Advances in Federated Optimization: Efficiency, Resiliency, and Privacy

Yuejie Chi and Zhize Li

Carnegie Mellon University

ICASSP Tutorial
June 2023
Acknowledgements

Boyue Li
CMU→Apple

Haoyu Zhao
Princeton

This work is supported in part by NSF, ONR, and AFRL.
Introduction
Empirical Risk Minimization (ERM)

Given a set of data $\mathcal{M}$,

$$\minimize_{x} \quad f(x) = \frac{1}{N} \sum_{z \in \mathcal{M}} \ell(x; z)$$

Here, $N =$ number of total samples.

- **convex**: least squares, logistic regression
- **non-convex**: PCA, training neural networks (focus of this tutorial)
Let’s go distributed

**Distributed/Federated learning:** due to privacy and scalability, data are distributed at multiple locations / workers / agents.

Let $\mathcal{M} = \bigcup_i \mathcal{M}_i$ be a data partition with equal splitting:

$$f(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}), \quad \text{where} \quad f_i(\mathbf{x}) := \frac{1}{(N/n) m} \sum_{z \in \mathcal{M}_i} \ell(\mathbf{x}; z).$$

$n$ = number of agents

$N/n = \underbrace{\text{number of local samples}}_{m}$
Federated learning is deployed nowadays by companies in many areas, e.g., on-device inference.
Multi-agent and distributed information processing

Decentralized processing without central coordination in wireless sensor networks, internet of things, swarms, ...
Two distributed schemes

Server/client model

PS coordinates *global* information sharing

Network/decentralized model

agents share *local* information over a graph topology
Two data regimes

Entities

cross-silo
small $n$, large $m$

Devices

cross-device:
small $m$, large $n$
Challenges in federated/decentralized learning

- **Communication efficiency**: limited bandwidth, stragglers, ...

- **Heterogeneity**: non-iid data and systems across the agents

- **Privacy**: does not come for free without sharing data
Communication efficiency

Communication cost = Communication rounds × Cost per round

- **Local method**: perform more local computation to reduce communication rounds, e.g. FedAvg *(McMahan et al., 2016)*.

- **Communication compression**: compress the message into fewer bits, e.g. sparsification or quantization *(Alistarh et al., 2017)*.

---

**How to design communication-efficient algorithms?**
Data heterogeneity

Heterogeneity measure

\[ local \ objective \neq global \ objective \]

— Can we tame the data heterogeneity?
A little privacy, please

Privacy guarantees are becoming increasingly critical!

— Can we design privacy-preserving algorithms?
Tutorial outline

Part 0: Primer on centralized nonconvex optimization

Part 1: Efficient federated optimization via local methods
  • Federated averaging
  • SCAFFOLD: dealing with heterogeneity via variance reduction

Part 2: Communication-compressed federated optimization
  • How do we compress? the role of error feedback
  • Dealing with data heterogeneity

Part 3: Private federated optimization
  • Differential privacy
  • Understanding gradient clipping
Part 0: A Primer on Centralized Nonconvex Optimization
Unconstrained optimization

Consider an unconstrained optimization problem

\[
\text{minimize}_x \quad f(x)
\]

**Definition (first-order critical points)**

A first-order critical point of \( f \) satisfies

\[
\nabla f(x) = 0
\]

How do we converge to first-order critical points?
A function $f(x)$ is $L$-smooth if

$$\|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L \|x_1 - x_2\|_2$$
Convergence of gradient descent (GD)

Gradient descent (GD):

\[ x_{t+1} = x_t - \eta_t \nabla f(x_t) \]

where \( \eta_t \) is the learning rate.

**Theorem (Convergence of GD)**

Suppose \( f^* = \min_x f(x) > -\infty \). Setting \( \eta_t = \eta = 1/L \), it satisfies

\[ \frac{1}{T} \sum_{t=0}^{T-1} \| \nabla f(x_t) \|_2^2 \leq \frac{2L\Delta}{T}, \]

where \( \Delta = f(x_0) - f^* \).

- GD converges at the rate \( O(1/T) \) in terms of the average squared gradient norm.
- For finite-sum problems of size \( n \), the IFO complexity of GD is \( O(n\varepsilon^{-1}) \) to reach \( \mathbb{E}\| \nabla f(x^{\text{output}}) \|_2^2 \leq \varepsilon \).
Convergence of GD under smoothness

- By smoothness,

\[ f(x_{t+1}) - f(x_t) = f(x_t - \eta \nabla f(x_t)) - f(x_t) \]
\[ \leq \langle \nabla f(x_t), -\eta \nabla f(x_t) \rangle + \frac{L}{2} \| \eta \nabla f(x_t) \|^2 \]
\[ = -\left( \eta - \frac{\eta^2 L}{2} \right) \| \nabla f(x_t) \|^2 \]
\[ \leq -\frac{\eta}{2} \| \nabla f(x_t) \|^2 \]

as long as \( \eta \leq 1/L \).

- Telescoping \( t = 0, 1, \ldots, T - 1 \) gives

\[ \frac{\eta}{2} \sum_{t=0}^{T-1} \| \nabla f(x_t) \|^2 \leq \sum_{t=0}^{T-1} (f(x_{t+1}) - f(x_t)) = f(x_0) - f(x_T) \leq \Delta. \]

Setting \( \eta = 1/L \) finishes the proof.
Stochastic gradient descent (SGD):

\[ x_{t+1} = x_t - \eta_t \nabla \ell(x_t; z_t), \quad z_t \sim \mathcal{M} \]

where \( \eta_t \) is the learning rate.

- **Unbiasedness:**
  \[ \mathbb{E}_z[\nabla \ell(x; z)] = \nabla f(x). \]

- Additional assumption is needed for convergence analysis.

**Definition (Bounded gradient assumption)**

For any \( x, z \in \mathbb{R}^d \), there exists some \( G > 0 \) such that

\[ \| \nabla \ell(x, z) \|_2 \leq G. \]
Convergence of SGD under bounded gradient

**Theorem (Convergence of SGD)**

Suppose $f^* = \min_x f(x) > -\infty$ and $\|\nabla \ell(x, z)\|_2 \leq G$ for any $x, z \in \mathbb{R}^d$. Setting $\eta = \sqrt{\frac{2\Delta}{G^2 LT}}$, it satisfies

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(x_t)\|_2^2 \leq G\sqrt{\frac{2L\Delta}{T}}.$$ 

- SGD converges at the rate $O(1/\sqrt{T})$ in terms of the expected average squared gradient norm, which is slower than GD.

- For finite-sum problems of size $n$, the IFO complexity of SGD is $O(\varepsilon^{-2})$ to reach $\mathbb{E}\|\nabla f(x^{\text{output}})\|_2^2 \leq \varepsilon$. 

Convergence of SGD

• By smoothness,

\[
\begin{align*}
    f(x_{t+1}) - f(x_t) &= f(x_t - \eta \nabla \ell(x_t; z_t)) - f(x_t) \\
    &\leq \langle \nabla f(x_t), -\eta \nabla \ell(x_t; z_t) \rangle + \frac{L}{2} \left\| \eta \nabla \ell(x_t; z_t) \right\|^2 \\
    &\leq -\eta \langle \nabla f(x_t), \nabla \ell(x_t; z_t) \rangle + \frac{\eta^2 G^2 L}{2}.
\end{align*}
\]

• Taking conditional expectation at the \( t \)-th iterate,

\[
\mathbb{E}_t f(x_{t+1}) - f(x_t) \leq -\eta \left\| \nabla f(x_t) \right\|^2 + \frac{\eta^2 G^2 L}{2}.
\]

• Telescoping \( t = 0, 1, \ldots, T - 1 \) gives

\[
\frac{1}{T} \mathbb{E} \sum_{t=0}^{T-1} \left\| \nabla f(x_t) \right\|^2 \leq \frac{(f(x_0) - \mathbb{E} f(x_T))}{\eta T} + \frac{\eta G^2 L}{2} \leq \frac{\Delta}{\eta T} + \frac{\eta G^2 L}{2}.
\]

Setting \( \eta = \sqrt{\frac{2\Delta}{G^2 LT}} \) finishes the proof.
## Bounded variance assumption

### Definition (Bounded variance assumption)
For any $\mathbf{x} \in \mathbb{R}^d$, there exists some $\sigma > 0$ such that
\[
\mathbb{E}_z \| \nabla \ell(\mathbf{x}, z) - \nabla f(\mathbf{x}) \|^2 \leq \sigma^2.
\]

- Under unbiasedness, this assumption is equivalent to
\[
\mathbb{E}_z \| \nabla \ell(\mathbf{x}, z) \|^2 \leq \| \nabla f(\mathbf{x}) \|^2 + \sigma^2.
\]

The convergence of SGD can be established under the relaxed bound variance assumption (Ghadimi and Lan, 2013):
\[
\mathbb{E} \| \nabla f(\mathbf{x}^{\text{output}}) \|^2 \lesssim \frac{L \Delta}{T} + \sigma \sqrt{\frac{\Delta}{LT}}
\]

- By picking large enough batch size to make $\sigma$ sufficiently small, the rate matches that of GD.
Can we achieve faster rate?

**Variance reduction:** perform SGD with a carefully designed stochastic gradient (SG) \( g_t \):

\[
x_{t+1} = x_t - \eta_t g_t
\]

**SVRG** (Johnson and Zhang, 2013) assumes \((x_0, \nabla f(x_0))\) is a reference point,

\[
g_t = \nabla \ell(x_t; z_t) - \nabla \ell(x_0; z_t) + \nabla f(x_0)
\]

- **Unbiased:** \( \mathbb{E}[g_t] = \nabla f(x^t) \);
- **Variance:**

\[
g_t - \nabla f(x_t) = [\nabla \ell(x_t; z_t) - \nabla f(x_t)] - [\nabla \ell(x_0; z_t) - \nabla f(x_0)]
\]

if the two terms are positively correlated, then variance reduction occurs, i.e. \( \text{Var}[g_t] \ll \text{Var}[\nabla \ell(x_t; z_t)] \).
- **Update the reference** \( x_0 \) periodically.
Can we achieve faster rate?

**Variance reduction:** perform SGD with a carefully designed stochastic gradient (SG) $g_t$:

$$x_{t+1} = x_t - \eta_t g_t$$

**SAGA** (Defazio et al., 2014) maintains a table of stochastic gradient $g(z)$ at each sample $z$:

$$g_t = \nabla \ell(x_t; z_t) - g(z_t) + \frac{1}{n} \sum_{z \in M} g(z)$$

$$g(z_t) \leftarrow \nabla \ell(x_t; z_t)$$

For finite-sum problems of size $n$, the IFO complexity of SVRG/SAGA achieves the rate $O(n + n^{2/3} \varepsilon^{-1})$ to reach $\mathbb{E}\|\nabla f(x^{\text{output}})\|_2^2 \leq \varepsilon$, which is sub-optimal.
Can we achieve the optimal rate?

**Variance reduction:** perform SGD with a carefully designed stochastic gradient (SG) $g_t$:

$$x_{t+1} = x_t - \eta_t g_t$$

**SARAH/Spider** (Nguyen et al., 2017; Fang et al., 2019) assumes $(x_0, \nabla f(x_0))$ is a reference point,

$$g_t = \nabla \ell(x_t; z_{i_t}) - \nabla \ell(x_{t-1}; z_{i_t}) + g_{t-1}$$

where $g_0 = \nabla f(x^0)$.

- Biased: $\mathbb{E}[g_t] \neq \nabla f(x_t)$;
- The stochastic gradient is recursive.

For finite-sum problems of size $n$, the IFO complexity of SARAH/Spider achieves the optimal rate $O(n + n^{1/2} \varepsilon^{-1})$ to reach $\mathbb{E}\|\nabla f(x^{\text{output}})\|_2^2 \leq \varepsilon$. 
The IFO complexity to reach $\mathbb{E}\|\nabla f(x^{\text{output}})\|_2^2 \leq \varepsilon$ under smoothness.

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
<th>Additional assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>$\frac{n}{\varepsilon}$</td>
<td>none</td>
</tr>
<tr>
<td>SGD</td>
<td>$\frac{1}{\varepsilon^2}$</td>
<td>bounded gradient or bounded variance</td>
</tr>
<tr>
<td>SVRG/SAGA</td>
<td>$n + \frac{n^{2/3}}{\varepsilon}$</td>
<td>none</td>
</tr>
<tr>
<td>SARAH/Spider</td>
<td>$n + \frac{n^{1/2}}{\varepsilon}$</td>
<td>none</td>
</tr>
<tr>
<td>Lower bound</td>
<td>$n + \frac{n^{1/2}}{\varepsilon}$</td>
<td>none</td>
</tr>
</tbody>
</table>
Part 1: Communication-efficient Federated Optimization via Local Methods
Federated Averaging (FedAvg): the first FL algorithm (McMahan et al., 2016) that alternates between local updates and global averaging.

- Also known as local SGD: the number of local updates $= E$.
- When $E = 1$, reduces to distributed SGD:

$$x_{t+1} = x_t - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_t)$$
Convergence guarantees of FedAvg

Definition (Bounded gradient dissimilarity)

There exist constants $G \geq 0$ and $B \geq 1$ such that for all $x \in \mathbb{R}^d$:

$$\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(x)\|_2^2 \leq G^2 + B^2 \|\nabla f(x)\|_2^2.$$

- Treating $f_i(x)$ as sampling $f(x)$, this assumption mimics the bounded variance assumptions. When $B = 1$,

$$\mathbb{E}_{i \sim [n]} \|\nabla f_i(x) - \nabla f(x)\|_2^2 = \frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(x)\|_2^2 - \|\nabla f(x)\|_2^2 \leq G^2.$$

When $f_i = f$, set $G = 0$ and $B = 1$. 
Theorem (Karimireddy et al., 2019)

To achieve \( \mathbb{E}\|\nabla f(x^{\text{output}})\|^2 \leq \varepsilon \), FedAvg takes at most an order of

\[
\frac{\sigma^2}{mE\varepsilon^2} + \frac{G}{\varepsilon^{3/2}} + \frac{B^2}{\varepsilon}
\]

iterations, where \( \sigma^2 \) is the local sampling variance.

- \( \frac{\sigma^2}{mE\varepsilon^2} \): error due to local stochasticity
- \( \frac{G}{\varepsilon^{3/2}} + \frac{B^2}{\varepsilon} \): error due to client heterogeneity
FedAvg is sensitive to data heterogeneity. The performance gets worse with more local updates for heterogeneous data.

Figure credit: (Hsu et al., 2019)
FedAvg is sensitive to data heterogeneity

Client drift: the average of the local optima is not the global optimum!

How to design better algorithms that are more resilient to heterogeneous data?
SCAFFOLD: leveraging variance reduction

**SCAFFOLD** (Karimireddy et al., 2019): federated SAGA (Defazio et al., 2014)

- Client $i$ performs $K$ steps of SGD using local control variate $c_i$

\[
y_t^i \leftarrow y_t^i - \eta (g_i(y_t^i) + \underbrace{c - c_i}_\text{correction})
\]

- $c$: estimated update direction for server
- $c_i$: estimated update direction for client $i$

---

**Algorithm 1** SCAFFOLD: Stochastic Controlled Averaging for federated learning

1. **server input**: initial $x$ and $c$, and global step-size $\eta_g$
2. **client $i$’s input**: $c_i$, and local step-size $\eta_l$
3. **for** each round $r = 1, \ldots, R$ **do**
4. **sample clients** $S \subseteq \{1, \ldots, N\}$
5. **communicate** $(x, c)$ to all clients $i \in S$
6. **on client** $i \in S$ **in parallel do**
   7. initialize local model $y_i \leftarrow x$
   8. **for** $k = 1, \ldots, K$ **do**
   9. compute mini-batch gradient $g_i(y_i)$
   10. $y_i \leftarrow y_i - \eta_l (g_i(y_i) - c_i + c)$
   **end for**
   11. $c_i^+ \leftarrow$ (i) $g_i(x)$, or (ii) $c_i - c + \frac{1}{K\eta_l} (x - y_i)$
   12. **communicate** $(\Delta y_i, \Delta c_i) \leftarrow (y_i - x, c_i^+ - c_i)$
   13. $c_i \leftarrow c_i^+$
   **end on client**
14. $(\Delta x, \Delta c) \leftarrow \frac{1}{|S|} \sum_{i \in S} (\Delta y_i, \Delta c_i)$
15. $x \leftarrow x + \eta_g \Delta x$ and $c \leftarrow c + \frac{|S|}{N} \Delta c$
16. **end for**
Convergence of SCAFFOLD

**Theorem (Karimireddy et al., 2019)**

To achieve $\mathbb{E}\|\nabla f(x^{\text{output}})\|^2 \leq \varepsilon$, SCAFFOLD takes at most an order of

$$\frac{\sigma^2}{mE\varepsilon^2} + \frac{1}{\varepsilon}$$

iterations, where $\sigma^2$ is the local sampling variance.

- Handles arbitrary data heterogeneity: does not require the bounded dissimilarity assumption!
- Also allows client sampling; details in the paper.
FedAvg versus SCAFFOLD

Figure credit: (Karimireddy et al., 2019)
Key takeaways and further pointers

**Key takeaways:**

- Local updates help improve communication efficiency
- FedAvg is sensitive to data heterogeneity
- Leverage variance reduction to deal with heterogeneity

**Further pointers:**

- Client sampling
Part 2: Communication-compressed federated optimization
Communication compression

Communication compression is a popular approach to reduce communication cost (e.g., (Alistarh et al., 2017); (Koloskova et al., 2019)).

\[ \mathbb{E}\|C(x) - x\|^2 \leq (1 - \alpha)\|x\|^2 \]

- **random sparsification**: \( \alpha = k/d \) measures the compression ratio.
- Other examples: random quantization, top-\( k \) quantization, etc.
A prelude: what should we compress?

What about

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \frac{1}{n} \sum_{i=1}^{n} C(\nabla f_i(\mathbf{x}^t))$$?

Somewhat surprisingly, direct compression may not work!
A counter-example

Consider $n = 3$ and let $f_i(x) = (a_i^T x)^2 + \frac{1}{2} \|x\|^2$, where

$$a_1 = (-4, 3, 3)^\top, \quad a_2 = (3, -4, 3)^\top \quad \text{and} \quad a_3 = (3, 3, -4)^\top.$$ 

• Let $x^0 = (b, b, b)$, and the compressor be $\text{top}_1$,

$$\nabla f_1(x^0) = b(-15, 13, 13)^\top \quad \rightarrow \quad C(\nabla f_1(x^0)) = b(-15, 0, 0)^\top$$

$$\nabla f_2(x^0) = b(13, -15, 13)^\top \quad \rightarrow \quad C(\nabla f_2(x^0)) = b(0, -15, 0)^\top$$

$$\nabla f_3(x^0) = b(13, 13, -15)^\top \quad \rightarrow \quad C(\nabla f_3(x^0)) = b(0, 0, -15)^\top$$

• The next iteration

$$x^1 = x^0 - \eta \frac{1}{3} \sum_{i=1}^{3} C(\nabla f_i(x^0)) = (1 + 5\eta)x^0,$$

and then $x^t = (1 + 5\eta)^t x^0$ diverges exponentially.
A better scheme: shift compression / error feedback

(Stich et al., 2018; Richtárik et al., 2021)

• The PS updates the model:

\[ x^{t+1} = x^t - \frac{\eta}{n} \sum_{i=1}^{n} g_i^t \]

— \( g_i^t \) is the compressed surrogate of \( \nabla f_i(x^t) \)

• Clients update \( g_i^t \) with a shift compression:

\[ g_i^{t+1} = g_i^t + C(\nabla f_i(x^{t+1}) - g_i^t) \]

— \( g_i^t \) is constructed accumulatively over time
Let’s revisit the example

- Let \( x^0 = (b, b, b) \), and the compressor be \( \text{top}_1 \), \( g_i^0 = C(\nabla f_i(x^0)) \), and the first iteration is still \( x^1 = (1 + 5\eta)x^0 \).

- **Error feedback:**

\[
\nabla f_1(x^1) - g_1^0 = b \begin{bmatrix} -75\eta \\ 13(1 + 5\eta) \\ 13(1 + 5\eta) \end{bmatrix}
\]

and as long as \( \eta < 13/30 \):

\[
C (\nabla f_1(x^1) - g_1^0) = b \begin{bmatrix} 0 \\ 13(1 + 5\eta) \\ 0 \end{bmatrix}
\]

receiving information from coordinates other than the first one, leading to a better compressed gradient!
Let’s revisit the example

We’ll consider algorithms using shifted compression!
Case study: decentralized nonconvex optimization

- The mixing of information is characterized by a mixing matrix $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ aligned with the network topology.
- The spectral quantity, which we call the spectral gap,
  \[ \rho \triangleq 1 - |\lambda_2(W)| \in (0, 1] \]
  captures how fast information mixes over the network.

Goal: design fast-converging algorithms with communication compression
Data heterogeneity

Heterogeneity measure

\[ E_i \| \nabla f_i(x) - \nabla f(x) \|_2^2 \leq G^2 \]

- **local obj.**
- **global obj.**

— *G can be unbounded!*
Prior art

CHOCO-SGD (Koloskova et al., 2019) / DeepSqueeze (Tang et al., 2019):
- slow convergence rates (need more communication rounds) and
- Incompatible with heterogeneity

Can we converge at the rate $O\left(\frac{1}{\varepsilon}\right)$ under arbitrary heterogeneity?

Yes, by using gradient tracking!
Detour: DGD with gradient tracking

**Centralized Gradient Descent (GD):**

\[ x^t = x^{t-1} - \eta \nabla f(x^{t-1}) \]

Constant step size, linear convergence for strongly convex problems.

**Decentralized Gradient Descent (DGD):**

\[ x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta \nabla f_i(x_i^{t-1}) \]

Where:
- **mixing**
- **local gradient**

Constant step size, does not converge!

At optimal point \( x^* \): \( \nabla f(x^*) = 0 \), but \( \nabla f_i(x^*) \neq 0 \)

How do we fix this?
DGD with gradient tracking

Use dynamic average consensus (Zhu and Martinez, 2010) to track the global gradient $s^t_i$:

$$x^t_i = \sum_j w_{ij}x^{t-1}_j - \eta s^t_i$$  

mixing

$$s^t_i = \sum_j w_{ij}s^{t-1}_i + \nabla f_i(x^t_i) - \nabla f_i(x^{t-1}_i)$$  

mixing gradient tracking

This trick, and other alternatives, have been used extensively to fix the non-convergence issue in decentralized optimization.

- EXTRA (Shi, Ling, Wu and Yin, 2015); NEXT (Di Lorenzo and Scutari, 2016);
- NIDS (Li, Shi, Yan, 2017); ADD-OPT (Xi, Xin, and Khan, 2017); DIGING (Nedic, Olshevsky, and Shi, 2017); DGD (Qu and Li, 2018);
- many, many more...
BEER: gradient tracking + shift compression

\[ X = [x_1, x_2, \cdots, x_n]: \text{local models.} \]
\[ \nabla F(X) = [\nabla f_1(x_1), \nabla f_2(x_2), \cdots, \nabla f_n(x_n)]: \text{local gradients.} \]

• **model update:**

\[
X^{t+1} = X^t + \gamma H^t(W - I) - \eta V^t
\]

where \( H^t \) is the accumulated compressed surrogate of \( X^t \), and \( V^t \) is the global gradient estimates across the agents.

• **gradient tracking:**

\[
V^{t+1} = V^t + \gamma G^t(W - I) + \nabla F(X^{t+1}) - \nabla F(X^t),
\]

where \( G^t \) is the accumulated compressed surrogate of \( V^t \).

• Both \( H^t \) and \( G^t \) are updated using **shift compression**.
Theoretical convergence of BEER

Theorem (Zhao et al., NeurIPS 2022)

To achieve $\mathbb{E}\|\nabla f(x^{\text{output}})\|^2 \leq \varepsilon$, BEER requires at most

$$O\left(\frac{1}{\rho^3 \alpha \varepsilon}\right)$$

communication rounds, without the bounded heterogeneity assumption. Here, $\alpha$ is the compression ratio, $\beta$ is the spectral gap of the network.

- Assuming constant $\alpha$ and $\rho$, the convergence rate of BEER is

$$O\left(\frac{1}{\varepsilon}\right).$$

- Can also be extended to using stochastic gradients.
Theoretical convergence of BEER

BEER converges at the rate $O\left(\frac{1}{\varepsilon}\right)$ under arbitrary heterogeneity!
Figure: Training gradient norm and testing accuracy against communication rounds for classification on the unshuffled MNIST dataset using a simple neural network. Both BEER and CHOCO-SGD employ the biased gsgd\textsubscript{b} compression with \( b = 20 \).
Key takeaways and further pointers

Key takeaways:
• Compression can greatly improve communication efficiency without hurting performance
• Compressing the error, not the gradient
• Accelerating decentralized optimization via gradient tracking

Further pointers:
• Biased versus unbiased compression
• Uplink and downlink compression
Part 3: Private federated optimization
Privacy guarantees are becoming increasingly critical!
Differential privacy (Dwork, 2006) is a popular approach for preserving privacy in practice, and widely adopted by Google, Apple, US Census, etc.

**Definition (Differential privacy (DP))**

A randomized mechanism $\mathcal{M} : \mathcal{Z} \rightarrow \mathcal{R}$ satisfies $(\epsilon, \delta)$-DP, if for any two neighboring dataset $\mathcal{Z}, \mathcal{Z}_i \in \mathcal{Z}$ and any outputs $\mathcal{R} \subseteq \mathcal{R}$, it holds that

$$P(\mathcal{M}(\mathcal{Z}) \in \mathcal{R}) \leq e^\epsilon P(\mathcal{M}(\mathcal{Z}_i) \in \mathcal{R}) + \delta.$$ 

The neighboring datasets are defined as $\mathcal{Z} = \{z_1, \ldots, z_n\}$ and $\mathcal{Z}_i = \{z_1, \ldots, z'_i, \ldots, z_n\}$, which means $\mathcal{Z}$ and $\mathcal{Z}_i$ are only different at one sample.

- Probabilistic definition: making it hard to tell if a data sample is used or not.
- Suitable to protect the privacy of individual records (cross-silo).
Gaussian mechanism: add noise to each sample gradient:

\[ g_{\text{DP}}(x_t; z) \leftarrow \nabla \ell(x_t; z) + w_t, \]

where \( w_t \sim \mathcal{N}(0, \sigma_{\text{DP}}^2 I) \).

- The noise level \( \sigma_{\text{DP}} \) depends on the size of \( \nabla \ell(x_t; z) \) → requiring bounded gradient assumption.

- or, clip the gradient before adding the noise

\[ g_{\text{DP}}(x_t; z) \leftarrow \text{Clip}_\tau(\nabla \ell(x_t; z)) + w_t \]

→ harder to analyze due to clipping!
A baseline: single-machine DP-SGD

Differentially private SGD (Abadi et al., 2016) in a single-machine setting:

\[ g_{\text{DP}}(x_t; z) \leftarrow \text{Clip}_\tau(\nabla \ell(x_t; z)) + w_t \]

**Theorem (Abadi et al., 2016)**

Assume the bounded gradient assumption holds. DP-SGD achieves \((\epsilon, \delta)\)-DP, and the utility

\[
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \| \nabla f(x^t) \|^2_2 \lesssim \frac{\sqrt{d \log(1/\delta)}}{m \epsilon} =: \phi_m
\]

within \( T \approx \frac{m \epsilon}{\sqrt{d \log(1/\delta)}} = \phi_m^{-1} \) rounds.

- Base utility \( \phi_m = \frac{\sqrt{d \log(1/\delta)}}{m \epsilon} \): lower is better.
- Stronger privacy, worse utility (accuracy), less communication.
- \( \sigma_{\text{DP}} \approx \frac{G \phi_m \sqrt{T}}{d} \), \( G \) is the gradient norm: add more noise when running the algorithm longer.
Local differential privacy (McMahan et al., 2018) protect from leaking info to other agents.

**Definition (Local differential privacy (LDP))**

A randomized mechanism $\mathcal{M} : \mathcal{Z} \rightarrow \mathcal{R}$ satisfies $(\epsilon, \delta)$-LDP for client $i$, if for any two neighboring dataset $\mathcal{Z}, \mathcal{Z}_i \in \mathcal{Z}$ and any outputs $\mathcal{R} \subseteq \mathcal{R}$, it holds that

$$\Pr(\mathcal{M}(\mathcal{Z}) \in \mathcal{R}) \leq e^\epsilon \Pr(\mathcal{M}(\mathcal{Z}_i) \in \mathcal{R}) + \delta.$$

The neighboring datasets are defined as $\mathcal{Z} = \{z_1, \ldots, z_n\}$ and $\mathcal{Z}_i = \{z_1, \ldots, z'_i, \ldots, z_n\}$, which means $\mathcal{Z}$ and $\mathcal{Z}_i$ are only different at agent $i$. 
Protecting privacy via Gaussian mechanism

Introducing local differential privacy to guarantee the client privacy

— used by Google, Apple, etc in products
Warm-up: a direct compression approach (CDP-SGD)

Theorem (Li et al., NeurIPS 2022)

Assume the bounded gradient assumption holds. CDP-SGD achieves $(\epsilon, \delta)$-LDP, and the utility

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \| \nabla f(x_t) \|^2_2 \lesssim \frac{1}{\sqrt{\alpha n}} \cdot \phi_m,$$

within communication complexity on the order of

$$dn^{3/2} \alpha^{3/2} \phi_m^{-1} + \alpha n d \phi_m^{-2}.$$ 

• Larger $\phi_m = \sqrt{d \log(1/\delta) / m \epsilon}$ gives stronger privacy, worse accuracy, fewer communication.

• Caveat: the communication complexity is $O(\phi_m^{-2})$ when the local data size $m$ is dominating.
Better compression and compute: a unified framework?

- **Compression**: shift compression with many options, e.g. sparsification or quantization

- **Computation**: stochastic local gradient estimators with many options, e.g. SGD, SVRG or SAGA

Can we develop a unified framework for private FL with compression, with a characterization of the privacy-utility-communication trade-off?
SoteriaFL: a unified framework for compressed private FL

Highlights of SoteriaFL:

- Flexible local gradient estimators
- Protect local data privacy
- State-of-the-art shift compression scheme
- Privacy-utility-communication trade-offs
Theorem (Li et al., NeurIPS 2022)

Assume the bounded gradient assumption holds. When $n \geq 1/\alpha^3$, SoteriaFL—with SGD, GD, SVRG, SAGA—achieves $(\epsilon, \delta)$-LDP, and the utility

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \| \nabla f(x^t) \|_2^2 \lesssim \frac{1}{\sqrt{\alpha n}} \cdot \phi_m$$

with communication complexity on the order of

$$dn^{3/2} \alpha^{3/2} \phi_m^{-1}.$$ 

- Communication complexity is linear in $\phi_m^{-1}$, better than CDP-SGD!
- This analysis applies to unbiased compressions, and adapts to other gradient estimators too.
Privacy-utility-communication trade-off

- Stronger privacy, worse accuracy, fewer communication
- More compression, worse accuracy, fewer communication
Numerical experiments

Compression preserves privacy at a better communication complexity.

Figure: Shallow NN training on the MNIST dataset under $(1, 10^{-3})$-LDP.
Introducing local differential privacy in BEER to guarantee client privacy
PORTER: BEER meets differential privacy

Theorem (Li and Chi, 2023)

Assuming bounded gradient assumption holds. PORTER achieves \((\epsilon, \delta)\)-LDP, and the utility

\[
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \| \nabla f(x_t) \|_2^2 \lesssim \frac{1}{(1 - \alpha)^{8/3} \rho^{4/3}} \cdot \phi_m
\]

within communication complexity on the order of \(\phi_m^{-2}\). Here, \(\alpha\) is the compression ratio, \(\beta\) is the spectral gap of the network.

- Captures the trade-off with network connectivity.
- Communication complexity degenerates to \(\phi_m^{-2}\), due to dealing with the decentralized setting.
Definition (Clipping operator)

\[ \text{Clip}_\tau(x) = \frac{\tau}{\tau + \|x\|_2} x \]

- The norm of a clipped vector is bounded by \( \tau \), i.e. \( \|\text{Clip}_\tau(x)\|_2 \leq \tau \).
- Can also use a hard thresholding operator for clipping.
Clipping is widely used in practice

**One stone, two birds:** clipping is widely used for two reasons (and they differ when using mini-batches).

- Privacy-preserving via per-sample clipping:

  \[
  g_{DP}(x_t; z) \leftarrow \frac{1}{|I_t|} \sum_{z \in I_t} \text{Clip}_\tau(\nabla \ell(x_t; z)) + w_t
  \]

- Stabilize training via per-batch clipping:

  \[
  g_{GC}(x_t; z) \leftarrow \text{Clip}_\tau \left( \frac{1}{|I_t|} \sum_{z \in I_t} \nabla \ell(x_t; z) \right)
  \]

How does clipping impact the performance of federated optimization?

*Let's take a detour to understand clipping!*
Understanding gradient clipping with batch gradient

Clipping only impacts the size of the gradient, but not the direction.

\[ x_{t+1} = x_{t+1} - \eta_t \text{Clip}_\tau (\nabla f(x_t)) \]

- Define \( \delta_t = \frac{\tau}{\tau + \|\nabla f(x_t)\|_2} \).

\[
f(x_{t+1}) - f(x_t) = f(x_t - \eta_t \text{Clip}_\tau (\nabla f(x_t))) - f(x_t)
\leq \langle \nabla f(x_t), -\eta_t \text{Clip}_\tau (\nabla f(x_t)) \rangle + \frac{L}{2} \|\eta_t \text{Clip}_\tau (\nabla f(x_t))\|_2^2
\]

\[
= -\eta_t \delta_t \langle \nabla f(x_t), \nabla f(x_t) \rangle + \frac{\eta_t^2 \delta_t^2 L}{2} \|\nabla f(x_t)\|_2^2
\]

\[
= - \left( \eta_t \delta_t - \frac{\eta_t^2 \delta_t^2 L}{2} \right) \|\nabla f(x_t)\|_2^2
\]

\[
\leq -\frac{\eta_t \delta_t}{2} \|\nabla f(x_t)\|_2^2
\]

as long as \( \eta_t \delta_t < 1/L \).
Gradient clipping: a contradiction argument

When \( \| \nabla f(x_t) \|_2 \geq \nu \),

\[
\frac{\delta_t}{2} \| \nabla f(x_t) \|_2^2 = \frac{1}{2} \frac{\tau \| \nabla f(x_t) \|_2^2}{\tau + \| \nabla f(x_t) \|_2^2} \geq \frac{\tau}{\tau + \nu} \cdot \frac{\nu^2}{2} \geq \frac{\tau}{\max\{\tau, \nu\}} \cdot \frac{\nu^2}{4}
\]

where (i) holds since \( h(x) = \frac{x^2}{c+x} \) is convex and increases monotonically when \( x \geq 0 \). Then, choose any \( \tau \geq \nu \), the function value decrease can be bounded by

\[
f(x_{t+1}) - f(x_t) \leq -\frac{\nu^2}{4L},
\]

which can not decrease for more than \( T = O\left(\frac{L \Delta}{\nu^2}\right) \) iterations.
Gradient clipping in the decentralized setting

- Let’s consider a toy example:
  - Number of agents $n = 3$, problem dimension $d = 1$
  - Local models are $x_1 = x_2 = x_3 = x^*$
  - Local gradients are $g_1 = 8, g_2 = -2, g_3 = -6$

- The global gradient is

$$g = \frac{1}{3}(g_1 + g_2 + g_3) = 0.$$  

- Apply $\text{Clip}_2(\cdot)$, the global gradient becomes

$$g' = \frac{1}{3}(\text{Clip}_2(g_1) + \text{Clip}_2(g_2) + \text{Clip}_2(g_3)) = \frac{1}{3}(1.6 - 1 - 1.25) = -0.22.$$  

**Definition (Bounded dissimilarity)**

The local and global objectives satisfy the following:

$$\|\nabla f_i(x) - \nabla f(x)\|_2 \leq \frac{1}{12}\|\nabla f(x)\|_2.$$
PORTER with per-batch clipping

**Theorem (Li and Chi, 2023)**

Assuming bounded local gradient variance and bounded dissimilarity assumptions hold. PORTER with gradient clipping achieves

\[
\min_{t \in [T]} \mathbb{E} \left\| \nabla f(x_t) \right\|_2 \lesssim \frac{1}{(1 - \alpha)^{4/3} \rho^{2/3}} \cdot \frac{1}{T^{1/2}}
\]

under appropriate parameter choices and large enough batch size.

- Matches the rate \( O(1/T^{1/2}) \) of centralized SGD as long as the mini-batch size is large enough and the local datasets are not too dissimilar.

- First convergence guarantee of decentralized optimization with gradient clipping and communication compression.
PORTER with per-sample clipping

Theorem (Li and Chi, 2023)
Assuming bounded local gradient variance and bounded dissimilarity assumptions hold. PORTER achieves $(\epsilon, \delta)$-LDP, and the utility

$$\min_{t \in [T]} \mathbb{E} \| \nabla f(x_t) \|_2^2 \lesssim \frac{1}{(1 - \alpha)^{8/3} \rho^{4/3}} \cdot \phi_m^{1/2}$$

within communication rounds $\phi_m^{-2}$.

- Dependencies on mixing rate and compression match previous results.
Numerical experiments

**Figure:** Shallow NN training on the MNIST dataset under $(10^{-2}, 10^{-3})$-LDP. Both PORTER and SoteriaFL-SGD employ random compression.
Concluding remarks
Federated optimization: let’s make it efficient, resilient and private!
Key algorithmic pillars and trade-offs

It’s all about trade-offs:

- Computation
- Communication
- Privacy
- Performance

Algorithmic ideas to probe the trade-offs:

- Local updates
- Compression
- Variance reduction
- Error feedback
- Gradient tracking
- Differential privacy
- ...
The end? Or just the beginning?
Robustness to adversary

adversarial client

Man-in-the-middle

Robust algorithms that are oblivious to adversarial clients/attack?
Asynchronous updates

Synchronous update

Asynchronous update

Credit: (Huba et al., 2022)

Asynchronous updates to the rescue!
Credit: (Arivazhagan et al., 2019)

Shared the representation, personalize the prediction
How to design efficient algorithms for feature-distributed data?
Semi-decentralized topology

Can we combining the best of worlds?
Federated reinforcement learning: enables multiple agents to collaboratively learn a global model without sharing datasets.
Disclaimer: this straw-man list is by no means exhaustive (in fact, it is quite the opposite given the fast pace of the field), and biased towards materials most related to this tutorial; readers are invited to further delve into the references therein to gain a more complete picture.

Monographs:


Primer on nonconvex optimization:


**Local methods:**


Communication compression:

**Differentially-private federated optimization:**


Thank you!

https://users.ece.cmu.edu/~yuejie