Tutorial for Federated Learning

UNIST 인공지능대학원

윤 성 환

May. 2022 @ AI Frontiers Summit

Contents

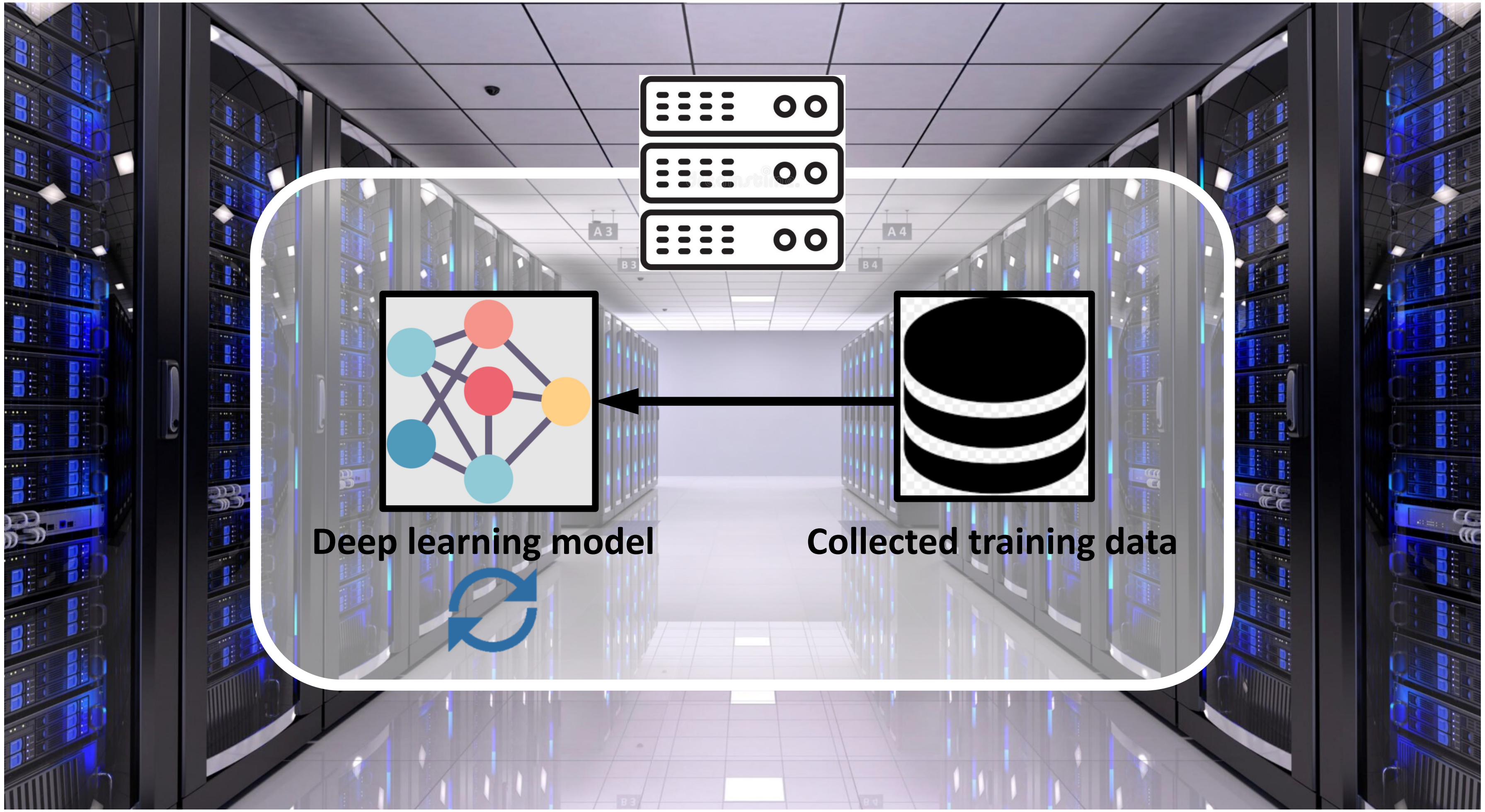
I. Federated Learning

- Motivations of Federated Learning

- Federated Averaging (FedAvg)

Motivation: Learning on Distributed Clients

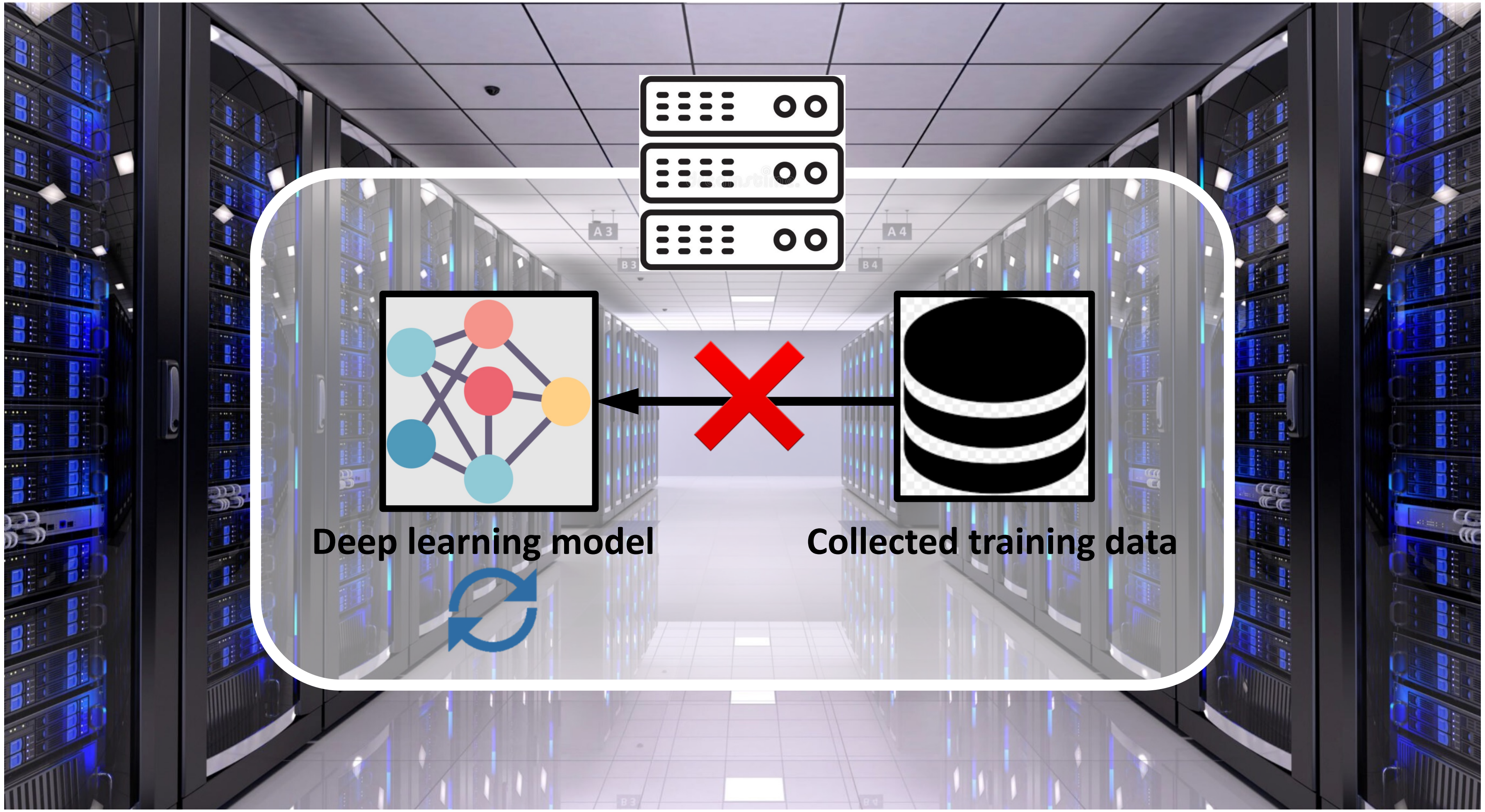
Many deep learning algorithms are trained/evaluated in a centralized learning framework.



ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

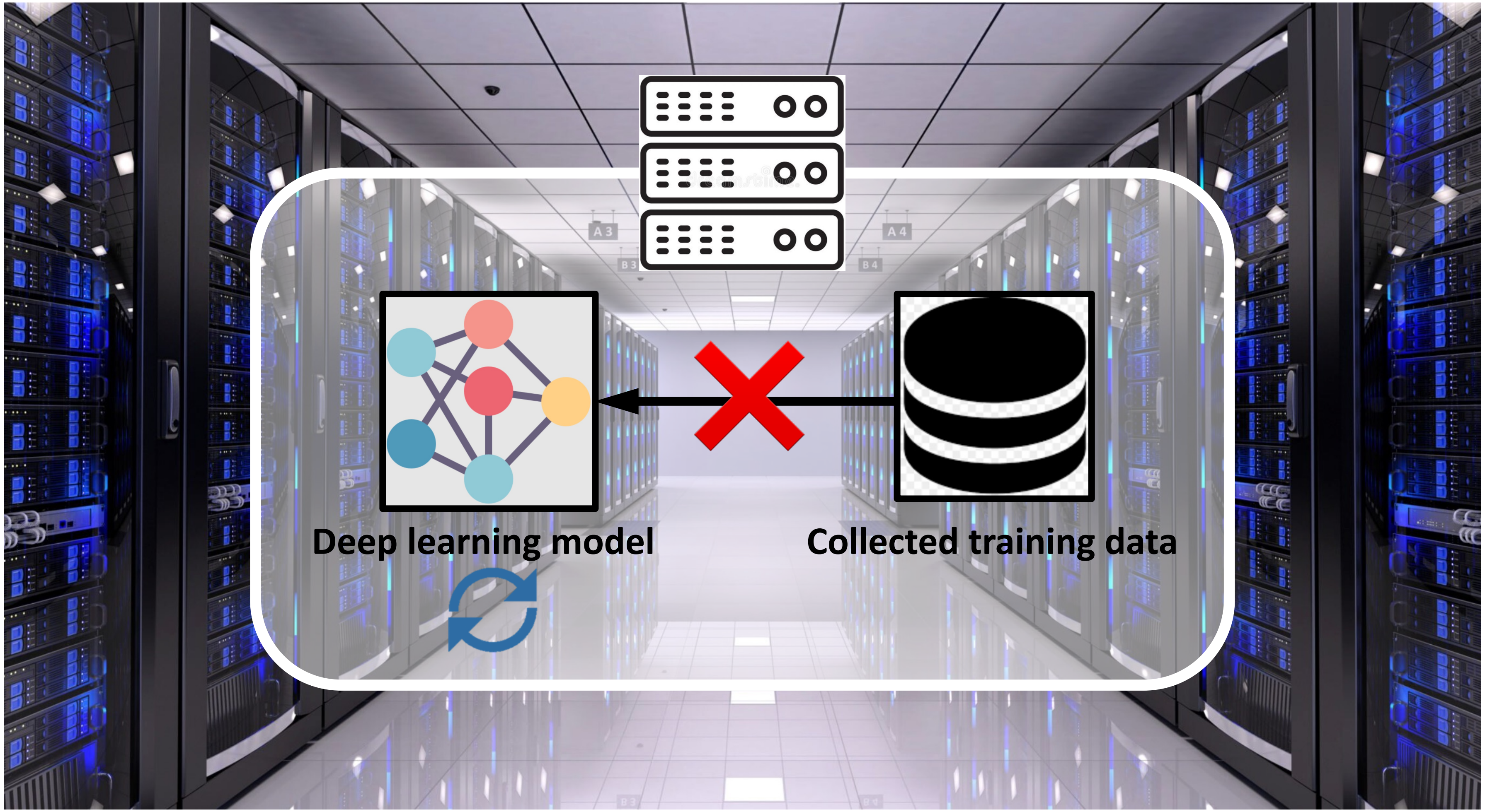
Motivation: Learning on Distributed Clients

However, real-world data cannot be collected on a central server in many cases.
Mainly due to the ***Data Privacy issue***.



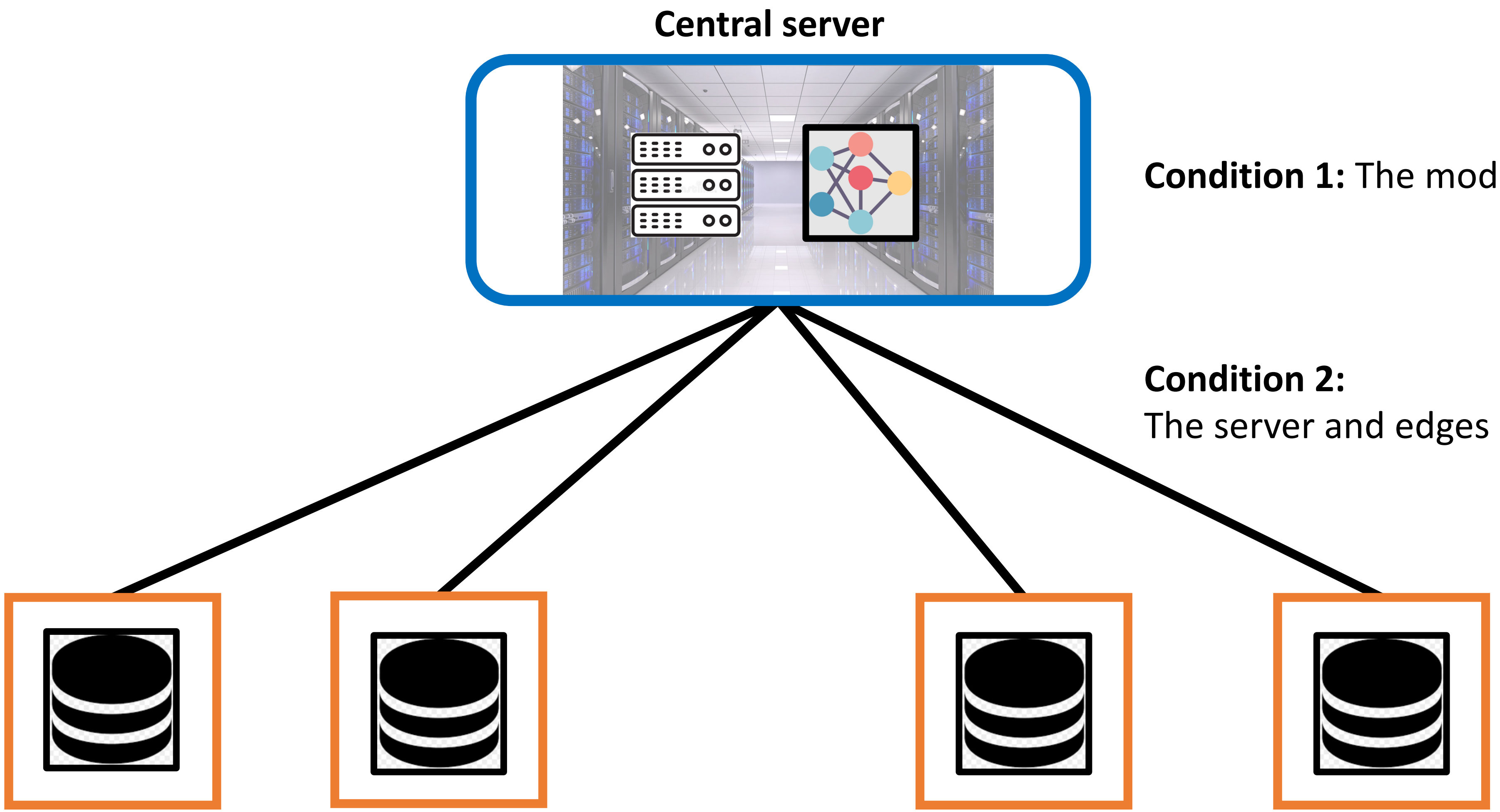
Motivation: Learning on Distributed Clients

However, real-world data cannot be collected on a central server in many cases.
Sometimes, due to *the* ***Computation Complexity*****.**



Motivation: Learning on Distributed Clients

Let us consider a distributed environment where training data is on edges (or clients).



Condition 1: The model is on the server.

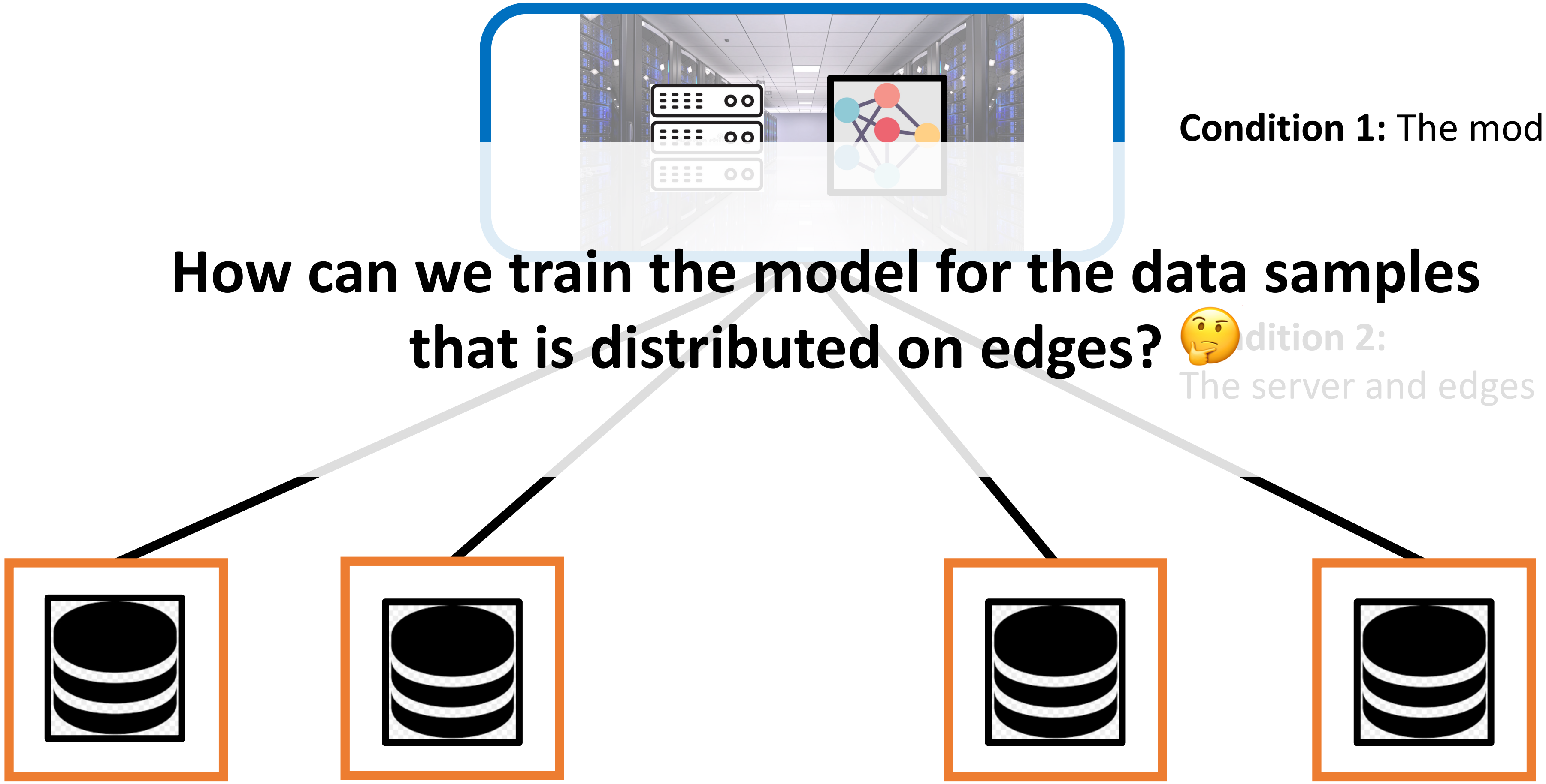
Condition 2:
The server and edges can communicate.

Condition 3:
Data stays on the edges.

Motivation: Learning on Distributed Clients

Let us consider a distributed environment where training data is on edges (or clients).

Central server



Condition 1: The model is on the server.

How can we train the model for the data samples that is distributed on edges? 🤔

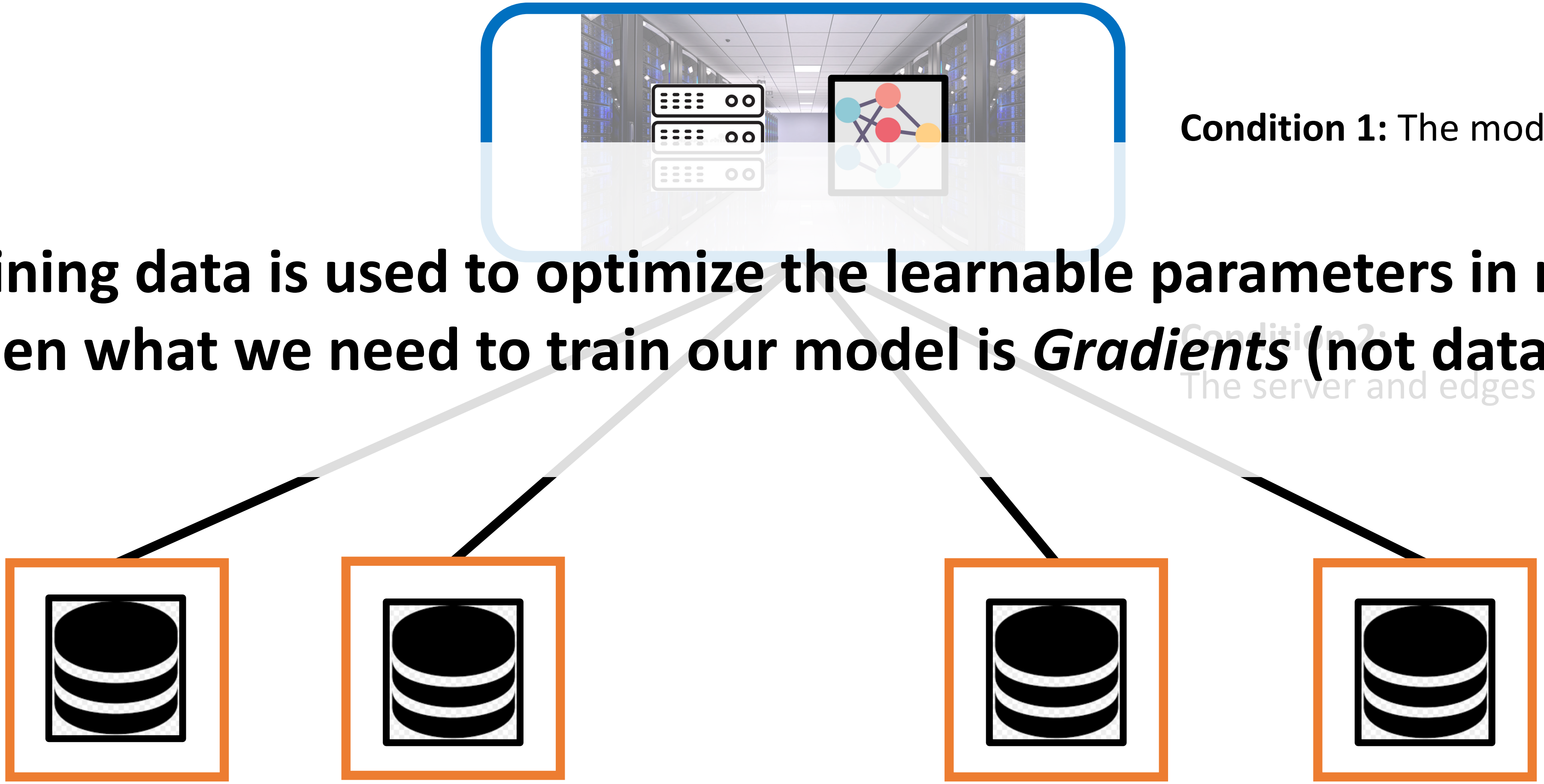
dition 2:
The server and edges can communicate.

Condition 3:
Data stays on the edges.

Motivation: Learning on Distributed Clients

Let us consider a distributed environment where training data is on edges (or clients).

Central server



Condition 1: The model is on the server.

Training data is used to optimize the learnable parameters in models.
Then what we need to train our model is *Gradients* (not data itself)

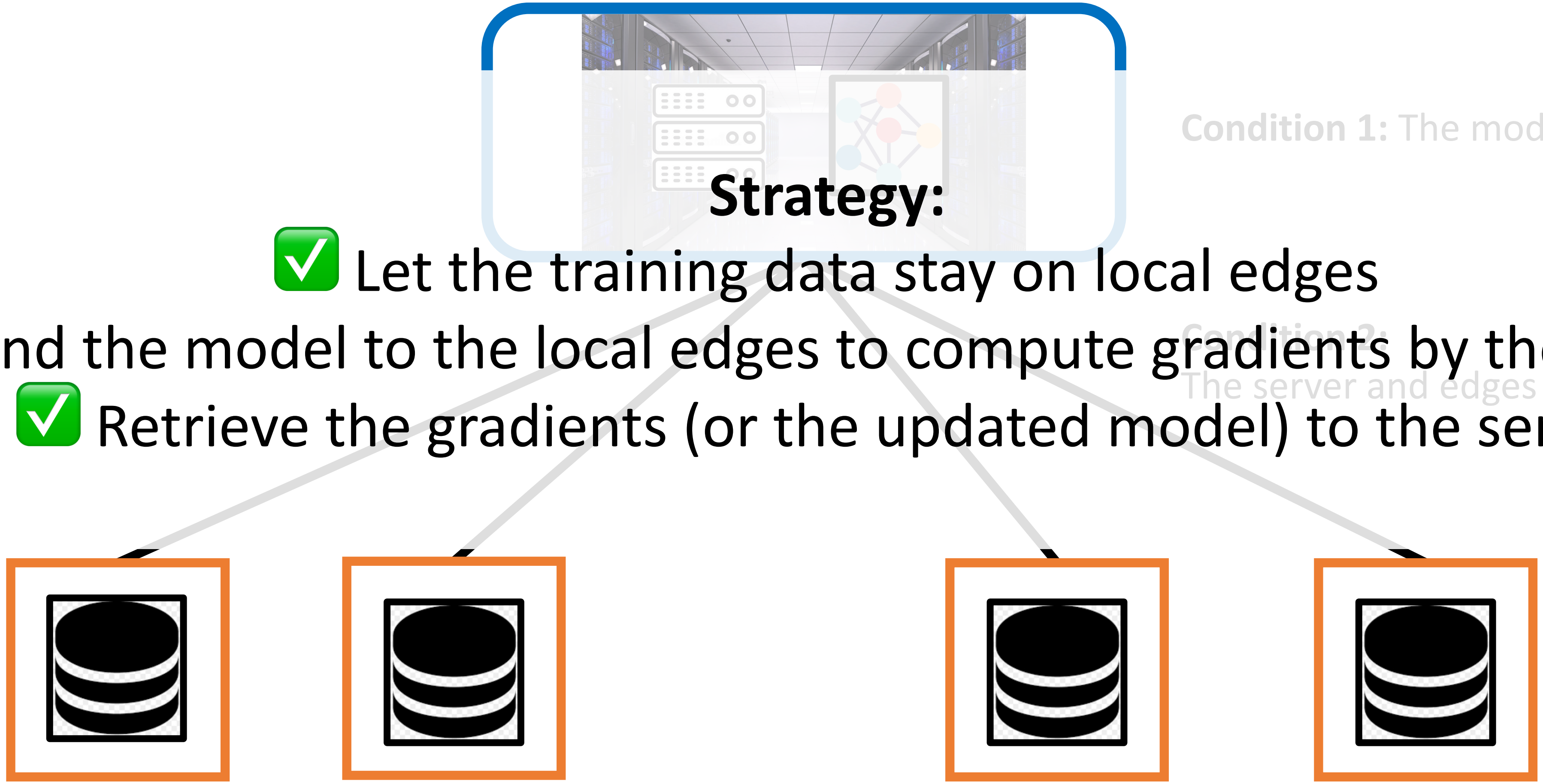
Condition 2:
The server and edges can communicate.

Condition 3:
Data stays on the edges.

Motivation: Learning on Distributed Clients

Let us consider a distributed environment where training data is on edges (or clients).

Central server



Condition 1: The model is on the server.

Strategy:

✅ Let the training data stay on local edges

✅ Send the model to the local edges to compute gradients by themselves.

✅ Retrieve the gradients (or the updated model) to the server

Condition 2: The server and edges can communicate.

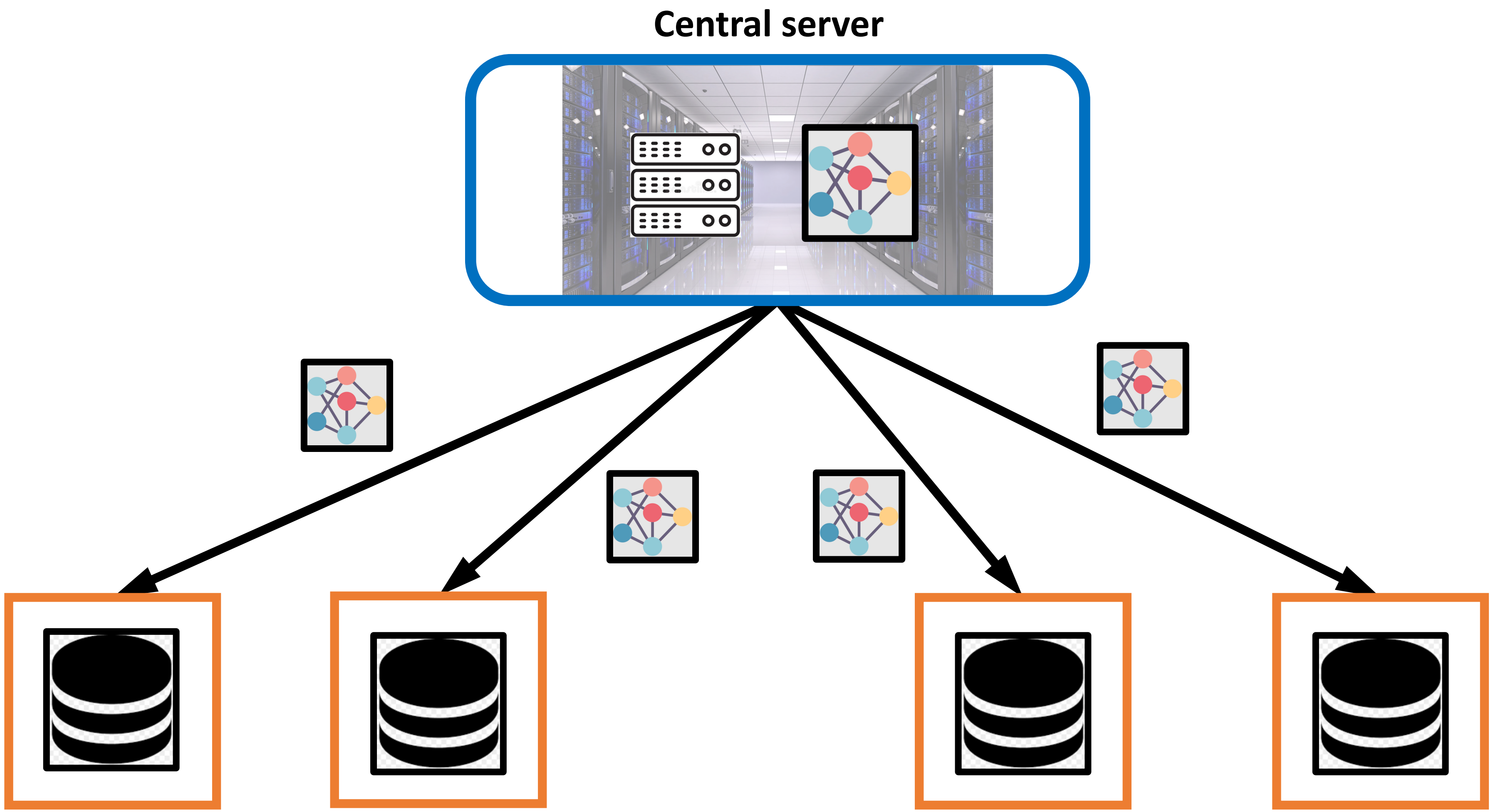
Condition 3: Data stays on the edges.

I. Federated Learning

- Motivations of Federated Learning
- Federated Averaging (FedAvg)

Federated Averaging of Models [FedAvg’17]

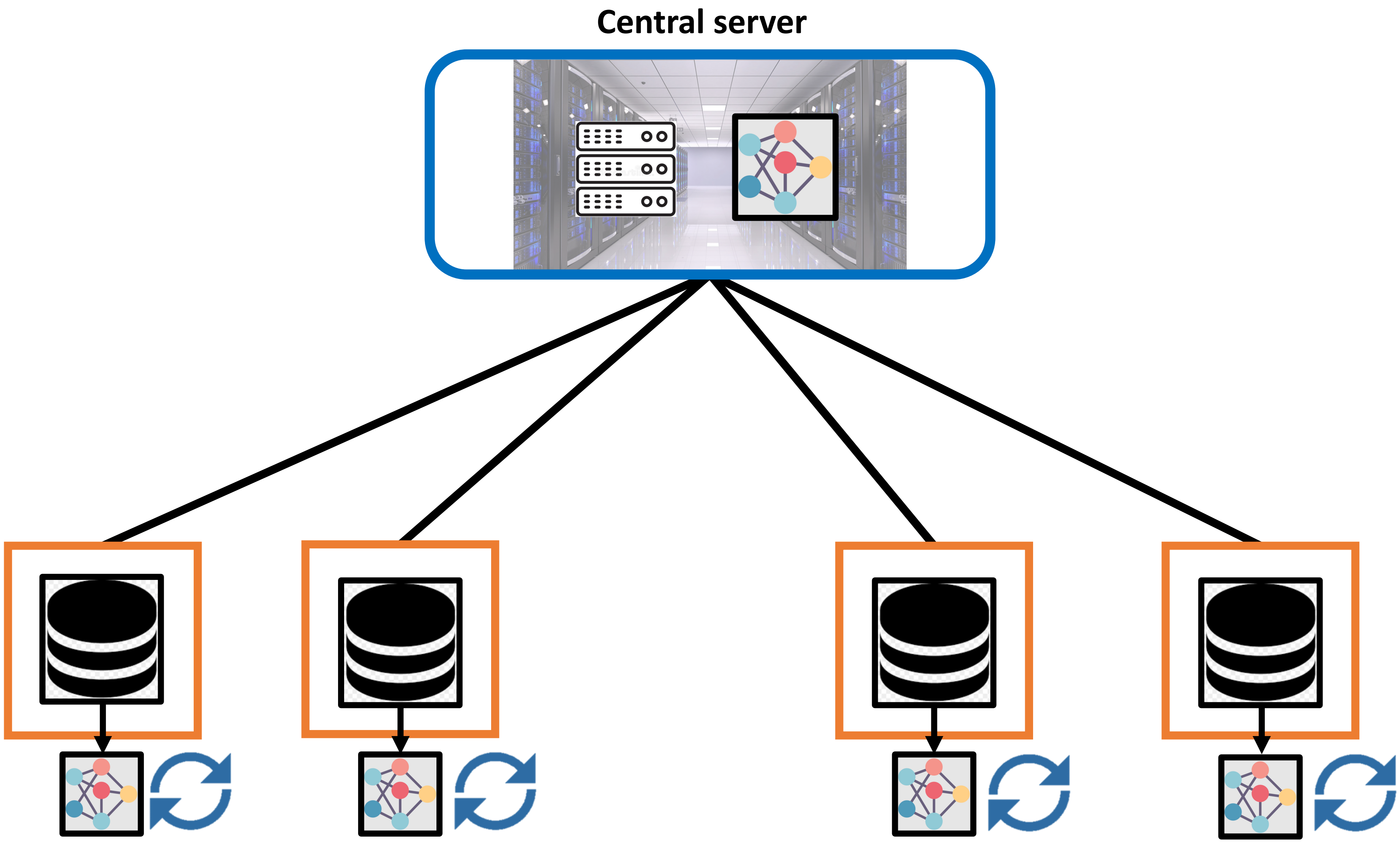
Step 1: Transmit the model from the server to all edges



[FedAvg’17] H. B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” AISTATS 2017

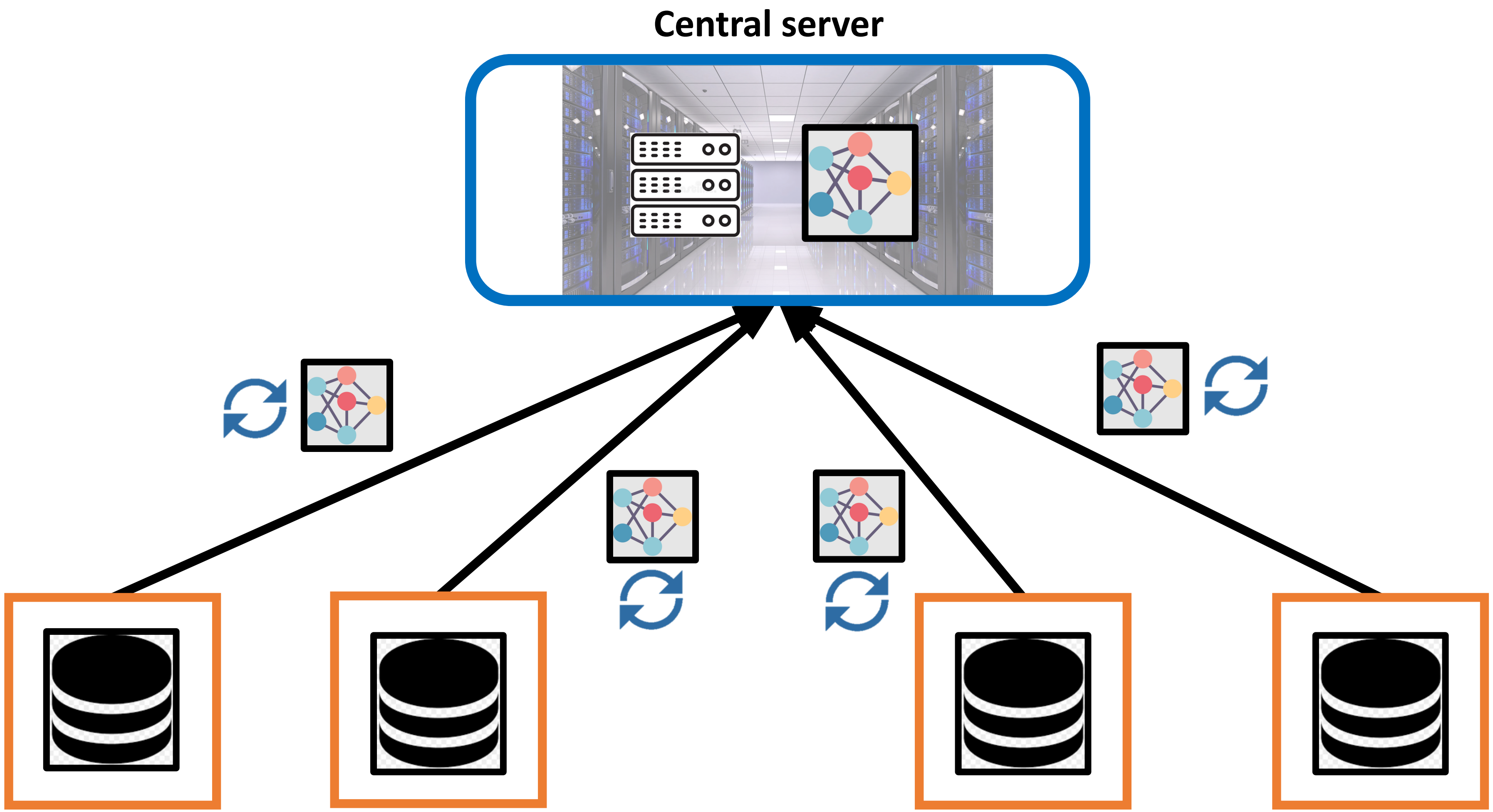
Federated Averaging of Models [FedAvg'17]

Step 2: Update the model by processing local data (or compute gradients)



Federated Averaging of Models [FedAvg'17]

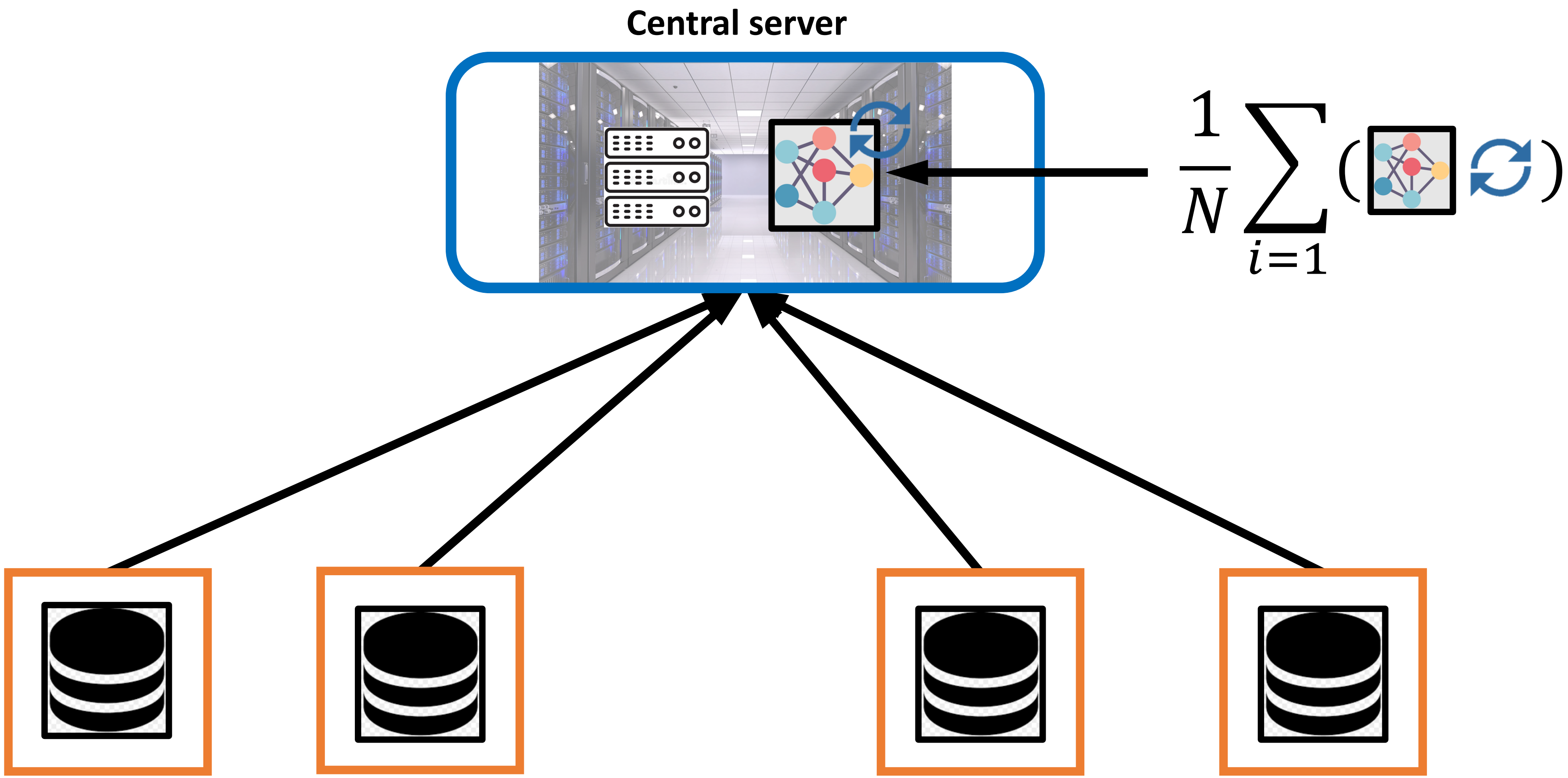
Step 3: Send back the updated model to the server



[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

Federated Averaging of Models [FedAvg'17]

Step 4: Merge the updated models to obtain a global model (e.g., averaging)

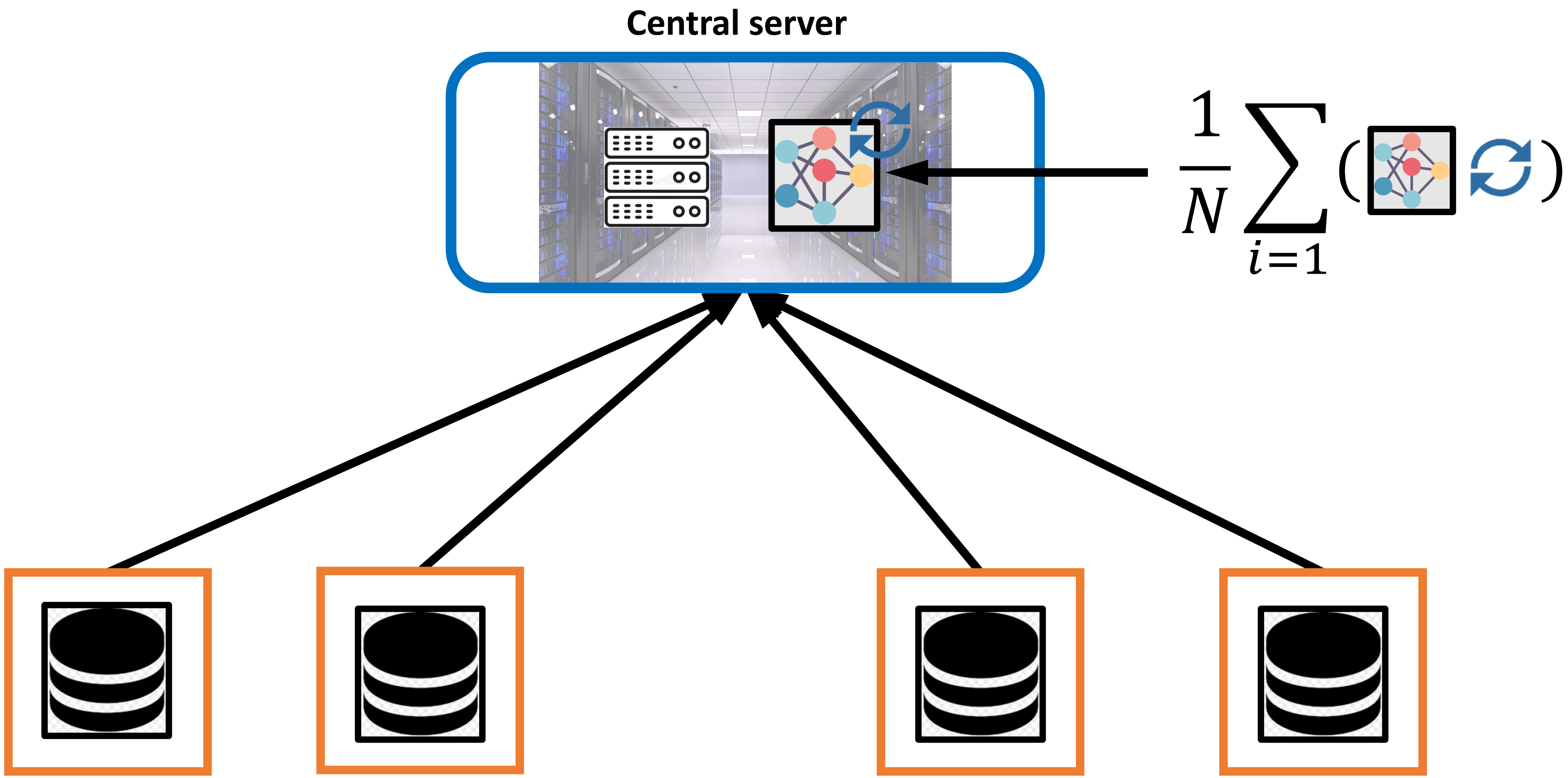


[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Federated Averaging of Models [FedAvg’17]

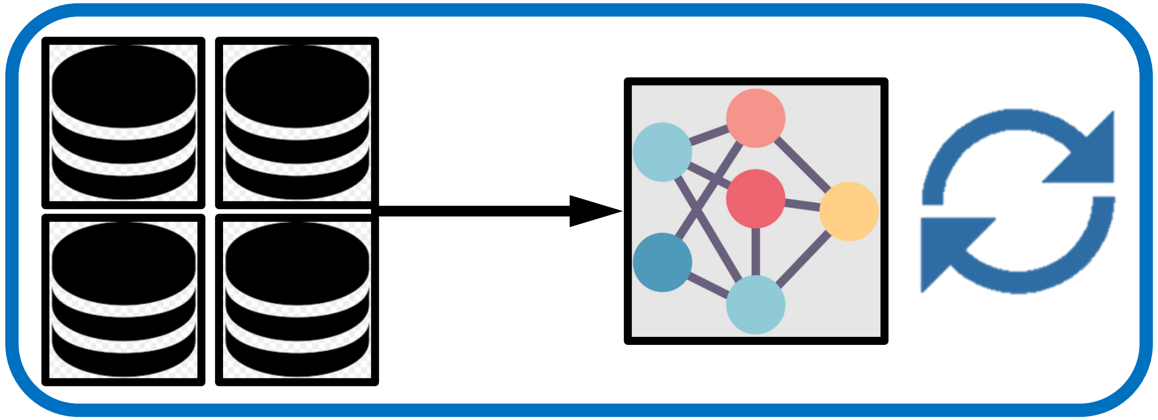
Step 5: Repeat step 1-4, until the model converges



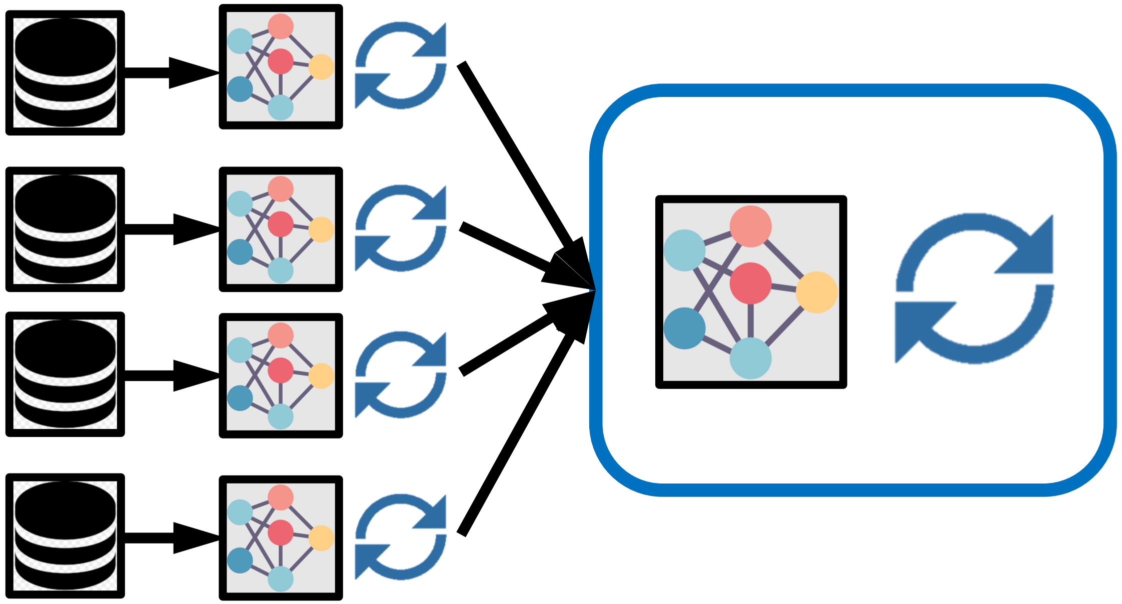
[FedAvg’17] H. B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” AISTATS 2017

Federated Averaging of Models [FedAvg'17]

Centralized training:



Learning on Distributed Clients:



[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

Federated Averaging of Models [FedAvg'17]

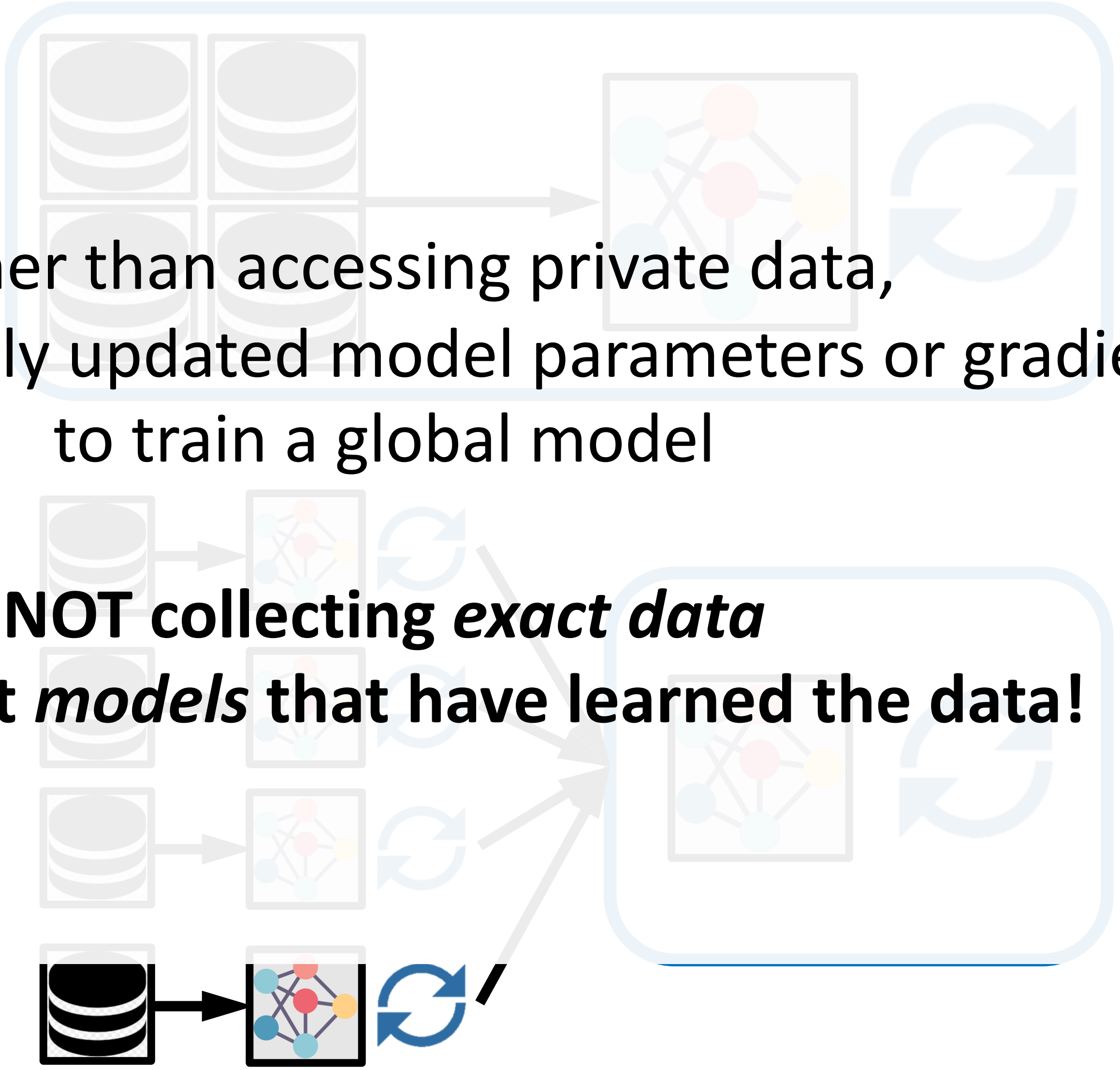
Centralized training:

Rather than accessing private data,
just collect locally updated model parameters or gradients
to train a global model

NOT collecting *exact data*

BUT collect *models* that have learned the data!

Learning on
Distributed Clients:



Federated Averaging of Models [FedAvg'17]

Pseudocode of FedAvg (taken from [FedAvg'17])

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w^k_{t+1} \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1}$

ClientUpdate(k, w)**:** *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

In addition, FedAvg considers some realistic settings:

✅ **Active clients -** not all clients participate in training (C-ratio)

✅ **Local computation** - local client processes B-size local minibatch
- Local client trains E local epoch

✅ **Data distribution (IID or non-IID)**

[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

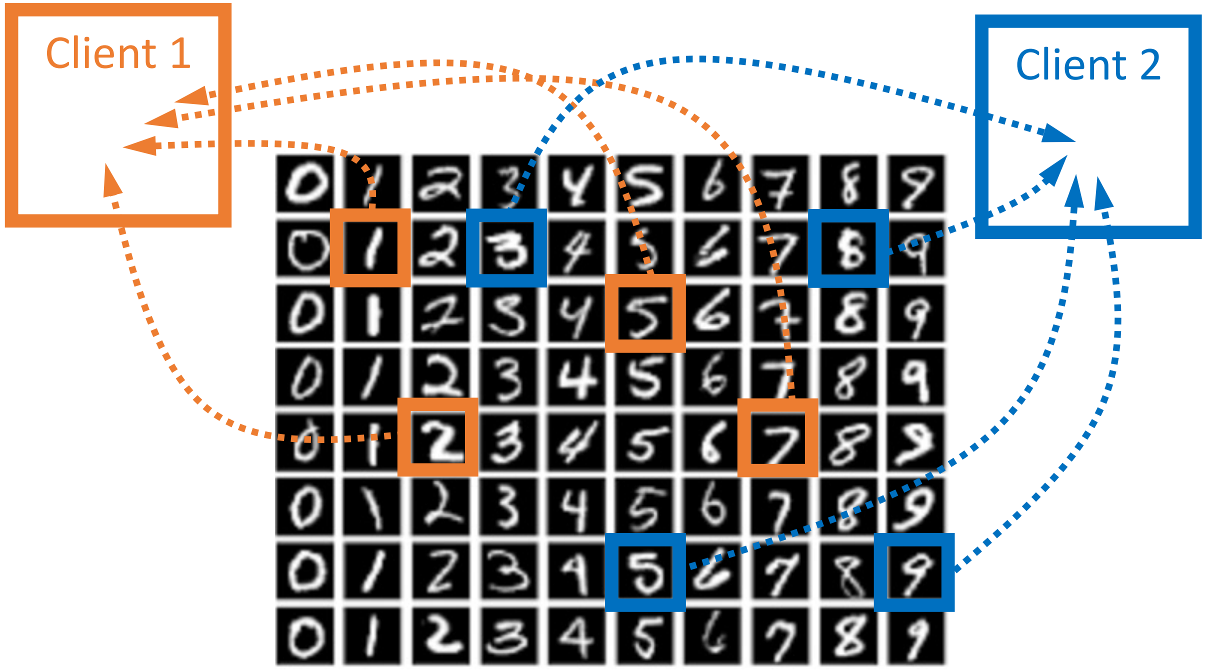
ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Federated Averaging of Models [FedAvg'17]

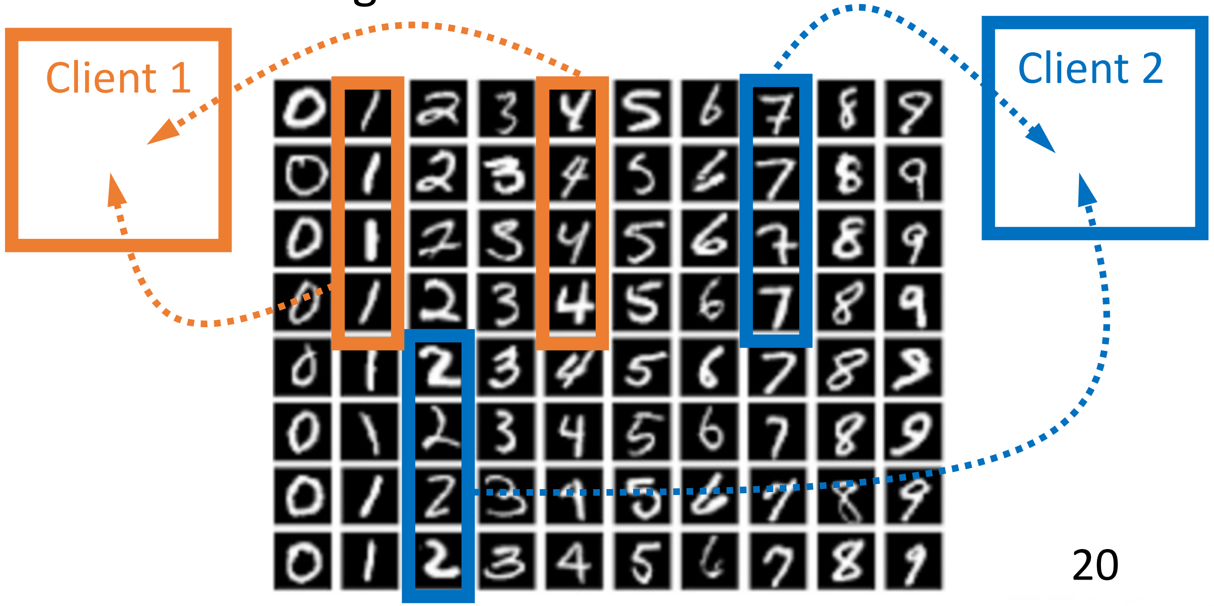
In addition, FedAvg considers some realistic settings:

✅ **Data distribution (IID or non-IID)**

IID case: data distribution on clients are IID
i.e., shuffle all images and partitioned them into clients



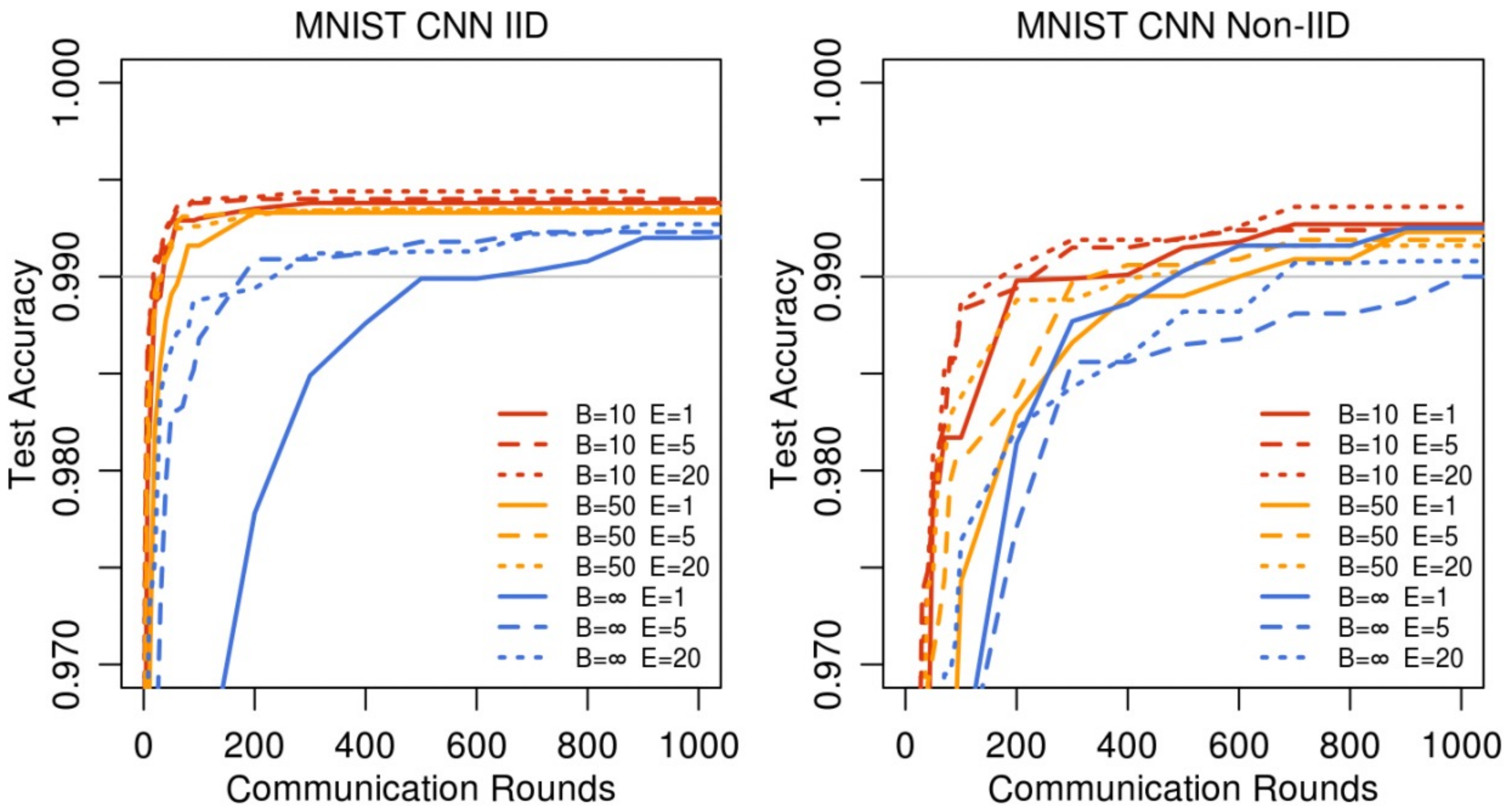
non-IID case: data distribution on clients are IID,
i.e., for each digit shuffle images and partition them into two shards, then each client selects two shards among all.



[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

Federated Averaging of Models [FedAvg'17]

FedAvg achieves a converging global model without collecting the local data samples (The test is done by the global model).



Taken from [FedAvg'17]

$C = 0.1$, i.e., 10% of clients are active in each round.

[FedAvg'17] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS 2017

II. Challenges of Federated Learning

- Heterogeneous FL, Personalization FL, Deep Leakage of FL

Federated Averaging on Heterogeneous Settings

When data distribution is non-IID, we have heterogeneity of data distribution.

In the work of FedAvg, authors consider a simple non-IID case.
However, a stronger heterogeneity can be introduced.

Many articles reported that
the strong heterogeneity hinders the convergence of FedAvg.

When local data is heterogenous, then the local gradients probably diverge.
It will hinder the fast convergence of FedAvg.

Federated Averaging on Heterogeneous Settings

In [ArXiv'18], significant performance degradation is observed for non-IID settings.
Here, non-IID(k) indicates that each client contains k-class images.

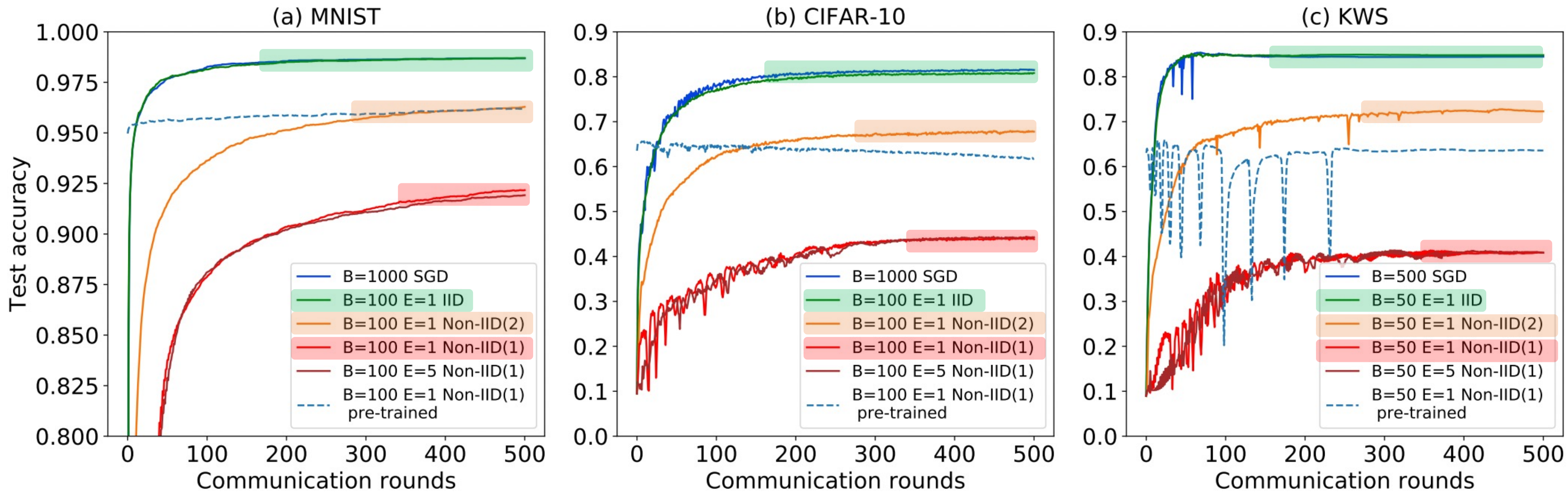


Figure 1: Test accuracy over communication rounds of *FedAvg* compared to SGD with IID and non-IID data of (a) MNIST (b) CIFAR-10 and (c) KWS datasets. Non-IID(2) represents the 2-class non-IID and non-IID(1) represents the 1-class non-IID.

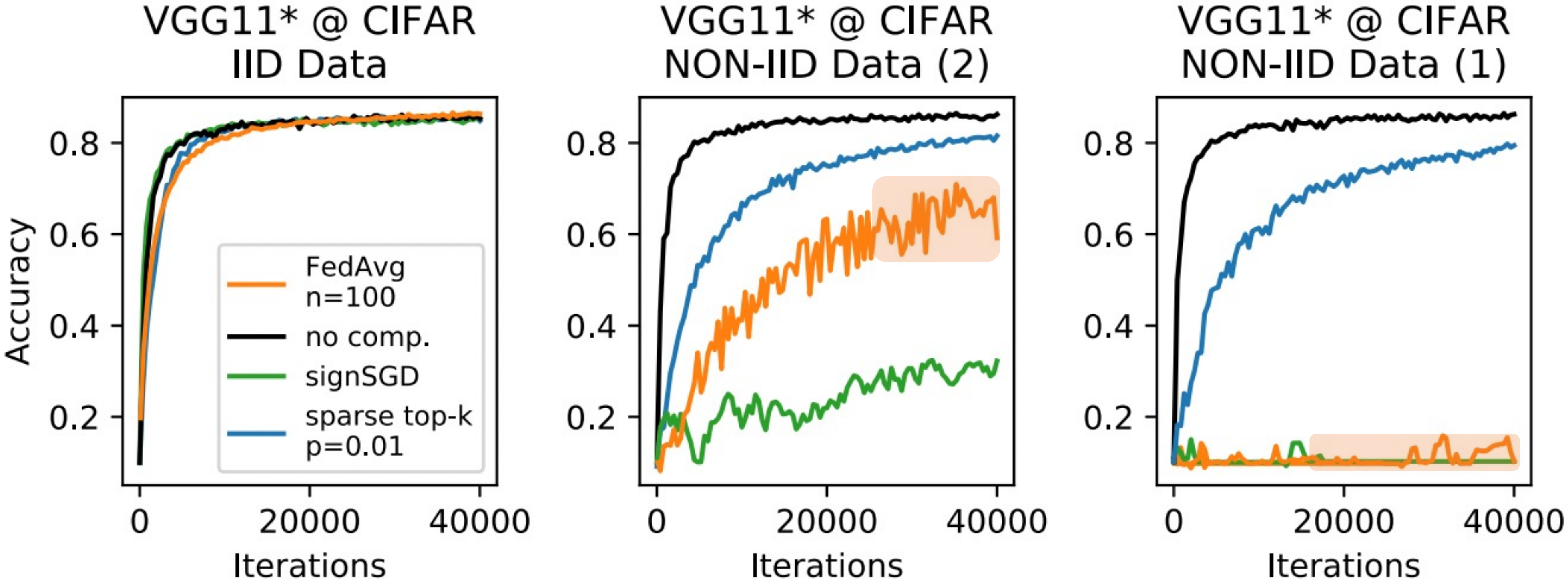
Taken from [ArXiv'18]

[ArXiv'18] Y. Zhao et al., "Federated Learning with Non-IID Data," arXiv 2018.

KWS: keyword spotting dataset

Federated Averaging on Heterogeneous Settings

In [T-NNLS'20], authors also point out that FedAvg suffers from non-IID settings.



Taken from [T-NNLS'20]

[T-NNLS'20] Y. Zhao et al., "Federated Learning with Non-IID Data," arXiv 2018.

KWS: keyword spotting dataset

Convergence FedAvg [ICLR'20]

In [ICLR'20], the convergence of FedAvg on non-IID is theoretically analyzed.

In summary, the convergence rate of FedAvg on non-IID is $\mathcal{O}(\frac{1}{T})$.

T means the total updates processed by clients.

Main contribution of [ICLR'20]:

$$\frac{T}{E} = \mathcal{O}\left[\frac{1}{\epsilon}\left(\left(1+\frac{1}{K}\right)EG^2 + \frac{\sum_{k=1}^{N} p_k^2\sigma_k^2 + \Gamma + G^2}{E}\right)\right]$$

ϵ: precision (lower means converging to the optimal)
E: updates done by each client per round.
T/E: total communication rounds

[ICLR'20] X. Li et al., "On the Convergence of FedAvg on Non-IID Data," ICLR 2020.

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Convergence FedAvg [ICLR'20]

Main contribution of [ICLR'20]:

$$\boxed{\frac{T}{E}} = \mathcal{O}\left[\frac{1}{\boxed{\epsilon}}\left(\left(1+\frac{1}{K}\right)EG^2 + \frac{\sum_{k=1}^{N} p_k^2\sigma_k^2 + \boxed{\Gamma} + G^2}{E}\right)\right]$$

ϵ: precision (lower means converging to the optimal)
E: updates processed by each client per round.
T/E: total communication rounds
Γ: a term quantifying the degree of non-IID

$$\Gamma = F^* - \sum_{k=1}^{N} p_k F_k^*$$

The minima of averaged loss

The average of local minima of loss

[ICLR'20] X. Li et al., "On the Convergence of FedAvg on Non-IID Data," ICLR 2020.

Personalized Federated Learning

Personalized federated learning is to provide local model for each client by federating the updates from all clients.

By leveraging the knowledge across clients, let us train strong local models.

Federation across clients ·····················► **Client-specific model**

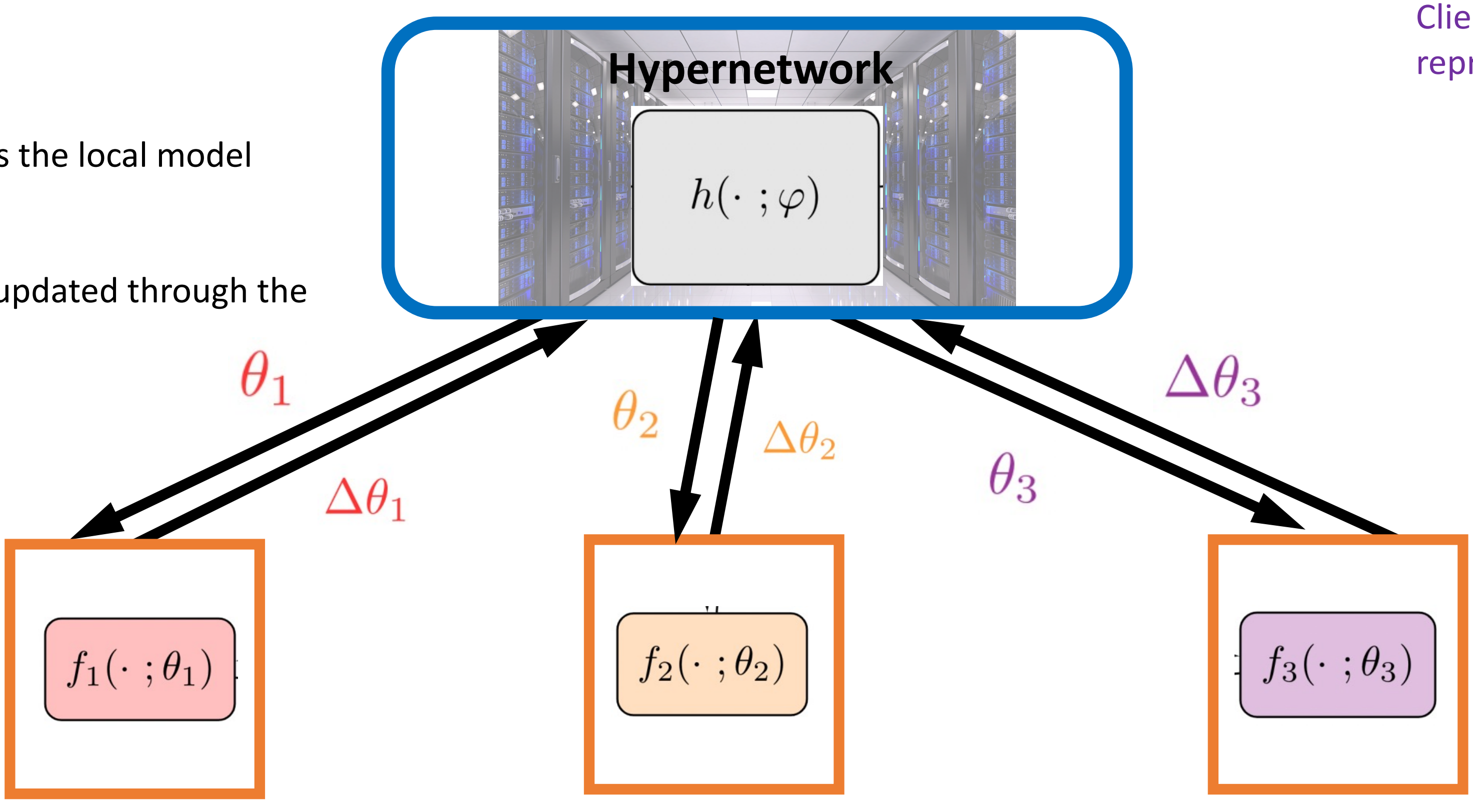
$$\Theta^* = \arg\min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{x},y\sim\mathcal{P}_i}[\ell_i(\boldsymbol{x}_j, y_j; \boxed{\theta_i})]$$

Personalized FL with Hypernetworks [pFedHN'21]

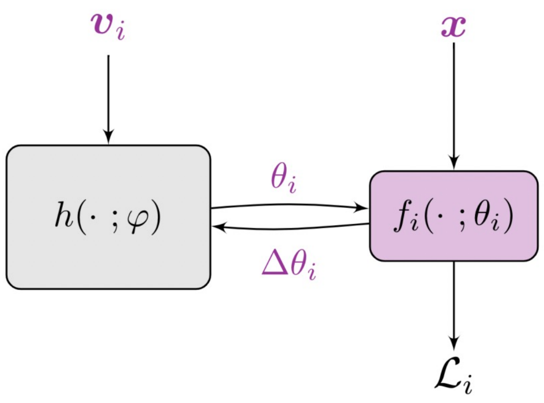
pFedHN of [pFedHN'21] trains hypernetworks through the federation across clients.
The FL-based hypernetworks are trained to generate local model weights.

Hypernetwork returns the local model parameters (θ_i).

The hypernetwork is updated through the gradients from locals.



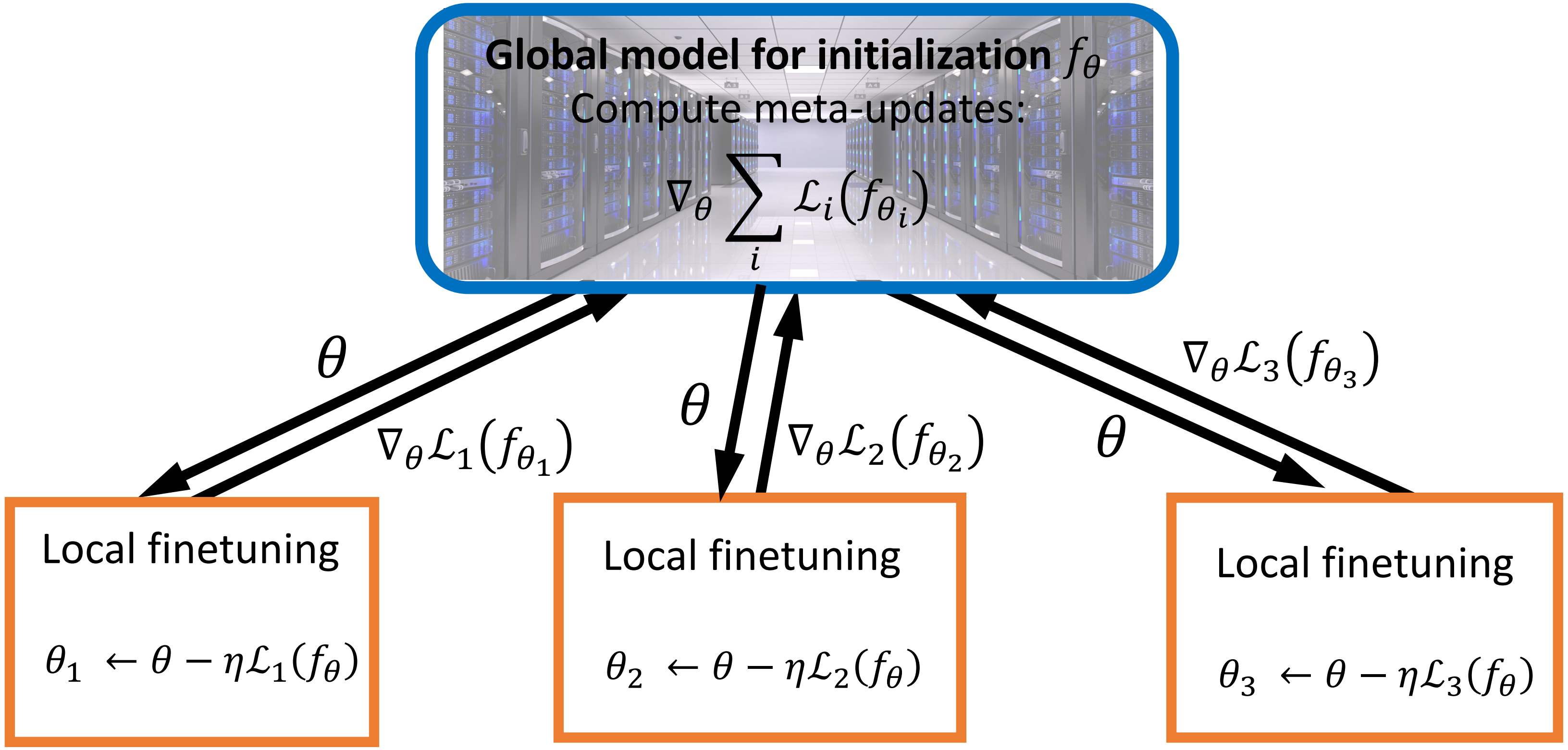
Client-specific representation vector



[pFedHN'21] A. Shamsian et al., "Personalized Federated Learning using Hypernetworks," ICML 2021.

Personalized FL with Meta-Learning [Per-FedAvg’20]

The strong ability for adaptation from meta-learning can be used for PFL.
In Per-FedAvg, Model-agnostic meta-learner (MAML) adopts few-shot-based finetuning for building local model from the federated global model.

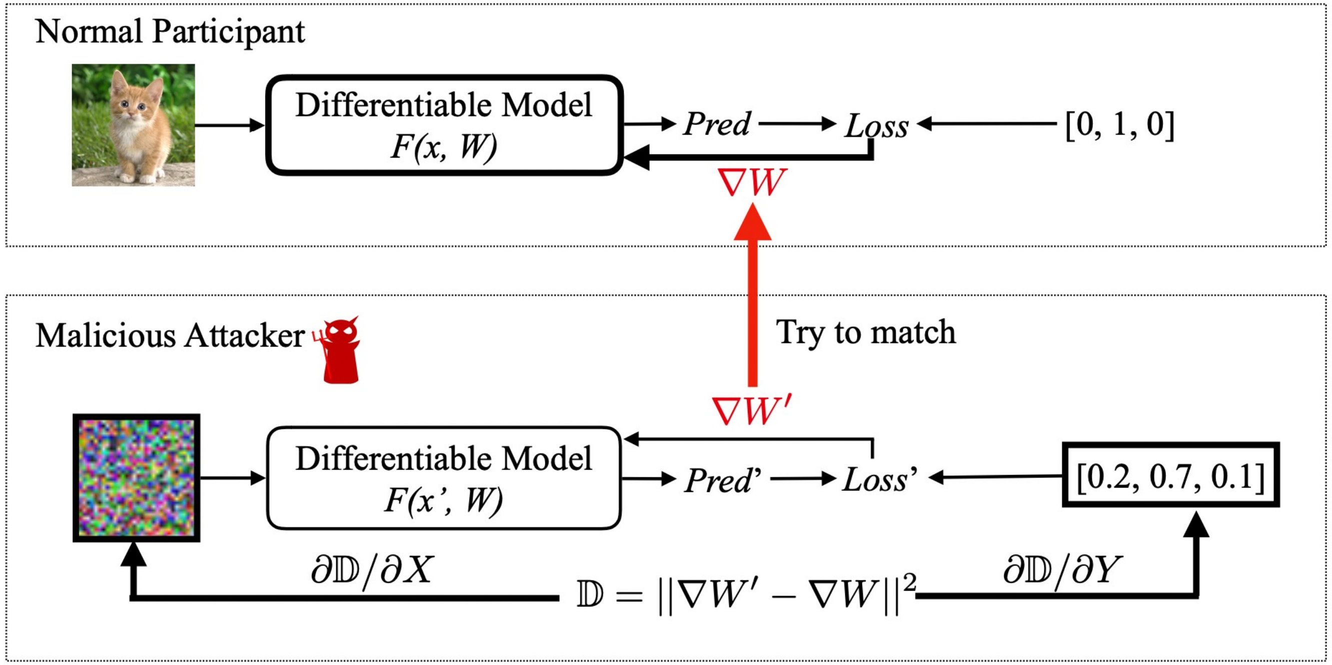


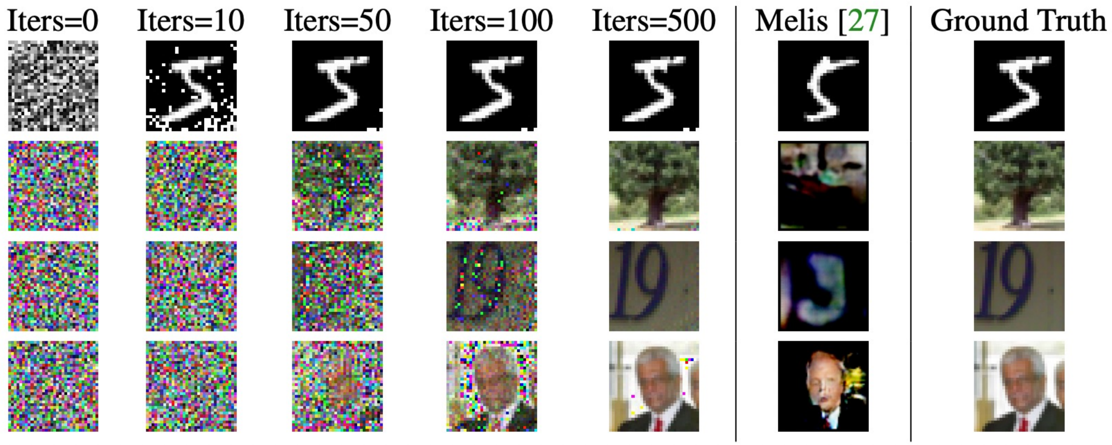
[Per-FedAvg’20] A. Fallah et al., “Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach,” NeurIPS 2020.

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Deep Leakage from Gradients [NeurIPS'19]

FedAvg relies on the belief that private data cannot be reconstructed from its gradients.
However, it is shown that gradients can be used to reconstruct original images.
By optimizing the dummy images to show a similar gradient to the original one,
server can reconstruct original images.





Figures taken from [NeurIPS'19]

[NeurIPS'19] L. Zhu et al., "Deep Leakage from Gradients," NeurIPS 2019.

III. Conclusions

- Federated Learning in Real-World Settings

Conclusions

- **Remaining Challenges of FL**
  - Handling data & model Heterogeneities across clients
    - Balancing between global and local optimization
    - Difference in model architecture across clients
  - Model splitting for better efficiency
    - Splitting model architecture into server/edge sides
  - Dynamic system variations
    - Federated learning on dynamic systems (structured and dynamic server-edge environment)
  - Robustness to adversarial attacks
    - Preventing deep leakage from gradients
    - Robustness to contaminated gradients from compromised nodes