Communication-Efficient Learning of Deep Networks from Decentralized Data

Idriss Tondji

African Master in Machine Intelligence, AMMI-Senegal

Bootcamp-Week2

August 27, 2021

Overview

Overview

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.
- But in the real world data is often decentralized across many parties.

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.
- But in the real world data is often decentralized across many parties.
- **Why can't we just centalize the data?**

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.
- But in the real world data is often decentralized across many parties.
- **Why can't we just centalize the data?**
- Sending the data may be **too costly** (will require a large amount of space in the data center), **too sensitive** (privacy issue).

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.
- But in the real world data is often decentralized across many parties.
- **Why can't we just centalize the data?**
- Sending the data may be **too costly** (will require a large amount of space in the data center), **too sensitive** (privacy issue).
- **How about each party learn on its own ?** .

Introduction

Motivation/Problem

- The standard setting in ML considers a centralized dataset.
- But in the real world data is often decentralized across many parties.
- **Why can't we just centalize the data?**
- Sending the data may be **too costly** (will require a large amount of space in the data center), **too sensitive** (privacy issue).
- **How about each party learn on its own ?** .
- The local dataset may be too small (Non-statistically significant results).
- The local dataset may be biased (not representative of the target distribution).

Motivation/Problem

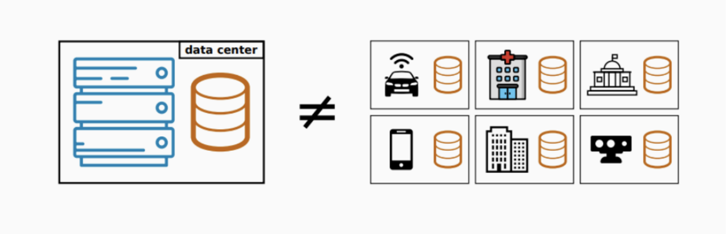


Figure 1: Illustration of centralized/decentralized data.

Overview

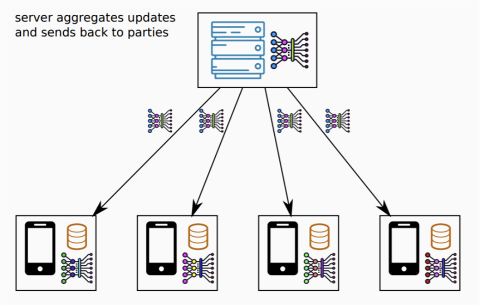
Introduction

Goal

- Collaboratively train a ML model while keeping the data decentralized (keeping data private and secure).

Introduction
Goal

- Collaboratively train a ML model while keeping the data decentralized (keeping data private and secure).



-

Figure 2: Architecture of the method.

Overview

1. Introduction
   - Motivation/Problem
   - Goal
2. Approach
   - Federated Learning
   - Federated Optimization
3. Experiments
4. Results
5. Conclusion

Approach

We label this decentralized learning approach mentioned above as **Federated Learning**.

The name Federated Learning comes from the fact that the learning task is solved by a loose federation of participating devices. We train several local update on different devices that contain data and then we aggregate them. We are able to do this with the help of a new algorithm called **FederatedAveraging (FedAvg)**.

Approach

The main idea of Federated Learning (FL) is the concept of keeping user data private. Even if the data in the data centers is "anonymized", it can still put the users at risk via join with other data.

In FL, the information transmitted is the minimal update necessary to improve a particular model. The updates themselves don't contain extra information than what is actually required.

Overview

1. Introduction
   - Motivation/Problem
   - Goal
2. Approach
   - Federated Learning
   - Federated Optimization
3. Experiments
4. Results
5. Conclusion

Federated Optimization

Optimization problem in FL is referred to as **Federated Optimization**.

There are several key properties in Federated Optimization. For this talk, we will be focusing on two:

- **Non-IID**: Any particular user's local dataset will note be representative of the population. distribution.
- **Unbalanced:** Some users will make much heavier use of the service or app than others, leading to varying amounts of local training data.

Algorithm

FederatedAveraging (FedAvg).

Algorithm 1 FedAvg (server-side)

initialize x_0
for each round $t = 1, \dots, T$ **do**
  $M \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of M clients)
  for each client $k \in S_t$ **in parallel do**
    $x^k_{t+1} \leftarrow ClientUpdate(k, x_t)$
  end for
  $x_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} x^k_{t+1}$
end for

Algorithm 2 ClientUpdate(k,x)

// Run on client k
for local step $j = 1, \dots, L$ **do**
  $z \leftarrow$ mini-batch of $|\mathcal{P}_k|$ examples from $\mathcal{P}_k$
  $x \leftarrow x - \eta \nabla f(x; z)$
end for
send x to the server

K	Number of clients
C	Global batch size, $C \leq 1$
M	Clients per round
T	Total communication rounds
L	Local steps per round

Figure 3: FL Algorithm.

- For $L = 1$ and $C = 1$, it is equivalent to classic parallel SGD.
- For $L > 1$: each client performs multiple local SGD steps before communicating.

Experiments

Dataset

MNIST Handwritten Digit Classification (1 channel).

Experiments

Dataset

MNIST Handwritten Digit Classification (1 channel).

Model

A simple multilayer perceptron with 2 hidden layers with 200 units each using ReLU activations.

Experiments

Dataset

MNIST Handwritten Digit Classification (1 channel).

Model

A simple multilayer perceptron with 2 hidden layers with 200 units each using ReLU activations.

Data Partitioning

- **IID**: data is shuffled, partitioned into 100 clients. Each client receives 600 examples.
- **Non-IID**: Sort the data by digit label, divide it into 200 shards. Each shard is of size 300. Assign each of 100 clients 2 shards. .

Hyperparameters

- Learning Rate: 0.01
- Local Epochs: 5
- Local Batch Size: 10
- Client Fraction: 0.1
- Optimizer: SGD

Results

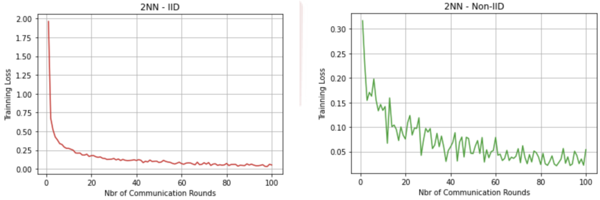
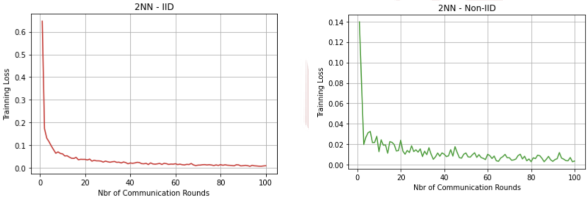


Figure 4: Learning curve for E=1 (up) and E=5 (down).



AIMS African Institute for Mathematical Sciences SENEGAL

Results

Personal experiments

- MNIST 2NN with E=1.
  - IID Parition: (Test Accuracy 97.72%)
  - Non-IID ParTition: (Test Accuracy 95.82%)
- MNIST 2NN with E= 5.
  - IID Parition: (Test Accuracy 97.97%)
  - Non-IID Parition: (Test Accuracy 95.31%)

Conclusion

- Federated Learning can be practical with the new defined algorithm Federated Averaging.
- FL is able to train high-quality models using relatively few rounds of communication.

Conclusion

- Federated Learning can be practical with the new defined algorithm Federated Averaging.
- FL is able to train high-quality models using relatively few rounds of communication.

We can extend this work by doing more experiments with differents hyperparameters and see their behavior and convergence rate.

References

H. Brendan McMahan and al.
Communication-Efficient Learning of Deep Networks from Decentralized Data.
2017.

Acknowledgements

Thanks for your attention!