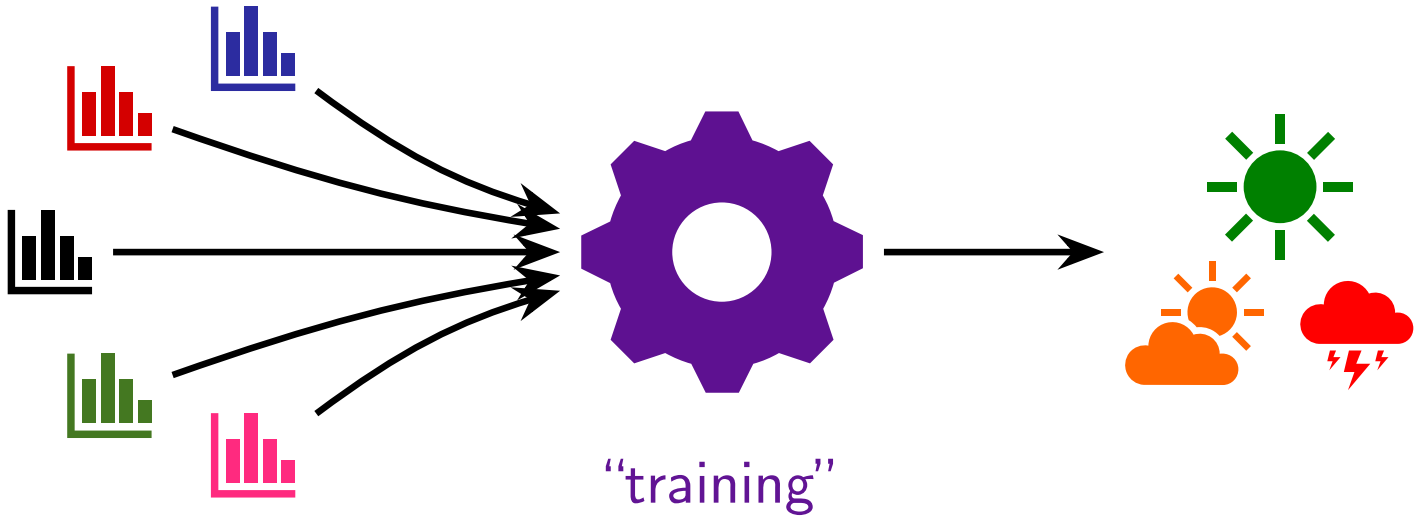
All Russian Seminar on Optimization

Decentralized Deep Learning on Heterogeneous Data

Sebastian U. Stich
www.sstich.ch

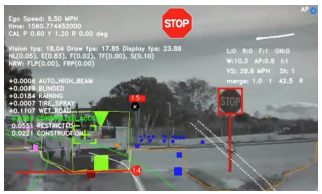
EPFL.ch → CISPA.de
postdoc & PhD positions available!
(or internship projects)

Learning From (Decentralized) Data

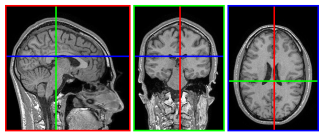


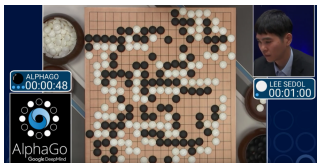
"training"

data | statistical model | task

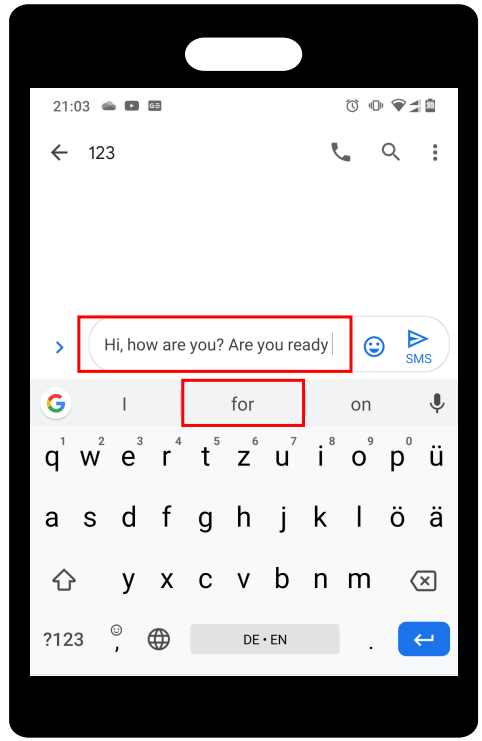


Bonjour $\mapsto$ Hej $\mapsto$ приветствие





Federated Learning



- private data stays on device
- server coordinates training and aggregates focused updates

[McMahan+ 16, FedAvg] **[Kairouz+ 19, Advances in FL]** ← hyperlinks!

Training Objective (in this talk)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} \underbrace{f_i(\mathbf{x})}_{\text{data } \mathcal{D}_i \text{ on client } i} \right] \qquad f_i(\mathbf{x}) = \mathbb{E}_{\xi\sim\mathcal{D}_i} f_i(\mathbf{x},\xi)$$

- Collaboratively solve **a (joint)** machine learning problem
- **efficiently**, in terms of:
  - computation (stochastic gradients, mini-batches),
  - communication (server $\leftrightarrow$ client).

Other very relevant scenarios:

• personalization • heterogenity • privacy • robustness

Base-Algorithm

Stochastic Gradient Descent:

$f(\mathbf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}} f(\mathbf{x}, \xi)$ *loss* function

$\xi \sim \mathcal{D}$ (unknown) data distribution

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

γ stepsize

$$\underbrace{\xi_t \sim \mathcal{D},}_{\text{uniform data sample, mini-batch}} \qquad \mathbf{x}_{t+1} := \underbrace{\mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t, \xi_t)}_{\text{model update}}$$

In practice:

- SGD with momentum, ADAM, AdaGrad
- or your favorite algorithm for single machine/server training **[Karimireddy+ 21, MIME]**

[Duchi+ 11, AdaGrad] [Kingma+ 14, ADAM] [Cutkosky+ 19, STORM]

Background I: SGD convergence ($n = 1$)

$$f(\mathbf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}} f(\mathbf{x}, \xi)$$

- **(Standard) Assumptions** $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$
  - $\mathbf{Var}\left[\nabla f(\mathbf{x}, \xi)\right] \leq \sigma^2$ bounded noise
  - $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$ L-smoothness

• Convergence	criterion	gradient computations
L-smooth	$\mathbb{E}\|\nabla f(\mathbf{x}_{\text{out}})\|^2 \leq \epsilon$	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$
+ μ-star convex[1]	$\mathbb{E} f(\mathbf{x}_{\text{out}}) - f(\mathbf{x}^\star) \leq \epsilon$	$\mathcal{O}\left(\frac{\sigma^2}{\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

- **Caveats:**
  - $\mathbf{x}_{\text{out}}$ is not the last iterate (typically a random iterate)
  - assumes *tuned* stepsize γ
  - assumptions might not hold in practice!

[Lan 11, Accelerated SGD] [Bottou+ 16, book] **[S 19, lecture notes]**

[1] $\exists \mathbf{x} \in \mathbb{R}^d$ s.t. $\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq \mu \|\mathbf{x} - \mathbf{x}^\star\|^2, \forall \mathbf{x} \in \mathbb{R}^d$

Background II: mini-batch SGD baseline

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} \left\{ f_i(\mathbf{x}) := \mathbb{E}_{\xi^i\sim\mathcal{D}_i}\left[f_i\left(\mathbf{x},\xi^i\right)\right] \right\} \right]$$

- **Mini-batch SGD**
  - compute (mini-batch) gradients on each client

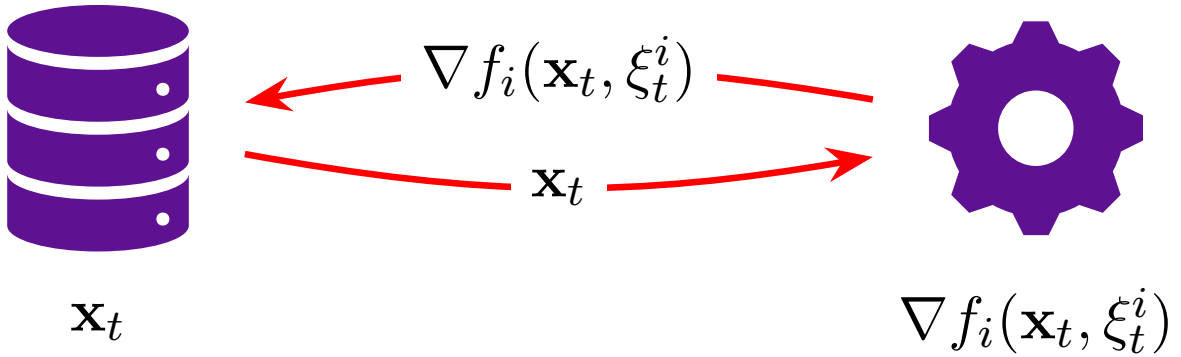
$$\xi_t^i \sim \mathcal{D}_i \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma}{n}\sum_{i=1}^{n} \nabla f_i\left(\mathbf{x}_t,\xi_t^i\right)$$

- **Convergence:**

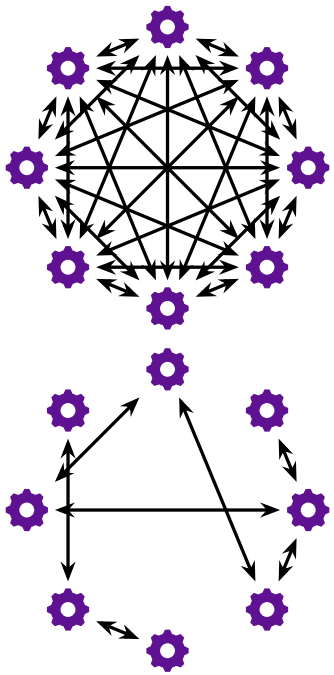
$$\mathcal{O}\Bigg(\underbrace{\frac{\sigma^2}{n\mu\epsilon}}_{\text{linear speedup}} + \frac{L}{\mu}\log\frac{1}{\epsilon} \Bigg)$$

[Dekel+ 10] [Goyal+ 17] **[Lin+ 20] [S+ 21]**

Training limited by communication bottleneck:



- bottleneck at the central server
- all-to-all communication is expensive
  $\Rightarrow$ arbitrary communication topology
- communication rounds too frequent
  $\Rightarrow$ less communication/local steps
- messages $\in \mathbb{R}^d$
  $\Rightarrow$ compression



Part 0 Intro & Motivation
mini-batch SGD refresher

Part I decentralized SGD on arbitrary networks
analysis, framework & limitations

Part II methods that work well for heterogeneous client data
SCAFFOLD, Gradient Tracking, RelaySGD

Decentralized SGD

A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi and S.U. Stich, A Unified Theory of Decentralized SGD with Changing Topology and Local Updates, ICML 2020.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} \underbrace{f_i(\mathbf{x})}_{\text{data } \mathcal{D}_i \text{ on client } i} \right] \qquad f_i(\mathbf{x}) = \mathbb{E}_{\xi^i\sim\mathcal{D}_i} f_i(\mathbf{x},\xi^i)$$

Decentralized SGD: model $\mathbf{x}_t^i$ on node $i \in [n]$

$$\xi_t^i \sim \mathcal{D}_i \qquad \mathbf{x}_{t+1}^i = \sum_{i=1}^{n} \left(\mathbf{x}_t^i - \gamma \nabla f_i\left(\mathbf{x}_t^i, \xi_t^i\right)\right) w_{ij}$$

matrix notation $\Leftrightarrow$

$$\mathbf{X}_{t+1} = \left(\mathbf{X}_t - \gamma \nabla \mathbf{F}(\mathbf{X}_t, \boldsymbol{\xi}_t)\right) \cdot \mathbf{W}$$

for mixing weights $\sum_i w_{ij} = 1$, $\sum_j w_{ij} = 1$, $w_{ij} = w_{ji}$

Example $\mathbf{W}$: "uniform average" with all neighbors

[Tsitsiklis, Bertsekas 86], [Xiao, Boyd 04], [Nedić, Ozdaglar 09], [Chen, Sayed 12], ...

Consensus with Linear Mixing

For the special case $\gamma = 0$, $\mathbf{X}_t \overset{t\to\infty}{\to} \bar{\mathbf{X}}$ where $\bar{\mathbf{X}} := \mathbf{X}_0 \cdot (\frac{1}{n}\mathbf{1})\mathbf{1}^\top$.

- $\bar{\mathbf{X}}_{t+1} = \bar{\mathbf{X}}_t = \cdots = \bar{\mathbf{X}}_0$
- $$\begin{aligned} \left\|\mathbf{X}_{t+1} - \bar{\mathbf{X}}\right\|_F^2 &= \left\|\mathbf{X}_t\mathbf{W} - \bar{\mathbf{X}}\right\|_F^2 \\ &= \left\|(\mathbf{X}_t - \bar{\mathbf{X}})(\mathbf{W} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\right\|_F^2 \\ &\leq \underbrace{\left\|\mathbf{W} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top\right\|^2}_{\text{'spectral gap'} \leq (1-p)} \left\|\mathbf{X}_t - \bar{\mathbf{X}}\right\|_F^2 \end{aligned}$$

Classic assumption: $\exists p > 0$:

$$\left\|\mathbf{X}\mathbf{W} - \bar{\mathbf{X}}\right\|_F^2 \leq (1-p)\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|_F^2 \qquad \forall \mathbf{X} \in \mathbb{R}^{d\times n}$$

Guarantees linear convergence of $\mathbf{X}_t \overset{t\to\infty}{\to} \bar{\mathbf{X}}$.

[Xiao, Boyd 04], [Lian+ 17]

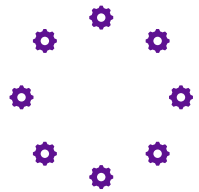
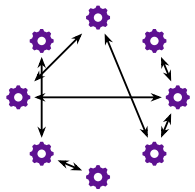
Expected Linear Mixing

Allow time-varying mixing matrix $\mathbf{W}_t$:

Our Assumption: $\exists\, \tau \geq 1$, $p > 0$, local steps, time-varying graphs

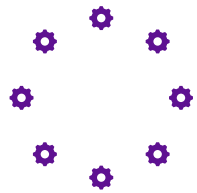
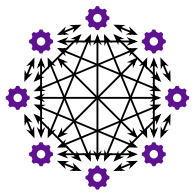
$$\mathbb{E}\left\|\mathbf{X}(\mathbf{W}_t \cdots \mathbf{W}_{t+\tau}) - \bar{\mathbf{X}}\right\|_F^2 \leq (1-p)\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|_F^2 \quad \forall \mathbf{X} \in \mathbb{R}^{d\times n}, t \geq 0$$

- $\tau = 1$: 'classic assumption **in expectation**'
  - random network typologies/mixing weights
  - no communication for (expected) $\frac{1}{p}$ steps

  with prob. $(1-p)$:  with prob. p: 
- $\tau > 1$: 'derandomization'
  - deterministic patterns
  - no communication for τ steps

  $\tau = \frac{1}{p}$

  Local Updates/Local SGD/Federated Learning

  for $(\tau - 1)$ steps  every τ-th step 

Coupling of SGD and Consensus

Technical step I:
descent lemma for the average SGD iterate $\bar{\mathbf{x}}_t = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_t^i$

$$\underbrace{\mathbb{E} f(\bar{\mathbf{x}}_{t+1}) \leq \mathbb{E} f(\bar{\mathbf{x}}_t) - \gamma \mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}_t)\right\|^2 + \gamma^2 \frac{L\sigma^2}{n}}_{\text{mini-batch SGD}} + \gamma \frac{L^2}{n} \mathbb{E} \underbrace{\left\|\mathbf{X}_t - \bar{\mathbf{X}}_t\right\|_F^2}_{\textbf{consensus error}}$$

Technical step II: control consensus error with mixing assumption

$$\mathbb{E}\left\|\mathbf{X}_t - \bar{\mathbf{X}}_t\right\|_F^2 \leq \gamma^2 \cdot \underbrace{\frac{\tau^2}{p^2}}_{\textbf{mixing}} \cdot \underbrace{\zeta^2}_{\textbf{dissimilarity}}$$

where $\zeta^2 = \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(\mathbf{x}^\star)\right\|^2$. dissimilarity measure

Convergence Results:

μ-strongly convex, L-smooth:

$$\mathcal{O}\left(\frac{\sigma^2}{\mu n \epsilon} + \frac{\sqrt{L}(\zeta\tau + \sigma\sqrt{p\tau})}{\mu p \sqrt{\epsilon}} + \frac{L\tau}{\mu p} \log \frac{1}{\epsilon} \right)$$

non-convex, L-smooth:

$$\mathcal{O}\left(\underbrace{\frac{\sigma^2}{n\epsilon^2}}_{\text{stochastic}} + \underbrace{\frac{\zeta\tau + \sigma\sqrt{\tau p}}{p\epsilon^{3/2}}}_{\text{'drift'}} + \underbrace{\frac{\tau}{p\epsilon}}_{\text{deterministic}} \right) \cdot L$$

Insights:

- optimal! [Woodworth+ 20b]
- does only depend on the ratio $\frac{\tau}{p}$ as expected
- if the stochastic term dominates:
  - linear speedup in n
  - network independent convergence guarantees!

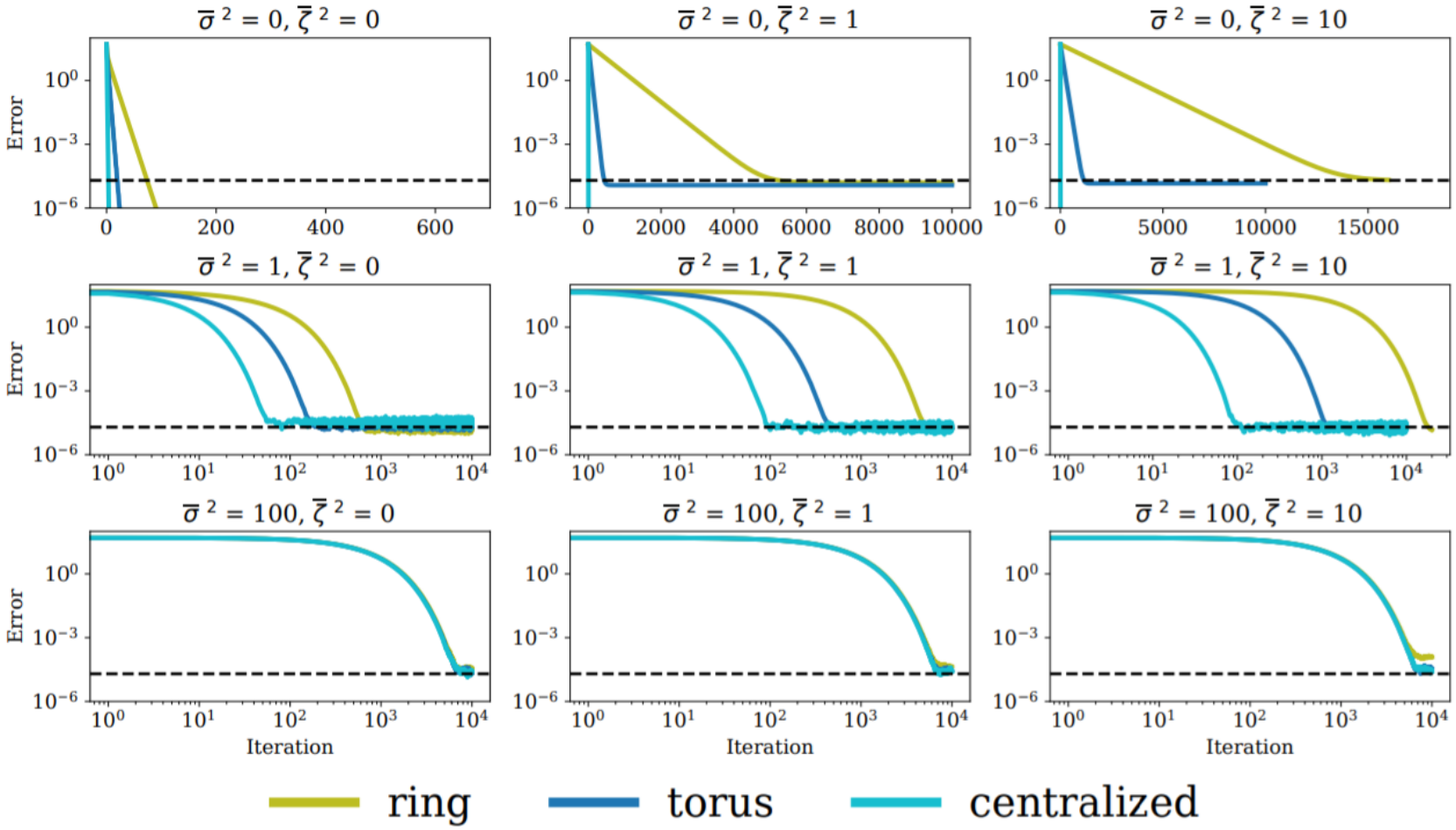
Impact of the drift ζ

- no impact if σ^2 dominates
- often noticeably impact in practice

Numerical Results

synthetic data on $n = 25$ nodes

$f_i(\mathbf{x}) = \left\|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i^\top\mathbf{x}\right\|^2$, $\mathbf{b}_i$ carefully chosen to control ζ



Part II:

methods agnostic to heterogeneous client data

The cause of drift

Stationary points, $\nabla f(\mathbf{x}^\star) = 0$, are not fixed points!

$$(\mathbf{X}^\star - \gamma \nabla F(\mathbf{X}^\star))\mathbf{W} = \mathbf{X}^\star - \gamma \underbrace{\nabla F(\mathbf{X}^\star)\mathbf{W}}_{\neq 0}$$

$\nabla f_i(\mathbf{x}^\star) \neq 0$ in general

Example: τ local steps

estimate of the consensus distance (intuitively):

$$\mathbb{E}\left\| \sum_{t=1}^{\tau} (\approx \gamma \nabla f_i(\mathbf{x}^\star)) \right\|^2 \approx \gamma^2 \tau^2 \mathbb{E} \left\| \nabla f_i(\mathbf{x}^\star) \right\|^2 = \gamma^2 \tau^2 \zeta^2$$

with $\zeta^2 = \frac{1}{n} \sum_{i=1}^n \left\| \nabla f_i(\mathbf{x}^\star) \right\|^2$.

Option I: Drift correction

if we would know $\nabla F(\mathbf{X}^\star)$, we could compensate:

$$\mathbf{X}_{t+1} = (\mathbf{X}_t - \nabla F(\mathbf{X}_t) + \mathbf{C})\mathbf{W}$$

for $\mathbf{C} = \nabla F(\mathbf{X}^\star)$.

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i - \gamma\Big(\underbrace{\nabla f_i(\mathbf{x}_t^i, \xi_t^i)}_{\text{normal update}} - \underbrace{\underbrace{\mathbf{c}_t^i}_{\substack{\mathbf{c}_t^i \approx \nabla f_i(\bar{\mathbf{x}}_t) \\ \text{local drift}}} + \underbrace{\mathbf{c}_t}_{\substack{\mathbf{c}_t \approx \nabla f(\bar{\mathbf{x}}_t) \\ \text{global drift}}}}_{\textbf{drift correction}} \Big)$$

- correction does not depend on local steps and is *unbiased!*

Similarities to variance reduction in server-only optimization, like in SVRG, SAGA, SCSG, etc.

[Johnson+ 13, SVRG] [Defazio+ 14, SAGA] [Lei+ 16, SCSG]

Implementation Sketch (with server): Bias Estimation

- if n small, SVRG/SAGA-type correction **[SCAFFOLD]**
  $\mathbf{c}_t^i = \mathbf{g}_t^i$, $\mathbf{c}_t = \frac{1}{n}\sum_{i=1}^n \mathbf{c}_t^i$
- if n is huge, SCSG-type correction **[Mime]**
  $\mathbf{c}_t^i = \mathbf{g}_t^i$, $\mathbf{c}_t = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t} \mathbf{c}_t^i$, for active clients $\mathcal{S}_t \subset [n]$

Here $\mathbf{g}_t^i$ denotes a (stochastic (possibly mini-batch) or full batch) gradient.

Theoretical Results with Bias Correction

algorithm	rounds
mini-batch SGD batch size τ	$\mathcal{O}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$
SCAFFOLD τ local steps + proper init	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

Implementation Sketch (without server): Bias Tracking

Use Gradient Tracking/NEXT to estimate the bias locally:

$$\mathbf{x}_t^i = \sum_{j=1}^n w_{ij}\left(\mathbf{x}_t^j - \mathbf{y}_t^j\right) \qquad \mathbf{y}_t^i = \sum_{j=1}^n w_{ij}\mathbf{y}_{t-1}^j + \mathbf{g}_t^i - \mathbf{g}_{t-1}^i$$

Here $\mathbf{g}_t^i$ denotes a (stochastic (possibly mini-batch) or full batch) gradient.

Theoretical Results with Gradient Tracking [KLS, 21]

algorithm	rounds
mini-batch SGD batch size τ	$\mathcal{O}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$
Gradient Tracking $p = \frac{1}{\tau}$ mixing + proper init	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{\sqrt{L\tau}\sigma}{\mu\sqrt{\epsilon}} + \frac{L\tau}{\mu}\log\frac{1}{\epsilon}\right)$

NB: We improve the last two terms by a factor of τ.

[Nedić+, DIGing, 16] [Lorenzo+, NEXT, 16] [Koloskova+, 21]

Option II: better mixing (small $\frac{\tau}{p}$)

R rounds of communication:

$$(\mathbf{X}^\star - \gamma \nabla F(\mathbf{X}^\star))\mathbf{W}^R = \mathbf{X}^\star - \gamma \underbrace{\nabla F(\mathbf{X}^\star)\mathbf{W}^R}_{\approx 0}$$

- Worst-case: $R = \tilde{\Omega}\left(\frac{\tau}{p}\right)$ rounds:

$$(1-p)^{\frac{c}{p}} \leq e^{-c}$$

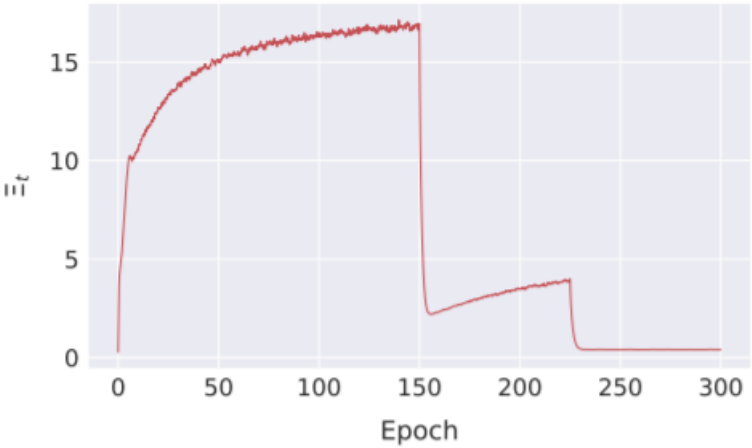
- Closer inspection: it suffices to pick R s.t.

$$\left\|\mathbf{X}\mathbf{W}^R - \bar{\mathbf{X}}\right\|_F^2 \leq \frac{\gamma\sigma^2}{Ln} + \frac{1}{L^2}\left\|\nabla f(\bar{\mathbf{x}})\right\|^2$$

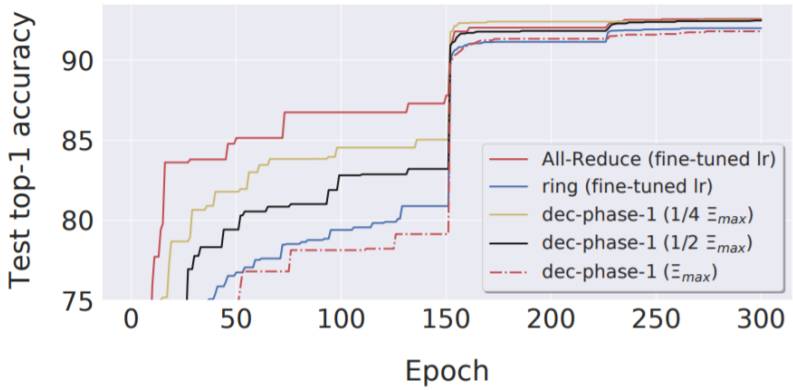
[Tsianos+ 16] [Scaman+, 17] [Kong+ 20]

Consensus Control

ResNet-20 on CIFAR-10, $n = 32$, ring topology



evolution of the consensus distance



test-accuracy

L. Kong, T. Lin, A. Koloskova, M. Jaggi and S.U. Stich, Consensus Control for Decentralized Deep Learning ICML 2021.

Option III: alternative mixing mechanisms

Gossip averaging: $\nabla F(\mathbf{X}^\star)\mathbf{W} \neq 0$.

$$\mathbf{x}_{t+1}^i = w_{ii}(\mathbf{x}_t^i - \gamma \nabla f_i(\mathbf{x}_t^i)) + \sum_{j \neq i} w_{ij}(\mathbf{x}_t^j - \gamma \nabla f_j(\mathbf{x}_t^j))$$

Idea:

Uniform (but delayed) averaging over all workers:

$$\mathbf{x}_{t+1}^i = \frac{1}{n} \sum_{j=1}^n \big(\mathbf{x}_{t-\tau_{ij}}^i - \gamma \nabla f_j(\mathbf{x}_{t-\tau_{ij}})\big)$$

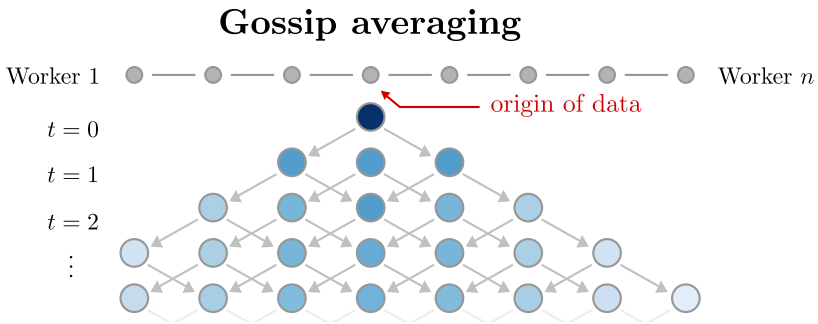
τ_{ij} the delay for routing the message from note j to i.

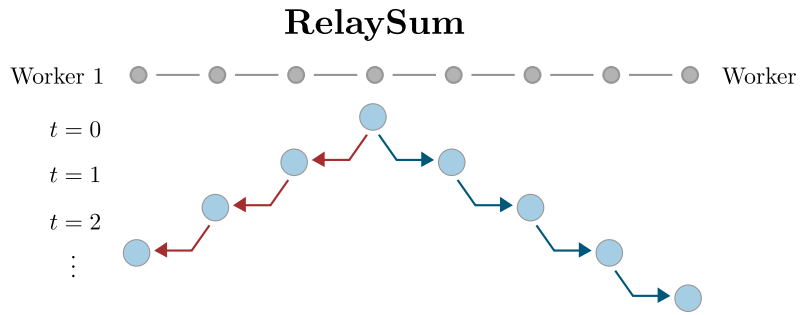
Sanity check: $\frac{1}{n} \sum_{j=1}^n \big(\mathbf{x}_\star^i - \gamma \nabla f_j(\mathbf{x}_\star)\big) = \mathbf{x}_\star^i \qquad (\mathbf{x}_{\star - \tau_{ij}} = \mathbf{x}_\star)$

Relay SGD

T. Vogels, L. He, A. Koloskova, T. Lin, P. Karimireddy, S.U. Stich and Martin Jaggi, RelaySum for Decentralized Deep Learning on Heterogeneous Data, NeurIPS, 2021.

Relay Sum

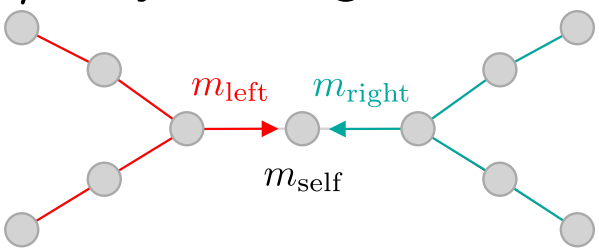


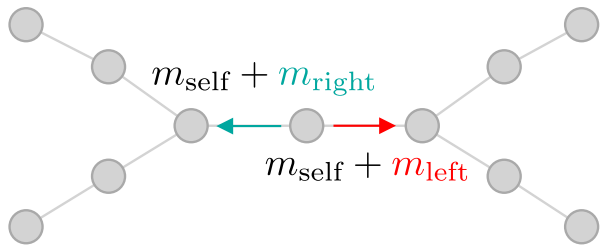


$$\mathbf{x}_{t+1}^i = \frac{1}{n}\sum_{j=1}^{n}\left(\mathbf{x}_{t-\tau_{ij}}^i - \gamma \nabla f_j\left(\mathbf{x}_{t-\tau_{ij}}\right)\right)$$

Efficient implementation:

- pick a *spanning tree* in the network
- route/relay messages:





[Ko 10] [Georgopoulos 11] [Hendrickx+ 14]

Proof?

Technical step I: Formulate as linear update on an augmented graph on $n \times (\tau_{\max} + 1)$ nodes, where $\tau_{\max} \geq \tau_{ij}$.

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t \mathbf{W} - \gamma \mathbf{G}_t \tilde{\mathbf{W}}$$

Technical step II: Consensus for the augmented mixing matrix $\mathbf{W} \in n(\tau_{\max} + 1) \times n(\tau_{\max} + 1)$.

- this matrix is not doubly stochastic!
- therefore $\mathbf{X}\mathbf{W}^t \overset{(t \to \infty)}{\not\to} \bar{\mathbf{X}}$!

Let $\boldsymbol{\pi}$ be a right eigenvector of $\mathbf{W}$ such that $\boldsymbol{\pi}^\top \mathbf{1} = 1$.

Lemma: $\exists m \geq 1$ and $p > 0$, such that:

$$\left\| \mathbf{X}\mathbf{W}^m - \mathbf{X}\boldsymbol{\pi}\mathbf{1}^\top \right\|^2 \leq (1-p)^m \left\| \mathbf{X} - \mathbf{X}\boldsymbol{\pi}\mathbf{1}^\top \right\|^2$$

Convergence Results:

μ-strongly convex, L-smooth:	$\mathcal{O}\left(\frac{\sigma^2}{\mu n \epsilon} + \frac{\sigma\sqrt{LC})}{\mu p \sqrt{\epsilon}} + \frac{LC}{\mu p} \log \frac{1}{\epsilon} \right)$
non-convex, L-smooth:	$\mathcal{O}\left(\underbrace{\frac{\sigma^2}{n\epsilon^2}}_{\text{stochastic}} + \frac{\sigma C}{p\epsilon^{3/2}} + \underbrace{\frac{C}{p\epsilon}}_{\text{deterministic}} \right) \cdot L$

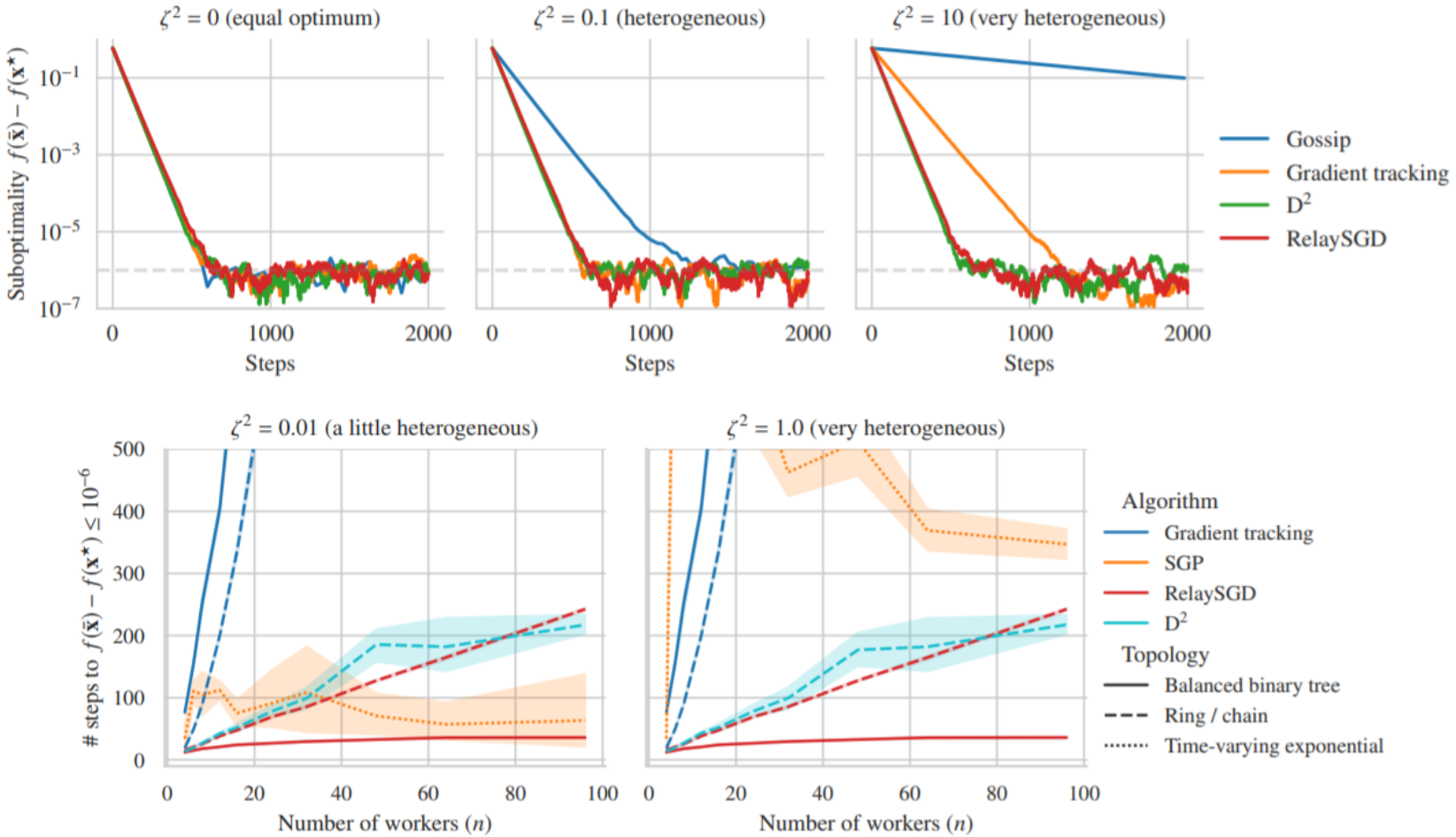
where $C \leq \tau_{\max}^{3/2}$.

no dependency on ζ

Numerical Results

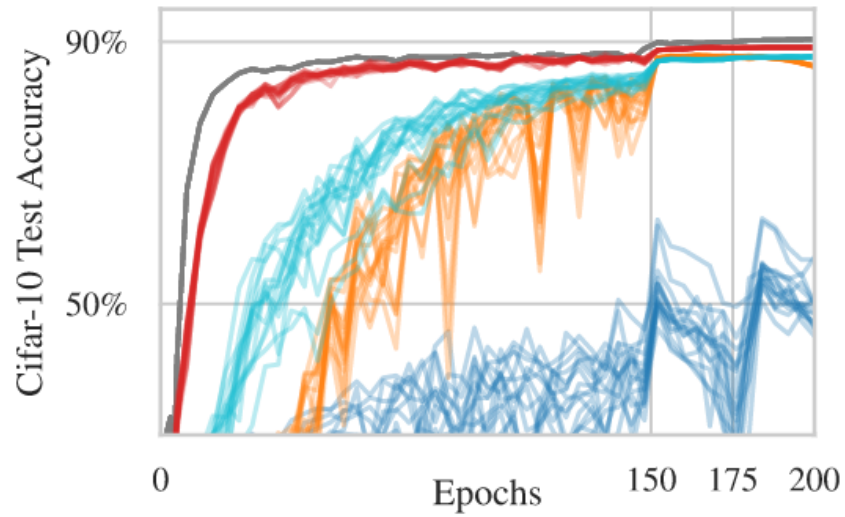
Quadratic Problem—Ring topology on $n = 32$ nodes

$f_i(\mathbf{x}) = \left\|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i^\top\mathbf{x}\right\|^2$, $\mathbf{b}_i$ carefully chosen to control ζ



VGG-11 on CIFAR-10, $n = 16$

(random data assignment with Dirichlet $\alpha = 0.01$—a parameter that controls data-heterogenity)



Heterogeneously distributed Cifar-10 with VGG-11

Topologies are optimized, but use fixed and sparse, with equal communication

All-to-all (baseline) DP-SGD D2 SGP RelaySGD

Algorithm	Ring	Chain (= spanning tree of ring)	Double binary trees	Time-varying exponential [2]
RelaySGD	Unsupported	Worse than double b. trees (E.1)	**Best result**	Unsupported
DP-SGD	**Best result**	Worse than ring	Worse than ring (E.1)	Unsupported
D^2	**Best result**	Worse than ring	Worse than ring (E.1)	Unsupported
SGP	Equivalent to DP-SGD	Equivalent to DP-SGD	Equivalent to DP-SGD	**Best result**

Conclusion

- framework to analyze decentralized SGD methods on time-varying networks
  - local SGD, pairwise gossip, ...
  - Gradient Tracking, bias correction methods, consensus control
  - single-stochastic mixing matrices
- relaySum as an alternative to gossip averaging

Main references and collaborators


A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi and S.U. Stich, A Unified Theory of Decentralized SGD with Changing Topology and Local Updates, ICML 2020.


L. Kong, T. Lin, A. Koloskova, M. Jaggi and S.U. Stich, Consensus Control for Decentralized Deep Learning ICML 2021.


P. Karimireddy, S. Kale, M. Mohri, S.J. Reddi, S.U. Stich and A.T. Suresh, SCAFFOLD: Stochastic Controlled Averaging for Federated Learning, ICML 2020.


A. Koloskova, T. Lin and S.U. Stich, An Improved Analysis of Gradient Tracking for Decentralized Machine Learning, NeurIPS, 2021.


T. Vogels, L. He, A. Koloskova, T. Lin, P. Karimireddy, S.U. Stich and Martin Jaggi, RelaySum for Decentralized Deep Learning on Heterogeneous Data, NeurIPS, 2021.