Federated Learning

Project Management and Software Development for Medical Applications

General Info
Contact Person: Rami Eisawy
Contact Email: rami@deepc.ai

Project Abstract
Federated learning is quickly becoming a key area of interest for its ability to potentially tackle the issue of training a global machine learning model without the need for clients to share data externally. A common challenge in this distributed learning setting is 1) how do you effectively combine the client models into a global one (aggregation scheme) and 2) the heterogeneity of the data distribution (non-iid) among the clients negatively affects the performance of the server-side model. This project aims to create a federated learner setting, benchmark the performance of two recent aggregation schemes on an internal model and (time and student progress dependent) develop a new aggregation scheme that outperforms current works.

Background and Motivation
A single algorithm/model has the potential to be deployed to multiple sites either via site-specific calibration and/or domain adaptation. This provides a favorable environment for clients in terms of performance, privacy and data ownership. However, there is the possibility to leverage information across all installations in the form of distributed training. In this setting, we can consider these various site installations as edge devices. The challenge remains in producing a global (server) model that is robust and efficient in terms of its aggregation scheme, tackles the heterogeneous nature of the nodes and whether or not the difficulty in data distribution across clients can be addressed at the server level to lead to a more accurate model. The focus of the project will be the aggregation scheme. Recent projects have tackled this by simply using an average function (FedAvg), personalisation layers (FedPer) and matched averaging (FedMA) to name a few. An internal CT brain model will be provided and ideally, the (Clara Train SDK) is utilised. Previous experience in another framework that supports custom components/models would also be acceptable and speed up development time.

Student’s Tasks Description
- Simulate a distributed training setting using nvidia’s Clara SDK to create a global model (1W)
- Evaluate performance of internal model in this setting under various aggregation schemes (AS) i.e FedMA, FedAvg (1-2.5W)
- (BONUS) Develop a new AS that is superior in performance vs client side (1.5W)
- Remainder of 0.5W can be used to summarise results/key findings

Students will learn about federated learning and why it is important in the healthcare domain given data privacy concerns, data sharing and performance. They will learn about how in the real world, data distribution may not be iid and thus a federated learner will most likely have poorer performance when compared to a homogeneous client-side model. They will learn about potential solutions to overcome this issue and potentially develop an aggregation scheme that is able to overcome these challenges. Students will get to utilize and learn about the Clara SDK from nvidia.
Technical Prerequisites

- Experience in docker, python
- Prior experience in Clara SDK beneficial
- Prior experience in data analysis
- Sound knowledge about deep learning, linear algebra and probability theory

References

[1] [1602.05629] Communication-Efficient Learning of Deep Networks from Decentralized Data (arxiv.org)

[2] [1912.00818] Federated Learning with Personalization Layers (arxiv.org)
