FedNL: Making Newton-Type Methods Applicable to Federated Learning

Mher Safaryan

Rustem Islamov

Xun Qian

Peter Richtárik

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

MIPT
MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

Mher Safaryan

Postdoctoral fellow

Xun Qian

Postdoctoral fellow

Peter Richtárik

Professor of Computer Science

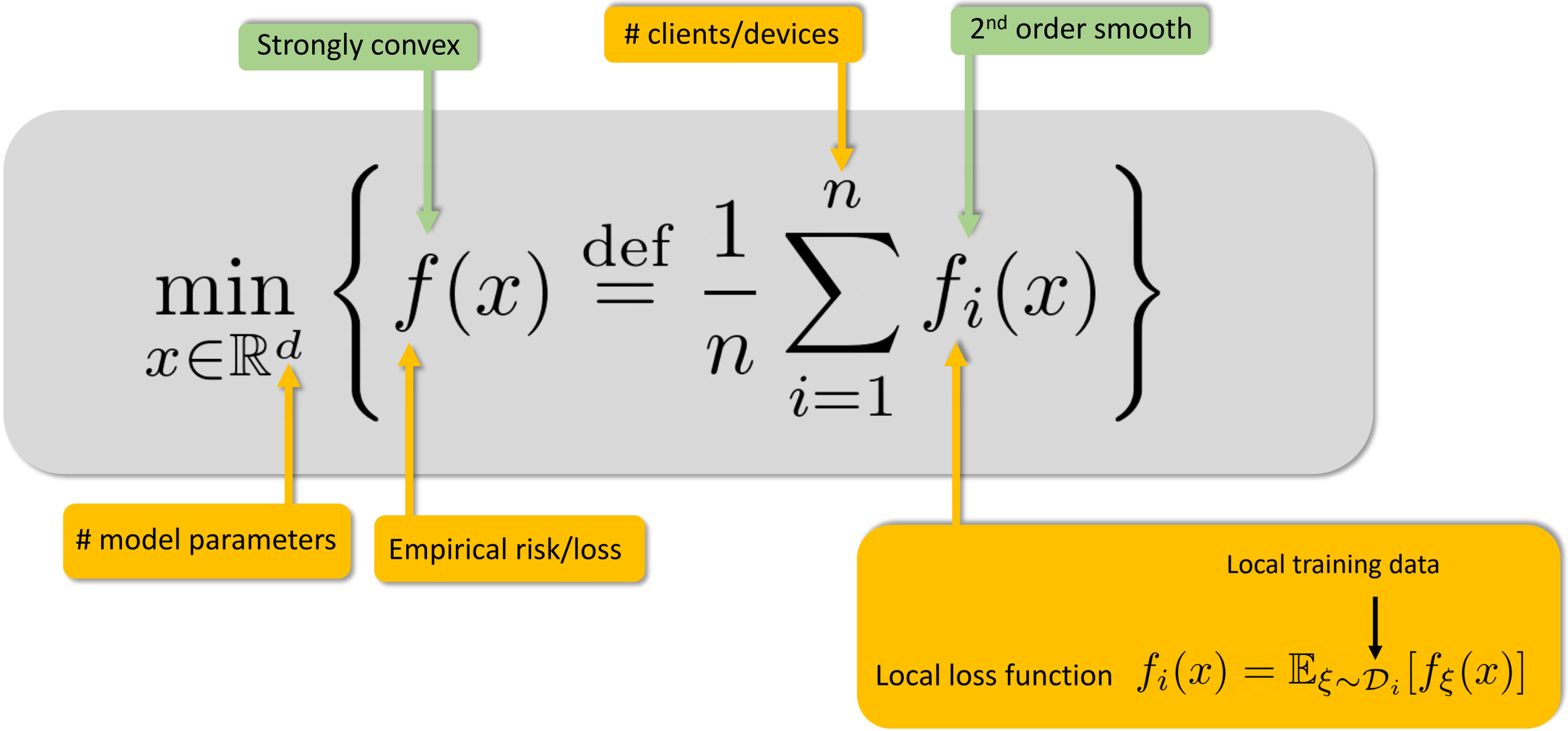
Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

Outline

1. **The Problem**
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

The Problem


Strongly convex
clients/devices
2nd order smooth
$$\min_{x \in \mathbb{R}^d} \left\{ f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$
model parameters
Empirical risk/loss
Local training data
Local loss function $f_i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_\xi(x)]$

Outline

1. **The Problem**
2. **Brief Comparison with Related Works**
3. **The 3 Special Newton-type Methods**
4. **Federated Newton Learn (FedNL)**
5. **Extensions (PP, LS, CR, BC)**
6. **Numerical Experiments**

Brief Comparison with Related Works

Method	Problem	Assumptions	CC[1]	Rate	Comments
GIANT [Wang et al., 2018] *NeurIPS*	GLM[3]	LipC[2] 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 and Roosta, 2019] *NeurIPS*	GFS[4]	Moral Smoothness[5], $\approx$ strong convexity[6]	$\mathcal{O}(d)$	Global linear rate. No fast local rate.	Operates full gradients, Hessian-vector products, Hessian pseudo-inverse and vector products.
DAN [Zhang et al., 2020] *IEEE, Decision and Control*	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] *IEEE, Decision and Control*	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 ?
NEWTON-LEARN [Islamov et al., 2021] *ICML*	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] *ICML*	GFS	LipC Hessian, LipC gradient, strong convexity[6]	$\widetilde{\mathcal{O}}(d^2)$	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] **CC** = **C**ommunication **C**ost per iteration. [2] **LipC** = **Lip**schitz **C**ontinuous.
[3] GLM = Generalized Linear Model, e.g. $loss_j(x; a_j) = \phi_j(a_j^\top x) + \lambda\|x\|^2$. [4] GFS = General Finite Sum.
[5] Moral Smoothness: $\|\nabla^2 f(x)\nabla f(x) - \nabla^2 f(y)\nabla f(y)\| \leq L\|x - y\|$. [6] Applies to local loss functions for all clients/devices.
† Partial Participation, Globalization (via Line Search and Cubic Regularization) and Bidirectional Compression.

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

Newton Method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^k)\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k)\right)$$

Can be computed locally

Expensive to communicate: $\mathcal{O}(d^2)$

Can be computed locally

Easy to communicate: $\mathcal{O}(d)$

- $\mathcal{O}(d)$ communication cost per round
- Implementability in practice
- Local quadratic convergence rate independent of the condition number

Hessian Lipschitz constant

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

Strong convexity constant

Local quadratic rate

Newton Star Method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*)\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k)\right)$$

Can NOT be computed locally

Single communication of $\mathcal{O}(d^2)$

Can be computed locally

Easy to communicate: $\mathcal{O}(d)$

Rustem Islamov, Xun Qian and Peter Richtárik
Distributed second order methods with fast rates and compressed communication,
ICML 2021.

- $\mathcal{O}(d)$ communication cost per round
- Implementability in practice
- Local quadratic convergence rate independent of the condition number

Hessian Lipschitz constant

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

Strong convexity constant

Local quadratic rate

Newton Zero Method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^0)\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k)\right)$$

Can be computed locally

Single communication of $\mathcal{O}(d^2)$

Can be computed locally

Easy to communicate: $\mathcal{O}(d)$

- $\mathcal{O}(d)$ communication cost per round*
- Implementability in practice
- Local quadratic convergence rate

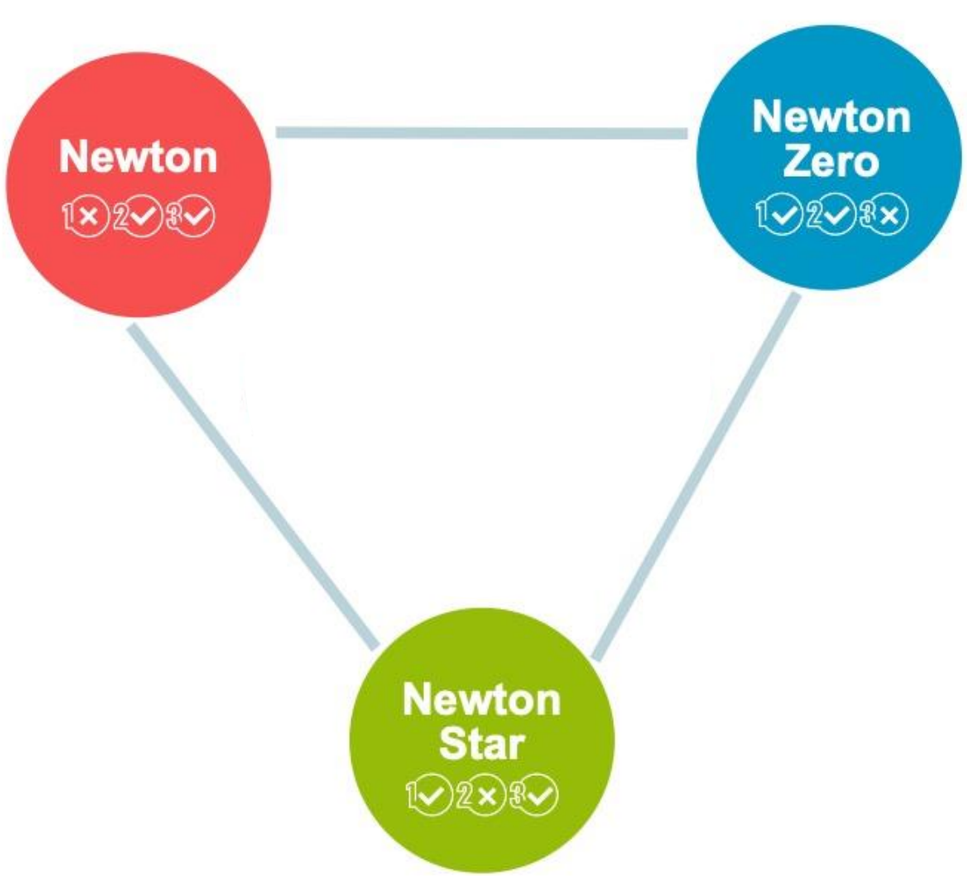
$$\|x^{k+1} - x^*\| \leq \frac{1}{2}\|x^k - x^*\|$$

Local (fixed) linear rate: $\|x^0 - x^*\| \leq \frac{\mu}{2L}$

Strong convexity constant

Hessian Lipschitz constant

“Newton Triangle”



1. *O(d)* communication cost per round
2. Implementability in practice
3. Local quadratic rate

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

Learning the Optimal Hessian Matrices

Newton Star

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*)\right)^{-1} \nabla f(x^k)$$

Idea! Learn the optimal Hessians $\nabla^2 f_i(x^*)$ in communication efficient manner:

$(i)\ \mathbf{H}_i^k \to \nabla^2 f_i(x^*) \text{ as } k \to \infty \qquad (ii)\ \mathbf{H}_i^{k+1} - \mathbf{H}_i^k \text{ is compressed}$

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{H}_i^k\right)^{-1} \nabla f(x^k)$$

$$= x^k - \left(\mathbf{H}^k\right)^{-1} \nabla f(x^k)$$

Rustem Islamov, Xun Qian and Peter Richtárik
Distributed second order methods with fast rates and compressed communication,
ICML 2021.

FedNL: Two Options for Updating the Global Model

Option 1

$$x^{k+1} = x^k - \left(\left[\mathbf{H}^k \right]_{\mu} \right)^{-1} \nabla f(x^k)$$

Projection onto the cone of positive definite matrices

Option 2

$$x^{k+1} = x^k - \left(\mathbf{H}^k + l^k \mathbf{I} \right)^{-1} \nabla f(x^k)$$

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

FedNL: New Hessian Learning Technique

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

Compression operator

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč and Peter Richtárik, **Distributed learning with compressed gradient differences,** arXiv:1901.09269, 2019.

Contractive compressor $\mathbb{C}(\delta),\ \delta \in [0, 1)$

$$\|\mathcal{C}(\mathbf{M})\|_{\mathrm{F}} \leq \|\mathbf{M}\|_{\mathrm{F}}$$

$$\|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_{\mathrm{F}}^2 \leq (1-\delta)\|\mathbf{M}\|_{\mathrm{F}}^2 \quad \forall\, \mathbf{M} \in \mathbb{R}^{d\times d}$$

Unbiased compressor $\mathbb{B}(\omega),\ \omega \geq 0$

$$\mathbb{E}[\mathcal{C}(\mathbf{M})] = \mathbf{M}$$

$$\mathbb{E}\left[\|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_{\mathrm{F}}^2\right] \leq \omega\|\mathbf{M}\|_{\mathrm{F}}^2 \quad \forall\, \mathbf{M} \in \mathbb{R}^{d\times d}$$

Does not rely on Error Feedback mechanism

Current theory supports only random sparsification

FedNL: Hessian Learning Rate Options

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

Stepsize depends only on the compression

Assumption 3.4. $\mathcal{C}_i^k \in \mathbb{C}(\delta)$ *for all* $i \in [n]$ *and* k. *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 *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* $k \geq 0$.

Algorithm 1 FedNL (Federated Newton Learn)

Parameters: Hessian learning rate $\alpha \geq 0$; compression operators $\{\mathcal{C}_1^k, \ldots, \mathcal{C}_n^k\}$
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 each node $i \in [n]$
  $\nabla f(x^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^k)$, $\mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k$, $l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$, $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$
  Option 1: $x^{k+1} = x^k - \left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$
  Option 2: $x^{k+1} = x^k - \left[\mathbf{H}^k + l^k \mathbf{I}\right]^{-1} \nabla f(x^k)$

FedNL: Assumptions

FedNL: Local Convergence Theory

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

Local (fixed) linear rate

Lyapunov function

$$\Phi^k := 6BH^2\|x^k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2$$

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

Local linear rates

$$x^k \to x^*$$
$$\mathbf{H}_i^k \to \nabla^2 f_i(x^*) \text{ for all } i \in [n]$$

Constant depending on the choice of the compressor and stepsize

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

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

Local superlinear rate

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

Extension: Partial Participation

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

Parameters: Hessian learning rate $\alpha > 0$; compression operators $\{\mathcal{C}_1^k, \ldots, \mathcal{C}_n^k\}$; number of participating devices $\tau \in \{1, 2, \ldots, n\}$

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 $S^k \subseteq \{1, \ldots, n\}$ 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 $i = 1, \ldots, n$ 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 $l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$

Extension: Partial Participation

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

$$\Psi^k := \mathcal{H}^k + BL_{\mathrm{F}}^2 \mathcal{W}^k.$$

Theorem C.1. *Let Assumption 3.1 holds and further assume that all loss functions f_i are μ-convex. Suppose $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{4(L_* + 2L_{\mathrm{F}})^2}$ 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

$$\mathbb{E}\left[\Psi^k\right] \leq \left(1 - \frac{\tau}{n}\min\left\{A, \frac{1}{2}\right\}\right)^k \Psi^0, \tag{25}$$

$$\mathbb{E}\left[\frac{\|x^{k+1} - x^*\|^2}{\mathcal{W}^k}\right] \leq \left(1 - \min\left\{A, \frac{1}{2}\right\}\right)^k \left(\frac{(L_* + 2L_{\mathrm{F}})^2}{2BL_{\mathrm{F}}^2} + 8\right)\frac{\Psi^0}{\mu^2}. \tag{26}$$

Extension: Globalization via Line Search

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

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 $\gamma \in (0, 1)$

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; compute $f_i(x^k)$, $\nabla f_i(x^k)$ and $\nabla^2 f_i(x^k)$

  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

  Update local Hessian shifts $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$

end for

on server

  Get $f_i(x^k)$, $\nabla f_i(x^k)$ and $\mathbf{S}_i^k$ from all devices $i \in [n]$

  $f(x^k) = \frac{1}{n} \sum_{i=1}^n f_i(x^k)$, $\nabla f(x^k) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k)$, $\mathbf{S}^k = \frac{1}{n} \sum_{i=1}^n \mathbf{S}_i^k$

  Compute search direction $d^k = -\left[\mathbf{H}^k\right]_\mu^{-1} \nabla f(x^k)$

  Find the smallest integer $s \geq 0$ satisfying $f(x^k + \gamma^s d^k) \leq f(x^k) + c\gamma^s \left\langle \nabla f(x^k), d^k \right\rangle$

  Update global model to $x^{k+1} = x^k + \gamma^s d^k$

  Update global Hessian shift to $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$

Extension: Globalization via Line Search

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_g-smoothness we assume Lipschitz continuity of gradients with Lipschitz constant L_g.

Theorem D.1. *Let Assumption 3.1 hold, function f be L_g-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*

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L_g} \min\left\{\frac{\mu}{\widetilde{L}}, 1\right\}\right)^k \left(f(x^0) - f(x^*)\right) \tag{31}$$

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

Lemma D.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 $\widetilde{L} \leq dL_\infty R + \|\nabla^2 f(x^*)\|$.*

Extension: Globalization via 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 $H \geq 0$ 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 $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$

Extension: Globalization via Cubic Regularization

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

Theorem E.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*

$$f(x^k) - f(x^*) \leq \frac{9lR^2}{k} + \frac{9L_*R^3}{k^2} + \frac{3\left(f(x^0) - f(x^*)\right)}{k^3}, \tag{33}$$

where $R := \{\|x - x^*\| : f(x) \leq f(x^0)\}$*. 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*

$$\mathcal{O}\left(\left(\frac{l}{\mu} + \sqrt{\frac{L_*R}{\mu}} + 1\right)\log\frac{f(x^0) - f(x^*)}{\varepsilon}\right) \tag{34}$$

iterations. Furthermore, if $\|x^0 - x^*\|^2 \leq \frac{\mu^2}{20(L_*^2 + 8L_{\mathrm{F}}^2)}$ *and* $\mathcal{H}^k \leq \frac{\mu^2}{160}$ *for all* $k \geq 0$*, then we have the same local rates (6), (7) and (8).*

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

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

Extension: Bidirectional Compression

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

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 $\mathcal{C}_{\mathrm{M}}^k$
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 $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$; $\xi^0 = 1$
for each device $i = 1, \ldots, n$ in parallel **do**
  Get ξ^k from the server
  if $\xi^k = 1$
    Compute local gradient $\nabla f_i(z^k)$ and send to the server
    $g_i^k = \nabla f_i(z^k), \quad w^{k+1} = z^k$
  if $\xi^k = 0$
    $g_i^k = \mathbf{H}_i^k(z^k - w^k) + \nabla f_i(w^k), \quad w^{k+1} = w^k$
  Compute local Hessian $\nabla^2 f_i(z^k)$
  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
  Update local Hessian shift to $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \mathbf{S}_i^k$
end for
on server
  $g^k = \frac{1}{n}\sum_{i=1}^n g_i^k, \quad \mathbf{S}^k = \frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^k, \quad l^k = \frac{1}{n}\sum_{i=1}^n l_i^k$
  Option 1: $x^{k+1} = z^k - \left[\mathbf{H}^k\right]_\mu^{-1} g^k$
  Option 2: $x^{k+1} = z^k - \left[\mathbf{H}^k + l^k \mathbf{I}\right]^{-1} g^k$
  Update global Hessian shifts $\mathbf{H}^{k+1} = \mathbf{H}^k + \alpha \mathbf{S}^k$
  Send $s^k := \mathcal{C}_{\mathrm{M}}^k(x^{k+1} - z^k)$ to all devices $i \in [n]$
  Update the model $z^{k+1} = z^k + \eta s^k$
  Sample $\xi^{k+1} \sim$ Bernoulli(p) and send to all devices $i \in [n]$
for each device $i = 1, \ldots, n$ in parallel **do**
  Get s^k from the server and update the model $z^{k+1} = z^k + \eta s^k$
end for

Extension: Bidirectional Compression

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 F.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*:*

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

Summary of Theoretical Results

Method	Convergence: result †	type	rate	Rate independent of the condition # (left) # training data (middle) compressor (right)	Theorem
Newton Zero N0 (Equation (9))	$r_k \leq \frac{1}{2^k} r_0$	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	✓ ✓ ✓	C.1
	$\Phi_2^k \leq \theta^k \Phi_2^0$	local	linear	✓ ✓ ✗	C.1
	$r_{k+1} \leq c\theta^k \mathcal{W}_k$	local	linear	✓ ✓ ✗	C.1
Line Search FedNL-LS (Algorithm 3)	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	D.1
Cubic Regularization FedNL-CR (Algorithm 4)	$\Delta_k \leq c/k$	global	sublinear	✗ ✓ ✓	E.1
	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	E.1
	$\Phi_1^k \leq \theta^k \Phi_1^0$	local	linear	✓ ✓ ✗	E.1
	$r_{k+1} \leq c\theta^k r_k$	local	superlinear	✓ ✓ ✗	E.1
Bidirectional Compression FedNL-BC (Algorithm 5)	$\Phi_3^k \leq \theta^k \Phi_3^0$	local	linear	✓ ✓ ✗	F.4
Newton Star NS (Equation (55))	$r_{k+1} \leq c r_k^2$	local	quadratic	✓ ✓ ✓	G.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 $\Phi_1^k := c\|x^k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2$; $\Phi_2^k := c\mathcal{W}^k + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2$; $\Phi_3^k := \|z^k - x^*\|^2 + c\|w^k - x^*\|^2$. (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 in this table. Refer to the precise statements of the theorems for the exact values.

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Federated Newton Learn (FedNL)
5. Extensions (PP, LS, CR, BC)
6. Numerical Experiments

Experiments: Regularized Logistic Regression

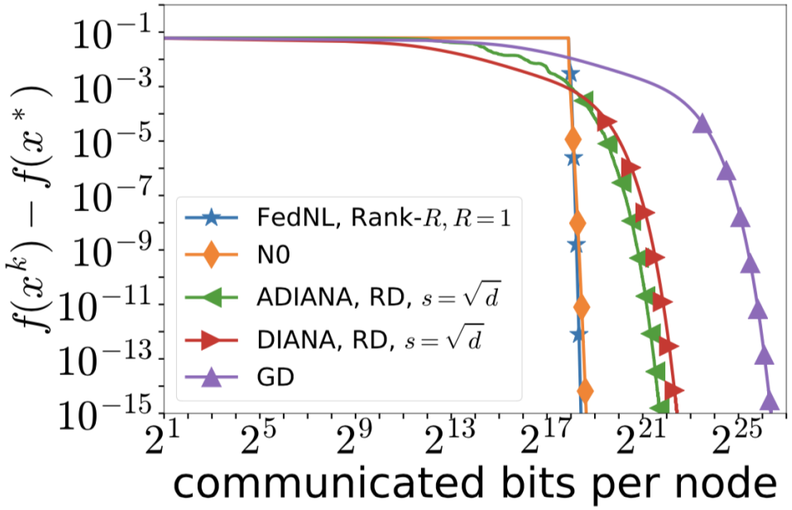
Regularization parameter

Training data points

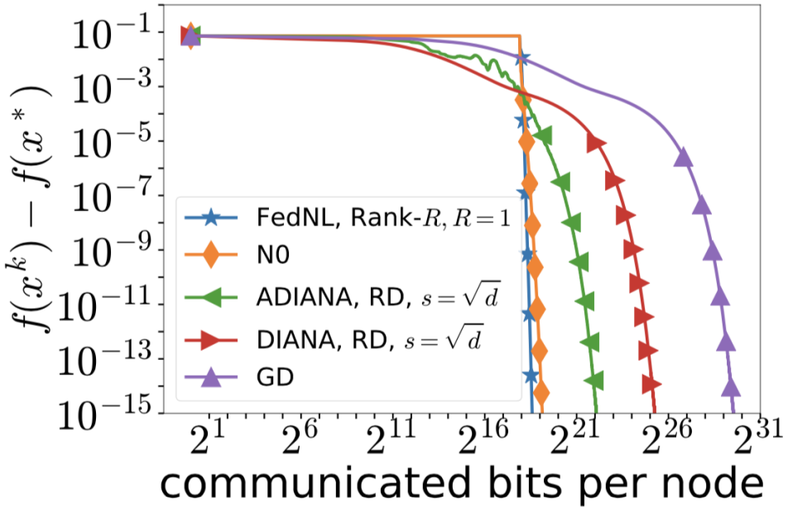
$$\min_{x\in\mathbb{R}^d}\left\{f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\frac{\lambda}{2}\|x\|^2\right\},\qquad f_i(x)=\frac{1}{m}\sum_{j=1}^{m}\log\left(1+\exp(-b_{ij}a_{ij}^{\top}x)\right),$$

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

Experiments: Local Comparisong against FOM

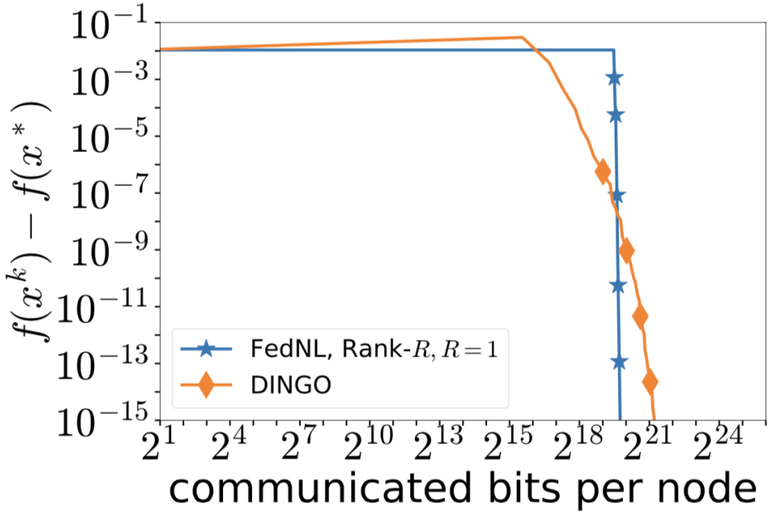


(a) a1a, $\lambda = 10^{-3}$

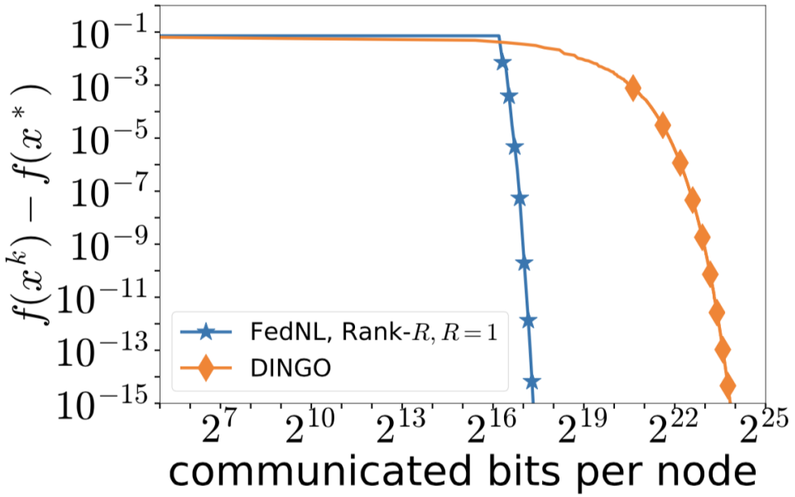


(b) a9a, $\lambda = 10^{-4}$

Experiments: Local Comparisong against DINGO

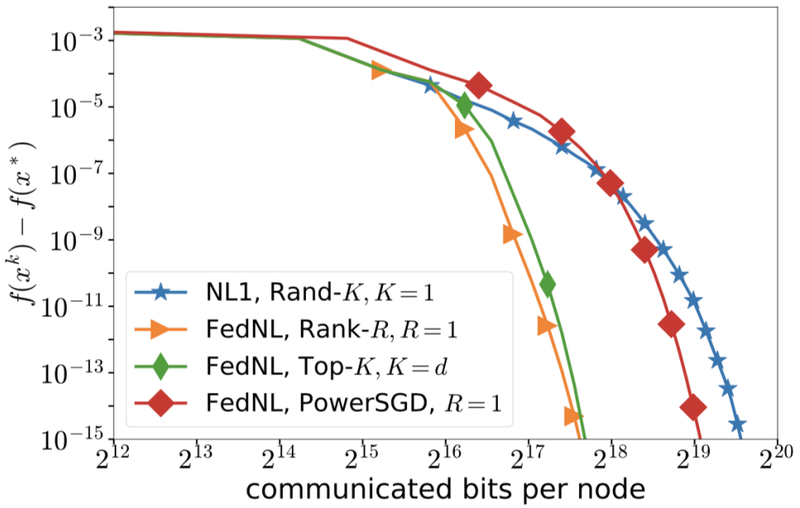


(c) w8a, $\lambda = 10^{-3}$

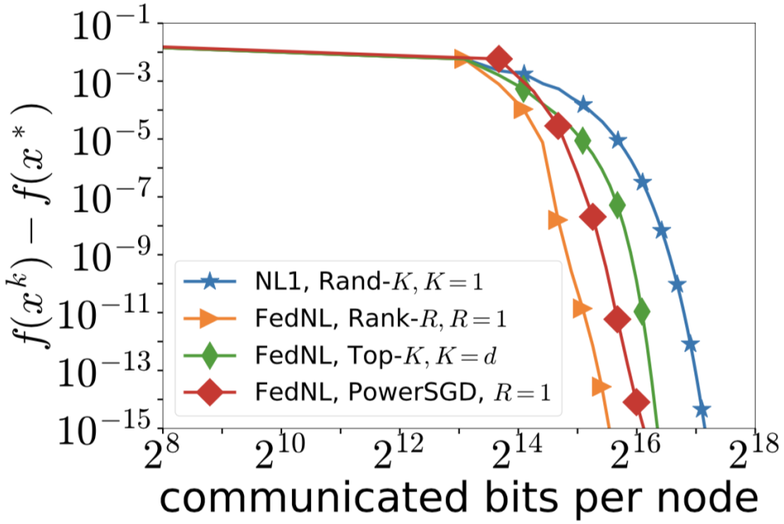


(d) phishing, $\lambda = 10^{-4}$

Experiments: Local Comparisong against NEWTON-LEARN(NL)

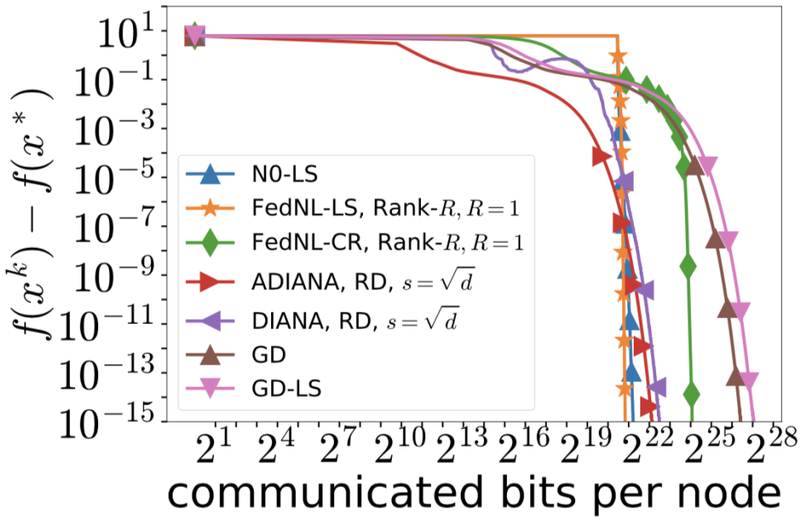


(a) w8a, $\lambda = 10^{-3}$

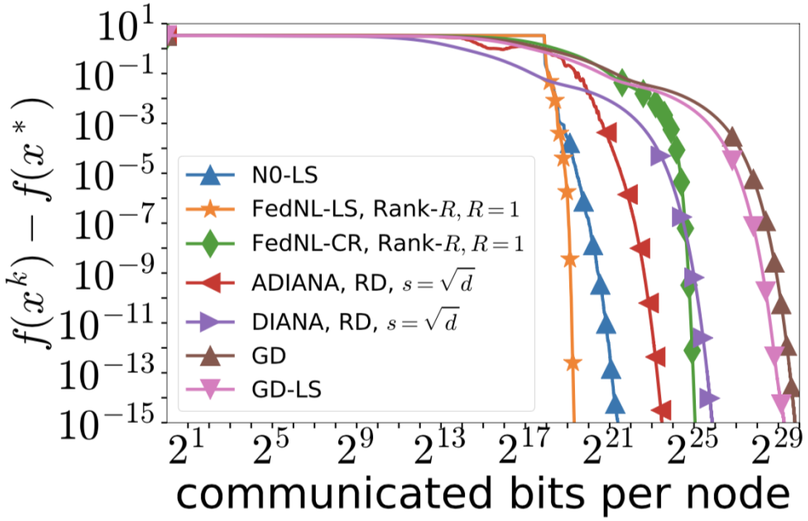


(b) phishing, $\lambda = 10^{-3}$

Experiments: Global Comparison against FOM

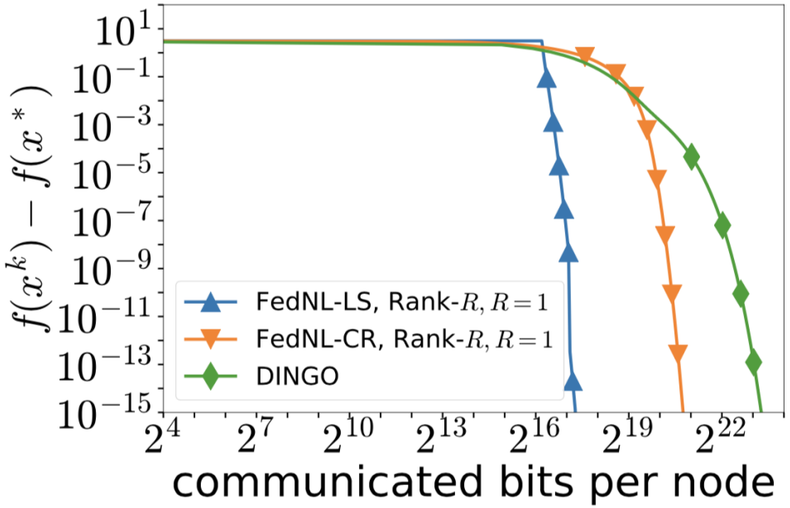


(a) w7a, $\lambda = 10^{-3}$

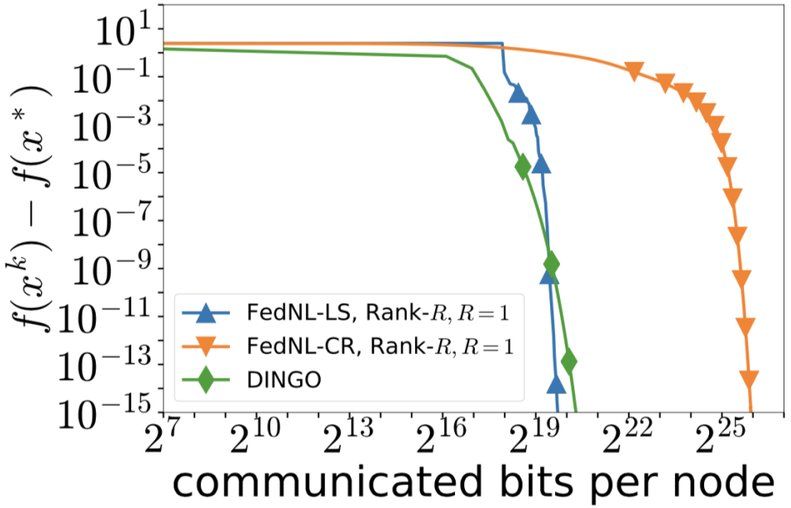


(b) a1a, $\lambda = 10^{-4}$

Experiments: Global Comparison against DINGO



(c) phishing, $\lambda = 10^{-3}$



(d) a9a, $\lambda = 10^{-4}$

Thank you