Deployment of Federated Learning Infrastructure

Mohamed Hemdan
Supervised by Jose Holgueras and Sofia Vallecorsa

Openlab summer student program 2022

14th of September 2022
What is Federated Learning?

Standard Machine Learning Setting

Next Word Prediction Model

Mohamed Hemdan - Deployment of Federated Learning Infrastructure
What is Federated Learning?

**Standard Machine Learning Setting**

**Federated Learning Setting**

Machine 1

Machine 2

Machine 3

Machine 4
Objective

My Federated Learning Tool

Build an easy-to-use tool that enable creating flexible deployment of federated learning models
Help researchers to try out different scenarios and even new applications without the need to be expert in federated learning

Desired Characteristics:

Environment Independent
- Work in any environment
- No dependency problems

Ease to Deploy
- Don’t care about how the communication is done. Just deploy

Scalable
- Run on any number of clients, it’s not time-consuming

Adaptable
- Easy to redeploy in case of model changes
Project Solution

Implementation Design

Environment-Independent
We ship the client and server in **docker containers** to the desired machines
So, no dependency problems

Scalable and Adaptable
We used Ansible as a **configuration tool** to do the orchestration of client and server containers. This makes the tool scalable to the number of clients and still adaptable to any changes in model or data

Ease to deploy
We ship the tool in a **python package** with simple APIs to use. Allowing researchers and non-expert users to use federating learning
Project Solution

Code Example (easy-to-use)

Server Code (.server.py)

```python
scontrol = ServerController()
scontrol.set_model(model)
scontrol.set_data(X, y)
scontrol.set_ips(server_ip, monitor_ip)
scontrol.start(num_rounds, evaluate_func)
```

Main Code (.main.py)

```python
runner = ContainerRunner()
runner.add_client(client_ip, data_dir)
runner.server_config(server_ip, data_dir)
runner.run_server('./server.py')
runner.run_client('./client.py')
```

Client Code (.client.py)

```python
client_tr = ClientTrainer()
client_tr.set_data(x_train, x_test, y_train, y_test)
client_tr.set_model(model)
client_tr.set_ips(server_ip, monitor_ip, client_id)
client_tr.train(train_func)
```

Just Run the Code. That's it!

```bash
$ python main.py
```
High Energy Physics (HEP) provides a very valid example for federated learning, due to:

- Huge amount of generated data
  - making it distributed by nature
- Difficulty to collect the data into one place

Federated Learning would be a good solution
Application

**HEP: Simulation of Particle Transport through detectors**

- Particle Transport Simulation using Monte Carlo is very *computationally expensive*.
- To Solve the problem, Previous Research* uses GANs to simulate Monte Carlo:
  - Since Monte Carlo is usually done on a cluster making the data distributed across nodes, GANs are trained on a specific portion of the data.
  - This separation of the training data and the training generated data creates a limiting factor to the amount of data the centralized node can use for training.
- Federated Learning could provide a solution for this:
  - You can train your model on these distributed data.

---

Application

**HEP: Simulation of Particle Transport through detectors**

- We run the model on 3 clients and one server and compared it
- Results (loss vs. time)
Conclusion

And Final remarks …

• Not Every ML can be run on a single machine and there is the need sometimes to use federated learning
• My Federated learning tool can provide an easy-to-use tool for researchers with no expertise in FL to do FL
• The field of High Energy Physics is a very potential application to federated learning
• Federated learning proves to be a good application (specifically the Simulation of Particle Transport through detectors)
QUESTIONS?

Mohamed Hemdan
mhemdan@cern.ch
mashrafhemdan@aucegypt.edu