FedNL:
Making Newton-Type Methods Applicable to Federated Learning

Mher Safaryan

Postdoctoral Research Fellow

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

ICML
International Conference
On Machine Learning

Rustem Islamov

Master's student

MIPT
MOSCOW INSTITUTE
OF PHYSICS AND TECHNOLOGY

INSTITUT POLYTECHNIQUE DE PARIS

Xun Qian

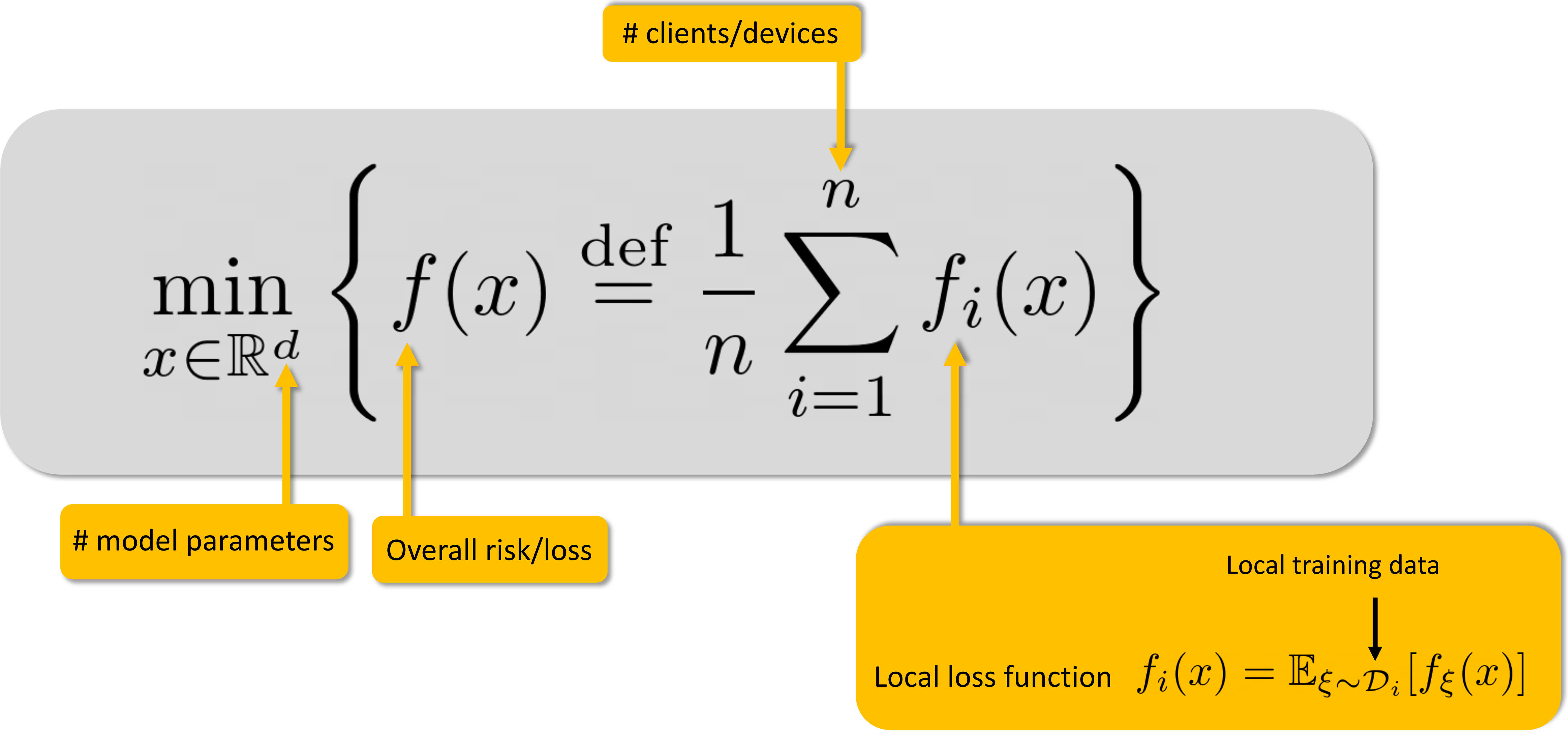
Research Scientist

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Peter Richtárik

Professor of Computer Science

The Problem



clients/devices

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

model parameters

Overall risk/loss

Local training data

Local loss function $f_i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[f_\xi(x)]$

The Problem

Hessians are Lipschitz continuous

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq L\|x-y\|$$

2nd order smooth

non-convex

$$\min_{x\in\mathbb{R}^d}\left\{ f(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\}$$

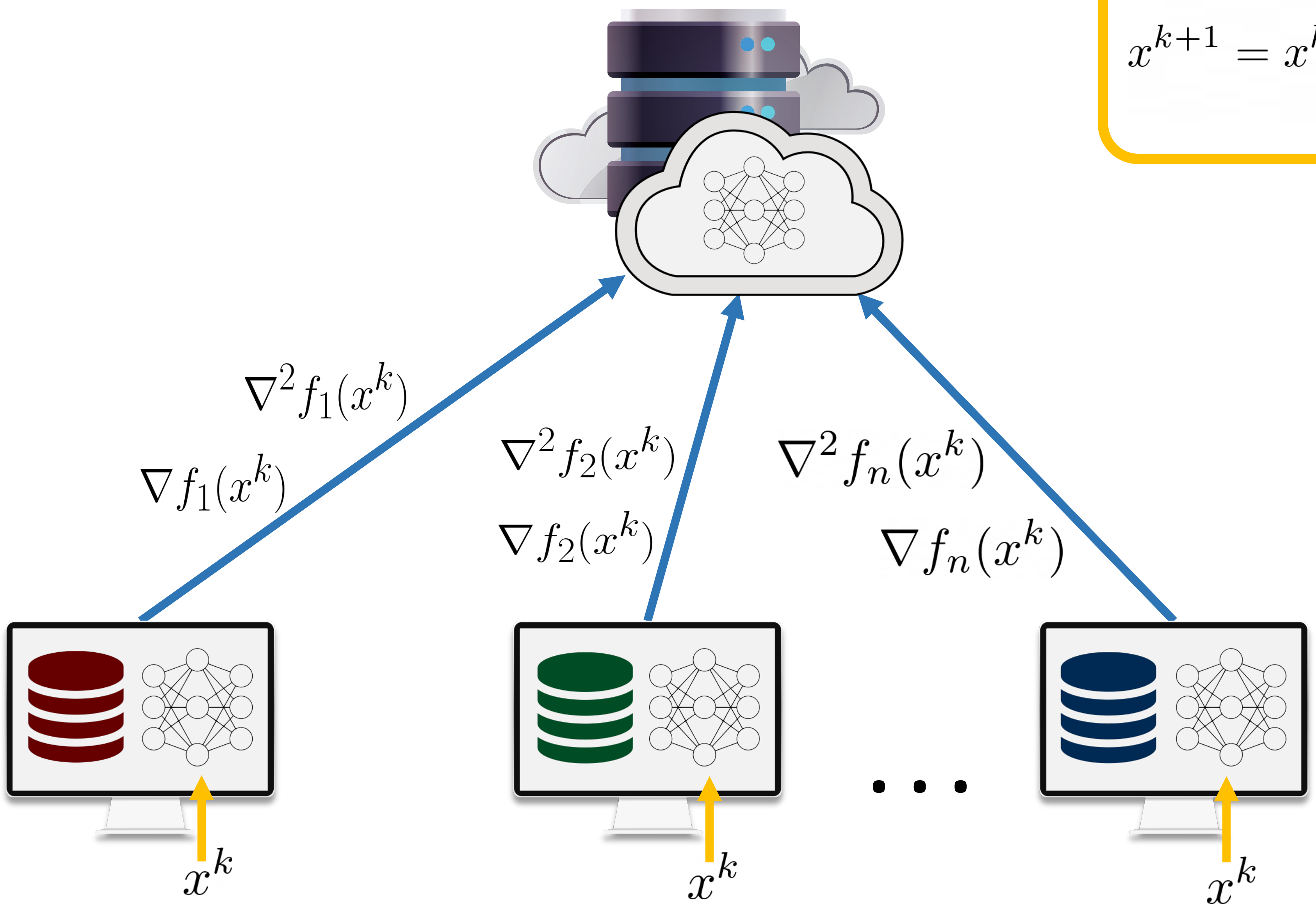
Strongly convex

for some $\mu > 0$ and for any $x, y \in \mathbb{R}^d$

$$f(x) \geq f(y) + \langle \nabla f(x), x-y\rangle + \tfrac{\mu}{2}\|x-y\|^2$$

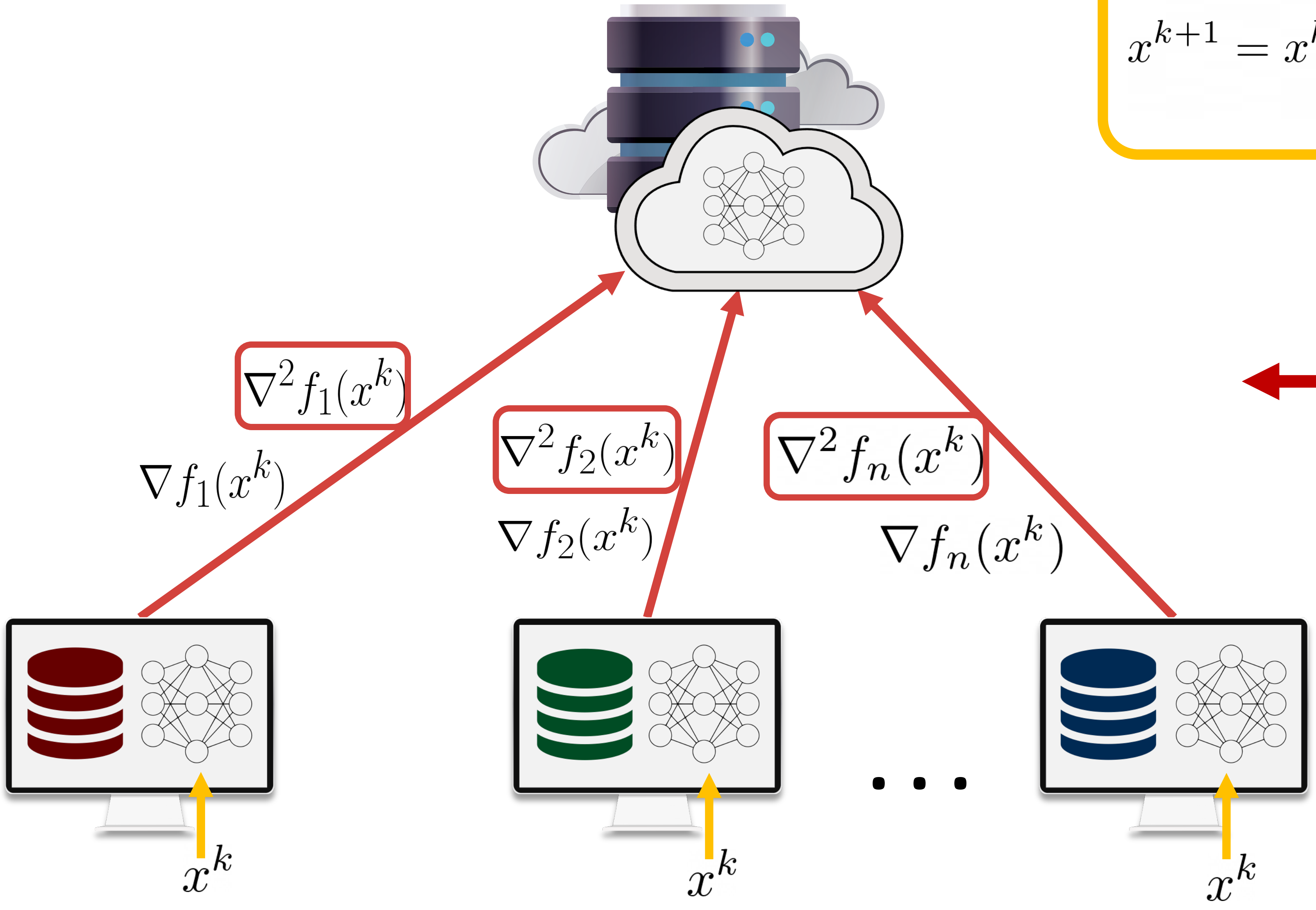
Distributed Implementation of Newton's method

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



Distributed Implementation of Newton's method

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



Communication issue

$\nabla^2 f_i(x^k)$ — d^2 floats

Newton’s Method

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

Newton's 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

The unique minimizer of $f(x)$

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

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

Newton Star Method

The unique minimizer of $f(x)$

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

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

Learning the Optimal Hessian Matrices

Newton Star

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

Learning the Optimal Hessian Matrices

Newton Star

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

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

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

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

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

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: 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, e.g.,

$$\alpha = 1$$

$$\alpha = 1 - \sqrt{1 - \delta}$$

$$\alpha = \frac{1}{\omega + 1}$$

FedNL: New Hessian Learning Technique

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

Compression operator

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

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

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

Greedy sparsification (Top-K)

$$\begin{bmatrix} -0.4 & 12.1 & 0.76 \\ 2.8 & -9.7 & -1.1 \\ 0.24 & 4.5 & 0.9 \end{bmatrix} \xrightarrow{K=2} \begin{bmatrix} 0 & 12.1 & 0 \\ 0 & -9.7 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Random sparsification (Rand-K)

$$\begin{bmatrix} -0.4 & 12.1 & 0.76 \\ 2.8 & -9.7 & -1.1 \\ 0.24 & 4.5 & 0.9 \end{bmatrix} \xrightarrow{K=2} \frac{9}{2} \begin{bmatrix} -0.4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 4.5 & 0 \end{bmatrix}$$

factor preserving unbiasedness

FedNL: Local Convergence Theory

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

Local (fixed) linear rate

FedNL: Local Convergence Theory

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

Local (fixed) linear rate

Lyapunov function

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

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

Local linear rates

$x^k \to x^*$

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

Constant depending on the choice of the compressor and stepsize

FedNL: Local Convergence Theory

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

Local (fixed) linear rate

Lyapunov function

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

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

Local linear rates

$x^k \to x^*$

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

Constant depending on the choice of the compressor and stepsize

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

Local superlinear rate

Experiments: Regularized Logistic Regression

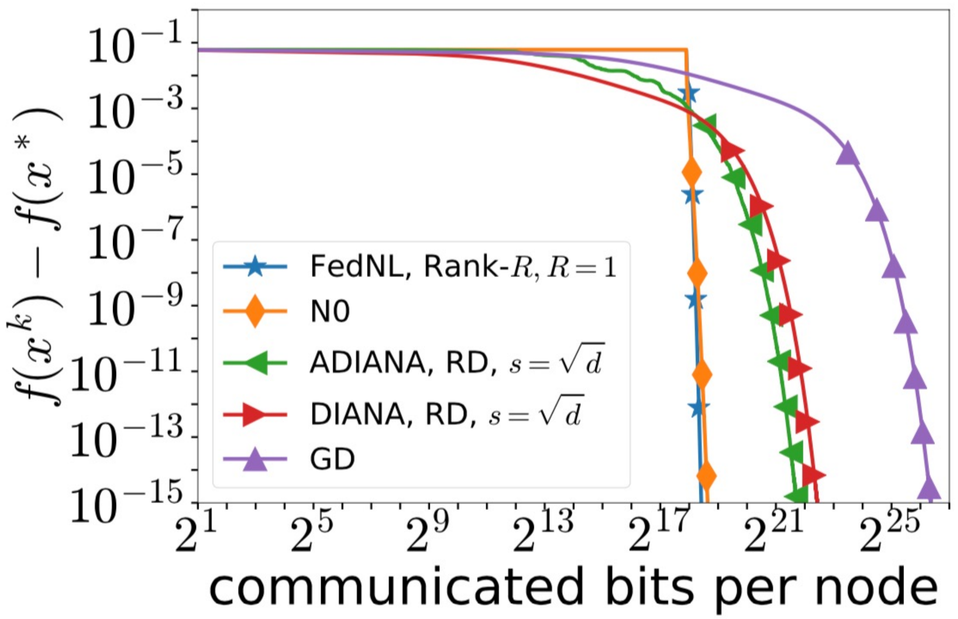
Regularization parameter

Training data points

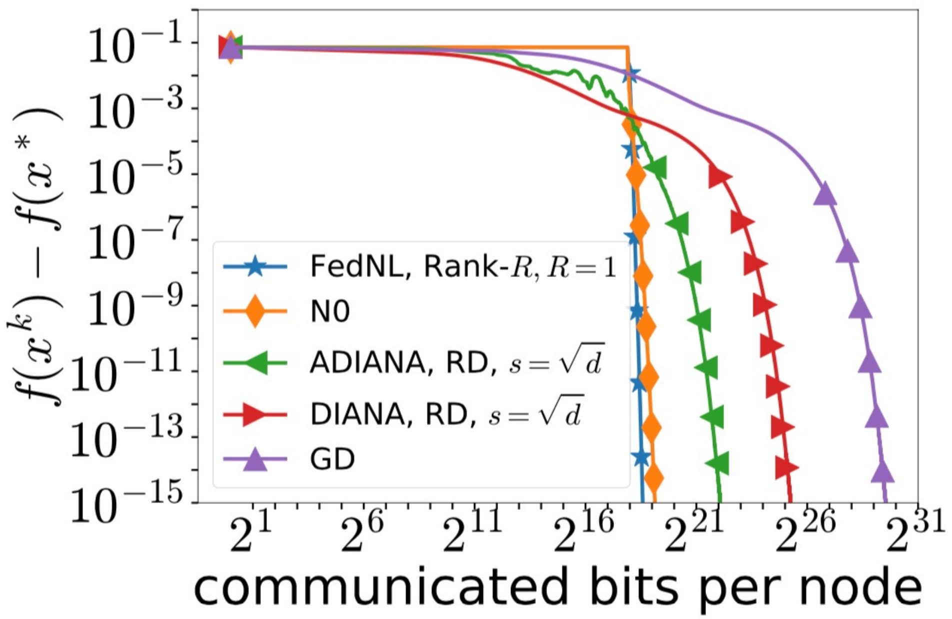
$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \overset{\downarrow}{\frac{\lambda}{2}}\|x\|^2\right\}, \qquad f_i(x) = \frac{1}{m}\sum_{j=1}^{m}\log\Big(1+\exp(-b_{ij}\overset{\downarrow}{a_{ij}^\top} x)\Big),$$

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: FedNL vs Gradient-Type Methods

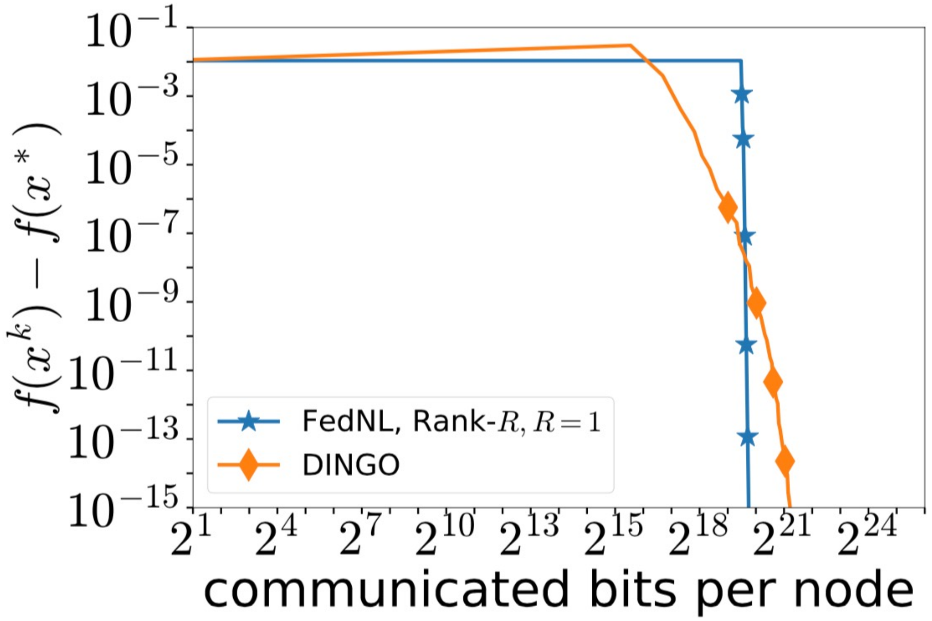


(a) a1a, $\lambda = 10^{-3}$

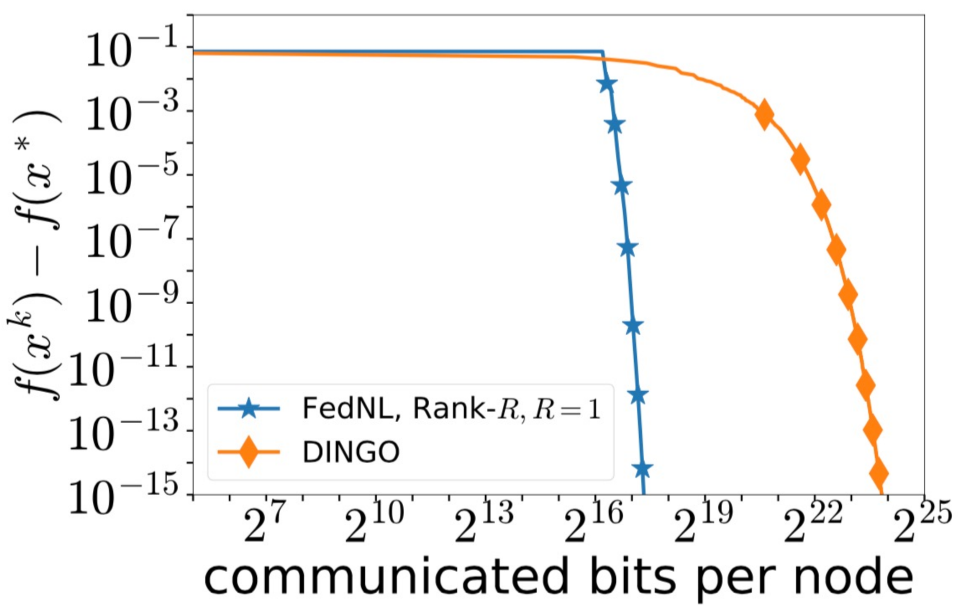


(b) a9a, $\lambda = 10^{-4}$

Experiments: FedNL vs DINGO

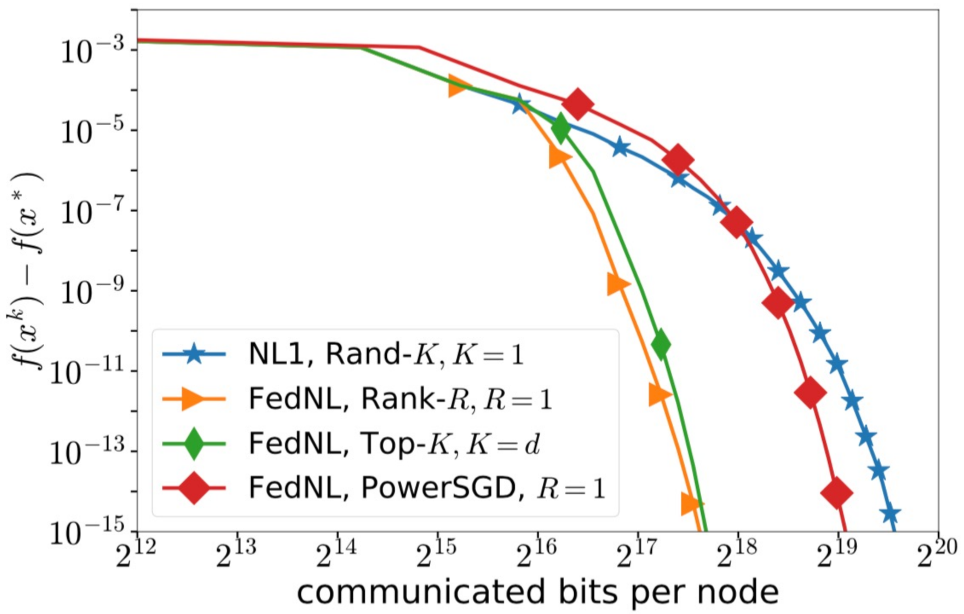


(c) w8a, $\lambda = 10^{-3}$

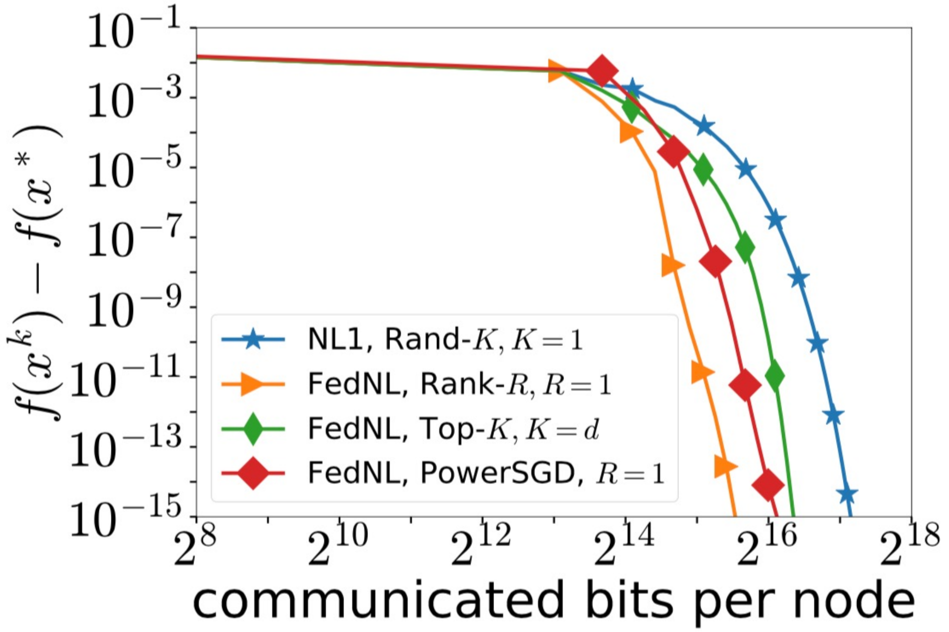


(d) phishing, $\lambda = 10^{-4}$

Experiments: FedNL vs NEWTON-LEARN (NL)



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



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

Thank you