Optimization in the Federated Setting

Virginia Smith
Carnegie Mellon University
ML workflow

- data & problem
- ML model
- optimization algorithm

\[ \min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, x_i) + g(\mathbf{w}) \]
system-aware optimization

\[
\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, x_i) + g(\mathbf{w})
\]
federated optimization

*training machine learning models at the edge*

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

raw data

cloud-based training

model

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

cloud-based training

raw data

predictions

federated training

model updates

global model

raw data

predictions

[McMahan et al., AISTATS 16]
federated optimization

training machine learning models at the edge

why? ✓ quickly incorporate new data ✓ reduce strain on network ✓ privacy

[McMahan et al., AISTATS 16]
federated optimization

example applications
federated optimization

example applications

language modeling for voice recognition on mobile phones
federated optimization

example applications

language modeling for voice recognition on mobile phones

adapting to pedestrian behavior in autonomous vehicles
federated optimization

**example applications**

- language modeling for voice recognition on mobile phones
- adapting to pedestrian behavior in autonomous vehicles
- predicting low blood sugar via wearable devices
federated optimization

example applications

- language modeling for voice recognition on mobile phones
- adapting to pedestrian behavior in autonomous vehicles
- predicting low blood sugar via wearable devices

assumptions? ✓ local data is important ✓ labels available ✓ privacy is a concern
last 10+ years: distributed optimization in the data center
beyond the data center: challenges in federated optimization
beyond the data center: challenges in federated optimization

communication is expensive
beyond the data center: **challenges in federated optimization**

**communication is expensive**

- millions of devices, slower networks
beyond the data center: challenges in federated optimization

communication is expensive
  ➢ millions of devices, slower networks

performance is highly variable
beyond the data center: challenges in federated optimization

**communication is expensive**
- millions of devices, slower networks

**performance is highly variable**
- systems: device heterogeneity, network unreliability, fault tolerance
beyond the data center: challenges in federated optimization

communication is expensive
▷ millions of devices, slower networks

performance is highly variable
▷ systems: device heterogeneity, network unreliability, fault tolerance
▷ statistical: unbalanced data
beyond the data center: challenges in federated optimization

communication is expensive
- millions of devices, slower networks

performance is highly variable
- systems: device heterogeneity, network unreliability, fault tolerance
- statistical: unbalanced data

personalization is key
beyond the data center: **challenges in federated optimization**

*communication is expensive*
- millions of devices, slower networks

*performance is highly variable*
- systems: device heterogeneity, network unreliability, fault tolerance
- statistical: unbalanced data

*personalization is key*
- non-IID data, underlying statistical structure
beyond the data center: challenges in federated optimization

communication is expensive
- millions of devices, slower networks

performance is highly variable
- systems: device heterogeneity, network unreliability, fault tolerance
- statistical: unbalanced data

personalization is key
- non-IID data, underlying statistical structure
FedCoCoA: communication-efficient federated optimization

[Smith, Chiang, Sanjabi, Talwalkar, NIPS 2017]
[Smith, Forte, Ma, Takac, Jordan, Jaggi, JMLR 2018]
first approach: data center optimization methods
first approach: data center optimization methods

data center setting

![Graph](image-url)
first approach: data center optimization methods
first approach: data center optimization methods

data center setting
first approach: data center optimization methods

data center setting

distance to optimal

time
first approach: data center optimization methods

data center setting

distance to optimal vs. time for different methods: CoCoA, Mb-CD, Mb-SGD, GD
first approach: data center optimization methods

data center setting

40x speedup
first approach: data center optimization methods

data center setting

40x speedup
2 hours → 3 minutes
first approach: data center optimization methods

✓ data center setting

40x speedup

2 hours → 3 minutes
first approach: data center optimization methods

✅ data center setting

X federated setting

40x speedup

2 hours → 3 minutes
first approach: data center optimization methods

✓ data center setting

2 hours → 3 minutes

40x speedup

✗ federated setting
prior work: distributed data center optimization
prior work: distributed data center optimization
prior work: distributed data center optimization
prior work: distributed data center optimization
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( w = w + \sum_k \Delta w_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce:

$$w = w + \sum_k \Delta w_k$$

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce:

\[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

Reduce:

\[ w = w + \sum_k \Delta w_k \]

✓ convergence guarantees
✗ slow

"always communicate"
prior work: distributed data center optimization

✓ convergence guarantees
✗ slow

reduce: \( w = w + \sum_k \Delta w_k \)

“always communicate”
prior work: distributed data center optimization

reduce: \( w = w + \sum_k \Delta w_k \)
prior work: distributed data center optimization

reduce: $\mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k$

✓ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( w = w + \sum_k \Delta w_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

"always communicate"
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✓ convergence guarantees
✗ slow

"always communicate"
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

(always communicate)
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✓ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

\[ w = w + \sum_k \Delta w_k \]

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \)

✔ convergence guarantees
✗ slow

“always communicate”
prior work: distributed data center optimization

reduce: \( \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \)

average: \( \mathbf{w} := \frac{1}{K} \sum_k \mathbf{w}_k \)

✔ convergence guarantees
✗ slow

"always communicate"
prior work: distributed data center optimization

reduce: \( w = w + \sum_k \Delta w_k \)

average: \( w := \frac{1}{K} \sum_k w_k \)

✔ convergence guarantees
✗ slow

✔ fast
✗ convergence not guaranteed

“always communicate”

“never communicate”
CoCoA: communication-efficient distributed optimization

mini-batch methods

one-shot communication
CoCoA: communication-efficient distributed optimization

key idea: control communication
CoCoA: subproblems
CoCoA: subproblems
CoCoA: subproblems
CoCoA: subproblems
CoCoA: subproblems

W

W_1 W_2 W_3 \ldots W_K
CoCoA: subproblems

$W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow \ldots \rightarrow W_K$
CoCoA: subproblems

\[ w = w + \sum_k \Delta w_k \]
CoCoA: subproblems

\[ w = w + \sum_{k} \Delta w_k \]
CoCoA: communication parameter
Main assumption:

Each subproblem is solved to accuracy $\Theta$.
CoCoA: communication parameter

Main assumption:

*each subproblem is solved to accuracy $\Theta$*
CoCoA: communication parameter

Main assumption:

Each subproblem is solved to accuracy $\Theta$

$\Theta \in [0, 1)$
Main assumption:
each subproblem is solved to accuracy $\Theta$

$\Theta \in [0, 1)$
Main assumption: each subproblem is solved to accuracy $\Theta$

$$\Theta \in [0, 1)$$

- exactly solve
- inexactley solve
CoCoA: communication parameter

Main assumption:
*each subproblem is solved to accuracy* \( \Theta \)

\[ \Theta \in [0, 1) \approx \text{amount of local computation vs. communication} \]
CoCoA: communication parameter

Main assumption:

challenge: make communication more flexible

exactly solve

inexactly solve
FedCoCoA: per-device, per-iteration approximations
FedCoCoA: per-device, per-iteration approximations

Old assumption: each subproblem is solved to accuracy $\theta \in [0, 1)$
FedCoCoA: per-device, per-iteration approximations

Old assumption:
each subproblem is solved to accuracy $\theta \in [0, 1)$

**Stragglers (Statistical heterogeneity)**
- Difficulty of solving subproblem
- Size of local dataset
FedCoCoA: per-device, per-iteration approximations

Old assumption: each subproblem is solved to accuracy $\theta \in [0, 1)$

Stragglers (Statistical heterogeneity)
- Difficulty of solving subproblem
- Size of local dataset

Stragglers (Systems heterogeneity)
- Hardware (CPU, memory)
- Network connection (3G, LTE, …)
- Power (battery level)
FedCoCoA: per-device, per-iteration approximations

Old assumption: each subproblem is solved to accuracy $\theta \in [0, 1)$

Stragglers (Statistical heterogeneity)
- Difficulty of solving subproblem
- Size of local dataset

Stragglers (Systems heterogeneity)
- Hardware (CPU, memory)
- Network connection (3G, LTE, …)
- Power (battery level)

Fault tolerance
- Devices going offline
FedCoCoA: per-device, per-iteration approximations

New assumption: each subproblem is solved to accuracy

\[ \theta^h_t \in [0, 1] \]
\[ \theta \in [0, 1) \]

Stragglers (Statistical heterogeneity)
- Difficulty of solving subproblem
- Size of local dataset

Stragglers (Systems heterogeneity)
- Hardware (CPU, memory)
- Network connection (3G, LTE, …)
- Power (battery level)

Fault tolerance
- Devices going offline
convergence
convergence

New assumption: each subproblem is solved to accuracy $\theta^h_t$
convergence

New assumption:

each subproblem is solved to accuracy $\theta_t^h$

and assume: $\mathbb{P}[\theta_t^h := 1] < 1$
convergence

New assumption:
each subproblem is solved to accuracy $\theta^h_t$

and assume: $\mathbb{P}[\theta^h_t := 1] < 1$

Theorem 1. Let $\ell_t$ be $L$-Lipschitz, then

$$T \geq \frac{1}{(1-\Theta)} \left( \frac{8L^2n^2}{\epsilon} + \tilde{c} \right)$$
convergence

New assumption: each subproblem is solved to accuracy $\theta_t^h$

and assume: $\mathbb{P}[\theta_t^h := 1] < 1$

Theorem 1. Let $\ell_t$ be $L$-Lipschitz, then

$$T \geq \frac{1}{(1 - \Theta)} \left( \frac{8L^2n^2}{\epsilon} + \tilde{c} \right)$$

1/$\epsilon$ rate
New assumption: each subproblem is solved to accuracy $\theta_t^h$

and assume: $\mathbb{P}[\theta_t^h := 1] < 1$

**Theorem 1.** Let $\ell_t$ be $L$-Lipschitz, then

$$T \geq \frac{1}{(1-\Theta)} \left( \frac{8L^2n^2}{\epsilon} + \tilde{c} \right)$$

**Theorem 2.** Let $\ell_t$ be $(1/\mu)$-smooth, then

$$T \geq \frac{1}{(1-\Theta)} \frac{\mu + n}{\mu} \log \frac{n}{\epsilon}$$

$1/\epsilon$ rate
convergence

New assumption:
each subproblem is solved to accuracy $\theta_t^h$

and assume: $\mathbb{P}[\theta_t^h := 1] < 1$

Theorem 1. Let $\ell_t$ be $L$-Lipschitz, then

$$T \geq \frac{1}{(1 - \Theta)} \left( \frac{8L^2n^2}{\epsilon} + \tilde{c} \right)$$

1/\epsilon rate

Theorem 2. Let $\ell_t$ be $(1/\mu)$-smooth, then

$$T \geq \frac{1}{(1 - \Theta)} \frac{\mu + n}{\mu} \log \frac{n}{\epsilon}$$

linear rate
<table>
<thead>
<tr>
<th></th>
<th>devices</th>
<th>features</th>
<th>min $n_t$</th>
<th>max $n_t$</th>
<th>std dev</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human Activity</strong></td>
<td>30</td>
<td>561</td>
<td>210</td>
<td>306</td>
<td>26.75</td>
</tr>
<tr>
<td><strong>Google Glass</strong></td>
<td>38</td>
<td>180</td>
<td>524</td>
<td>581</td>
<td>11.07</td>
</tr>
<tr>
<td><strong>Land Mine</strong></td>
<td>29</td>
<td>9</td>
<td>333</td>
<td>517</td>
<td>65.39</td>
</tr>
<tr>
<td><strong>Vehicle Sensor</strong></td>
<td>23</td>
<td>100</td>
<td>872</td>
<td>1,933</td>
<td>267.47</td>
</tr>
</tbody>
</table>
### federated datasets

<table>
<thead>
<tr>
<th></th>
<th>devices</th>
<th>features</th>
<th>$\text{min } n_t$</th>
<th>$\text{max } n_t$</th>
<th>std dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Activity</td>
<td>30</td>
<td>561</td>
<td>210</td>
<td>306</td>
<td>26.75</td>
</tr>
<tr>
<td>Google Glass</td>
<td>38</td>
<td>180</td>
<td>524</td>
<td>581</td>
<td>11.07</td>
</tr>
<tr>
<td>Land Mine</td>
<td>29</td>
<td>9</td>
<td>333</td>
<td>517</td>
<td>65.39</td>
</tr>
<tr>
<td>Vehicle Sensor</td>
<td>23</td>
<td>100</td>
<td>872</td>
<td>1,933</td>
<td>267.47</td>
</tr>
</tbody>
</table>
simulations
FedCoCoA converges two orders of magnitude more quickly than all competitors.
FedCoCoA converges two orders of magnitude more quickly than all competitors and is robust to dropped nodes.
MOCHA: personalization

[Smith, Chiang, Sanjabi, Talwalkar, NIPS 2017]
system-aware optimization

\[
\min_w \sum_{i=1}^{n} \ell(w, x_i) + g(w)
\]
how to model federated data?
how to model federated data?

local

W_1

W_2

W_3

W_4

W_5

W_6

W_7

W_8
how to model federated data?

- local

- ✓ personalized models
- ✗ don’t learn from peers
how to model federated data?

✓ personalized models
✗ don’t learn from peers
how to model federated data?

- **local**
  - ✓ personalized models
  - × don’t learn from peers

- **global**
  - × non-personalized models
  - ✓ learn from peers
how to model federated data?

- **local**
  - ✓ personalized models
  - ✗ don’t learn from peers

- **global**
  - ✗ non-personalized models
  - ✓ learn from peers

- **??**
how to model federated data?

local

- ✓ personalized models
- ✓ learn from peers
- ✓ personalized models
- ✓ learn from peers

non-personalized models

- × don’t learn from peers
- × non-personalized models

global

- ✓ learn from peers
- ✓ learn from peers

??

- ✓ personalized models
- ✓ learn from peers
multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + \mathcal{R}(W, \Omega)
\]
multi-task learning

$$\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)$$
multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)
\]
multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)
\]
multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + \mathcal{R}(W, \Omega)
\]
multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)
\]

all tasks related

multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + \mathcal{R}(W, \Omega)
\]

models

all tasks related

outlier tasks

multi-task learning

$$\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)$$

models

losses

regularizer

all tasks related

outlier tasks

clusters / groups

multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t, x^i_t) + R(W, \Omega)
\]

all tasks related
outlier tasks
clusters / groups
asymmetry

example: prediction error on federated data

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>Local</th>
<th>MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Activity</td>
<td>2.23</td>
<td>1.34</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>(0.30)</td>
<td>(0.21)</td>
<td>(0.11)</td>
</tr>
<tr>
<td>Google Glass</td>
<td>5.34</td>
<td>4.92</td>
<td>2.02</td>
</tr>
<tr>
<td></td>
<td>(0.26)</td>
<td>(0.26)</td>
<td>(0.15)</td>
</tr>
<tr>
<td>Land Mine</td>
<td>27.72</td>
<td>23.43</td>
<td>20.09</td>
</tr>
<tr>
<td></td>
<td>(1.08)</td>
<td>(0.77)</td>
<td>(1.04)</td>
</tr>
<tr>
<td>Vehicle Sensor</td>
<td>13.4</td>
<td>7.81</td>
<td>6.59</td>
</tr>
<tr>
<td></td>
<td>(0.26)</td>
<td>(0.13)</td>
<td>(0.21)</td>
</tr>
</tbody>
</table>
example: prediction error on federated data

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>Local</th>
<th>MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Activity</td>
<td>2.23</td>
<td>1.34</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>(0.30)</td>
<td>(0.21)</td>
<td>(0.11)</td>
</tr>
<tr>
<td>Google Glass</td>
<td>5.34</td>
<td>4.92</td>
<td>2.02</td>
</tr>
<tr>
<td></td>
<td>(0.26)</td>
<td>(0.26)</td>
<td>(0.15)</td>
</tr>
<tr>
<td>Land Mine</td>
<td>27.72</td>
<td>23.43</td>
<td>20.09</td>
</tr>
<tr>
<td></td>
<td>(1.08)</td>
<td>(0.77)</td>
<td>(1.04)</td>
</tr>
<tr>
<td>Vehicle Sensor</td>
<td>13.4</td>
<td>7.81</td>
<td>6.59</td>
</tr>
<tr>
<td></td>
<td>(0.26)</td>
<td>(0.13)</td>
<td>(0.21)</td>
</tr>
</tbody>
</table>
MOCHA: federated optimization for multi-task learning
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)
\]
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)
\]

Key idea: Solve for \( W, \Omega \) in an alternating fashion
MOCHA: federated optimization for multi-task learning

\[\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + \mathcal{R}(W, \Omega)\]

Key idea: Solve for \(W, \Omega\) in an alternating fashion

\(\Omega\) can be updated centrally
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)
\]

- Key idea: Solve for $W, \Omega$ in an alternating fashion
- $\Omega$ can be updated centrally
- $W$ needs to be solved in federated setting
MOCHA: federated optimization for multi-task learning

$$\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)$$

- Key idea: Solve for $W, \Omega$ in an alternating fashion
  - $\Omega$ can be updated centrally
  - $W$ needs to be solved in federated setting
- How to extend FedCoCoA to MTL?
FedCoCoA: subproblems

\[ \mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k \]
FedCoCoA: primal-dual framework
FedCoCoA: primal-dual framework
FedCoCoA: primal-dual framework

**PRIMAL**

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^T \mathbf{x}_i) + \lambda g(\mathbf{w})
\]
FedCoCoA: primal-dual framework

PRIMAL \geq \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i) + \lambda g(w)
FedCoCoA: primal-dual framework

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^T \mathbf{x}_i) + \lambda g(\mathbf{w})
\]
FedCoCoA: primal-dual framework

\[
\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i) + \lambda g(w) \quad \geq \quad \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \alpha)
\]
FedCoCoA: primal-dual framework

\[
\begin{align*}
\min_{w \in \mathbb{R}^d} & \quad \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i) + \lambda g(w) \\
\max_{\alpha \in \mathbb{R}^n} & \quad -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \alpha)
\end{align*}
\]
FedCoCoA: primal-dual framework

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^T \mathbf{x}_i) + \lambda g(\mathbf{w}) \quad \geq \quad \max_{\mathbf{\alpha} \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\mathbf{\alpha}_i) - \lambda g^*(\mathbf{X}, \mathbf{\alpha})
\]

\[
\sum_{k=1}^{K} \tilde{g}^*(X[k], \alpha[k])
\]
FedCoCoA: primal-dual framework

\[
\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i) + \lambda g(w) \quad \geq \quad \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \alpha)
\]
FedCoCoA: primal-dual framework

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^T \mathbf{x}_i) + \lambda g(\mathbf{w}) \quad \geq \quad \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \alpha)
\]

\[\alpha_k(t)\]
FedCoCoA: primal-dual framework

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^T \mathbf{x}_i) + \lambda g(\mathbf{w}) \quad \geq \quad \max_{\mathbf{\alpha} \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\mathbf{\alpha}_i) - \lambda g^*(\mathbf{X}, \mathbf{\alpha})
\]

\[\alpha_k(t) \quad \alpha_k(t+1)\]
FedCoCoA: primal-dual framework

\[
\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i) + \lambda g(w) \quad \geq \quad \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \alpha) \\
\sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \alpha_{[k]})
\]

\[ \alpha_k(t) \quad \alpha_k(t+1) \quad \alpha_k^* \]
FedCoCoA: primal-dual framework

challenge: extend to MTL setup
MOCHA: federated optimization for multi-task learning
MOCHA: federated optimization for multi-task learning

$$\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w^T_t x^i_t) + R(W, \Omega)$$
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w^T x_t^i) + R(W, \Omega)
\]

- Solve for \( W, \Omega \) in an alternating fashion
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)
\]

- Solve for \( W, \Omega \) in an alternating fashion
- Modify FedCoCoA to solve \( W \) in the federated setting
MOCHA: federated optimization for multi-task learning

\[ \min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega) \]

- Solve for \( W, \Omega \) in an alternating fashion
- Modify FedCoCoA to solve \( W \) in the federated setting

\[ \text{dual} \quad \min_{\alpha} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\alpha_t^i) + R^*(X\alpha) \]
MOCHA: federated optimization for multi-task learning

\[
\min_{W, \Omega} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(w_t^T x_t^i) + R(W, \Omega)
\]

- Solve for \(W, \Omega\) in an alternating fashion
- Modify FedCoCoA to solve \(W\) in the federated setting

\[
\text{dual} \quad \min_{\alpha} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\alpha_t^i) + R^*(X\alpha)
\]

\[
\text{subproblem} \quad \min_{\Delta \alpha_t} \sum_{i=1}^{n_t} \ell_t^*(-\alpha_t^i - \Delta \alpha_t^i) + \langle w_t(\alpha), X_t \Delta \alpha_t \rangle + \frac{\sigma'}{2} \|X_t \Delta \alpha_t\|_{M_t}^2
\]
beyond the data center: **challenges in federated optimization**

- **communication is expensive**
  - millions of devices, slower networks

- **performance is highly variable**
  - systems: device heterogeneity, network unreliability, fault tolerance
  - statistical: unbalanced data

- **personalization is key**
  - non-IID data, underlying statistical structure
what’s next?
additional challenges in federated learning
additional challenges in federated learning

non-convex optimization

deep learning
additional challenges in federated learning

non-convex optimization
  ➤ deep learning

privacy & security
  ➤ differential privacy
  ➤ secure aggregation
additional challenges in federated learning

- non-convex optimization
  - deep learning
- privacy & security
  - differential privacy
  - secure aggregation
- personalization
  - model fine-tuning
additional challenges in federated learning

- non-convex optimization
  - deep learning
- privacy & security
  - differential privacy
  - secure aggregation
- personalization
  - model fine-tuning

- reduced communication
  - compression
  - lazy aggregation
additional challenges in federated learning

- non-convex optimization
  - deep learning
- privacy & security
  - differential privacy
  - secure aggregation
- personalization
  - model fine-tuning
- reduced communication
  - compression
  - lazy aggregation
- ML pipeline
  - on-device feature extraction
  - on-device inference
LEAF: a benchmark for learning in the federated setting
LEAF: a benchmark for learning in the federated setting

suite of open-source datasets
LEAF: a benchmark for learning in the federated setting

- suite of open-source datasets
- evaluation framework with statistical and systems metrics to assess competing solutions
LEAF: a benchmark for learning in the federated setting

- suite of open-source datasets
- evaluation framework with statistical and systems metrics to assess competing solutions
- online presence to encourage participation and reproducibility
LEAF: a benchmark for learning in the federated setting

- suite of open-source datasets
- evaluation framework with statistical and systems metrics to assess competing solutions
- online presence to encourage participation and reproducibility

v1 coming soon!
SysML 2019

- New conference targeting research at the intersection of systems and machine learning
New conference targeting research at the intersection of systems and machine learning

Last year: 1-day conference, 300+ attendees, 200+ submissions
New conference targeting research at the intersection of systems and machine learning

Last year: 1-day conference, 300+ attendees, 200+ submissions

This year: 3 day conference with published proceedings
SysML 2019

- New conference targeting research at the intersection of systems and machine learning
- Last year: 1-day conference, 300+ attendees, 200+ submissions
- This year: 3 day conference with published proceedings

Dates: March 31—April 2, 2019
Location: Stanford, CA
Submissions due: September 28, 2018
Info & call for papers: www.sysml.cc
paper & code:

ece.cmu.edu/~smithv

Virginia Smith
Carnegie Mellon University
smithv@cmu.edu
Thank you!