Investigating Data Distribution Variability Across Devices in Federated Learning: Comparative Analysis of Algorithm Performance

L. Duda, K. Alibabaei, L. Berberi, V. Kozlov, A. Streit
Karlsruhe Institute of Technology, Scientific Computing Center (SCC), Baden-Württemberg, Germany

Abstract: Federated Learning (FL) enables distributed training on multiple devices, enhancing privacy and conserving resources by sharing model updates instead of data. Using NVFlare, we distributed the training of a CNN for brain tumor detection, keeping sensitive data local. We evaluated different FL algorithms (FedAvg, FedOpt, FedProx, Scaffold) and found that complex algorithms like Scaffold perform better among heterogeneous data distributions.

Introduction

Traditional Machine Learning Approach:

- Train a model such that it recognizes a pattern or behavior
- Labeled data (supervised learning) or unlabeled data (unsupervised learning) needed
- Data is centralized in one spot

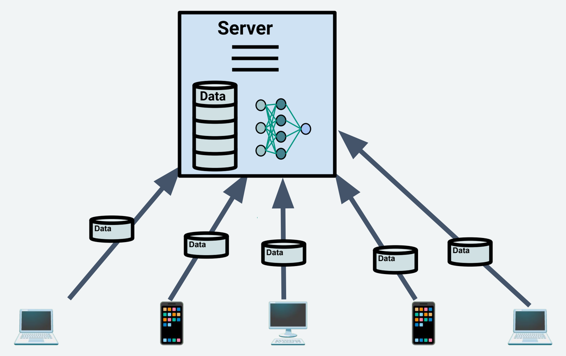


Figure: Centralized Data Storage for Machine Learning Training

Problem:

What if data can not be shared and allocated in one spot due to privacy, data restrictions or resource limitations?

Federated Learning:

An approach enabling multiple peers to collaboratively learn a shared prediction model by sharing the weights of the model but not the data itself.

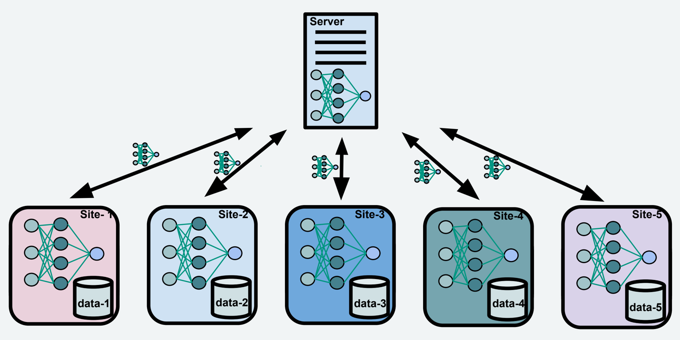


Figure: Scatter and Gather workflow in Federated Learning

Research Objective

Research Statement:

Traditional distributed learning assumes:

- Data on different nodes (computing units) originate from the same distribution.
- Data on different nodes have a similar size.

Real-world scenarios often feature:

- Significant data imbalances.
- Size differences in data across nodes.

Federated Learning method:

Tailored to handle unbalanced and non-identically distributed (non-IID) data using different **Aggregation Algorithms**.

Key Algorithms in FL:

- **FedAvg:**
  - Collects and aggregates local weights using a weighted average after local training.
- **FedProx:**
  - Adds a loss function to penalize local weights deviating from the global model.
- **FedOpt:**
  - Allows use of a specified optimizer and learning rate scheduler (e.g., SGD) to aggregate model weights.
- **Scaffold:**
  - Adds a correction term to neural network parameters during local training by calculating the discrepancy between global parameters.

OBJECTIVE:

Comparison of Federated Learning and Centralized Learning Methods. Moreover, evaluate different data distributions among the devices in combination with different FL Algorithms: FedAvg, FedOpt, FedProx and Scaffold.

DL Methodology

Dataset:

- Splitting data into 80% training data and 20% test data

Model:

- A simple convolutional neural network with 5 layers and 3687745 parameters using Pytorch Framework

Hyperparameters:

- Batch size of 32
- 15 epochs of training
- Learning rate of 0.001

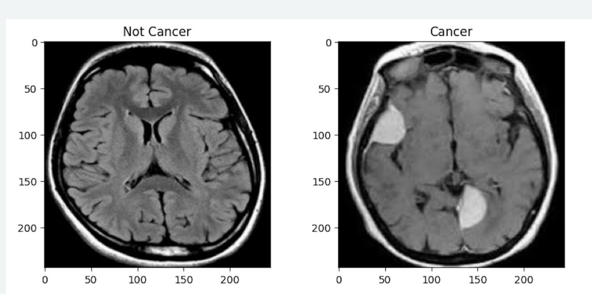


Figure: x-ray images of the human brain
source: https://www.kaggle.com/

FL Methodology

FL Framework:

NVIDIA FLARE from NVIDIA

- Open-source library for federated learning
- Allows adapting an existing machine learning workflow to a federated paradigm

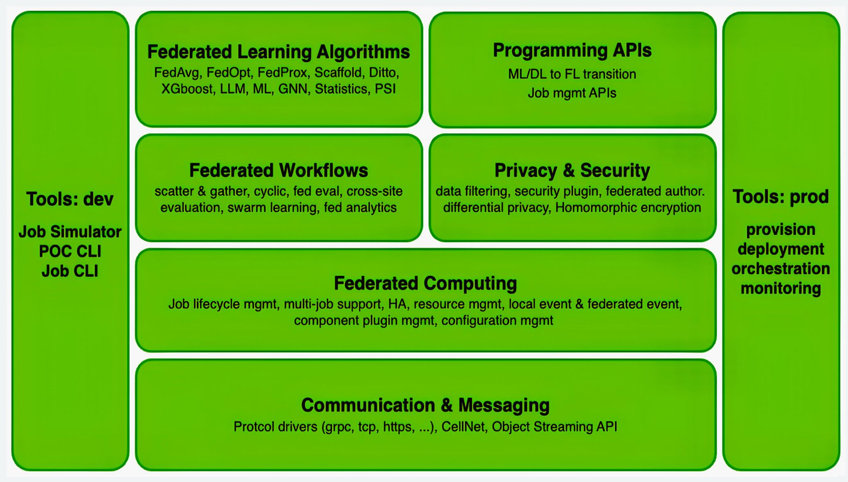


Figure: High-level System Architecture of NVFlare, source: [3]

Data splitting methods For FL:

Various splitting methods were used to simulate balanced and imbalanced data distributions among the sites.

- **Uniform**: No imbalance, each site has the same amount of data.
- **Square**: The amount of data is correlated with the site-ID in a squared fashion (1^2 to ID^2)
- **Exponential**: The amount of data is correlated with the site-ID in an exponential fashion (exp(1) to exp(ID))

Hyperparameters for experiments:

- Distribution of data per site varies from 5% to 50% (of the training data)
- Batch size of 32, 15 epochs, Learning rate of 0.001
- 5 rounds of training (Receive model, train locally and share updates)
- 5, 7 Clients

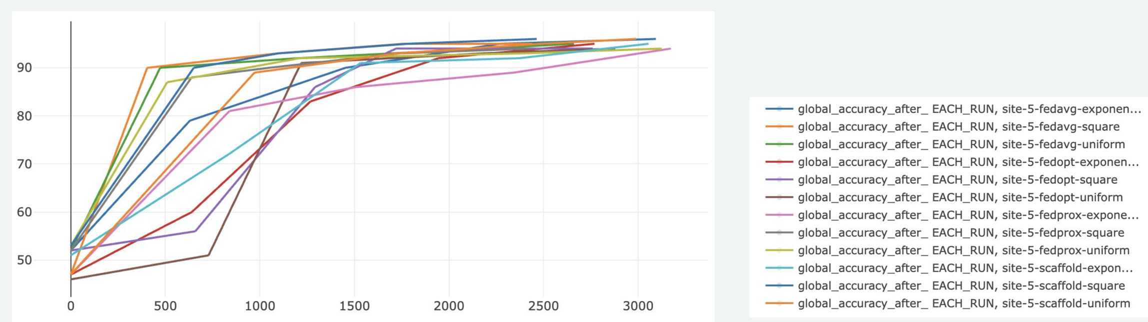
Experiment Tracking:

- Usage of the MLflow instance provided by AI4EOSC[3] Project
- Tracking of loss and accuracy for training and validation and other parameters

Results

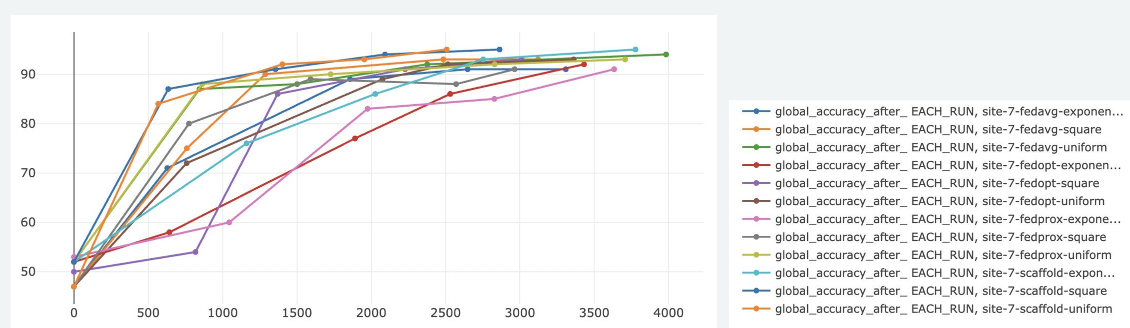
Global Model Performance for 5 clients

algorithm	Global accuracy		
	Uniform	Square	Exponential
FedAvg	95	95	96
FedOpt	94	94	95
FedProx	94	94	94
Scaffold	96	96	95
Centralized	97		



Global Model Performance for 7 clients

algorithm	Global accuracy		
	Uniform	Square	Exponential
FedAvg	94	93	91
FedOpt	93	93	92
FedProx	93	91	91
Scaffold	95	95	95
Centralized	97		



Discussion:

- The federated version of the machine learning workflow can keep up with the base line depending on the number of clients in combination with the right algorithm and data distribution
- Accuracy with FedAvg gets worse the more unequal the distribution of the data
- Scaffold performs better for heterogeneous data
- For a small count of clients the algorithms had just a marginal difference

Outlooks

- Compare different FL libraries and evaluate them (privacy, platform compatibility, versatility, …)
- Investigating the FL methods in more complex task such as object detection and segmentation.
- Test the shown use case on real world sites and compare the results to the simulation (speed, overhead, accuracy, …)
- Further investigation into different FL algorithms and development of new approaches

References

[1] Roth, H. R., et al. (2022). NVIDIA FLARE: Federated Learning from Simulation to Real-World. *arXiv*. https://arxiv.org/abs/2210.13291
[2] NVFlare GitHub Repository: https://github.com/NVIDIA/NVFlare
[3] AI4EOSC Documentation https://docs.ai4os.eu/en/latest/user/howto/mlops/mlflow.html
[4] H. Brendan McMahan, et al. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv*. https://arxiv.org/abs/1602.05629
[5] Tian Li, Anit Kumar Sahu, et al. (2020). Federated Optimization in Heterogeneous Networks. *arXiv*. https://arxiv.org/abs/1812.06127
[6] Sashank Reddi, et al. (2021). Adaptive Federated Optimization. *arXiv*. https://arxiv.org/abs/2003.00295
[7] Sai Praneeth Karimireddy, et al. (2021). SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. *arXiv*. https://arxiv.org/abs/1910.06378
[8] GitHub Repository of this work: https://github.com/LeoDuda/Practical-Introduction-into-Federated-Learning-with-NVFlare

Funded by the European Union

AI4EOSC receives funding from the European Union's Horizon Europe 2022 research and innovation programme under agreement *101058593*