Distributed and Stochastic Optimization Methods with Gradient Compression and Local Steps

Eduard Gorbunov

Ph.D. defense

Scientific supervisor: Alexander Gasnikov

Scientific advisor: Peter Richtárik

MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

December 23, 2021

Outline

1. Unified theory of SGD
2. Distributed Optimization
3. Unified theory of Error-Feedback SGD
4. Unified theory of Local-SGD

$\left.\right\}$ convex and strongly convex problems

5. Faster distributed methods with compression for non-convex optimization
6. Decentralized fault-tolerant optimization

1. Unified Theory of SGD

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. *"A Unified Theory of SGD: Variance reduction, Sampling, Quantization and Coordinate Descent."* In International Conference on Artificial Intelligence and Statistics, pp. 680-690. 2020.\

Stochastic/Finite-Sum Optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

Stochastic optimization

$$f(x) = \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_\xi(x) \right]$$

Finite-sum optimization

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

- $\nabla f(x)$ is too expensive to compute
- An unbiased stochastic estimator of $\nabla f(x)$ can be computed efficiently

Stochastic Gradient Descent

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right)$$

$$x^{k+1} = x^k - \gamma g^k$$

Stochastic gradient

Stochastic Gradient Descent

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right)$$

$$x^{k+1} = x^k - \gamma g^k$$

Stochastic gradient

How to choose the stochastic gradient?

Stochastic Gradient

Infinitely many ways of getting unbiased estimator with «good» properties

- Flexibility to construct stochastic gradients in order to target desirable properties:
  - convergence speed
  - iteration cost
  - overall complexity
  - parallelizability
  - communication cost and etc.

Stochastic Gradient

Infinitely many ways of getting unbiased estimator with «good» properties

- Flexibility to construct stochastic gradients in order to target desirable properties:
  - convergence speed
  - iteration cost
  - overall complexity
  - parallelizability
  - communication cost and etc.

- Too many methods
  - hard to keep up with new results
  - challenges in terms of the analysis
  - problems with a fair comparison: different assumptions are used in different papers

The First Problem

A single unifying theoretical framework for different variants of SGD is required

The first contribution of the dissertation

Key Parametric Assumption

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right)$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \le 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B\sigma_k^2 + D_1$$

$$\mathbb{E}\left[\sigma_{k+1}^2 \mid x^k, \sigma_k^2\right] \le (1-\rho)\sigma_k^2 + 2C\left(f\left(x^k\right) - f\left(x^*\right)\right) + D_2$$

Key Parametric Assumption

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right)$$

$$\boxed{\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \leq 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B\sigma_k^2 + D_1}$$

$$\mathbb{E}\left[\sigma_{k+1}^2 \mid x^k, \sigma_k^2\right] \leq (1-\rho)\sigma_k^2 + 2C\left(f\left(x^k\right) - f\left(x^*\right)\right) + D_2$$

Reflects smoothness properties of the problem and noises introduced by stochastic gradients

Key Parametric Assumption

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right)$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \le 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B\sigma_k^2 + D_1$$

$$\mathbb{E}\left[\sigma_{k+1}^2 \mid x^k, \sigma_k^2\right] \le (1-\rho)\sigma_k^2 + 2C\left(f\left(x^k\right) - f\left(x^*\right)\right) + D_2$$

Reflects smoothness properties of the problem and noises introduced by stochastic gradients

Describes the process of variance reduction

Additional Assumption

Generalization of strong convexity – quasi-strong convexity:

$$f\left(x^{*}\right) \geq f(x)+\langle\nabla f(x), x^{*}-x\rangle+\frac{\mu}{2}\left\|x^{*}-x\right\|^{2}$$

Main Theorem

If the stepsize satisfies

$$0 < \gamma \leq \min\left\{\frac{1}{\mu}, \frac{1}{A + CM}\right\},$$ where $$M > \frac{B}{\rho}$$

Main Theorem

If the stepsize satisfies

$$0 < \gamma \leq \min\left\{\frac{1}{\mu}, \frac{1}{A + CM}\right\}, \quad \text{where} \quad M > \frac{B}{\rho}$$

then the iterates of SGD satisfy

$$\mathbb{E}\left[V^k\right] \leq \max\left\{(1-\gamma\mu)^k, \left(1 + \frac{B}{M} - \rho\right)^k\right\} V^0 + \frac{(D_1 + MD_2)\,\gamma^2}{\min\left\{\gamma\mu, \rho - \frac{B}{M}\right\}}$$

where $$V^k \stackrel{\text{def}}{=} \left\|x^k - x^*\right\|^2 + M\gamma^2\sigma_k^2$$

Table 2.1: List of specific existing (in some cases generalized) and new methods which fit our general analysis framework. VR = variance reduced method, AS = arbitrary sampling, Quant = supports gradient quantization, RCD = randomized coordinate descent type method. [a] Special case of SVRG with 1 outer loop only; [b] Special case of DIANA with 1 node and quantization of exact gradient.

Problem	Method	Alg #	Citation	VR?	AS?	Quant?	RCD?	Section	Result
(2.1)+(2.2)	SGD	Alg 1	[153]	✗	✗	✗	✗	2.6.1	Cor 2.6.2
(2.1)+(2.3)	SGD-SR	Alg 2	[60]	✗	✓	✗	✗	2.6.2	Cor 2.6.5
(2.1)+(2.3)	SGD-MB	Alg 3	**NEW**	✗	✗	✗	✗	2.6.3	Cor 2.6.9
(2.1)+(2.3)	SGD-star	Alg 4	**NEW**	✓	✓	✗	✗	2.6.4	Cor 2.6.12
(2.1)+(2.3)	SAGA	Alg 5	[35]	✓	✗	✗	✗	2.6.5	Cor 2.6.15
(2.1)+(2.3)	N-SAGA	Alg 6	**NEW**	✗	✗	✗	✗	2.6.6	Cor 2.6.17
(2.1)	SEGA	Alg 7	[66]	✓	✗	✗	✓	2.6.7	Cor 2.6.19
(2.1)	N-SEGA	Alg 8	**NEW**	✗	✗	✗	✓	2.6.8	Cor 2.6.21
(2.1)+(2.3)	SVRG[a]	Alg 9	[79]	✓	✗	✗	✗	2.6.9	Cor 2.6.23
(2.1)+(2.3)	L-SVRG	Alg 10	[74]	✓	✗	✗	✗	2.6.10	Cor 2.6.25
(2.1)+(2.3)	DIANA	Alg 11	[136]	✗	✗	✓	✗	2.6.11	Cor 2.6.28
(2.1)+(2.3)	DIANA[b]	Alg 12	[136]	✓	✗	✓	✗	2.6.11	Cor 2.6.29
(2.1)+(2.3)	Q-SGD-SR	Alg 13	**NEW**	✗	✓	✓	✗	2.6.12	Cor 2.6.31
(2.1)+(2.3)+(4.3)	VR-DIANA	Alg 14	[76]	✓	✗	✓	✗	2.6.13	Cor 2.6.34
(2.1)+(2.3)	JacSketch	Alg 15	[59]	✓	✓✗	✗	✗	2.6.14	Cor 2.6.37

In one theorem, we recover the sharpest rates for all known special cases

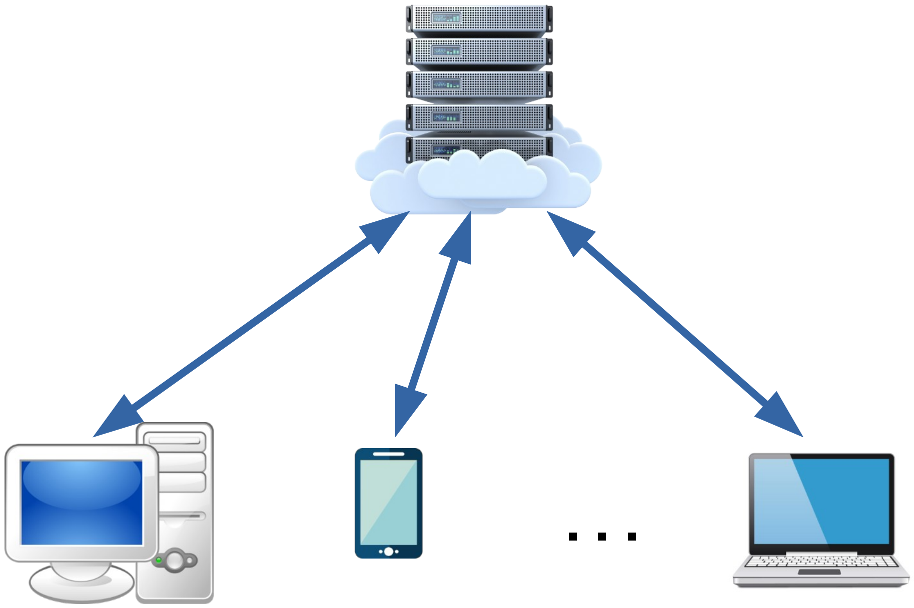
2. Distributed Optimization

Distributed Optimization

- Some problems cannot be solved on a single a machine in a reasonable time (deep learning models with billions of parameters and gigabytes of data)

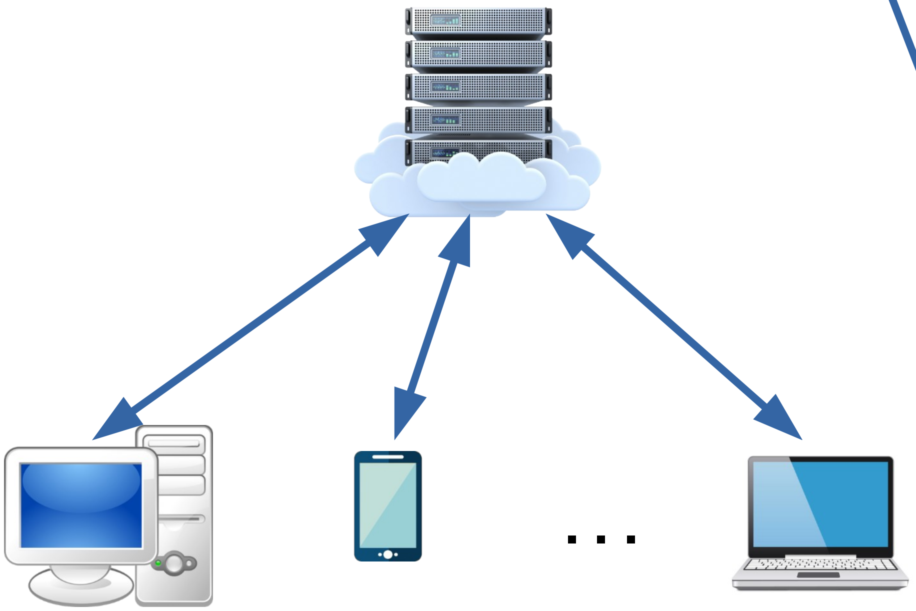
- There exist such problems where the data that defines the optimization problem is private and distributed among several machines (federated learning)

These problems are typically solved in a distributed way



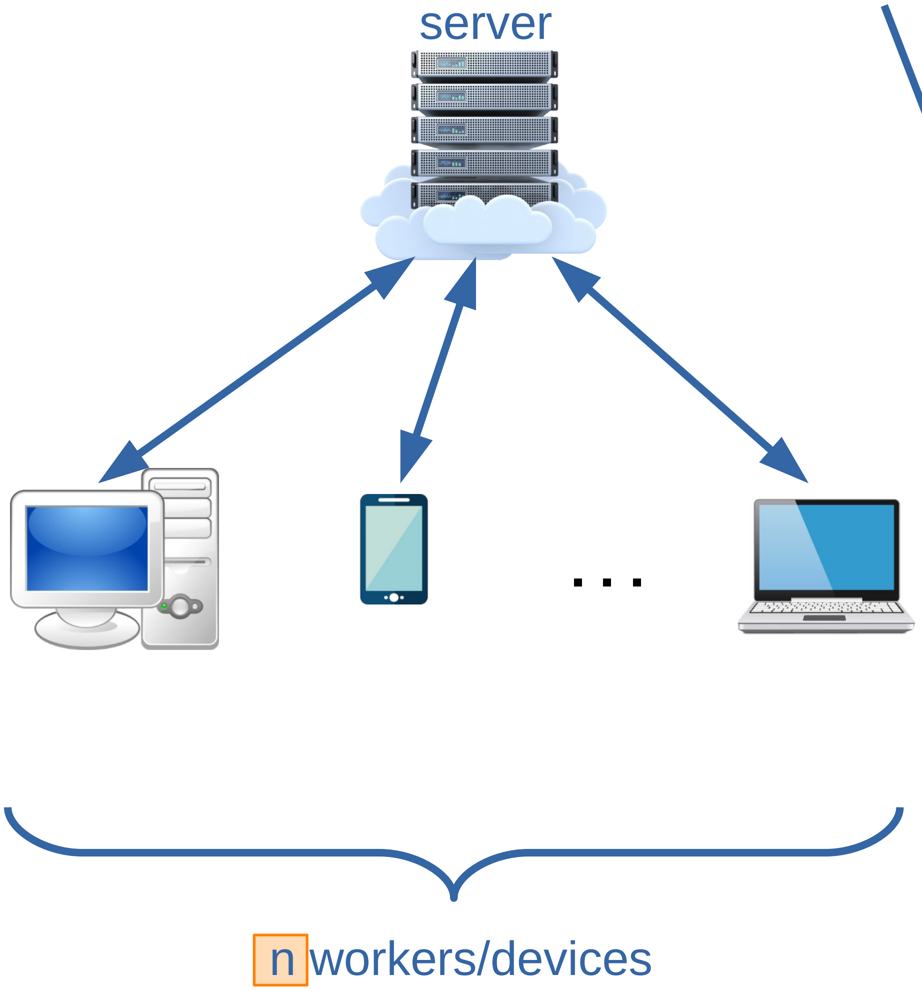
$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}$$

parameters of the model



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

# of parameters

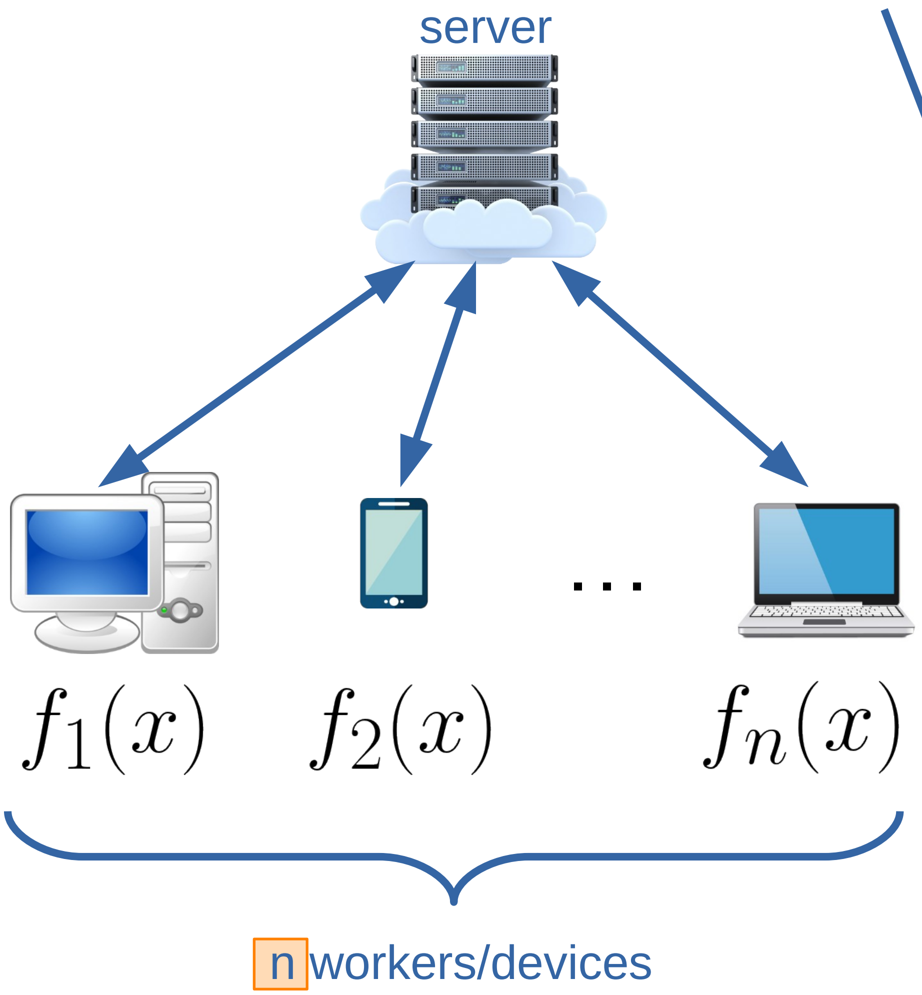


parameters of the model

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

# of parameters

# of workers/devices



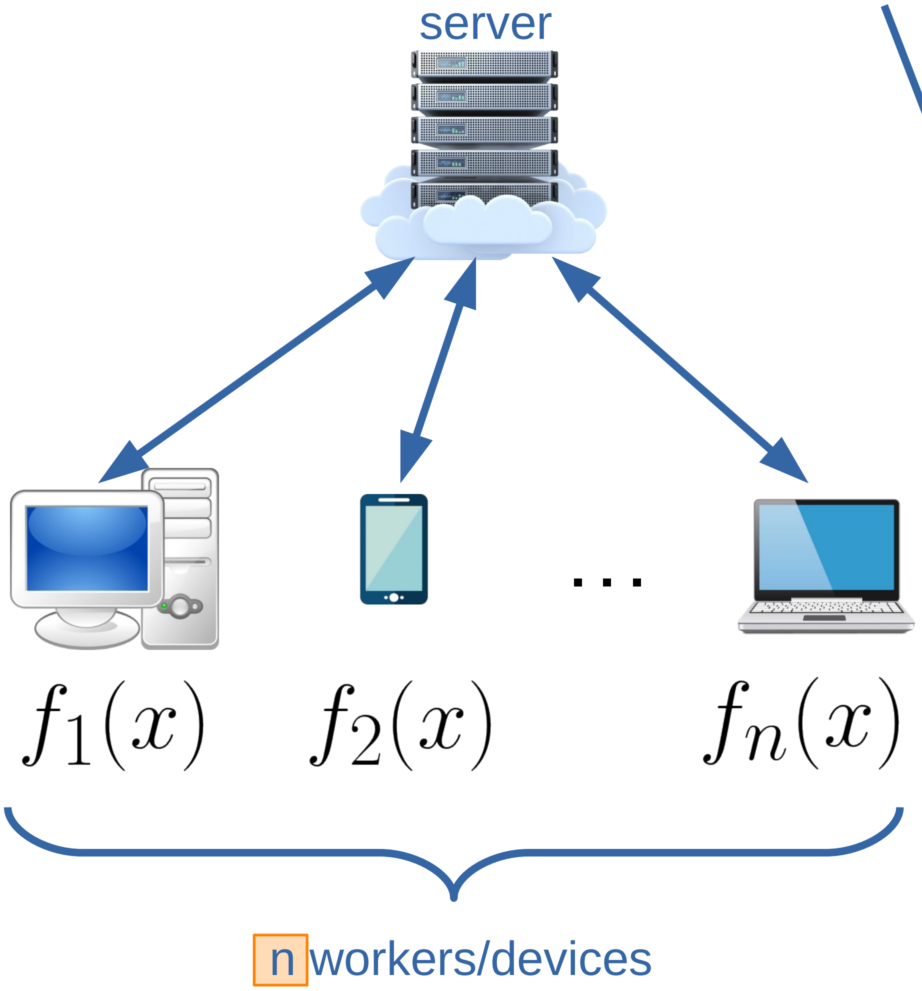
parameters of the model

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

# of parameters

# of workers/devices

loss on the data accessible by worker *i*



parameters of the model

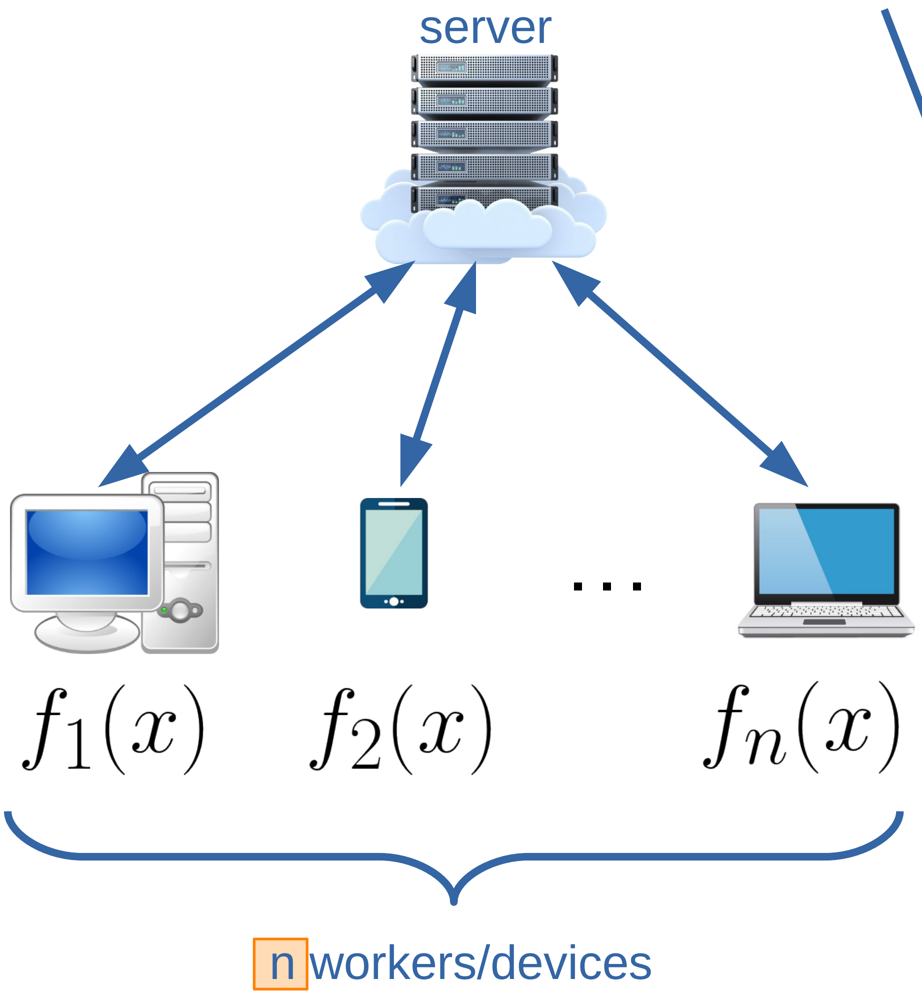
$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{\boxed{n}}\sum_{i=1}^{\boxed{n}} f_i(x)\right\}$$

# of parameters

# of workers/devices

loss on the data accessible by worker *i*

- $f_i(x) = \mathbf{E}_{\xi_i\sim\mathcal{D}_i}\left[f_{\xi_i}(x)\right]$



parameters of the model

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

\# of parameters

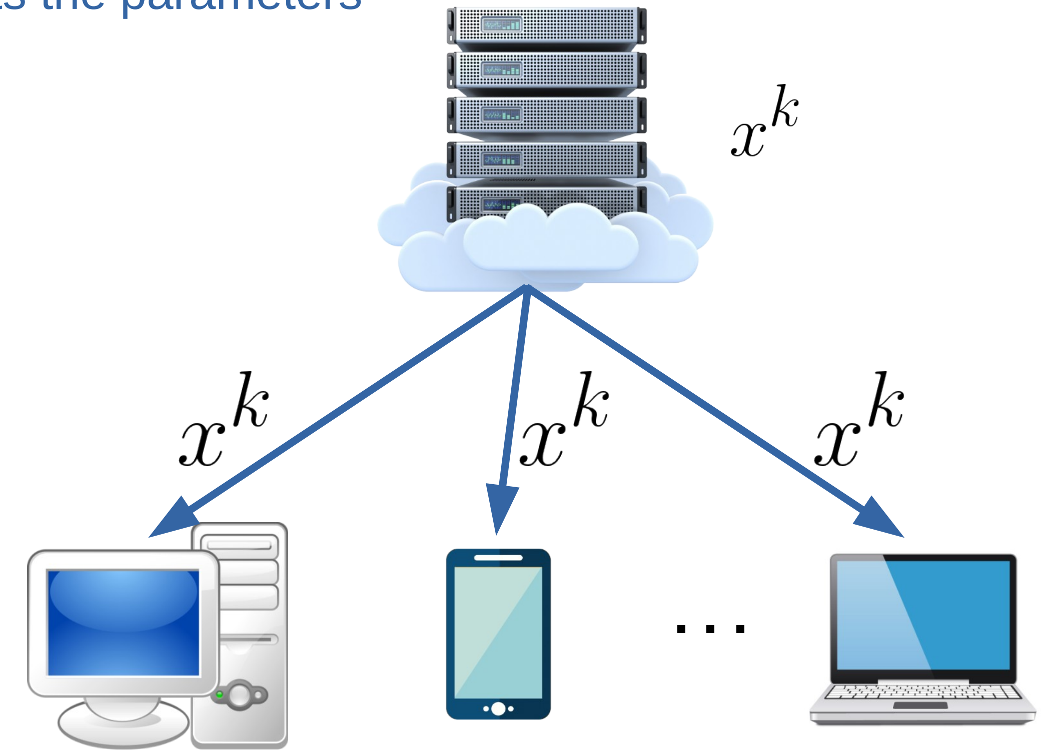
\# of workers/devices

loss on the data accessible by worker *i*

- $f_i(x) = \mathbf{E}_{\xi_i \sim \mathcal{D}_i} \left[f_{\xi_i}(x) \right]$
- $f_i(x) = \frac{1}{m} \sum_{j=1}^{m} f_{ij}(x)$

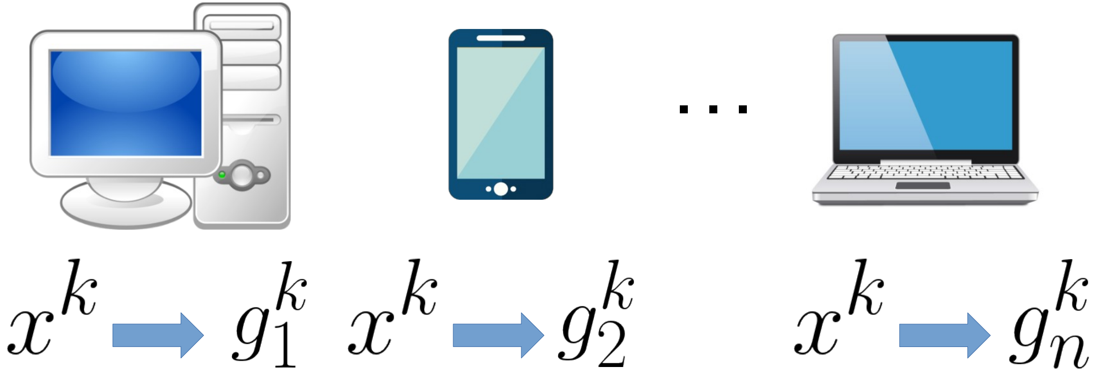
Parallel SGD

1 Server broadcasts the parameters



Parallel SGD

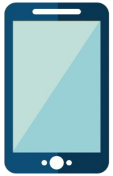
1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel



Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel

x^k



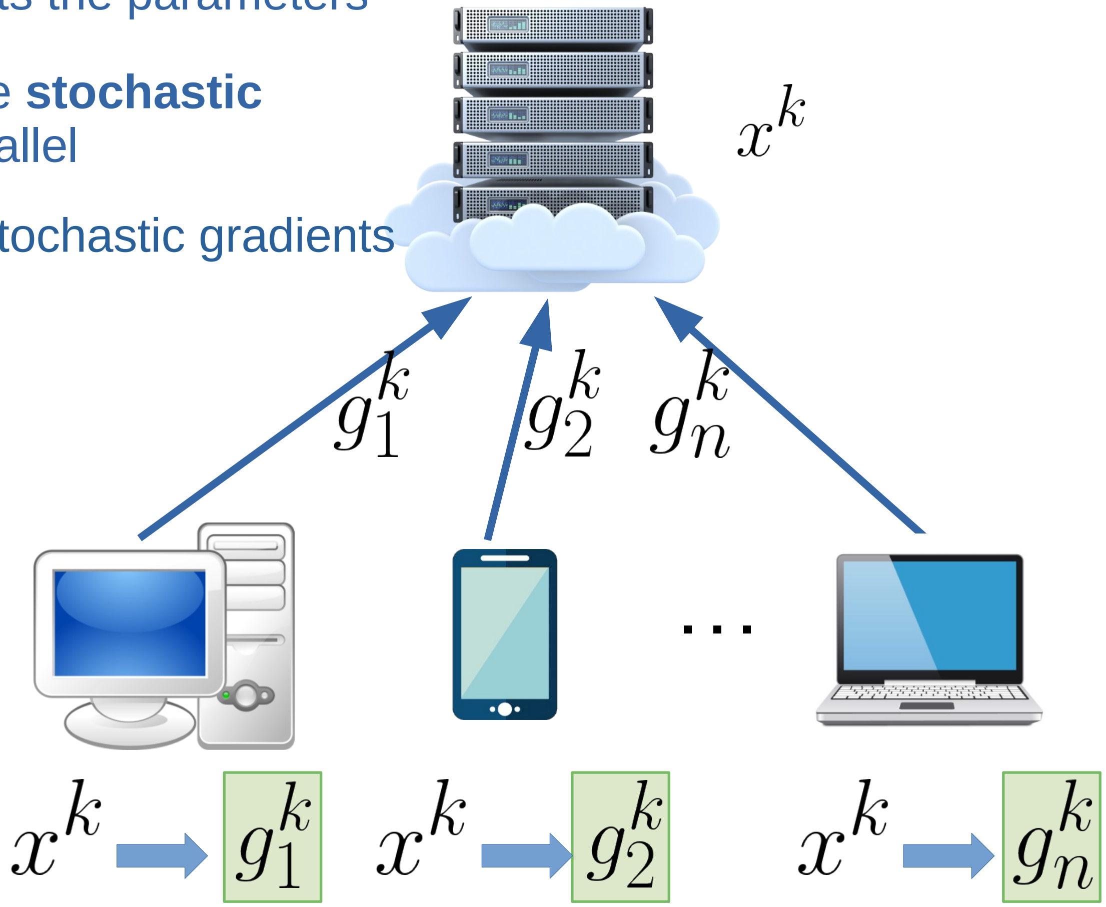
$\cdots$

$$x^k \rightarrow \boxed{g_1^k} \quad x^k \rightarrow \boxed{g_2^k} \quad x^k \rightarrow \boxed{g_n^k}$$

$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients

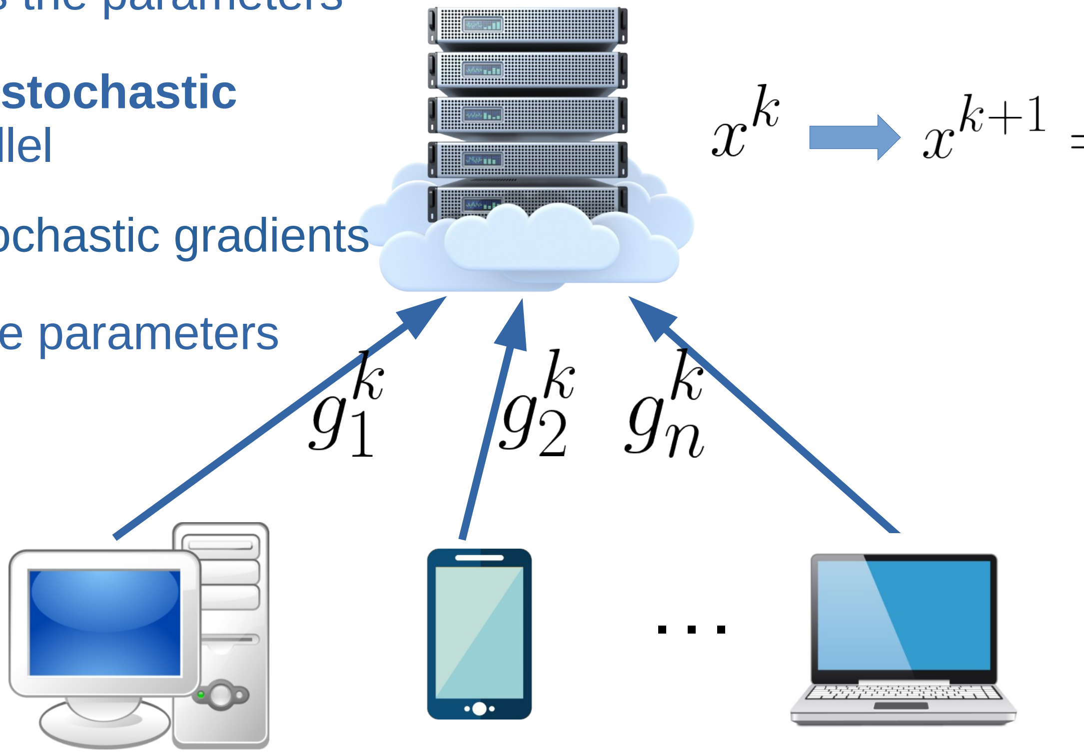


$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters

$$x^k \Rightarrow x^{k+1} = x^k - \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} g_i^k$$



$$x^k \Rightarrow \boxed{g_1^k} \qquad x^k \Rightarrow \boxed{g_2^k} \qquad x^k \Rightarrow \boxed{g_n^k}$$

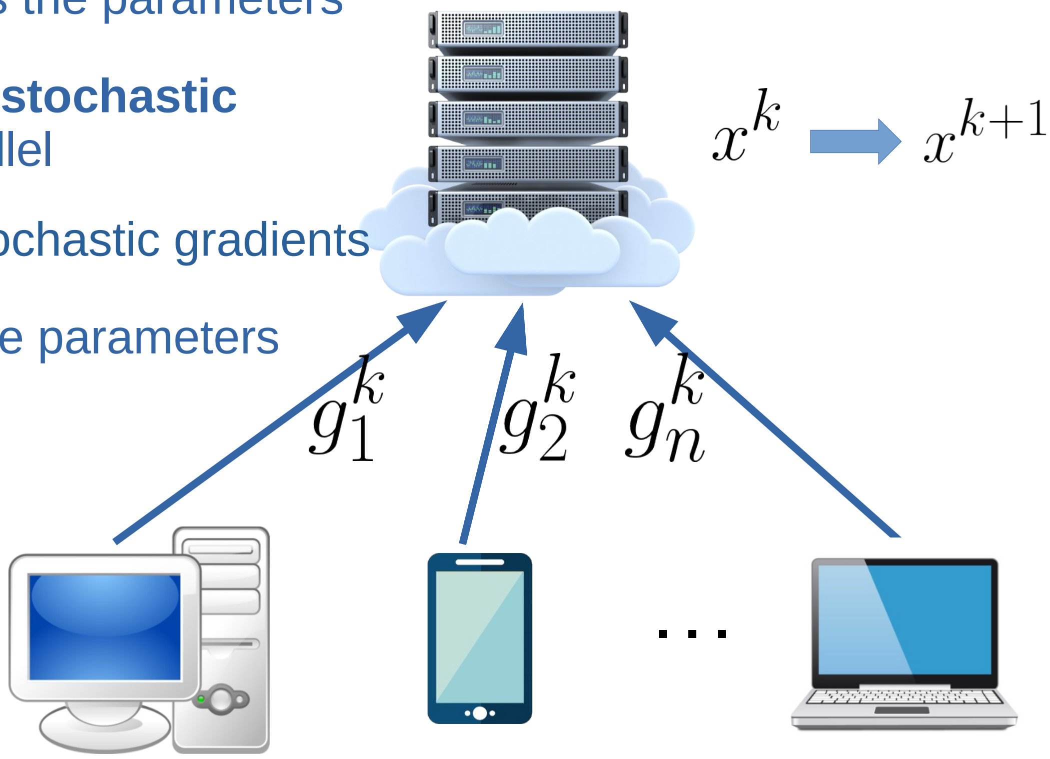
$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters

stepsize

$$x^k \Rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$



$$x^k \Rightarrow \boxed{g_1^k} \qquad x^k \Rightarrow \boxed{g_2^k} \qquad x^k \Rightarrow \boxed{g_n^k}$$

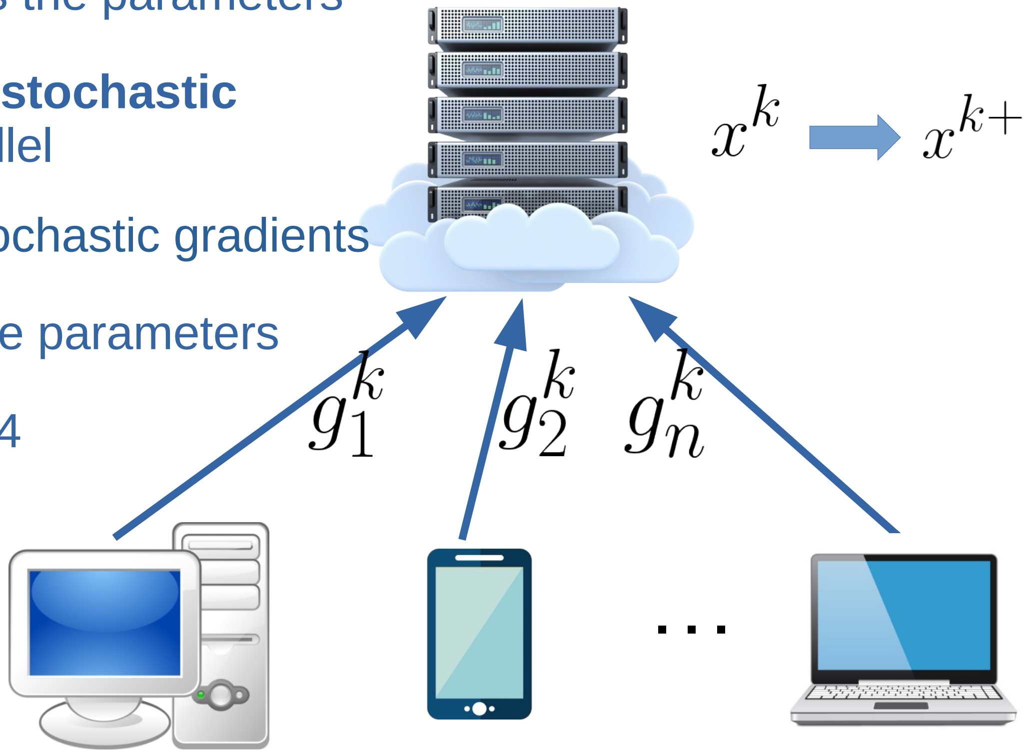
$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters
5. Repeat steps 1 – 4

stepsize

$$x^k \Rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$



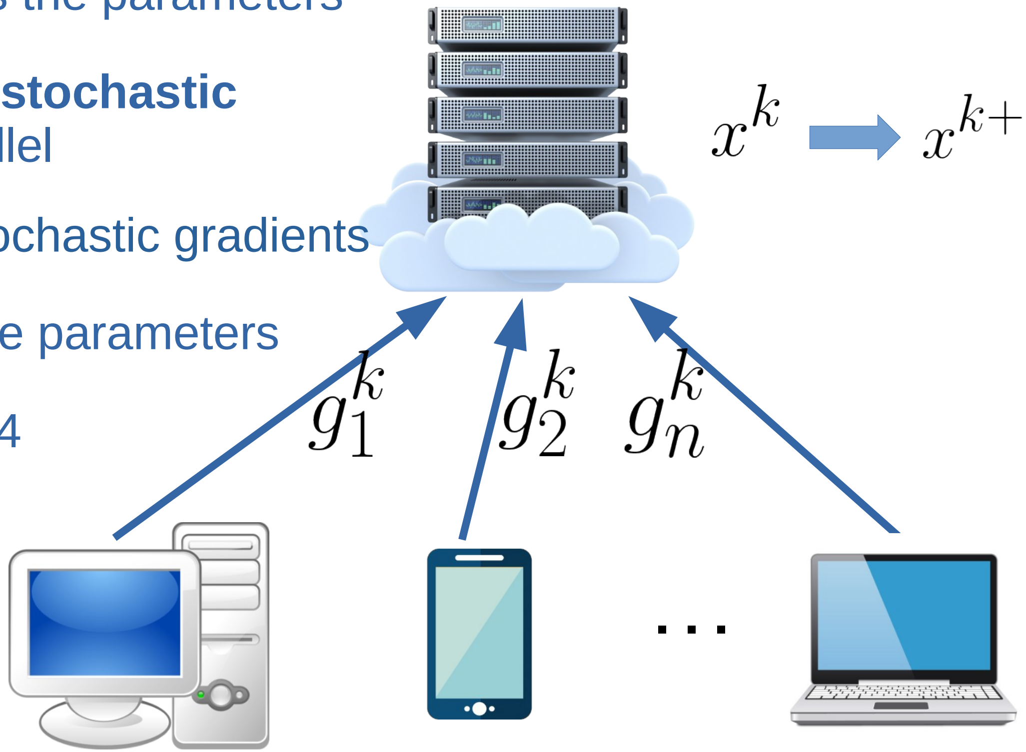
$g_1^k \quad g_2^k \quad g_n^k$

$x^k \Rightarrow \boxed{g_1^k} \qquad x^k \Rightarrow \boxed{g_2^k} \qquad \ldots \qquad x^k \Rightarrow \boxed{g_n^k}$

$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters
5. Repeat steps 1 – 4



stepsize

$$x^k \Rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$

Good news:

- Very simple algorithm
- Can be much faster than non-parallel SGD

$$x^k \Rightarrow \boxed{g_1^k} \qquad x^k \Rightarrow \boxed{g_2^k} \qquad x^k \Rightarrow \boxed{g_n^k}$$

$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

Parallel SGD

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters
5. Repeat steps 1 – 4

Good news:

- Very simple algorithm
- Can be much faster than non-parallel SGD

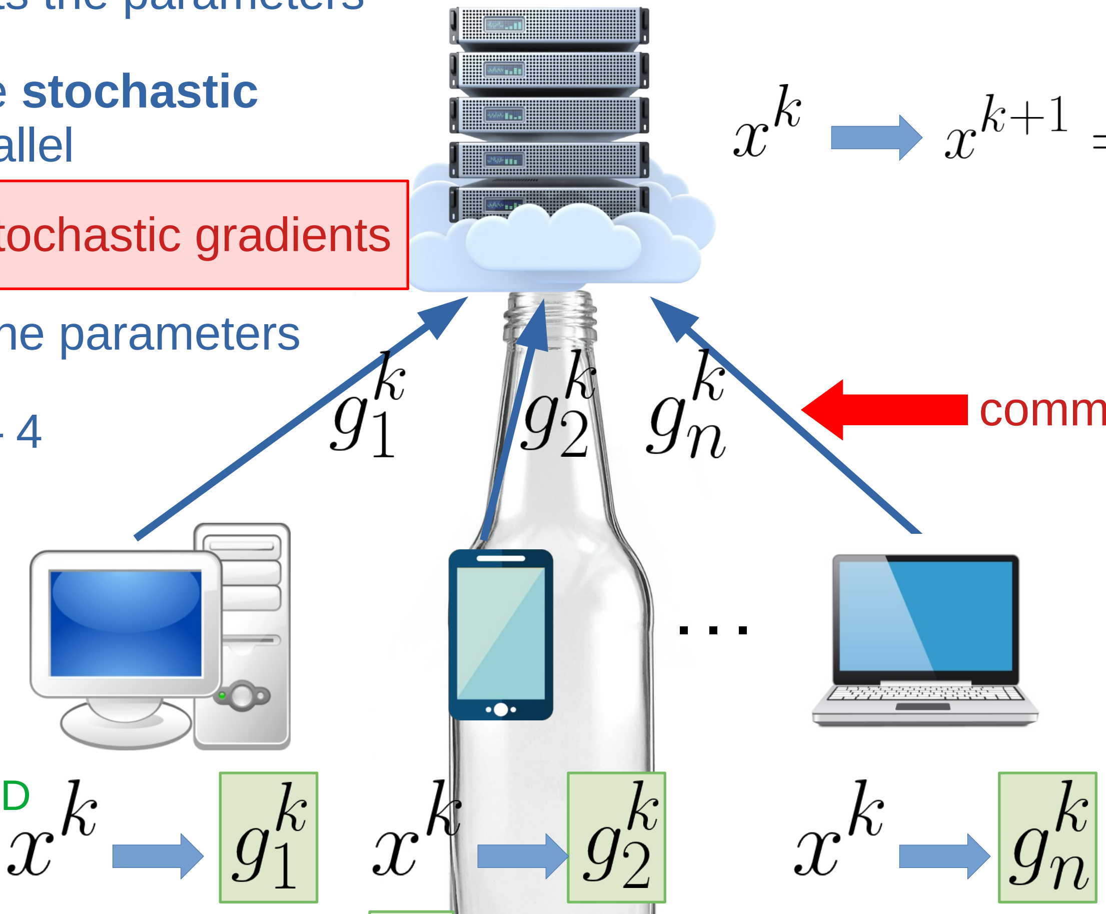
Issues:

- Overload of the server

stepsize

$$x^k \rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$

communication is a bottleneck



$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

3. Unified theory of Error-Feedback SGD

Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtarik, *Linearly Converging Error Compensated SGD*. Advances in Neural Information Processing Systems, 33, 2020.

Compression Operators

Unbiased compressors
(quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

Biased compressors

$$x \to \mathcal{C}(x)$$

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega \|x\|^2$$

Biased compressors

$$x \to \mathcal{C}(x)$$

Compression Operators

Unbiased compressors
(quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega\|x\|^2$$

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2$$

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega\|x\|^2$$

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2$$

Example: RandK (for K = $\boxed{2}$)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \frac{5}{\boxed{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix}$$

for unbiasedness

Pick K = $\boxed{2}$ components uniformly at random

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega \|x\|^2$$

Example: RandK (for K = $\boxed{2}$)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \frac{5}{\boxed{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix}$$

for unbiasedness

Pick K = $\boxed{2}$ components uniformly at random

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2$$

Example: TopK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 10 \end{pmatrix}$$

Pick K = 2 components with largest absolute value

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \le \omega \|x\|^2$$

Well studied in the (strongly) convex case

Example: RandK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \frac{5}{2} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix}$$

for unbiasedness

Pick K = 2 components uniformly at random

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \le (1-\delta)\|x\|^2$$

Example: TopK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 10 \end{pmatrix}$$

Pick K = 2 components with largest absolute value

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \le \omega\|x\|^2$$

Well studied in the (strongly) convex case

Example: RandK (for K = $\boxed{2}$)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \frac{5}{\boxed{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix}$$

for unbiasedness

Pick K = $\boxed{2}$ components uniformly at random

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \le (1-\delta)\|x\|^2$$

Much less is known, e.g., no linearly converging methods are developed

Example: TopK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 10 \end{pmatrix}$$

Pick K = 2 components with largest absolute value

The Second Problem

Theory of distributed methods with *biased* compression requires improvements

The second contribution of the dissertation

Parallel SGD with Biased Compressor Can Diverge at Exponential Rate

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

$$n = d = 3$$

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2$$

$$a = (-3, 2, 2)^\top \qquad b = (2, -3, 2)^\top \qquad c = (2, 2, -3)^\top$$

$$x^0 = (t, t, t)^\top$$

Parallel SGD with Biased Compressor Can Diverge at Exponential Rate

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

$$n = d = 3$$

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2$$

$$a = (-3, 2, 2)^\top \qquad b = (2, -3, 2)^\top \qquad c = (2, 2, -3)^\top$$

$$x^0 = (t, t, t)^\top$$

In this case Parallel SGD with Top1 compression operator satisfies

$$x^k = \left(1 + \frac{11\gamma}{6}\right)^k x^0$$

Parallel SGD with Biased Compressor Can Diverge at Exponential Rate

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

$$n = d = 3$$

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2$$

$$a = (-3, 2, 2)^\top \qquad b = (2, -3, 2)^\top \qquad c = (2, 2, -3)^\top$$

$$x^0 = (t, t, t)^\top$$

In this case Parallel SGD with Top1 compression operator satisfies

$$x^k = \left(1 + \frac{11\gamma}{6}\right)^k x^0$$

One can fix this using one special trick called ***error-compensation***

Error-Compensated SGD

Seide, Frank, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. **"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.

Stich, Sebastian U., Jean-Baptiste Cordonnier, and Martin Jaggi. **"Sparsified SGD with memory."** *In Advances in Neural Information Processing Systems*, pp. 4447-4458. 2018.

Karimireddy, Sai Praneeth, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. **"Error Feedback Fixes SignSGD and other Gradient Compression Schemes."** *In International Conference on Machine Learning*, pp. 3252-3261. 2019.

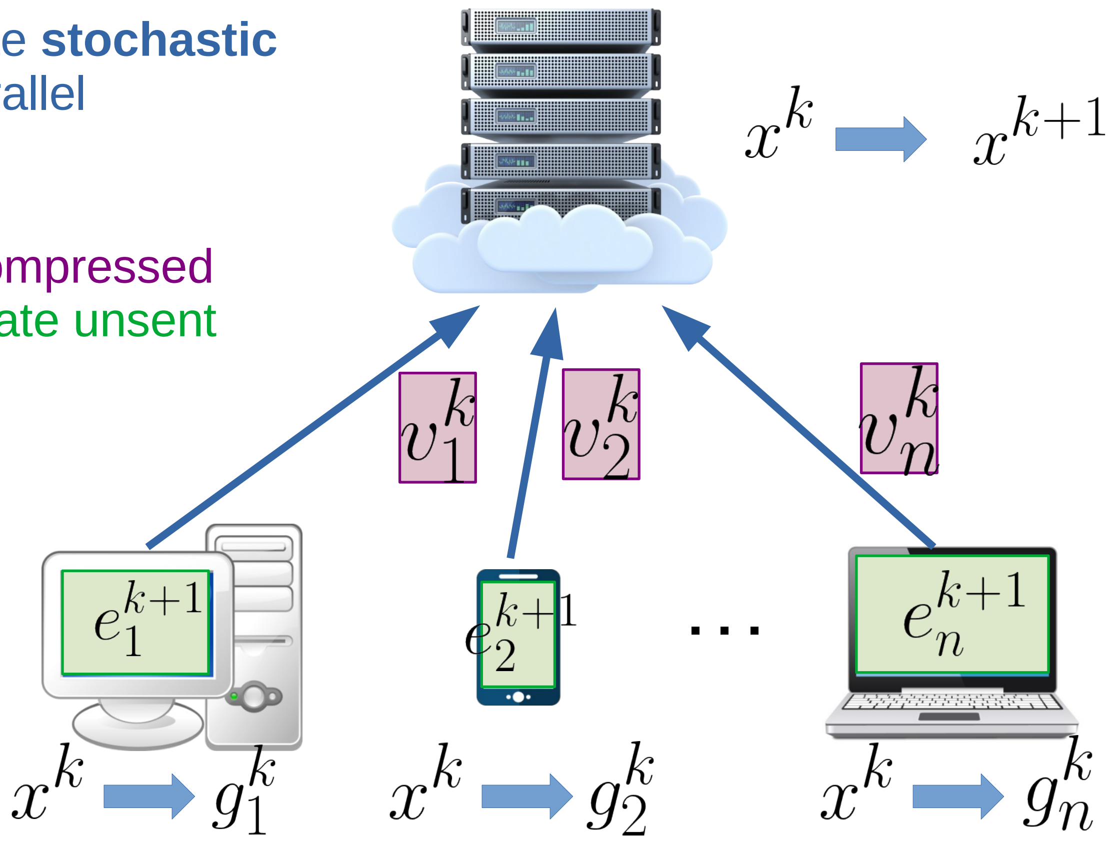
Stich, Sebastian U., and Sai Praneeth Karimireddy. **"The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication."** arXiv preprint arXiv:1909.05350 (2019).

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

Step k+1

1. Server broadcasts new parameters
2. Workers compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information
5. Server gathers the information and updates the parameters
6. Repeat steps 1 – 5

$$x^k \rightarrow x^{k+1} = x^k - \frac{1}{n}\sum_{i=1}^{n} v_i^k$$



$$\boxed{v_i^k = \mathcal{C}\left(e_i^k + \gamma g_i^k\right)} \qquad \boxed{e_i^{k+1} = e_i^k + \gamma g_i^k - v_i^k}$$

Key Assumption

$$g^k = \frac{1}{n}\sum_{i=1}^n g_i^k, \quad \mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right) \quad \bar{g}_i^k = \mathbb{E}\left[g_i^k \mid x^k\right]$$

$$\frac{1}{n}\sum_{i=1}^n \left\|\bar{g}_i^k\right\|^2 \le 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1\sigma_{1,k}^2 + B_2\sigma_{2,k}^2 + D_1$$

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|g_i^k - \bar{g}_i^k\right\|^2 \mid x^k\right] \le 2\widetilde{A}\left(f\left(x^k\right) - f\left(x^*\right)\right) + \widetilde{B}_1\sigma_{1,k}^2 + \widetilde{B}_2\sigma_{2,k}^2 + \widetilde{D}_1$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \le 2A'\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1'\sigma_{1,k}^2 + B_2'\sigma_{2,k}^2 + D_1'$$

$$\mathbb{E}\left[\sigma_{1,k+1}^2 \mid \sigma_{1,k}^2, \sigma_{2,k}^2\right] \le \left(1-\rho_1\right)\sigma_{1,k}^2 + 2C_1\left(f\left(x^k\right) - f\left(x^*\right)\right) + G\rho_1\sigma_{2,k}^2 + D_2$$

$$\mathbb{E}\left[\sigma_{2,k+1}^2 \mid \sigma_{2,k}^2\right] \le \left(1-\rho_2\right)\sigma_{2,k}^2 + 2C_2\left(f\left(x^k\right) - f\left(x^*\right)\right)$$

Reflects smoothness properties of the problem and noises introduced by compressions and stochastic gradients

Describes the process of variance reduction of the variance coming from compressions

Describes the process of variance reduction of the variance coming from stochastic gradients

Main Theorem

Some quantity depending only on the
starting point and stepsize

$$\mathbb{E}\left[f\left(\bar{x}^K\right) - f\left(x^*\right)\right] \leq (1-\eta)^K \frac{\Psi\left(x^0, \gamma\right)}{\gamma} + \gamma \Phi\left(D_1, \widetilde{D}_1, D_1', D_2\right)$$

$$\eta = \min\left\{\frac{\gamma\mu}{2}, \frac{\rho_1}{4}, \frac{\rho_2}{4}\right\}$$

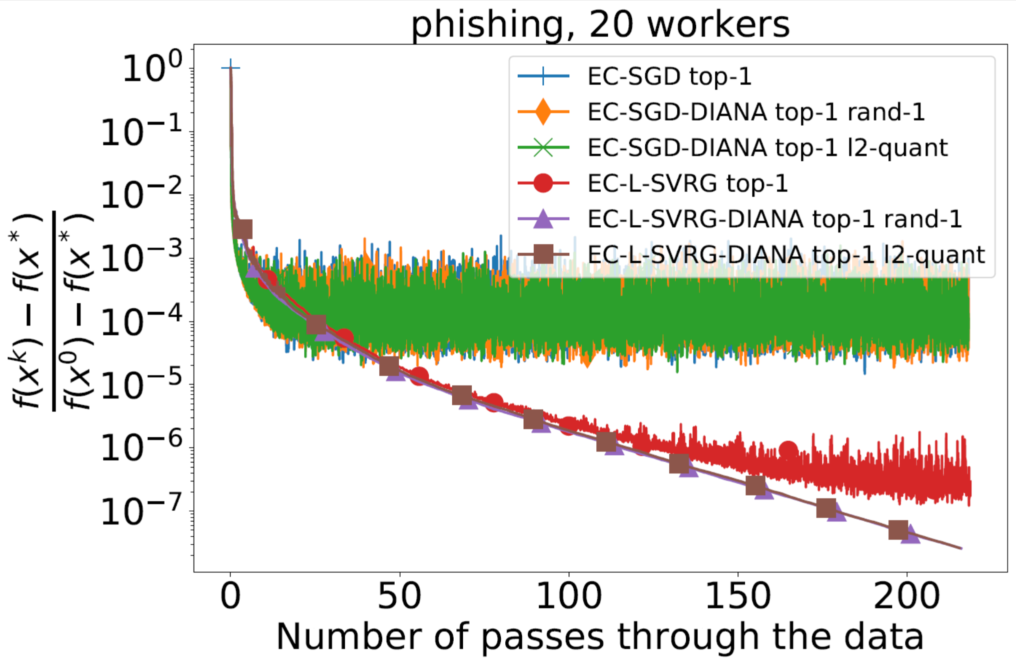
Linear function

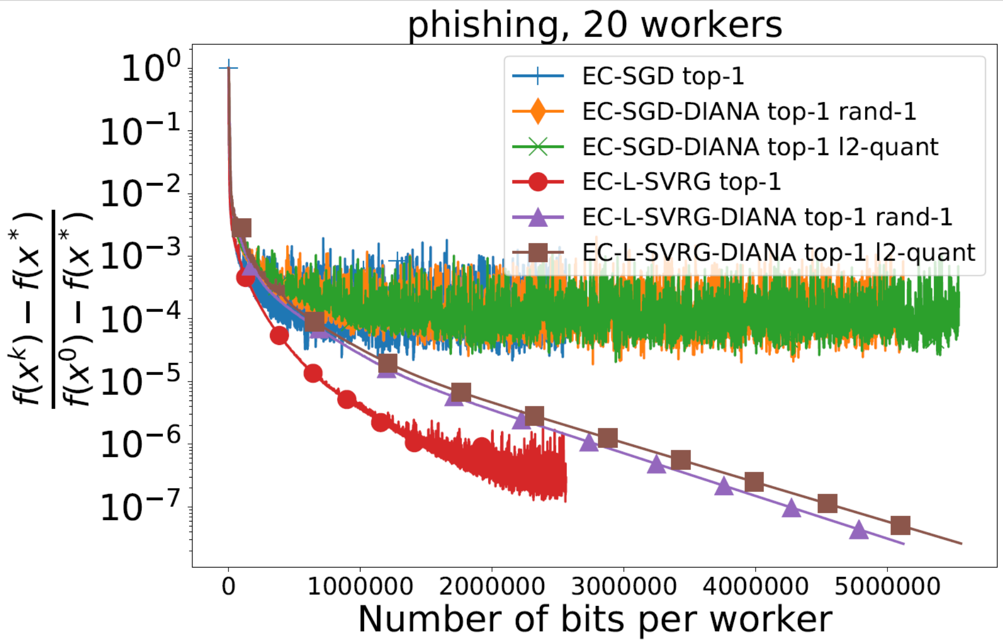
Methods with Error Compensation Covered by Our Framework

Problem	Method	Alg #	Citation	Sec #	Rate (constants ignored)
(3.1)+(3.3)	EC-SGDsr	Alg 19	**new**	3.8.1	$\widetilde{\mathcal{O}}\left(\frac{\mathcal{L}}{\mu}+\frac{L+\sqrt{\delta L \mathcal{L}}}{\delta\mu}+\frac{\sigma_*^2}{n\mu\varepsilon}+\frac{\sqrt{L(\sigma_*^2+\zeta_*^2/\delta)}}{\mu\sqrt{\delta\varepsilon}}\right)$
(3.1)+(3.2)	EC-SGD	Alg 20	[206]	3.8.2	$\widetilde{\mathcal{O}}\left(\frac{\kappa}{\delta}+\frac{\sigma_*^2}{n\mu\varepsilon}+\frac{\sqrt{L(\sigma_*^2+\zeta_*^2/\delta)}}{\delta\mu\sqrt{\varepsilon}}\right)$
(3.1)+(3.3)	EC-GDstar	Alg 21	**new**	3.8.3	$\mathcal{O}\left(\frac{\kappa}{\delta}\log\frac{1}{\varepsilon}\right)$
(3.1)+(3.2)	EC-SGD-DIANA	Alg 22	**new**	3.8.4	Opt. I: $\widetilde{\mathcal{O}}\left(\omega+\frac{\kappa}{\delta}+\frac{\sigma^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$ Opt. II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta}+\frac{\kappa}{\delta}+\frac{\sigma^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$
(3.1)+(3.3)	EC-SGDsr-DIANA	Alg 23	**new**	3.8.5	Opt. I: $\widetilde{\mathcal{O}}\left(\omega+\frac{\mathcal{L}}{\mu}+\frac{\sqrt{L\mathcal{L}}}{\delta\mu}+\frac{\sigma_*^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma_*^2}}{\delta\mu\sqrt{\varepsilon}}\right)$ Opt. II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta}+\frac{\mathcal{L}}{\mu}+\frac{\sqrt{L\mathcal{L}}}{\delta\mu}+\frac{\sigma_*^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma_*^2}}{\mu\sqrt{\delta\varepsilon}}\right)$
(3.1)+(3.2)	EC-GD-DIANA†	Alg 22	**new**	3.8.4	$\mathcal{O}\left(\left(\omega+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(3.1)+(3.3)	EC-LSVRG	Alg 24	**new**	3.8.6	$\widetilde{\mathcal{O}}\left(m+\frac{\kappa}{\delta}+\frac{\sqrt{L\zeta_*^2}}{\delta\mu\sqrt{\varepsilon}}\right)$
(3.1)+(3.3)	EC-LSVRGstar	Alg 25	**new**	3.8.7	$\mathcal{O}\left(\left(m+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(3.1)+(3.3)	EC-LSVRG-DIANA	Alg 26	**new**	3.8.8	$\mathcal{O}\left(\left(\omega+m+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$

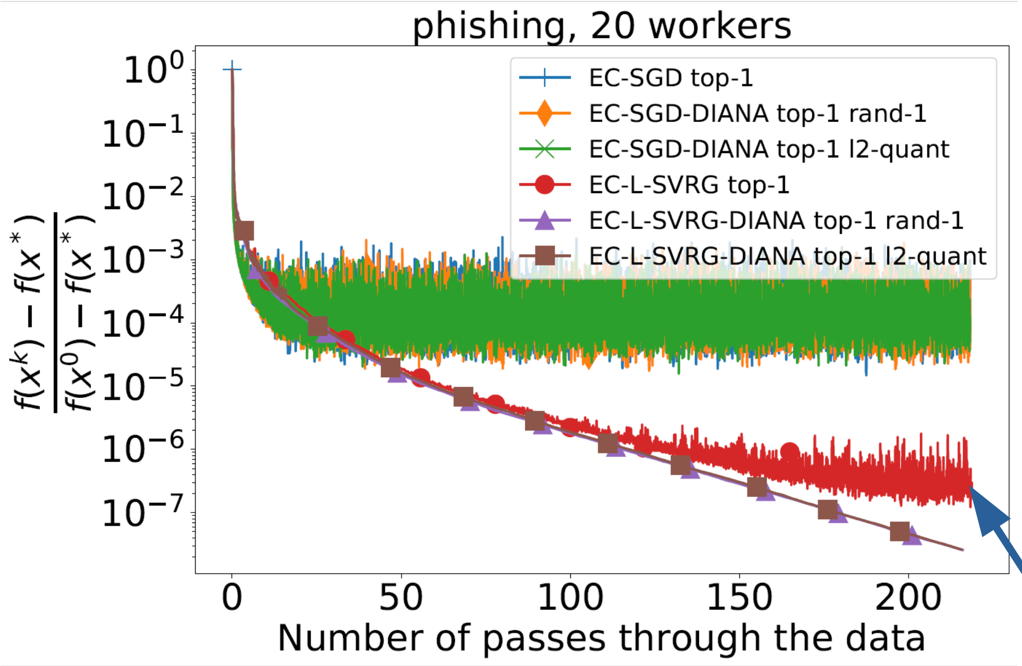
Our framework covers even methods without error compensation and methods with delayed updates

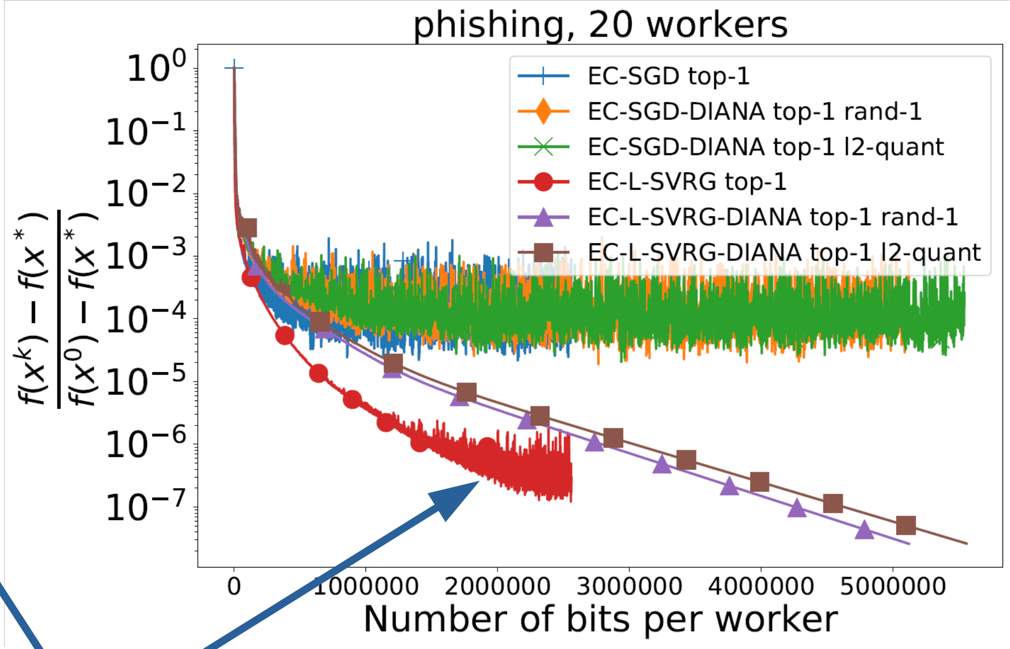
Logistic Regression with l2-regularization





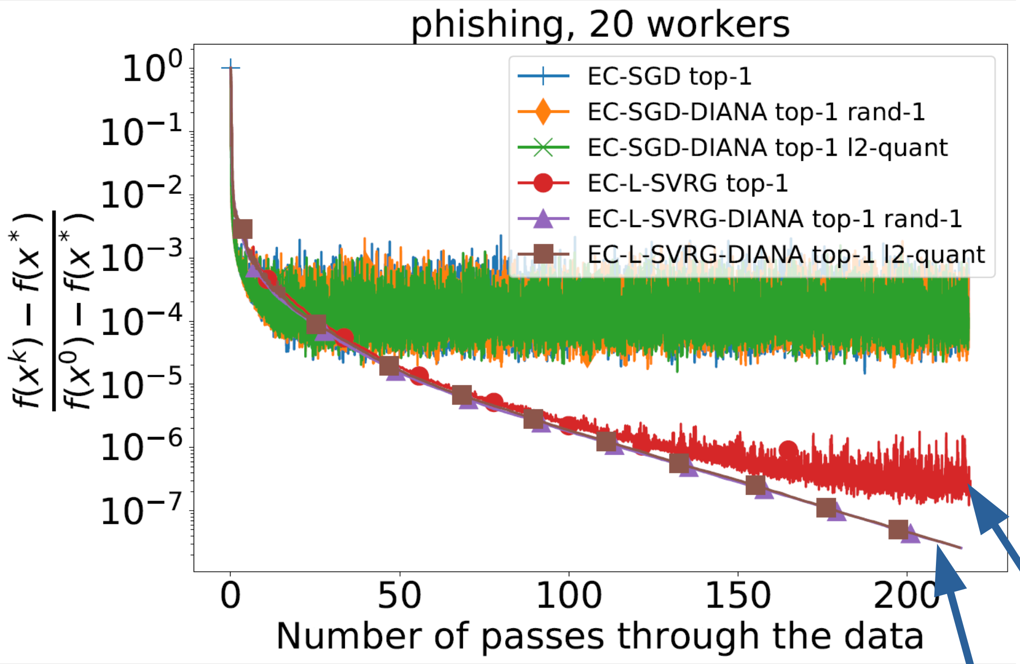
Logistic Regression with l2-regularization

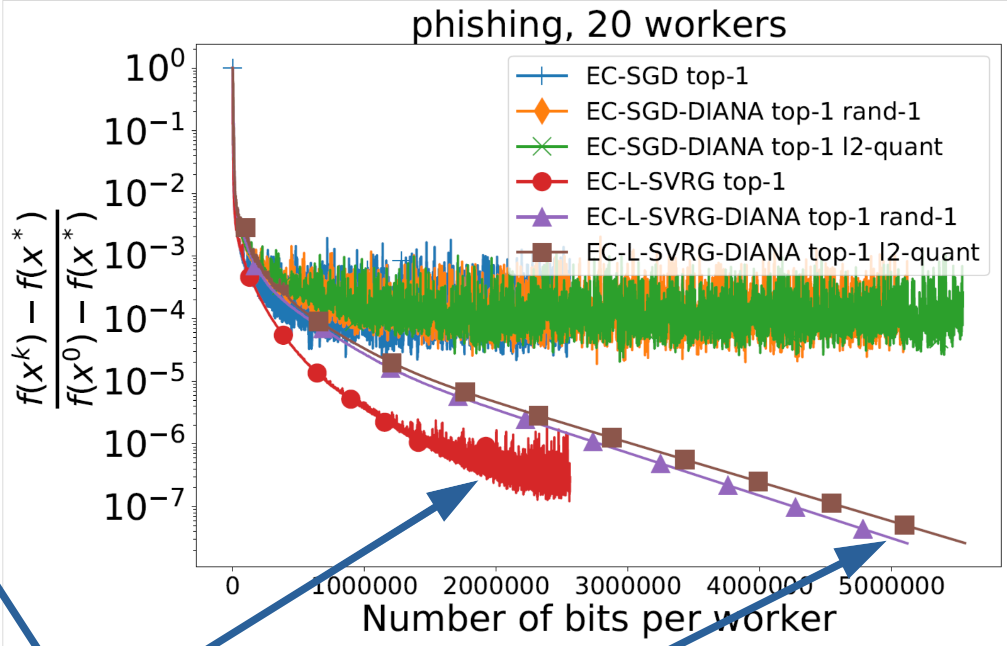




partial variance reduction

Logistic Regression with l2-regularization





partial variance reduction

full variance reduction

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

- Methods without error feedback: SGD, SGD-SR (arbitrary sampling), SAGA, SVRG, L-SVRG, QSGD, TernGrad, DQGD, DIANA, **DIANAsr-DQ,** VR-DIANA, JacSketch, SEGA

bold font = new method

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

- Methods without error feedback: SGD, SGD-SR (arbitrary sampling), SAGA, SVRG, L-SVRG, QSGD, TernGrad, DQGD, DIANA, **DIANAsr-DQ,** VR-DIANA, JacSketch, SEGA
- Methods with delayed updates: D-SGD, **D-SGD-SR** (arbitrary sampling), **D-QSGD, D-SGD-DIANA, D-LSVRG, D-QLSVRG, D-LSVRG-DIANA**

bold font = new method

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

- Methods without error feedback: SGD, SGD-SR (arbitrary sampling), SAGA, SVRG, L-SVRG, QSGD, TernGrad, DQGD, DIANA, **DIANAsr-DQ,** VR-DIANA, JacSketch, SEGA

- Methods with delayed updates: D-SGD, **D-SGD-SR** (arbitrary sampling), **D-QSGD, D-SGD-DIANA, D-LSVRG, D-QLSVRG, D-LSVRG-DIANA**

- ✓ In one theorem, we recover the sharpest rates for all known special cases

- ✓ Our analysis works for non-strongly convex objectives as well

bold font = new method

4. Unified theory of Local-SGD

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. *Local SGD: Unified Theory and New Efficient Methods.* International Conference on Artificial Intelligence and Statistics. PMLR, 2021.

Local-SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \mod \tau \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma g_i^k\right), & \text{if } k+1 \mod \tau = 0 \end{cases}$$

Local First-Order Methods

- A lot of results are already known...

Local First-Order Methods

- A lot of results are already known...
- … but many fruitful directions were **unexplored**
  - better understanding of the local shifts
  - importance sampling
  - variance reduction
  - variable number of local steps
  - general theory for multiple data similarity types

The Third Problem

A single unifying theoretical framework for different variants of Local-SGD for heterogeneous/homogeneous problems is required

↓

The third contribution of the dissertation

Standard Assumptions

$f_1, f_2, \ldots, f_n$ – L-smooth and strongly quasi-convex

Standard Assumptions

$$\boxed{\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|}$$

$f_1, f_2, \ldots, f_n$ – L-smooth and strongly quasi-convex

Standard Assumptions

$$\boxed{\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|}$$

$$f_1, f_2, \ldots, f_n$$ – L-smooth and strongly quasi-convex

$$\boxed{f_i\left(x^*\right) \geq f_i(x) + \langle \nabla f_i(x), x^* - x\rangle + \frac{\mu}{2}\left\|x - x^*\right\|^2}$$

the solution of the problem

Key Assumption: “Unbiasedness“

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}\left[g_i^k \mid x_1^k,\ldots,x_n^k\right] = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i\left(x_i^k\right)$$

However, in general, $\mathbf{E}\left[g_i^k \mid x_1^k,\ldots,x_n^k\right] \neq \nabla f_i(x_i^k)$

needed to prevent clients’ drift via local shifts

Key Assumption: Bounded Second Moments

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}\left[\|g_i^k\|^2\right] \leq 2A\mathbf{E}\left[f\left(x^k\right)-f\left(x^*\right)\right]+B\mathbf{E}\left[\sigma_k^2\right]+F\mathbf{E}\left[V_k\right]+D_1$$

$$\mathbf{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}g_i^k\right\|^2\right] \leq 2A'\mathbf{E}\left[f\left(x^k\right)-f\left(x^*\right)\right]+B'\mathbf{E}\left[\sigma_k^2\right]+F'\mathbf{E}\left[V_k\right]+D_1'$$

Key Assumption: Bounded Second Moments

virtual iterates: $\boxed{x^k} = \frac{1}{n}\sum_{i=1}^{n} x_i^k$

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}\left[\|g_i^k\|^2\right] \leq 2A\mathbf{E}\left[f\left(\boxed{x^k}\right) - f\left(x^*\right)\right] + B\mathbf{E}\left[\sigma_k^2\right] + F\mathbf{E}\left[V_k\right] + D_1$$

$$\mathbf{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} g_i^k\right\|^2\right] \leq 2A'\mathbf{E}\left[f\left(\boxed{x^k}\right) - f\left(x^*\right)\right] + B'\mathbf{E}\left[\sigma_k^2\right] + F'\mathbf{E}\left[V_k\right] + D_1'$$

Key Assumption: Bounded Second Moments

virtual iterates: $\boxed{x^k} = \frac{1}{n}\sum_{i=1}^{n} x_i^k$

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{E}\left[\|g_i^k\|^2\right] \leq 2A\mathbf{E}\left[f\left(\boxed{x^k}\right) - f\left(x^*\right)\right] + B\mathbf{E}\left[\sigma_k^2\right] + F\mathbf{E}\left[\boxed{V_k}\right] + D_1$$

$$\mathbf{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} g_i^k\right\|^2\right] \leq 2A'\mathbf{E}\left[f\left(\boxed{x^k}\right) - f\left(x^*\right)\right] + B'\mathbf{E}\left[\sigma_k^2\right] + F'\mathbf{E}\left[\boxed{V_k}\right] + D_1'$$

$$\boxed{V_k} = \frac{1}{n}\sum_{i=1}^{n} \left\|x_i^k - \boxed{x^k}\right\|^2$$

workers' iterates discrepancy

Key Assumption: Shifts and Variance Reduction

$$\mathbf{E}\left[\sigma_{k+1}^2\right] \leq (1-\rho)\mathbf{E}\left[\sigma_k^2\right] + 2C\mathbf{E}\left[f\left(x^k\right) - f\left(x^*\right)\right] + G\mathbf{E}\left[V_k\right] + D_2$$

Key Assumption: Iterates Discrepancy

workers' iterates discrepancy

$$\boxed{V_k} = \frac{1}{n}\sum_{i=1}^{n}\left\|x_i^k - x^k\right\|^2$$

$$2L\sum_{k=0}^{K} w_k \mathbf{E}\left[\boxed{V_k}\right] \le \frac{1}{2}\sum_{k=0}^{K} w_k \mathbf{E}\left[f\left(x^k\right) - f\left(x^*\right)\right] + 2LH\mathbf{E}\sigma_0^2 + 2LD_3\gamma^2 W_K$$

Main Theorem: Simplified Version

depends only on the starting point and stepsize

$$\mathbf{E}\left[f(\bar{x}^K)\right] - f(x^*) \leq \left(1 - \min\left\{\gamma\mu, \frac{\rho}{4}\right\}\right)^K \frac{\Phi^0(x^0, \gamma)}{\gamma} + \gamma\Psi^0(D_1', D_2, D_3)$$

Linear function

S-Local-SVRG: Update Rule

Finite-sum case: $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{with prob. } 1-p \\ \frac{1}{n}\sum_{i=1}^n \left(x_i^k - \gamma g_i^k\right), & \text{with prob. } p \end{cases}$$

S-Local-SVRG: Update Rule

Finite-sum case: $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{with prob. } 1-p \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma g_i^k\right), & \text{with prob. } p \end{cases}$$

$$g_i^k = \nabla f_{i,j_i}\left(x_i^k\right) - \nabla f_{i,j_i}\left(y^k\right) + \nabla f\left(y^k\right)$$

S-Local-SVRG: Update Rule

Finite-sum case: $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{with prob. } 1-p \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma g_i^k\right), & \text{with prob. } p \end{cases}$$

$$g_i^k = \nabla f_{i,\boxed{j_i}}\left(x_i^k\right) - \nabla f_{i,\boxed{j_i}}\left(y^k\right) + \nabla f\left(y^k\right) \qquad \boxed{j_i} \sim \{1,\ldots,m\}$$

uniformly at random

S-Local-SVRG: Update Rule

Finite-sum case: $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{with prob. } 1-p \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma g_i^k\right), & \text{with prob. } p \end{cases}$$

$$g_i^k = \nabla f_{i,\boxed{j_i}}(x_i^k) - \nabla f_{i,\boxed{j_i}}\left(\boxed{y^k}\right) + \nabla f\left(\boxed{y^k}\right)$$

$$\boxed{j_i} \sim \{1,\ldots,m\}$$

uniformly at random

$$\boxed{y^{k+1}} = \begin{cases} x^k, & \text{with prob. } \boxed{q} \\ \boxed{y^k}, & \text{with prob. } 1-q \end{cases}$$

$$\boxed{q} = \frac{1}{m}$$

S-Local-SVRG: Rate of Convergence

S-Local-SVRG finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

$$\mathcal{O}\left(\left(m + \frac{L}{p\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{(1-p)L\max L_{ij}}}{p\mu}\right)\log\frac{1}{\varepsilon}\right)$$

iterations/oracle calls per node

S-Local-SVRG: Rate of Convergence

S-Local-SVRG finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \le \varepsilon$ after

$$\mathcal{O}\left(\left(m + \frac{L}{p\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{(1-p)L\max L_{ij}}}{p\mu}\right)\log\frac{1}{\varepsilon}\right)$$

iterations/oracle calls per node

The first linearly converging local method for heterogeneous data

Methods Covered by Our Framework

Method	a_i^k, b_i^k, l_i^k	Complexity	Setting	Sec
`Local-SGD` Alg. 27, [225]	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L\tau(\sigma^2+\tau\zeta^2)}{\mu^2\varepsilon}}$	UBV, ζ-Het	4.5.1
`Local-SGD` Alg. 27, [94]	$f_{\xi_i}(x_i^k), 0, -$	$\frac{\tau L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma^2+(\tau-1)\zeta_*^2)}{\mu^2\varepsilon}}$	UBV, Het	4.5.1
`Local-SGD` Alg. 27, [86]♣	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L+\mathcal{L}/n+\sqrt{(\tau-1)L\mathcal{L}}}{\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \frac{L\zeta^2(\tau-1)}{\mu^2\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma_*^2+\zeta_*^2)}{\mu^2\varepsilon}}$	ES, ζ-Het	4.5.1
`Local-SGD` Alg. 27, [86]♣	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L\tau+\mathcal{L}/n+\sqrt{(\tau-1)L\mathcal{L}}}{\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma_*^2+(\tau-1)\zeta_*^2)}{\mu^2\varepsilon}}$	ES, Het	4.5.1
`Local-SVRG` Alg. 28, (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y_i^k) + \nabla f_i(y_i^k)$, 0, $-$	$m + \frac{L+\max L_{ij}/n+\sqrt{(\tau-1)L\max L_{ij}}}{\mu} + \frac{L\zeta^2(\tau-1)}{\mu^2\varepsilon} + \sqrt{\frac{L(\tau-1)\zeta_*^2}{\mu^2\varepsilon}}$	simple, ζ-Het	4.5.2
`Local-SVRG` Alg. 28, (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y_i^k) + \nabla f_i(y_i^k)$, 0, $-$	$m + \frac{L\tau+\max L_{ij}/n+\sqrt{(\tau-1)L\max L_{ij}}}{\mu} + \sqrt{\frac{L(\tau-1)^2\zeta_*^2}{\mu^2\varepsilon}}$	simple, Het	4.5.2
`S*-Local-SGD` Alg. 29, (NEW)	$f_{\xi_i}(x_i^k), \nabla f_i(x^*), -$	$\frac{\tau L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)\sigma^2}{\mu^2\varepsilon}}$	UBV, Het	4.5.3
`SS-Local-SGD` Alg. 30, [83]	$f_{\xi_i}(x_i^k), h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k, \nabla f_{\tilde{\xi}_i^k}(y_i^k)$	$\frac{L}{p\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(1-p)\sigma^2}{p\mu^2\varepsilon}}$	UBV, Het	4.5.4
`SS-Local-SGD` Alg. 30, (NEW)	$f_{\xi_i}(x_i^k), h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k, \nabla f_{\tilde{\xi}_i^k}(y_i^k)$	$\frac{L}{p\mu} + \frac{\mathcal{L}}{n\mu} + \frac{\sqrt{L\mathcal{L}(1-p)}}{p\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \sqrt{\frac{L(1-p)\sigma_*^2}{p\mu^2\varepsilon}}$	ES, Het	4.5.4
`S*-Local-SGD*` Alg. 31, (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(x^*) + \nabla f_i(x^*)$, $\nabla f_i(x^*)$, $-$	$\left(\frac{\tau L}{\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{(\tau-1)L\max L_{ij}}}{\mu}\right)\log\frac{1}{\varepsilon}$	simple, Het	4.5.5
`S-Local-SVRG` Alg. 32, (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y^k) + \nabla f_i(y^k)$, $h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k, \nabla f_i(y^k)$	$\left(m + \frac{L}{p\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{L\max L_{ij}(1-p)}}{p\mu}\right)\log\frac{1}{\varepsilon}$	simple, Het	4.5.6

Our framework covers even methods without local updates

5. Faster Distributed Methods with Compression for Non-Convex Optimization

Eduard Gorbunov, Konstantin P. Burlachenko, Zhize Li, Peter Richtarik. *MARINA: Faster Non-Convex Distributed Learning with Compression*, Proceedings of the 38th International Conference on Machine Learning, PMLR 139:3788-3798, 2021.

Unbiased compression (quantization)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega \|x\|^2$$

Example: RandK (for K = $\boxed{2}$)

$$\boxed{d = 5} \left\{ \begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \right. \Rightarrow \frac{\boxed{5}}{\boxed{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix} \qquad \omega = \frac{\boxed{d}}{\boxed{K}} - 1$$

for unbiasedness

Pick K = $\boxed{2}$ components uniformly at random

Known Results for Non-Convex Problems

The best-known complexity results in the non-convex case

$$\sim \omega^{3/2}$$

Known Results for Non-Convex Problems

The best-known complexity results in the non-convex case

$$\sim \omega^{3/2}$$

For Rand*1*

$$\sim d^{3/2}$$

The Fourth Problem

New distributed methods with compression with better convergence guarantees are needed for distributed non-convex optimization

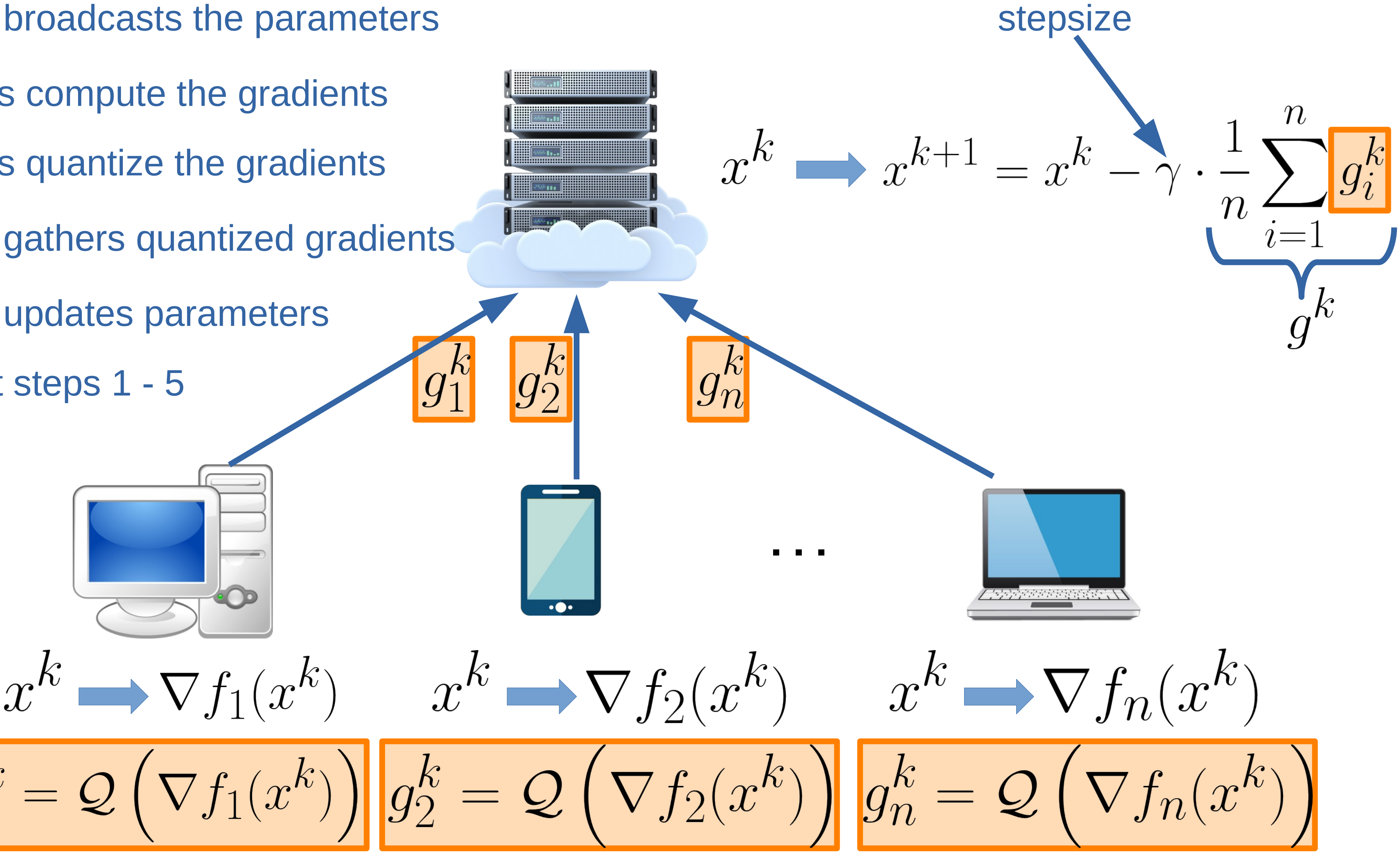
↓

The fourth contribution of the dissertation

Quantized Gradient Descent (QGD)

Alistarh, Dan, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. **"QSGD: Communication-efficient SGD via gradient quantization and encoding."** *In Advances in Neural Information Processing Systems*, pp. 1709-1720. 2017.

1. Server broadcasts the parameters
2. Devices compute the gradients
3. Devices quantize the gradients
4. Server gathers quantized gradients
5. Server updates parameters
6. Repeat steps 1 - 5



Assumptions

1. Uniform lower bound: $$\exists f_* \in \mathbb{R} : \forall x \in \mathbb{R}^d \quad f(x) \geq f_*$$

2. Smoothness: $$||\nabla f_i(x) - \nabla f_i(y)|| \leq L_i ||x - y||$$

Complexity Bound for QGD

Khaled, Ahmed, and Peter Richtárik. **"Better theory for SGD in the nonconvex world."** arXiv preprint arXiv:2002.03329 (2020).

QGD finds such $\hat{x}$ that $\mathbb{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \varepsilon^2$ after

Complexity Bound for QGD

Khaled, Ahmed, and Peter Richtárik. **"Better theory for SGD in the nonconvex world."** arXiv preprint arXiv:2002.03329 (2020).

QGD finds such $\hat{x}$ that $\mathbb{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \varepsilon^2$ after

$$\mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2} + \frac{(1+\omega)\Delta_0^2}{\varepsilon^4 n} + \frac{(1+\omega)\Delta_0\Delta_f^*}{\varepsilon^4 n}\right)$$

communication rounds

Complexity Bound for QGD

Khaled, Ahmed, and Peter Richtárik. **"Better theory for SGD in the nonconvex world."** arXiv preprint arXiv:2002.03329 (2020).

QGD finds such $\hat{x}$ that $\mathbb{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \boxed{\varepsilon^2}$ after

Hides numerical factors and smoothness constants

$$\mathcal{O}\left(\frac{\boxed{\Delta_0}}{\boxed{\varepsilon^2}} + \frac{(1+\boxed{\omega})\boxed{\Delta_0^2}}{\boxed{\varepsilon^4}n} + \frac{(1+\boxed{\omega})\boxed{\Delta_0}\boxed{\Delta_f^*}}{\boxed{\varepsilon^4}n}\right)$$

communication rounds

$$\boxed{\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega\|x\|^2}$$

$$\boxed{\Delta_0 = f(x^0) - f_*}$$

$$\boxed{\Delta_f^* = f_* - \frac{1}{n}\sum_{i=1}^n f_{i,*}}$$

Complexity Bound for QGD

Khaled, Ahmed, and Peter Richtárik. **"Better theory for SGD in the nonconvex world."** arXiv preprint arXiv:2002.03329 (2020).

QGD finds such $\hat{x}$ that $\mathbb{E}\left[\|\nabla f(\hat{x})\|^2\right] \leq \boxed{\varepsilon^2}$ after

Hides numerical factors and smoothness constants

$$\mathcal{O}\left(\frac{\boxed{\Delta_0}}{\boxed{\varepsilon^2}} + \frac{(1+\boxed{\omega})\boxed{\Delta_0^2}}{\boxed{\varepsilon^4} n} + \frac{(1+\boxed{\omega})\boxed{\Delta_0}\boxed{\Delta_f^*}}{\boxed{\varepsilon^4} n}\right)$$

communication rounds

$$\mathbb{E}\|Q(x)-x\|^2 \leq \omega\|x\|^2$$

$$\Delta_0 = f(x^0) - f_*$$

Not optimal!

$$\Delta_f^* = f_* - \frac{1}{n}\sum_{i=1}^{n} f_{i,*}$$

DIANA

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

Horváth, Samuel, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. "Stochastic distributed learning with gradient quantization and variance reduction." arXiv preprint arXiv:1904.05115 (2019).

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k$$

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k$$

QGD: $$g_i^k = \mathcal{Q}\left(\nabla f_i(x^k)\right)$$

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k$$

QGD: $$g_i^k = \mathcal{Q}\left(\nabla f_i(x^k)\right)$$

DIANA: $$g_i^k = \boxed{h_i^k} + \mathcal{Q}\left(\nabla f_i(x^k) - \boxed{h_i^k}\right)$$

learnable local shifts

$$\boxed{h_i^{k+1} = h_i^k + \alpha \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)}$$

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} g_i^k$$

QGD: $$g_i^k = \boxed{\mathcal{Q}\left(\nabla f_i(x^k)\right)}$$

vectors that devices have to send

DIANA: $$g_i^k = \boxed{h_i^k} + \boxed{\mathcal{Q}\left(\nabla f_i(x^k) - \boxed{h_i^k}\right)}$$

learnable local shifts

$$\boxed{h_i^{k+1} = h_i^k + \alpha \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)}$$

Complexity Bounds for DIANA and QGD

QGD: $$\mathcal{O}\left(\frac{\Delta_0}{\boxed{\varepsilon^2}} + \frac{(1+\boxed{\omega})\Delta_0^2}{\boxed{\varepsilon^4}n} + \frac{(1+\boxed{\omega})\Delta_0\Delta_f^*}{\boxed{\varepsilon^4}n}\right)$$

DIANA: $$\mathcal{O}\left(\frac{\Delta_0\left(1+(1+\boxed{\omega})\sqrt{\boxed{\omega}/n}\right)}{\boxed{\varepsilon^2}}\right)$$

Complexity Bound for DIANA

QGD: $\mathcal{O}\left(\frac{\Delta_0}{\varepsilon^2} + \frac{(1+\omega)\Delta_0^2}{\varepsilon^4 n} + \frac{(1+\omega)\Delta_0\Delta_f^*}{\varepsilon^4 n}\right)$

Is it possible to get better rates?

DIANA: $\mathcal{O}\left(\frac{\Delta_0\left(1+(1+\omega)\sqrt{\omega/n}\right)}{\varepsilon^2}\right)$

New Method: MARINA

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k$$

DIANA: $g_i^k = h_i^k + \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)$

$$h_i^{k+1} = h_i^k + \alpha \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)$$

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} g_i^k$$

DIANA: $g_i^k = h_i^k + \mathcal{Q}\left(\nabla f_i(x^k) - \boxed{h_i^k}\right)$

$$h_i^{k+1} = h_i^k + \alpha \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)$$

typically small

MARINA: $g_i^k = \begin{cases} \nabla f_i\left(x^k\right) & \text{w.p. } p \\ g^{k-1} + \mathcal{Q}\left(\nabla f_i\left(x^k\right) - \boxed{\nabla f_i\left(x^{k-1}\right)}\right) & \text{w.p. } 1-p \end{cases}$

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} g_i^k$$

DIANA: $g_i^k = h_i^k + \boxed{\mathcal{Q}\left(\nabla f_i(x^k) - \boxed{h_i^k}\right)}$

vectors that devices have to send

$$h_i^{k+1} = h_i^k + \alpha \mathcal{Q}\left(\nabla f_i(x^k) - h_i^k\right)$$

typically small

MARINA: $g_i^k = \begin{cases} \boxed{\nabla f_i\left(x^k\right)} & \text{w.p. } p \\ g^{k-1} + \boxed{\mathcal{Q}\left(\nabla f_i\left(x^k\right) - \boxed{\nabla f_i\left(x^{k-1}\right)}\right)} & \text{w.p. } 1-p \end{cases}$

Complexity Bounds for MARINA and DIANA

DIANA: $$\mathcal{O}\left(\frac{\Delta_0\left(1+(1+\boxed{\omega})\sqrt{\boxed{\omega}/n}\right)}{\varepsilon^2}\right)$$

MARINA: $$\mathcal{O}\left(\frac{\Delta_0\left(1+\boxed{\omega}/\sqrt{n}\right)}{\varepsilon^2}\right)$$

The Dissertation Also Contains

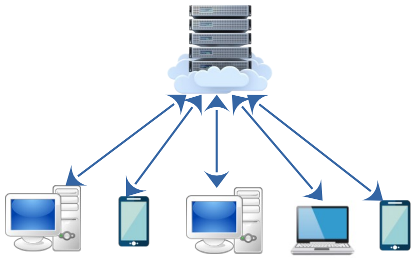
- Variance Reduced MARINA (uses stochastic gradients instead of full gradients)
- MARINA with partial participation of clients
- Rates under Polyak- Lojasiewicz Condition
- Explicit dependencies on smoothness constants, non-uniform sampling
- Numerical experiments with generalized linear models and neural networks

6. Decentralized Fault-Tolerant Optimization

Max Ryabinin, **Eduard Gorbunov**, Vsevolod Plokhotnyuk, and Gennady Pekhimenko. *Moshpit SGD: Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices*, **accepted to NeurIPS 2021**.

Communication

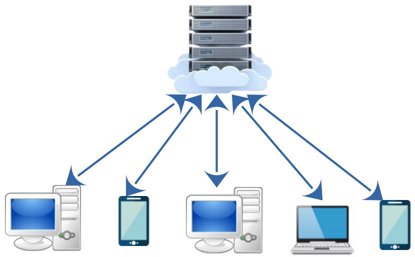
- With Parameter-Server (PS):
  - ✓ Simple and widely applicable approach
  - ✗ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

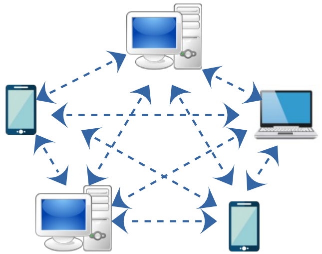
Communication

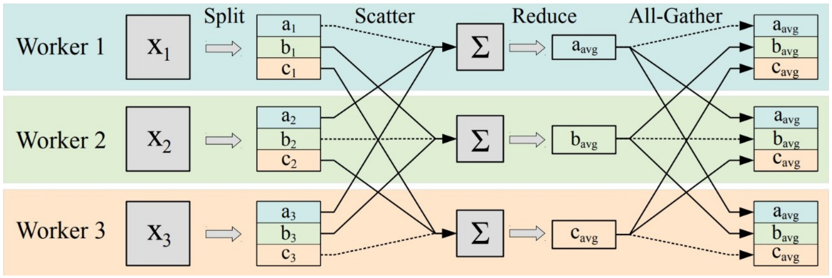
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

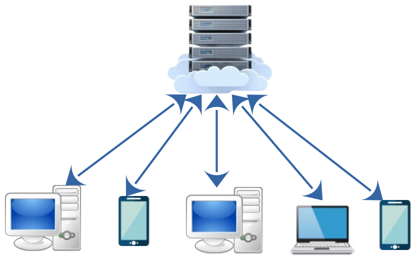
- Without PS via All-Reduce:





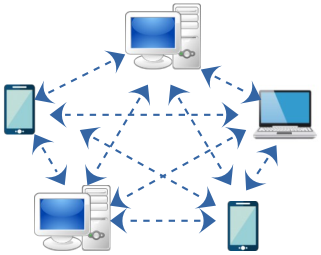
Communication

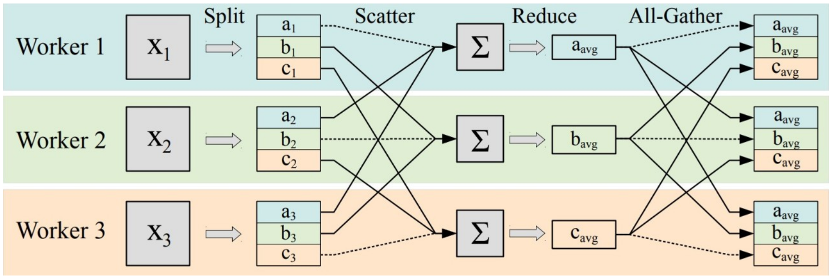
- With Parameter-Server (PS):
  - ✔ Simple and widely applicable approach
  - ✘ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

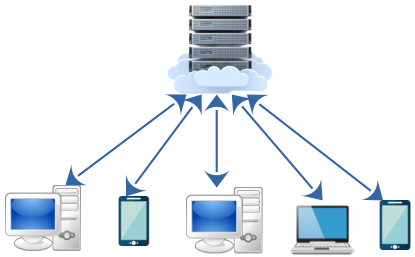
- Without PS via All-Reduce:
  - ✔ Scalable approach
  - ✘ Not robust to faults





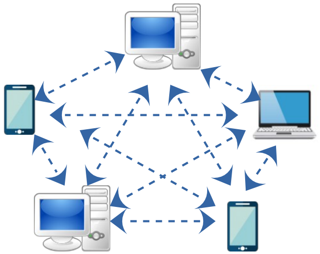
Communication

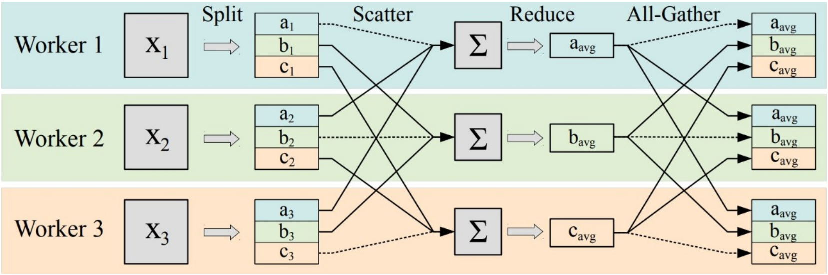
- With Parameter-Server (PS):
  - ✔ Simple and widely applicable approach
  - ✘ Not scalable: for large number of participants the communication is a bottleneck



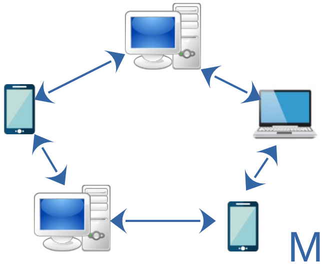
Devices send and receive full vectors

- Without PS via All-Reduce:
  - ✔ Scalable approach
  - ✘ Not robust to faults





- Without PS via gossip:
  - ✔ Scalable approach
  - ✘ Inevitable dependence on mixing matrix and graph structure



$$g_i^{k+1} = \sum_{j=1}^{n} \mathbf{M}_{ij} g_j^k$$

Devices send and receive full vectors

Mixing matrix defines the communication pattern

The Fifth Problem

New scalable decentralized fault-tolerant algorithm with better convergence guarantees than for gossip-based methods is required

↓

The fifth contribution of the dissertation

Moshpit All-Reduce: Main Idea

- All-Reduce protocols are fragile: the fault of 1 worker affects all other workers

Moshpit All-Reduce: Main Idea

- All-Reduce protocols are fragile: <u>the fault of 1 worker affects all other workers</u>
- The idea: execute <u>All-Reduce in small groups</u>

Moshpit All-Reduce: Main Idea

- All-Reduce protocols are fragile: the fault of 1 worker affects all other workers
- The idea: execute All-Reduce in small groups

The fault of one peer affects only its group

Moshpit All-Reduce: General Case

Algorithm 37 `Moshpit All-Reduce` (for i-th peer)

Input: parameters $\{x_j\}_{j=1}^n$, number of peers n, N, M, number of iterations T, peer index i

$x_i^0 := x_i$

$C_i^0 :=$ `get_initial_index(i)`

for $t \in 1 \dots T$ **do**

DHT$[C_i^{t-1}, t]$.add(address$_i$)

`/* wait for peers to assemble */`

`peers`$_t :=$ DHT.get$([C_i^{t-1}, t])$

$x_i^t, c_i^t :=$ `AllReduce`$(x_i^{t-1}, $ `peers`$_t)$

$C_i^t := (C_i^{t-1}$`[1:]`$, c_i^t)$ // same as eq. (1)

end for

Return x_i^T

$$\text{get\_initial\_index}\ (i) = \left(\left\lfloor i/M^{N-1} \right\rfloor \mod M\right)_{j \in \{1,\dots,N\}}$$

$$C_i^t := \left(c_i^{t-N+1}, c_i^{t-N+2}, \dots, c_i^t\right)$$

Moshpit All-Reduce: Theoretical Properties

- If n = M^N and there are no faults, then Moshpit All-Reduce finds an exact average after *N* steps

- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers_t is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \boxed{\frac{1}{n}\sum_i \theta_i^0} \right\|_2^2 \xrightarrow[t\to\infty]{} 0$$

- **Exponential convergence to the average:** for a version of Moshpit All-Reduce with random splitting into *r* groups at each step, we have

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left\|\theta_i^T - \boxed{\bar{\theta}}\right\|^2\right] = \left(\frac{r-1}{n} + \frac{r}{n^2}\right)^T \frac{1}{n}\sum_{i=1}^{n}\left\|\theta_i - \boxed{\bar{\theta}}\right\|^2$$

Moshpit SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \bmod \tau \neq 0 \\ \texttt{Moshpit All-Reduce}_{j \in P_{k+1}} (x_j - \gamma g_j^k), & \text{if } k+1 \bmod \tau = 0 \end{cases}$$

Number of active workers
at iteration *k+1*

Local-SGD with Moshpit All-Reduce instead of averaging

Assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_n(x) = f(x)$$

- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \le \sigma^2$$

- Effect of peers' vanishing is bounded: $$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \boxed{\widehat{x}^{k+1}}, \boxed{x^{k+1}} + \boxed{\widehat{x}^{k+1}} - 2x^* \right\rangle\right] \le \Delta_{pv}^k$$

$$\boxed{n_k = |P_k|}$$

$$\boxed{x^{k+1} = \frac{1}{\boxed{n_{k+1}}} \sum_{i \in P_{k+1}} x_i^{k+1}}$$

$$\boxed{\widehat{x}^{k+1} = \frac{1}{n_k} \sum_{i \in P_k} \left(x_i^k - \gamma g_i^k\right)}$$

Assumptions

- Function *f* is (strongly) convex
- Averaging quality:

$$\mathbb{E}\left[\frac{1}{n_{a\tau}}\sum_{i\in P_{a\tau}}\left\|x_i^{a\tau}-x^{a\tau}\right\|^2\right]\leq\gamma^2\delta_{aq}^2$$

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

$$\widetilde{\mathcal{O}}\left(\frac{L}{(1-\delta_{pv,1})\,\mu} + \frac{\delta_{pv,2}^2 + \sigma^2/n_{\min}}{(1-\delta_{pv,1})\,\mu\varepsilon} + \sqrt{\frac{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}{(1-\delta_{pv,1})^2\,\mu^2\varepsilon}}\right)$$

iterations when $\mu > 0$

$$\mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{R_0^2\left(\delta_{pv,2}^2 + \sigma^2/n_{\min}\right)}{\varepsilon^2} + \frac{R_0^2\sqrt{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}}{\varepsilon^{3/2}}\right)$$

iterations when $\mu = 0$

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

$$\widetilde{\mathcal{O}}\left(\frac{L}{(1-\delta_{pv,1})\,\mu} + \frac{\delta_{pv,2}^2 + \sigma^2/n_{\min}}{(1-\delta_{pv,1})\,\mu\varepsilon} + \sqrt{\frac{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}{(1-\delta_{pv,1})^2\,\mu^2\varepsilon}}\right)$$

iterations when $\mu > 0$

$$\mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{R_0^2\left(\delta_{pv,2}^2 + \sigma^2/n_{\min}\right)}{\varepsilon^2} + \frac{R_0^2\sqrt{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}}{\varepsilon^{3/2}}\right)$$

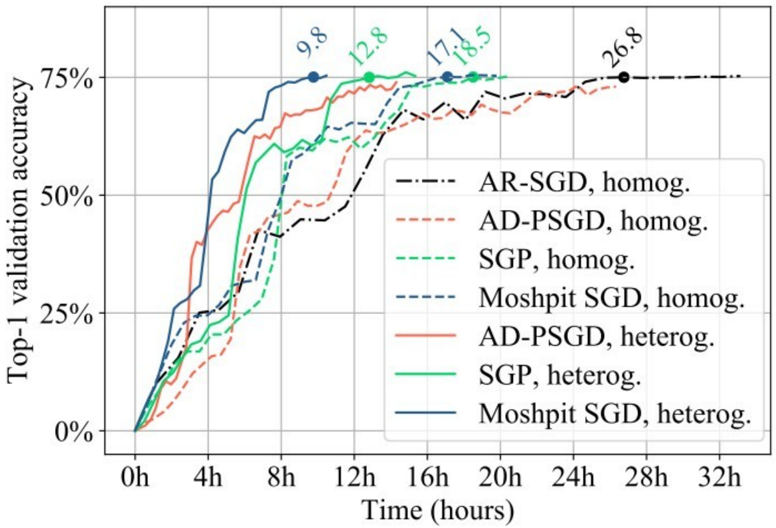
iterations when $\mu = 0$

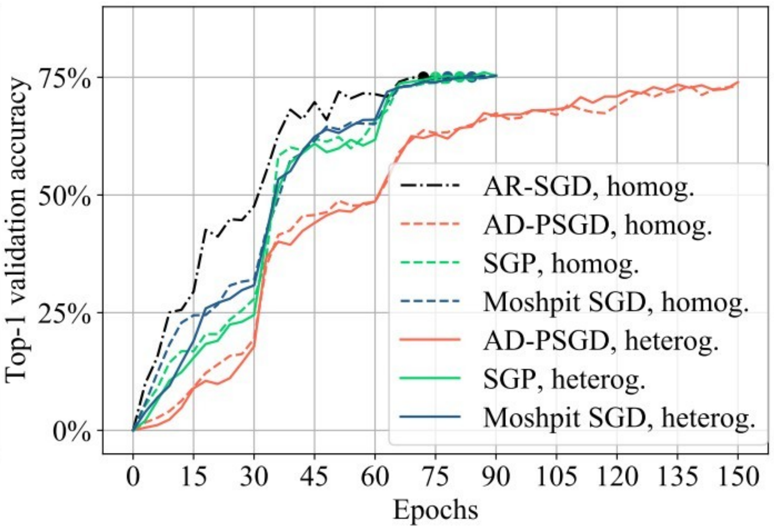
If $\delta_{pv,1} \leq 1/2,\ n_{\min} = \Omega(n),\ \delta_{pv,2}^2 = \mathcal{O}\left(\sigma^2/n_{\min}\right),\ \delta_{aq}^2 = \mathcal{O}\left((\tau-1)\sigma^2\right)$, then

the complexity of Moshpit SGD matches the complexity of <u>centralized</u> Local-SGD

Moshpit SGD: ResNet-50 on Imagenet

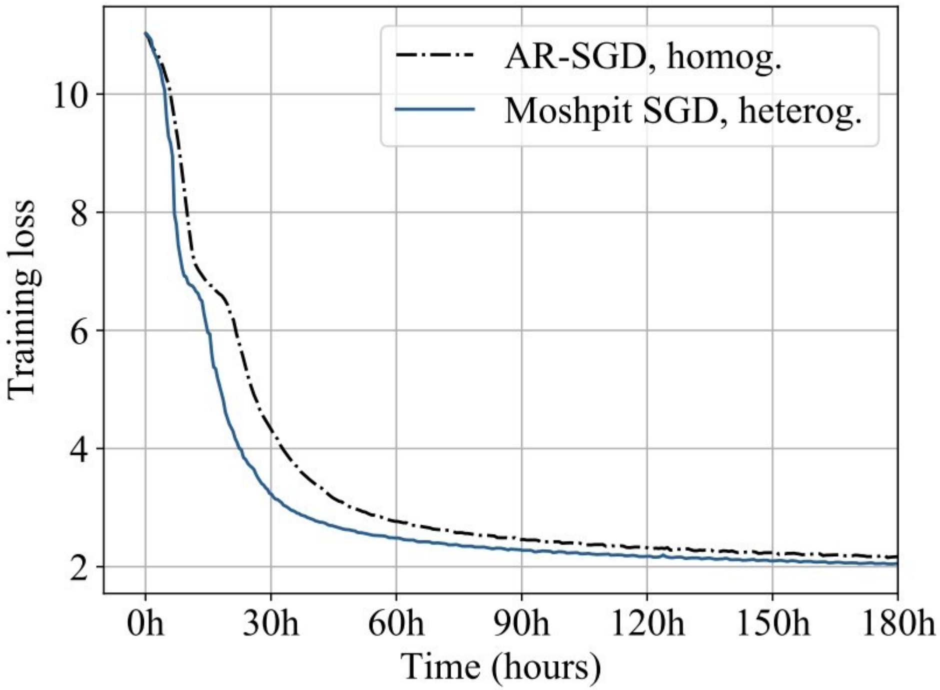
- We evaluate Moshpit SGD and several baselines in two environments
- (16 nodes with 1xV100 and 64 workers with 81 different GPUs)
- Comparable to All-Reduce in terms of iterations, faster in terms of time
- Decentralized methods run faster, but achieve worse results





Moshpit SGD: ALBERT on BookCorpus

- Baseline: All-Reduce on 8 V100
- Moshpit SGD: 66 preemptible GPUs
- Cost of spot instances are much smaller, yet we converge 1.5x faster



7. Conclusion

Short Summary of the Results

- Unified theory of SGD methods (5 new methods were proposed and analyzed)
- Unified theory of methods with error feedback and delayed updates (16 new methods were proposed and analyzed)
- Unified theory of Local-SGD methods (4 new methods were proposed and analyzed)
- Faster methods for non-convex distributed optimization with compression (3 new methods were proposed and analyzed)
- New efficient fault-tolerant method for decentralized optimization was proposed and analyzed
- New methods were tested numerically

I express my deepest gratitude to my supervisors Alexander Gasnikov and Peter Richtárik. I have learned a lot from both of you about various aspects of being a researcher. Thank you a lot for your guidance, encouragement, and opportunities that you provided. This all allowed me to realize my potential.

Next, I am grateful to Pavel Dvurechensky for all his help and guidance, especially during the work on our first papers.

I thank all my co-authors for their work, fruitful discussions and great impact on my research (in the order of appearance of joint works): Evgeniya Vorontsova, Dmitry Kovalev, Elnur Gasanov, Ahmed Mohammed, Elena Chernousova, Konstantin Mishchenko, Martin Takáč, El Houcine Bergou, Darina Dvinskikh, César A. Uribe, Filip Hanzely, Adel Bibi, Ozan Sener, Sergey Guz, Maksim Shirobokov, Egor Shulgin, Aleksandr Beznosikov, Marina Danilova, Dmitry Makarenko, Alexander Rogozin, Sergey Guminov, Dmitry Kamzolov, Innokentiy Shibaev, Konstantin Burlachenko, Zhize Li, Max Ryabinin, Vsevolod Plokhotnyuk, Gennady Pekhimenko, Alexander Borzunov, and Michael Diskin

I also express my gratitude to Artem Babenko, Francis Bach, Aymeric Dieuleveut, Ilyas Fatkhulin, Samuel Horváth, Praneeth Karimireddy, Eric Moulines, Anton Osokin, Alexander Panin, Liudmila Prokhorenkova, Igor Sokolov, Adrien Taylor, and Aleksei Ustimenko for fruitful discussions. Further, I owe a great thanks to my internship advisor Gauthier Gidel and to Nicolas Loizou, who I actively collaborated with during my internship. I learned a lot from you!

Finally, It is hard to express how much I appreciate all the support I received from MIPT and, in particular, from Andrei M. Raigorodskii and the Phystech School of Applied Mathematics and Informatics.