Federated Machine Learning via *Over-the-Air Computation*

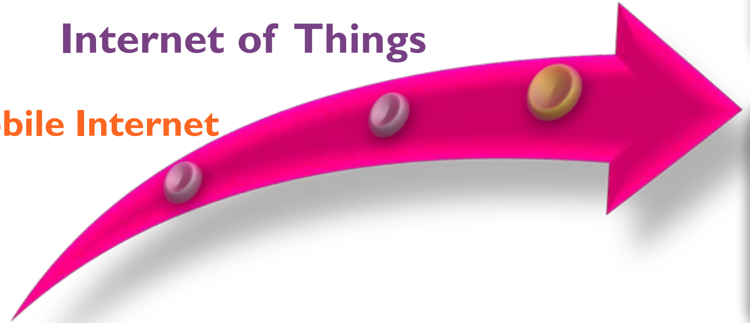
Yuanming Shi

ShanghaiTech University

Outline

- **Motivations**
  - Big data, IoT, AI
- **Three vignettes:**
  - **Federated machine learning**
    - Federated model aggregation
  - **Over-the-air computation**
    - Joint device selection and beamforming design
  - **Sparse and low-rank optimization**
    - Difference-of-convex programming algorithm

Intelligent IoT ecosystem



Develop computation, communication & AI technologies:
enable smart IoT applications to make ***low-latency decision*** on streaming data

Intelligent IoT applications

Autonomous vehicles

Smart home

Smart city

Smart health

Smart agriculture

Smart drones

Challenges

- Retrieve or infer information from high-dimensional/large-scale data

2.5 exabytes of data
are generated every day (2012)

exabyte → zettabyte → yottabyte...??

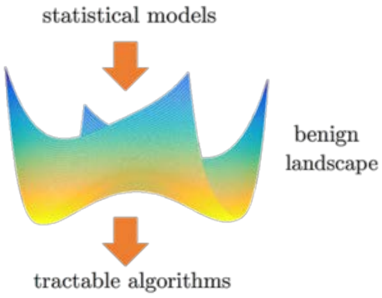
We're interested in the ***information*** rather than the data

limited processing ability
(computation, storage, ...)

Challenges:

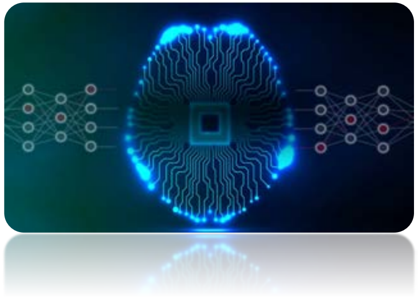
- High computational cost
- Only limited memory is available
- Do NOT want to compromise statistical accuracy

High-dimensional data analysis



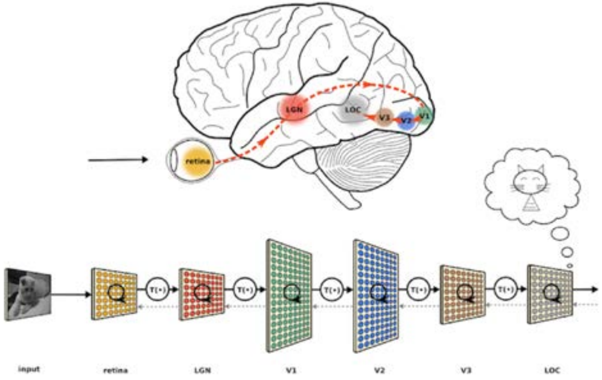
(big) data

Models: (deep) machine learning



Methods:
1. Large-scale optimization
2. High-dimensional statistics
3. Device-edge-cloud computing

Deep learning: next wave of AI



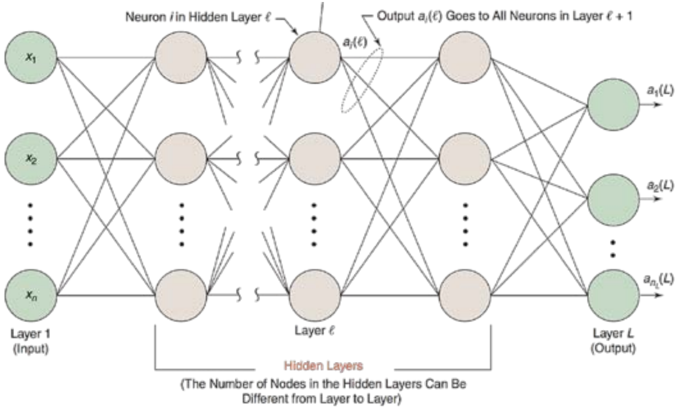


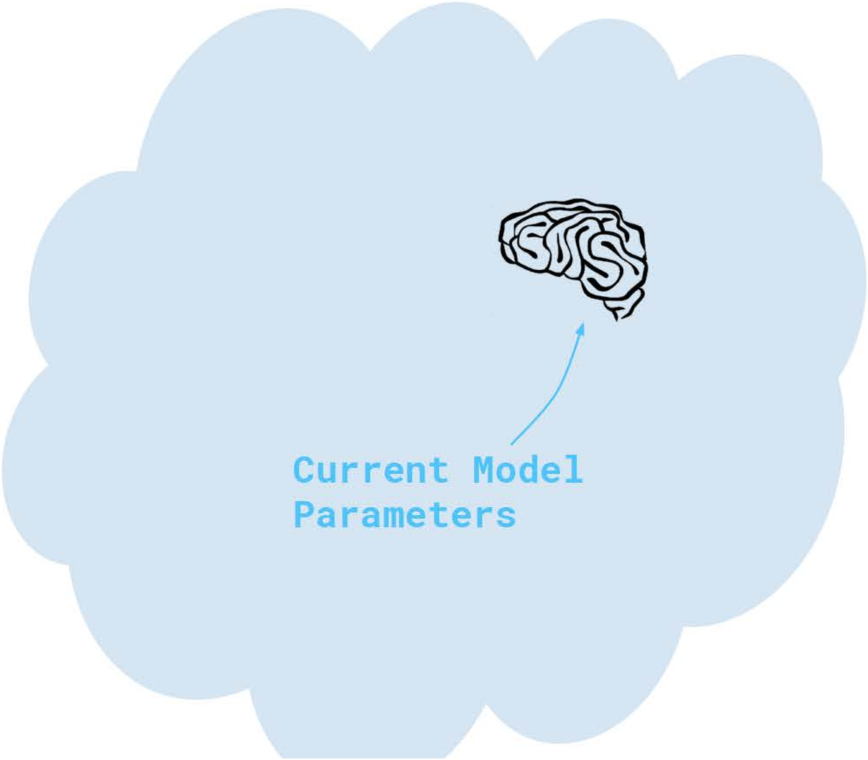
image
recognition

speech
recognition

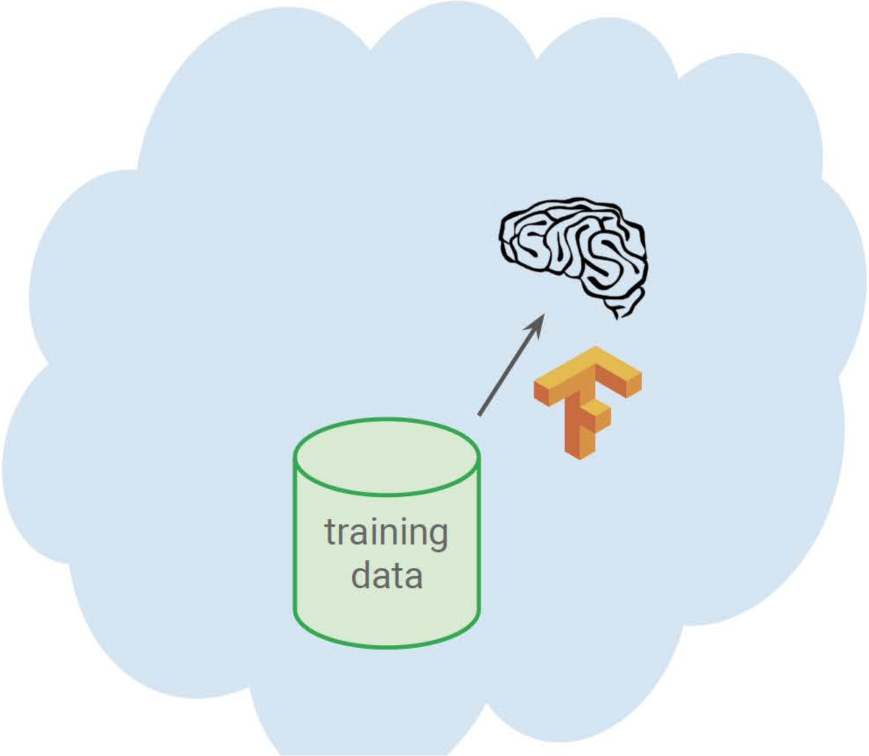
natural language
processing

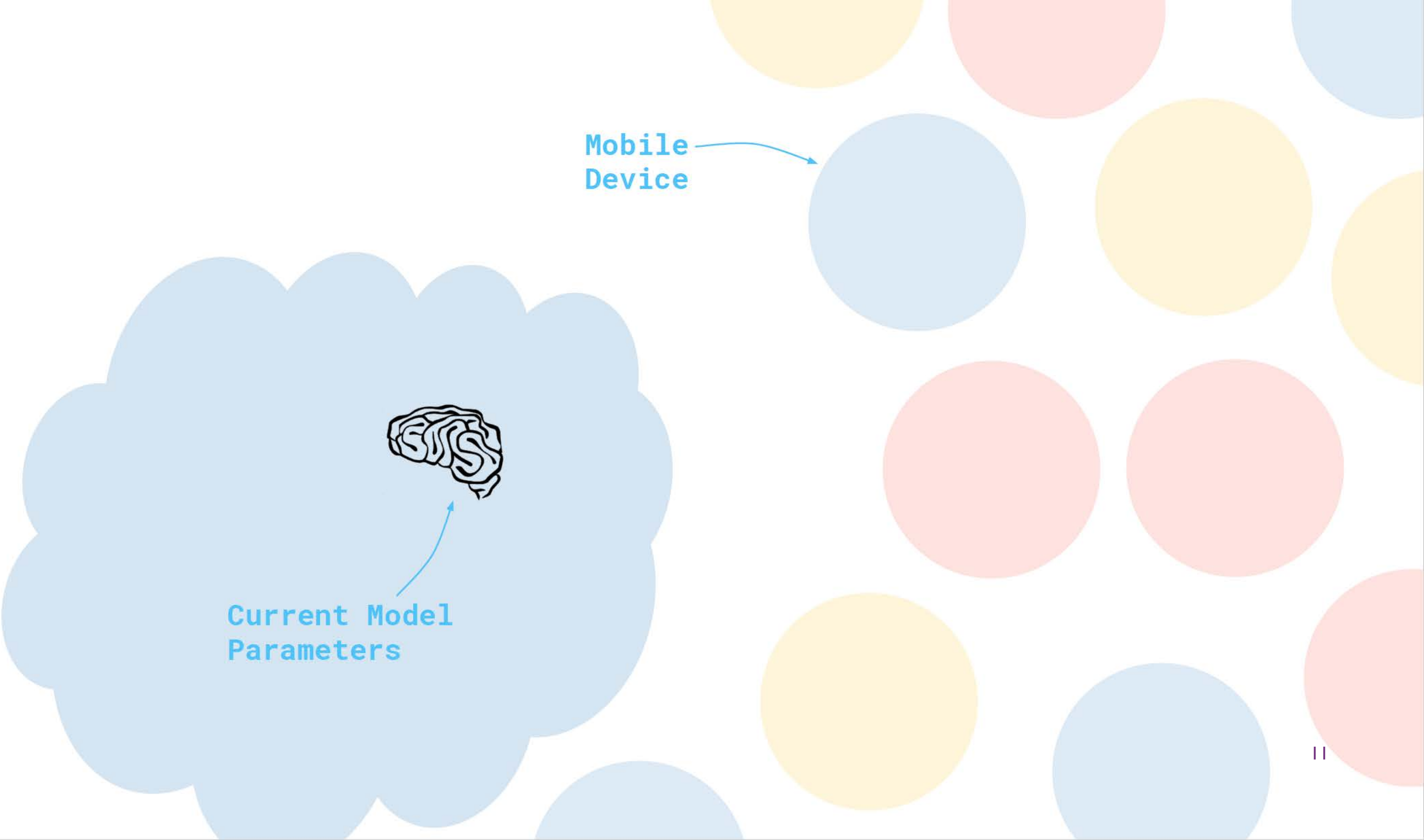
Cloud-centric machine learning

The model lives in the cloud

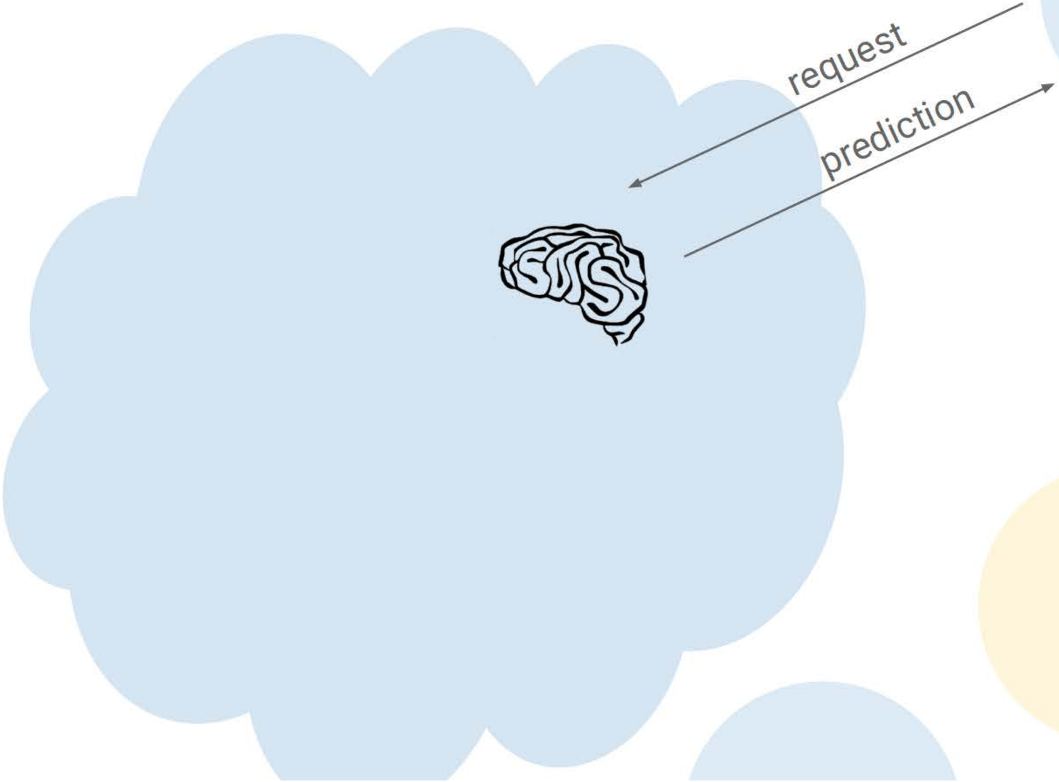


We train models in the cloud

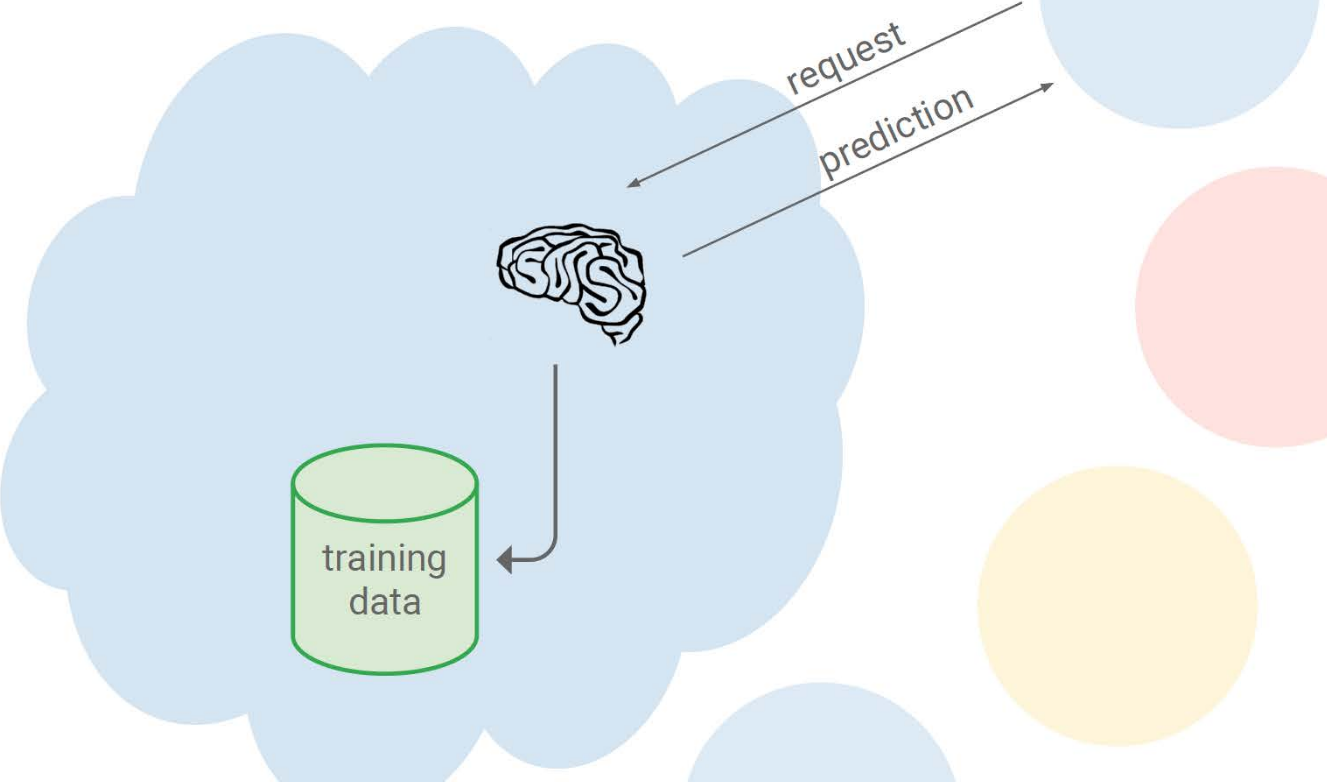



Mobile
Device
Current Model
Parameters

Make predictions in the cloud



Gather training data in the cloud



And make the models better

Why edge machine learning?

Learning on the edge

- The emerging high-stake AI applications: low-latency, privacy,…

phones

drones

robots

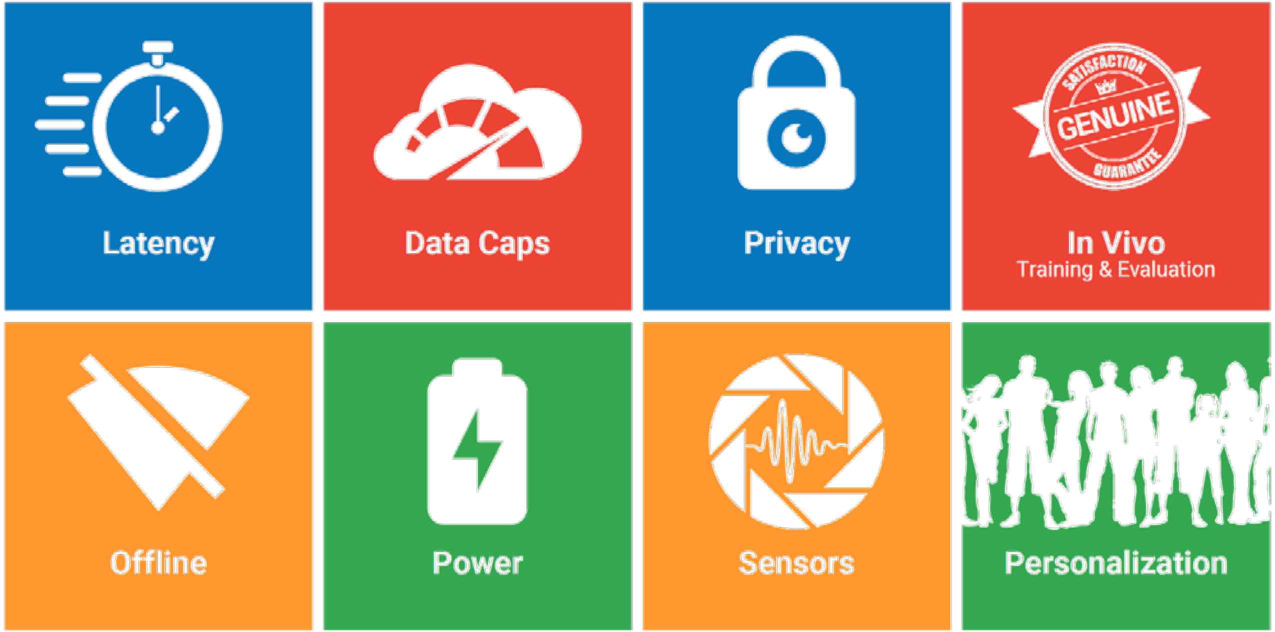
glasses

self driving cars

where to compute?

Mobile edge AI

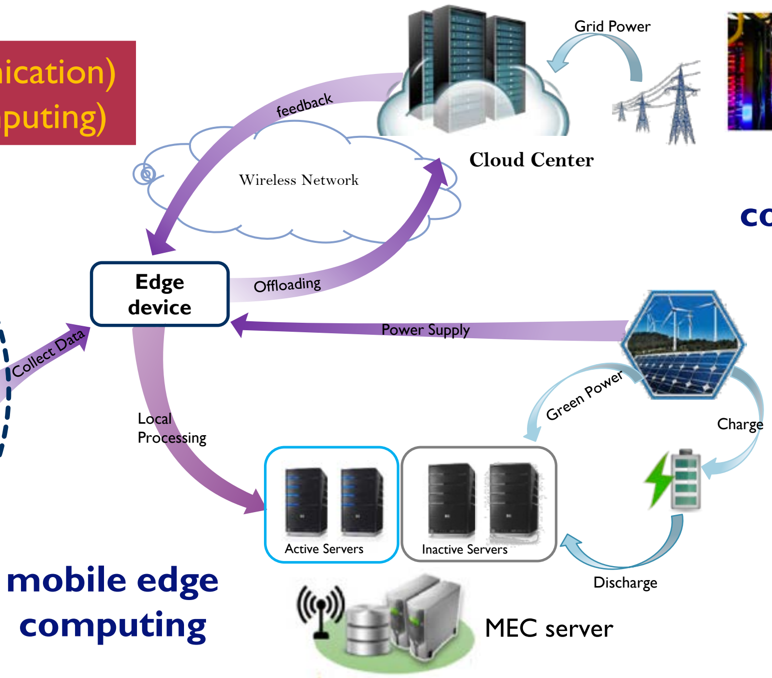
- Processing at "edge" instead of "cloud"



Edge computing ecosystem

- **“Device-edge-cloud”** computing system for mobile AI applications

Shannon (communication)
meets **Turing (computing)**



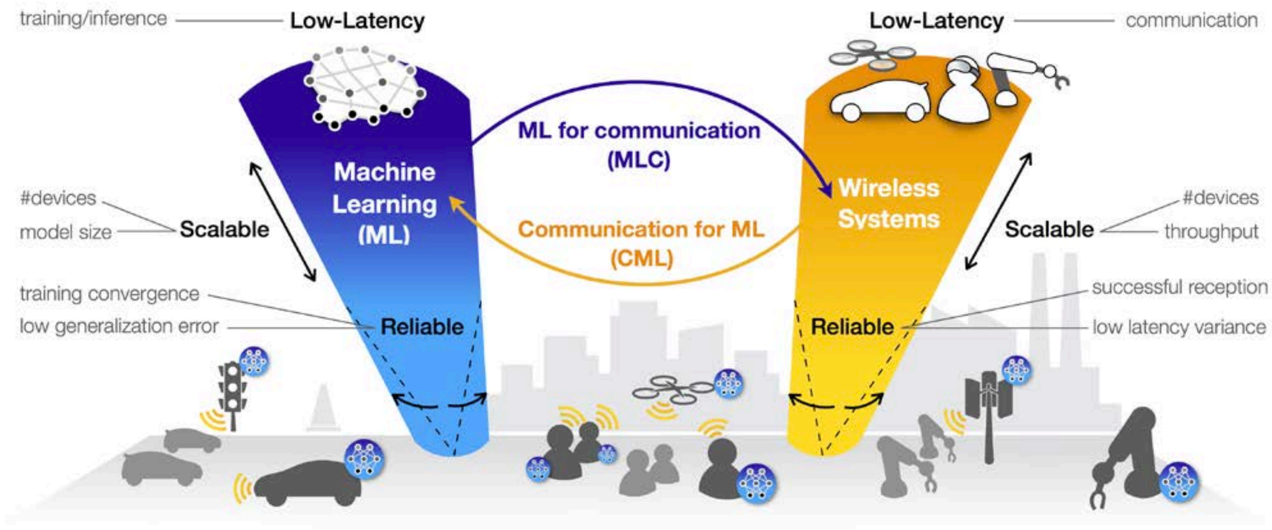
cloud computing

on-device computing

mobile edge computing

Edge machine learning

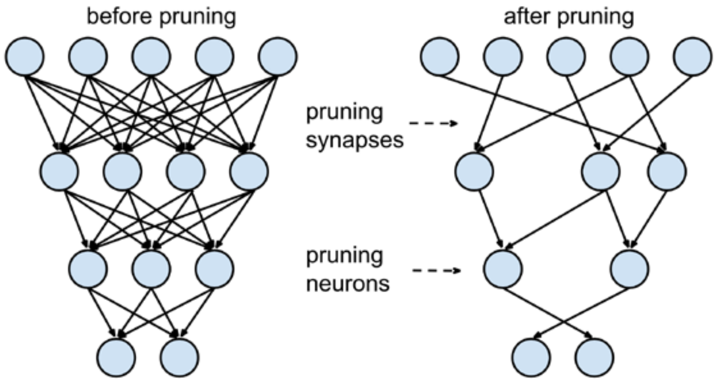
- **Edge ML:** both ML inference and training processes are pushed down into the network edge (bottom)



On-device inference

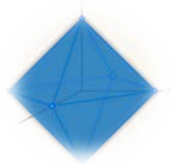
Deep model compression

- Layer-wise deep neural network pruning via sparse optimization



sparse optimization

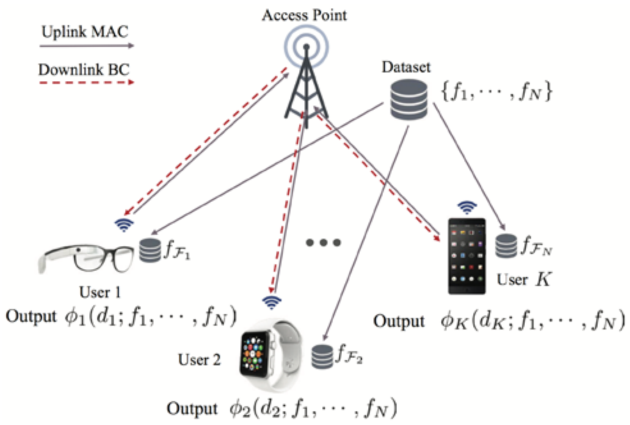
$$\begin{aligned} \underset{\boldsymbol{W} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}}{\text{minimize}} \quad & \|\boldsymbol{W}\|_1 \\ \text{subject to} \quad & \|\max(\boldsymbol{W}^T \boldsymbol{X}_{\ell-1}, 0) - \boldsymbol{X}_\ell\|_F \leq \epsilon \end{aligned}$$



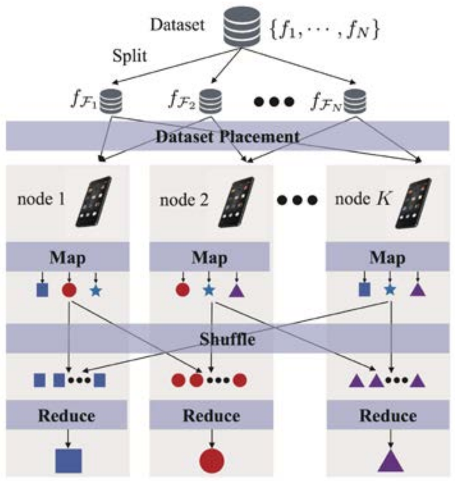
[Ref] T. Jiang, X. Yang, Y. Shi, and H. Wang, "Layer-wise deep neural network pruning via iteratively reweighted optimization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, UK, May 2019.

Edge distributed inference

- Wireless MapReduce for on-device distributed inference process



wireless distributed computing system

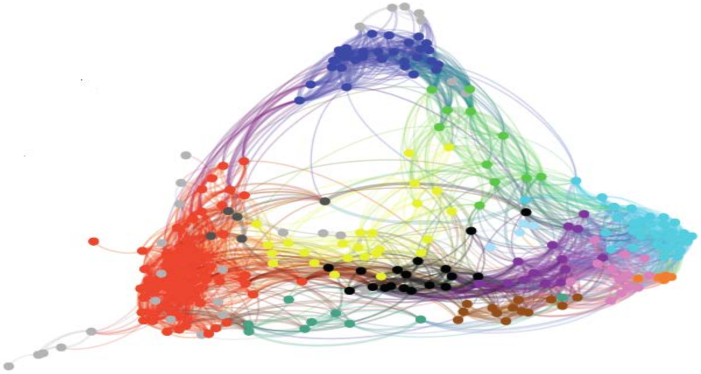


distributed computing model

[Ref] K. Yang, Y. Shi, and Z. Ding, "Data shuffling in wireless distributed computing via low-rank optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3087-3099, Jun., 2019.

<u>This talk: On-device training</u>

Vignettes A: Federated machine learning



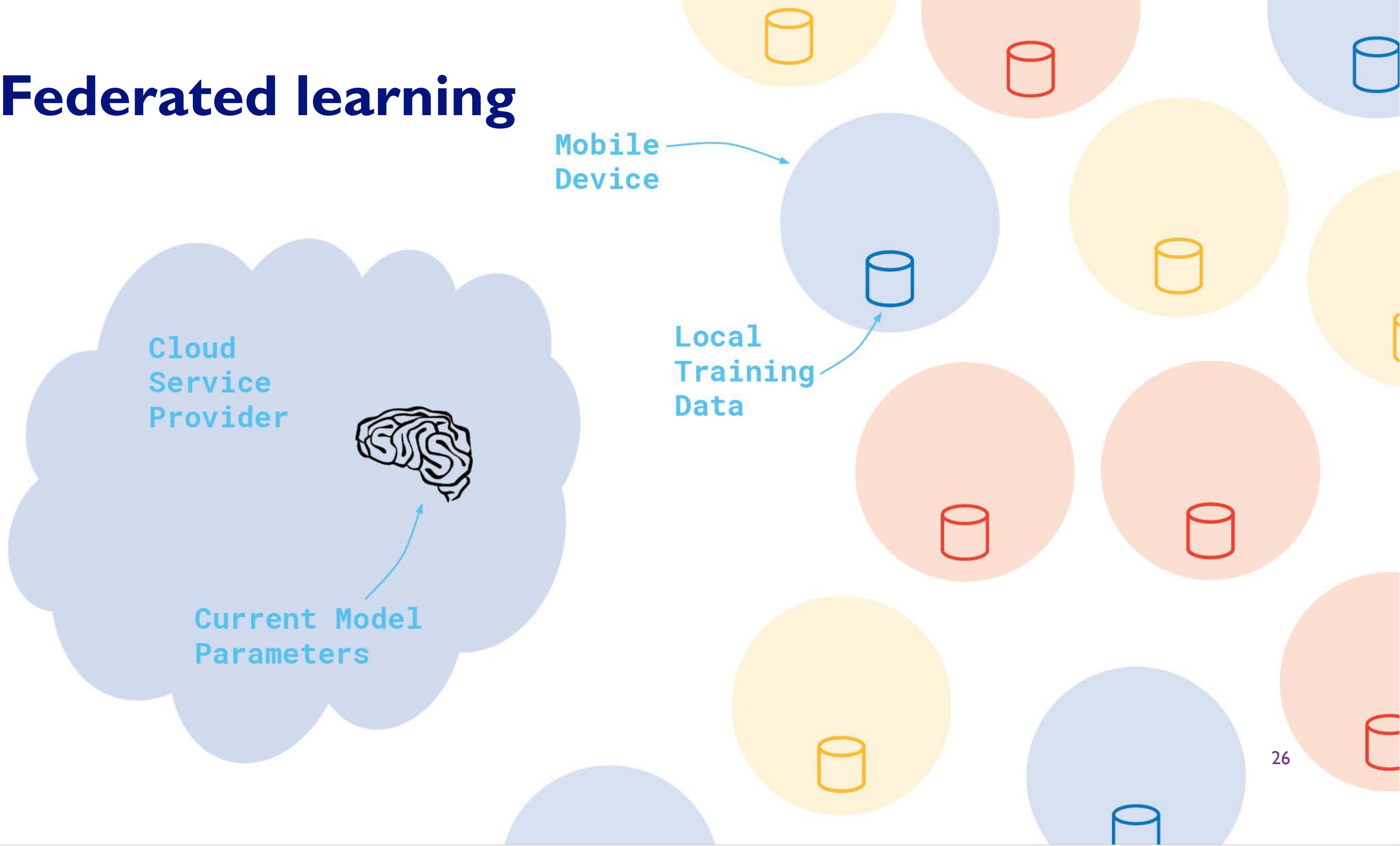
Federated computation and learning

- **Goal:** imbue mobile devices with state of the art machine learning systems ***without centralizing data*** and with ***privacy*** by default

- **Federated computation:** a server coordinates a fleet of participating devices to compute aggregations of devices' private data

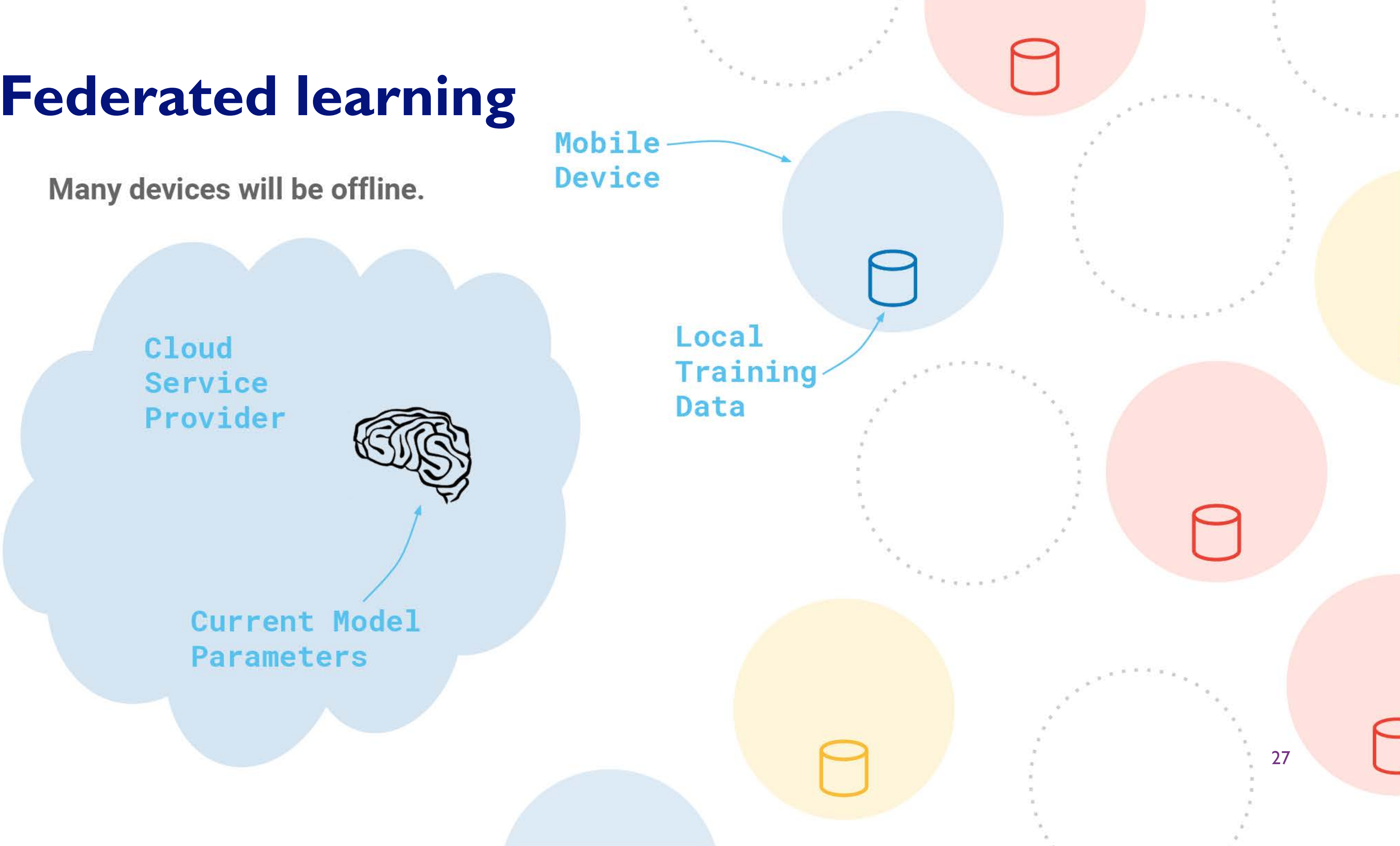
- **Federated learning:** a shared global model is trained via federated computation

Federated learning

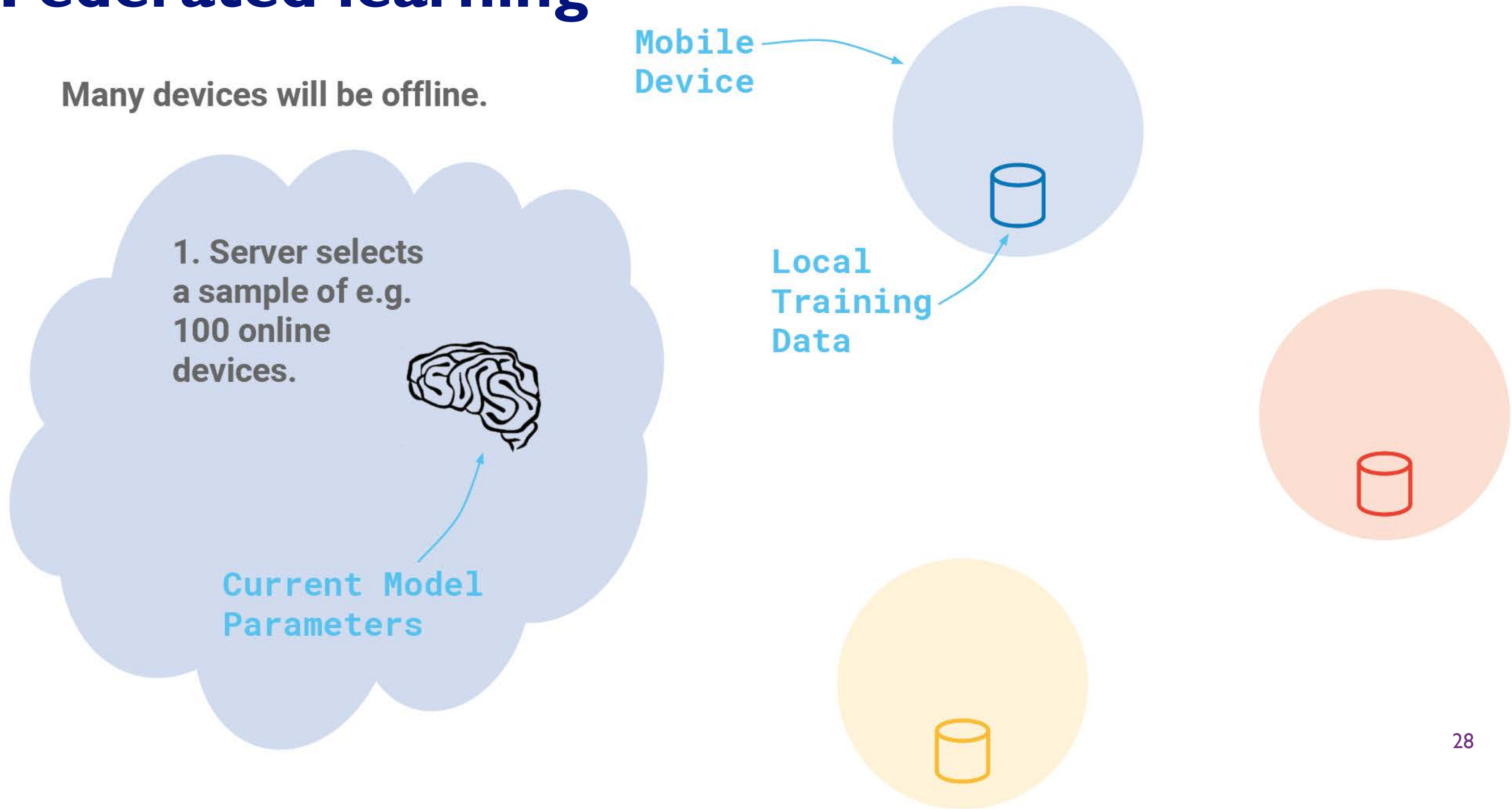


Federated learning

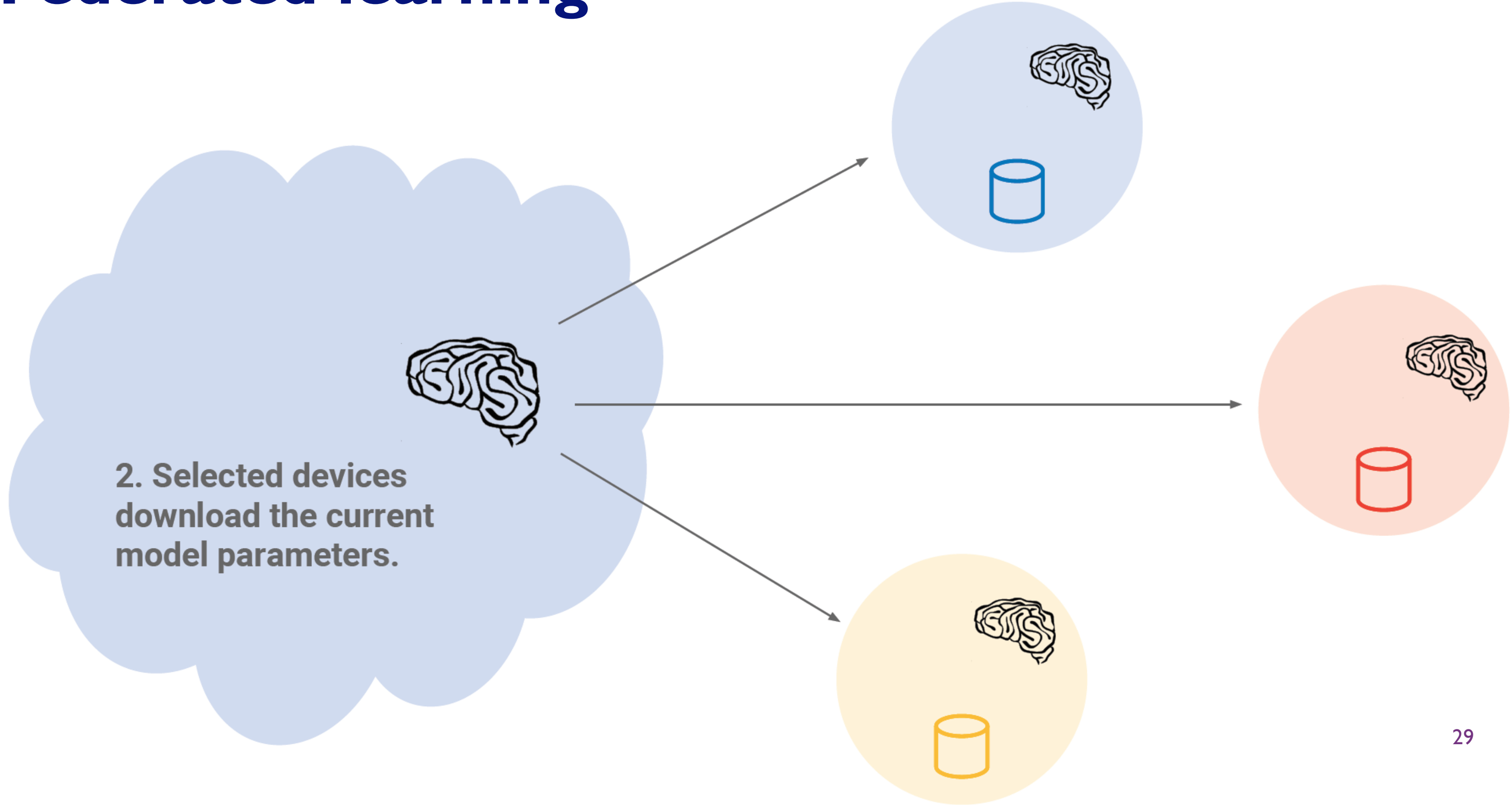
Many devices will be offline.



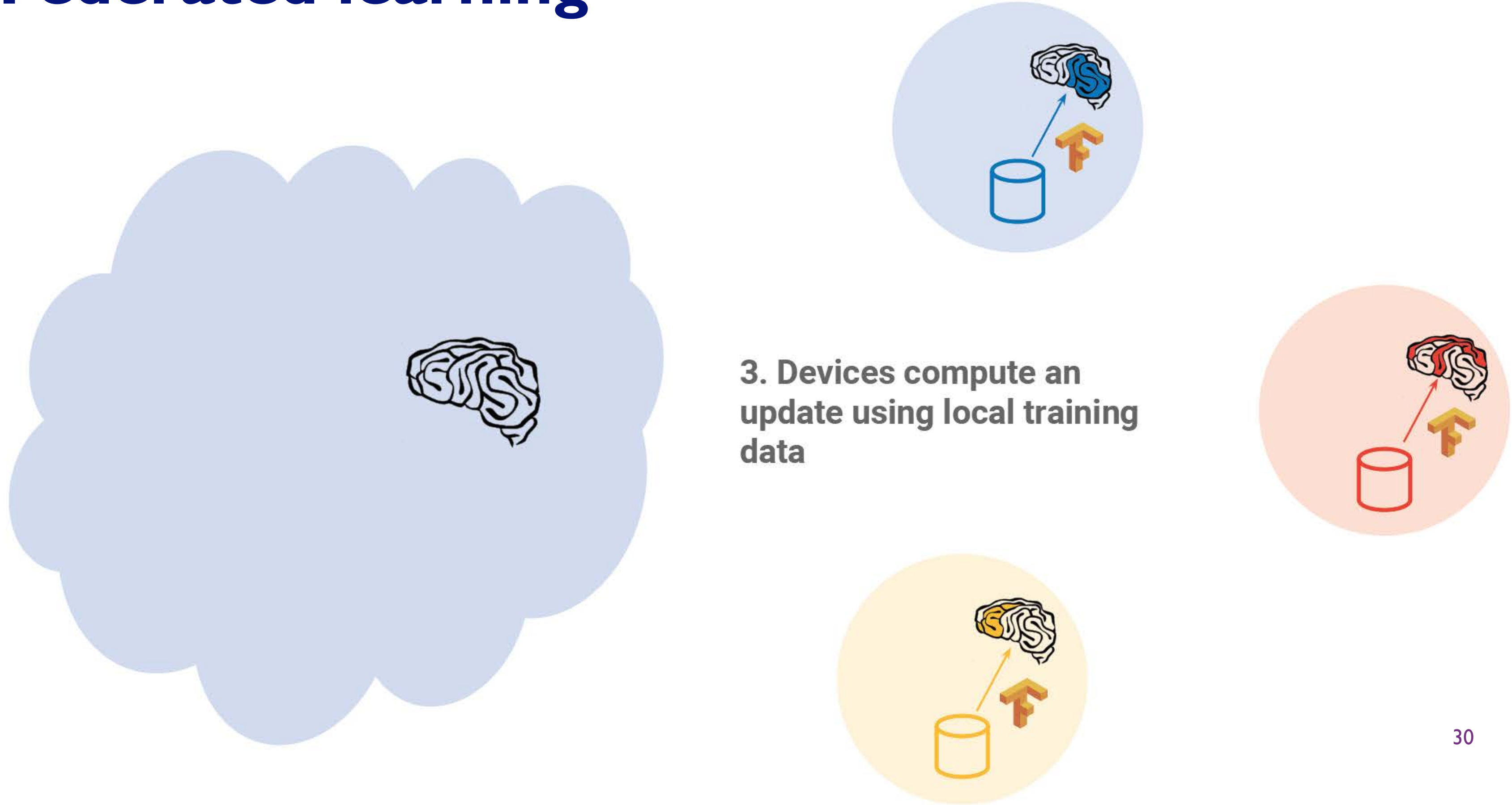
Federated learning



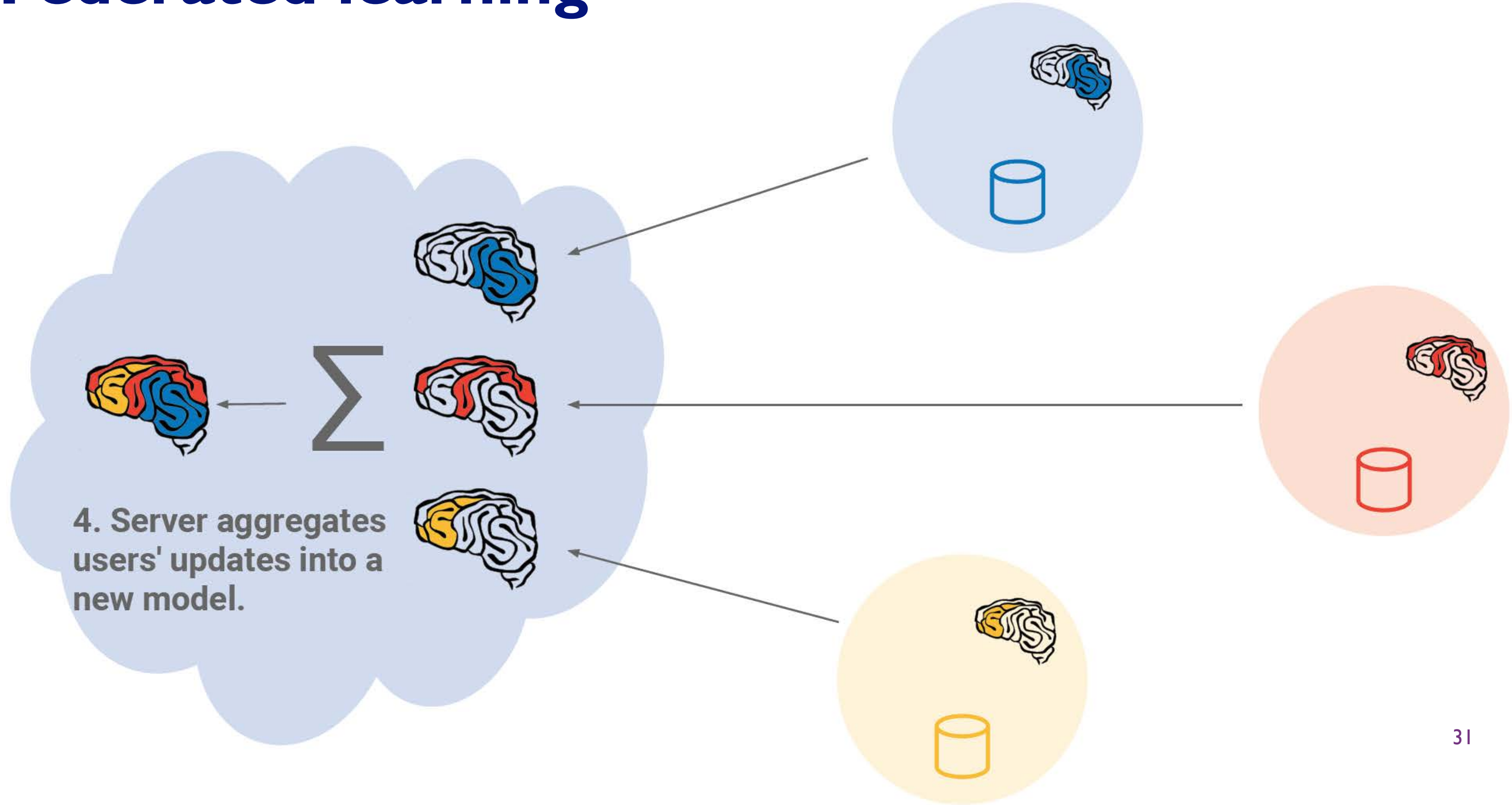
Federated learning



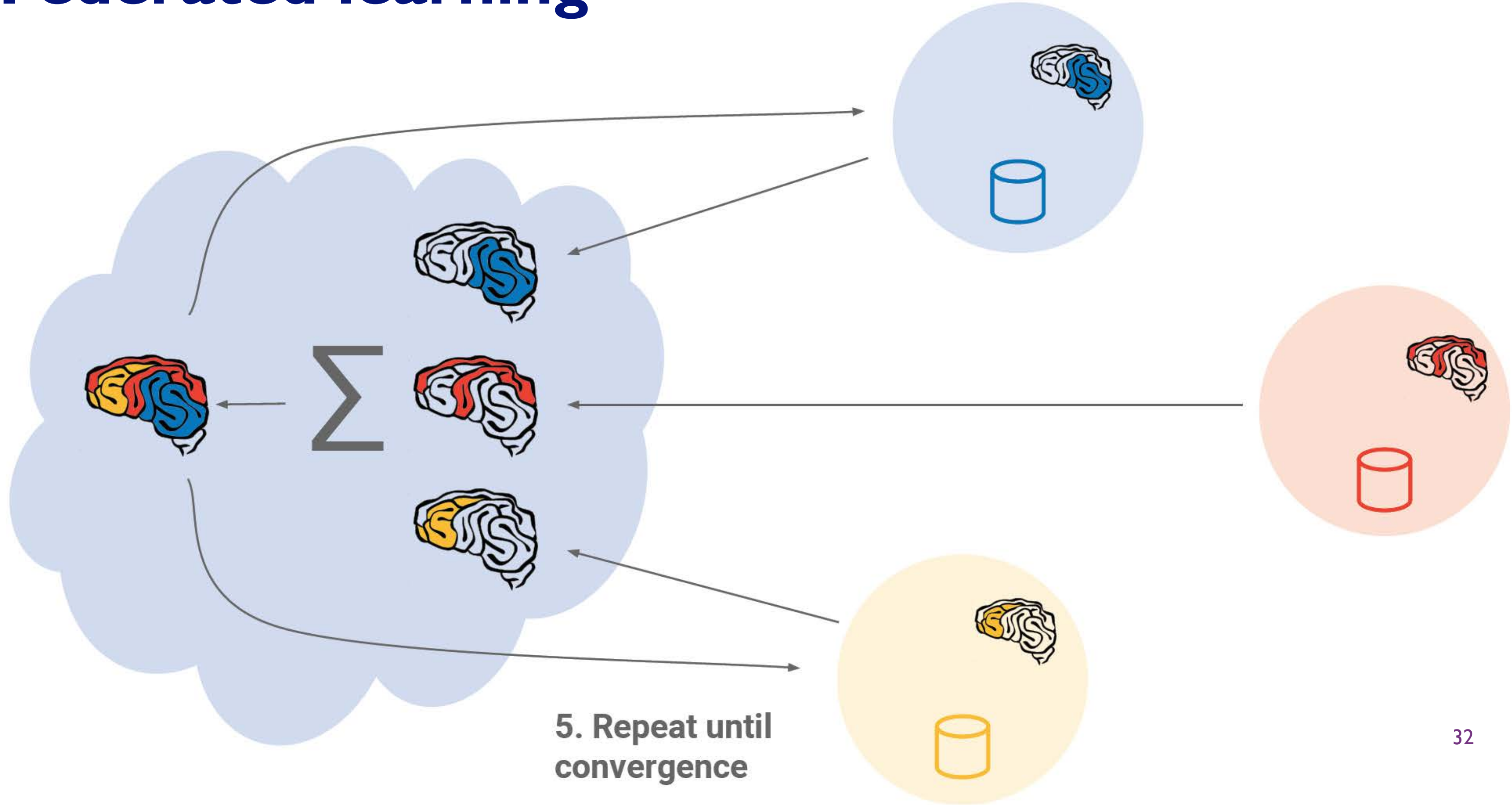
Federated learning



Federated learning



Federated learning

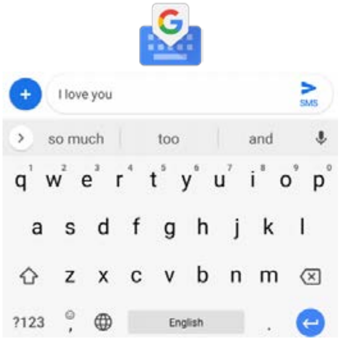


Federated learning: applications

- **Applications:** where the data is generated at the mobile devices and is undesirable/infeasible to be transmitted to centralized servers

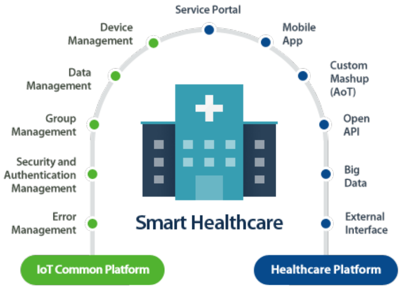
WeBank
微众银行

financial services



keyboard prediction

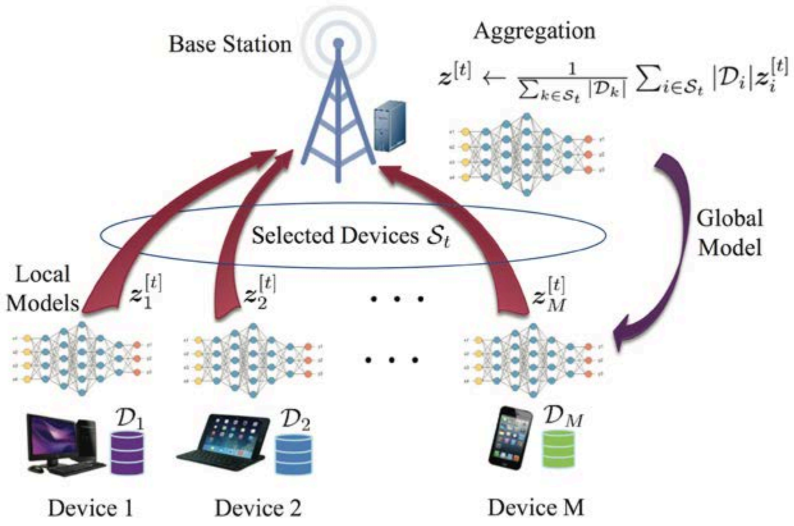
smart retail



smart healthcare

Federated learning over wireless networks

- **Goal:** train a shared global model via ***wireless*** federated computation

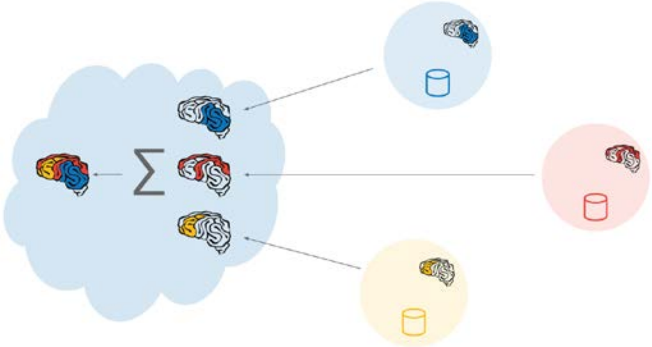


on-device distributed federated learning system

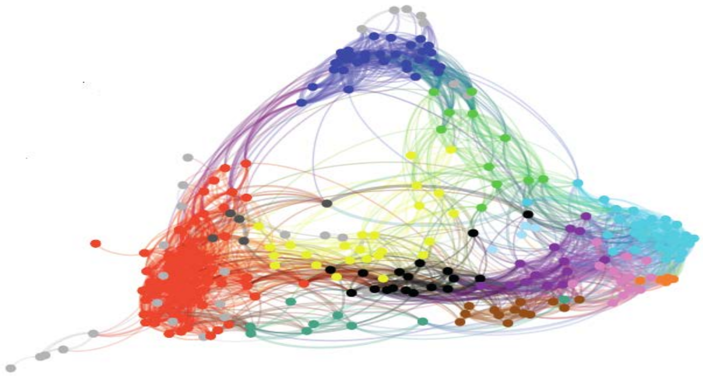
System challenges
- Massively distributed
- Node heterogeneity

Statistical challenges
- Unbalanced
- Non-IID
- Underlying structure

<u>How to efficiently aggregate models over wireless networks?</u>



Vignettes B: Over-the-air computation

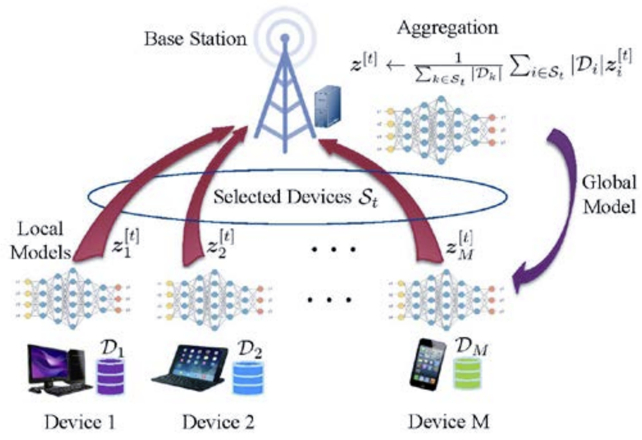


Model aggregation via over-the-air computation

- Aggregating local updates from mobile devices

$$\boxed{z \leftarrow \frac{1}{\sum_{k\in\mathcal{S}}|\mathcal{D}_k|}\sum_{k\in\mathcal{S}}|\mathcal{D}_k|z_k}$$

  - weighted sum of messages
  - M mobile devices and one N-antenna base station
  - $\mathcal{S}\subseteq\{1,\cdots,M\}$ is the set of selected devices
  - $|\mathcal{D}_k|$ is the data size at device k



Over-the-air computation: explore signal superposition of a wireless multiple-access channel for model aggregation

Over-the-air computation

- The estimated value before post-processing at the BS

$$\hat{g} = \frac{1}{\sqrt{\eta}}\boldsymbol{m}^{\mathsf{H}}\boldsymbol{y} = \frac{1}{\sqrt{\eta}}\boldsymbol{m}^{\mathsf{H}}\sum_{i\in\mathcal{S}}\boldsymbol{h}_i b_i z_i + \frac{\boldsymbol{m}^{\mathsf{H}}\boldsymbol{n}}{\sqrt{\eta}}$$

  - b_i is the transmitter scalar, $\boldsymbol{m}$ is the received beamforming vector, η is a normalizing factor
  - target function to be estimated: $g = \sum_{i\in\mathcal{S}}|\mathcal{D}_i|z_i$
  - recovered aggregation vector entry via post-processing: $\hat{z} = \frac{1}{\sum_{i\in\mathcal{S}}|\mathcal{D}_i|}\hat{g}$

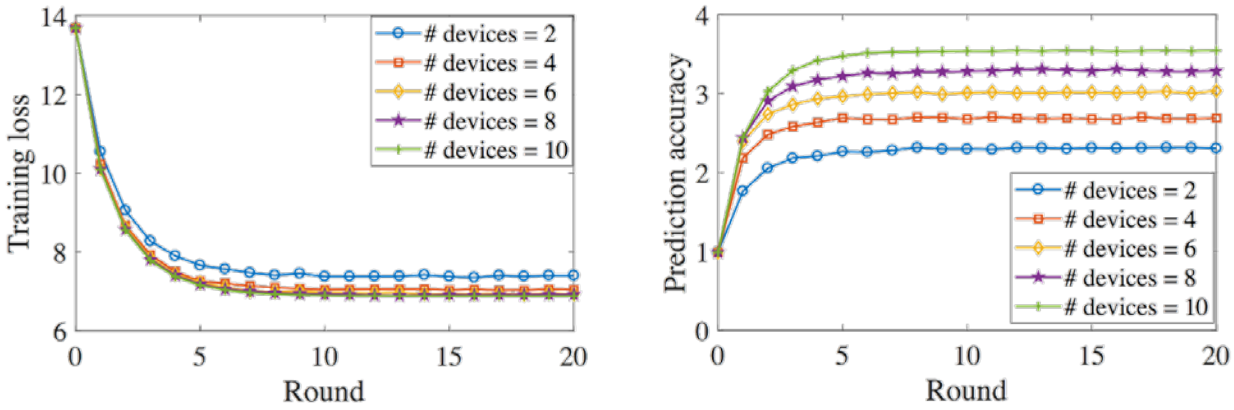
- **Model aggregation error:**

$$\mathsf{MSE}(\hat{g}, g; \mathcal{S}, \boldsymbol{m}) = \frac{\|\boldsymbol{m}\|^2\sigma^2}{\eta} = \frac{\sigma^2}{P_0}\max_{i\in\mathcal{S}}|\mathcal{D}_i|^2\frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2}$$

  - Optimal transmitter scalar: $b_i = \sqrt{\eta}|\mathcal{D}_i|\frac{(\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i)^{\mathsf{H}}}{\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2}$

Problem formulation

- **Key observations:**
  - ➢ More selected devices yield fast convergence rate of the training process
  - ➢ Aggregation error leads to the deterioration of model prediction accuracy



(a) Training loss (b) Relative prediction accuracy

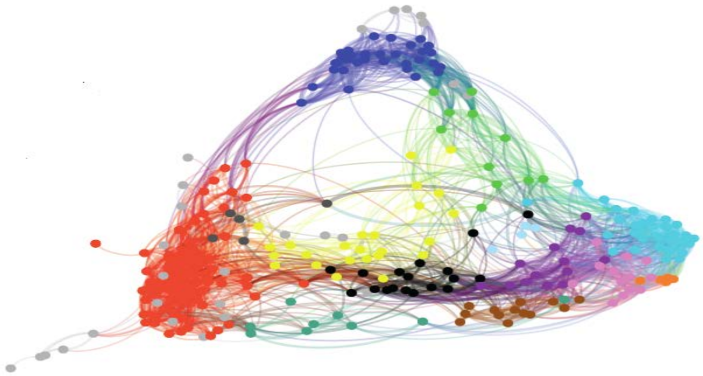
Problem formulation

- **Goal:** maximize the number of selected devices under target MSE constraint

$$\underset{\mathcal{S},\boldsymbol{m}\in\mathbb{C}^N}{\text{maximize}} \quad |\mathcal{S}| \qquad \text{subject to} \quad \left(\max_{i\in\mathcal{S}} |\mathcal{D}_i|^2 \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2}\right) \leq \gamma$$

  - Joint device selection and received beamforming vector design
  - Improve convergence rate in the ***training process***, guarantee prediction accuracy in the ***inference process***
  - **Mixed combinatorial optimization problem**

Vignettes C: Sparse and low-rank optimization



Sparse and low-rank optimization

- Sparse and low-rank optimization for on-device federated learning

$$
\begin{aligned}
\underset{\mathcal{S},\boldsymbol{m}\in\mathbb{C}^N}{\text{maximize}} \quad & |\mathcal{S}| \\
\text{subject to} \quad & \left(\max_{i\in\mathcal{S}} |\mathcal{D}_i|^2 \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2}\right) \le \gamma
\end{aligned}
$$

multicasting duality

$$
\begin{aligned}
\underset{\mathcal{S},\boldsymbol{m}\in\mathbb{C}^N}{\text{maximize}} \quad & |\mathcal{S}| \\
\text{subject to} \quad & \|\boldsymbol{m}\|^2 - \gamma_i\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2 \le 0, i\in\mathcal{S} \\
& \|\boldsymbol{m}\|^2 \ge 1
\end{aligned}
$$

sum of feasibilities

$$
\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}_+^M,\boldsymbol{m}\in\mathbb{C}^N}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\
\text{subject to} \quad & \|\boldsymbol{m}\|^2 - \gamma_i\|\boldsymbol{m}^{\mathsf{H}}\boldsymbol{h}_i\|^2 \le x_i, \forall i \\
& \|\boldsymbol{m}\|^2 \ge 1
\end{aligned}
$$

$\boldsymbol{M} = \boldsymbol{m}\boldsymbol{m}^{\mathsf{H}}$

matrix lifting

$$
\boxed{
\begin{aligned}
\mathscr{P}: \underset{\boldsymbol{x}\in\mathbb{R}_+^M,\boldsymbol{M}\in\mathbb{C}^{N\times N}}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\
\text{subject to} \quad & \mathrm{Tr}(\boldsymbol{M}) - \gamma_i\boldsymbol{h}_i^{\mathsf{H}}\boldsymbol{M}\boldsymbol{h}_i \le x_i, \\
& \boldsymbol{M} \succeq \boldsymbol{0}, \mathrm{Tr}(\boldsymbol{M}) \ge 1 \\
& \mathrm{rank}(\boldsymbol{M}) = 1
\end{aligned}
}
$$

Problem analysis

- **Goal:** induce sparsity while satisfying fixed-rank constraint

$$
\begin{aligned}
\mathscr{P}: \underset{\boldsymbol{x}\in\mathbb{R}_+^M, \boldsymbol{M}\in\mathbb{C}^{N\times N}}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\
\text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \le x_i, \forall i \\
& \boldsymbol{M} \succeq \mathbf{0}, \operatorname{Tr}(\boldsymbol{M}) \ge 1 \\
& \operatorname{rank}(\boldsymbol{M}) = 1
\end{aligned}
$$

- Limitations of existing methods
  - Sparse optimization: iterative reweighted algorithms are parameters sensitive
  - Low-rank optimization: semidefinite relaxation (SDR) approach (i.e., drop rank-one constraint) has the poor capability of returning rank-one solution

Difference-of-convex functions representation

- Ky Fan k-norm [Fan, PNAS'1951]: the sum of largest-k absolute values

$$\|\boldsymbol{x}\|_k = \sum_{i=1}^{k} |x_{\pi(i)}|$$

convex function

  - π is a permutation of $\{1, \cdots, M\}$, where $|x_{\pi(1)}| \geq \cdots \geq |x_{\pi(M)}|$

*MAXIMUM PROPERTIES AND INEQUALITIES FOR THE EIGENVALUES OF COMPLETELY CONTINUOUS OPERATORS**

By Ky Fan

Department of Mathematics, University of Notre Dame

Communicated by John von Neumann, September 8, 1951

PNAS'1951

Difference-of-convex functions representation

- DC representation for sparsity function

$$\|\boldsymbol{x}\|_0 = \min\{k : \|\boldsymbol{x}\|_1 - |||\boldsymbol{x}|||_k = 0, 0 \le k \le M\}$$

- DC representation for rank-one positive semidefinite matrix

$$\operatorname{rank}(\boldsymbol{M}) = 1 \Leftrightarrow \operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 = 0$$

  - where $\operatorname{Tr}(\boldsymbol{M}) = \sum_{i=1}^{N} \sigma_i(\boldsymbol{M})$ and $\|\boldsymbol{M}\|_2 = \sigma_1(\boldsymbol{M})$

algorithmic advantages?

[Ref] J.-y. Gotoh, A. Takeda, and K. Tono, "DC formulations and algorithms for sparse optimization problems," *Math. Program.*, vol. 169, pp. 141– 176, May 2018.

A DC representation framework

- A two-step framework for device selection

Induce sparsity structure of vector $\boldsymbol{x}$ via solving problem $\mathscr{P}_{\mathrm{S1}}$ $\xrightarrow{\boldsymbol{x}^\star}$ Check the feasibility of selected devices via solving problem $\mathscr{P}_{\mathrm{S2}}$

- **Step 1:** obtain the sparse solution such that the objective value achieves zero through increasing k from 0 to M

$$\begin{aligned}\mathscr{P}_{\mathrm{S1}}: \underset{\boldsymbol{x},\boldsymbol{M}}{\text{minimize}} \quad & \underline{\|\boldsymbol{x}\|_1 - \|\boldsymbol{x}\|_k} + \underline{\mathrm{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2} \qquad \text{zero?} \\ \text{subject to} \quad & \mathrm{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \le x_i, \forall i = 1, \cdots, M \\ & \boldsymbol{M} \succeq \boldsymbol{0}, \quad \mathrm{Tr}(\boldsymbol{M}) \ge 1, \boldsymbol{x} \succeq \boldsymbol{0}\end{aligned}$$

A DC representation framework

- **Step II:** feasibility detection
  - Ordering $\boldsymbol{x}$ in descending order as $x_{\pi(1)} \geq \cdots \geq x_{\pi(M)}$
  - Increasing k from 1 to M, choosing $\mathcal{S}^{[k]}$ as $\{\pi(k), \pi(k+1), \cdots, \pi(M)\}$
- Feasibility detection via DC programming

$$\begin{aligned} \text{find} \quad & \boldsymbol{M} \\ \text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \forall i \in \mathcal{S}^{[k]} \\ & \boldsymbol{M} \succeq \boldsymbol{0}, \operatorname{Tr}(\boldsymbol{M}) \geq 1, \operatorname{rank}(\boldsymbol{M}) = 1 \end{aligned}$$

$$\begin{aligned} \mathscr{P}_{\mathrm{S2}} : \underset{\boldsymbol{M}}{\text{minimize}} \quad & \underline{\operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2} \quad \text{zero?} \\ \text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \forall i \in \mathcal{S}^{[k]} \\ & \boldsymbol{M} \succeq \boldsymbol{0}, \quad \operatorname{Tr}(\boldsymbol{M}) \geq 1 \end{aligned}$$

DC algorithm with convergence guarantees

- $\mathscr{P}_{\mathrm{S1}}$ and $\mathscr{P}_{\mathrm{S2}}$: minimize the difference of two strongly convex functions

$$\underset{\boldsymbol{X}\in\mathbb{C}^{m\times n}}{\text{minimize}} \quad f(\boldsymbol{X}) = g(\boldsymbol{X}) - h(\boldsymbol{X})$$

  - e.g., $g = \mathrm{Tr}(\boldsymbol{M}) + I_{\mathcal{C}_2}(\boldsymbol{M}) + \frac{\alpha}{2}\|\boldsymbol{M}\|_F^2$ and $h = \|\boldsymbol{M}\|_2 + \frac{\alpha}{2}\|\boldsymbol{M}\|_F^2$

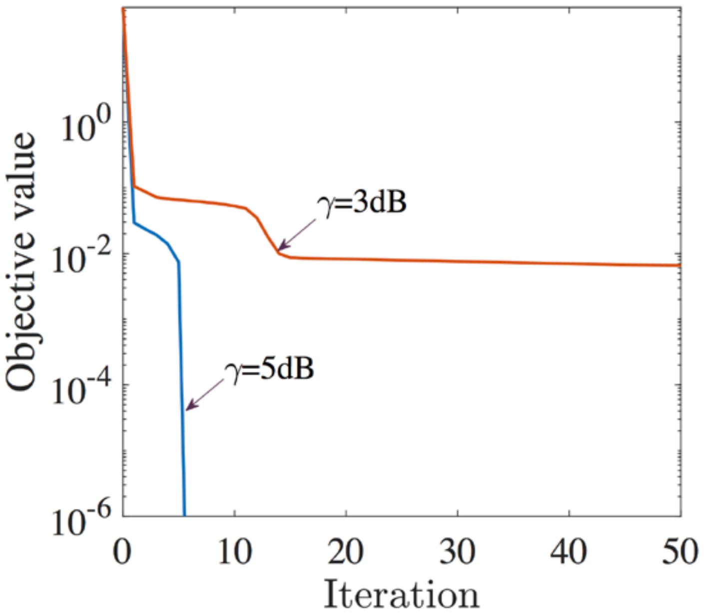
- The DC algorithm via linearizing the concave part

$$\boldsymbol{X}^{[t+1]} = \arg\inf_{\boldsymbol{X}\in\mathcal{X}} \; g(\boldsymbol{X}) - [h(\boldsymbol{X}^{[t]}) + \langle \boldsymbol{X} - \boldsymbol{X}^{[t]}, \partial_{\boldsymbol{X}^{[t]}} h\rangle]$$

  - converge to a critical point with speed $\mathcal{O}(1/t)$

Numerical results

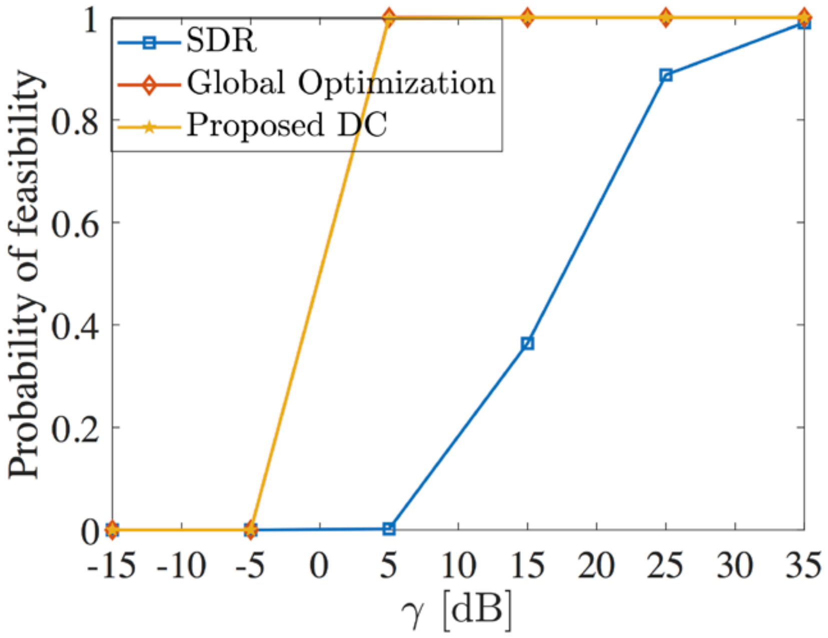
- Convergence of the proposed DC algorithm for problem $\mathscr{P}_{\mathrm{S2}}$



$$\begin{aligned} \mathscr{P}_{\mathrm{S2}}: \underset{\boldsymbol{M}}{\text{minimize}} \quad & \mathrm{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 \\ \text{subject to} \quad & \mathrm{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \\ & \boldsymbol{M} \succeq \boldsymbol{0}, \quad \mathrm{Tr}(\boldsymbol{M}) \geq 1 \end{aligned}$$

Numerical results

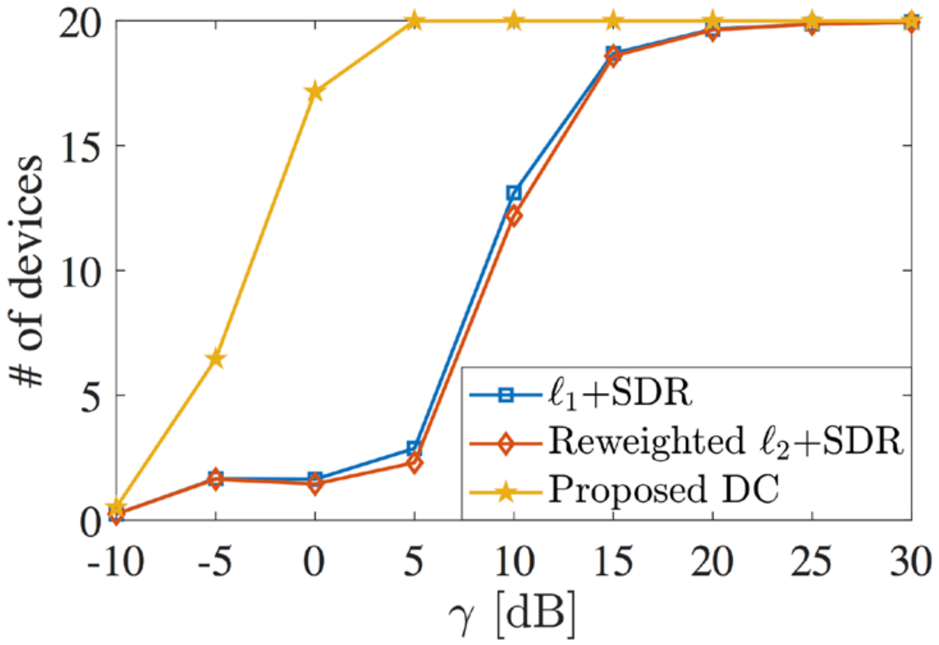
- Probability of feasibility with different algorithms



$$
\begin{aligned}
\text{find} \quad & \boldsymbol{M} \\
\text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \le 0, \forall i \in \mathcal{S}^{[k]} \\
& \boldsymbol{M} \succeq \boldsymbol{0}, \operatorname{Tr}(\boldsymbol{M}) \ge 1, \operatorname{rank}(\boldsymbol{M}) = 1
\end{aligned}
$$

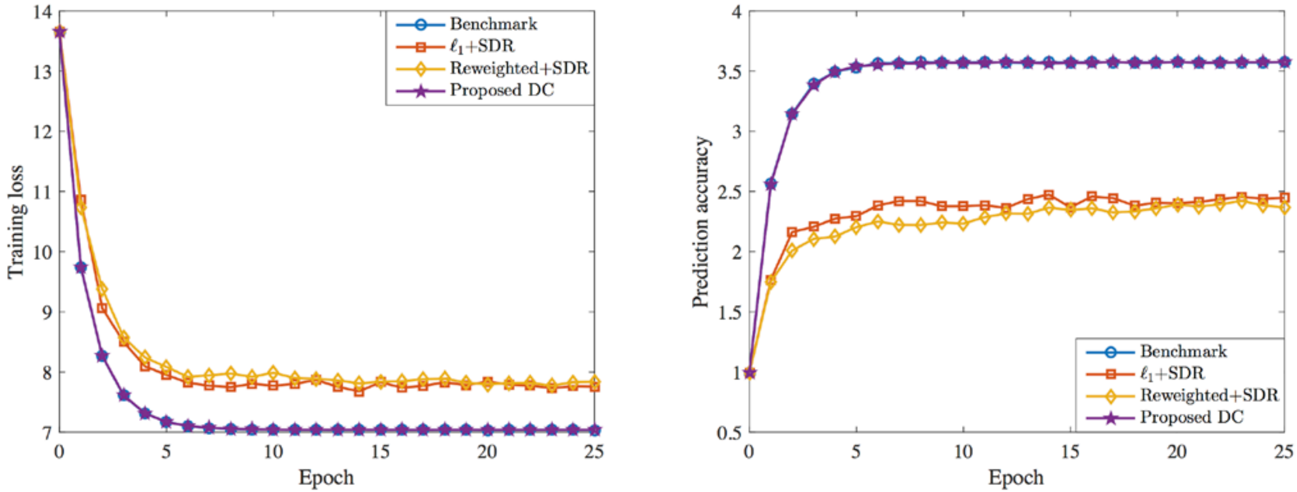
Numerical results

- Average number of selected devices with different algorithms



Numerical results

- Performance of proposed fast model aggregation in federated learning
  - Training an SVM classifier on CIFAR-10 dataset



(a) Training loss

(b) Relative prediction accuracy

Concluding remarks

- **Wireless communication meets machine learning**
  - Over-the-air computation for fast model aggregation

- **Sparse and low-rank optimization framework**
  - Joint device selection and beamforming design

- **A unified DC programming framework**
  - DC representation for sparse and low-rank functions

Future directions

- **Federated learning**
  - security, provable guarantees, …
- **Over-the-air computation**
  - channel uncertainty, synchronization,…
- **Sparse and low-rank optimization via DC programming**
  - optimality, scalability,…

To learn more…

- **Papers:**
- K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, DOI10.1109/TWC.2019.2961673, Jan. 2020.
- K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning based on over-the-air computation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019.

http://shiyuanming.github.io/home.html

Thanks