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)

\[
\min_{x \in \mathbb{R}^d} \left[ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right] \quad f_i(x) = \mathbb{E}_{\xi \sim D_i} F(x, \xi)
\]

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

Other very relevant scenarios: future talks today ;-

- personalization • heterogeneity • privacy • robustness
Base-Algorithm

Stochastic Gradient Descent:

\[ f(x) = \mathbb{E}_{\xi \sim D} F(x, \xi) \]  

loss function

\[ \xi \sim D \] (unknown) data distribution

\[ \min_{x \in \mathbb{R}^d} f(x) \quad \gamma \text{ stepsize} \]

\[ \xi^{(t)} \sim D, \quad x^{(t+1)} := x^{(t)} - \gamma \nabla F(x^{(t)}, \xi^{(t)}) \]

uniform data sample, mini-batch

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 \((n = 1)\)

\[ f(x) = E_{\xi \sim \mathcal{D}} F(x, \xi) \]

- **(Standard) Assumptions**
  - \(\text{Var}[\nabla F(x, \xi)] \leq \sigma^2\)
  - \(\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|\)

- **Convergence**
  - \(L\)-smooth
    - \(\mathbb{E} \|\nabla f(x_{\text{out}})\|^2 \leq \epsilon\)
    - \(O\left(\frac{\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)\)
  - \(\mu\)-star convex\(^1\)
    - \(\mathbb{E}f(x_{\text{out}}) - f(x^*) \leq \epsilon\)
    - \(O\left(\frac{\sigma^2}{\mu \epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon}\right)\)

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

\([\text{Lan } 11, \text{ Accelerated SGD}] [\text{Bottou+ } 16, \text{ book}] [\text{S } 19]\)

\(^1\exists x \in \mathbb{R}^d \text{ s.t. } \langle \nabla f(x) - \nabla f(x^*), x - x^* \rangle \geq \mu \|x - x^*\|^2, \forall x \in \mathbb{R}^d\)
Background II: mini-batch SGD baseline

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

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

\[
\xi_i^{(t)} \sim D_i \quad x^{(t+1)} = x^{(t)} - \frac{\gamma}{n} \sum_{i=1}^{n} \nabla F(x^{(t)}, \xi_i^{(t)})
\]

- **Convergence:**

\[
O \left( \frac{\sigma^2}{n \mu \epsilon} + \frac{L}{\mu \log \frac{1}{\epsilon}} \right)
\]

linear speedup

[Dekel+ 10] [Goyal+ 17] [Lin+ 20] [S+ 21]
Training limited by communication bottleneck:

\[ \nabla F(x^{(t)}, \xi^{(t)}) \]

<table>
<thead>
<tr>
<th>algorithm</th>
<th>rounds</th>
<th>gradients (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mini-batch SGD</td>
<td>( O \left( \frac{\sigma^2}{n\mu\epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
<td>( O \left( \frac{\sigma^2}{\mu\epsilon} + \frac{nL}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>batch size ( \tau = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mini-batch SGD</td>
<td>( O \left( \frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
<td>( O \left( \frac{\sigma^2}{\mu\epsilon} + \frac{n\tau L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>batch size ( \tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Local Update Methods
+ increasing batch size reduces communication
- but no progress while computing the batch gradients

• for $\tau \rightarrow \infty$, stuck at $x_0$ forever!

\[
x_i^{(t+1)} = x_i^{(t)} - \gamma \nabla F(x_i^{(t)}, \xi_i^{(t)}) \times \tau \text{ times}
\]

+ use local gradients for local model updates

• for $\tau \rightarrow \infty$, each client converges to local solution $x_i^*$

- different models $x_i$ on clients!
Local SGD

\[ x_i^{(t+1)} := \begin{cases} 
  x_i^{(t)} - \gamma \nabla F(x_i^{(t)}, \xi_i^{(t)}) & \text{if } t + 1 \notin \{0, \tau, 2\tau, \ldots\} \\
  \frac{1}{n} \sum_{i=1}^{n} \left( x_i^{(t)} - \gamma \nabla F(x_i^{(t)}, \xi_i^{(t)}) \right) & \text{if } t + 1 \in \{0, \tau, 2\tau, \ldots\} 
\end{cases} \]

in general: \( x_i^{(t)} \neq x_j^{(t)} \) for \( i \neq j \) and \( t + 1 \notin \{0, \tau, 2\tau, 3\tau, \ldots\} \)

- leverages parallelism (unlike single-machine SGD)
- makes \( \tau \) 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

\[ \bar{x}^{(t+1)} := \bar{x}^{(t)} - \frac{\gamma}{n} \sum_{i=1}^{n} \nabla F(x_i^{(t)}, \xi_i^{(t)}) \]

- \( \bar{x}^{(t)} \) behaves almost as ‘normal’ SGD \( \mathbb{E} \| \bar{x}^{(t)} - x^* \|^2 \to 0 \)
- the additional error term \( \mathbb{E} \| x_i^{(t)} - \bar{x}^{(t)} \|^2 \) can be controlled
  - IID data (\( D_i = D_j \))
    \[ \mathbb{E} \| x_i^{(t)} - \bar{x}^{(t)} \|^2 = \gamma^2 \cdot O \left( \tau^2 \| \nabla f(\bar{x}^{(t)}) \|^2 + \frac{\tau}{n} \sigma^2 \right) \]
  - non-IID data (\( D_i \neq D_j \)) additional properties
    \[ \mathbb{E} \| x_i^{(t)} - \bar{x}^{(t)} \|^2 = \gamma^2 \cdot O \left( \tau^2 \| \nabla f(\bar{x}^{(t)}) \|^2 + \tau^2 \zeta^2 + \frac{\tau}{n} \sigma^2 \right) \]
    \[ \mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \|^2 \leq \zeta^2 \] data-dissimilarity
    \[ \text{inter-client variance} \]

[S 19] [S+ 20] [Koloskova+ 20]
Data-dissimilarity $\zeta^2 > 0$ causes drift when doing local steps.

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

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!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>mini-batch SGD</td>
<td>( O \left( \frac{\sigma^2}{n\mu\epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>batch size ( \tau = 1 )</td>
<td></td>
</tr>
<tr>
<td>mini-batch SGD</td>
<td>( O \left( \frac{\sigma^2}{n\tau\mu\epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>batch size ( \tau )</td>
<td></td>
</tr>
<tr>
<td>local SGD</td>
<td>( 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) )</td>
</tr>
<tr>
<td>( \tau ) local steps</td>
<td></td>
</tr>
</tbody>
</table>

- (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

\[ x_i^{(t+1)} = x_i^{(t)} - \gamma \left( \nabla F(x_i^{(t)}, \xi_i^{(t)}) - c_i^{(t)} + c^{(t)} \right) \]

- \( c_i^{(t)} \approx \nabla f_i(\bar{x}_i^{(t)}) \) local drift
- \( c^{(t)} \approx \nabla f(\bar{x}^{(t)}) \) global drift

\# drift correction

- correction does not depend on local steps and is \textit{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
  \[
  c_i^{(t)} = g_i^{(t)}, \quad c^{(t)} = \frac{1}{n} \sum_{i=1}^{n} c_i^{(t)}
  \]

- if \( n \) is huge, SCSG-type correction
  \[
  c_i^{(t)} = g_i^{(t)}, \quad c^{(t)} = \frac{1}{|S_t|} \sum_{i \in S_t} c_i^{(t)}, \text{ for active clients } S_t \subset [n]
  \]

Here \( g_i^{(t)} \) denotes a (stochastic (possibly mini-batch) or full batch) gradient.

Theoretical Results with Bias Correction

<table>
<thead>
<tr>
<th>algorithm</th>
<th>rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>mini-batch SGD</td>
<td>( \mathcal{O} \left( \frac{\sigma^2}{n \tau \mu \epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>batch size ( \tau )</td>
<td></td>
</tr>
<tr>
<td>SCAFFOLD</td>
<td>( \tilde{\mathcal{O}} \left( \frac{\sigma^2}{n \tau \mu \epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right) )</td>
</tr>
<tr>
<td>( \tau ) local steps + proper init</td>
<td></td>
</tr>
</tbody>
</table>
Benefits of Local Updates?
Intermezzo: Some Experiments

- drift correction accelerates training on non-IID data!

<table>
<thead>
<tr>
<th></th>
<th>non-IID data</th>
<th>IID data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% similarity (sorted)</td>
<td>10% similarity</td>
</tr>
<tr>
<td></td>
<td>Num. of rounds</td>
<td>Speedup</td>
</tr>
<tr>
<td>SGD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>317</td>
<td>(1×)</td>
</tr>
<tr>
<td>SCAFFOLD1</td>
<td>77</td>
<td>(4.1×)</td>
</tr>
<tr>
<td>5</td>
<td>152</td>
<td>(2.1×)</td>
</tr>
<tr>
<td>FedAvg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>258</td>
<td>(1.2×)</td>
</tr>
<tr>
<td>5</td>
<td>428</td>
<td>(0.7×)</td>
</tr>
</tbody>
</table>

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!

<table>
<thead>
<tr>
<th></th>
<th>Top-1 acc.</th>
<th>local gradients</th>
<th>communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini-batch SGD (n = 16, τ = 128)</td>
<td>92.5%</td>
<td>2048</td>
<td>-</td>
</tr>
<tr>
<td>Mini-batch SGD (n = 16, τ = 1024)</td>
<td>76.3%</td>
<td>16384</td>
<td>÷ 8</td>
</tr>
<tr>
<td>Local-SGD (n = 16, τ = 8 × 128)</td>
<td>92.0%</td>
<td>16384</td>
<td>÷ 8</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>mini-batch SGD</td>
<td>$\mathcal{O} \left( \frac{\sigma^2}{n \tau \mu \epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>local SGD</td>
<td>$\mathcal{O} \left( \frac{\sigma^2}{n \tau \mu \epsilon} + \frac{L}{\tau \mu} \log \frac{1}{\epsilon} \right)$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAFFOLD</td>
<td>$\tilde{\mathcal{O}} \left( \frac{\sigma^2}{n \tau \mu \epsilon} + \frac{L}{\tau \mu} \log \frac{1}{\epsilon} \right)$</td>
</tr>
</tbody>
</table>

Remarkable: benefit from parallelism and serial updates!

[Woodworth+ 20a] [Woodworth+ 20b] [Karimireddy+ 20a]
‘Breaking’ Lower Bounds!

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

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

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

- bounded Hessian dissimilarity
  - [SCAFFOLD] [Mime]

$$\| \nabla^2 f_i(x) - \nabla^2 f(x) \|^2 \leq \delta$$

- 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 $\frac{\sigma^2}{s\tau} + (1 - \frac{s}{n}) \frac{\zeta^2}{s}$) is still a strong baseline

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>mini-batch SGD</td>
<td>$\mathcal{O} \left( \frac{\sigma^2}{s\tau \mu \epsilon} + \left(1 - \frac{s}{n}\right) \frac{\zeta^2}{s\mu \epsilon} + \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>$\tau$ batch size</td>
<td></td>
</tr>
<tr>
<td>SCAFFOLD</td>
<td>$\tilde{\mathcal{O}} \left( \frac{\sigma^2}{s\tau \mu \epsilon} + \left(\frac{L}{\mu} + \frac{n}{s}\right) \log \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>$\tau$ local steps</td>
<td></td>
</tr>
</tbody>
</table>

[McMahan+ 16, FedAvg] [Woodworth+ 20b] [Karimireddy+ 20a]
Compression & Asynchronous Updates

standard: large and frequent updates

(expensive)

(cheap)

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:

![Diagram showing FEDAVG and MIME updates with server momentum and unbiased momentum](image)

server momentum (red) can enhances drift

unbiased momentum (red) can reduce drift

[Karimireddy+ 20b]
Decentralized Optimization

- Arbitrary communication topology (graph):
  - $G = ([n], E)$
  - communication only along the edges $E$,
    $i, j$ connected $\iff (i, j) \in E$
  - time-varying (or random) graphs possible
- FL is a special case!
  - no communication for $\tau - 1$ steps, fully-connected graph every $\tau$-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


B. Woodworth, K.K. Patel and N. Srebro, Minibatch vs Local SGD for Heterogeneous Distributed Learning, NeurIPS 2020.