \left\langle \nabla f(x^k), y - x^k \right\rangle + \frac{1}{2}\left\langle (\mathbf{H}^k + l^k \mathbf{I})(y - x^k), y - x^k \right\rangle + \frac{L_*}{6}\|y - x^k\|^3 \\
&\leq f(x^k) + \left\langle \nabla f(x^k), y - x^k \right\rangle + \frac{1}{2}\left\langle \nabla^2 f(x^k)(y - x^k), y - x^k \right\rangle + \frac{L_*}{6}\|y - x^k\|^3 \\
&\quad + \frac{1}{2}\left\|\mathbf{H}^k - \nabla^2 f(x^k)\right\| \|y - x^k\|^2 + \frac{1}{2} l^k \|y - x^k\|^2 \\
&\leq f(x^k) + \left\langle \nabla f(x^k), y - x^k \right\rangle + \frac{1}{2}\left\langle \nabla^2 f(x^k)(y - x^k), y - x^k \right\rangle + l^k\|y - x^k\|^2 + \frac{L_*}{6}\|y - x^k\|^3 \\
&\leq f(y) + \frac{L_*}{6}\|y - x^k\|^3 + l^k\|y - x^k\|^2 + \frac{L_*}{6}\|y - x^k\|^3 \\
&\leq f(y) + l\|y - x^k\|^2 + \frac{L_*}{3}\|y - x^k\|^3. \qquad (39)
\end{aligned}$$

Denote $a_k := k^2$ and

$$A_k := 1 + \sum_{i=1}^{k} a_i = 1 + \sum_{i=1}^{k} i^2 = 1 + \frac{k(k+1)(2k+1)}{6} \geq 1 + \frac{k^3}{3}.$$

Let $\sigma_k = \frac{a_{k+1}}{A_{k+1}} \in (0,1)$. Then we get $1 - \sigma_k = \frac{A_k}{A_{k+1}}$. Now we choose $y = \sigma_k x^* + (1-\sigma_k)x^k = x^k + \sigma_k(x^* - x^k)$. Using convexity of f, we get

$$\begin{aligned} f(x^{k+1}) &\leq f(y) + l\|y - x^k\|^2 + \frac{L_*}{3}\|y - x^k\|^3 \\ &\leq \sigma_k f(x^*) + (1-\sigma_k) f(x^k) + l\sigma_k^2\|x^k - x^*\|^2 + \frac{L_*}{3}\sigma_k^3\|x^k - x^*\|^3 \\ &\leq \sigma_k f(x^*) + (1-\sigma_k) f(x^k) + l\sigma_k^2 R^2 + \frac{L_*}{3}\sigma_k^3 R^3. \end{aligned} \tag{40}$$

Using the definition of σ_k and subtracting both sides by $A_k f(x^*)$ we get

$$A_{k+1}\left(f(x^{k+1}) - f(x^*)\right) \leq A_k\left(f(x^k) - f(x^*)\right) + lR^2\frac{a_{k+1}^2}{A_{k+1}} + \frac{L_* R^3}{3}\frac{a_{k+1}^3}{A_{k+1}^2},$$

repeated application of which provides us the following bound

$$A_k\left(f(x^k) - f(x^*)\right) \leq A_0\left(f(x^0) - f(x^*)\right) + lR^2\sum_{t=1}^{k}\frac{a_t^2}{A_t} + \frac{L_* R^3}{3}\sum_{t=0}^{k}\frac{a_t^3}{A_t^2}. \tag{41}$$

Next we upper bound the above two sums:

$$\sum_{t=1}^{k}\frac{a_t^2}{A_t} \leq \sum_{t=1}^{k}\frac{t^4}{1+\frac{t^3}{3}} \leq 3k^2, \quad \sum_{t=1}^{k}\frac{a_t^3}{A_t^2} \leq \sum_{t=1}^{k}\frac{t^6}{\left(1+\frac{t^3}{3}\right)^2} \leq 9k.$$

Hence the bound (41) can be transformed into

$$\begin{aligned} f(x^k) - f(x^*) &\leq \frac{1}{A_k}\left[\left(f(x^0) - f(x^*)\right) + 3k^2\cdot lR^2 + 3k\cdot L_* R^3\right] \\ &\leq \frac{9lR^2}{k} + \frac{9L_* R^3}{k^2} + \frac{3\left(f(x^0) - f(x^*)\right)}{k^3}. \end{aligned}$$

Thus, we have shown $\mathcal{O}(\frac{1}{k})$ rate for convex functions and it holds for any $k \geq 1$.

Global rate for strongly convex case ($\mu > 0$). We can turn this rate into a linear rate using strong convexity of f. Namely, in this case we have $R^2 \leq \frac{2}{\mu}(f(x^0) - f(x^*))$ and therefore

$$f(x^k) - f(x^*) \leq \left[\frac{18l}{k\mu} + \frac{18L_* R}{k^2\mu} + \frac{3}{k^3}\right]\left(f(x^0) - f(x^*)\right) \leq \frac{1}{2}\left(f(x^0) - f(x^*)\right),$$

if $k \geq K_1 := \max\left(\frac{108l}{\mu}, \sqrt{\frac{108L_* R}{\mu}}, 3\right)$. In other words, we half the error $f(x^k) - f(x^*)$ after K_1 steps. This implies the following linear rate

$$\mathcal{O}\left(\left(\frac{l}{\mu} + \sqrt{\frac{L_* R}{\mu}} + 1\right)\log\frac{1}{\varepsilon}\right).$$

Local rate for strongly convex case ($\mu > 0$).

From the definition of h^k direction, we have

$$\nabla T_k(h^k) = \nabla f(x^k) + (\mathbf{H}^k + l^k\mathbf{I})h^k + \frac{L_*}{2}\|h^k\|h^k = 0,$$

which implies the following equivalent update rule

$$\begin{aligned} x^{k+1} &= x^k + h^k \\ &= x^k - \left[\mathbf{H}^k + l^k\mathbf{I} + \frac{L_*}{2}\|x^{k+1} - x^k\|\right]^{-1} \nabla f(x^k). \end{aligned}$$

Then, using $\mu\mathbf{I} \preceq \nabla f(x^k) \preceq \mathbf{H}^k + l^k\mathbf{I}$, we have

$$\begin{aligned} & \|x^{k+1} - x^*\|^2 \\ = & \left\| x^k - x^* - \left[\mathbf{H}^k + l^k\mathbf{I} + \frac{L_*}{2}\|x^{k+1} - x^k\|\right]^{-1} \nabla f(x^k) \right\|^2 \\ \leq & \frac{1}{\mu^2} \left\| \left[\mathbf{H}^k + l^k\mathbf{I} + \frac{L_*}{2}\|x^{k+1} - x^*\| + \frac{L_*}{2}\|x^k - x^*\|\right](x^k - x^*) - \nabla f(x^k) \right\|^2 \\ \leq & \frac{5}{\mu^2} \left(\left\|\nabla^2 f(x^k)(x^k - x^*) - \nabla f(x^k) + \nabla f(x^*)\right\|^2 + \frac{L_*^2}{4}\|x^k - x^*\|^4 + \frac{L_*^2}{4}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 \right. \\ & \left. + \left\|\left(\mathbf{H}^k - \nabla^2 f(x^k)\right)(x^k - x^*)\right\|^2 + \left[l^k\right]^2 \|x^k - x^*\|^2 \right) \\ \leq & \frac{5}{\mu^2} \left(\frac{L_*^2}{4}\|x^k - x^*\|^4 + \frac{L_*^2}{4}\|x^k - x^*\|^4 + \frac{L_*^2}{4}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 \right. \\ & \left. + \left\|\mathbf{H}^k - \nabla^2 f(x^k)\right\|^2 \|x^k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^n \left\|\mathbf{H}_i^k - \nabla f_i(x^k)\right\|_{\rm F}^2 \|x^k - x^*\|^2 \right) \\ \leq & \frac{5}{\mu^2} \left(\frac{L_*^2}{2}\|x^k - x^*\|^4 + \frac{L_*^2}{4}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 + \frac{2}{n}\sum_{i=1}^n \left\|\mathbf{H}_i^k - \nabla f_i(x^k)\right\|_{\rm F}^2 \|x^k - x^*\|^2 \right) \\ \leq & \frac{5L_*^2}{2\mu^2}\|x^k - x^*\|^4 + \frac{5L_*^2}{4\mu^2}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 + \frac{20}{n\mu^2}\sum_{i=1}^n \left\|\mathbf{H}_i^k - \nabla f_i(x^*)\right\|_{\rm F}^2 \|x^k - x^*\|^2 \\ & + \frac{20}{n\mu^2}\sum_{i=1}^n \left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|_{\rm F}^2 \|x^k - x^*\|^2 \\ \leq & \frac{5L_*^2}{2\mu^2}\|x^k - x^*\|^4 + \frac{5L_*^2}{4\mu^2}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 + \frac{20}{\mu^2}\|x^k - x^*\|^2\mathcal{H}^k + \frac{20L_{\rm F}^2}{\mu^2}\|x^k - x^*\|^4 \\ \leq & \frac{5L_*^2}{4\mu^2}\|x^{k+1} - x^*\|^2\|x^k - x^*\|^2 + \frac{20}{\mu^2}\|x^k - x^*\|^2\mathcal{H}^k + \frac{5(L_*^2 + 8L_{\rm F}^2)}{2\mu^2}\|x^k - x^*\|^4. \end{aligned} \tag{42}$$

Using the assumptions we show that $\|x^k - x^*\|^2 \leq \frac{\mu^2}{20(L_*^2 + 8L_{\rm F}^2)}$ for all $k \geq 0$. We prove this again by induction on k. From $\|x^k - x^*\|^2 \leq \frac{\mu^2}{20(L_*^2 + 8L_{\rm F}^2)} \leq \frac{2\mu^2}{5L_*^2}$ and $\mathcal{H}^k \leq \frac{\mu^2}{160}$, it follows

$$\begin{aligned} & \|x^{k+1} - x^*\|^2 \\ \leq & \frac{5L_*^2}{4\mu^2}\|x^k - x^*\|^2\|x^{k+1} - x^*\|^2 + \frac{20}{\mu^2}\mathcal{H}^k\|x^k - x^*\|^2 + \frac{5(L_*^2 + 8L_{\rm F}^2)}{2\mu^2}\|x^k - x^*\|^2\|x^k - x^*\|^2 \\ \leq & \frac{1}{2}\|x^{k+1} - x^*\|^2 + \frac{1}{8}\|x^k - x^*\|^2 + \frac{1}{8}\|x^k - x^*\|^2 \\ \leq & \frac{1}{2}\|x^{k+1} - x^*\|^2 + \frac{1}{4}\|x^k - x^*\|^2. \end{aligned}$$

Hence

$$\|x^{k+1} - x^*\|^2 \leq \frac{1}{2}\|x^k - x^*\|^2 \leq \frac{\mu^2}{20(L_*^2 + 8L_{\rm F}^2)}. \tag{43}$$

By this we complete the induction and also derived the local linear rate for iterates. Moreover, (42) and (43) imply

$$\|x^{k+1} - x^*\|^2 \leq \frac{20}{\mu^2}\|x^k - x^*\|^2\mathcal{H}^k + \frac{3L_*^2 + 20L_{\rm F}^2}{\mu^2}\|x^k - x^*\|^4. \tag{44}$$

Choosing $y = x^k$ and $z = x^*$ in Lemma C.1, and noting that $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$, we get

$$\mathbb{E}_k\left[\mathcal{H}^{k+1}\right] \le (1-A)\mathcal{H}^k + BL_{\rm F}^2\|x^k - x^*\|^2.$$

Using the same Lyapunov function $\Phi^k = \mathcal{H}^k + 6BL_{\rm F}^2\|x^k - x^*\|^2$, from the above inequality and (43), we arrive at

$$\begin{aligned}\mathbb{E}_k\left[\Phi^{k+1}\right] &\le (1-A)\mathcal{H}^k + BL_{\rm F}^2\|x^k - x^*\|^2 + 3BL_{\rm F}^2\|x^k - x^*\|^2\\ &= (1-A)\mathcal{H}^k + \left(1-\frac{1}{3}\right)6BL_{\rm F}^2\|x^k - x^*\|^2\\ &\le \left(1-\min\left\{A, \frac{1}{3}\right\}\right)\Phi^k.\end{aligned}$$

Hence $\mathbb{E}[\Phi^k] \le \left(1-\min\left\{A,\frac{1}{3}\right\}\right)^k\Phi^0$. We further have $\mathbb{E}[\mathcal{H}^k] \le \left(1-\min\left\{A,\frac{1}{3}\right\}\right)^k\Phi^0$ and $\mathbb{E}[\|x^k - x^*\|^2] \le \frac{1}{6BL_{\rm F}^2}\left(1-\min\left\{A,\frac{1}{3}\right\}\right)^k\Phi^0$ for $k \ge 0$. Assume $x^k \neq x^*$ for all k. Then from (44), we have

$$\frac{\|x^{k+1}-x^*\|^2}{\|x^k - x^*\|^2} \le \frac{20}{\mu^2}\mathcal{H}^k + \frac{3L_*^2 + 20L_{\rm F}^2}{\mu^2}\|x^k - x^*\|^2,$$

and by taking expectation, we have

$$\begin{aligned}\mathbb{E}\left[\frac{\|x^{k+1}-x^*\|^2}{\|x^k - x^*\|^2}\right] &\le \frac{20}{\mu^2}\mathbb{E}[\mathcal{H}^k] + \frac{3L_*^2 + 20L_{\rm F}^2}{\mu^2}\mathbb{E}[\|x^k - x^*\|^2]\\ &\le \left(1-\min\left\{A,\frac{1}{3}\right\}\right)^k\left(20 + \frac{3L_*^2 + 20L_{\rm F}^2}{6BL_{\rm F}^2}\right)\frac{\Phi^0}{\mu^2}.\end{aligned}$$

To conclude, FedNL-CR method provably provides global rates (both for convex and strongly convex cases) and recovers the same local rates (7), (8) and (9) that we showed for FedNL. Note that constants A and B are the same, while C and D differ from (9).

F.6. Proof of Lemma F.2

Recall that $R = \sup\{\|x - x^*\|\colon f(x) \le f(x^0)\}$. Since $T_k(0) = 0$, from (38) we can show that $f(x^{k+1}) \le f(x^k) \le f(x^0)$, and hence $\|x^k - x^*\| \le R$ for all $k \ge 0$. Denote

$$\tilde{l}^k := \frac{1}{n}\sum_{i=1}^n \tilde{l}_i^k, \qquad \tilde{l}_i^k := \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}.$$

Notice that

$$\begin{aligned} l_i^k &= \|\nabla^2 f_i(x^k) - \mathbf{H}_i^k\|_{\rm F}\\ &\le \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F} + \|\nabla^2 f_i(x^k) - \nabla^2 f_i(x^*)\|_{\rm F}\\ &\le \tilde{l}_i^k + L_{\rm F}\|x^k - x^*\|\\ &\le \tilde{l}_i^k + L_{\rm F}R. \end{aligned} \tag{45}$$

Consider the case when compressors $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ and the learning rate is either $\alpha = 1 - \sqrt{1-\delta}$ or $\alpha = 1$. Using Lemma C.1 with $y = x^k$ and $z = x^*$, for both cases we get

$$\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 \le (1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2\|x^k - x^*\|^2. \tag{46}$$

Reusing (46) multiple times we get

$$\begin{aligned}\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 &\le (1-A)\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2R^2\\ &\le (1-A)^2\|\mathbf{H}_i^{k-1} - \nabla^2 f_i(x^*)\|_{\rm F}^2 + [1 + (1-A)]\,BL_{\rm F}^2R^2\\ &\le (1-A)^{k+1}\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 + BL_{\rm F}^2R^2\sum_{t=0}^{\infty}(1-A)^t\\ &\le \|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{B}{A}L_{\rm F}^2R^2,\end{aligned}$$

which implies boundedness of $\tilde{l}_i^k$:

$$\tilde{l}_i^k \leq \sqrt{\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\rm F}^2 + \frac{B}{A} L_{\rm F}^2 R^2} \leq \tilde{l}_i^0 + \sqrt{\frac{B}{A}} L_{\rm F} R.$$

From this we also conclude boundedness of l^k as follows

$$l^k = \frac{1}{n}\sum_{i=1}^n l_i^k \overset{(45)}{\leq} \frac{1}{n}\sum_{i=1}^n \tilde{l}_i^k + L_{\rm F} R \leq \tilde{l}^0 + \left(1 + \sqrt{\frac{B}{A}}\right) L_{\rm F} R.$$

We can further upper bound $\tilde{l}^0 \leq \sqrt{\mathcal{H}^0}$ and conclude $l \leq \sqrt{\mathcal{H}^0} + \left(1 + \sqrt{\frac{B}{A}}\right) L_{\rm F} R$.

Consider the case when compressors $\mathcal{C}_i^k \in \mathbb{B}(\omega)$ and the learning rate $\alpha \leq \frac{1}{\omega+1}$. As we additionally assume that $(\mathbf{H}_i^k)_{jl}$ is a convex combination of past Hessians $\{(\nabla^2 f_i(x^0))_{jl}, \dots, (\nabla^2 f_i(x^k))_{jl}\}$, we get

$$|(\mathbf{H}_i^k - \nabla^2 f_i(x^*))_{jl}|^2 \leq L_\infty^2 \max_{0 \leq t \leq k} \|x^t - x^*\|^2 \leq L_\infty^2 R^2.$$

Therefore

$$\left[\tilde{l}_i^k\right]^2 = \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2 \leq d^2 L_\infty^2 R^2,$$

from which

$$l^k = \frac{1}{n}\sum_{i=1}^n l_i^k \overset{(45)}{\leq} \frac{1}{n}\sum_{i=1}^n \tilde{l}_i^k + L_{\rm F} R \leq d L_\infty R + L_{\rm F} R = (d L_\infty + L_{\rm F}) R.$$

G. Extension: Bidirectional Compression (FedNL-BC)

Finally, we extend the vanilla FedNL to allow for an even more severe level of compression that can't be attained by compressing the Hessians only. This is achieved by compressing the gradients (uplink) and the model (downlink), in a "smart" way. Thus, in FedNL-BC (Algorithm 5) described below, both directions of communication are fully compressed.

Algorithm 5 FedNL-BC (Federated Newton Learn with Bidirectional Compression)

1: **Parameters:** Hessian learning rate $\alpha \geq 0$; model learning rate $\eta \geq 0$; gradient compression probability $p \in (0, 1]$; compression operators $\{\mathcal{C}_1^k, \ldots, \mathcal{C}_n^k\}$ and $\mathcal{C}_{\mathrm{M}}^k$
2: **Initialization:** $x^0 = w^0 = z^0 \in \mathbb{R}^d$; $\mathbf{H}_1^0, \ldots, \mathbf{H}_n^0 \in \mathbb{R}^{d \times d}$ and $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$; $\xi^0 = 1$
3: **for** each device $i = 1, \ldots, n$ in parallel **do**
4: Get ξ^k from the server
5: **if** $\xi^k = 1$
6: Compute local gradient $\nabla f_i(z^k)$ and send to the server
7: $g_i^k = \nabla f_i(z^k), \quad w^{k+1} = z^k$
8: **if** $\xi^k = 0$
9: $g_i^k = \mathbf{H}_i^k(z^k - w^k) + \nabla f_i(w^k), \quad w^{k+1} = w^k$
10: Compute local Hessian $\nabla^2 f_i(z^k)$
11: Send $\mathbf{S}_i^k := \mathcal{C}_i^k(\nabla^2 f_i(z^k) - \mathbf{H}_i^k)$ and $l_i^k := \|\nabla^2 f_i(z^k) - \mathbf{H}_i^k\|_{\mathrm{F}}$ to the server
12: Update local Hessian shift to $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$
13: **end for**
14: **on** server
15: $g^k = \frac{1}{n}\sum\limits_{i=1}^n g_i^k, \quad \mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k, \quad l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$
16: *Option 1:* $x^{k+1} = z^k - \left[\mathbf{H}^k\right]_\mu^{-1} g^k$
17: *Option 2:* $x^{k+1} = z^k - \left[\mathbf{H}^k + l^k \mathbf{I}\right]^{-1} g^k$
18: Update global Hessian shifts $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$
19: Send $s^k := \mathcal{C}_{\mathrm{M}}^k(x^{k+1} - z^k)$ to all devices $i \in [n]$
20: Update the model $z^{k+1} = z^k + \eta s^k$
21: Sample $\xi^{k+1} \sim \mathrm{Bernoulli}(p)$ and send to all devices $i \in [n]$
22: **for** each device $i = 1, \ldots, n$ in parallel **do**
23: Get s^k from the server and update the model $z^{k+1} = z^k + \eta s^k$
24: **end for**

G.1. Model learning technique

In FedNL-BC we introduced “smart" model learning technique, which is similar to the proposed Hessian learning technique. As in Hessian learning technique, the purpose of the model learning technique is learn the optimal model x^* in a communication efficient manner. This is achieved by maintaining and progressively updating global model estimates z^k for all nodes $i \in [n]$ and for the sever. Thus, the goal is to make updates from z^k to z^{k+1} easy to communicate and to induce $z^k \to x^*$ throughout the training process. Similar to the Hessian learning technique, the server operates its own compressors $\mathcal{C}_{\mathrm{M}}^k$ and updates the model estimates z^k via the rule $z^{k+1} = z^k + \eta s^k$, where $s^k = \mathcal{C}_{\mathrm{M}}^k(x^{k+1} - z^k)$ and $\eta > 0$ is the learning rate. Again, we reduce the communication cost by explicitly requiring the server to send compressed model information s^k to all clients.

G.2. Hessian corrected local gradients

The second key technical novelty in FedNL-PP is another structure of Hessian corrected local gradients

$$g_i^k = \mathbf{H}_i^k(z^k - w^k) + \nabla f_i(w^k)$$

(see line 9 of Algorithm 5). The intuition behind this form is as follows. Uplink gradient compression is done by Bernoulli compression synchronized by the server: namely, if the Bernoulli trial $\xi^k \sim \mathrm{Bernoulli}(p)$ is successful (i.e., $\xi^k = 1$, see line 5), then all clients compute and communicate the current true local gradients $\nabla f_i(z^k)$, otherwise (i.e., $\xi^k = 0$, see line 8) devices do not even compute the local gradient. In the latter case, devices approximate current local gradient $\nabla f_i(z^k)$ based on stale local gradient $\nabla f_i(w^k)$ and current Hessian estimate $\mathbf{H}_i^k$ via the rule $g_i^k = \mathbf{H}_i^k(z^k - w^k) + \nabla f_i(w^k)$, w^k is the last learned global model when Bernoulli trial was successful and local gradients are sent to the server.

To further motivate the structure of g_i^k, consider for a moment the case when $\mathbf{H}_i^k = \nabla^2 f_i(x^k)$. Then $g_i^k = \nabla^2 f_i(x^k)(z^k -$

$w^k) + \nabla f_i(w^k)$ is, indeed, approximates $\nabla f(z^k)$ as

$$\|\nabla f(z^k) - \nabla f_i(w^k) - \nabla^2 f_i(x^k)(z^k - w^k)\| \leq \frac{L_*}{2}\|z^k - w^k\|^2 \leq L_*\|z^k - x^*\|^2 + L_*\|w^k - x^*\|^2.$$

G.3. Local convergence theory

Similar to Assumptions 3.4 and 3.5, we need one of the following assumptions related to the compression done by the master.

Assumption G.1. Compressors $\mathcal{C}_{\mathrm{M}}^k \in \mathbb{C}(\delta_{\mathrm{M}})$ and learning rate (i) $\eta = 1 - \sqrt{1-\delta_{\mathrm{M}}}$ or (ii) $\delta_{\mathrm{M}} = 1$.

Assumption G.2. Compressors $\mathcal{C}_{\mathrm{M}}^k \in \mathbb{B}(\omega_{\mathrm{M}})$, learning rate $0 < \eta \leq \frac{1}{\omega_{\mathrm{M}}+1}$. Moreover, for all $j \in [d]$, each entry $(z^k)_j$ is a convex combination of $\{(x^t)_j\}_{t=0}^k$ for any $k \geq 0$.

Note that Assumption 3.5 assumes that $(\mathbf{H}_i^k)_{jl}$ is a convex combination of $\{(\nabla^2 f_i(x^t))_{jl}\}_{t=0}^k$ as the Hessian learning technique is based on exact Hessians $\nabla^2 f_i(x^k)$ at x^k. However, in FedNL-BC, the Hessian learning technique is based on Hessians $\nabla^2 f_i(z^k)$ at z^k. Hence, it makes sense to adapt Assumption 3.5 and assume that $(\mathbf{H}_i^k)_{jl}$ is a convex combination of $\{(\nabla^2 f_i(z^t))_{jl}\}_{t=0}^k$.

Moreover, we need alternatives to constants $A,\ B,\ C,\ D$ in this case, which we denote by $A_{\mathrm{M}},\ B_{\mathrm{M}},\ C_{\mathrm{M}},\ D_{\mathrm{M}}$ and define as follows

$$(A_{\mathrm{M}}, B_{\mathrm{M}}) := \begin{cases} (\eta^2, \eta) & \text{if Assumption } G.1\text{(i) holds} \\ (\frac{\delta_{\mathrm{M}}}{4}, \frac{6}{\delta_{\mathrm{M}}} - \frac{7}{2}) & \text{if Assumption } G.1\text{(ii) holds} \\ (\eta, \eta) & \text{if Assumption } G.2 \text{ holds} \end{cases} \tag{47}$$

$$(C_{\mathrm{M}}, D_{\mathrm{M}}) := \begin{cases} (24, 8L_{\mathrm{F}}^2 + 9/4 L_*^2) & \text{if } \textit{Option 1} \text{ is used in FedNL-BC} \\ (32, 16L_{\mathrm{F}}^2 + 9/4 L_*^2) & \text{if } \textit{Option 2} \text{ is used in FedNL-BC} \end{cases}. \tag{48}$$

Following the same steps of Lemma C.1, one can show the following lemma for different compressors applied by the master to handle $\mathbb{E}_k\left[\|z^k + \eta\mathcal{C}_{\mathrm{M}}^k(u - z^k) - v\|^2\right]$ term, where $\mathbb{E}_k[u] = u$ and $\mathbb{E}_k[v] = v$.

Lemma G.3. *For any $u,\, v \in \mathbb{R}^d$ such that $\mathbb{E}_k[u] = u$ and $\mathbb{E}_k[v] = v$, we have the following result combining three different cases from (47):*

$$\mathbb{E}_k\|z^k + \eta\mathcal{C}_{\mathrm{M}}^k(u - z^k) - v\|^2 \leq (1 - A_{\mathrm{M}})\|z^k - v\|^2 + B_{\mathrm{M}}\|u - v\|^2.$$

The proof of Lemma G.3 can be obtained by repeating the proof of Lemma C.1 with small modifications. Denote

$$r_k := \left\|x^k - x^*\right\|^2, \quad \nu_k := \left\|w^k - x^*\right\|^2, \quad \gamma_k = \left\|z^k - x^*\right\|^2.$$

$$E_1 := 16L_{\mathrm{F}}^2,\ E_2 := 16,\ E_3 := 16L_{\mathrm{F}}^2 + 8L_*^2.$$

We prove local linear rate for Lyapunov function $\Phi^k := \|z^k - x^*\|^2 + \frac{A_{\mathrm{M}}}{3p}\|w^k - x^*\|^2$. As a result, we show that both $z^k \to x^*$ and $w^k \to x^*$ converge locally linearly.

Theorem G.4. *Let Assumption 3.1 hold and assume that $\mathcal{H}^k \leq \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}$ and $\|z^k - x^*\|^2 \leq \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}$ for all $k \geq 0$. Then, we have the following linear rate for FedNL-BC:*

$$\mathbb{E}\left[\Phi^k\right] \leq \left(1 - \min\left\{\frac{A_{\mathrm{M}}}{3}, \frac{p}{2}\right\}\right)^k \Phi^0.$$

We assumed inequalities $\mathcal{H}^k \leq \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}$ and $\|z^k - x^*\|^2 \leq \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}$ hold for all $k \geq 0$. Next we prove these inequalities using initial conditions only.

Lemma G.5. *Let Assumptions 3.4 and G.1 hold. If*

$$\mathcal{H}^0 \leq \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}, \quad \|z^0 - x^*\| \leq \min\left\{\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}, \frac{A}{BL_{\mathrm{F}}^2}\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}\right\},$$

then the same upper bounds hold for all $k \geq 0$, i.e.,

$$\mathcal{H}^k \leq \frac{A_{\rm M}}{B_{\rm M}} \frac{\mu^2}{9C_{\rm M}}, \quad \|z^k - x^*\| \leq \min\left\{ \frac{A_{\rm M}}{B_{\rm M}} \frac{\mu^2}{9E_3}, \frac{A}{BL_{\rm F}^2} \frac{A_{\rm M}}{B_{\rm M}} \frac{\mu^2}{9C_{\rm M}} \right\}.$$

Lemma G.6. *Let Assumptions 3.5 and G.2 hold. If*

$$\|x^0 - x^*\| \leq \min\left\{ \frac{\mu^2}{9d^2E_3}, \frac{\mu^2}{9C_{\rm M}d^4L_\infty^2} \right\},$$

then the following upper bounds hold for all $k \geq 0$, i.e.,

$$\mathcal{H}^k \leq \frac{\mu^2}{9dC_{\rm M}}, \quad \|z^k - x^*\| \leq \min\left\{ \frac{\mu^2}{9dE_3}, \frac{\mu^2}{9C_{\rm M}d^3L_\infty^2} \right\}. \tag{49}$$

G.4. Proof of Theorem G.4

Consider *Option 1* first and expand $\|x^{k+1} - x^*\|^2$:

$$\left\|x^{k+1} - x^*\right\|^2 = \left\| \left[\mathbf{H}_\mu^k\right]^{-1} \left(\mathbf{H}_\mu^k(z^k - x^*) - g^k\right) \right\|^2 \leq \frac{1}{\mu^2} \left\|\mathbf{H}_\mu^k(z^k - x^*) - g^k\right\|^2.$$

Then we decompose the term $\mathbf{H}_\mu^k(z^k - x^*) - g^k$ as follows

$$\begin{aligned}
& \mathbf{H}_\mu^k(z^k - x^*) - g^k \\
= \;& (\mathbf{H}_\mu^k - \nabla^2 f(z^k))(z^k - x^*) + \left[\nabla^2 f(z^k) - \nabla f(z^k) + \nabla f(x^*)\right] + \left[\nabla f(z^k) - g^k\right] \\
= \;& (\mathbf{H}_\mu^k - \nabla^2 f(z^k))(z^k - x^*) + \left[\nabla^2 f(z^k)(z^k - x^*) - \nabla f(z^k) + \nabla f(x^*)\right] \\
& \quad + \left[\nabla f(z^k) - \nabla f(w^k) - \mathbf{H}^k(z^k - w^k)\right] \\
= \;& (\mathbf{H}_\mu^k - \nabla^2 f(z^k))(z^k - x^*) + \left[\nabla^2 f(z^k)(z^k - x^*) - \nabla f(z^k) + \nabla f(x^*)\right] \\
& \quad + \left[\nabla f(z^k) - \nabla f(w^k) - \nabla^2 f(z^k)(z^k - w^k)\right] + (\nabla^2 f(z^k) - \mathbf{H}^k)(z^k - w^k)
\end{aligned} \tag{50}$$

and apply back to the previous inequality

$$
\begin{aligned}
& \left\|x^{k+1}-x^*\right\|^2 \\
\leq\ & \frac{4}{\mu^2}\left(\left\|(\mathbf{H}_\mu^k-\nabla^2 f(z^k))(z^k-x^*)\right\|^2+\left\|\nabla^2 f(z^k)(z^k-x^*)-\nabla f(z^k)+\nabla f(x^*)\right\|^2\right. \\
& \quad \left.+\left\|\nabla f(z^k)-\nabla f(w^k)-\nabla^2 f(z^k)(z^k-w^k)\right\|^4+\left\|(\mathbf{H}^k-\nabla f(w^k))(z^k-w^k)\right\|^4\right) \\
\leq\ & \frac{4}{\mu^2}\left(\left\|\mathbf{H}^k-\nabla^2 f(z^k)\right\|_{\mathrm{F}}^2\left\|z^k-x^*\right\|^2+\left\|\mathbf{H}^k-\nabla^2 f(w^k)\right\|_{\mathrm{F}}\left\|z^k-w^k\right\|^2\right. \\
& \quad \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^2+\frac{L_*^2}{4}\left\|z^k-w^k\right\|^2\right) \\
\leq\ & \frac{4}{\mu^2}\left(2\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2\left\|z^k-x^*\right\|^2+2\left\|\nabla^2 f(z^k)-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2\left\|z^k-x^*\right\|^2\right. \\
& \quad +4\left[\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2+\left\|\nabla^2 f(w^k)-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2\right]\left[\left\|z^k-x^*\right\|^2+\left\|w^k-x^*\right\|^2\right] \\
& \quad \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^4+\frac{L_*^2}{4}\left\|z^k-w^k\right\|^4\right) \\
\leq\ & \frac{4}{\mu^2}\left(2\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2\left\|z^k-x^*\right\|^2+2L_{\mathrm{F}}^2\left\|z^k-x^*\right\|^4\right. \\
& \quad +4\left[\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\mathrm{F}}^2+L_{\mathrm{F}}^2\left\|w^k-x^*\right\|^2\right]\left[\left\|z^k-x^*\right\|^2+\left\|w^k-x^*\right\|^2\right] \\
& \quad \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^4+2L_*^2\left\|z^k-x^*\right\|^4+2H^2\left\|w^k-x^*\right\|^4\right) \\
\leq\ & \frac{4}{\mu^2}\left(2\mathcal{H}^k\gamma_k+2L_{\mathrm{F}}^2\gamma_k^2+4\left[\mathcal{H}^k+L_{\mathrm{F}}^2\nu_k\right](\gamma_k+\nu_k)+\frac{L_*^2}{4}\gamma_k^2+2H^2\gamma_k^2+2L_*^2\nu_k^2\right) \\
=\ & \gamma_k\left(\frac{24}{\mu^2}\mathcal{H}^k+\frac{8L_{\mathrm{F}}^2+{}^9\!/\!_4L_*^2}{\mu^2}\gamma_k+\frac{16L_{\mathrm{F}}^2}{\mu^2}\nu_k\right)+\frac{16}{\mu^2}\mathcal{H}^k\nu_k+\frac{16L_{\mathrm{F}}^2+8L_*^2}{\mu^2}\nu_k^2,
\end{aligned} \tag{51}
$$

where

$$
\begin{aligned}
\mathcal{H}^k &:= \frac{1}{n}\sum_{i=1}^n\left\|\mathbf{H}_i^k-\nabla^2 f_i(x^*)\right\|_{\mathrm{F}}^2, \\
r_k &:= \left\|x^k-x^*\right\|^2, \quad \nu_k := \left\|w^k-x^*\right\|^2, \quad \gamma_k=\left\|z^k-x^*\right\|^2.
\end{aligned}
$$

For *Option 2* we have similar bound with different constants. Recall that $\mu\mathbf{I} \preceq \nabla^2 f(x^k) \preceq \mathbf{H}^k + l^k\mathbf{I}$.

$$
\left\|x^{k+1}-x^*\right\| = \left\|\left[\mathbf{H}^k+l^k\mathbf{I}\right]^{-1}\left(\left[\mathbf{H}^k+l^k\mathbf{I}\right](z^k-x^*)-g^k\right)\right\| \leq \frac{1}{\mu}\left\|\left[\mathbf{H}^k+l^k\mathbf{I}\right](z^k-x^*)-g^k\right\|.
$$

Then we decompose the term $\left[\mathbf{H}^k+l^k\mathbf{I}\right](z^k-x^*)-g^k$ similar to (50):

$$
\begin{aligned}
& \left[\mathbf{H}^k+l^k\mathbf{I}\right](z^k-x^*)-g^k \\
=\ & (\mathbf{H}^k-\nabla^2 f(z^k))(z^k-x^*)+l^k(z^k-x^*)+\left[\nabla^2 f(z^k)(z^k-x^*)-\nabla f(z^k)+\nabla f(x^*)\right] \\
& \quad +\left[\nabla f(z^k)-\nabla f(w^k)-\nabla^2 f(z^k)(z^k-w^k)\right]+(\nabla^2 f(z^k)-\mathbf{H}^k)(z^k-w^k)
\end{aligned}
$$

and apply back to the previous inequality

$$
\begin{aligned}
& \left\|x^{k+1}-x^*\right\|^2 \\
\leq\ & \frac{5}{\mu^2}\left(\left\|(\mathbf{H}^k-\nabla^2 f(z^k))(z^k-x^*)\right\|^2+\|l^k(z^k-x^*)\|^2+\left\|\nabla^2 f(z^k)(z^k-x^*)-\nabla f(z^k)+\nabla f(x^*)\right\|^2\right. \\
& \left.+\left\|\nabla f(z^k)-\nabla f(w^k)-\nabla^2 f(z^k)(z^k-w^k)\right\|^4+\left\|(\mathbf{H}^k-\nabla f(w^k))(z^k-w^k)\right\|^4\right) \\
\leq\ & \frac{5}{\mu^2}\left(\left\|\mathbf{H}^k-\nabla^2 f(z^k)\right\|_{\rm F}^2\left\|z^k-x^*\right\|^2+\left[l^k\right]^2\|(z^k-x^*)\|^2+\left\|\mathbf{H}^k-\nabla^2 f(w^k)\right\|_{\rm F}\left\|z^k-w^k\right\|^2\right. \\
& \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^2+\frac{L_*^2}{4}\left\|z^k-w^k\right\|^2\right) \\
\leq\ & \frac{5}{\mu^2}\left(\frac{2}{n}\sum_{i=1}^n\left\|\mathbf{H}_i^k-\nabla^2 f_i(z^k)\right\|_{\rm F}^2\left\|z^k-x^*\right\|^2+\left\|\mathbf{H}^k-\nabla^2 f(w^k)\right\|_{\rm F}\left\|z^k-w^k\right\|^2\right. \\
& \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^2+\frac{L_*^2}{4}\left\|z^k-w^k\right\|^2\right) \\
\leq\ & \frac{5}{\mu^2}\left(\frac{4}{n}\sum_{i=1}^n\left\|\mathbf{H}_i^k-\nabla^2 f_i(x^*)\right\|_{\rm F}^2\left\|z^k-x^*\right\|^2+\frac{4}{n}\sum_{i=1}^n\left\|\nabla^2 f_i(z^k)-\nabla^2 f_i(x^*)\right\|_{\rm F}^2\left\|z^k-x^*\right\|^2\right. \\
& +4\left[\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\rm F}^2+\left\|\nabla^2 f(w^k)-\nabla^2 f(x^*)\right\|_{\rm F}^2\right]\left[\left\|z^k-x^*\right\|^2+\left\|w^k-x^*\right\|^2\right] \\
& \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^4+\frac{L_*^2}{4}\left\|z^k-w^k\right\|^4\right) \\
\leq\ & \frac{5}{\mu^2}\left(4\mathcal{H}^k\left\|z^k-x^*\right\|^2+4L_{\rm F}^2\left\|z^k-x^*\right\|^4\right. \\
& +4\left[\left\|\mathbf{H}^k-\nabla^2 f(x^*)\right\|_{\rm F}^2+L_{\rm F}^2\left\|w^k-x^*\right\|^2\right]\left[\left\|z^k-x^*\right\|^2+\left\|w^k-x^*\right\|^2\right] \\
& \left.+\frac{L_*^2}{4}\left\|z^k-x^*\right\|^4+2L_*^2\left\|z^k-x^*\right\|^4+2H^2\left\|w^k-x^*\right\|^4\right) \\
\leq\ & \frac{5}{\mu^2}\left(4\mathcal{H}^k\gamma_k+4L_{\rm F}^2\gamma_k^2+4\left[\mathcal{H}^k+L_{\rm F}^2\nu_k\right](\gamma_k+\nu_k)+\frac{9L_*^2}{4}\gamma_k^2+2L_*^2\nu_k^2\right) \\
=\ & \gamma_k\left(\frac{32}{\mu^2}\mathcal{H}^k+\frac{16L_{\rm F}^2+{}^9\!/\!_4L_*^2}{\mu^2}\gamma_k+\frac{16L_{\rm F}^2}{\mu^2}\nu_k\right)+\frac{16}{\mu^2}\mathcal{H}^k\nu_k+\frac{16L_{\rm F}^2+8L_*^2}{\mu^2}\nu_k^2.
\end{aligned} \tag{52}
$$

Combining (51) and (52) with (48), we have

$$
r_{k+1}\leq\gamma_k\left(\frac{C_{\rm M}}{\mu^2}\mathcal{H}^k+\frac{D_{\rm M}}{\mu^2}\gamma_k+\frac{E_1}{\mu^2}\nu_k\right)+\frac{E_2}{\mu^2}\mathcal{H}^k\nu_k+\frac{E_3}{\mu^2}\nu_k^2, \tag{53}
$$

where $E_1:=16L_{\rm F}^2,\ E_2:=16,\ E_3:=16L_{\rm F}^2+8L_*^2$.

Choosing $y=z^k$ and $z=x^*$ in Lemma C.1, we get the following recurrence for $\mathcal{H}^k$:

$$
\mathbb{E}_k\left[\mathcal{H}^{k+1}\right]\leq(1-A)\mathcal{H}^k+BL_{\rm F}^2\gamma_k. \tag{54}
$$

Choosing $u=x^{k+1}$ and $v=x^*$ in Lemma G.3, we get the following recurrence for γ_k:

$$
\begin{aligned}
& \mathbb{E}_k\left[\gamma_{k+1}\right] \\
\leq\ & (1-A_{\rm M})\gamma_k+B_{\rm M}r_{k+1} \\
\overset{(52)}{\leq}\ & (1-A_{\rm M})\gamma_k+\gamma_k\left(\frac{B_{\rm M}C_{\rm M}}{\mu^2}\mathcal{H}^k+\frac{B_{\rm M}D_{\rm M}}{\mu^2}\gamma_k+\frac{B_{\rm M}E_1}{\mu^2}\nu_k\right)+\frac{B_{\rm M}E_2}{\mu^2}\mathcal{H}^k\nu_k+\frac{B_{\rm M}E_3}{\mu^2}\nu_k^2.
\end{aligned} \tag{55}
$$

Assume that $\mathcal{H}^k\leq\frac{A_{\rm M}}{B_{\rm M}}\frac{\mu^2}{\max(9C_{\rm M},12E_2)}=\frac{A_{\rm M}}{B_{\rm M}}\frac{\mu^2}{9C_{\rm M}}$ and $\gamma_k\leq\frac{A_{\rm M}}{B_{\rm M}}\frac{\mu^2}{9\max(D_{\rm M},E_1,E_3)}=\frac{A_{\rm M}}{B_{\rm M}}\frac{\mu^2}{9E_3}$ for all $k\geq 0$. Then from the update rule of w^k we also have $\nu_k\leq\frac{\mu^2A_{\rm M}}{9B_{\rm M}\max(D_{\rm M},E_1,E_3)}$. Using this upper bounds we can simplify the recurrence relation for γ_k to the following

$$\mathbb{E}_k\left[\gamma_{k+1}\right] \le \left(1 - \frac{2A_{\mathrm{M}}}{3}\right)\gamma_k + \frac{A_{\mathrm{M}}}{6}\nu_k. \tag{56}$$

In addition, from the update rule of w^k we imply

$$\mathbb{E}_k\left[\nu_{k+1}\right] = (1-p)\nu_k + p\gamma_k.$$

Finally, for the Lyapunov function

$$\Phi^k = \gamma_k + \frac{A_{\mathrm{M}}}{3p}\nu_k,$$

we have

$$\begin{aligned}
\mathbb{E}_k\left[\Phi^{k+1}\right] &= \mathbb{E}_k\left[\gamma_{k+1}\right] + \frac{A_{\mathrm{M}}}{3p}\mathbb{E}_k\left[\nu_{k+1}\right] \\
&\le \left(1 - \frac{2A_{\mathrm{M}}}{3}\right)\gamma_k + \frac{A_{\mathrm{M}}}{6}\nu_k + \frac{A_{\mathrm{M}}}{3p}\left[(1-p)\nu_k + p\gamma_k\right] \\
&= \left(1 - \frac{A_{\mathrm{M}}}{3}\right)\gamma_k + \left(1 - \frac{p}{2}\right)\frac{A_{\mathrm{M}}}{3p}\nu_k \\
&\le \left(1 - \min\left\{\frac{A_{\mathrm{M}}}{3}, \frac{p}{2}\right\}\right)\Phi^k. \qquad (57)
\end{aligned}$$

G.5. Proof of Lemma G.5

We prove the lemma by induction. Let for some k we have $\mathcal{H}^k \le \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}$ and $\gamma_k \le \min\left\{\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}, \frac{A}{BL_{\mathrm{F}}^2}\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}\right\}$. Then, from the definition of w^k we have $\nu_k \le \min\left\{\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}, \frac{A}{BL_{\mathrm{F}}^2}\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}\right\}$. Since compressors $\mathcal{C}_{\mathrm{M}}^k$ are deterministic (Assumption G.1), from (56) we conclude

$$\gamma_{k+1} \le \left(1 - \frac{2A_{\mathrm{M}}}{3}\right)\gamma_k + \frac{A_{\mathrm{M}}}{6}\nu_k \le \max\{\gamma_k, \nu_k\} \le \min\left\{\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9E_3}, \frac{A}{BL_{\mathrm{F}}^2}\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}\right\}.$$

Since compressors $\mathcal{C}_i^k$ are deterministic (Assumption 3.4), from (54) we conclude

$$\mathcal{H}_{k+1} \le (1-A)\mathcal{H}^k + BL_{\mathrm{F}}^2\gamma_k \le (1-A)\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}} + BL_{\mathrm{F}}^2\frac{A}{BL_{\mathrm{F}}^2}\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}} = \frac{A_{\mathrm{M}}}{B_{\mathrm{M}}}\frac{\mu^2}{9C_{\mathrm{M}}}.$$

G.6. Proof of Lemma G.6

First note that in this case $A_{\mathrm{M}} = B_{\mathrm{M}} = \eta$ so that the ratio $\frac{A_{\mathrm{M}}}{B_{\mathrm{M}}} = 1$. From the Assumption G.2, we have $\mathbf{H}_i^0 = \nabla^2 f_i(z^0)$, from which we get

$$\|\mathbf{H}_i^0 - \nabla^2 f_i(x^*)\|_{\mathrm{M}}^2 \le L_\infty^2\|z^0 - x^*\|^2 \le \frac{\mu^2}{9dC_{\mathrm{M}}},$$

which implies $\mathcal{H}^0 \le \frac{\mu^2}{9dC_{\mathrm{M}}}$. Also notice that $x^0 = z^0$ so that (49) holds for $k=0$. Next we do induction. Let

$$\mathcal{H}^k \le \frac{\mu^2}{9dC_{\mathrm{M}}},\ \|z^k - x^*\| \le \min\left\{\frac{\mu^2}{9dE_3}, \frac{\mu^2}{9C_{\mathrm{M}}d^3L_\infty^2}\right\},\ \|x^k - x^*\| \le \min\left\{\frac{\mu^2}{9d^2E_3}, \frac{\mu^2}{9C_{\mathrm{M}}d^4L_\infty^2}\right\}.$$

hold for all $k \le K$ and prove it for $k = K+1$. Using bounds $\mathcal{H}^k \le \frac{\mu^2}{9dC_{\mathrm{M}}}$ and $\gamma_k \le \frac{\mu^2}{9dE_3}$ we deduce from (53) that

$$\|x^{K+1} - x^*\|^2 \le \frac{1}{3d}\gamma_K + \frac{1}{6d}\nu_K \le \frac{1}{d}\max\{\gamma_K, \nu_K\} \le \min\left\{\frac{\mu^2}{9d^2E_3}, \frac{\mu^2}{9C_{\mathrm{M}}d^4L_\infty^2}\right\}.$$

Since $(z^{K+1})_j$ is a convex combination of $\{(x^t)_j\}_{t=0}^{K+1}$, we get

$$\begin{aligned}
\|z^{K+1} - x^*\|^2 &= \sum_{j=1}^{d} |(z^{K+1} - x^*)_j|^2 \\
&\leq \sum_{j=1}^{d} \max_{0 \leq t \leq K+1} |(x^t - x^*)_j|^2 \\
&\leq d \max_{0 \leq t \leq K+1} \|x^t - x^*\|^2 \leq \min\left\{ \frac{\mu^2}{9dE_3}, \frac{\mu^2}{9C_{\mathrm{M}} d^3 L_\infty^2} \right\}.
\end{aligned}$$

Since $(\mathbf{H}_i^{K+1})_{jl}$ is a convex combination of $\{(\nabla^2 f_i(z^t))_{jl}\}_{t=0}^{K+1}$, we get

$$\begin{aligned}
\|\mathbf{H}_i^{K+1} - \nabla^2 f_i(x^*)\|^2 &= \sum_{j,l=1}^{d} |(\mathbf{H}_i^{K+1} - \nabla^2 f_i(x^*))_{jl}|^2 \\
&\leq d^2 L_\infty^2 \max_{0 \leq t \leq K+1} \|z^t - x^*\|^2 \leq \frac{\mu^2}{9dC_{\mathrm{M}}}.
\end{aligned}$$

The last three inequalities complete the induction step and we conclude the lemma.

H. Local Quadratic Rate of NEWTON-STAR for General Finite-Sum Problems

In their recent work, Islamov et al. (2021) proposed a novel Newton-type method, which does not update the Hessian estimator from iteration to iteration and, meanwhile, preserves fast local quadratic rate of convergence. The method can be describe with a single update rule preformed by the master:

$$x^{k+1} = x^k - \left[\nabla^2 f(x^*)\right]^{-1} \nabla f(x^k), \quad k \geq 0. \tag{58}$$

Note that parallel nodes need to send the master only gradient information $\nabla f_i(x^k)$. Then master aggregates them, performs the update step (58) and sends new parameters x^{k+1} to devices for the next round. While this scheme is very simple-looking, notice that the update rule (58) depends on the knowledge of $\nabla^2 f(x^*)$, where x^* is the (unique) solution of (1). As we do not know x^* (otherwise there is no sense to do any training), this method, called NEWTON-STAR, is practically useless and cannot be implemented. However, this method was quite useful in theory, since it led to a new practical method.

Now, the local quadratic rate of NEWTON-STAR was shown using some special structure of local loss functions $f_i(x)$. Here we provide a very simple proof of local quadratic rate which works for any smooth losses and does not need special structure of $f_i(x)$.

Theorem H.1. *Assume that $f\colon \mathbb{R}^d \to \mathbb{R}$ has L_*-Lipschitz Hessian and the Hessian at the optimum x^* is positive definite with parameter $\mu > 0$. Then local convergence rate of NEWTON-STAR (58) is quadratic, i.e., for any $k \geq 0$ and initial point $x^0 \in \mathbb{R}^d$ we have*

$$\|x^{k+1} - x^*\| \leq \frac{L_*}{2\mu} \|x^k - x^*\|^2.$$

Proof. As we do not have a regularization term in our ERM problem, we imply $\nabla f(x^*) = 0$. Hence

$$\begin{aligned}
\|x^{k+1} - x^*\| &= \left\| x^k - x^* - \left[\nabla^2 f(x^*)\right]^{-1} \nabla f(x^k) \right\| \\
&\leq \left\| \left[\nabla^2 f(x^*)\right]^{-1} \right\| \left\| \nabla^2 f(x^*)(x^k - x^*) - \nabla f(x^k) + \nabla f(x^*) \right\| \\
&\leq \frac{L_*}{2\mu} \|x^k - x^*\|^2,
\end{aligned}$$

where we used positive definiteness $\nabla^2 f(x^*) \succeq \mu \mathbf{I}$ and L_*-Lipschitzness of the Hessian $\nabla^2 f(x)$, namely

$$\left\| \nabla^2 f(y)(x - y) - \nabla f(x) + \nabla f(y) \right\| \leq \frac{L_*}{2} \|x - y\|^2, \quad x, y \in \mathbb{R}^d.$$

□

I. FedNL and the Newton "Triangle"

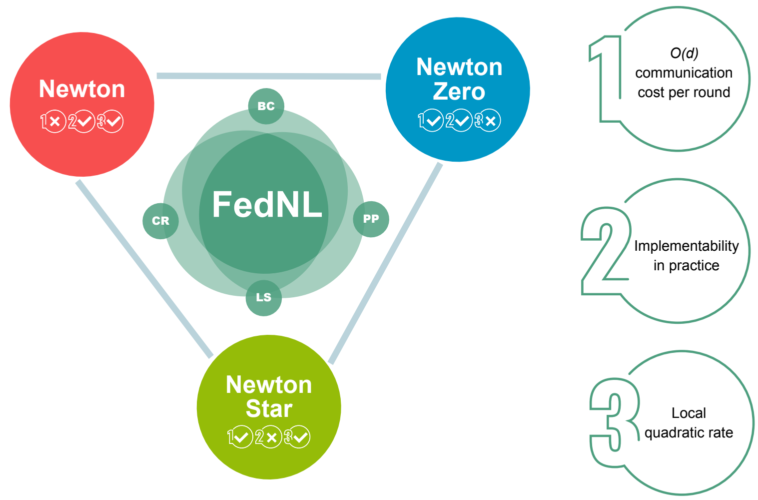


Figure 14: Visualization of the three special Newton-type methods—Newton (N), Newton Star (NS) and Newton Zero (N0)—as the vertices of a triangle capturing a subset of two of these three requirements: 1) $\mathcal{O}(d)$ communication cost per round, 2) implementability in practice, and 3) local quadratic rate. Indeed, each of these three methods satisfies *two* of these requirements only: N (2+3), NS (1+3) and N0 (1+2). Finally, the proposed FedNL framework with its four extensions to Partial Participation (FedNL-PP), globalization via Line Search (FedNL-LS), globalization via Cubic Regularization (FedNL-CR) and Bidirectional Compression (FedNL-BC) interpolates between these requirements.

J. Limitations

Here we discuss main limitations of our approach and directions which are not explored in this work.

- Our theory covers general convex (the rate (36)) and strongly convex (all other rates of this paper) loss functions. We do not consider non-convex objectives in this work.
- All the proposed methods are analyzed in the regime when the exact local gradients and exact local Hessians of local loss functions are computed for all participating devices. We do not consider stochastic gradient or stochastic Hessian oracles of local loss functions in our analyses.
- We present separate methods/extensions (FedNL, FedNL-PP, FedNL-CR, etc) for each setup (compressed communication, partial participation, globalization, etc) to make our contributions clearer. For practical purposes, however, one might need to combine these extensions in order to get a method which supports compressed communication, partial participation, globalization, etc at the same time. We do not design a single master method containing all these extensions as special cases.
- Finally, we do not provide strong (differential) privacy guarantees for our methods. Our privacy enhancement mechanism offers the most rudimentary level of privacy only: we forbid the devices do directly send/reveal their training data to the server.

K. Table of Frequently Used Notation

Table 7: Notation we use throughout the paper.

Basic		
d	number of the model parameters to be trained	
n	number of the devices/workers/clients in distributed system	
$[n]$	$:= \{1, 2, \ldots, n\}$	
f_i	local loss function associated with data stored on device $i \in [n]$	(1)
f	$:= \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, overall empirical loss/risk	(1)
x^*	trained model, i.e., the optimal solution to (1)	
ε	target accuracy	
$\mathbb{R}^{d\times d}$	the set of $d \times d$ square matrices	
$(\mathbf{M})_{jl}$	the element at j^{th} row and l^{th} column of matrix $\mathbf{M}$	
Standard		
μ	strong convexity parameter of f	Asm 3.1
L	Lipschitz constant of the gradient $\nabla f(x)$ w.r.t. the Euclidean norm	Thm E.1
L_*	Lipschitz constant of the Hessian $\nabla^2 f(x)$ w.r.t. the spectral norm	Asm 3.1
L_{F}	Lipschitz constant of the Hessian $\nabla^2 f(x)$ w.r.t. the Frobenius norm	Asm 3.1
L_∞	Lipschitz constant of the Hessian $\nabla^2 f(x)$ w.r.t. the $\max$ norm	Asm 3.1
$\mathcal{C}$	(possibly randomized) compression operator $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$	(3), (4)
$\mathbb{B}(\omega)$	class of unbiased compressors with bounded variance $\omega \geq 0$	Def 3.2
$\mathbb{C}(\delta)$	class of deterministic contractive compressors with contraction $\delta \in [0, 1]$	Def 3.3
Algorithm names		
GD	Gradient Descent	
GD-LS	GD with Line Search procedure	
DIANA	Compressed GD with variance reduction (Mishchenko et al., 2019)	
ADIANA	DIANA with Nesterov's acceleration (Li et al., 2020b)	
N	classical Newton	
NS	Newton Star	(58)
N0	Newton Zero **(new)**	(10)
N0-LS	Newton Zero with Line Search procedure **(new)**	
NL1, NL2	Newton Learn methods of Islamov et al. (2021)	
CNL	Cubic Newton Learn (Islamov et al., 2021)	
DINGO	Distributed Newton-type method of (Crane & Roosta, 2019)	
FedNL	Federated Newton Learn **(new)**	Alg 1
FedNL-PP	Extension to FedNL: Partial Participation **(new)**	Alg 2
FedNL-LS	Extension to FedNL: Globalization via Line Search **(new)**	Alg 3
FedNL-CR	Extension to FedNL: Globalization via Cubic Regularization **(new)**	Alg 4
FedNL-BC	Extension to FedNL: Bidirectional Compression **(new)**	Alg 5
Federated Newton Learn (FedNL)		
$\mathbf{H}_i^k$	estimate of the local optimal Hessian $\nabla^2 f_i(x^*)$ at client i in iteration k	
$\mathbf{H}^k$	estimate of the global optimal Hessian $\nabla^2 f(x^*)$ at the server in iteration k	
α	Hessian learning rate	
$\mathcal{C}_i^k$	compression operator applied by the client i in iteration k	
$\mathbf{S}_i^k$	$:= \mathcal{C}_i^k(\nabla^2 f_i(x^k) - \mathbf{H}_i^k)$ compressed second order information	
l_i^k	$:= \|\nabla^2 f_i(x^k) - \mathbf{H}_i^k\|_{\mathrm{F}}$ compression error	
$A,\ B$	constants depending on the choice of compressors $\mathcal{C}_i^k$ and learning rate α	(5)
$C,\ D$	constants depending on which option is chosen for the global update	(5)
Experiments		
$\{a_{ij}, b_{ij}\}$	j^{th} data point stored in device i	(12)
m	number of local training data points	(12)
λ	regularization parameter	(12)