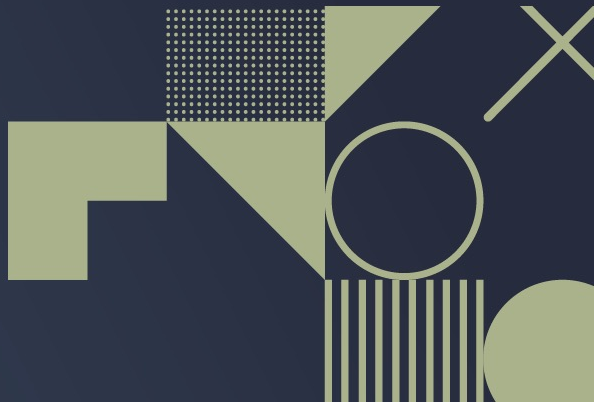
FedAT: A High-Performance and Communication-Efficient Federated Learning System with Asynchronous Tiers

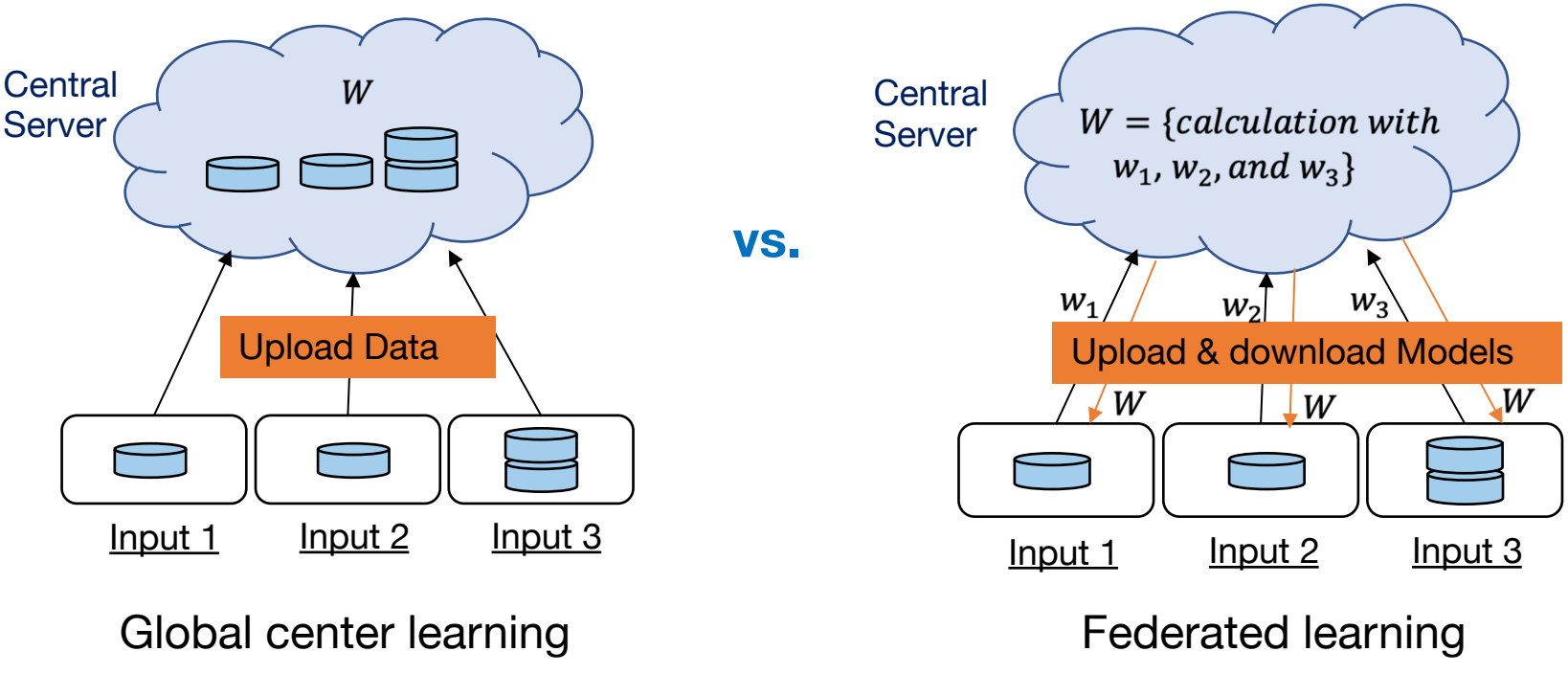
Zheng Chai[1], Yujing Chen[1], Ali Anwar[2], Liang Zhao[3], Yue Cheng[1], Huzefa Rangwala[1]

1 George Mason University
2 IBM Research
3 Emory University



Introduction and Motivation

Why federated learning?



Global center learning

Federated learning

Introduction and Motivation

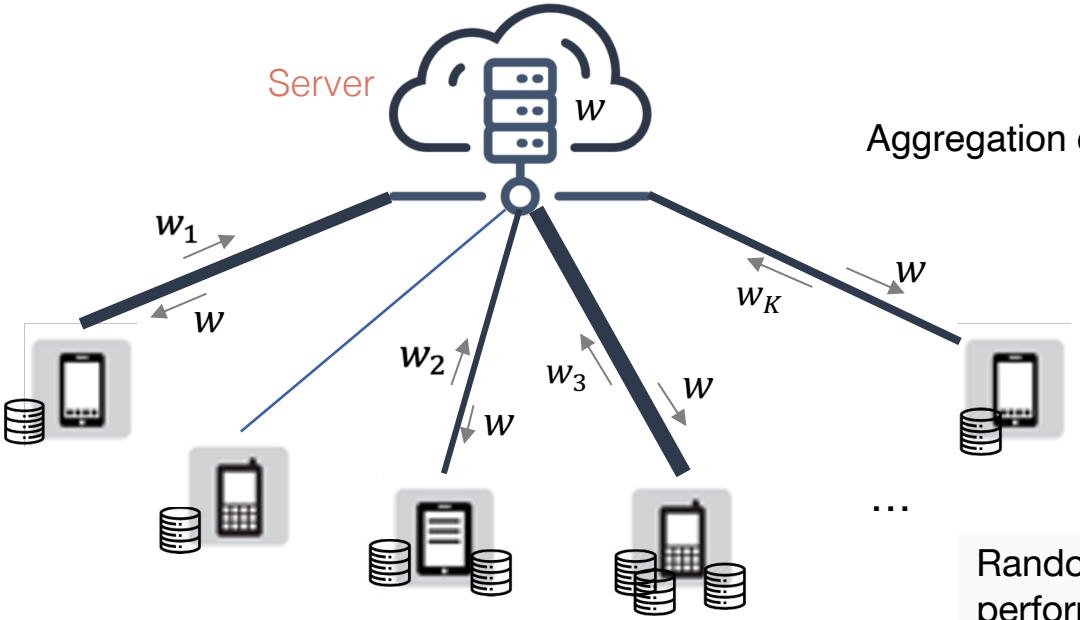
Why federated learning?

Federated learning

- is a practical approach for the *decentralized device data learning* of deep networks
- can *learn generalizable model* across all local inputs
- *protects sensitive user data* compared with the traditional central server training

Synchronous Federated Learning

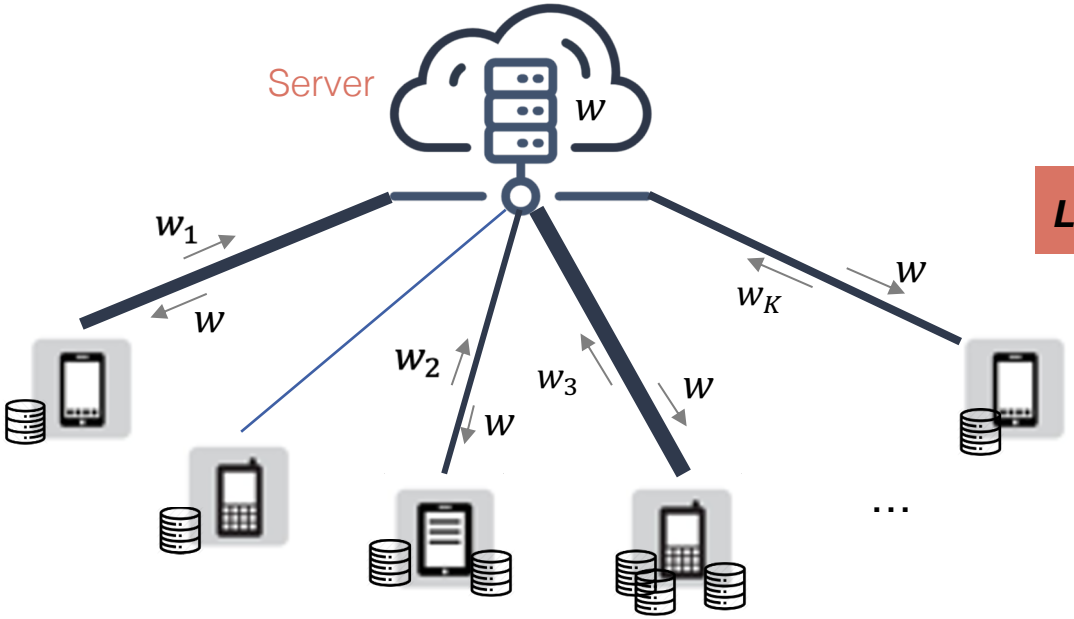
Vanilla approach: FedAvg (Federated Average)



Aggregation on server: $w = \sum_{k=1}^{K} \frac{n^k}{N} w_k$

Randomly select *a subset of local inputs* to perform local training at each round

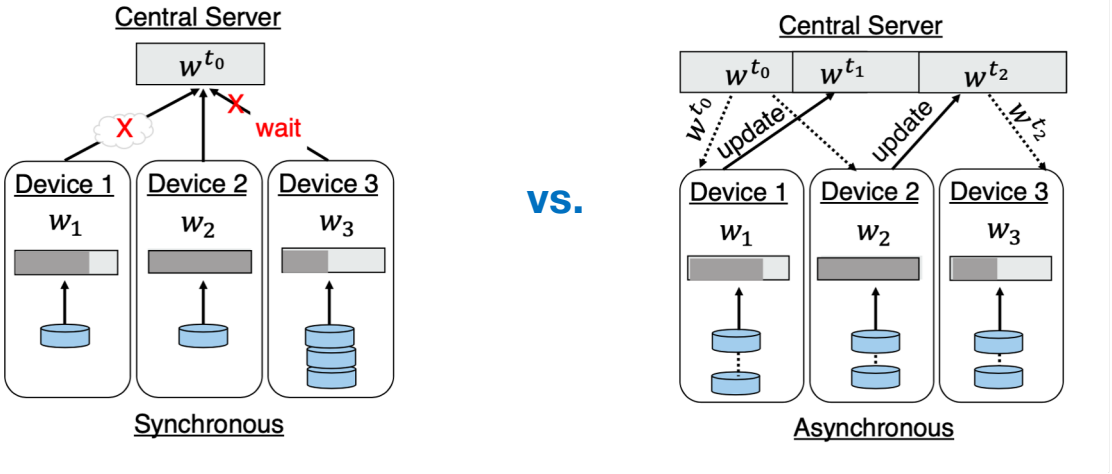
Synchronous Federated Learning



Limitation of Federated Average…

- Not perform well in heterogeneous federated environments

Limitations of synchronous FL frameworks



Synchronous model:

Lagging devices (stragglers) and dropouts will lead to idling and wastage of computing resources

Asynchronous model:

Server aggregation does not need to wait for lagging or dropped devices

Asynchronous Federated Learning

In asynchronous FLs, the server can aggregate without waiting for the straggling clients.

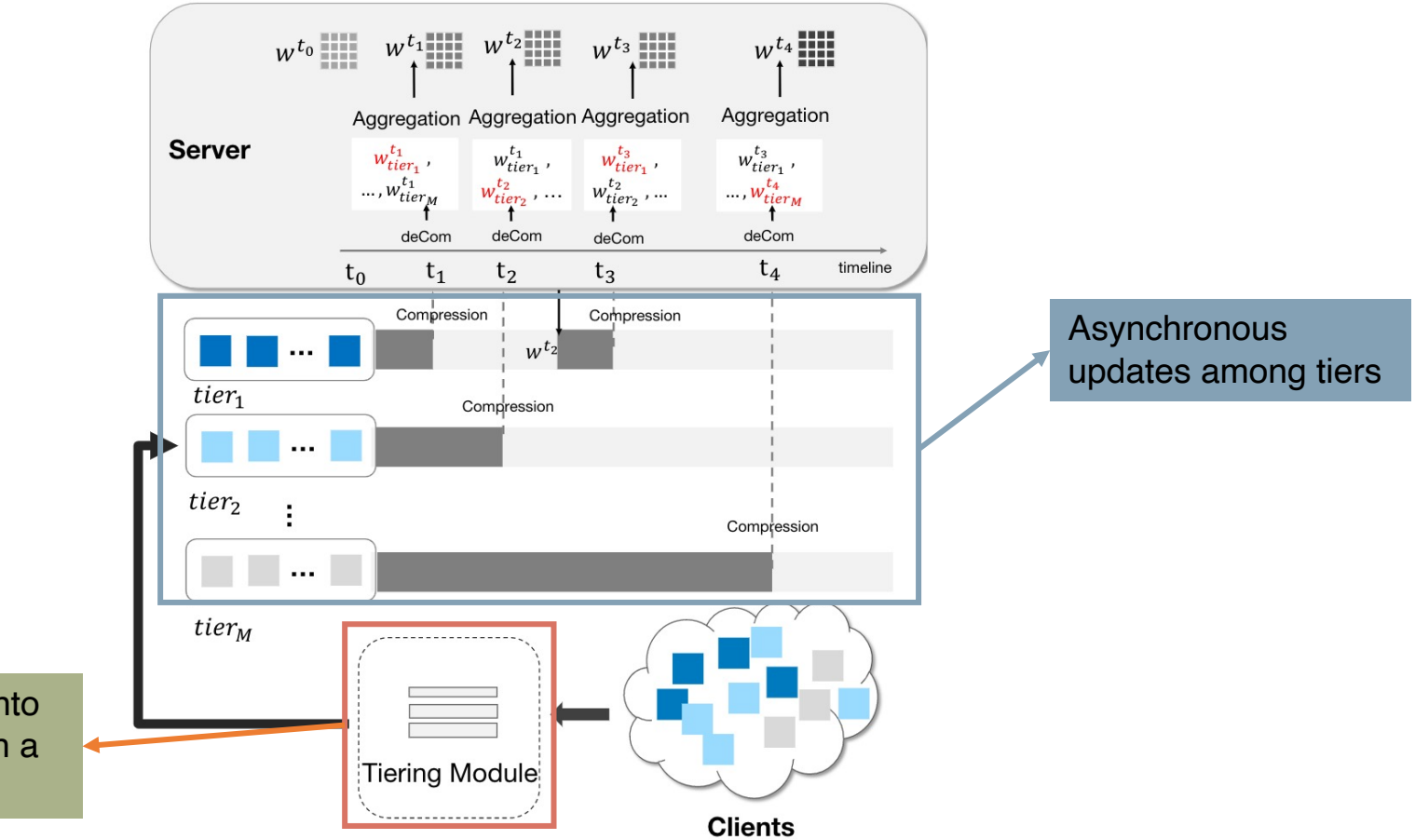
However…

- The server needs to communicate with each local device at each round, thus this model is not communication-efficient
- The server model may bias to devices which have more frequent updates

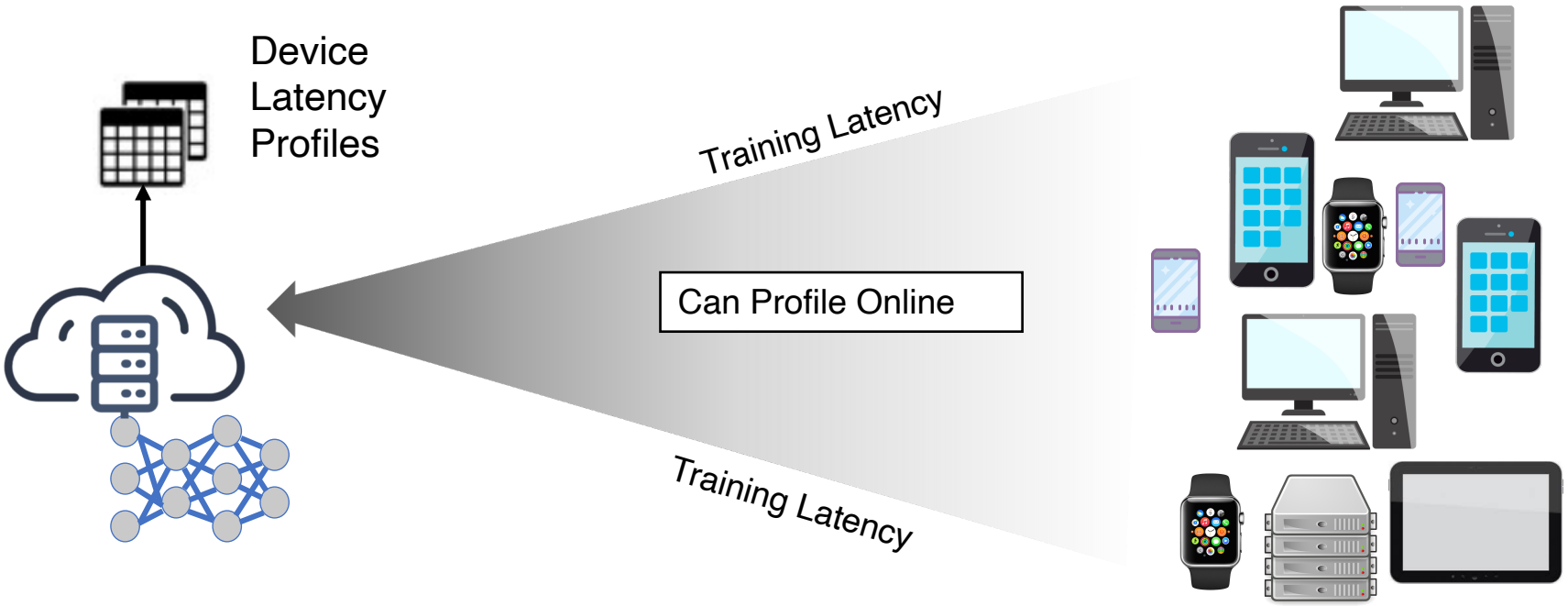
FedAT

A federated learning system that mitigates the stragglers, meanwhile maintains low communication costs with asynchronous tiers.

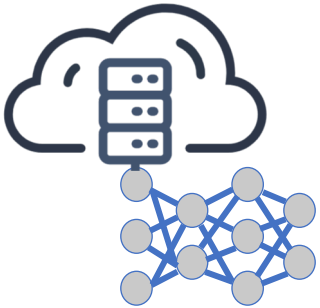
FedAT System Overview



FedAT: Profiling

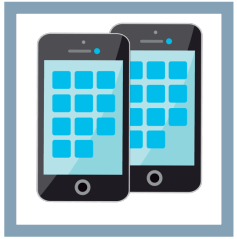


FedAT: Tiering



Devices with the same training latency are assigned into the same tier

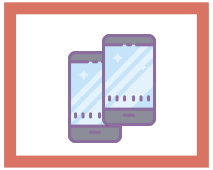
Tier 1



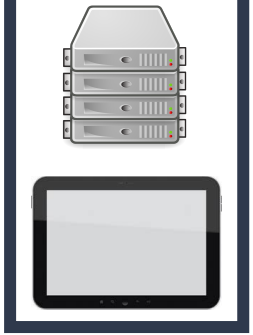
Tier 4

Tier 2

Tier 3



Tier 5



FedAT: Training Process

Algorithm 2: FedAT's Training Process

Input: w_{tier_m}, t, T and T_{tier_m}. w_{tier_m} denotes the weights of Tier m. t represents the global round t. T is the maximum global rounds. T_{tier_m} is the number of updates of tier m

Server: Initialize $w_{tier_1}, w_{tier_2} \ldots w_{tier_M}$ to w^{t_0}. Initialize $t, T_{tier_1} \ldots T_{tier_M}$ to 0

for *each tier* $m \in M$ ***in parallel*** **do**

  while $t < T$ **do**

    w^t = **WeightedAverage**$(w_{tier_1}, w_{tier_2} \ldots w_{tier_M})$

    $\mathcal{S}_m$ = (random set of clients from tier m)

    for *each client* $k \in \mathcal{S}_m$ ***in parallel*** **do**

      $n_k = |\mathcal{D}_k|$

      $w_k^{t+1} = w_k^t - \eta \nabla h(w^t)$

    $N_c = \sum_{k=1}^{|\mathcal{S}_m|} n_k$

    $w_{tier_m} = \sum_{k=1}^{|\mathcal{S}_m|} \frac{n_k}{N_c} \cdot w_k^{t+1}$

    $T_{tier_m} = T_{tier_m} + 1$

    $t = t + 1$

function WeightedAverage$(w_{tier_1}, w_{tier_2} \ldots w_{tier_M})$

if $t == 0$ **then**

  return w^{t_0}

else

  return $\sum_{m=1}^{M} \frac{T_{tier_{(M+1-m)}}}{T} \cdot w_{tier_m}$

Synchronous updates within each tier

FedAT: Local Learning

Local objective function:

$$h_k(w_k) = F_k(w_k) + \frac{\lambda}{2}||w_k - w||^2$$

Incorporating a constraint term to restrict the amount of local model deviation

Update of tier m:

$$f_{tier_m}(w) = \sum_{k=1}^{|\mathcal{S}_t|} \frac{n_k}{N_c} h_k(w_k) = \sum_{k=1}^{|\mathcal{S}_t|} \frac{n_k}{N_c} (F_k(w_k) + \frac{\lambda}{2}||w_k - w||^2).$$

FedAT: Aggregation on Server

Cross-Tier Weighted Aggregation

$$f(w) = \sum_{m=1}^{M} \frac{T_{tier_{(M+1-m)}}}{T} f_{tier_m}(w),$$

$\frac{T_{tier_{(M+1-m)}}}{T}$: relative weight of $tier_m$ and $\sum_{m=1}^{M} \frac{T_{tier_{(M+1-m)}}}{T} = 1$

Heuristic: a relatively slower tier would update less frequently thus it gets assigned a relatively larger weight value

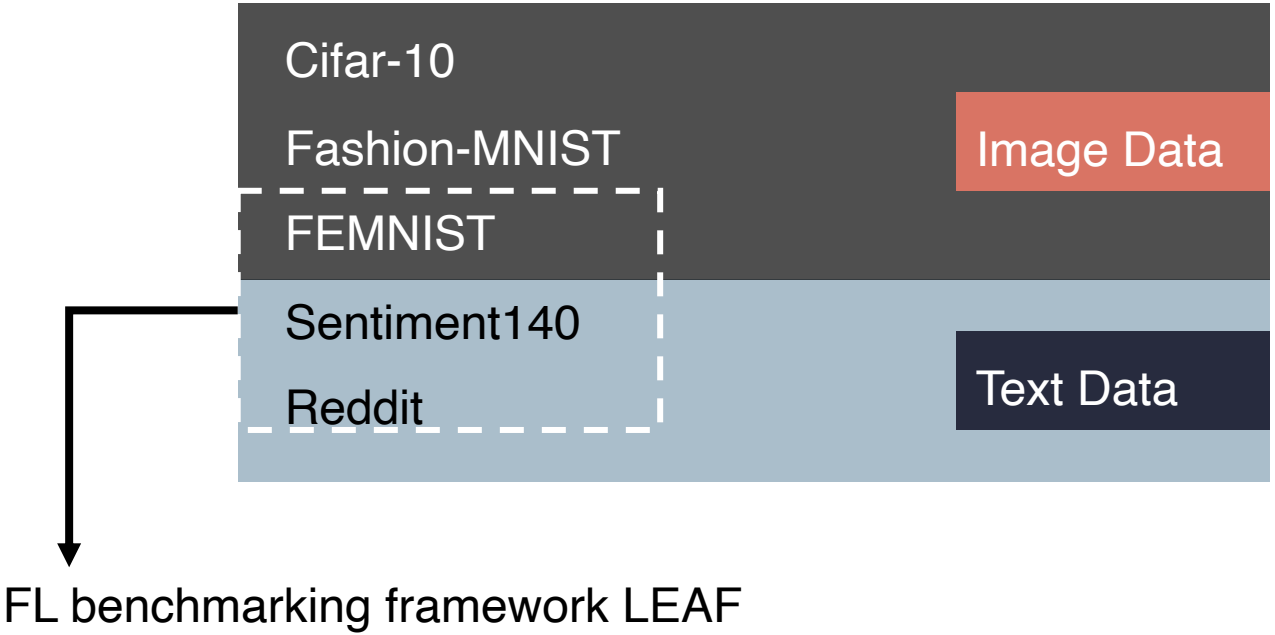
FedAT: Compression, Marshalling, and Unmarshalling

Polyline Encoding[1]

1. Lossy compression algorithm
2. Covert a series of numbers to a single string
3. Compress both uplink and downlink traffic

[1] https://developers.google.com/maps/documentation/utilities/polylinealgorithm

Datasets


Cifar-10
Fashion-MNIST
FEMNIST
Image Data
Sentiment140
Reddit
Text Data
FL benchmarking framework LEAF

Baseline FL Methods

Synchronous FL

FedAvg	The commonly used synchronous federated learning approach proposed by McMahan et al..
FedProx	Synchronous federated learning framework with a **proximal term** on the local objective function to mitigate the data heterogeneity problem and to improve the model stability compared to FedAvg.
TiFL	A synchronous FL method that partitions training clients into different tiers based on their responding latency. The aggregation method of TiFL is adopted from FedAvg.

Baseline FL Methods

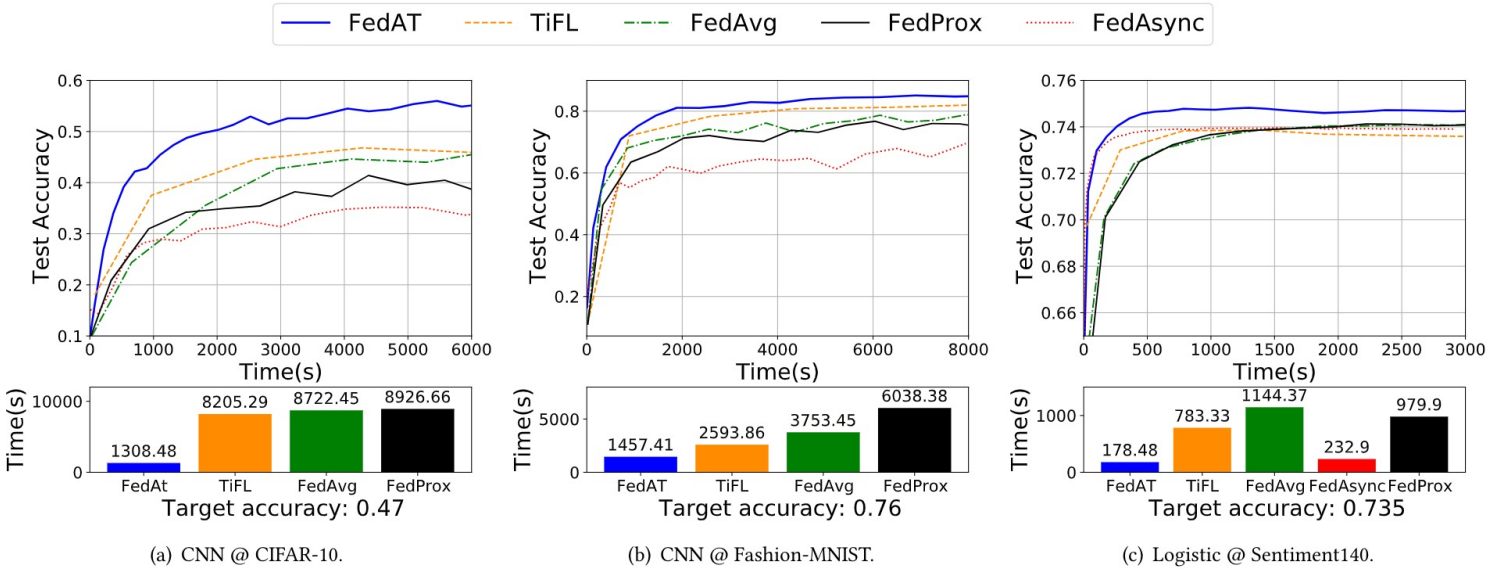
Asynchronous FL

FedAsync Asynchronous federated learning framework using a **weighted average** to update the server model with IID data.

ASO-Fed Asynchronous online federated learning approach with a more close to real-world assumption on the settings of edge devices.

Prediction Performance

Performance comparison of different FL methods on 2-class Non-i.i.d. CIFAR-10, Fashion-MNIST, and Sentiment140 datasets

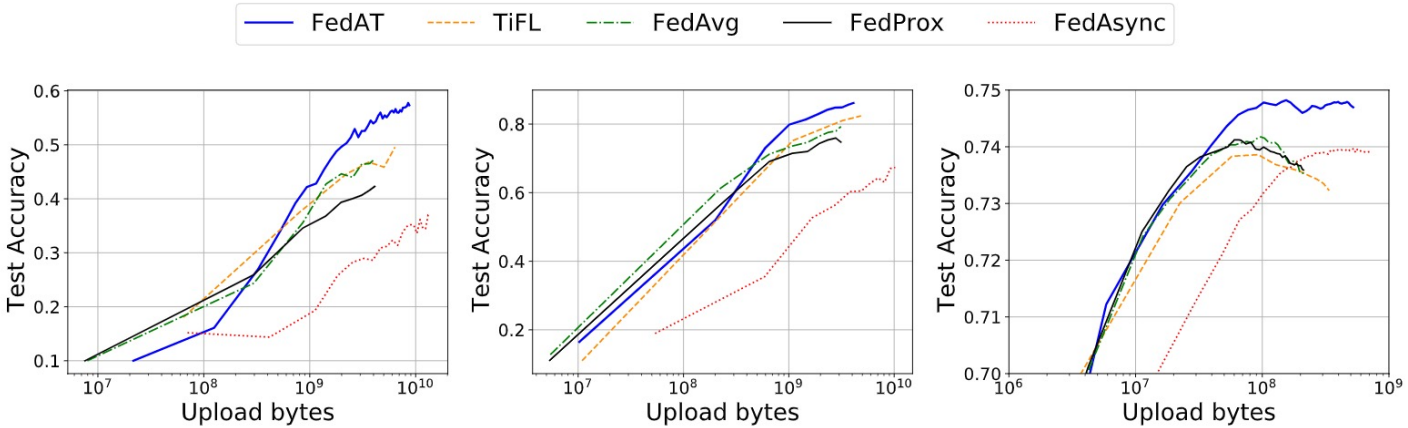


(a) CNN @ CIFAR-10. (b) CNN @ Fashion-MNIST. (c) Logistic @ Sentiment140.

Note: FedAsync is not able to reach the target accuracy for CIFAR-10 and Fashion-MNIST, thus is omitted

Communication Analysis

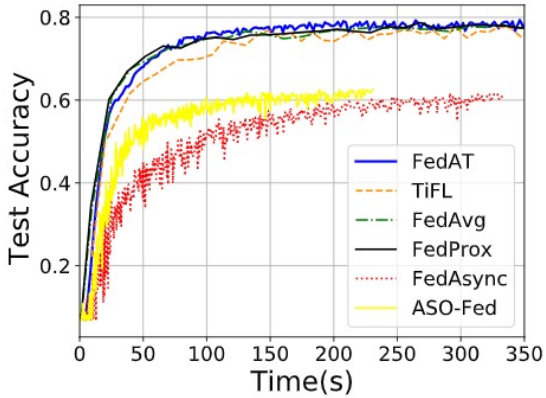
Test accuracy as a function of the cumulative amounts of data uploaded from clients to the server for 2-class Non-i.i.d.datasets.



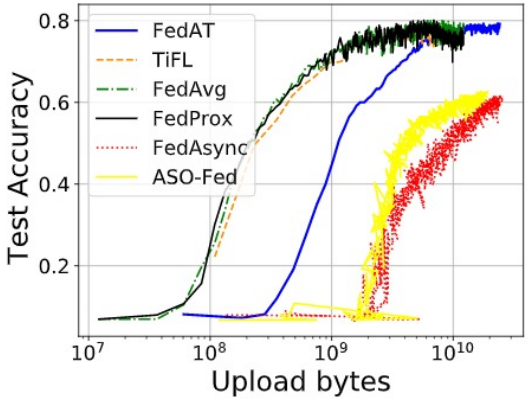
(a) CIFAR-10. (b) Fashion-MNIST. (c) Sentiment140.

Results of Large-scale Training

Prediction accuracy of FEMNIST as a function of training time (a) and cumulative amount of data uploaded from clients to the server (b).



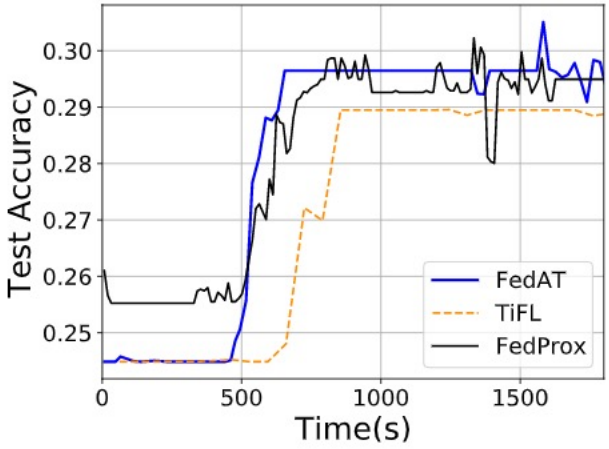
(a) Accuracy over time.



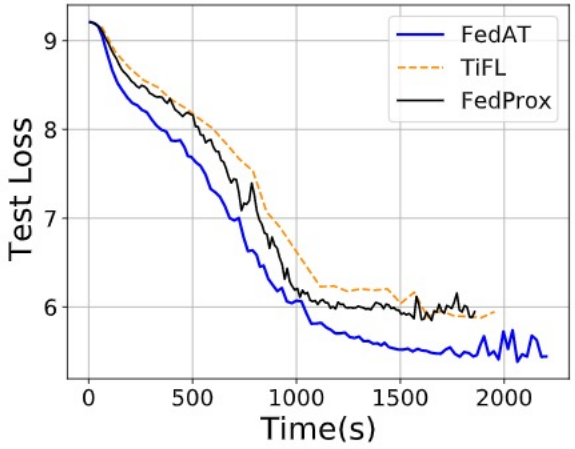
(b) Accuracy over uploaded bytes.

Results of Large-scale Training

Prediction accuracy of Reddit as a function of training time (a) and cumulative amount of data uploaded from clients to the server (b).



(a) Accuracy over time.



(b) Loss over time.

Summary

In this work, we propose FedAT, which

- is a novel, tiered, FL framework, which updates local model parameters *synchronously within tiers* and updates the global model *asynchronously across tiers*;
- has a *new optimization objective* with a weighted aggregation heuristic to speed up the model convergence and improve the prediction performance;
- has *provable convergence guarantee* for both convex and non-convex objectives;
- includes a lossy compression technique to compress the transferred model data between clients and server to *reduce the communication cost without affecting the model accuracy*.

Thank you!

Q&A