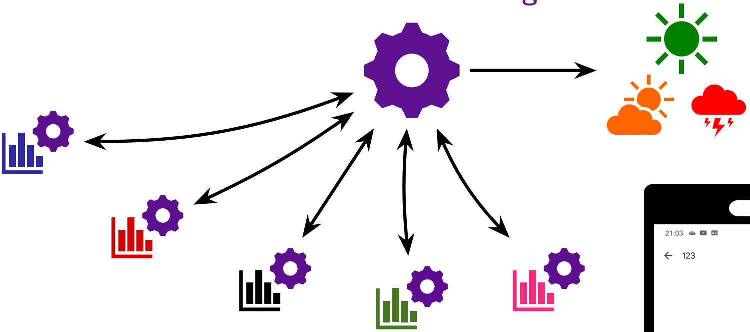
FL-ICML 2021 Workshop

Algorithms for Efficient Federated (and Decentralized) Learning

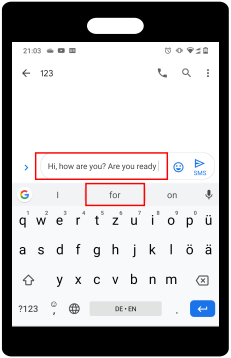
Sebastian U. Stich
www.sstich.ch

EPFL.ch → CISPA.de
postdoc & PhD positions available!

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)

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

 $+$  $+\cdots+$ 

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

Other very relevant scenarios: later talks today ;-)

• 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
- Adapt your favorite algorithm for single machine/sever training to the FL setting!

[Duchi+ 11, AdaGrad] [Kingma+ 14, ADAM] [Cutkosky+ 19, STORM]
[Karimireddy+ 20b, MIME]

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	$\mathbb{E}\left\|\nabla f(\mathbf{x}_{\text{out}})\right\|^2 \leq \epsilon$	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$
+ 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]**

[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

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

 $+$  $+\cdots+$ 

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

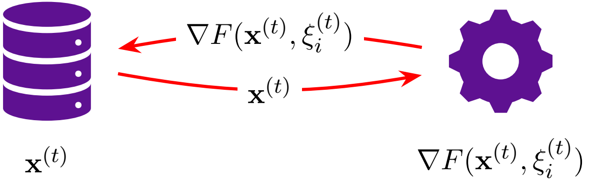
$$\xi_i^{(t)} \sim \mathcal{D}_i \qquad \mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \frac{\gamma}{n}\sum_{i=1}^{n}\nabla F\left(\mathbf{x}^{(t)}, \xi_i^{(t)}\right)$$

- **Convergence:**

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

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

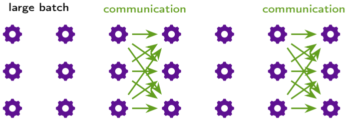
Training limited by communication bottleneck:



algorithm	rounds	gradients (total)
mini-batch SGD batch size $\tau = 1$	$\mathcal{O}\left(\frac{\sigma^2}{n\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\mu\epsilon} + \frac{nL}{\mu}\log\frac{1}{\epsilon}\right)$
mini-batch SGD batch size τ	$\mathcal{O}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\mu\epsilon} + \frac{n\tau L}{\mu}\log\frac{1}{\epsilon}\right)$
?	$\lll$	$\mathcal{O}\left(\frac{\sigma^2}{\mu\epsilon} + \frac{n\tau L}{\mu}\log\frac{1}{\epsilon}\right)$

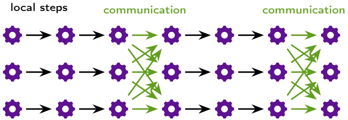
Local Update Methods

+ increasing batch size reduces communication
– but no progress while computing the batch gradients
- for $\tau \to \infty$, stuck at $\mathbf{x}_0$ forever!



Local Steps $$\left.\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \gamma \nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)})\right\} \times \tau \text{ times}$$

+ use local gradients for local model updates
- for $\tau \to \infty$, each client converges to local solution $\mathbf{x}_i^\star$
– different models $\mathbf{x}_i$ on clients!



Local SGD

$$\mathbf{x}_i^{(t+1)} := \begin{cases} \mathbf{x}_i^{(t)} - \gamma \nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) & \text{if } t+1 \notin \{0, \tau, 2\tau, \dots\} \\ \frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^{(t)} - \gamma \nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) \right) & \text{if } t+1 \in \{0, \tau, 2\tau, \dots\} \end{cases}$$

in general: $\mathbf{x}_i^{(t)} \neq \mathbf{x}_j^{(t)}$ for $i \neq j$ and $t+1 \notin \{0, \tau, 2\tau, 3\tau, \dots\}$

- leverages parallelism (unlike single-machine SGD)
- makes τ updates per round (unlike large-batch SGD)
- performs good in experiments

[Zinkevich+ 10] [McMahan+ 16] **[S 18]** [Wang+ 18] [Dieuleveut+ 19] [Haddadpour+ 19] [Woodworth+ 20a] [Khaled+ 20] **[Lin+ 20]** **[Koloskova+ 20]** [Woodworth+ 20b]

How to Analyze Local SGD?

Main idea: Study the virtual average

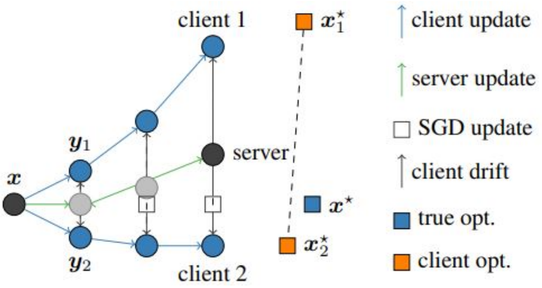
$$\bar{\mathbf{x}}^{(t+1)} := \bar{\mathbf{x}}^{(t)} - \frac{\gamma}{n} \sum_{i=1}^{n} \nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$$

- $\bar{\mathbf{x}}^{(t)}$ behaves almost as 'normal' SGD $\quad \mathbb{E}\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\|^2 \to 0$
- the additional error term $\mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2$ can be controlled
  - IID data $(\mathcal{D}_i = \mathcal{D}_j)$
    $\mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 = \gamma^2 \cdot \mathcal{O}\left(\tau^2 \left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tau}{n}\sigma^2\right)$
  - non-IID data $(\mathcal{D}_i \neq \mathcal{D}_j)$ additional properties
    $\mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 = \gamma^2 \cdot \mathcal{O}\left(\tau^2 \left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \tau^2\zeta^2 + \frac{\tau}{n}\sigma^2\right)$

$\mathbb{E}_i \left\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\right\|^2 \leq \zeta^2$ data-dissimilarity
inter-client variance

[S 19] [S+ 20] [Koloskova+ 20]

Data-dissimilarity $\zeta^2 > 0$ causes *drift* when doing local steps.



$$\mathbb{E}_i \left\| \nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x}) \right\|^2 \leq \zeta^2$$

Many **refinements/relaxations/alternatives** proposed and studied in the literature.

[Karimireddy+ 20a] [Koloskova+ 20] [Khaled+ 20] [Li+ 20] [Cen+ 20]

Theoretical Results on Local SGD

+ local SGD converges!

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

– (in the worst-case) slower than mini-batch SGD

– cannot be significantly improved [Woodworth+ 20b]

+ improved results under additional assumptions possible!

Mitigate Bias/Drift in Local Update Methods

Main Idea: Bias correction in local update

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

- 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: Estimate Bias

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

Here $\mathbf{g}_i^{(t)}$ 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	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

Benefits of Local Updates?

Intermezzo: Some Experiments

- drift correction accelerates training on non-IID data!

		non-IID data				IID data	
	Epochs	0% similarity (sorted) Num. of rounds	Speedup	10% similarity Num. of rounds	Speedup	100% similarity (i.i.d.) Num. of rounds	Speedup
SGD	1	317	(1×)	365	(1×)	416	(1×)
SCAFFOLD	1	77	(4.1×)	62	(5.9×)	60	(6.9×)
	5	152	(2.1×)	20	(18.2×)	10	(41.6×)
FedAvg	1	258	(1.2×)	74	(4.9×)	83	(5×)
	5	428	(0.7×)	34	(10.7×)	10	(41.6×)

less rounds with drift correction — no drift correction needed

Communication rounds to reach 0.5 test accuracy for logistic regression on EMNIST.

- local updates yield better generalization than huge batches!

	Top-1 acc.	local gradients	communication
Mini-batch SGD ($n = 16$, $\tau = 128$)	92.5%	2048	-
Mini-batch SGD ($n = 16$, $\tau = 1024$)	76.3%	16384	÷ 8
Local-SGD ($n = 16$, $\tau = 8 \times 128$)	92.0%	16384	÷ 8

ResNet-20 on CIFAR-10

Source: [Karimireddy+ 20a] [Lin+ 20]

Better theory?

- in the standard complexity classes, no benefit of local steps!
- however, on quadratic functions we hope to be better!
  - IID setting: (same Hessian, matching local minima)

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

  - non-IID setting: (same Hessian, different local minima)

SCAFFOLD τ local steps	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\tau\mu}\log\frac{1}{\epsilon}\right)$

Remarkable: benefit from parallelism and serial updates!

[Woodworth+ 20a] [Woodworth+ 20b] **[Karimireddy+ 20a]**

'Breaking' Lower Bounds!

- Additional Assumptions
  - Bounded third derivative, Hessian M-Lipschitz

$$\left\|\nabla^3 f(\mathbf{x})\right\| \leq \delta \quad \text{or} \quad \left\|\nabla^3 f(\mathbf{x}) - \nabla^3 f(\mathbf{y})\right\| \leq M \left\|\mathbf{x}-\mathbf{y}\right\|$$

  [Zhang+ 12, Savgm] [Shamir+ 13, Dane] [Dieuleveut+ 19, Local SGD] [Yuan+ 20, Accelerated Local SGD]

  - bounded Hessian dissimilarity **[SCAFFOLD] [Mime]**

$$\left\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f(\mathbf{x})\right\|^2 \leq \delta$$

- Stronger oracles? discussion in [Woodworth+ 20b]
  - second order
  - proximal oracles

Assumptions that hold in the DL setting?
(also for optimization in DL more generally...)

Orthogonal Techniques

Client Sampling, $s \leq n$ clients per round

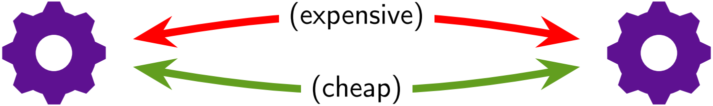
- standard in FedAvg to handle large number of clients
- does not significantly change the discussion (of here considered aspects)
- mini-batch SGD with client sampling (and variance $\frac{\sigma^2}{s\tau} + \left(1 - \frac{s}{n}\right)\frac{\zeta^2}{s}$) is still a strong baseline

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

[McMahan+ 16, FedAvg] [Woodworth+ 20b] **[Karimireddy+ 20a]**

Compression & Asynchronous Updates

standard: large and frequent updates



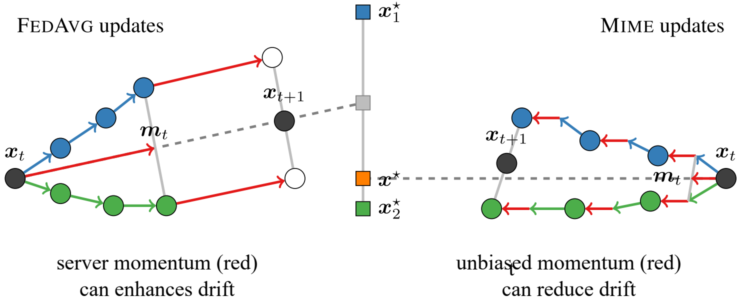
compressed and infrequent messages

- Compress (model updates)
  [S+ 18] [Alistarh+ 18] [Mishchenko+ 19] [Vogels+ 19] **[S 20]** **[S+ 20]**
- Asynchronous communication **[S+ 21]** [Aviv+ 21] [Nguyen+ 21]
- Adaptive Communication Frequency
  [Chen+ 18] [Haddadpour+ 19] **[Ghadikolaei+ 21]**

Some of these techniques have also been proven useful in data-center training [Assran+ 18] [Ramesh+ 21, Dall-E].

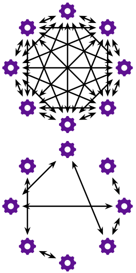
Local Momentum-SGD? Local-ADAM?

- We can translate the convergence of a generic base-algorithm (SGD with momentum, ADAM, etc.) from the centralized setting into convergence in the federated setting.
- momentum can help reduce client drift:



[Karimireddy+ 20b]

Decentralized Optimization



- **Arbitrary communication topology (graph):**
  - $\mathcal{G} = ([n], E)$
  - communication only along the edges E,
    i, j connected $\Leftrightarrow (i, j) \in E$
  - time-varying (or random) graphs possible
- **FL is a special case!**

  no communication for $\tau - 1$ steps, fully-connected graph every τ-th step

[Tsitsiklis+ 86] [Xiao+ 04] [Nedić+ 09] [Chen+ 12] [Pu+ 18] **[Koloskova+ 19] [Koloskova+ 20] [Kovalev+ 21] [Kong+ 21] [Lin+ 21]**

Conclusion

- **Rethinking assumptions!**
    - stronger assumptions better capture algorithm's behaviors
    - but we move even further away from DL?
- **Careful evaluation needed!**
- **Challenges of non-convex/DL optimization!**

Main references and collaborators

- 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.
- P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S.J. Reddi, S.U. Stich and A.T. Suresh, Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning, arXiv:2008.03606.
- 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.
- B. Woodworth, K.K. Patel and N. Srebro, Minibatch vs Local SGD for Heterogeneous Distributed Learning, NeurIPS 2020.
- S.U. Stich, Local SGD Converges Fast and Communicates Little, ICLR 2019.