FEDADMM-INSA: AN INEXACT AND SELF-ADAPTIVE ADMM FOR FEDERATED LEARNING

Yongcun Song
Department of Mathematics
City University of Hong Kong

Ziqi Wang
Chair for Dynamics, Control, Machine Learning and Numerics
AvH Professorship, Department of Mathematics
FAU Erlangen-Nürnberg

Enrique Zuazua
Chair for Dynamics, Control, Machine Learning and Numerics
AvH Professorship, Department of Mathematics
FAU Erlangen-Nürnberg

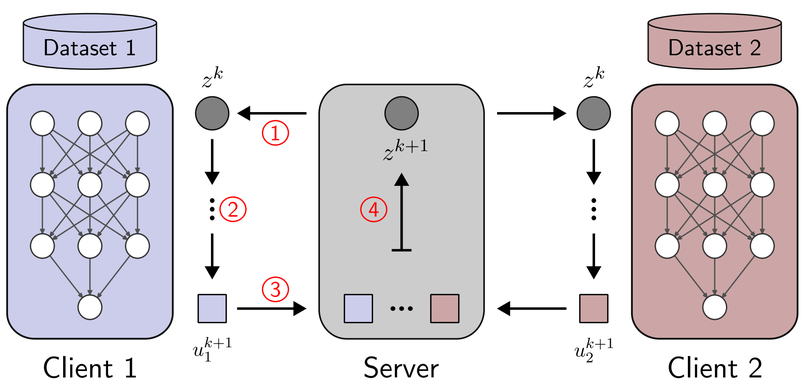
Introduction

Federated learning (FL) facilitates collaborative artificial intelligence (AI) model development among clients without sharing their data [2]. For example, hospitals can collaboratively develop AI tools for disease diagnosis without compromising patient confidentiality.

- **Problem formulation:** The goal of FL is to minimize the following problem:

$$\min_{z\in\mathbb{R}^n} \sum_{i=1}^{m} \alpha_i \ell_i(z), \tag{P1}$$

  where ℓ_i is each client's local loss function, and α_i is the weight.
- **Training paradigm:** The FL training consists of several communication rounds among multiple clients and a central server. Each round contains four steps:
  1. Server broadcasts global model parameters to the clients;
  2. Clients conduct local training/update in parallel;
  3. Clients send updated local model parameters back to the server;
  4. Server aggregates local model parameters to update global ones.



Challenges

The distributed model training process of FL imposes fundamental challenges including:

1. **Model performance:** The effectiveness of aggregated models in comparison to those trained centrally remains uncertain.
2. **Data heterogeneity:** Clients normally possess unbalanced and non-IID local datasets, impacting the performance of the trained model.
3. **Communication cost:** Delays or failures in clients' communication can disrupt the entire training process.

FedADMM-InSa algorithm design

- By using a consensus constraint, we first rewrite (P1) into the following form:

$$\min_{u_i,z\in\mathbb{R}^n} \sum_{i=1}^{m} \alpha_i \ell_i(u_i), \text{ s.t. } u_i = z, \forall\, i \in [m], \tag{P2}$$

  where u_i is client's local model parameter and z is server's global model parameter. For (P2), its augmented Lagrangian reads:

$$L(u,\lambda,z) = \sum_{i=1}^{m} \alpha_i \underbrace{\left[\overbrace{\ell_i(u_i)}^{\text{old loss}} - \overbrace{\lambda_i^\top (u_i - z)}^{\text{Lagrange term}} + \overbrace{\frac{\beta_i}{2}\|u_i - z\|^2}^{\text{penalty term}} \right]}_{\text{Each client's new loss } L_i(u_i,\lambda_i,z)}.$$

- Then, the vanilla FedADMM iteratively updates the variables $\{u, \lambda, z\}$ as follows:
  - Client's local update

$$u_i^{k+1} = \arg\min_{u_i\in\mathbb{R}^n} L_i\left(u_i, \lambda_i^k, z^k\right), \tag{S1}$$

$$\lambda_i^{k+1} = \lambda_i^k - \beta_i\left(u_i^{k+1} - z^k\right). \tag{S2}$$

  - Server's aggregation

$$z^{k+1} = \arg\min_{z\in\mathbb{R}^n} L\left(u^{k+1}, \lambda^{k+1}, z\right). \tag{S3}$$

- For FedADMM-**In**, we propose an **inexactness** criterion for solving (S1):
  - We first define residual $e_i^k(u_i) = \nabla_{u_i} L_i(u_i, \lambda_i^k, z^k)$ for each client i.
  - Then, each client i finds u_i^{k+1} such that

$$\|e_i^k(u_i^{k+1})\| \leqslant \sigma_i \|e_i^k(u_i^k)\|,$$

    where σ_i is a given constant satisfying

$$0 < \sigma_i < \frac{\sqrt{2}}{\sqrt{2} + \sqrt{\tilde{\beta}_i}} < 1,$$

    with $\tilde{\beta}_i = \beta_i / c_i$, and c_i is a constant.

FedADMM-InSa algorithm design – Cont'd

- For FedADMM-In**Sa**: we further design a **self-adaptive** penalty parameter scheme.
  - We define the primal residual p_i^k and the dual residual d_i^k as follows:

$$p_i^k = \beta_i^k \|u_i^{k+1} - u_i^k\|,$$
$$d_i^k = \|u_i^{k+1} - z^k\|.$$

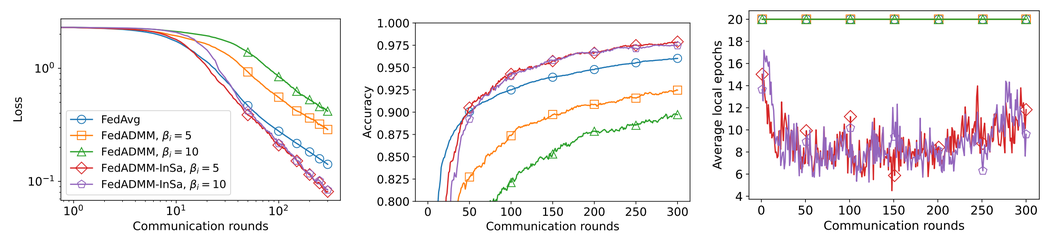
  - Then, each client updates β_i^k according to the following scheme:

$$\beta_i^{k+1} = \begin{cases} \beta_i^k \tau, & \text{if } d_i^k > \mu p_i^k, \\ \beta_i^k / \tau, & \text{if } p_i^k > \mu d_i^k, \\ \beta_i^k, & \text{otherwise,} \end{cases}$$

    where $\mu, \tau > 1$, e.g. $\mu = 5$ and $\tau = 2$.

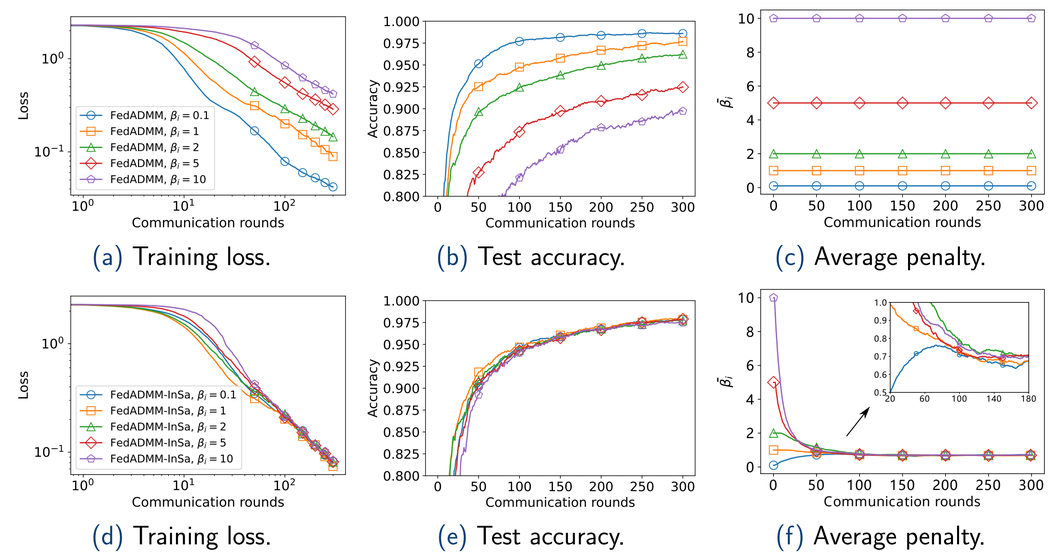
Experimental results

- We test image classification using convolutional neural networks (CNN) with the MNIST dataset. The MNIST dataset contains images of handwritten digits and the CNN architecture used is the same as that in [2].



(a) Training loss. (b) Test accuracy. (c) Average local epochs.

- The pre-selected penalty parameter β_i plays a crucial role in the performance of the FedADMM algorithm. Below we present the comparison results of FedADMM and FedADMM-InSa under different penalty parameters.



(a) Training loss. (b) Test accuracy. (c) Average penalty.

(d) Training loss. (e) Test accuracy. (f) Average penalty.

FedADMM (top row) vs. FedADMM-InSa (bottom row).

Conclusions and outlook

In this work, we introduce the FedADMM-InSa algorithm, a novel approach that leverages the alternating direction method of multipliers (ADMM) to address the challenges of federated learning (FL) in the presence of data and system heterogeneity. Note that here we assume each client is willing to cooperate with the server. In another work [1], we explore an FL scenario where clients' training efforts are shaped by the server's incentives and their training costs. Additionally, while FL offers privacy benefits by sharing model parameters instead of raw data, it remains vulnerable to data reconstruction attacks. For instance, we propose in [3] a weighted attack method for reconstructing private data in multiple-step local update scenarios, emphasizing the importance of developing privacy defenses in FL.

References

[1] Liu, K., Wang, Z., & Zuazua, E. (2024). Game Theory in Federated Learning: A Potential Game Perspective, submitted, under review.
[2] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics. PMLR.
[3] Song, Y., Wang, Z., & Zuazua, E. (2023). Approximate and Weighted Data Reconstruction Attack in Federated Learning, submitted, under review.
[4] Song, Y., Wang, Z., & Zuazua, E. (2024). FedADMM-InSa: An Inexact and Self-Adaptive ADMM for Federated Learning, submitted, under review.

MODCONFLEX
Modelling and control of flexible structures interacting with fluids
Funded by the European Union
Unterstützt von / Supported by
Alexander von Humboldt Stiftung/Foundation