Federated Learning

From the Perspective of Optimization

Li Ju

Uppsala University

October 11, 2025

Outline

1 From Centralized Learning to Federated Learning

2 Federated Optimization Problem

3 An Introduction to Our Work

Centralized Learning

Problem:
We have a dataset $\{(x_i, y_i)\}_{i=1}^{I}$, we want to model the unknown function $y = g(x)$

Neural Network:
A function approximator $f : \mathbb{R}^n \to \mathbb{R}^p$ parameterized by $W := \{A_k, b_k; k \in [K]\}$:

$$f(x) := f_K \circ f_{K-1} \cdots \circ f_1$$
$$\text{where } f_k = \sigma_k(A_k x + b_k) \text{ for } k \in [K]$$

Centralized Learning

Target parameters W^*:

With a defined loss function ℓ, we use empirical risk minimization:

$$W^* := \arg\min_W \sum_{i=1}^{I} \frac{\ell(f(x_i; W), y_i)}{I}$$

How to solve:

Generally first-order methods: (stochastic) gradient descent

$$W^t := W^{t-1} - \eta \nabla_W$$

Distributed Learning vs Federated Learning

Distributed Learning:

Challenges:

- Large models
- Large amount of data

Solutions:

- Model parallelization
- Data parallelization

`Distributed SGD:`

$$\nabla_W = \frac{\sum_{n=1}^{N} \nabla_W^n}{N}$$

$$W^t := W^{t-1} - \eta \nabla_W$$

Federated Learning:

Challenges:

- Ones from DL
- Intrinsic distributed data
- Security concern
- Prohibitive communication cost

How to do federated optimization?

Baseline Algorithm

Baseline algorithm: `FedAvg`[1]

Algorithm 1 `FedAvg`

Require: Initialize parameters W^0
  for round t in $\{1, \dots T\}$ **do**
    for client k in $\{1, \dots K\}$ **parallel do**
      Iterate `SGD` for n steps: $W_k^t = \texttt{SGD}(W^{t-1}, n)$ ▷ Client-side
      $(\Delta_k^t := W_k^t - W^t)$
    end for
    $W^t = \sum_{k=1}^{K} W_k^t / K$
    $(W^t = W^{t-1} + \sum_{k=1}^{K} \Delta_k^t / K)$ ▷ Server-side
  end for

In ideal cases, the communication cost is reduced to $\mathcal{O}(\frac{1}{n})$

[1] McMahan et al. 2016.

Generalized Framework

`FedOpt` from Adaptive Federated Optimization[2]:

Algorithm 2 `FedOpt`

Require: Initialize parameters W^0
  for round t in $\{1, \ldots T\}$ **do**
    for client k in $\{1, \ldots K\}$ **parallel do**
      $W_k^t := \textbf{ClientOpt}(W^{t-1})$ ▷ Client-side
      $\Delta_k^t := W_k^t - W^t$
    end for
    $\Delta^t := \textbf{Aggre}(\{\Delta_k^t, 0 \leq k < K\})$ ▷ Server-side
    $W^{t+1} := \textbf{ServerOpt}(\Delta^t)$
  end for

ServerOpt
GD, *Nestrv. GD*, *Adam*, etc.

ClientOpt
SGD, *Nestrv. SGD*, *Adam*, *AdaGrad*, etc.

Aggre
Averaging, *Medianing*, etc.

`FedAvg`: *SGD* + Averaging + *GD* with $\eta = -1$

[2] Reddi et al. 2020.

Problems of Federated Optimization

There are still problems in federated optimization:

- Statistical heterogeneity
- Computational heterogeneity
- Additional privacy constraints
- Communication efficiency
- ...

Model Fairness

Case Study: Non-iid partitioned MNIST + Multi-Layer Perceptron

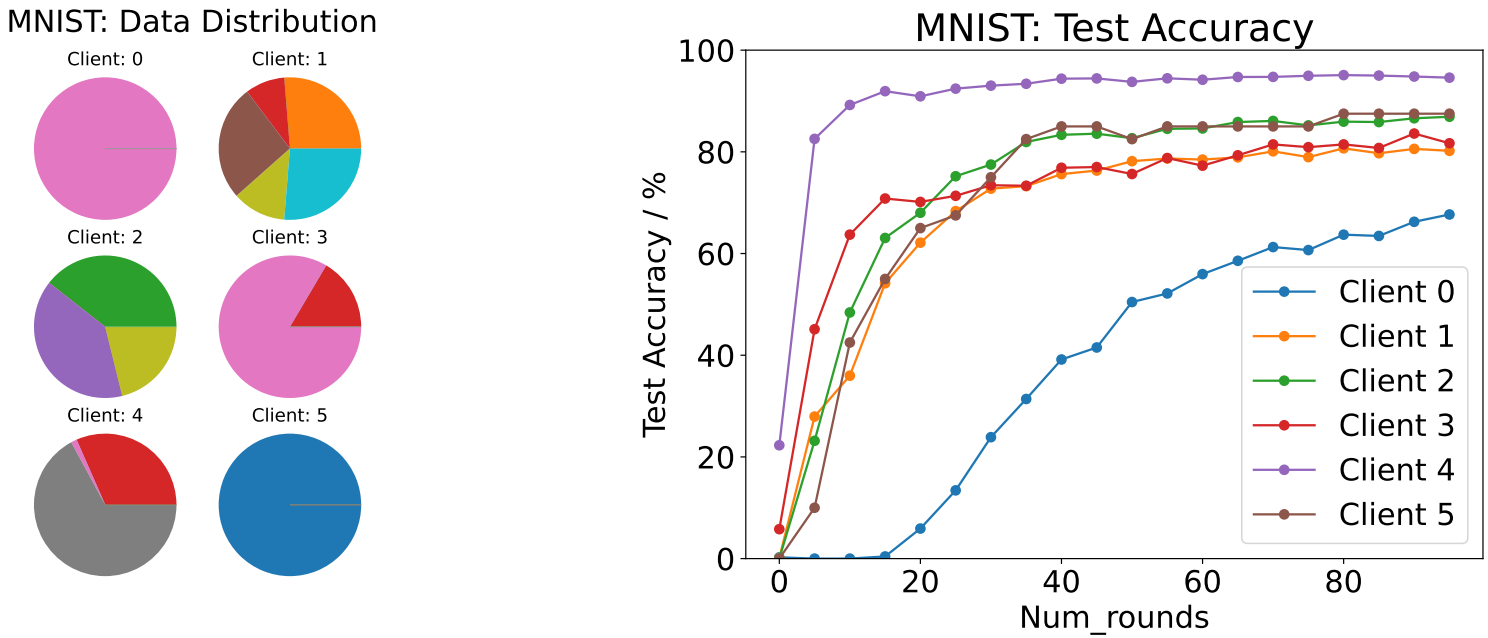


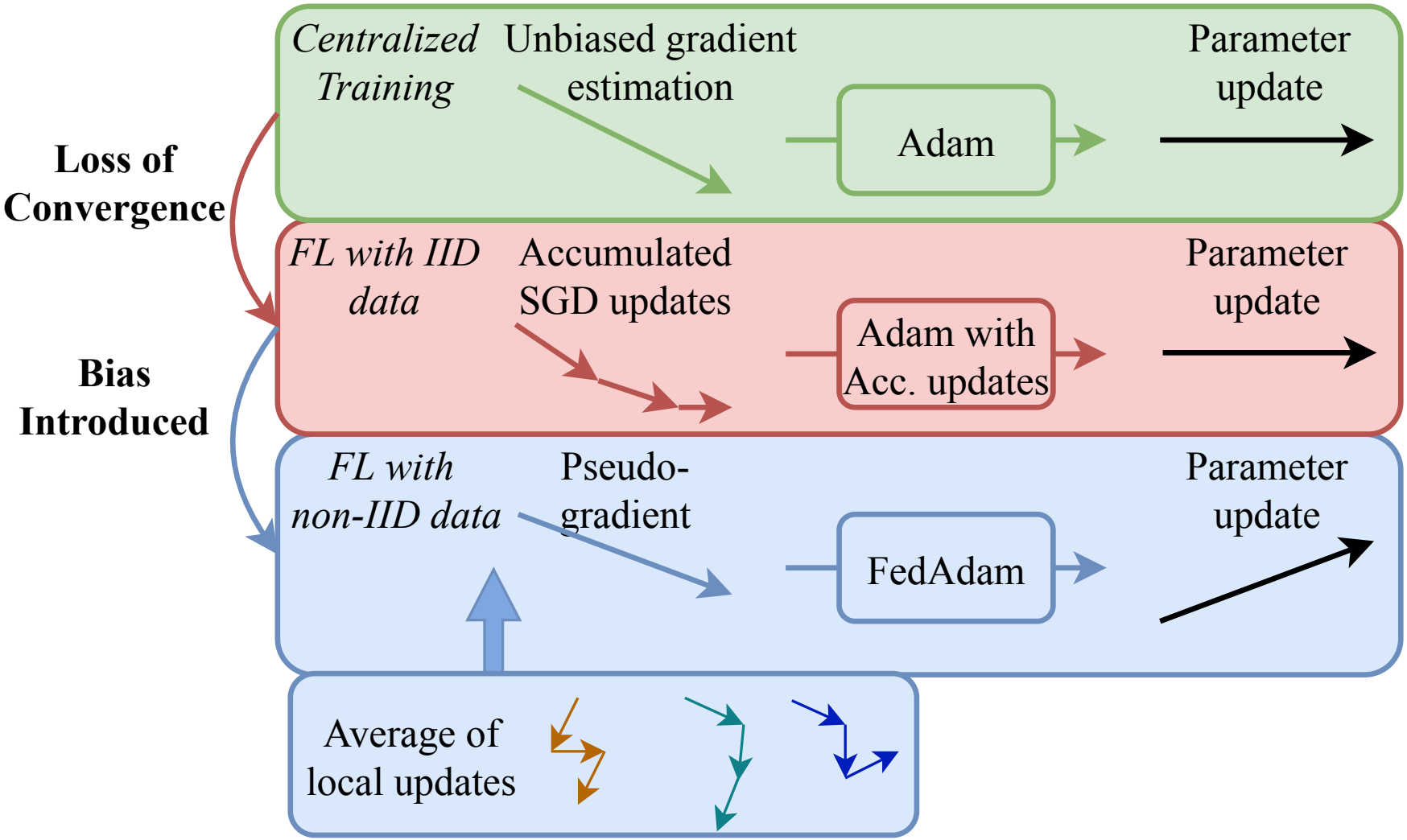
Figure: **Left**: Distributions of local datasets **Right**: Training curves

Fairness problem: Differences of model performance across participants in a federated training process.

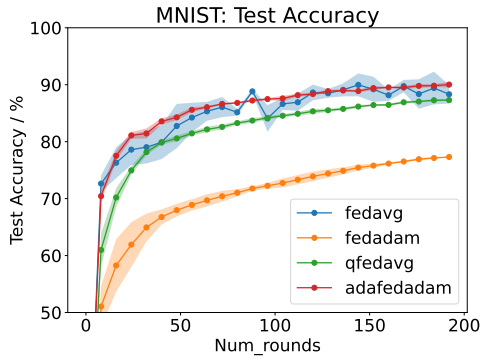
Our Contribution

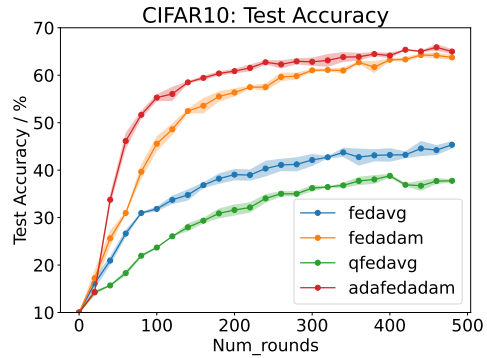
- Formulate fairness-controlled federated learning
- Provide the theoretical fairness guarantee for the solution of the reformulated problem
- Analyse the convergence of `Federated Adam`
- Propose `Adaptive Federated Adam` to optimize the problem with better convergence

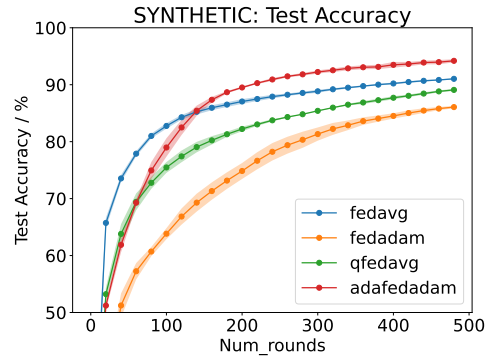
Analysis of FedAdam

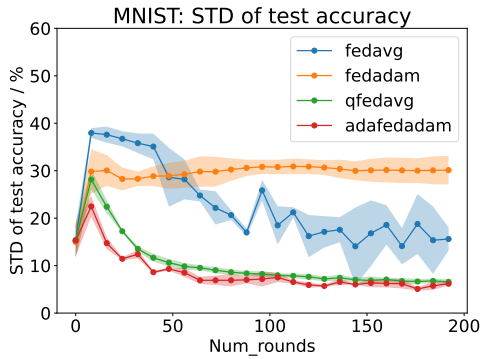


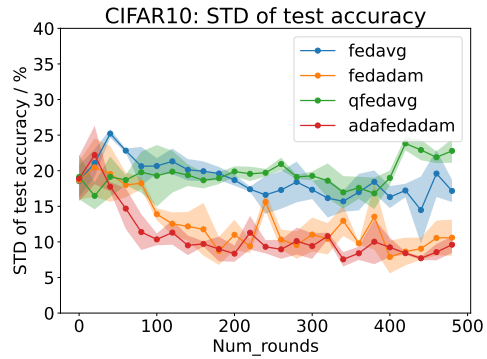
Convergence & Fairness

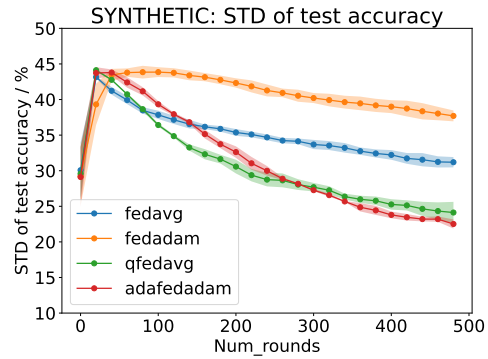


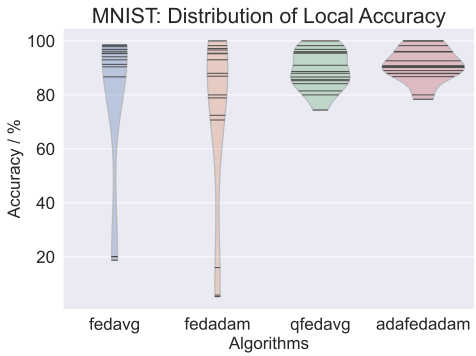


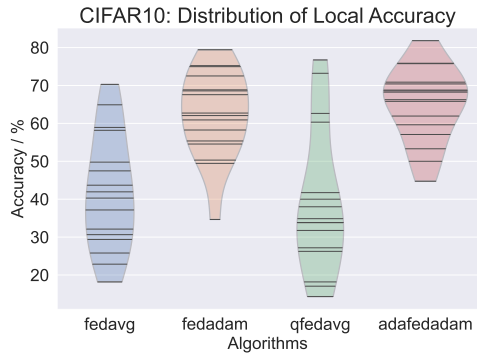


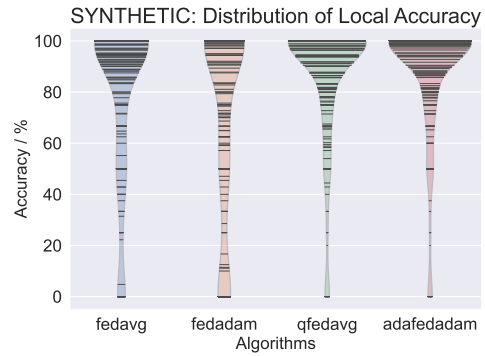






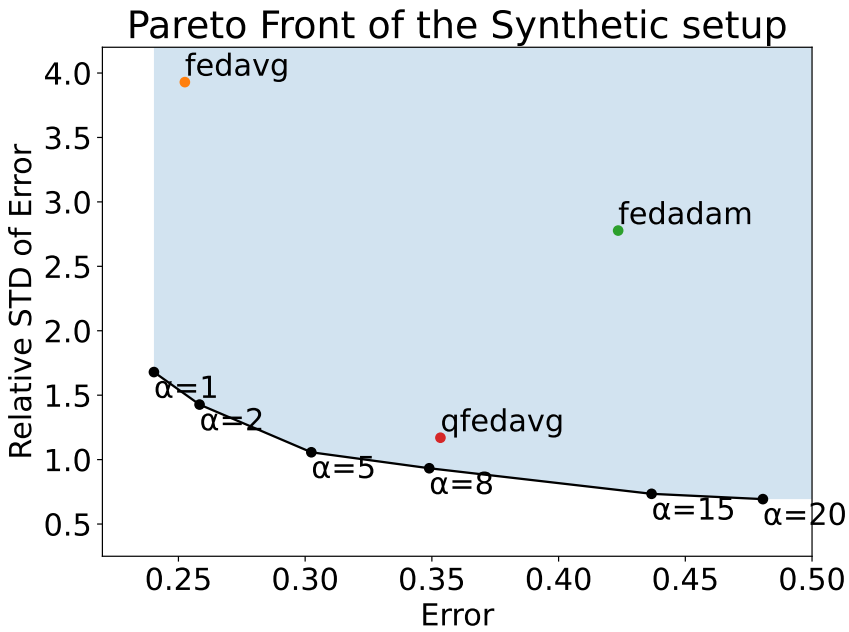






Optimality & Robustness

Pareto Optimality



Pareto Front of the Synthetic setup formed by `AdaFedAdam` with different α.

Robustness against partial participation & arbitrary numbers of local steps.

Thank you!

Questions?