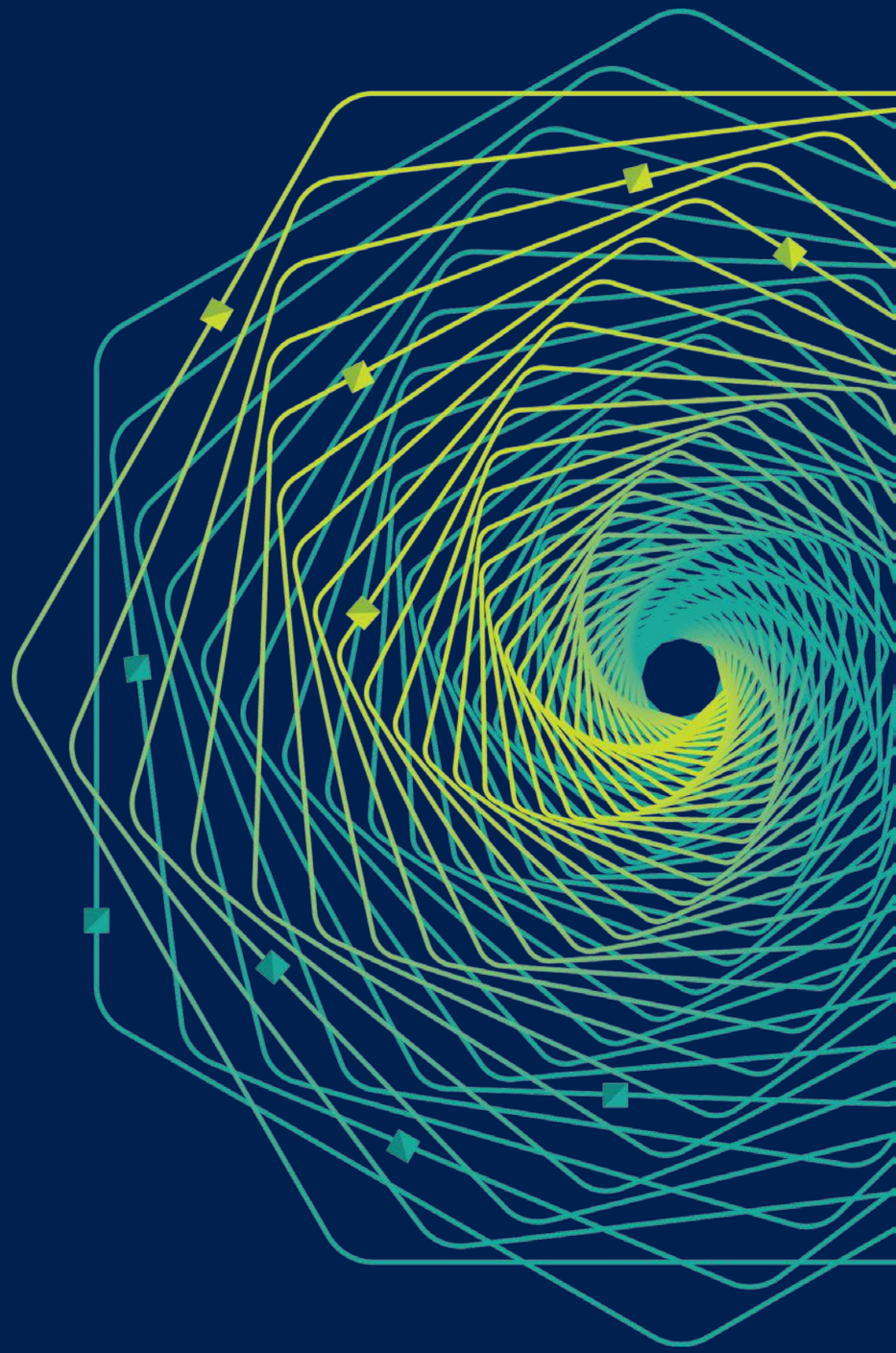
Microsoft

Research Faculty Summit 2018

Systems | Fueling future disruptions

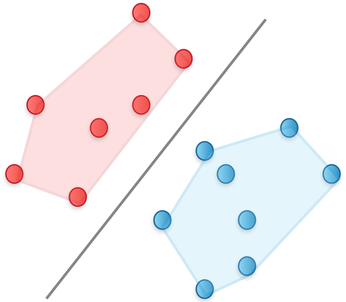


Optimization in the Federated Setting

Virginia Smith
Carnegie Mellon University

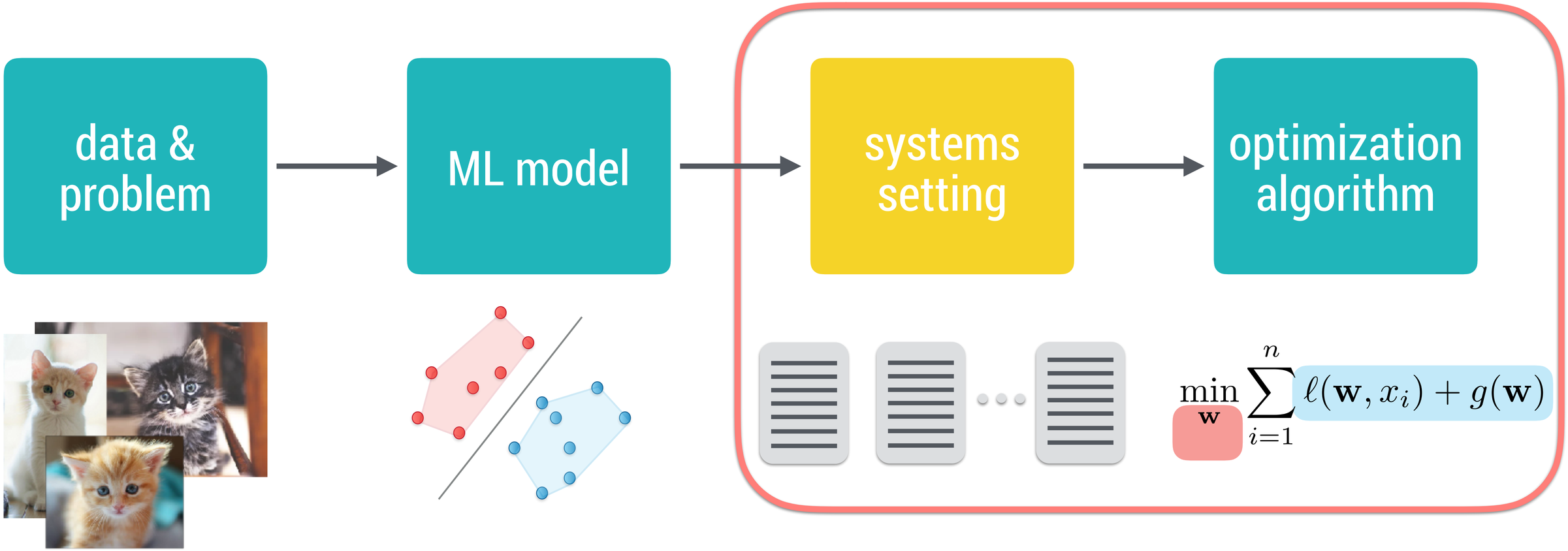
ML workflow

data & problem → ML model → optimization algorithm



$$\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, x_i) + g(\mathbf{w})$$

system-aware optimization



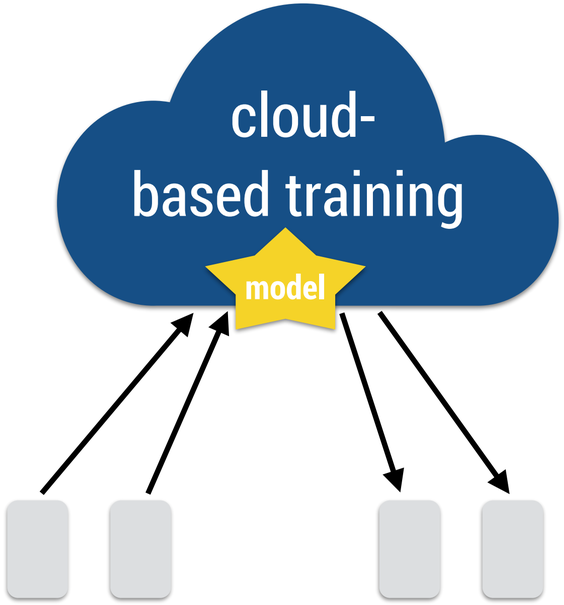
federated optimization

training machine learning models at the edge

[McMahan et al., AISTATS 16]

federated optimization

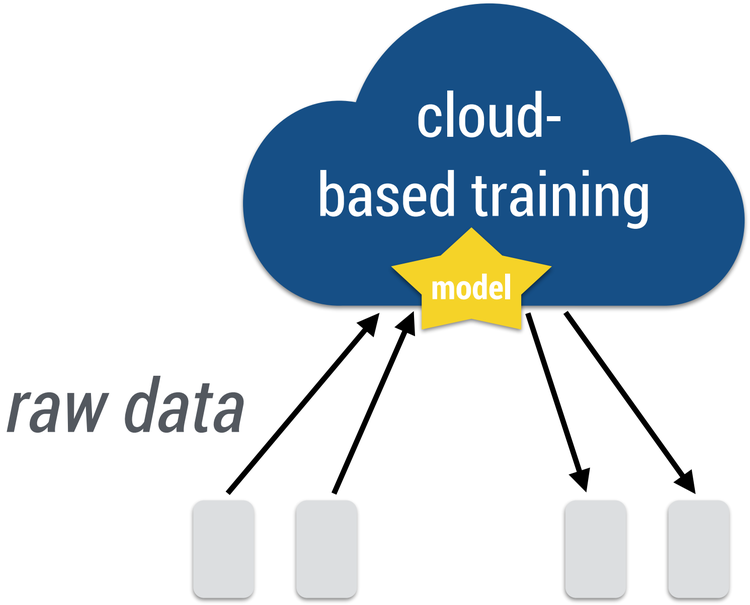
training machine learning models at the edge



[McMahan et al., AISTATS 16]

federated optimization

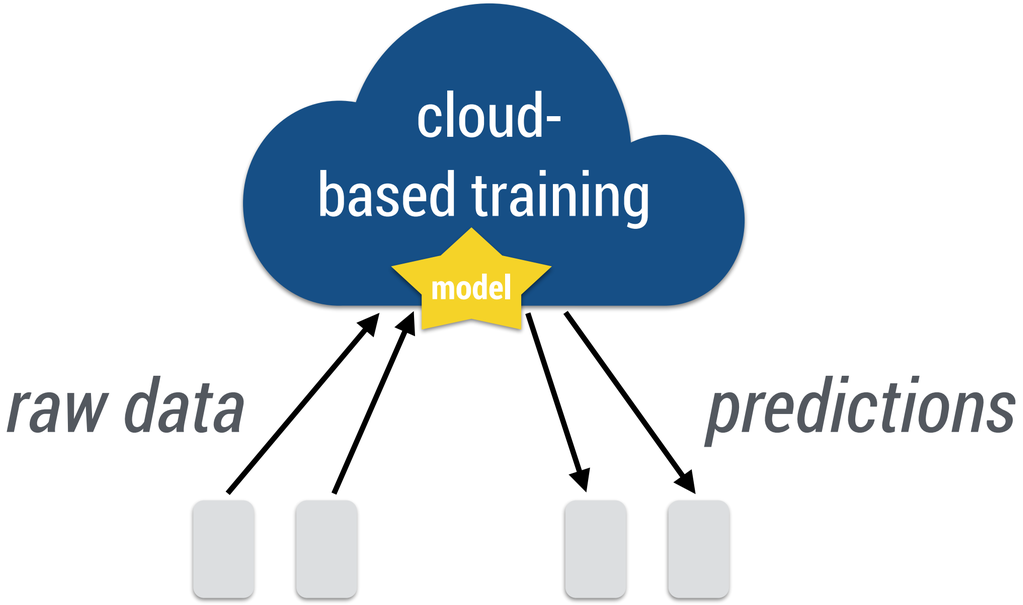
training machine learning models at the edge



[McMahan et al., AISTATS 16]

federated optimization

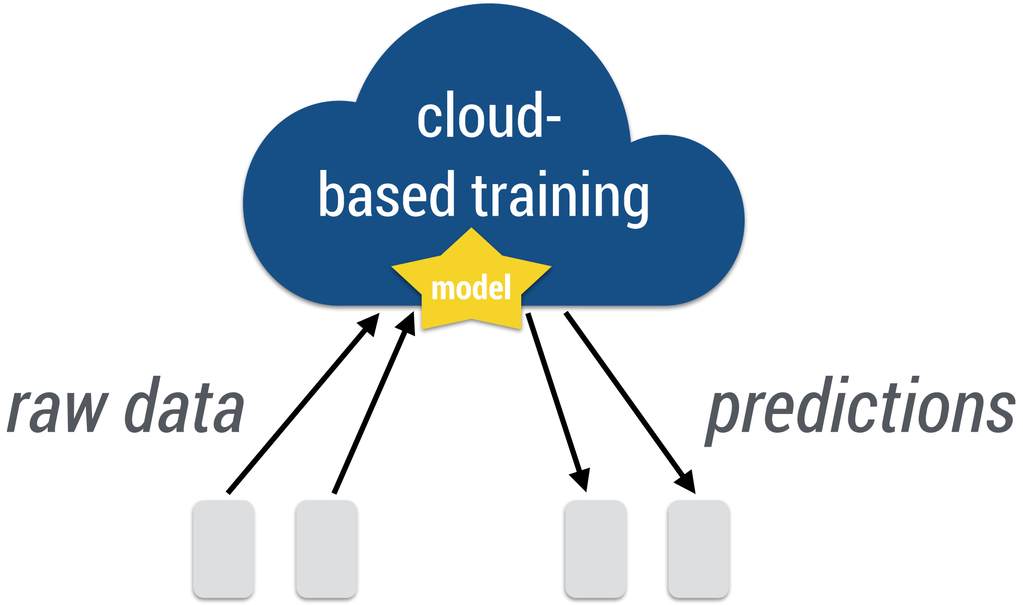
training machine learning models at the edge

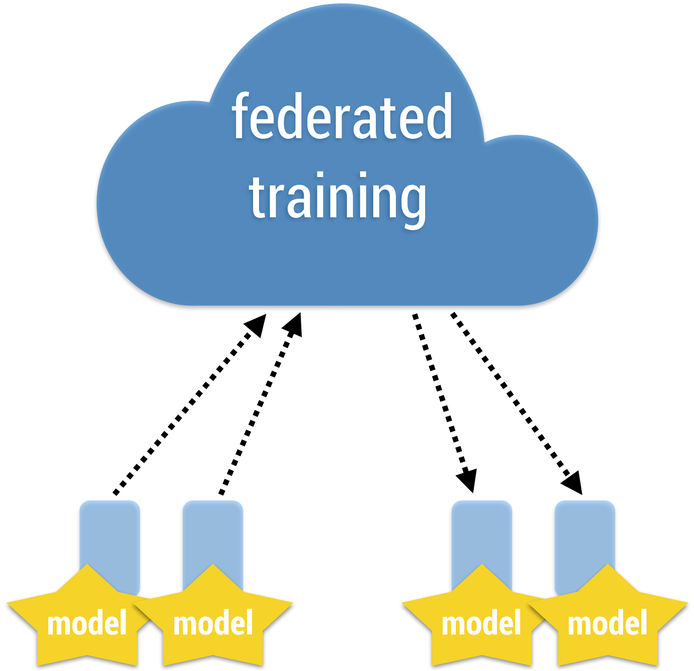


[McMahan et al., AISTATS 16]

federated optimization

training machine learning models at the edge

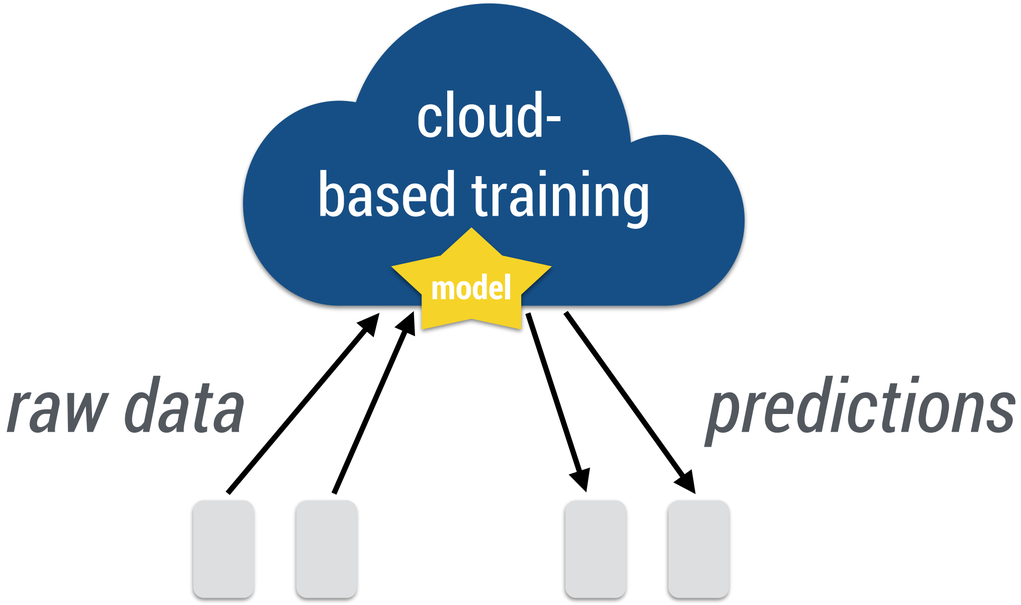


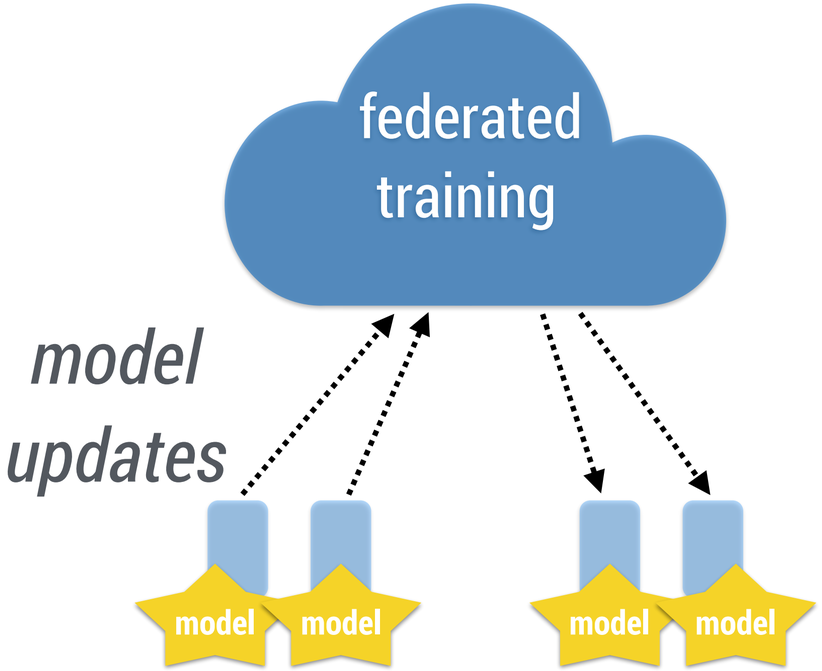


[McMahan et al., AISTATS 16]

federated optimization

training machine learning models at the edge

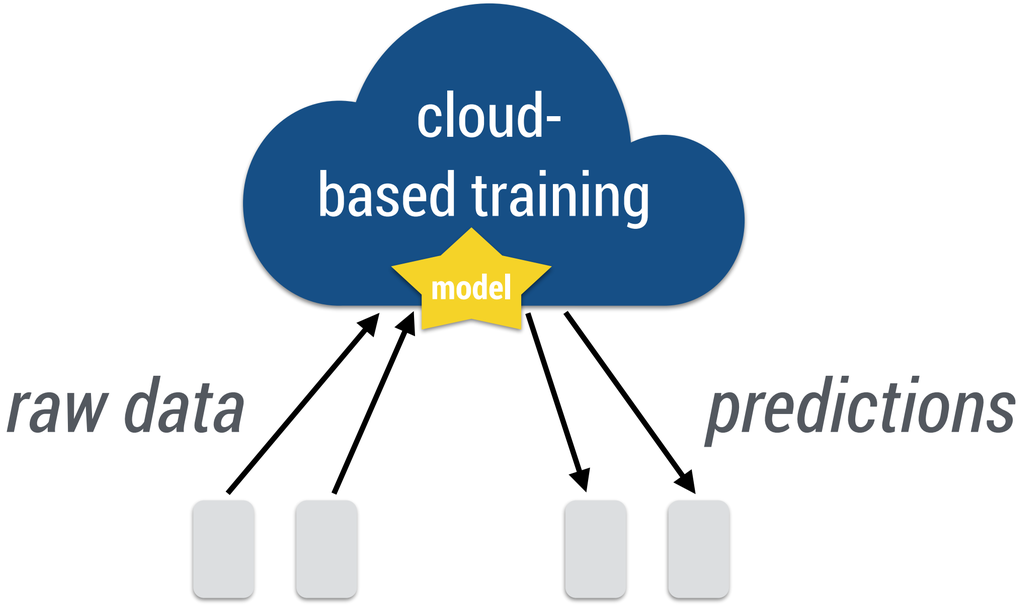


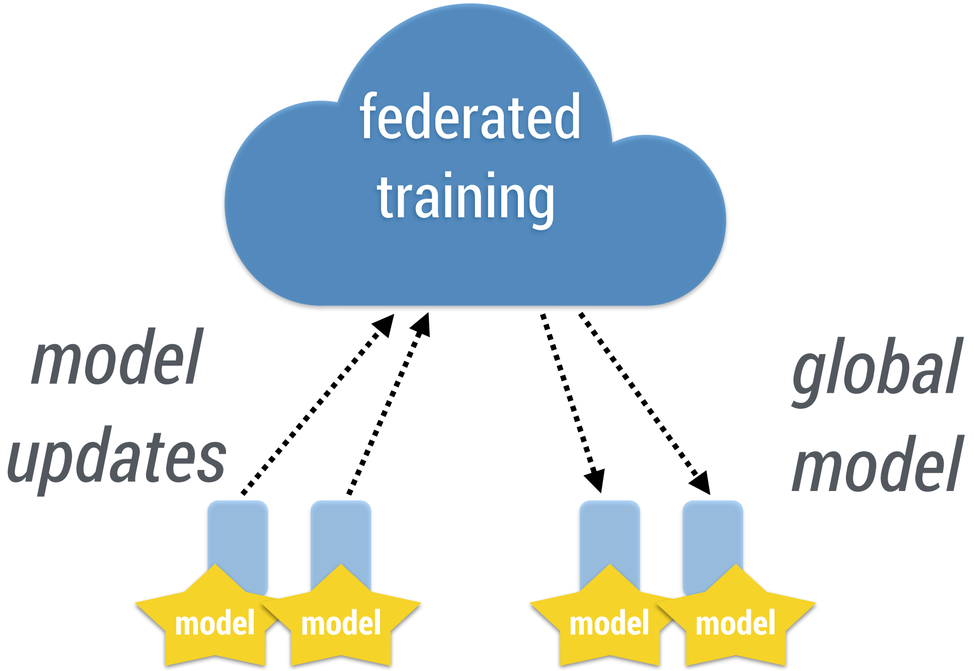


[McMahan et al., AISTATS 16]

federated optimization

training machine learning models at the edge

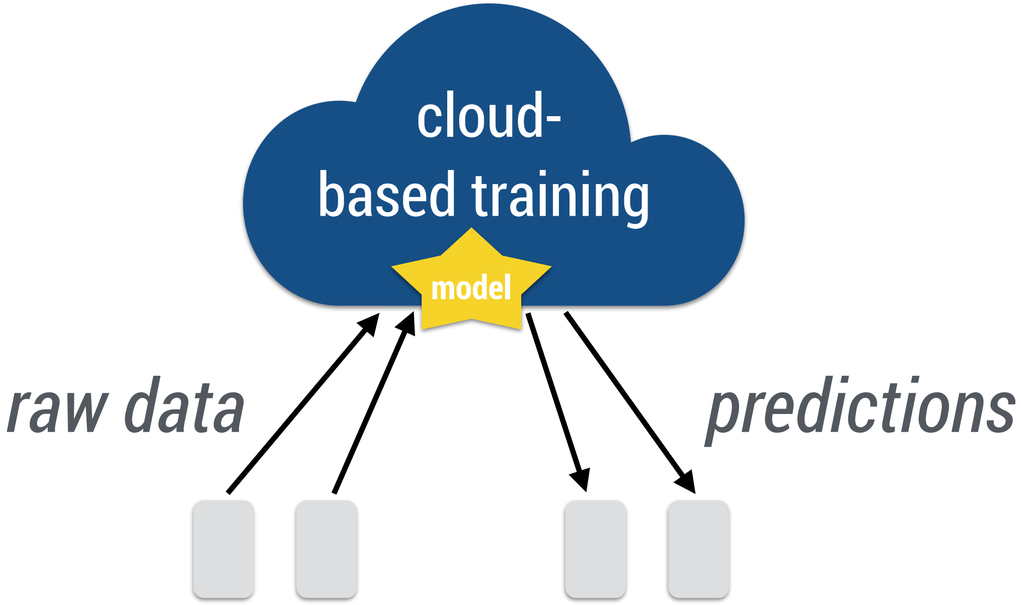


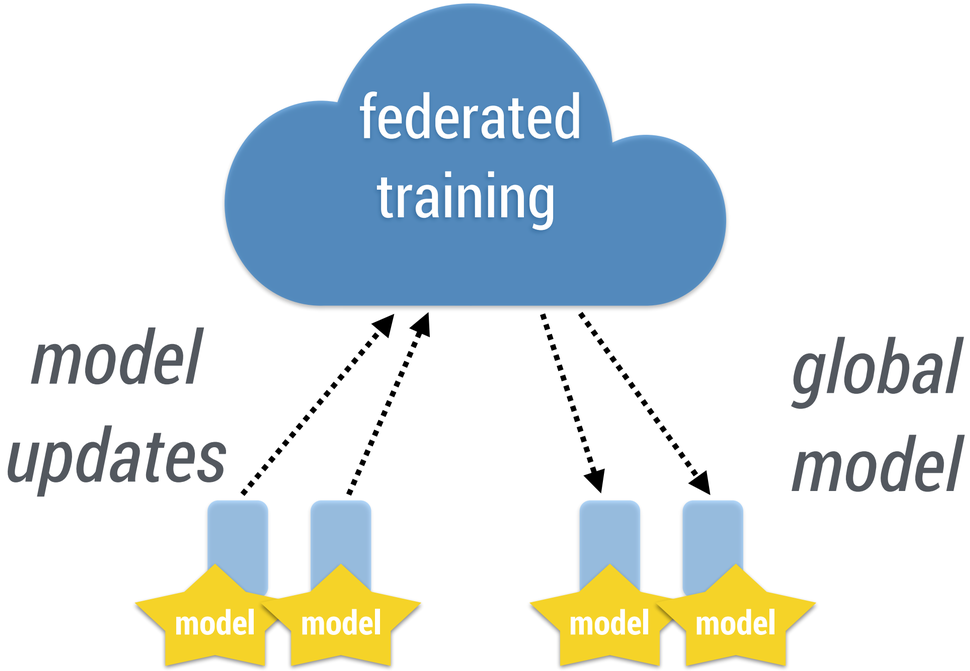


[McMahan et al., AISTATS 16]

federated optimization

training machine learning models at the edge





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

FedCoCoA

personalization is key

- non-IID data, underlying statistical structure

MOCHA

FedCoCoA: communication-efficient federated optimization

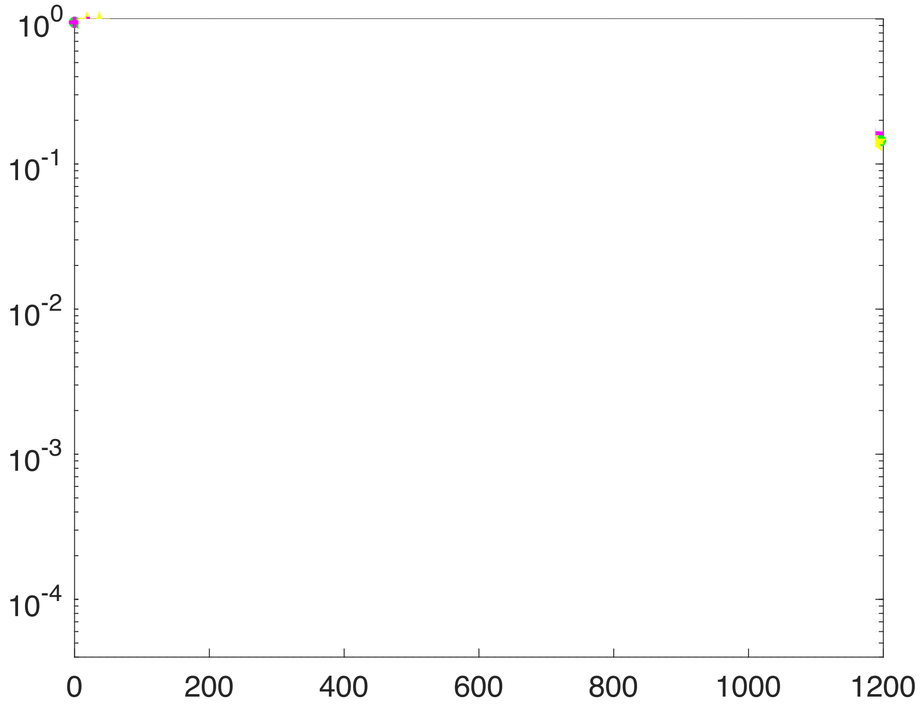
[Smith, Chiang, Sanjabi, Talwalkar, NIPS 2017]

[Smith, Forte, Ma, Takac, Jordan, Jaggi, JMLR 2018]

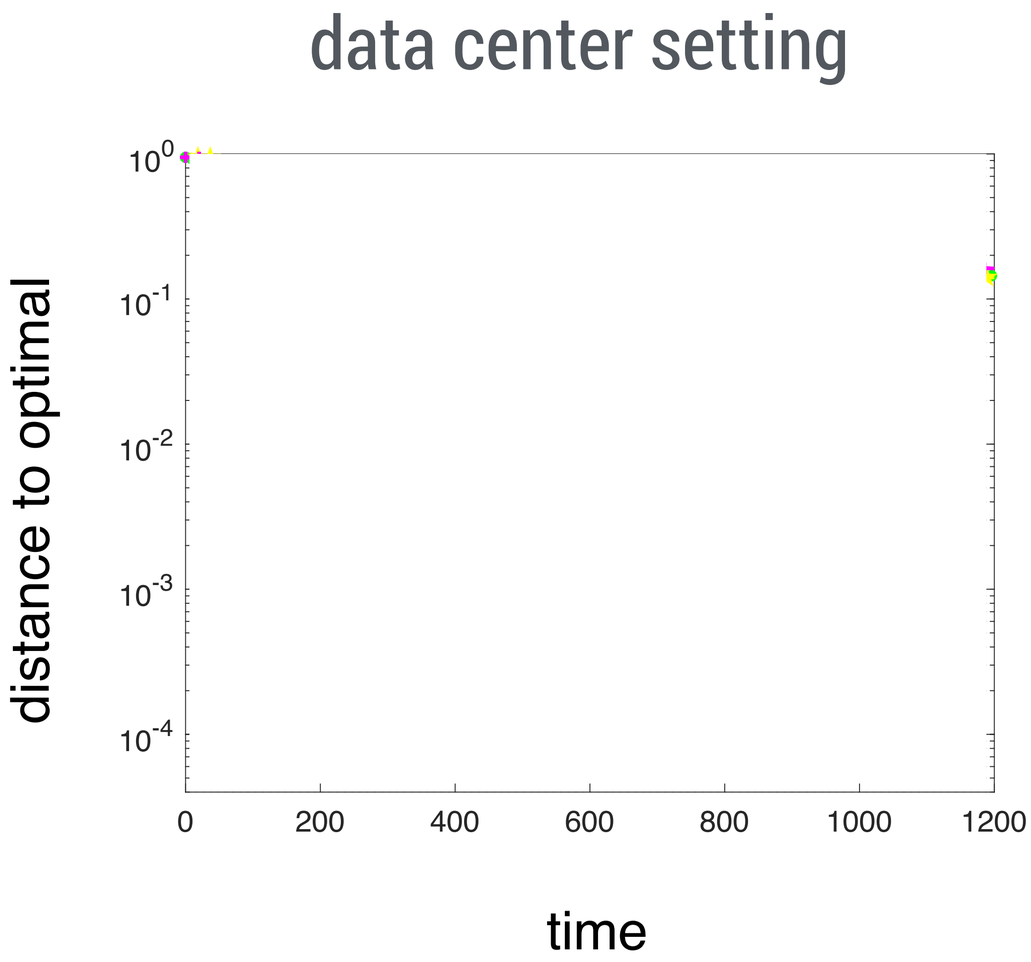
first approach: data center optimization methods

first approach: data center optimization methods

data center setting

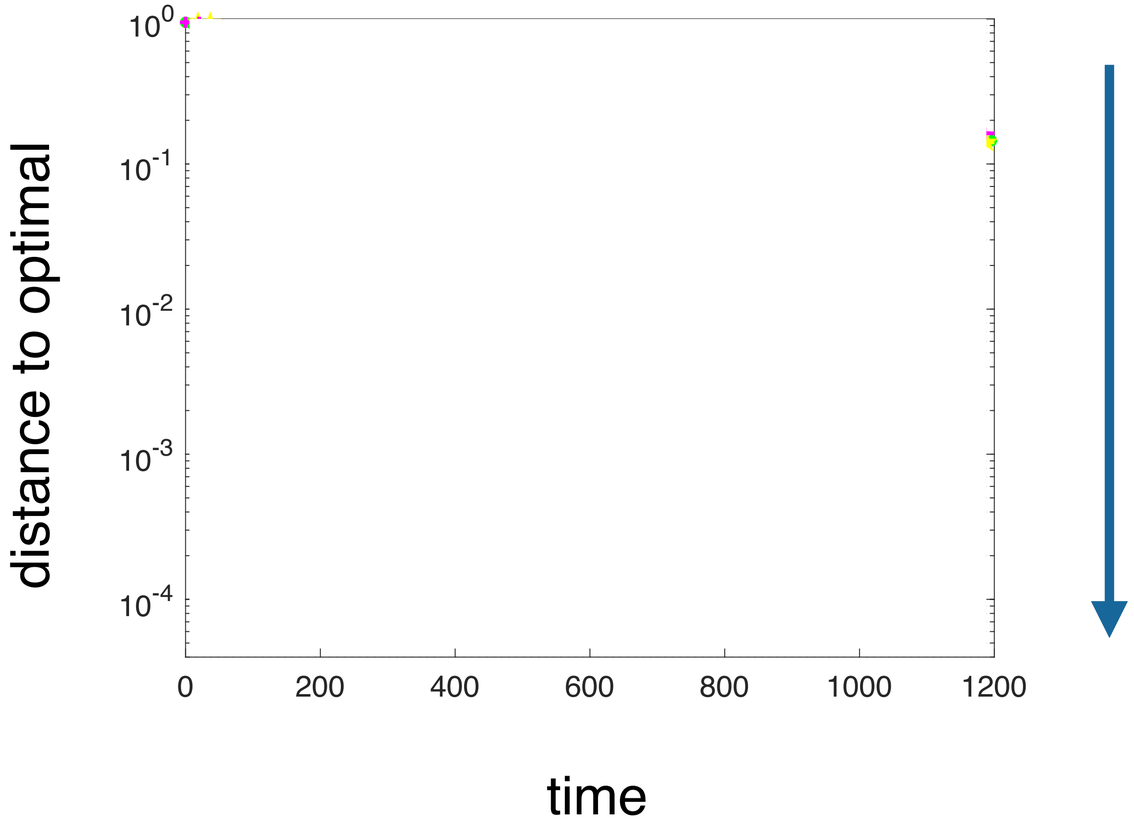


first approach: data center optimization methods



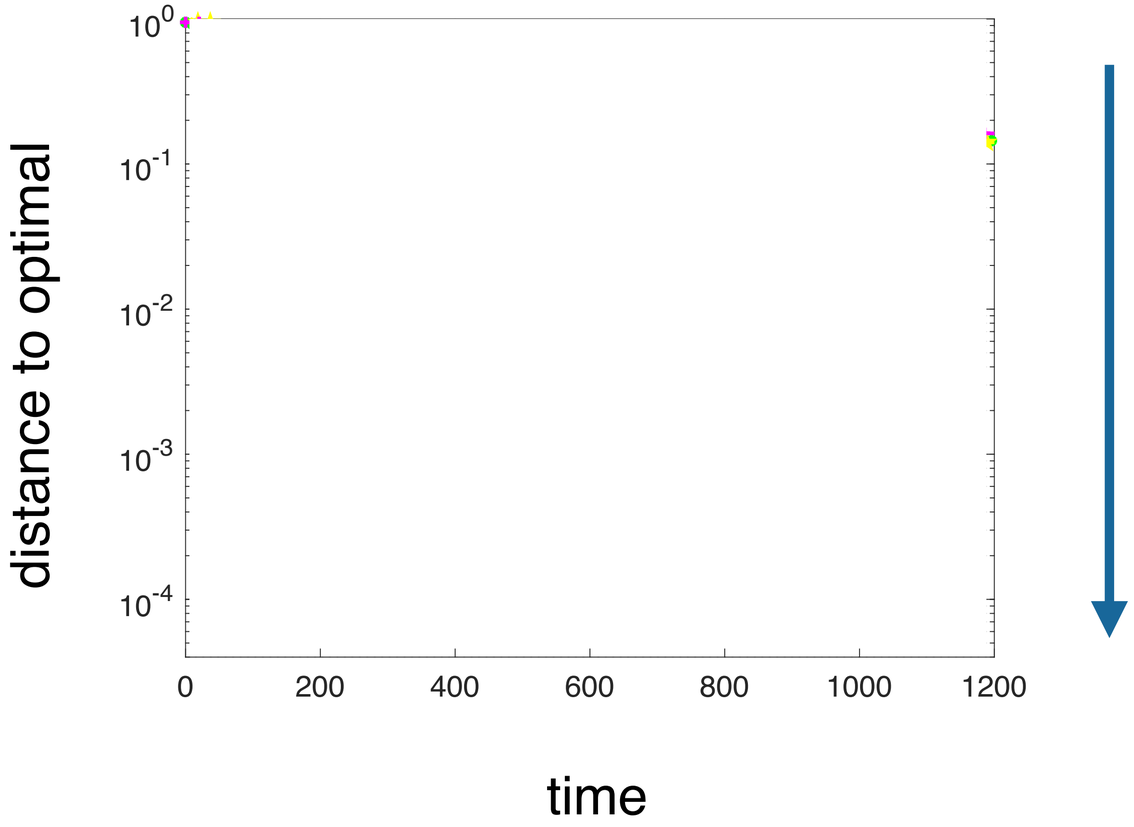
first approach: data center optimization methods

data center setting



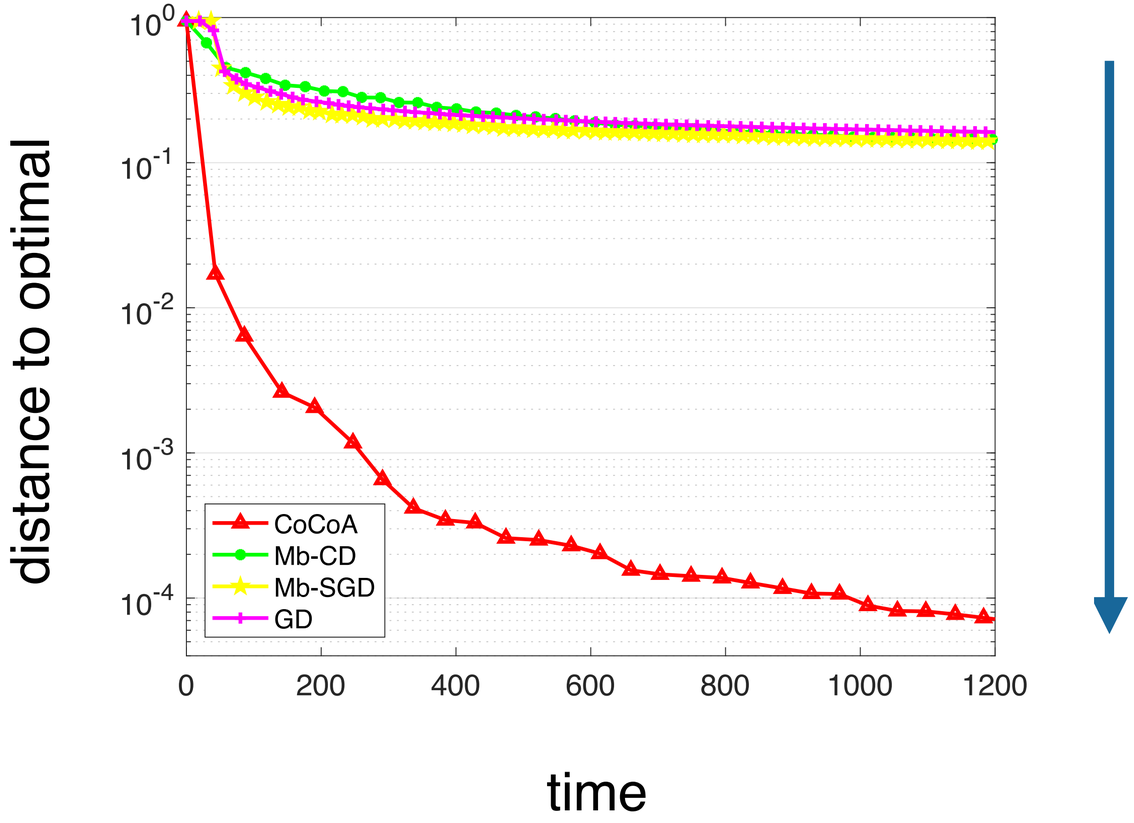
first approach: data center optimization methods

data center setting



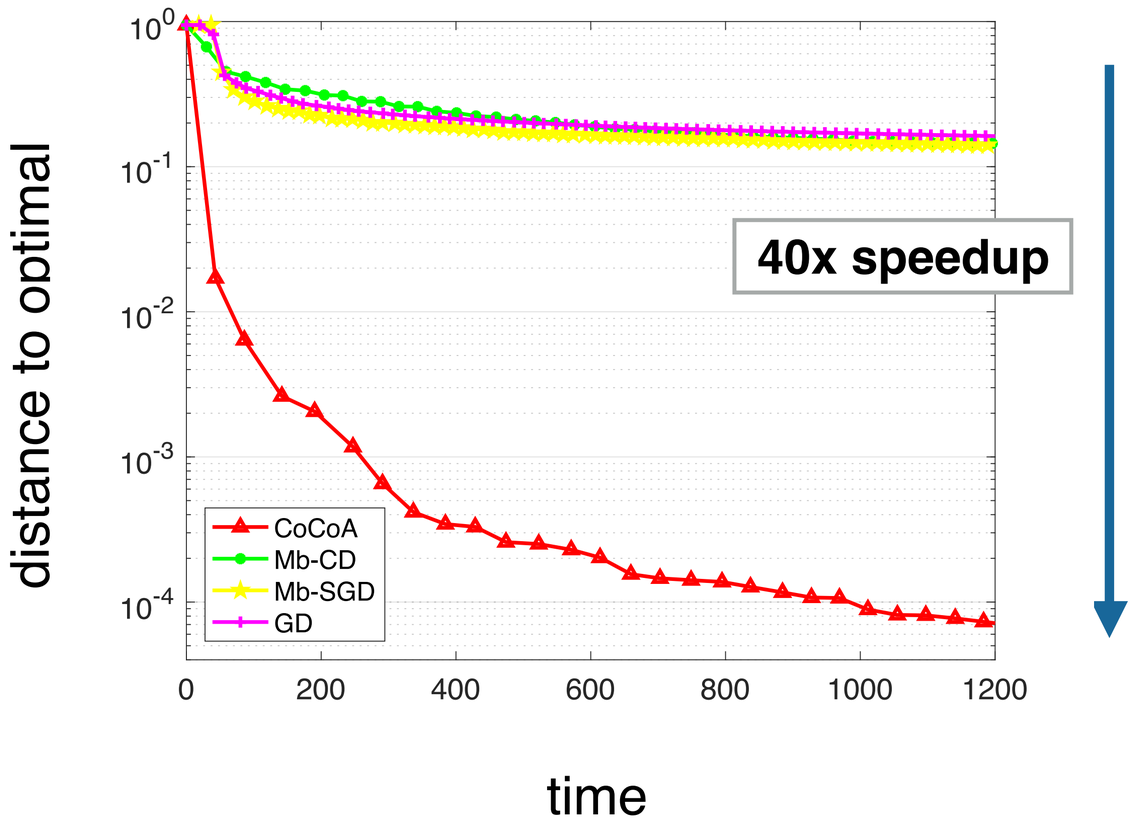
first approach: data center optimization methods

data center setting



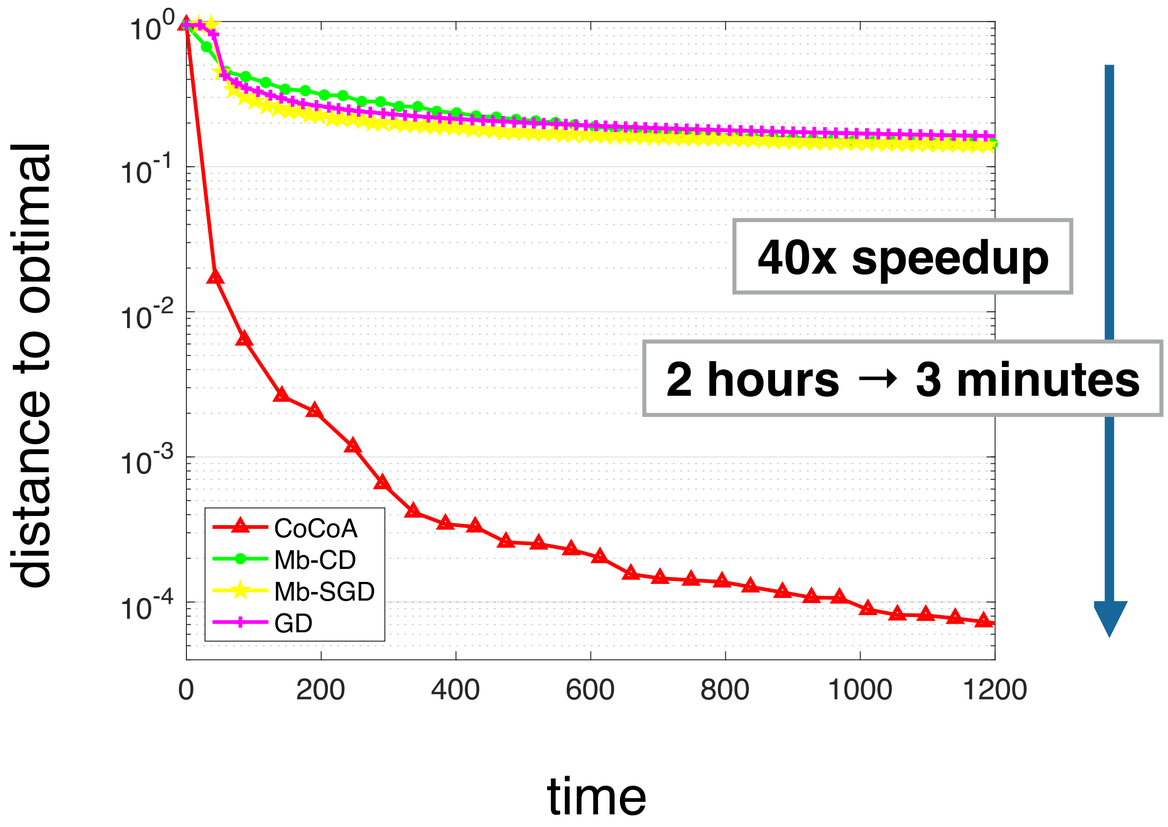
first approach: data center optimization methods

data center setting



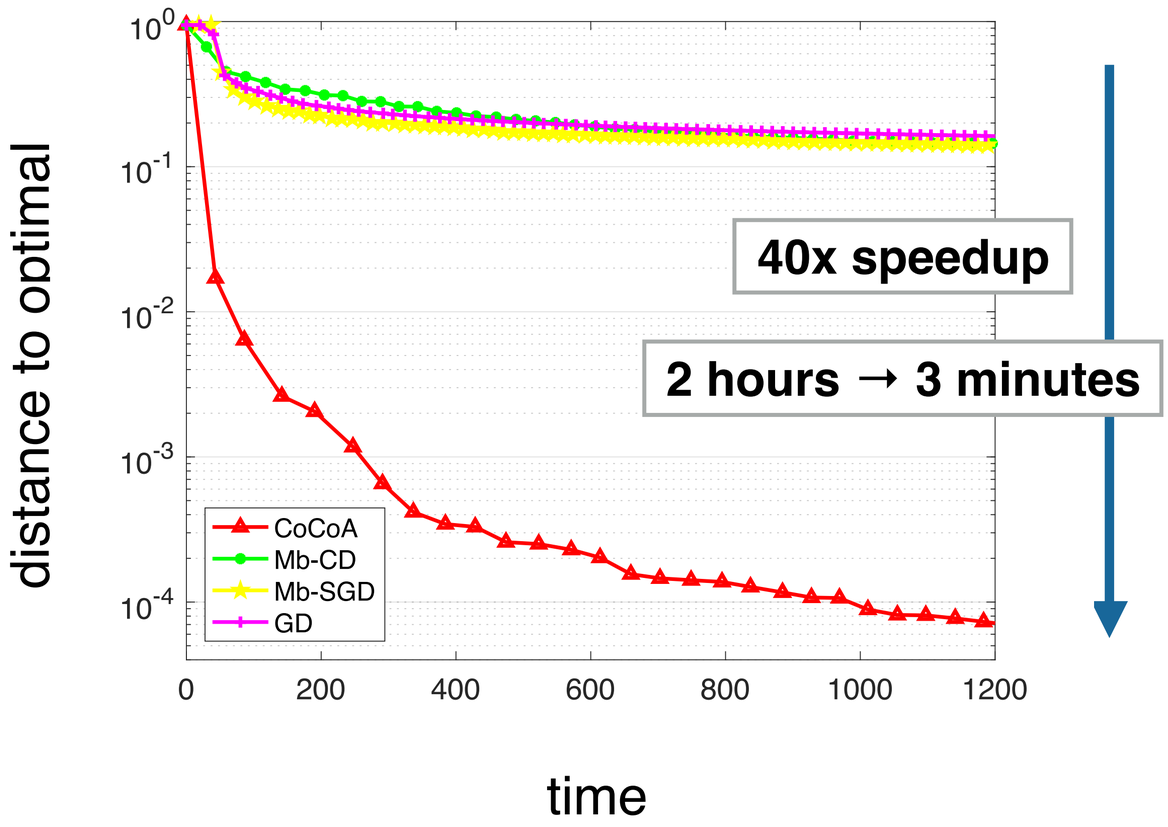
first approach: data center optimization methods

data center setting



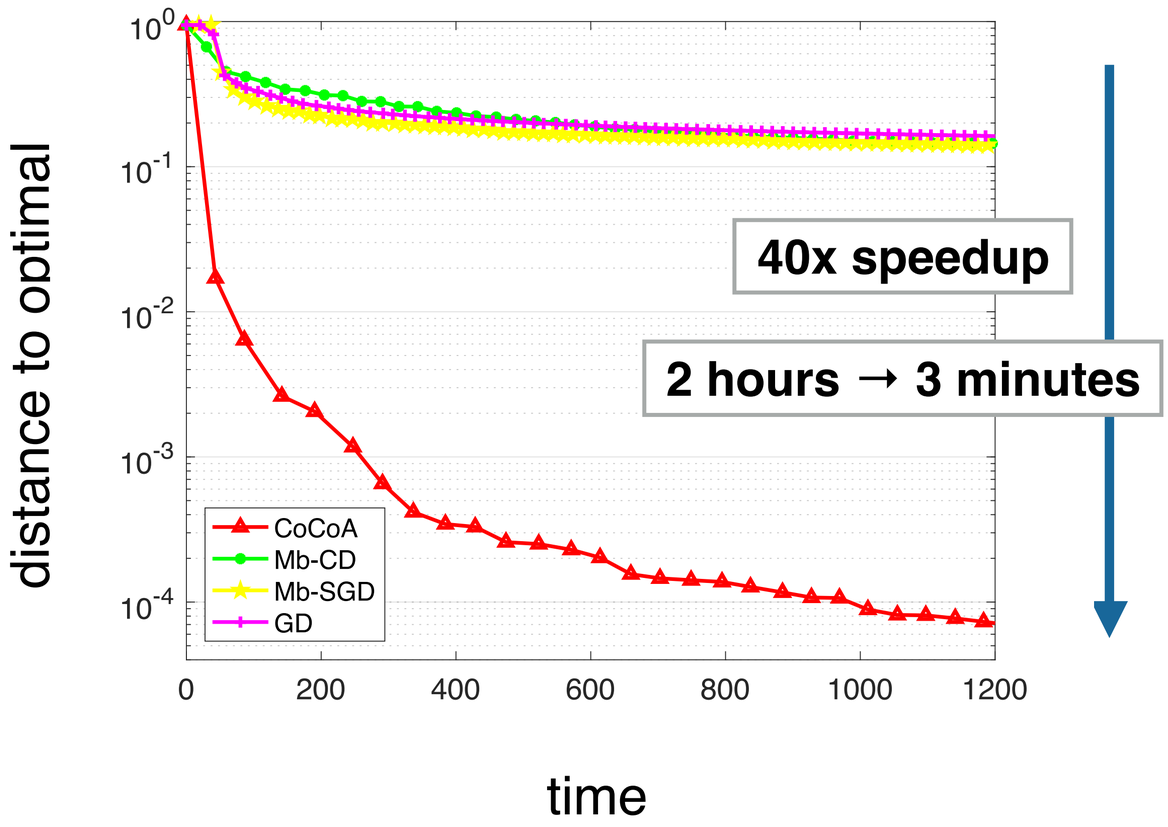
first approach: data center optimization methods

✓ data center setting

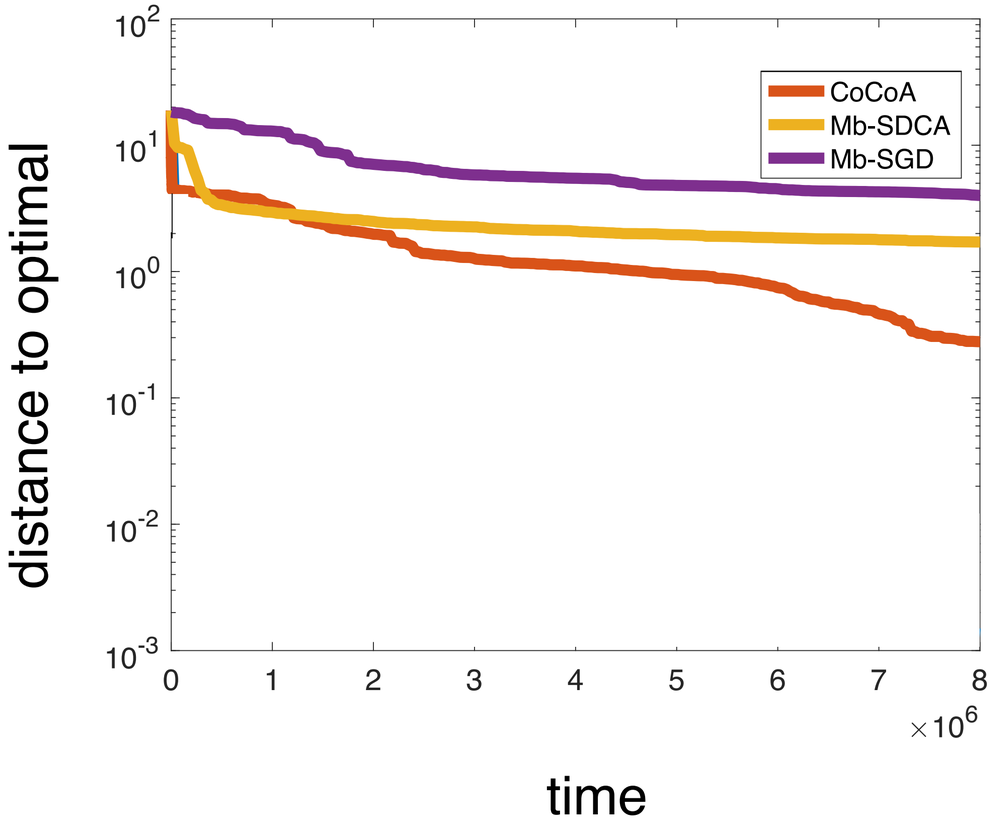


first approach: data center optimization methods

✓ data center setting

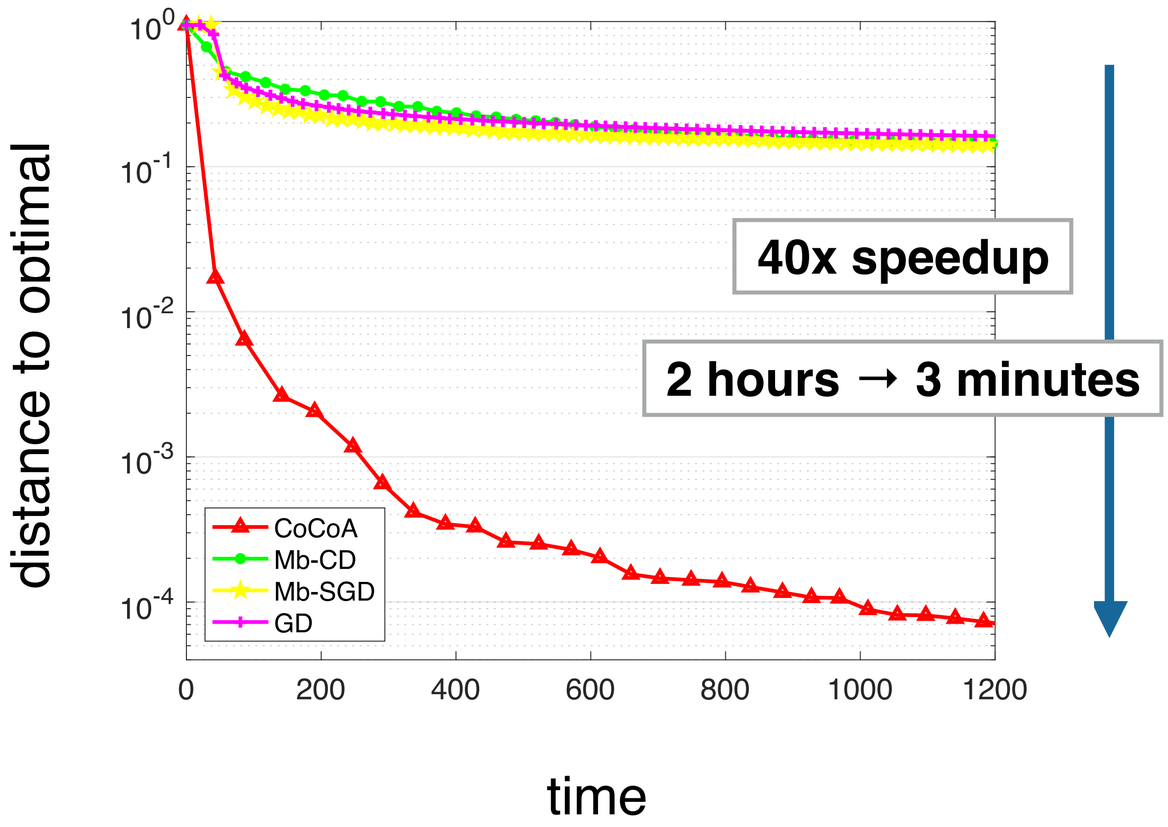


✗ federated setting

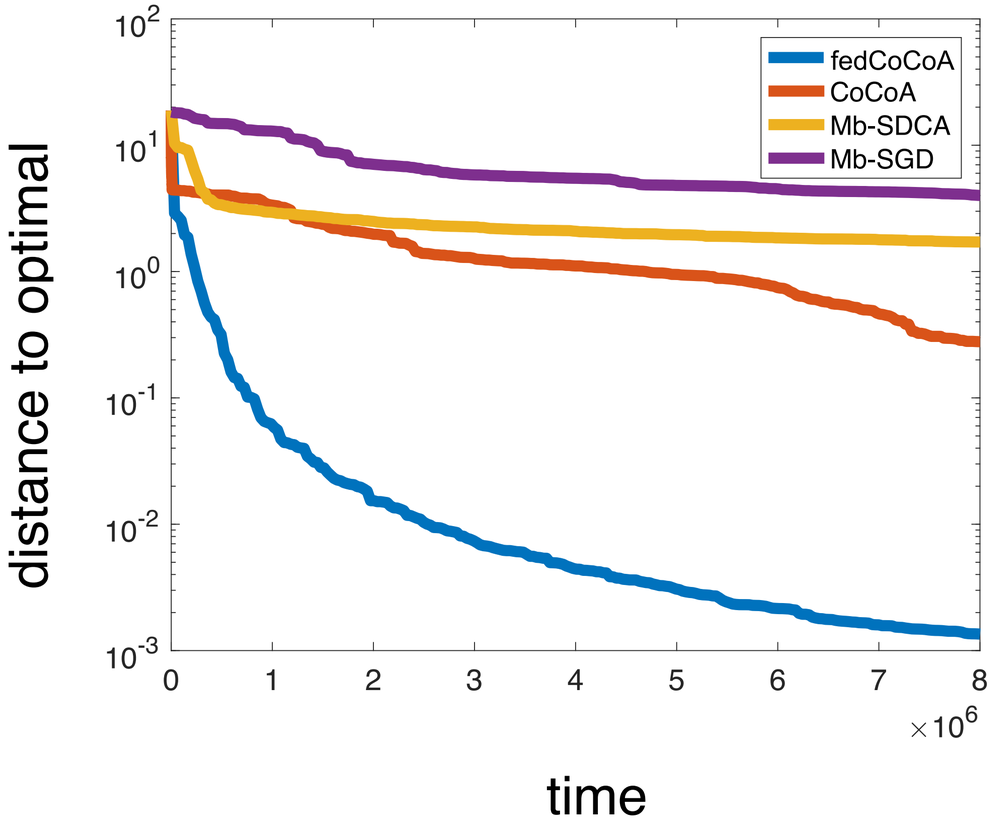


first approach: data center optimization methods

✓ data center setting



✗ federated setting

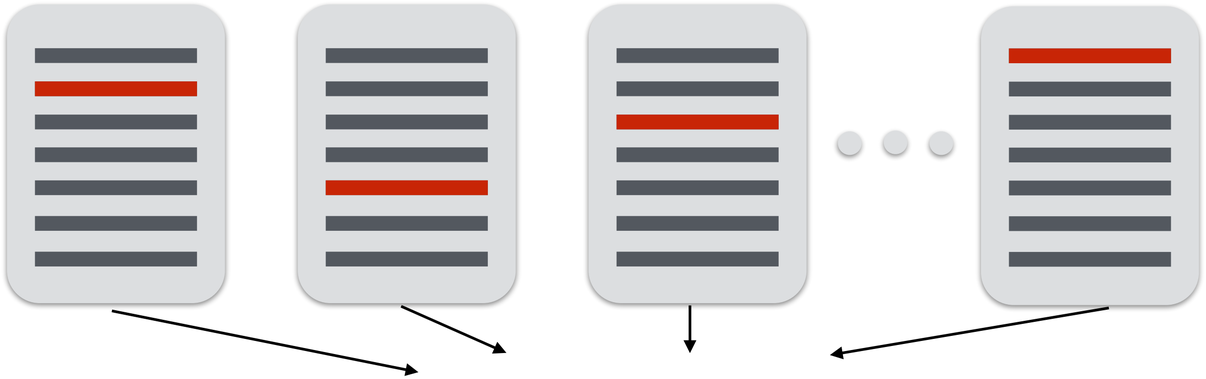


prior work: distributed data center optimization

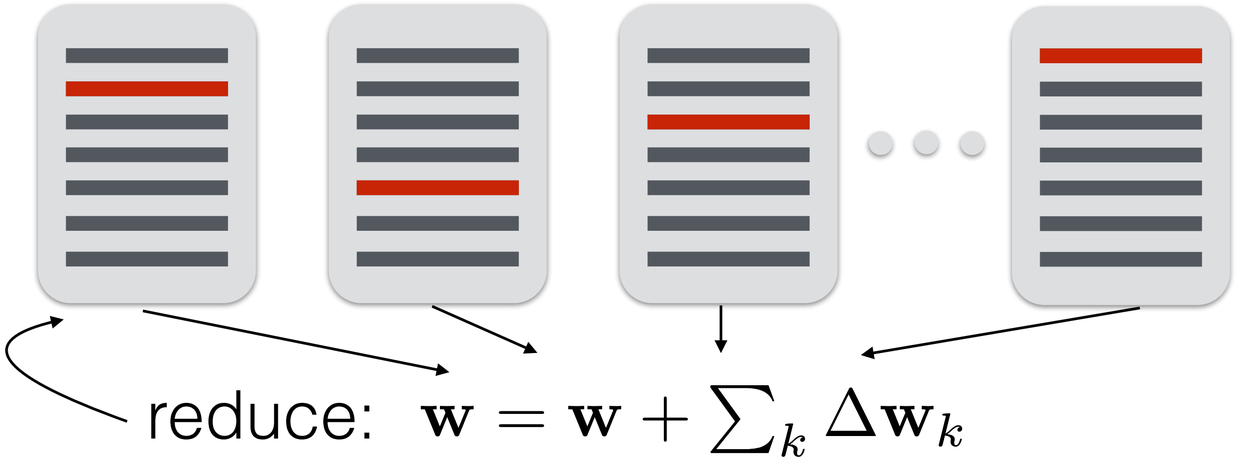
prior work: distributed data center optimization

prior work: distributed data center optimization

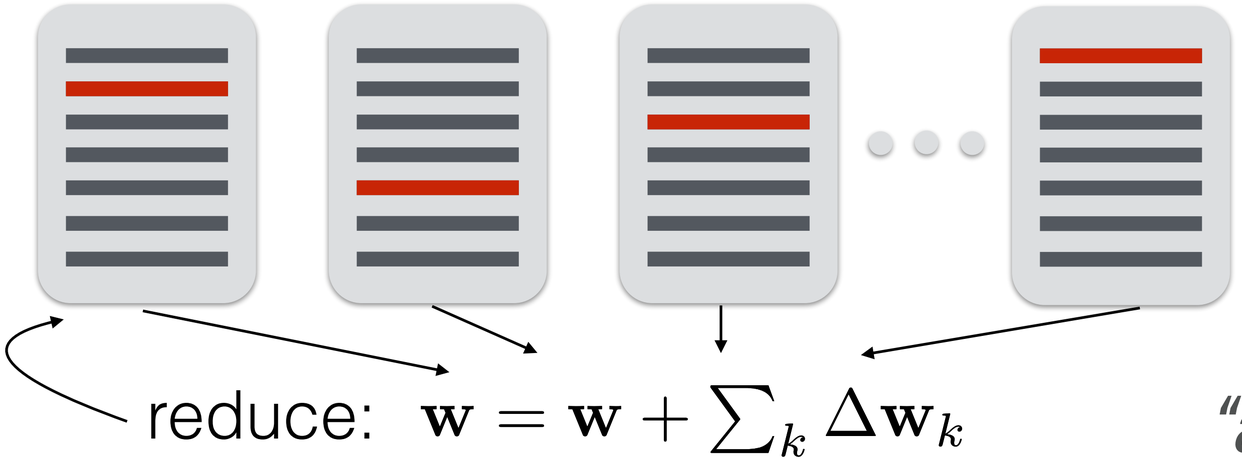
prior work: distributed data center optimization



prior work: distributed data center optimization



prior work: distributed data center optimization

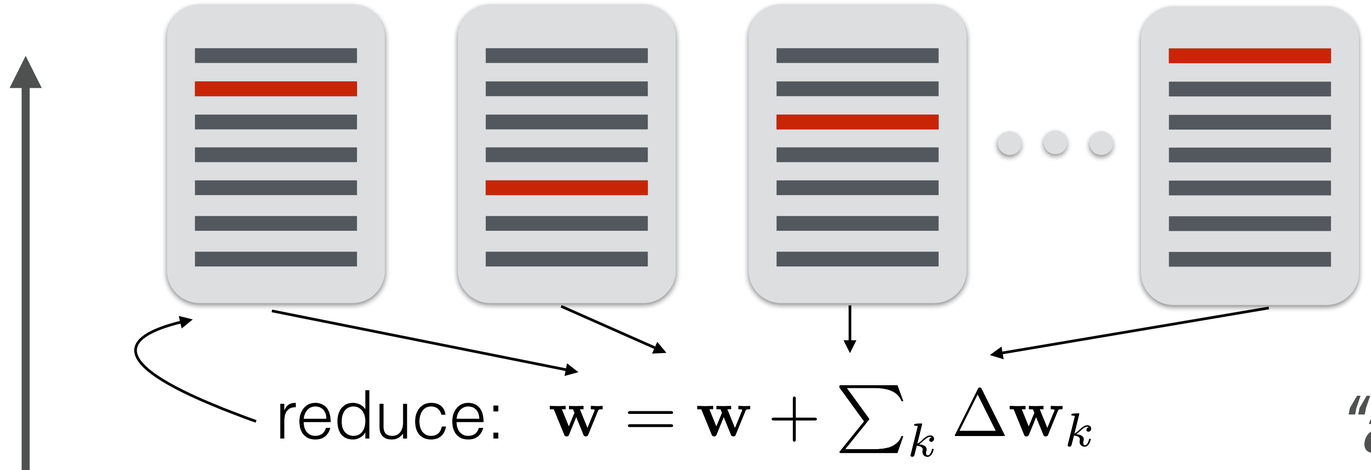


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization



✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

✔ convergence guarantees

✗ slow

"always communicate"

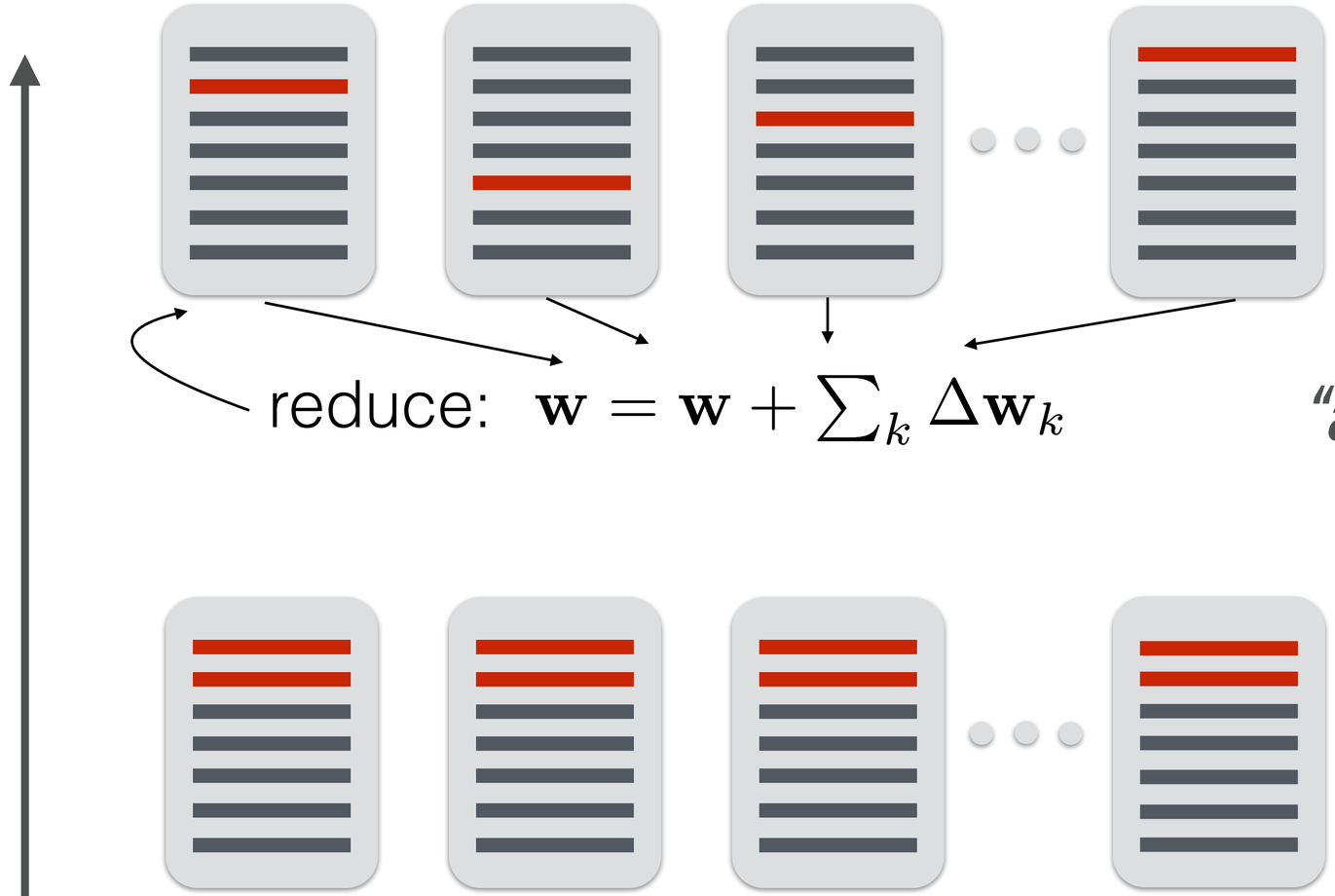
prior work: distributed data center optimization

✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

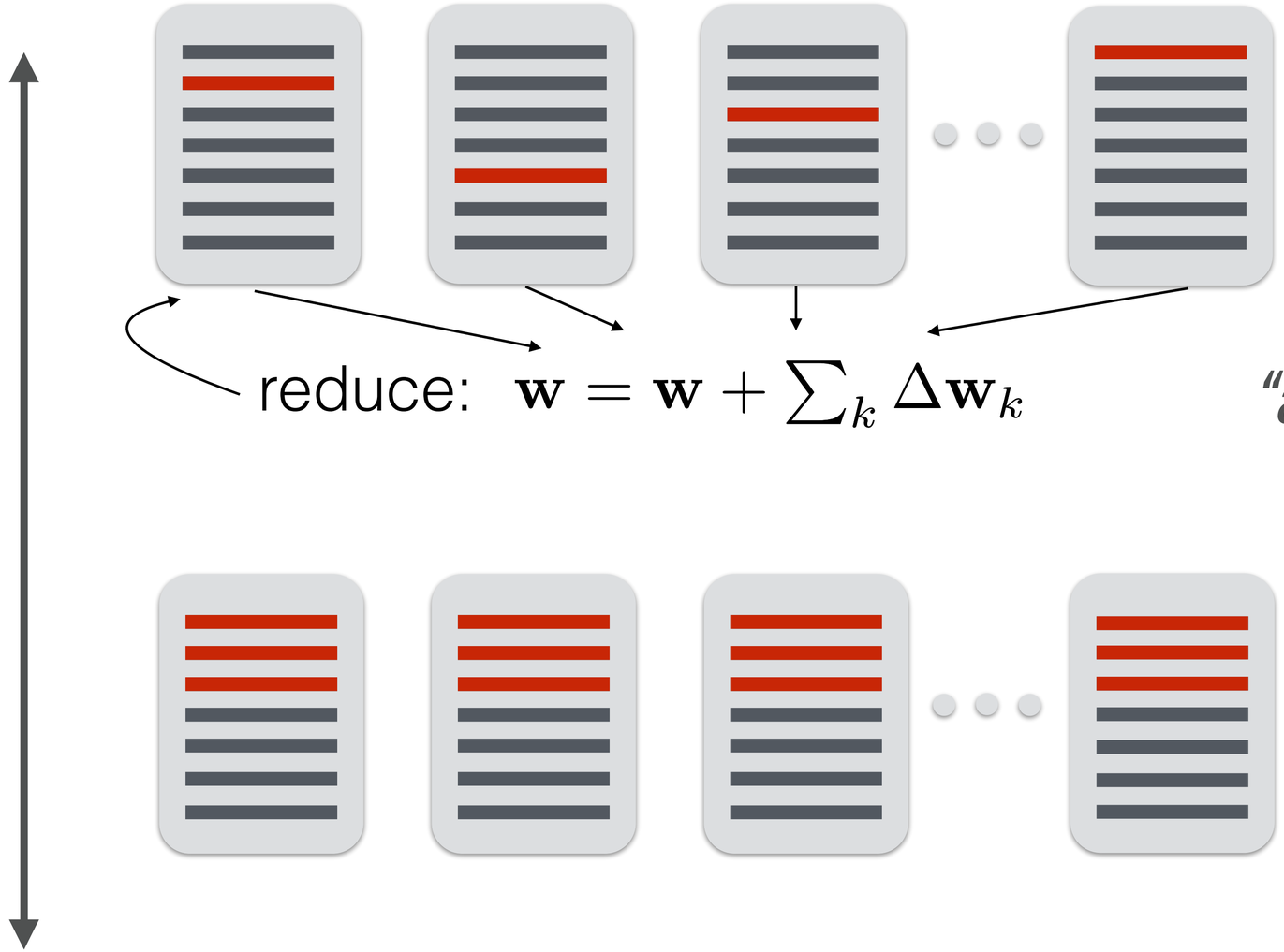


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

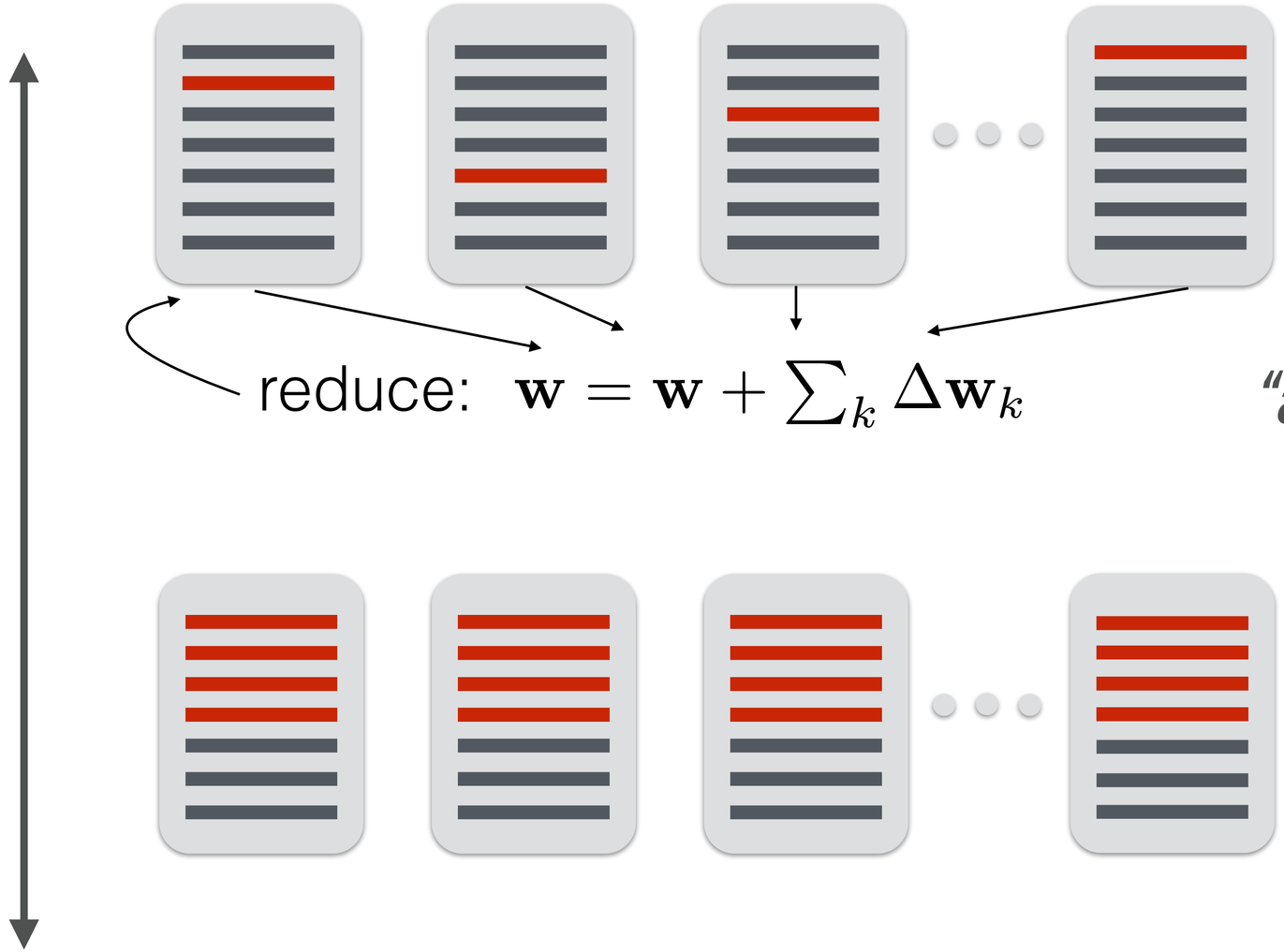


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

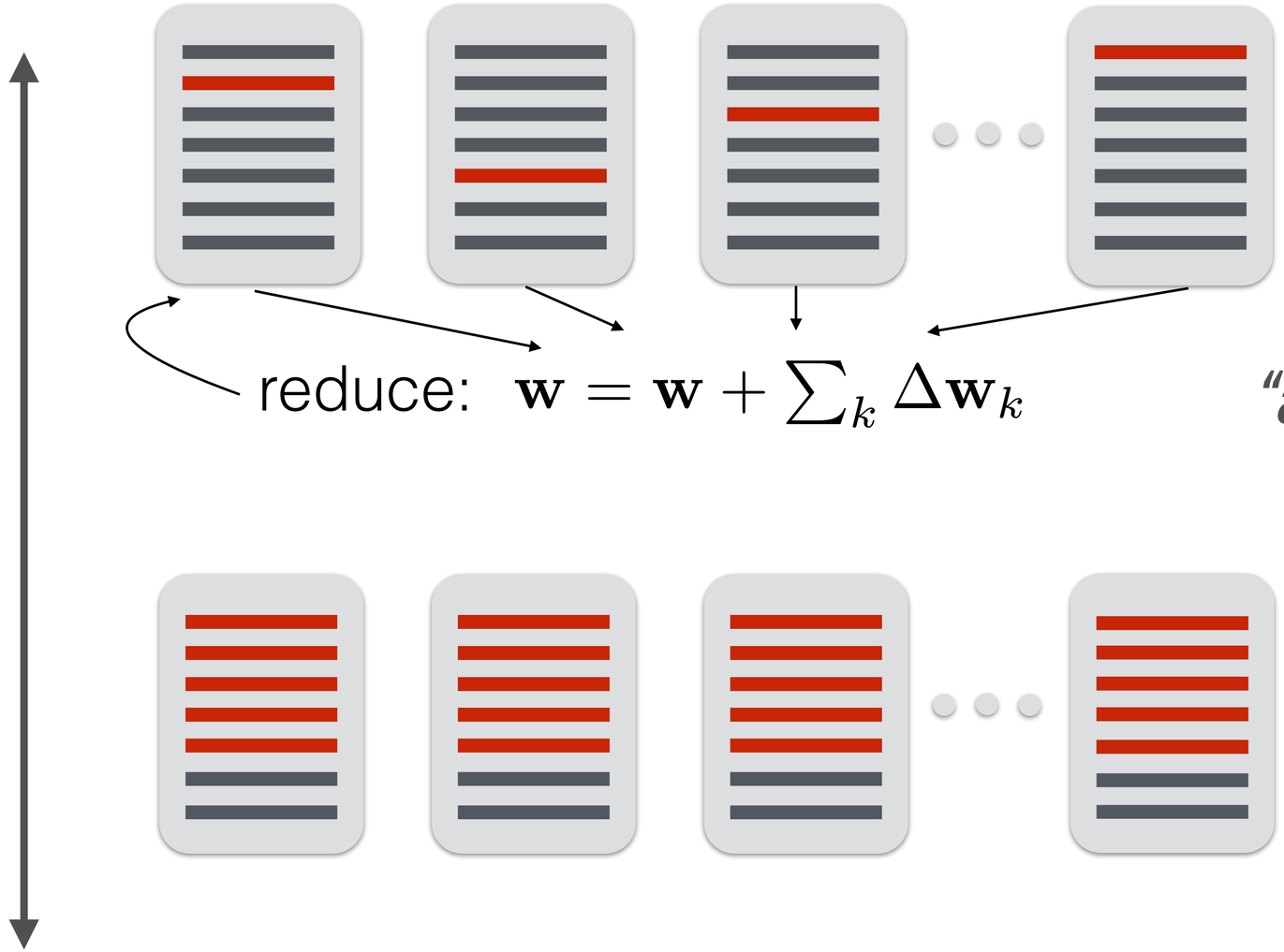


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

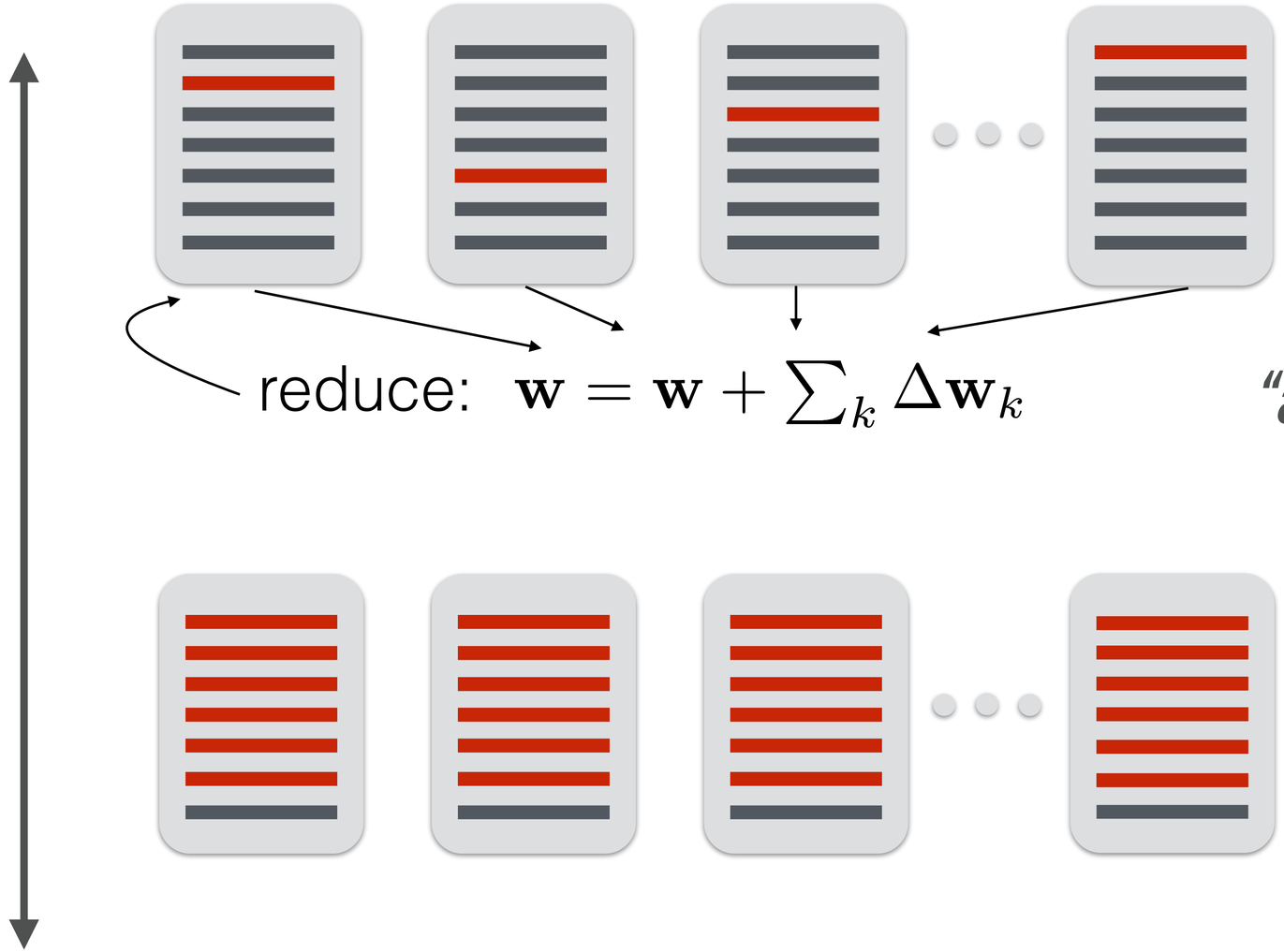


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

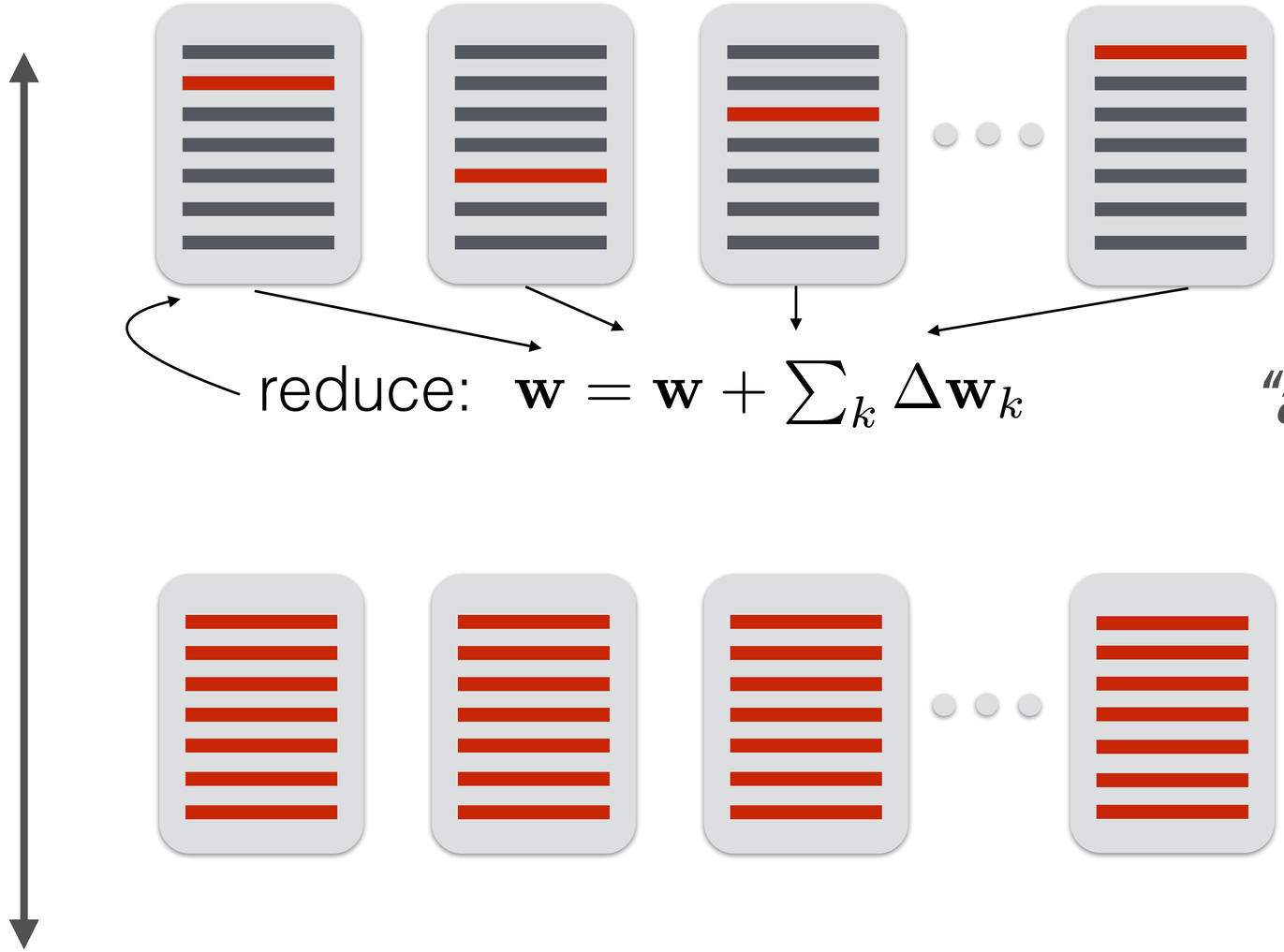


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

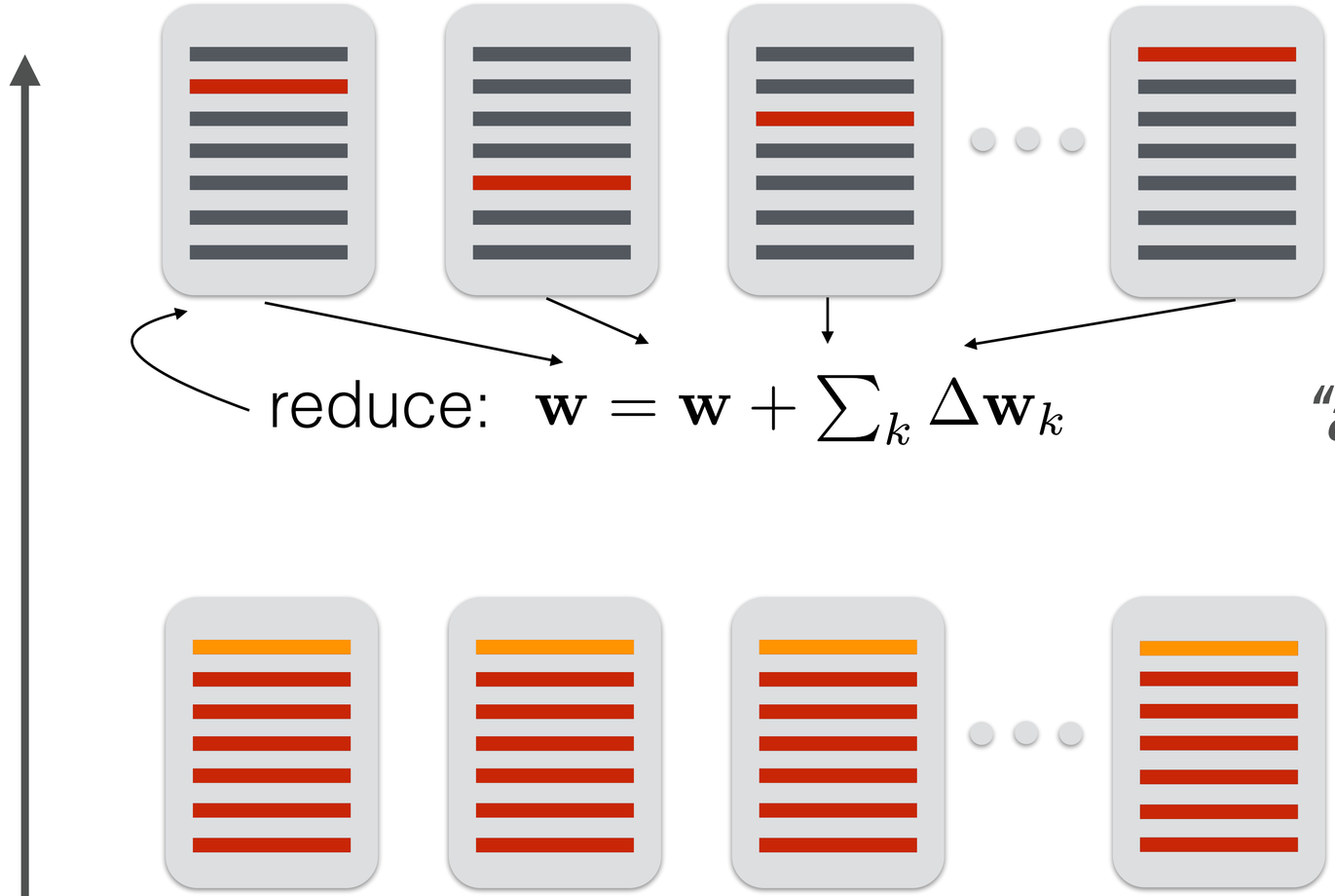


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

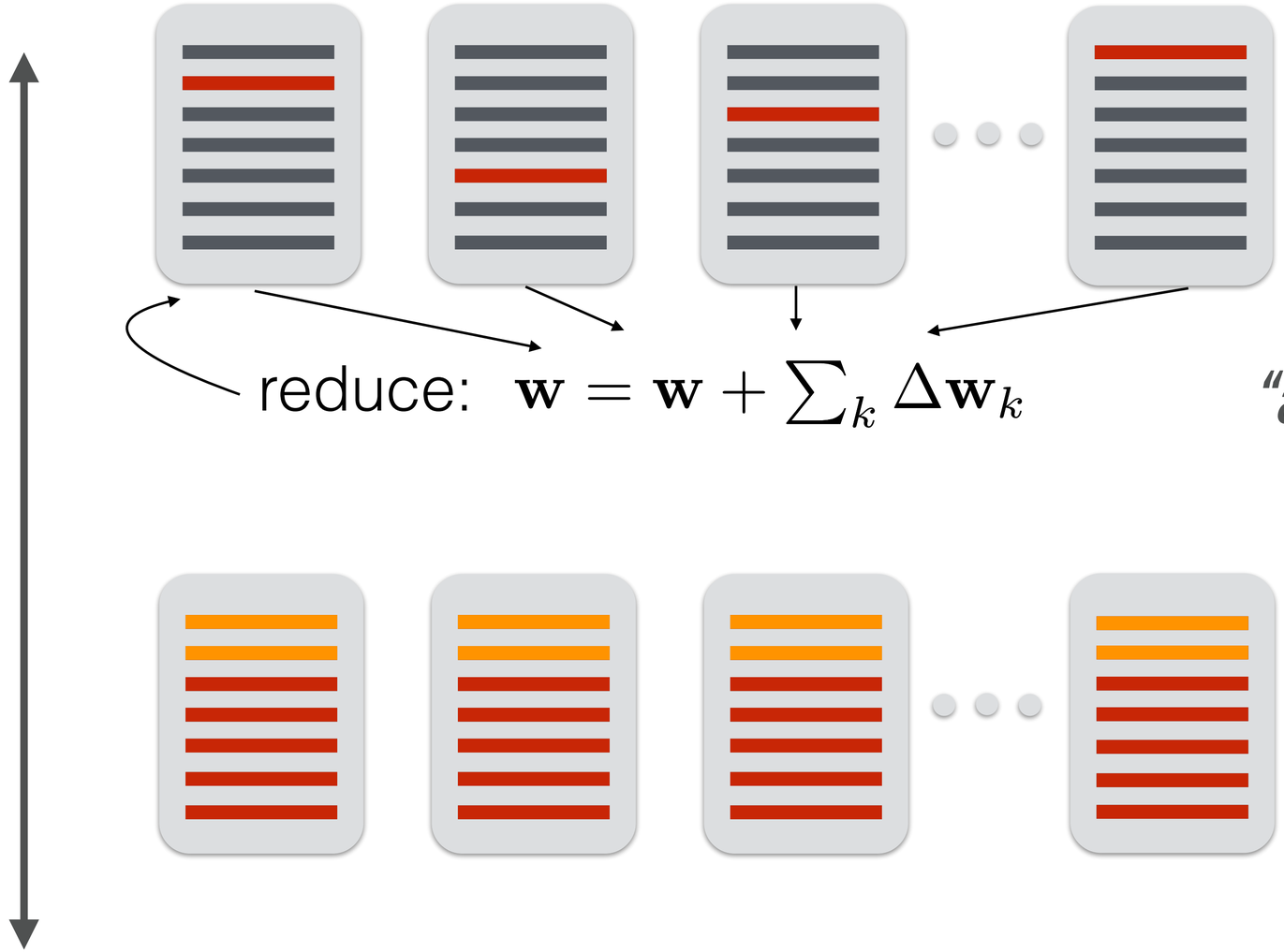


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

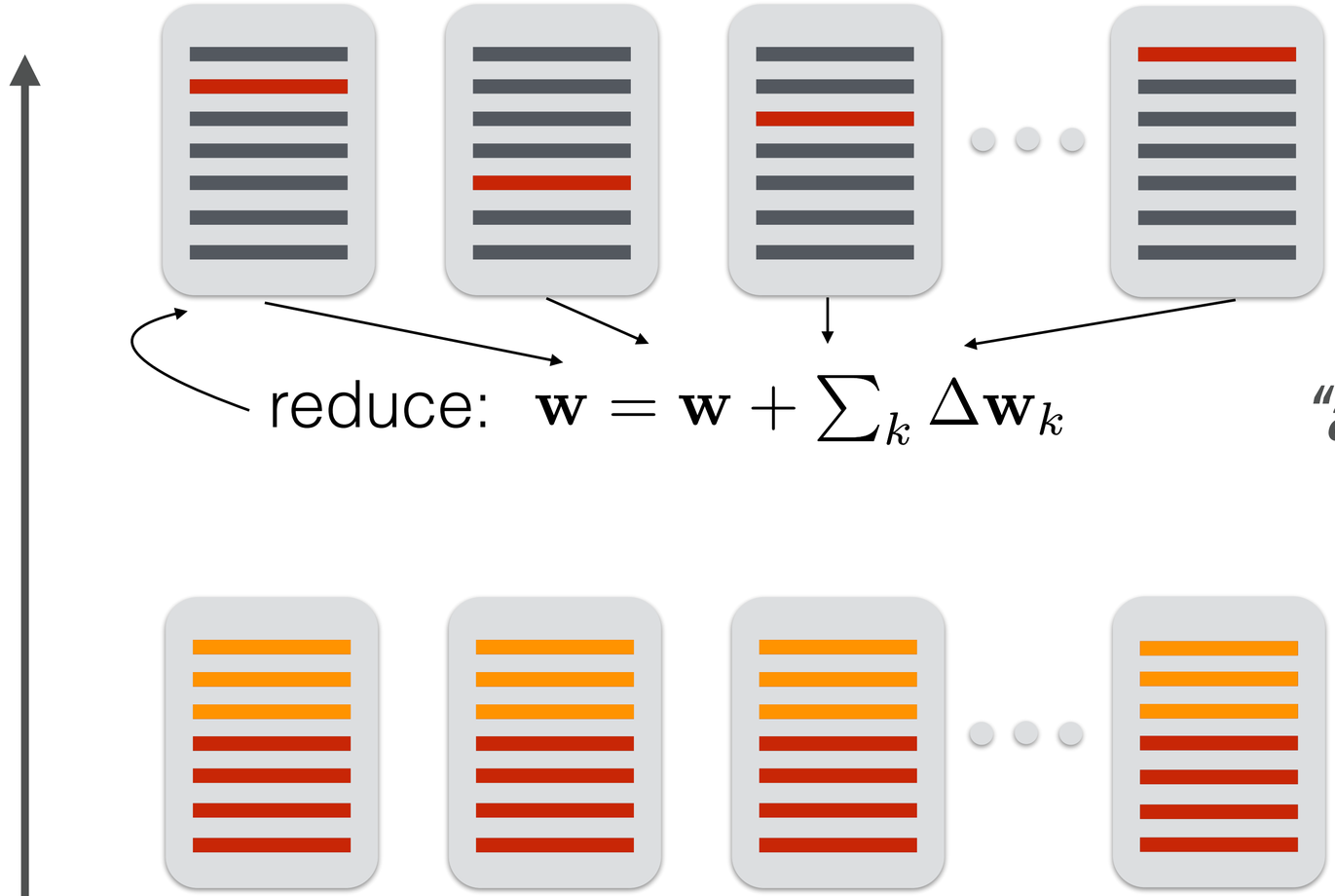


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

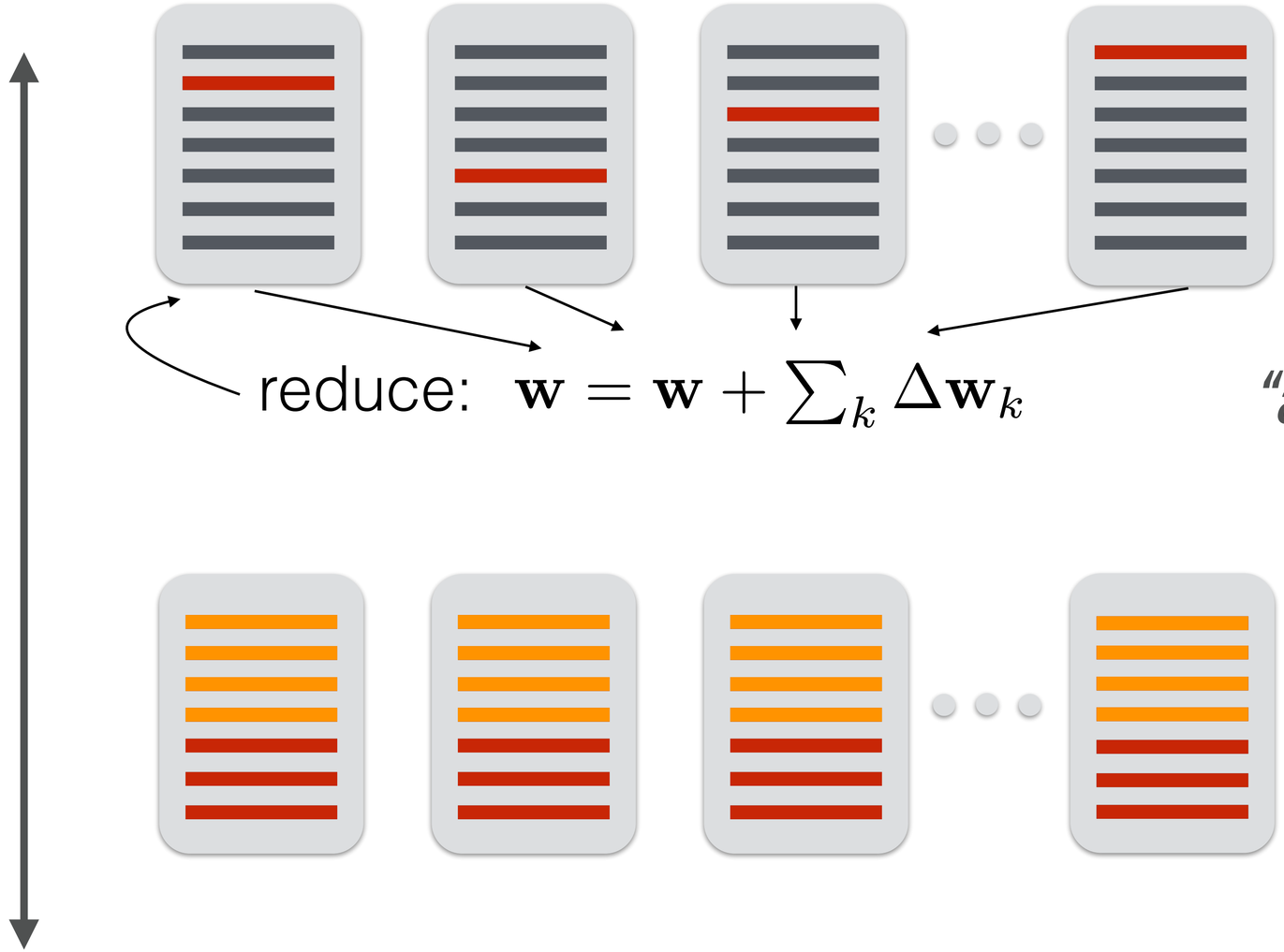


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

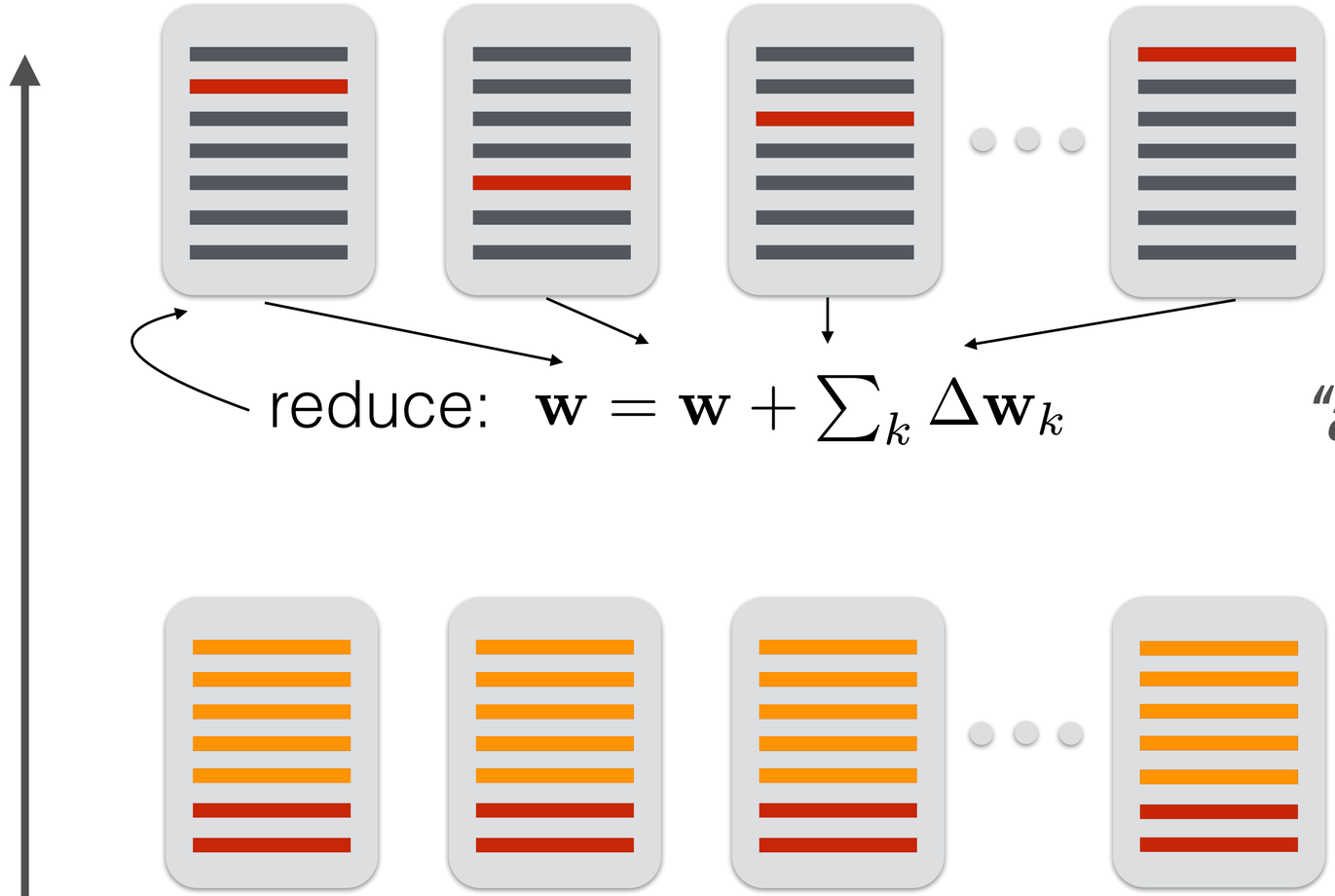


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

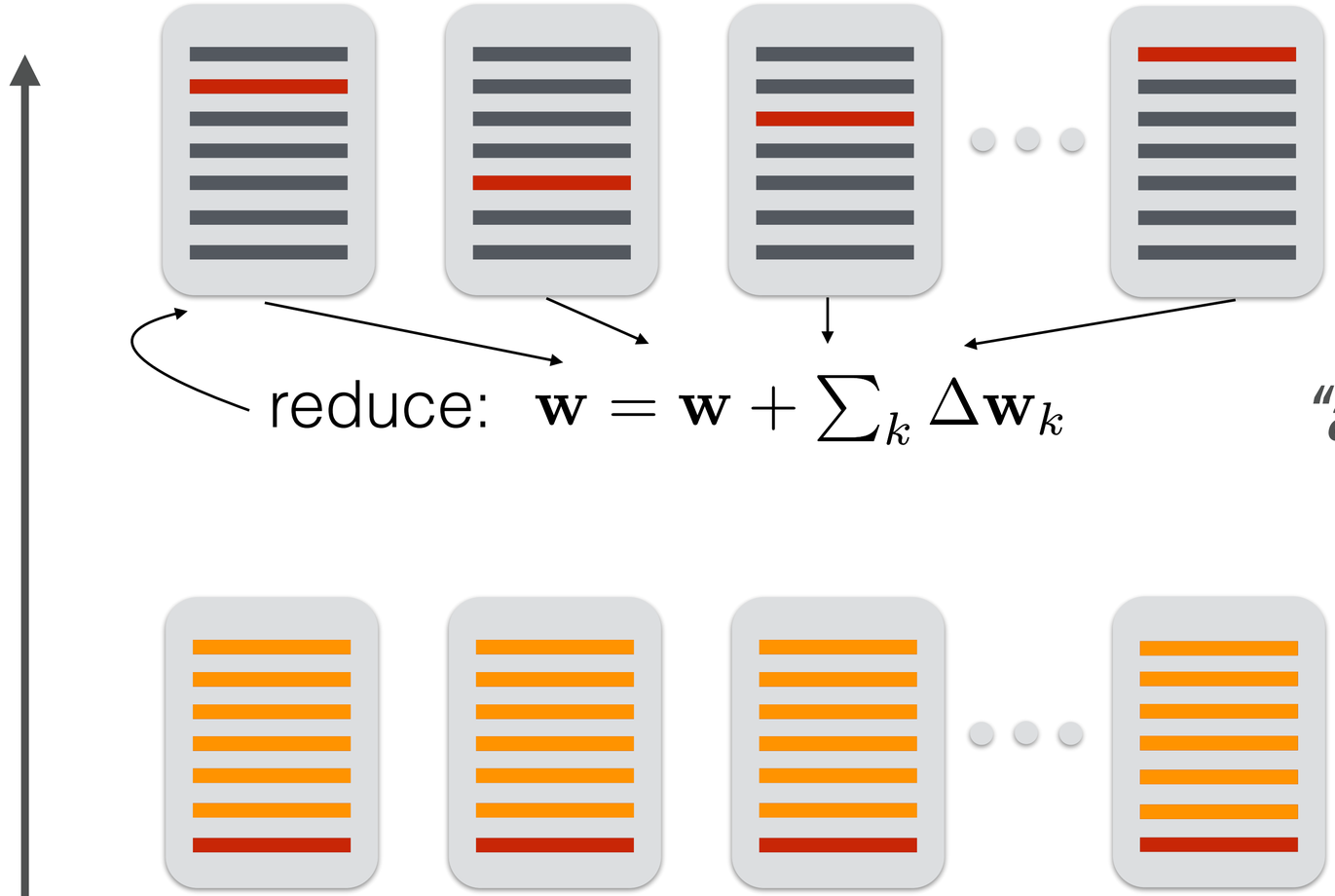


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization



✔ convergence guarantees

✗ slow

"always communicate"

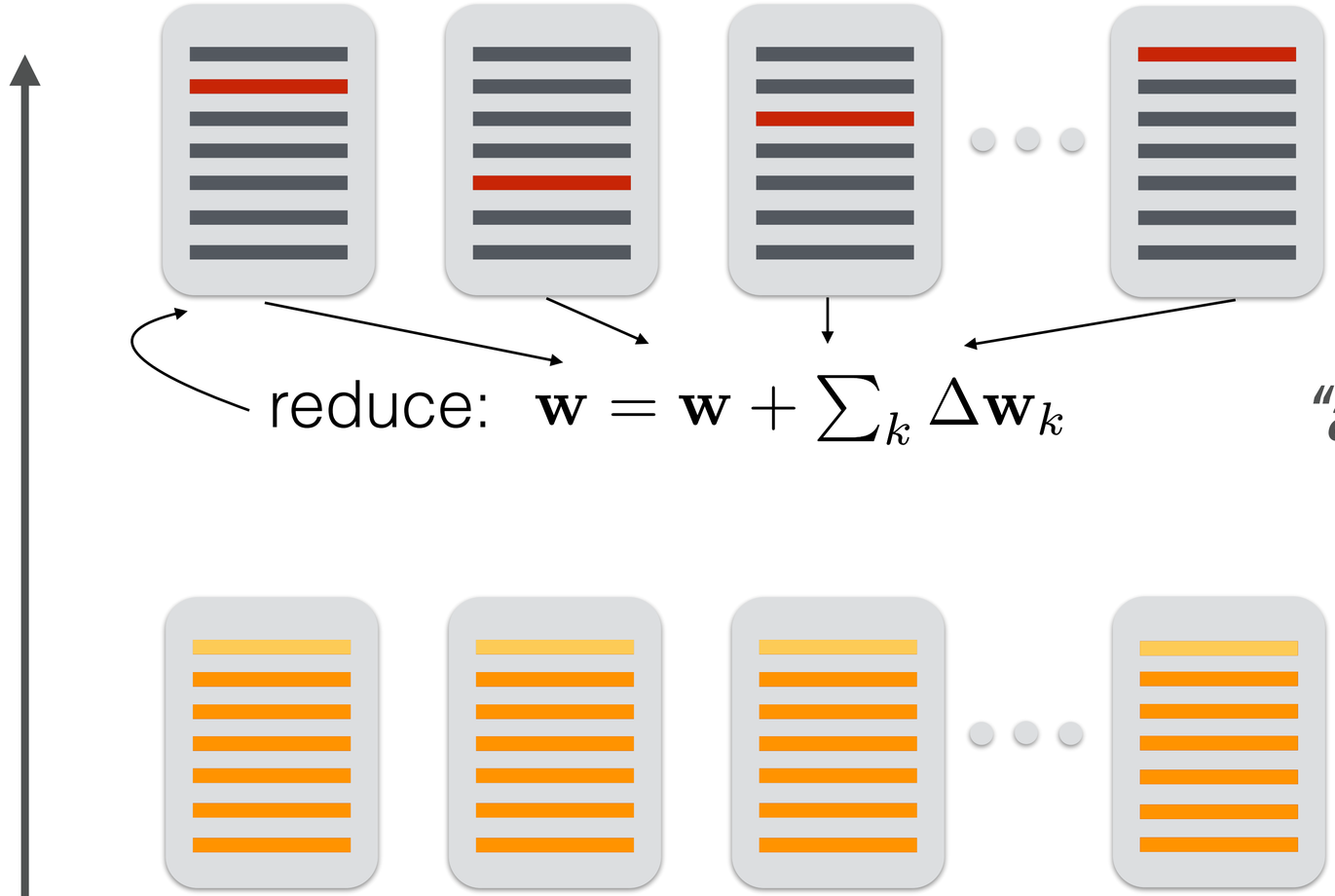
prior work: distributed data center optimization

✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

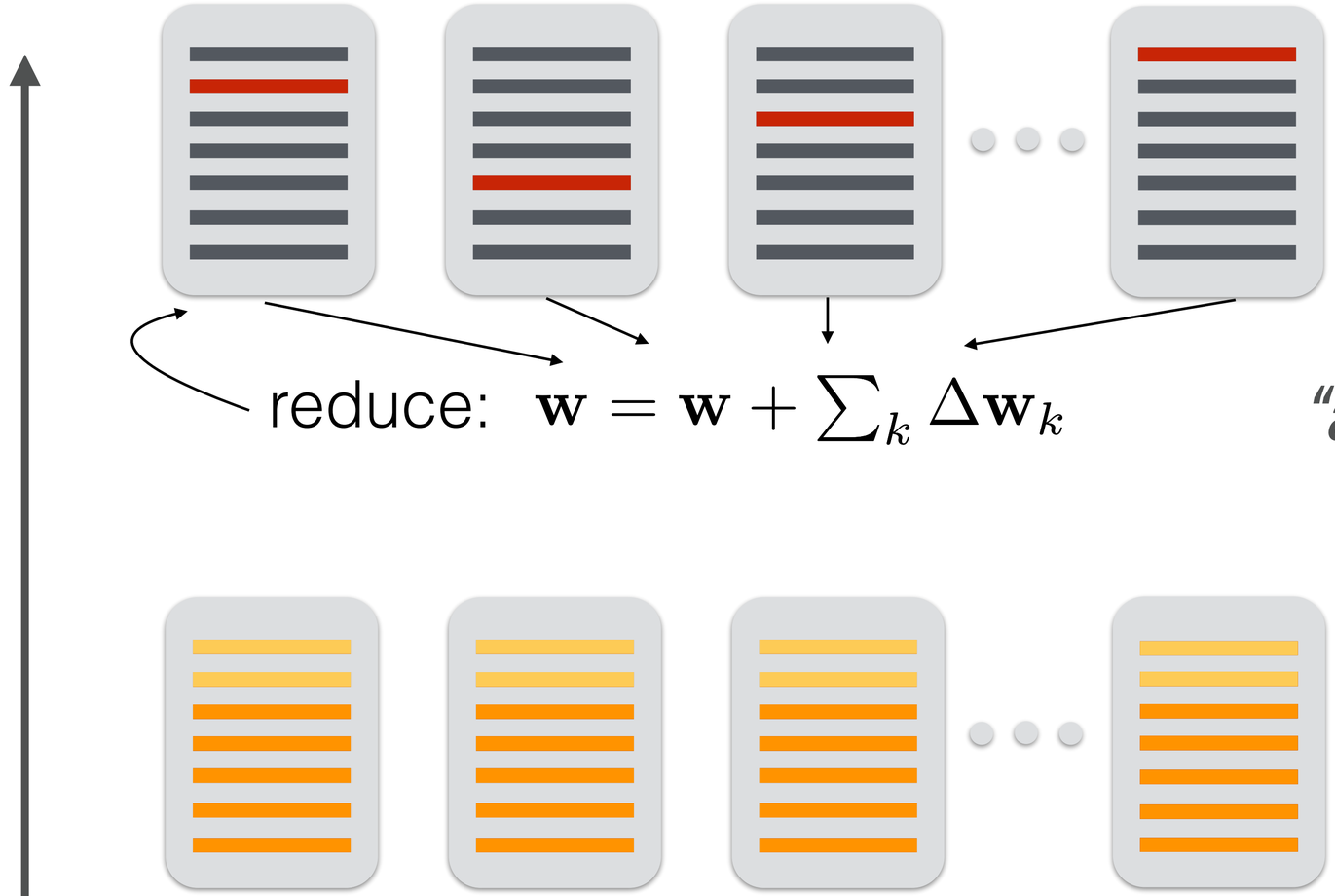


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

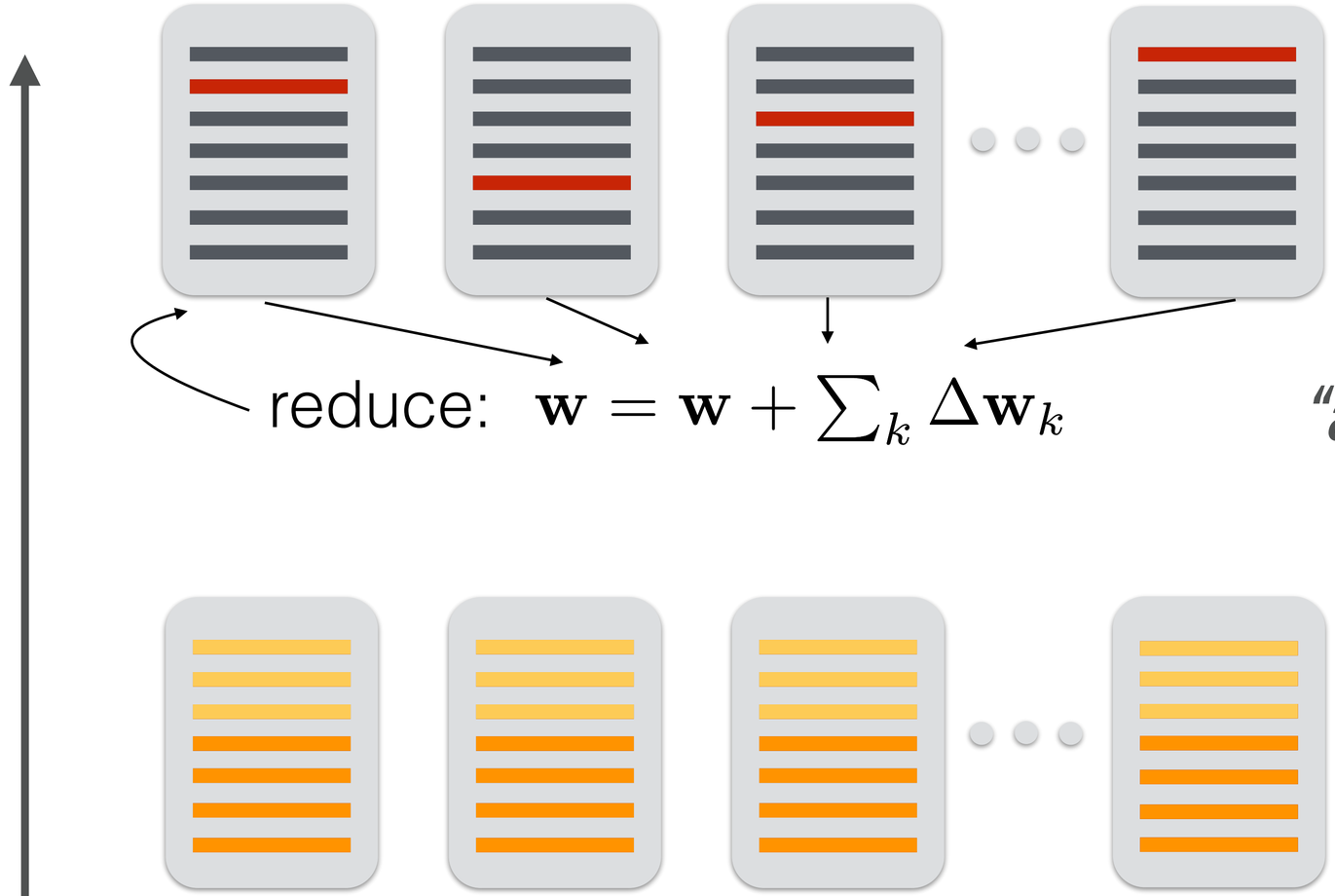


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

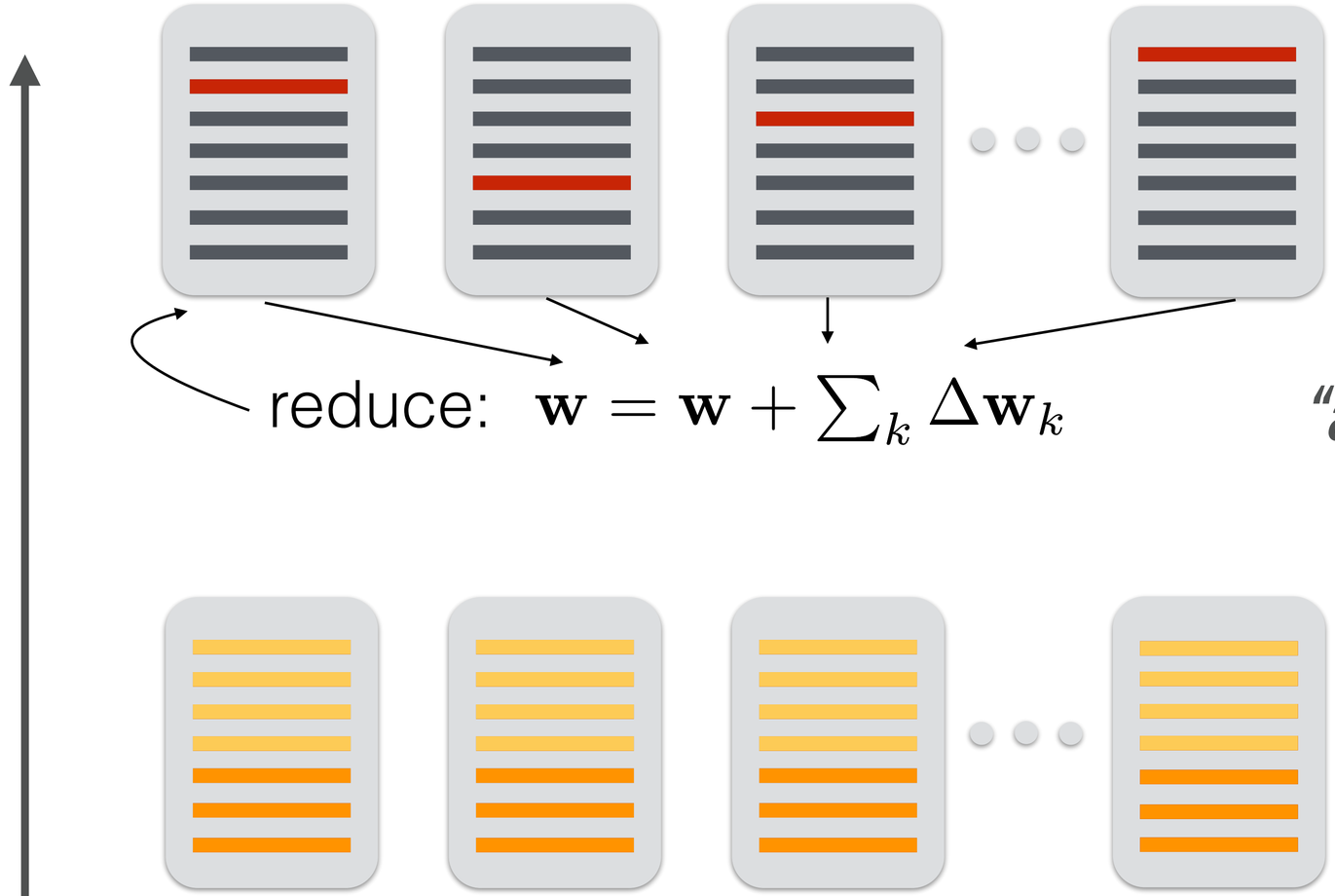


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

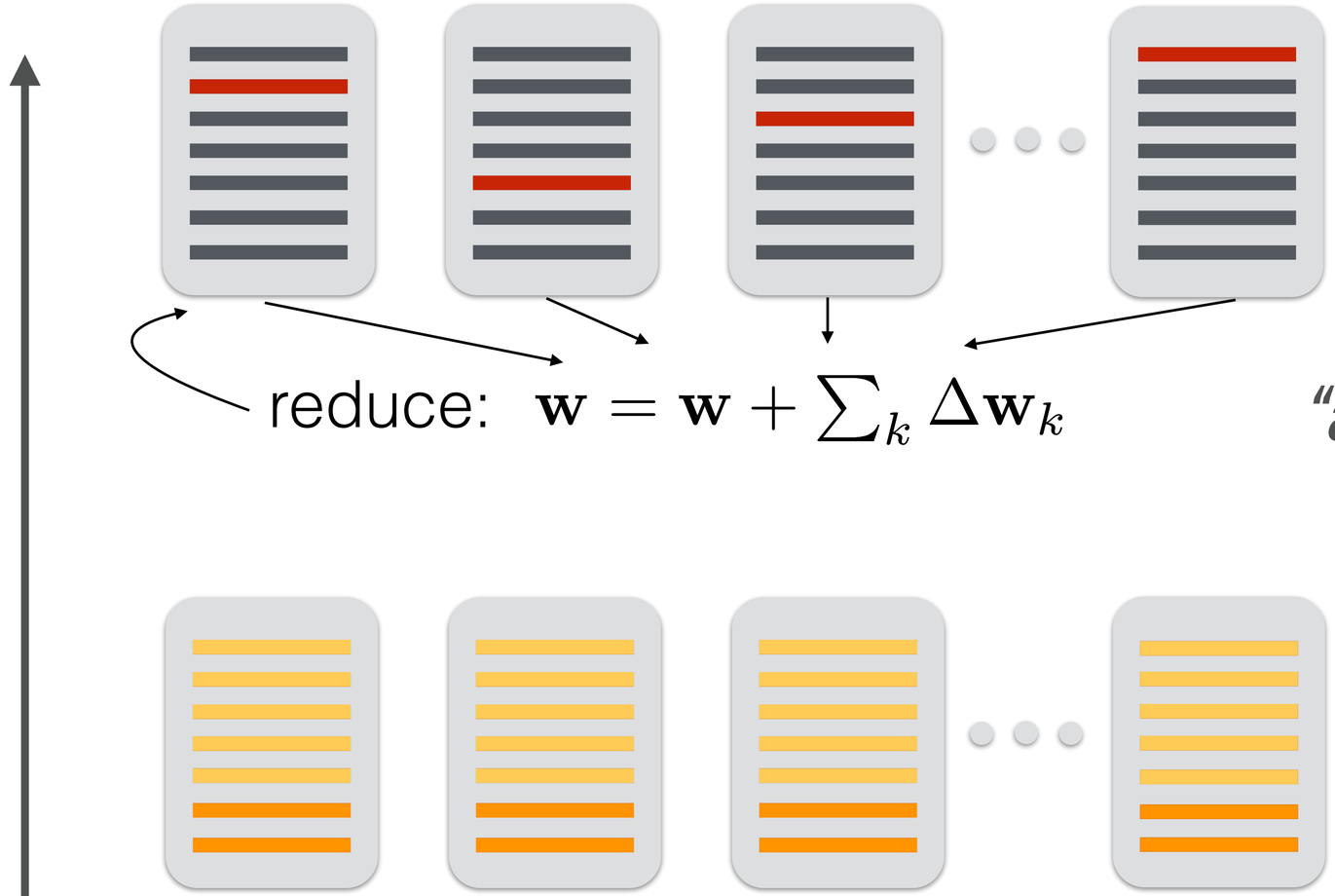


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

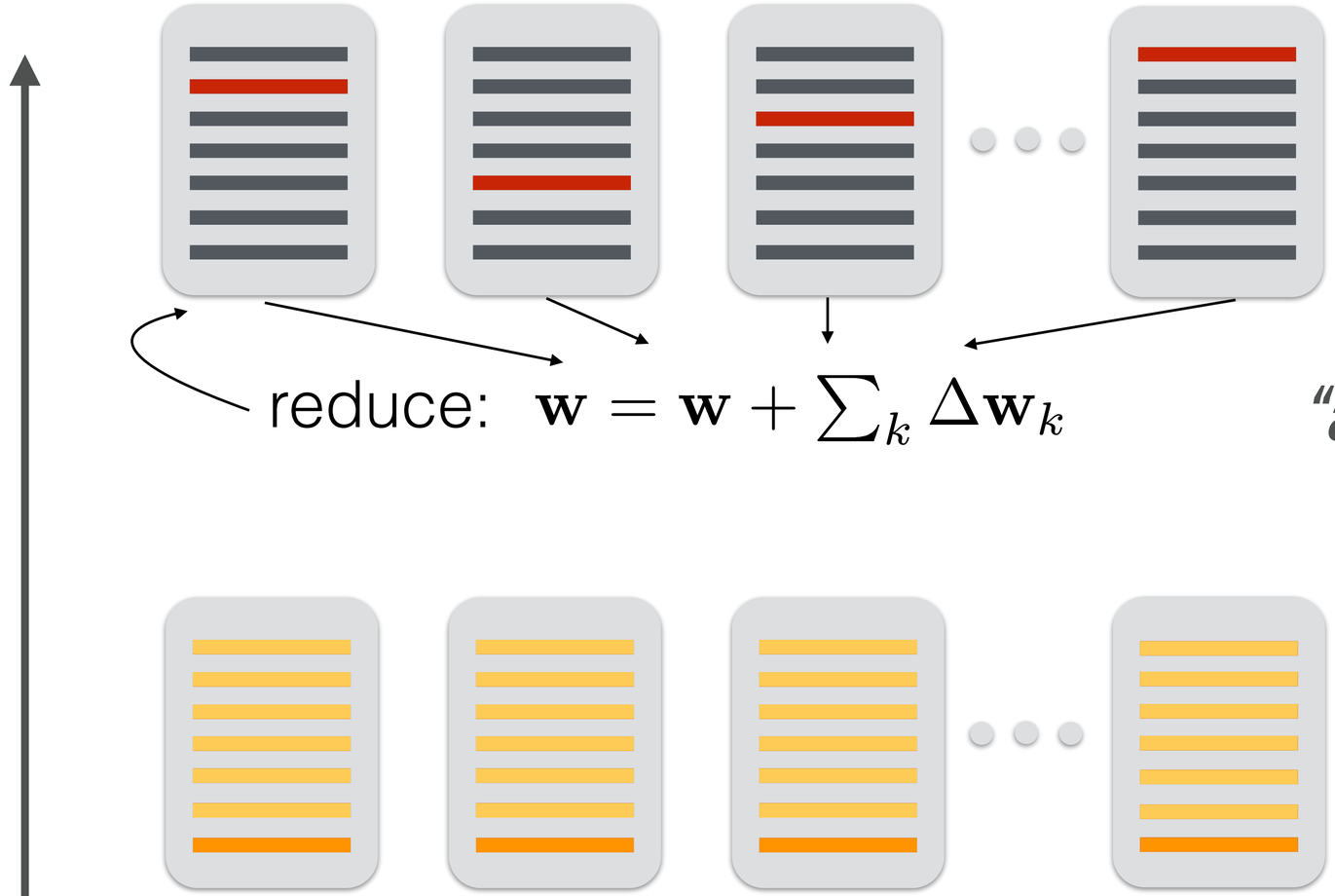


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

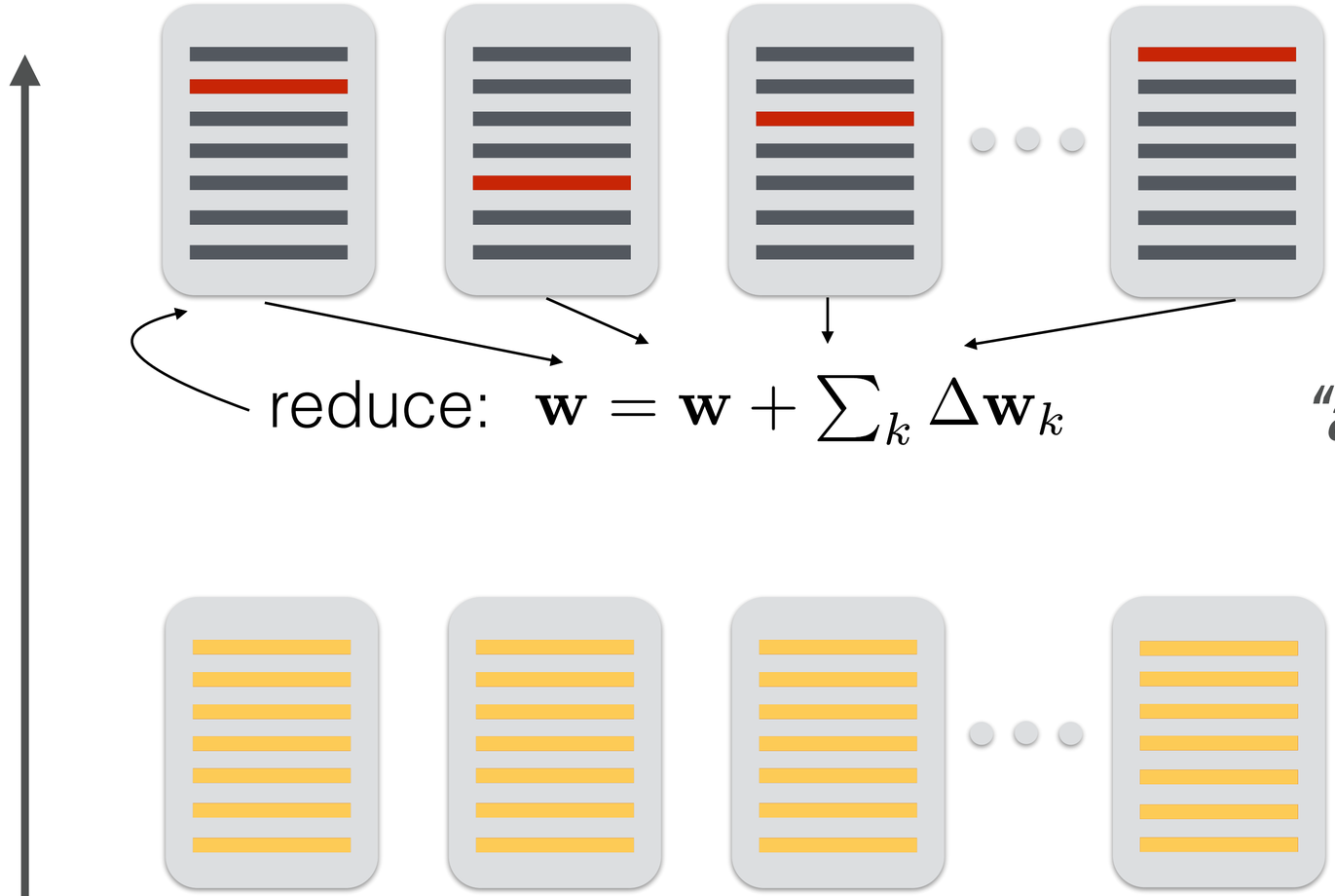


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

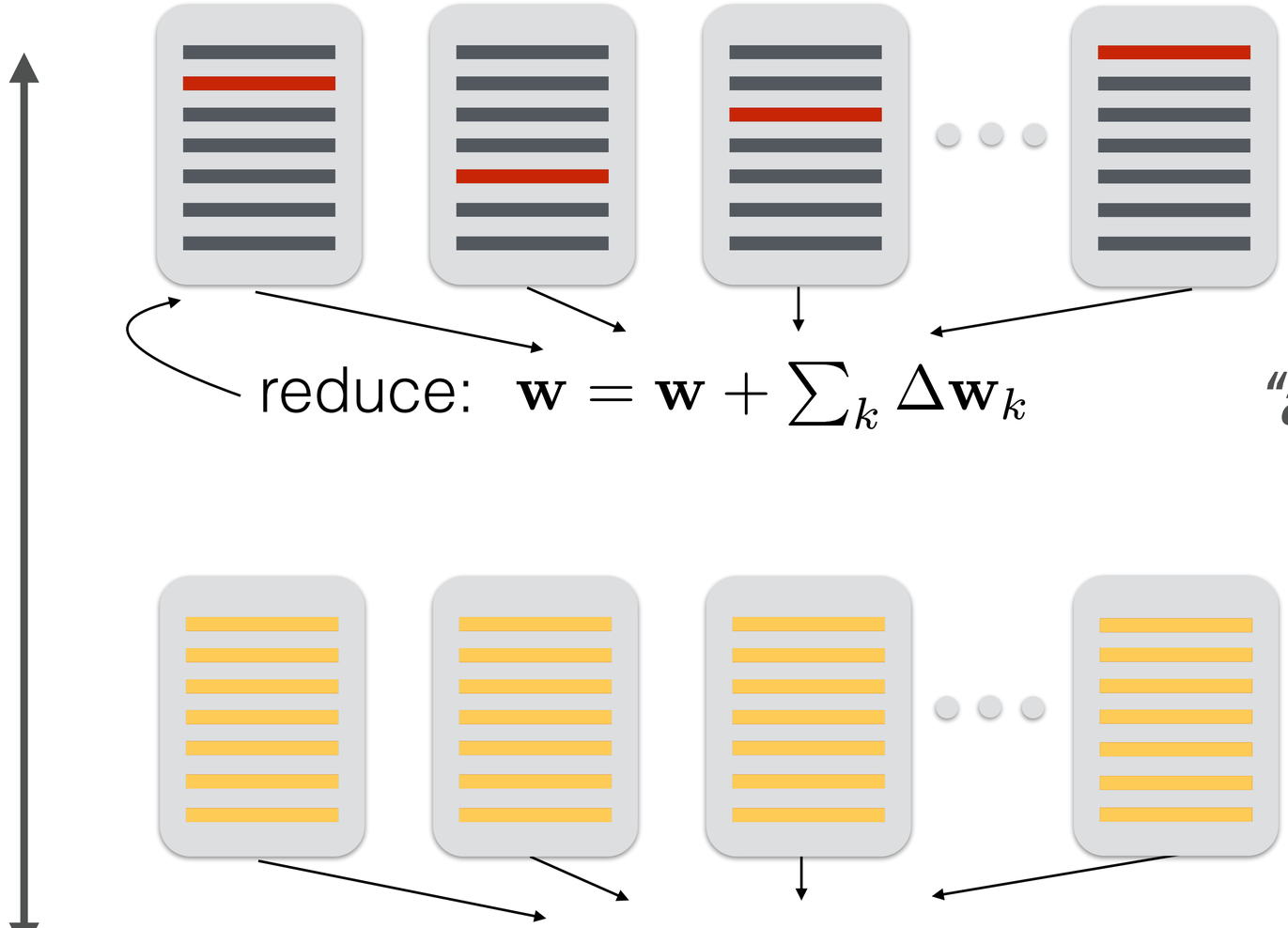


✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization



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

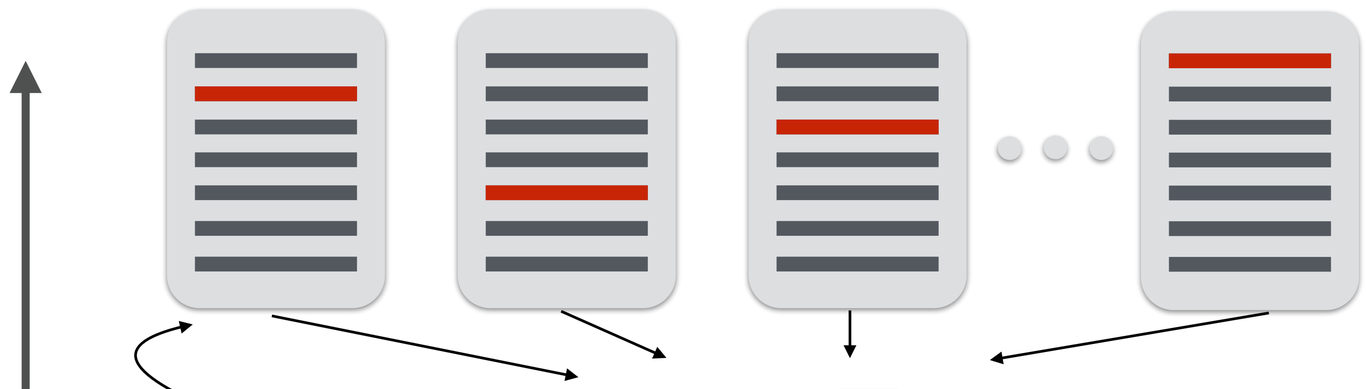
average: $\mathbf{w} := \frac{1}{K} \sum_k \mathbf{w}_k$

✔ convergence guarantees

✗ slow

"always communicate"

prior work: distributed data center optimization

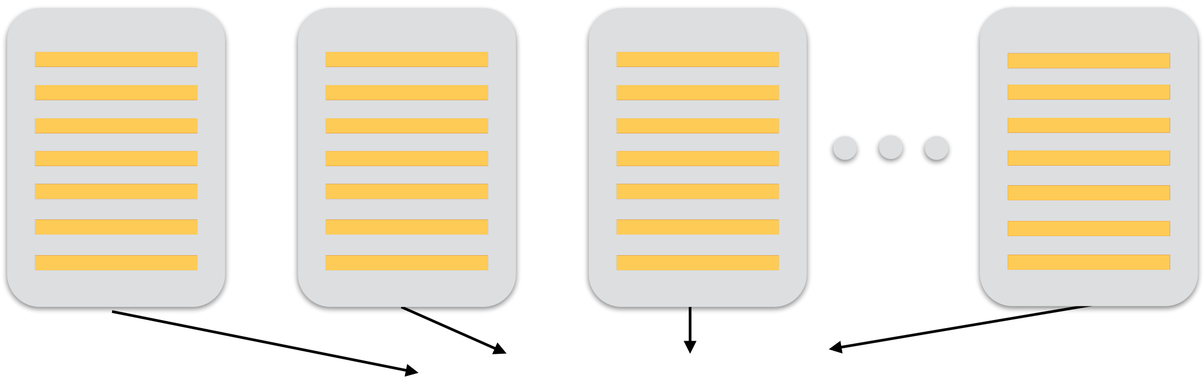


reduce: $\mathbf{w} = \mathbf{w} + \sum_k \Delta \mathbf{w}_k$

✔ convergence guarantees

✗ slow

"always communicate"



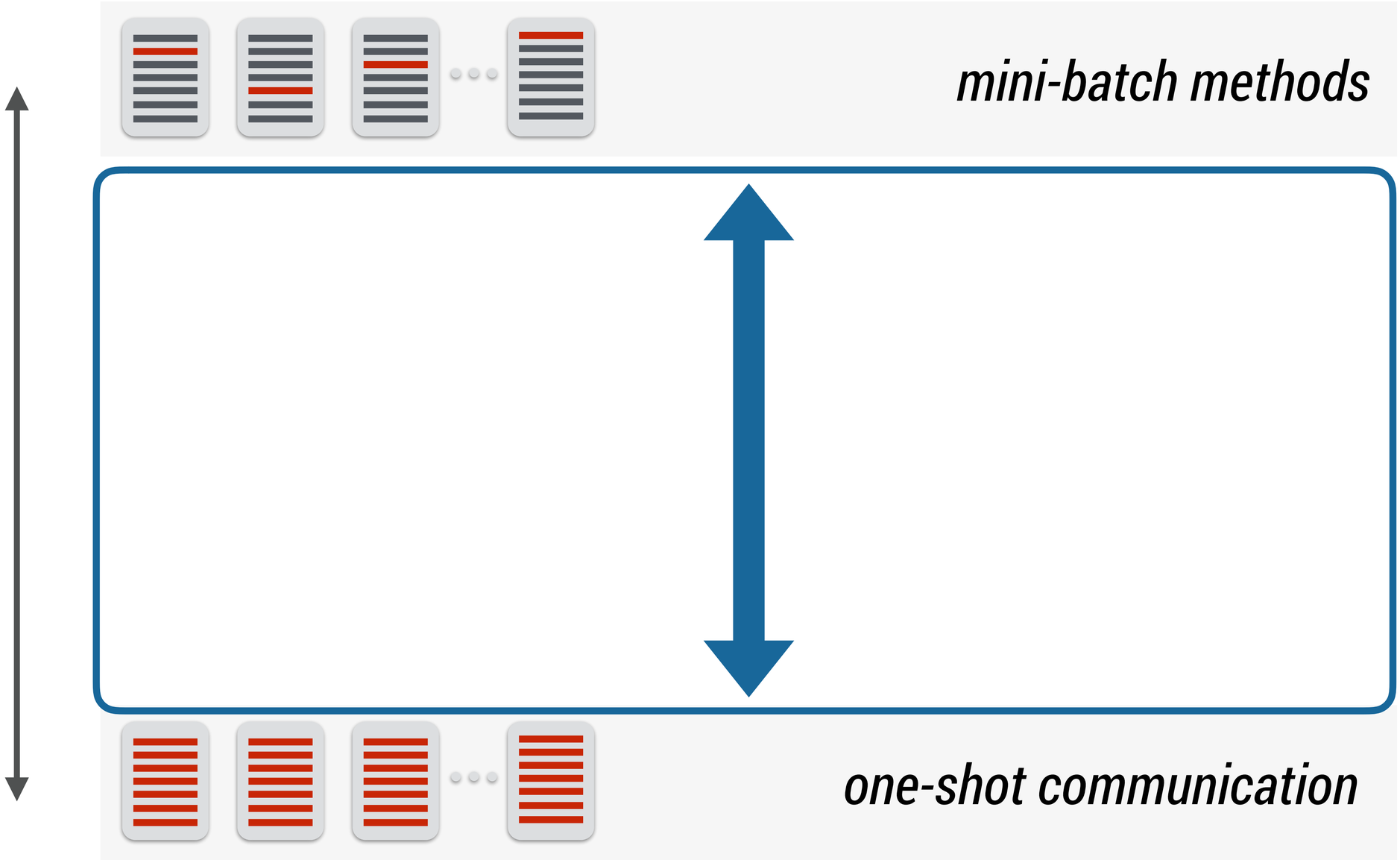
average: $\mathbf{w} := \frac{1}{K} \sum_k \mathbf{w}_k$

✔ fast

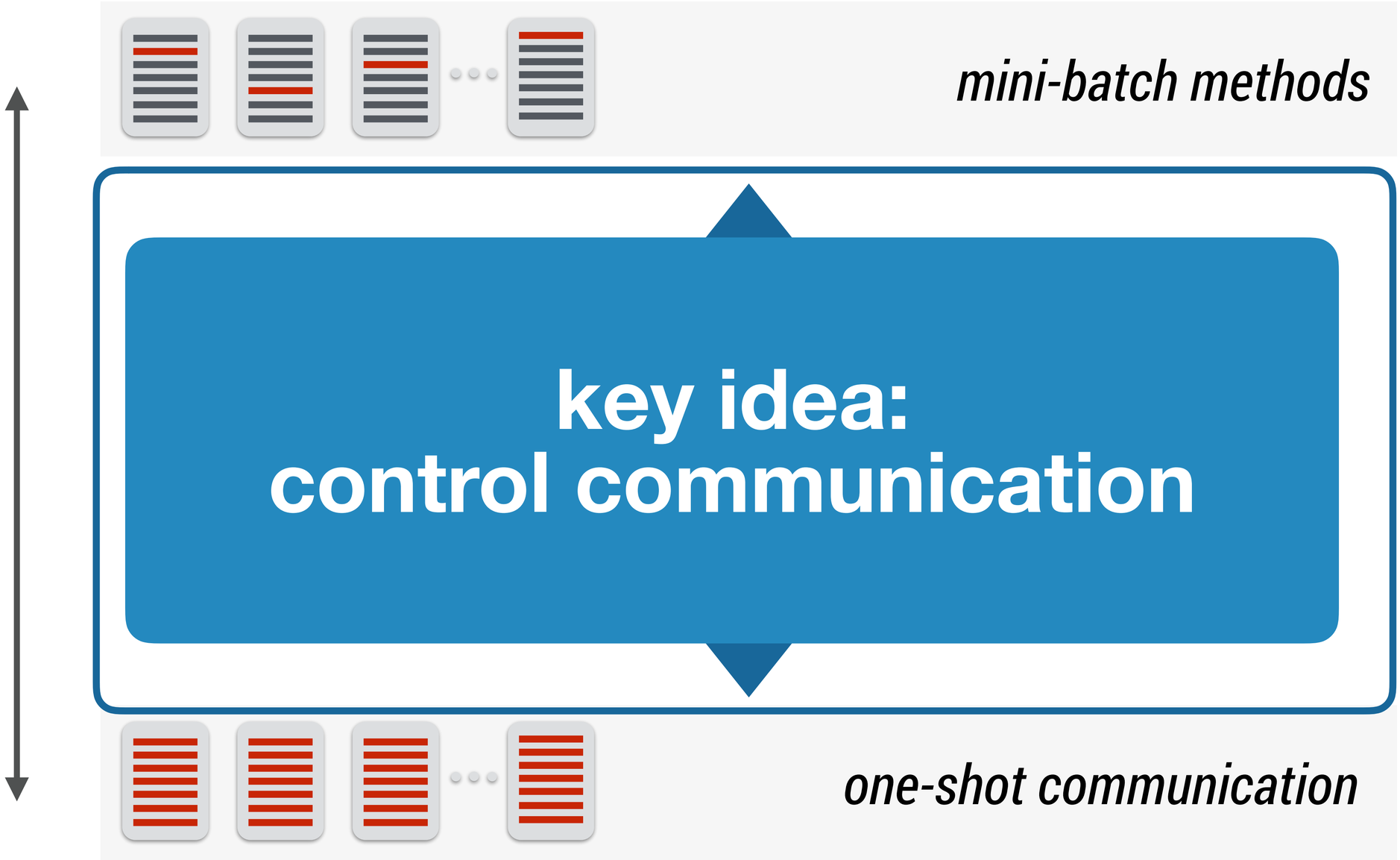
✗ convergence not guaranteed

"never communicate"

CoCoA: communication-efficient distributed optimization



CoCoA: communication-efficient distributed optimization



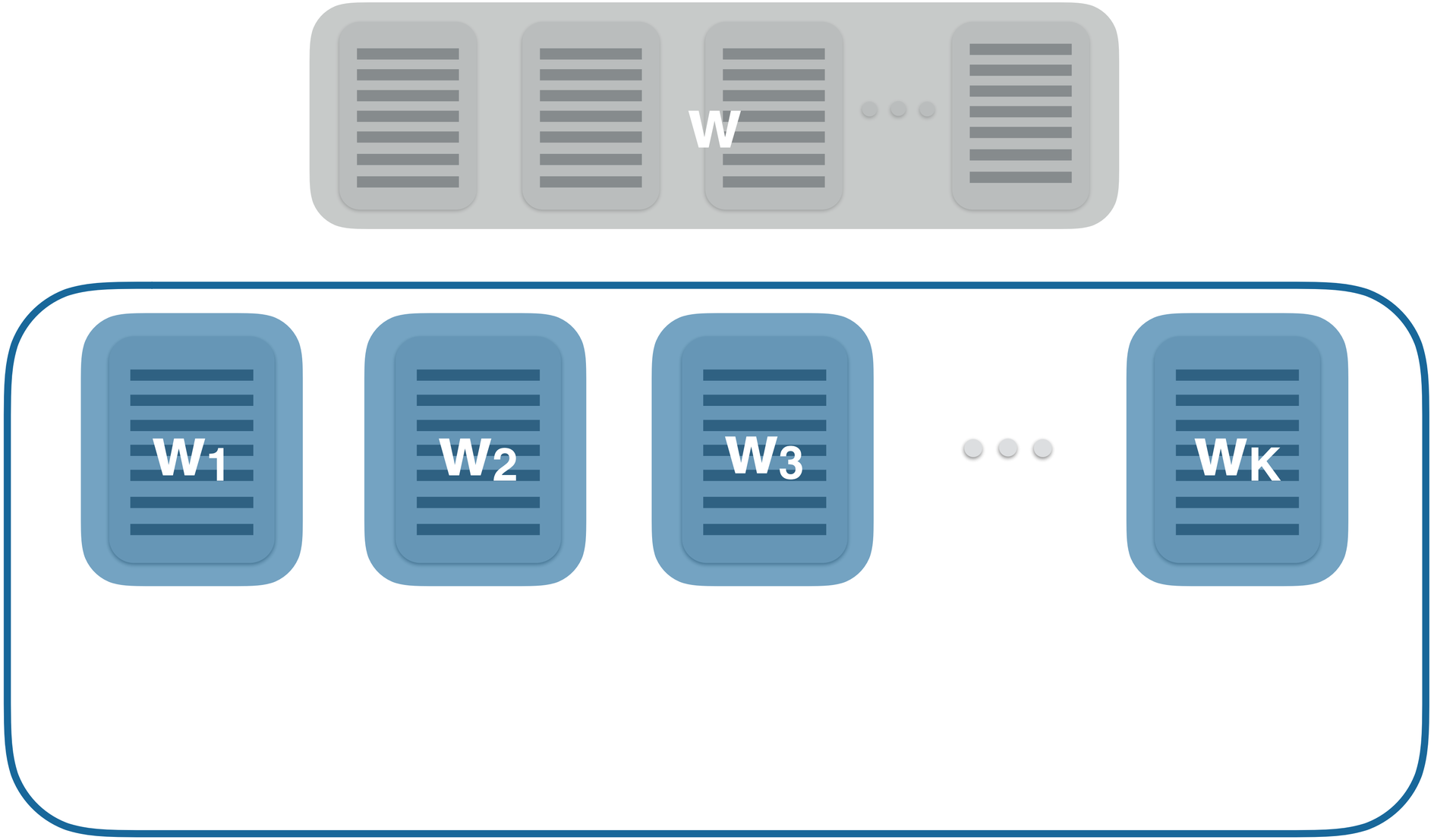
CoCoA: subproblems

CoCoA: subproblems

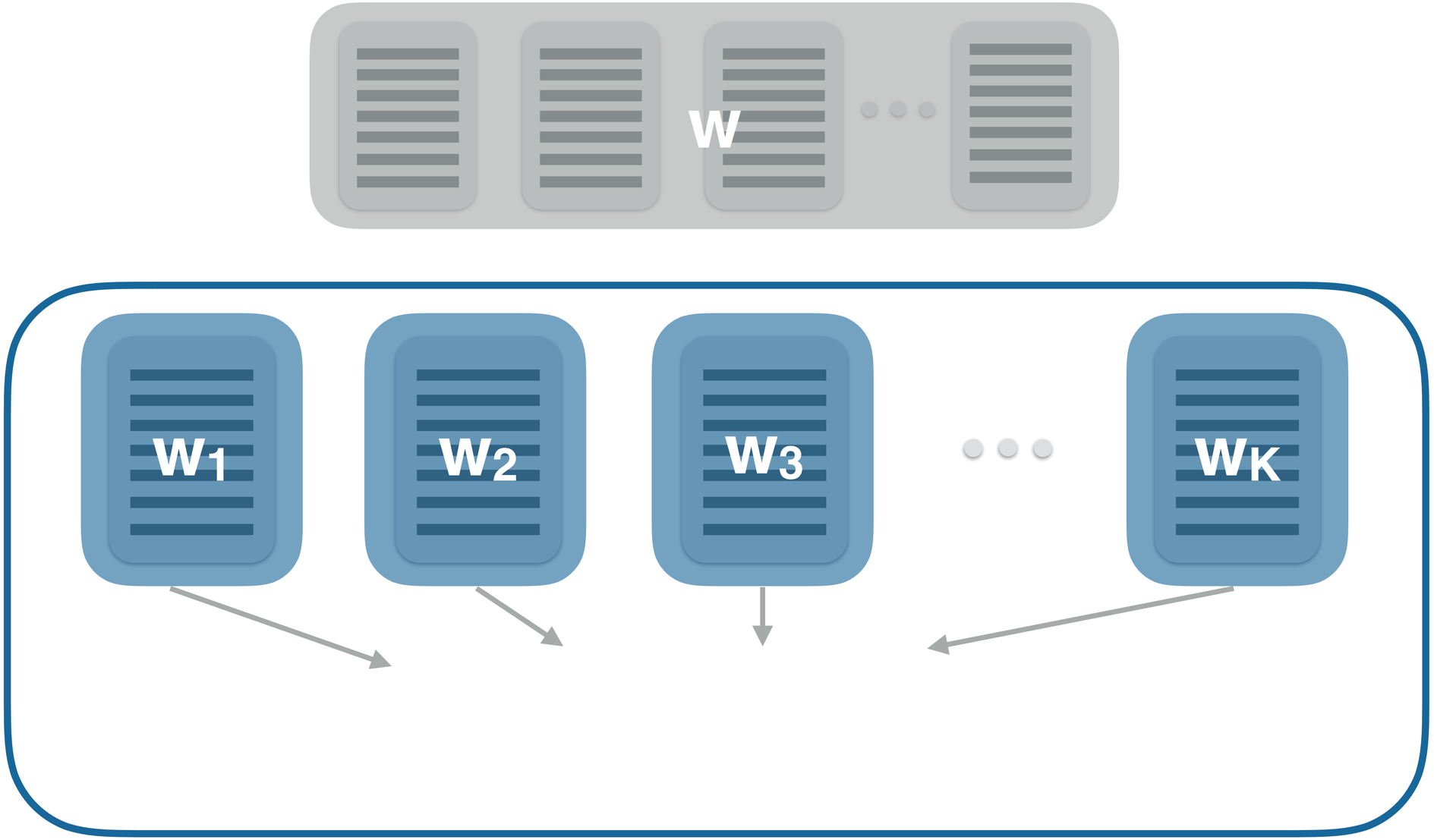
CoCoA: subproblems

CoCoA: subproblems

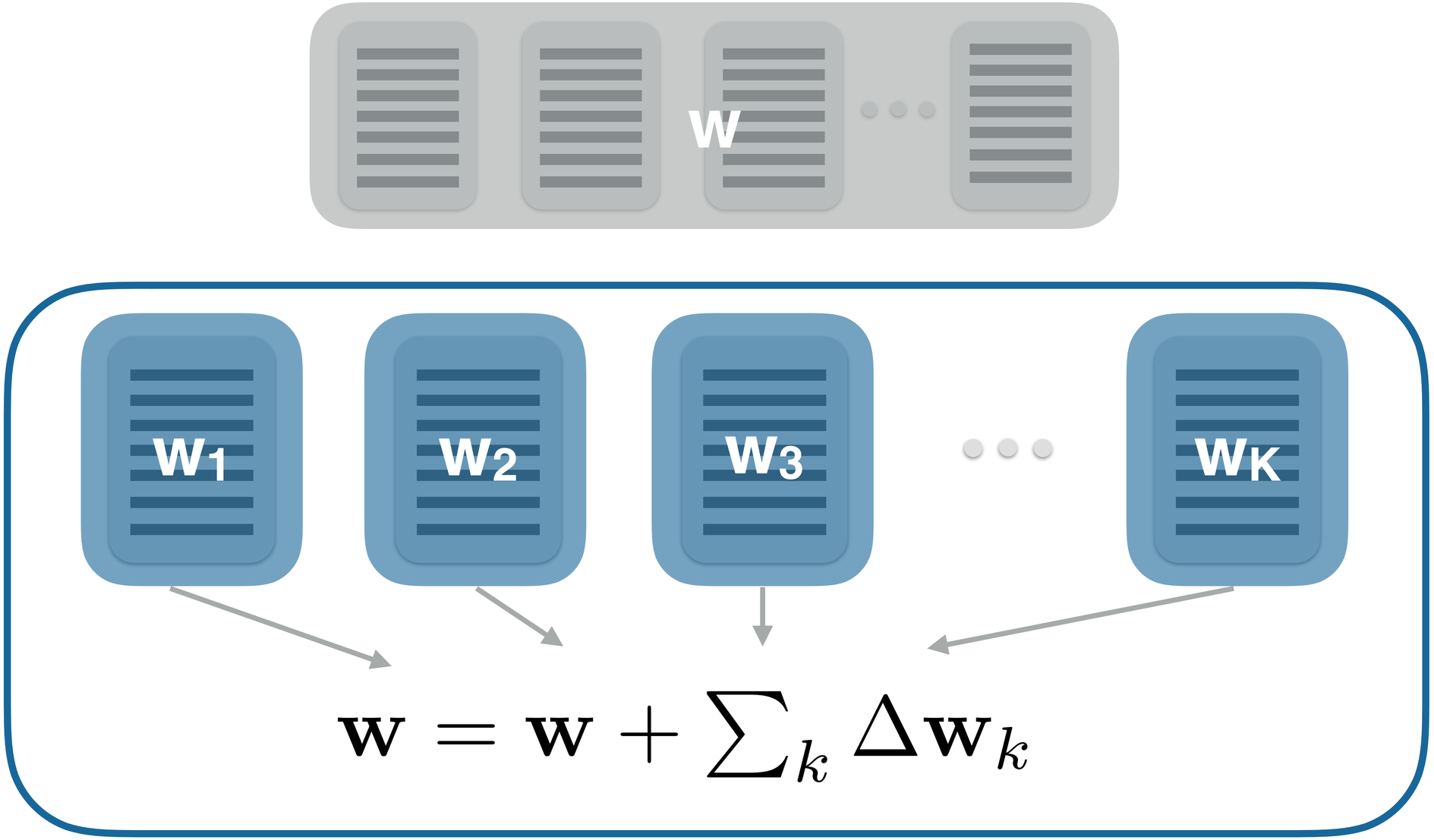
CoCoA: subproblems



CoCoA: subproblems

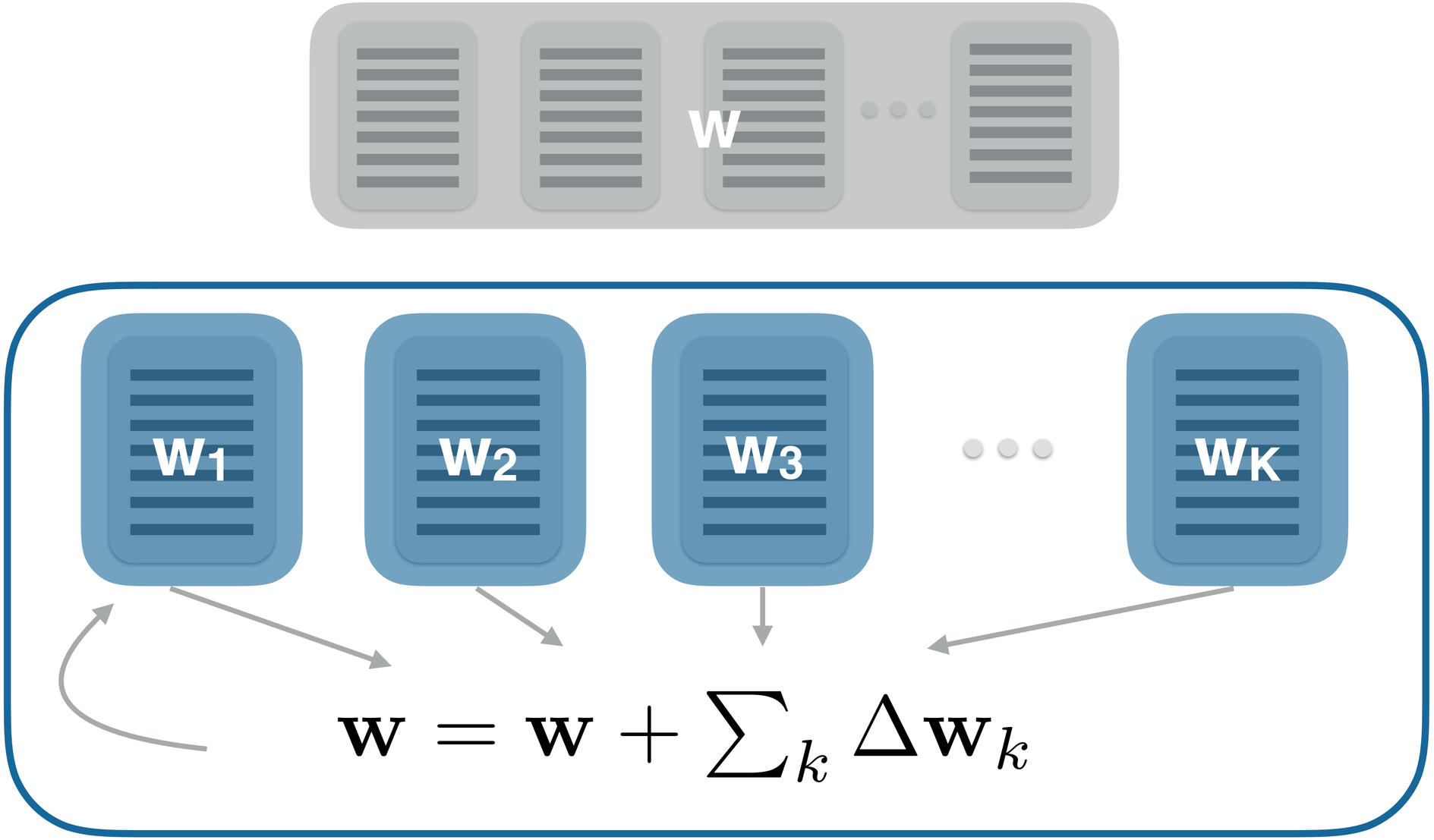


CoCoA: subproblems



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

CoCoA: subproblems



CoCoA: communication parameter

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy Θ

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy Θ

Θ

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} \in [0, 1)$$

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} \in [0, 1)$$

exactly solve

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} \in [0, 1)$$

exactly solve

inexactly solve

CoCoA: communication parameter

Main assumption:
each subproblem is solved to accuracy $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} \in [0, 1) \approx$$

amount of local
computation
vs.
communication

exactly solve

inexactly solve

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: $\theta_t^h \in [0, 1]$

each subproblem is solved to accuracy $\cancel{\theta \in [0, 1)}$

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 θ_t^h

convergence

New assumption:
each subproblem is solved to accuracy θ_t^h

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

convergence

New assumption:
each subproblem is solved to accuracy θ_t^h

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

Theorem 1. Let ℓ_t be L-Lipschitz, then

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

convergence

New assumption:
each subproblem is solved to accuracy θ_t^h

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

Theorem 1. Let ℓ_t be L-Lipschitz, then

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

1/ε rate

convergence

New assumption:
each subproblem is solved to accuracy θ_t^h

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

Theorem 1. Let ℓ_t be L-Lipschitz, then

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

1/ε rate

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

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

convergence

New assumption:
each subproblem is solved to accuracy θ_t^h

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

Theorem 1. Let ℓ_t be L-Lipschitz, then

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

1/ε rate

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

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

linear rate

federated datasets

	devices	features	min n_t	max n_t	std dev
Human Activity 	30	561	210	306	26.75
Google Glass 	38	180	524	581	11.07
Land Mine 	29	9	333	517	65.39
Vehicle Sensor 	23	100	872	1,933	267.47

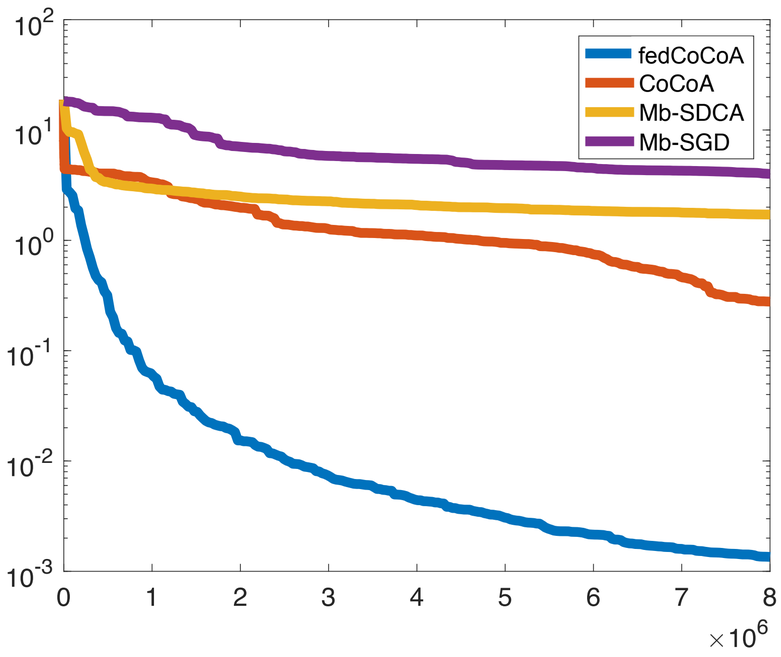
federated datasets

	devices	features	min n_t	max n_t	std dev
Human Activity 	30	561	210	306	26.75
Google Glass 	38	180	524	581	11.07
Land Mine 	29	9	333	517	65.39
Vehicle Sensor 	23	100	872	1,933	267.47

simulations

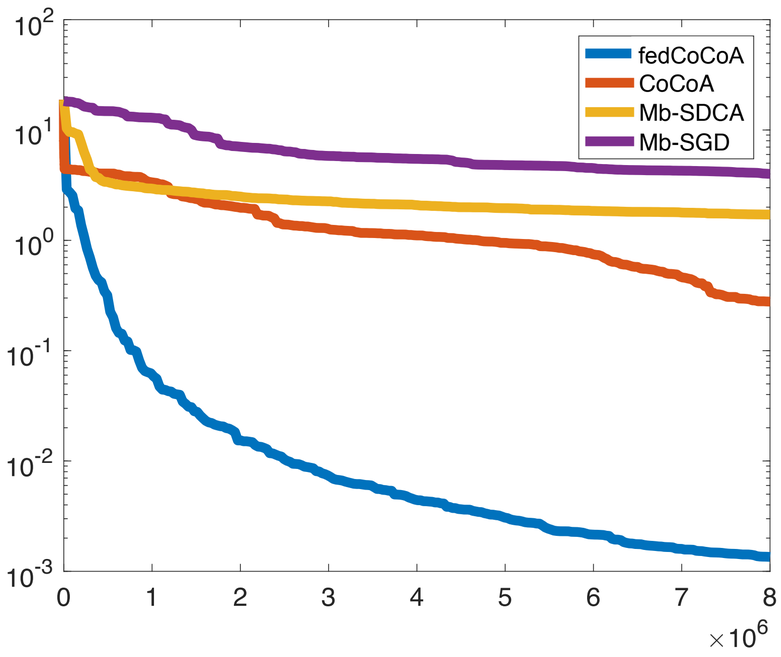
simulations

fedCoCoA converges **two orders of magnitude** more quickly than all competitors

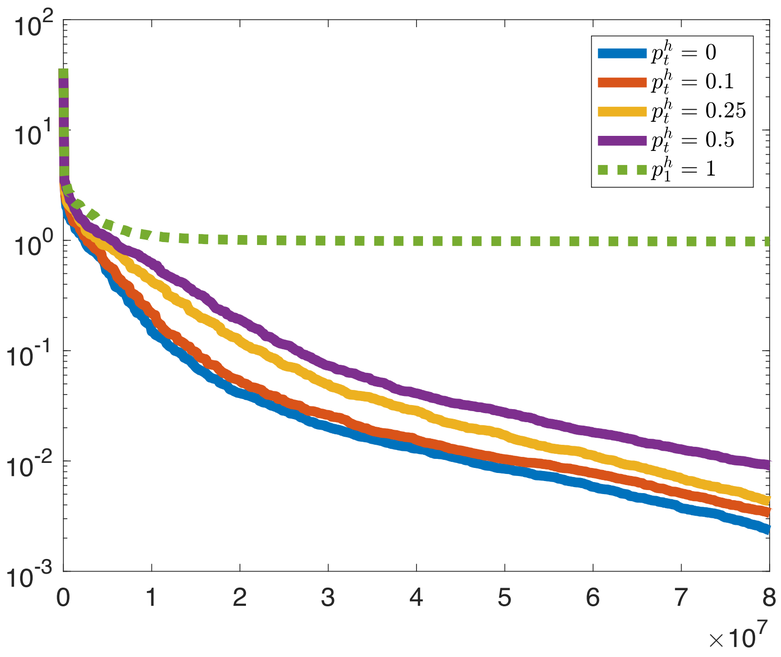


simulations

fedCoCoA converges **two orders of magnitude** more quickly than all competitors



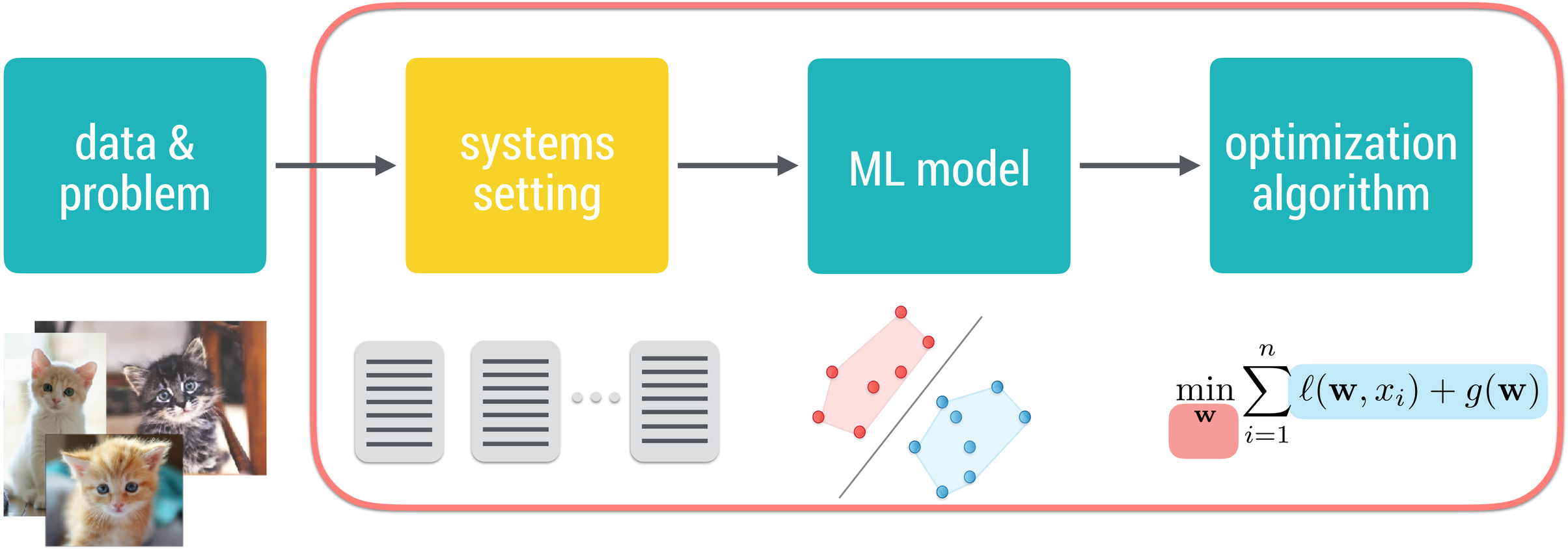
and is robust to dropped nodes



MOCHA: personalization

[Smith, Chiang, Sanjabi, Talwalkar, NIPS 2017]

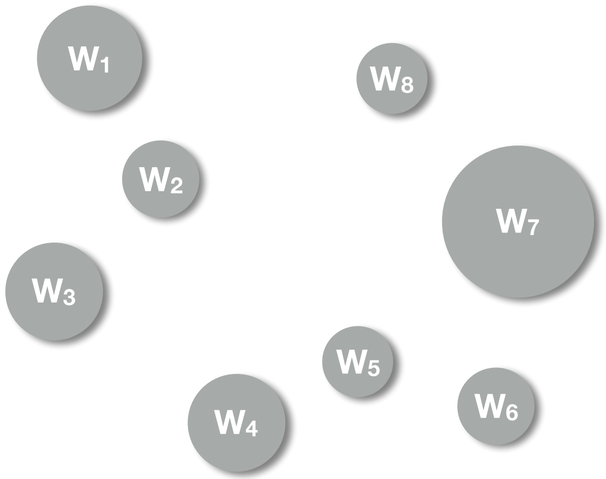
system-aware optimization



how to model federated data?

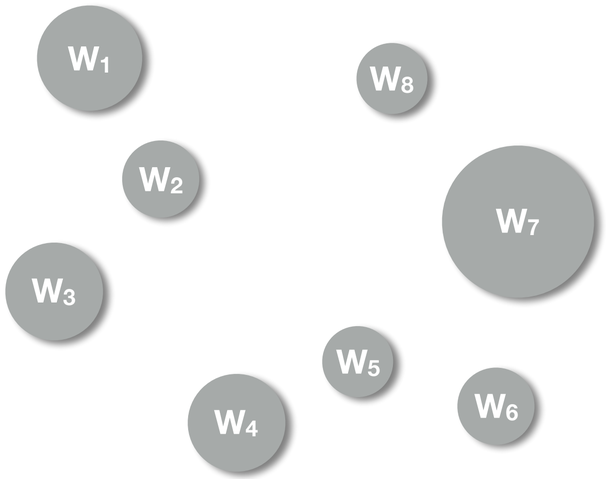
how to model federated data?

local



how to model federated data?

local

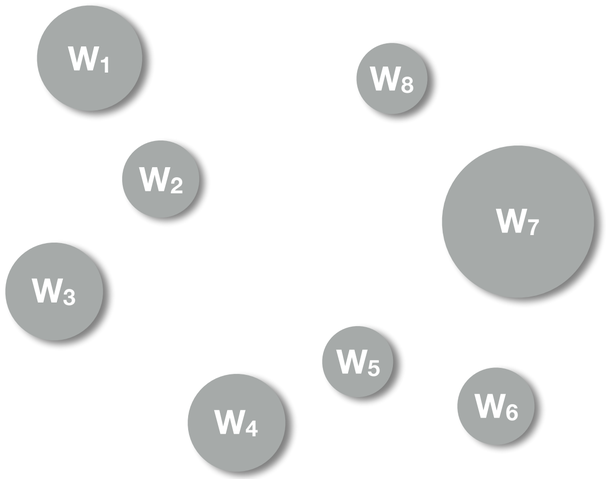


✓ *personalized models*

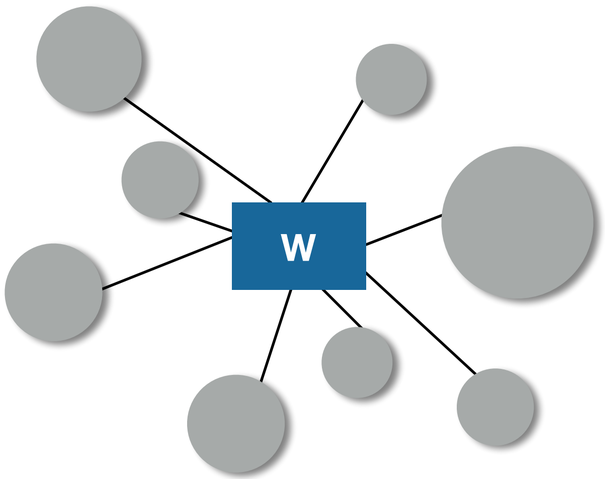
✗ *don't learn from peers*

how to model federated data?

local



global

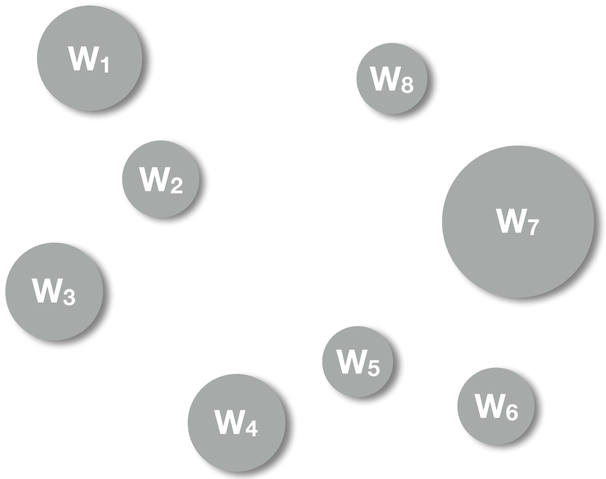


✓ *personalized models*

✗ *don't learn from peers*

how to model federated data?

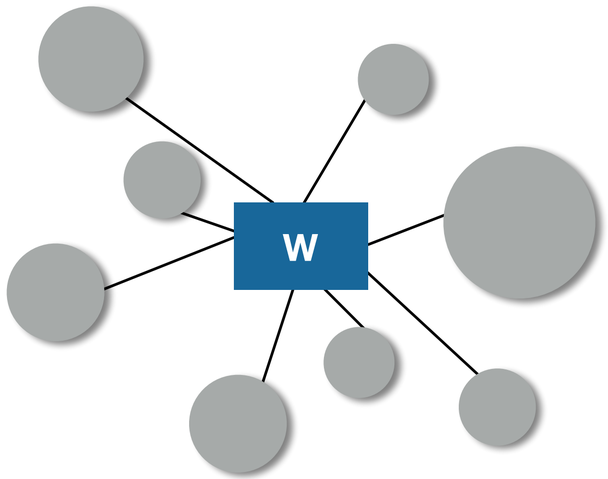
local



✓ *personalized models*

✗ *don't learn from peers*

global

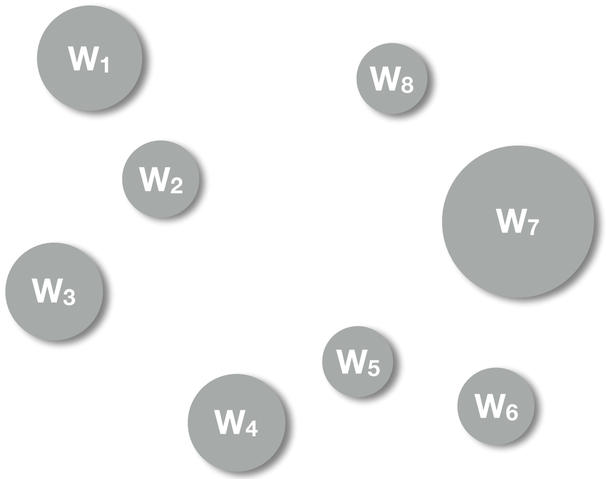


✗ *non-personalized models*

✓ *learn from peers*

how to model federated data?

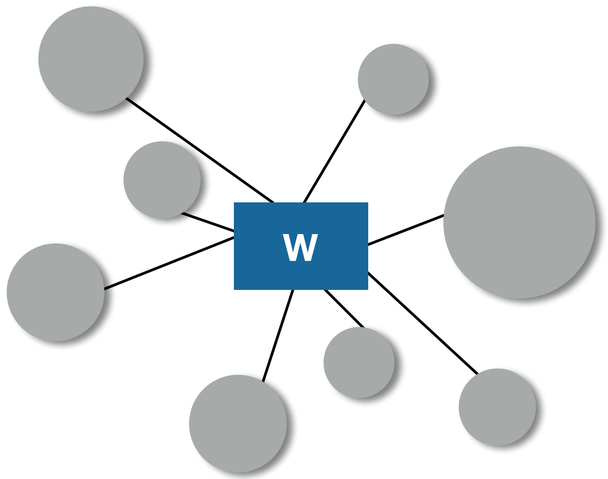
local



✓ *personalized models*

✗ *don't learn from peers*

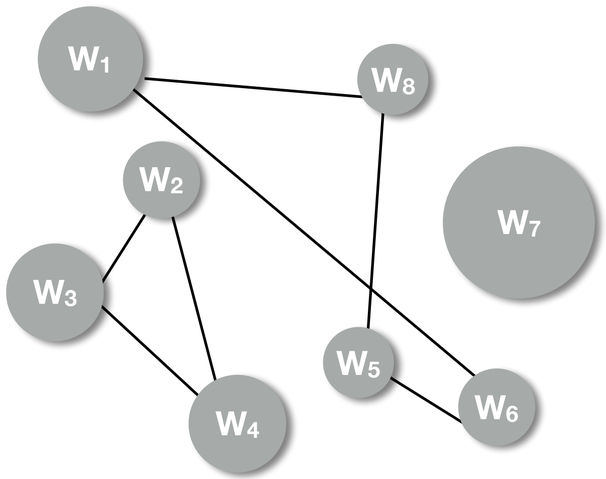
global



✗ *non-personalized models*

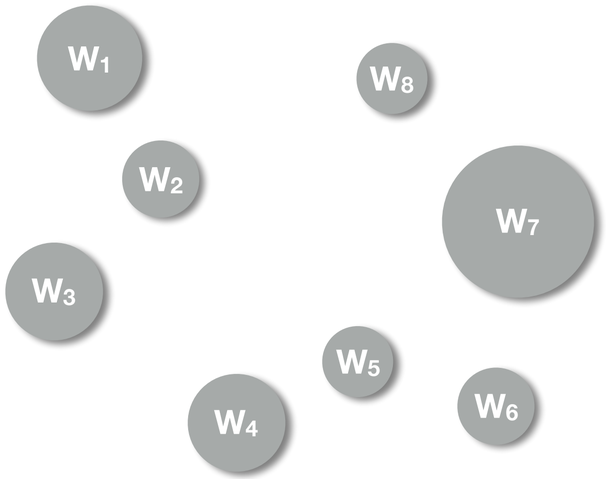
✓ *learn from peers*

??



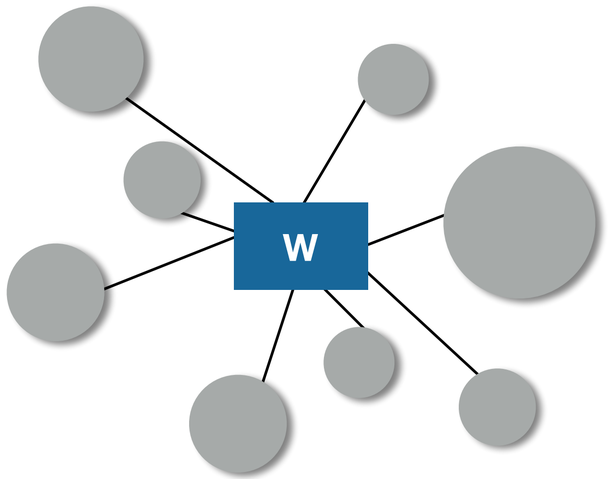
how to model federated data?

local



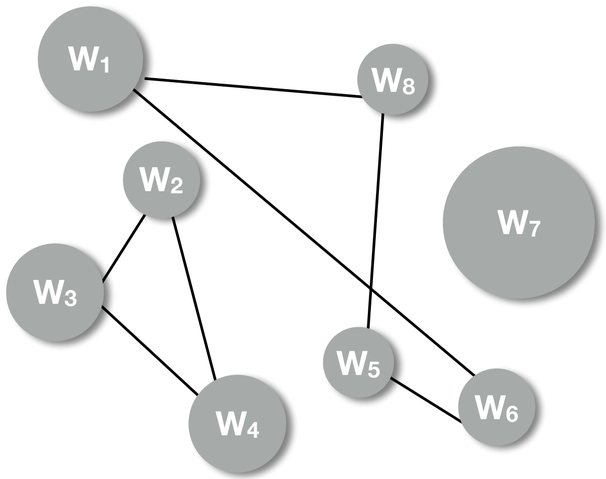
✓ *personalized models*
✗ *don't learn from peers*

global



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

??



✓ *personalized models*
✓ *learn from peers*

multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \underbrace{\ell_t(\mathbf{w}_t, \mathbf{x}_t^i)}_{\text{losses}} + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

models

losses

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

models

losses

regularizer

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

models

task relationship

losses

regularizer

[EP, KDD 2004; JVB, NIPS 2009; ZCY, NIPS 2011]

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

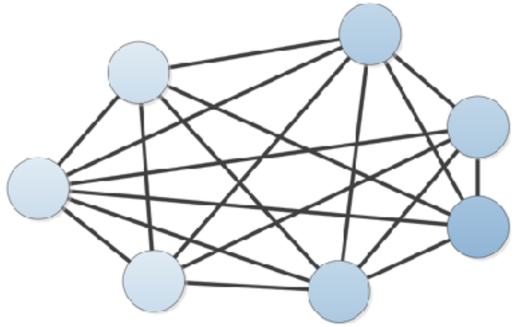
models

task relationship

losses

regularizer

all tasks related



[EP, KDD 2004; JVB, NIPS 2009; ZCY, NIPS 2011]

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

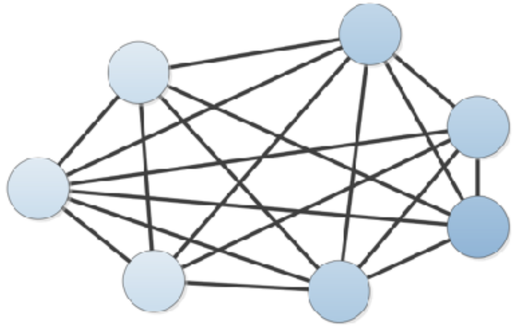
models

task relationship

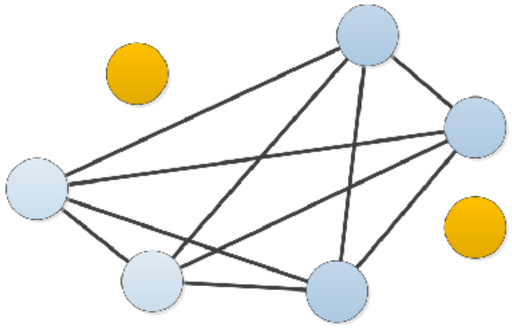
losses

regularizer

all tasks related



outlier tasks



[EP, KDD 2004; JVB, NIPS 2009; ZCY, NIPS 2011]

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

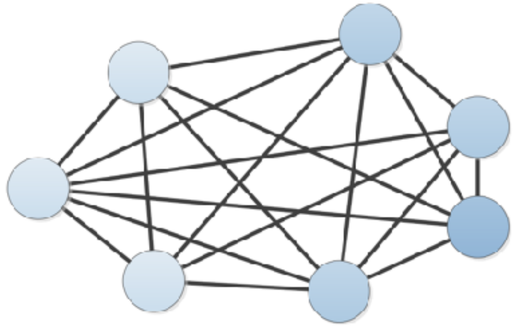
models

task relationship

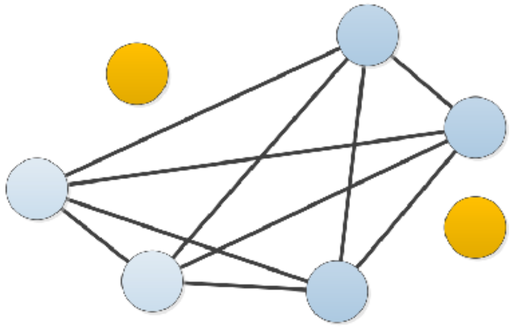
losses

regularizer

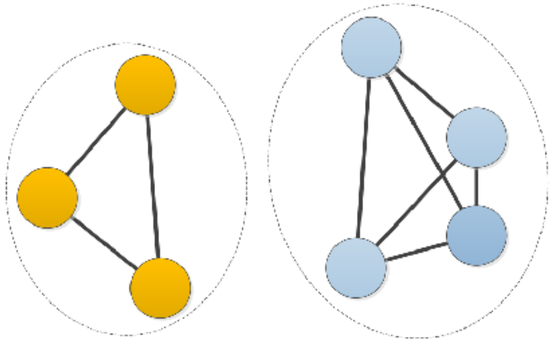
all tasks related



outlier tasks



clusters / groups



[EP, KDD 2004; JVB, NIPS 2009; ZCY, NIPS 2011]

multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t, \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

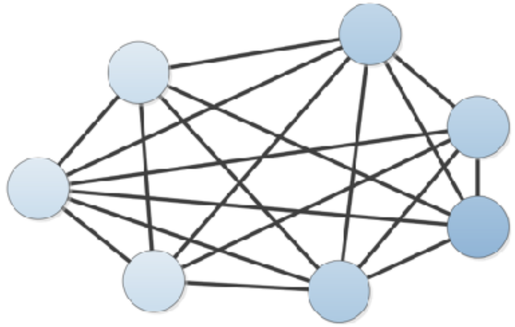
models

task relationship

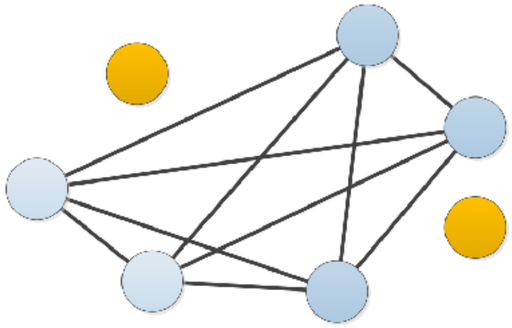
losses

regularizer

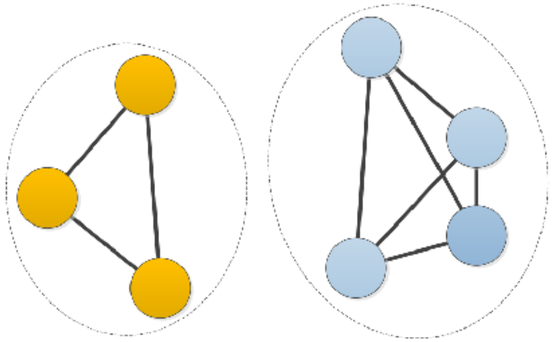
all tasks related



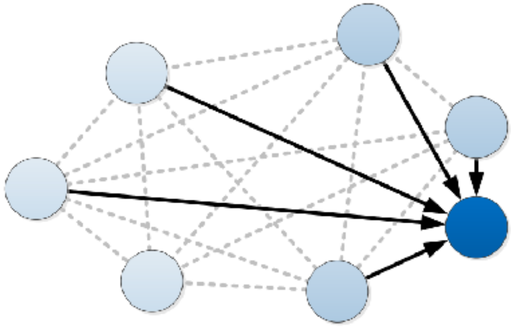
outlier tasks



clusters / groups



asymmetry



[EP, KDD 2004; JVB, NIPS 2009; ZCY, NIPS 2011]

example: prediction error on federated data

	Global	Local	MTL
Human Activity 	2.23 (0.30)	1.34 (0.21)	**0.46 (0.11)**
Google Glass 	5.34 (0.26)	4.92 (0.26)	**2.02 (0.15)**
Land Mine 	27.72 (1.08)	23.43 (0.77)	**20.09 (1.04)**
Vehicle Sensor 	13.4 (0.26)	7.81 (0.13)	**6.59 (0.21)**

example: prediction error on federated data

	Global	Local	MTL
Human Activity 	2.23 (0.30)	1.34 (0.21)	**0.46 (0.11)**
Google Glass 	5.34 (0.26)	4.92 (0.26)	**2.02 (0.15)**
Land Mine 	27.72 (1.08)	23.43 (0.77)	**20.09 (1.04)**
Vehicle Sensor 	13.4 (0.26)	7.81 (0.13)	**6.59 (0.21)**

MOCHA: federated optimization for multi-task learning

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

- Key idea: Solve for W, $\mathbf{\Omega}$ in an alternating fashion

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

- Key idea: Solve for W, $\mathbf{\Omega}$ in an alternating fashion
  - $\mathbf{\Omega}$ can be updated centrally

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

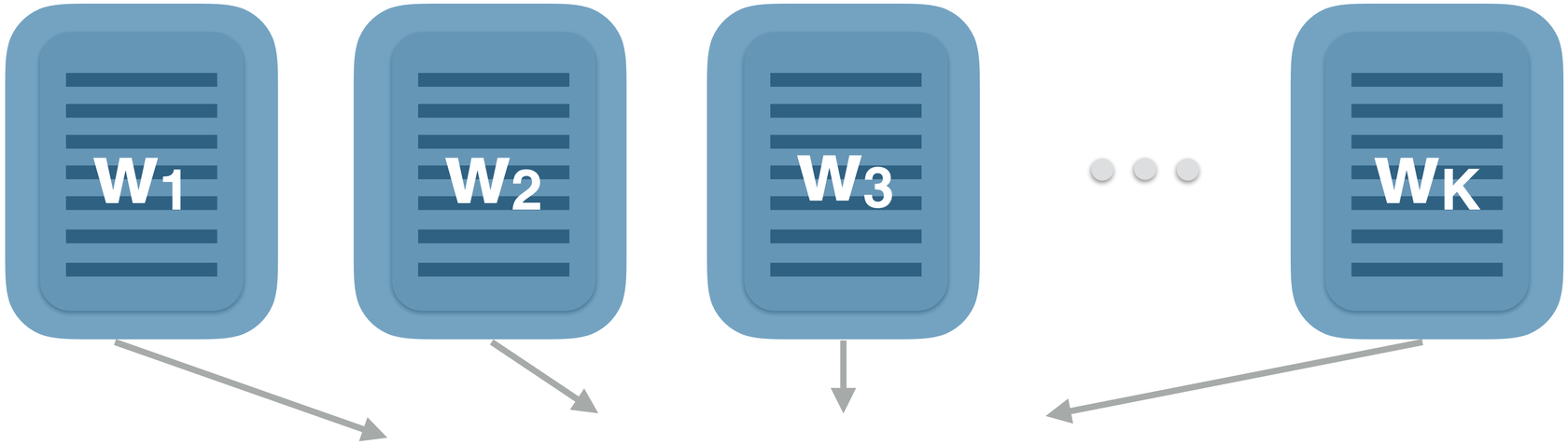
- Key idea: Solve for W, $\mathbf{\Omega}$ in an alternating fashion
  - $\mathbf{\Omega}$ can be updated centrally
  - *W needs to be solved in federated setting*

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

- Key idea: Solve for W, $\mathbf{\Omega}$ in an alternating fashion
  - $\mathbf{\Omega}$ 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

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

FedCoCoA: primal-dual framework

PRIMAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

FedCoCoA: primal-dual framework

PRIMAL

$\geq$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w}) \qquad \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) - \lambda g^*(X, \boldsymbol{\alpha})$$

FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w}) \qquad \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) - \cancel{\lambda g^*(X, \boldsymbol{\alpha})}$$

FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

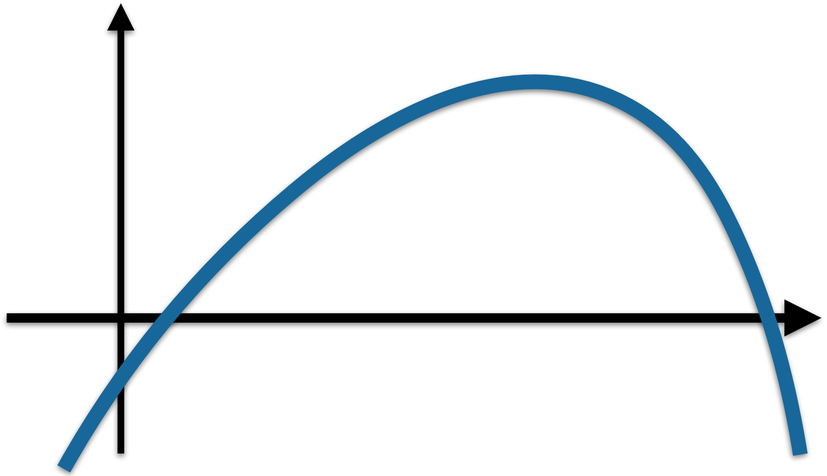
$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) \overset{\sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \boldsymbol{\alpha}_{[k]})}{-} \cancel{\lambda g^*(X, \boldsymbol{\alpha})}$$

FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) \overset{\sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \boldsymbol{\alpha}_{[k]})}{\cancel{-\lambda g^*(X, \boldsymbol{\alpha})}}$$

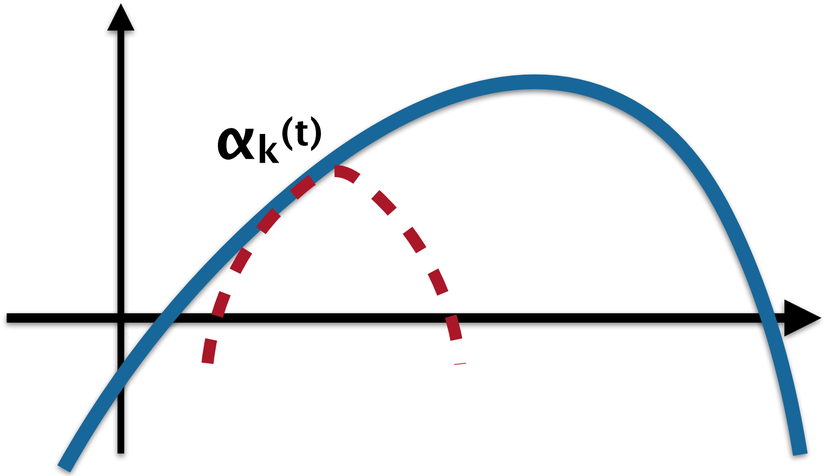


FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) - \cancel{\lambda g^*(X, \boldsymbol{\alpha})} \sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \boldsymbol{\alpha}_{[k]})$$

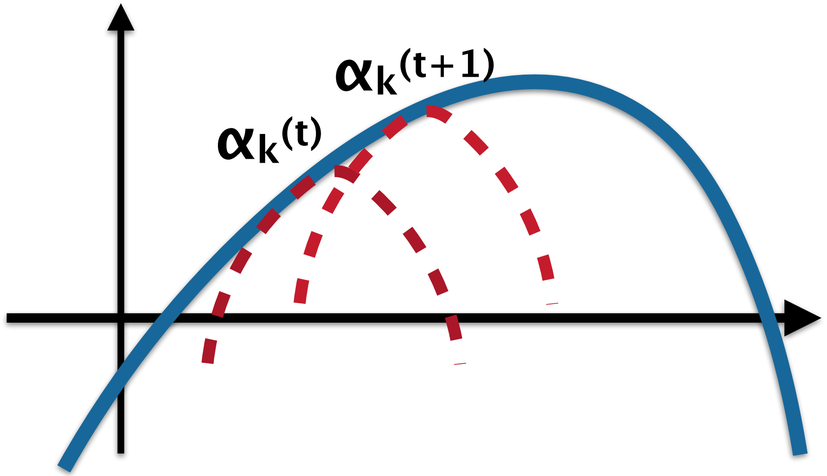


FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) - \cancel{\lambda g^*(X, \boldsymbol{\alpha})} \sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \boldsymbol{\alpha}_{[k]})$$

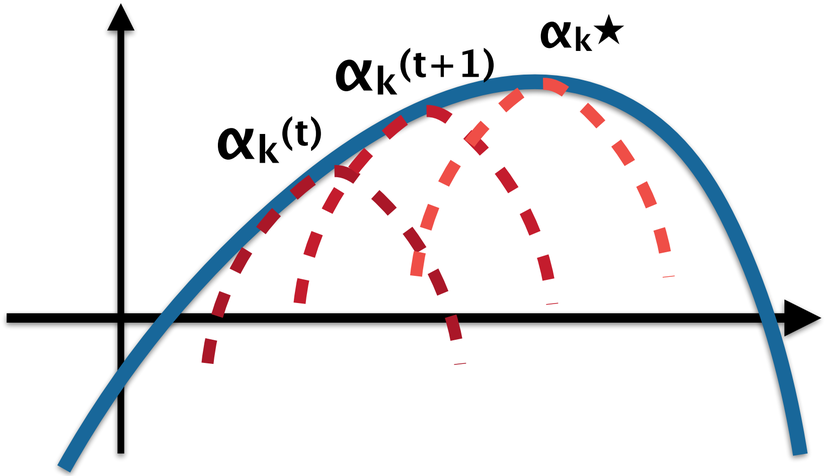


FedCoCoA: primal-dual framework

PRIMAL $\geq$ DUAL

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^T x_i) + \lambda g(\mathbf{w})$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^{n} \ell^*(-\alpha_i) - \cancel{\lambda g^*(X, \boldsymbol{\alpha})} \sum_{k=1}^{K} \tilde{g}^*(X_{[k]}, \boldsymbol{\alpha}_{[k]})$$

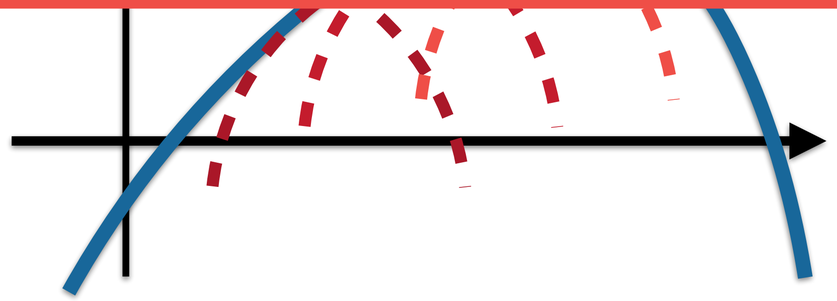


FedCoCoA: primal-dual framework

$[k], \boldsymbol{\alpha}_{[k]})$

$\cancel{X, \boldsymbol{\alpha})}$

challenge:
extend to MTL setup



MOCHA: federated optimization for multi-task learning

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

- Solve for W, $\mathbf{\Omega}$ in an alternating fashion

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega})$$

- Solve for W, $\mathbf{\Omega}$ in an alternating fashion
- Modify FedCoCoA to solve W in the federated setting

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

- Solve for W, $\boldsymbol{\Omega}$ in an alternating fashion
- Modify FedCoCoA to solve W in the federated setting

dual $$\min_{\boldsymbol{\alpha}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i) + \mathcal{R}^*(\mathbf{X}\boldsymbol{\alpha})$$

MOCHA: federated optimization for multi-task learning

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega})$$

- Solve for W, $\boldsymbol{\Omega}$ in an alternating fashion
- Modify FedCoCoA to solve W in the federated setting

dual $$\min_{\boldsymbol{\alpha}} \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i) + \mathcal{R}^*(\mathbf{X}\boldsymbol{\alpha})$$

subproblem $$\min_{\Delta\boldsymbol{\alpha}_t} \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i - \Delta\boldsymbol{\alpha}_t^i) + \langle \mathbf{w}_t(\boldsymbol{\alpha}), \mathbf{X}_t \Delta\boldsymbol{\alpha}_t \rangle + \frac{\sigma'}{2} \|\mathbf{X}_t \Delta\boldsymbol{\alpha}_t\|_{\mathbf{M}_t}^2$$

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

FedCoCoA

personalization is key

- non-IID data, underlying statistical structure

MOCHA

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

SysML 2019

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

SysML 2019

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

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

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!

Microsoft