FEDn Framework Tutorial for IADS Summer School 2022

Author1: Salman Toor
Organization: Uppsala University, Sweden
Email: Salman.Toor@it.uu.se

Author2: M AL-Naday
Organization:
Email: mfhaln@essex.ac.uk

Introduction

The aim of this tutorial is to gain technical experience with the FEDn Framework jointly designed and developed by researchers at Uppsala University and Scaleout Systems team. FEDn provides a complete solution for federated learning based on the Scaleout-led open source federated learning runtime FEDn. The solution handles geographically distributed environments, both for cross-device and cross-silo settings.

FEDn is the framework for federated machine learning model training. It is a modular, open-source ML-framework agnostic environment. The following Github links provide the latest version of the framework together with deployment instructions:

https://scaleoutsystems.github.io/fedn/#/
https://github.com/scaleoutsystems/test-examples
FEDn article: https://arxiv.org/pdf/2103.00148.pdf
FEDn video: https://www.youtube.com/watch?v=TJnkN4JyrRM

In the next 3 hours, we will accomplish the following three tasks:

Task 0 - All-in-one deployment (Only if you have a laptop or VM with Ubuntu installed on it)
Task 1 - Explore the FEDn interface
Task 2 - Train an MNIST model using two clients
Task 3 - Setup and connect a client to UEssex FEDn deployment
Task 4(Optional) - Prepare a new compute package with your own custom dataset

Prerequisites[Mays]

- Setup Virtual Box
- Setup the Virtual Machine

Task-0: All-in-one installation

Requirements

Operating System: Ubuntu 22.04 (recommended)
Memory: 4GB (recommended)
Disk space for FEDn: 5GB (recommended)

This deployment will include the base services and two clients. Please follow the quick start guide available at the link: https://github.com/scaleoutsystems/fedn

In case you find difficulties installing Python 3.8 or 3.9, Docker or Docker compose, the following steps will help you install the setup:

Steps to install python3.8

```
# sudo bash
# apt install software-properties-common
# add-apt-repository ppa:deadsnakes/ppa
# apt install python3.8
# update-alternatives --install /usr/bin/python python
/usr/bin/python3.10 1
# update-alternatives --install /usr/bin/python python
/usr/bin/python3.8 2
# update-alternatives --config python
```

Note: Confirm that the *python* executable points to python3.8

```
# apt install python3.8-distutils
# curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
# python get-pip.py
```

Note: Confirm that pip is using python3.8

```
 # apt-get install python3.8-venv
```

Steps to install Docker and Docker-Compose for Ubuntu 22.04

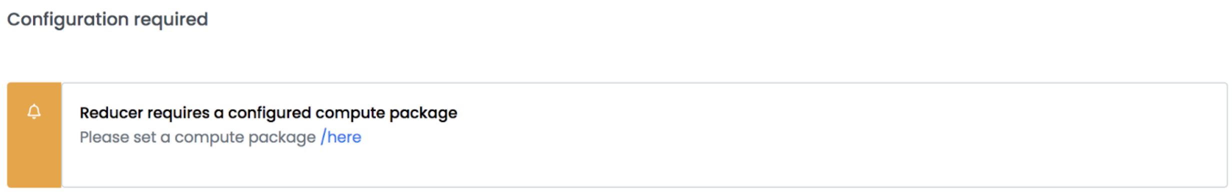
The instructions are available on the following link:
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-22-04

Steps to install Fedn framework

https://github.com/scaleoutsystems/fedn/tree/master/examples/mnist-keras

Task-1: Explore the FEDn interface

1 - Enter the URL: https://localhost:8090



Configuration required

Reducer requires a configured compute package

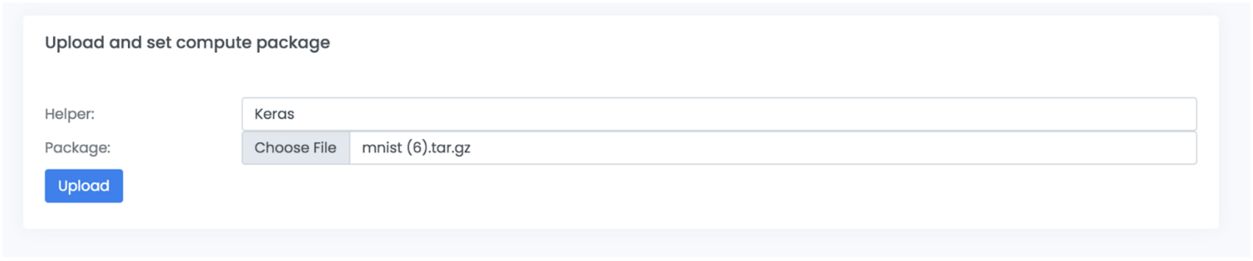
Please set a compute package /here

2 - On this page, we need to upload the **compute package**. The **compute package** contains all the necessary software modules that are required to run the model training process. We need to upload it to the **Reducer**, and it will be sent to all the clients when the training process will start. The **compute package** guarantees that all the clients participating in the training process will run exactly the same code for the training. However, the data will be never shared in this process and the clients will have complete autonomy of their datasets.

3 - For this task, we have prepared a **compute package** based on MNIST machine learning model, which you have generated in Task 0

Upload the archive file “package.tar.gz” to the Reducer.

Note: you need to select Keras in the Helper



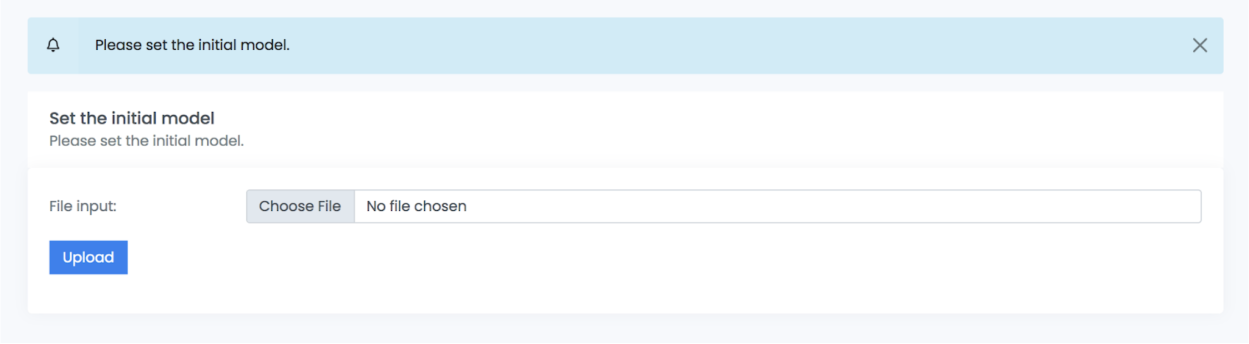
Upload and set compute package

Helper: Keras

Package: mnist (6).tar.gz

FEDn can support different ML-frameworks. The current deployment allows Keras and PyTorch ML-frameworks.

4 - Next step is to upload a **seed model**. This model will be a starting point for generating the global model based on the local models. For this task, we will upload the **MNIST seed model** to the Reducer. we have prepared a **seed model** based on MNIST machine learning model, which you have generated in Task 0:



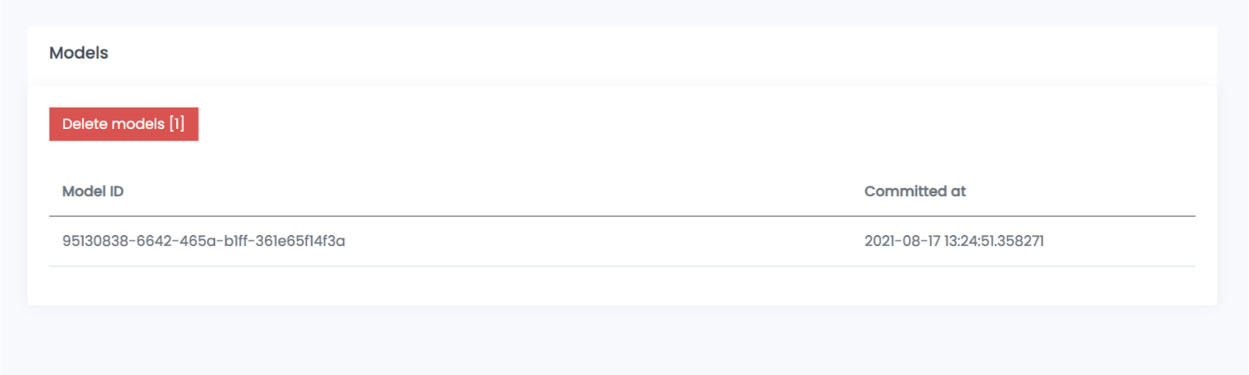
Please set the initial model.

Set the initial model

Please set the initial model.

File input: No file chosen

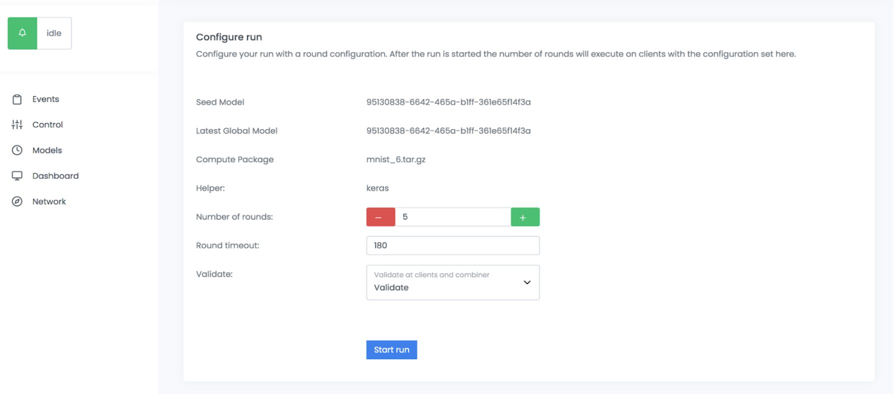
5 - Now we have a seed model available.



Models

Model ID	Committed at
95130838-6642-465a-b1ff-361e65f14f3a	2021-08-17 13:24:51.358271

6 - Now it’s time to click on different tabs and get familiar with the environment. Click on the left tabs and write a few sentences about each associated page.

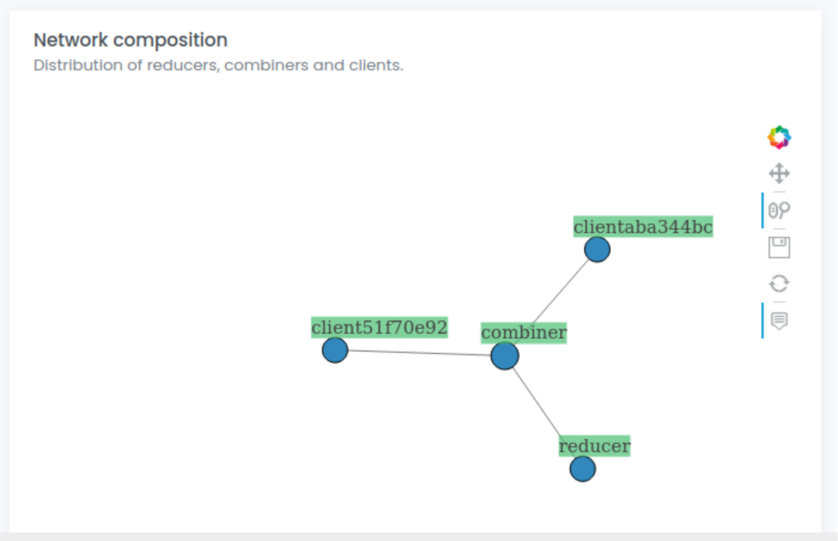


Configure run

Configure your run with a round configuration. After the run is started the number of rounds will execute on clients with the configuration set here.

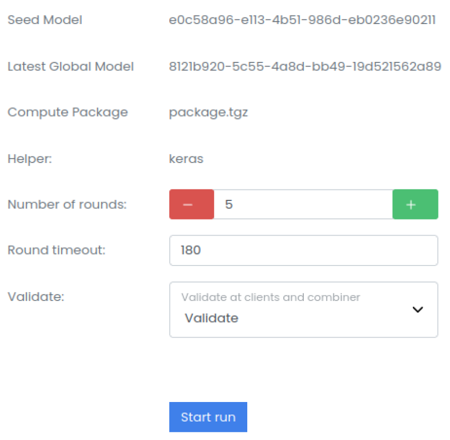
Seed Model	95130838-6642-465a-b1ff-361e65f14f3a
Latest Global Model	95130838-6642-465a-b1ff-361e65f14f3a
Compute Package	mnist_6.tar.gz
Helper:	keras
Number of rounds:	5
Round timeout:	180
Validate:	Validate at clients and combiner Validate

The *Network* tab will show you the deployed services and client connections.


Network composition

Distribution of reducers, combiners and clients.

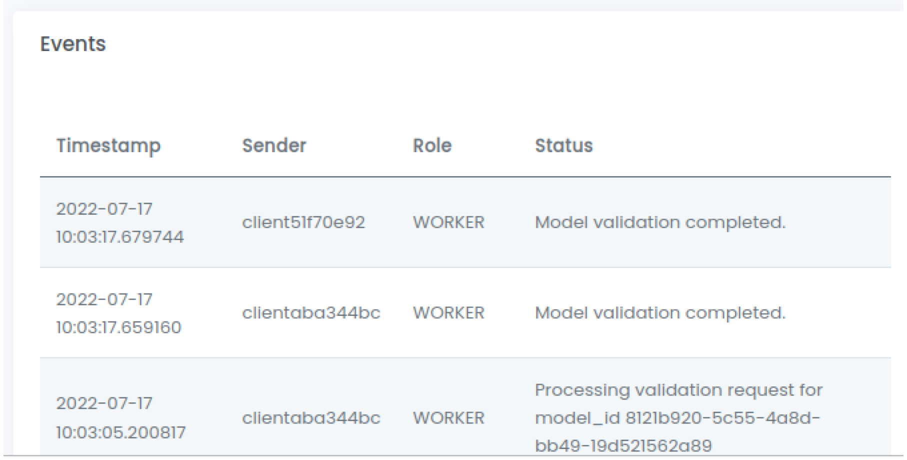
The *Control* tab will set the round max round time and start the training.



Seed Model	e0c58a96-e113-4b51-986d-eb0236e90211
Latest Global Model	8121b920-5c55-4a8d-bb49-19d521562a89
Compute Package	package.tgz
Helper:	keras
Number of rounds:	5
Round timeout:	180
Validate:	Validate at clients and combiner Validate

Task 2: Train an MNIST model using two clients

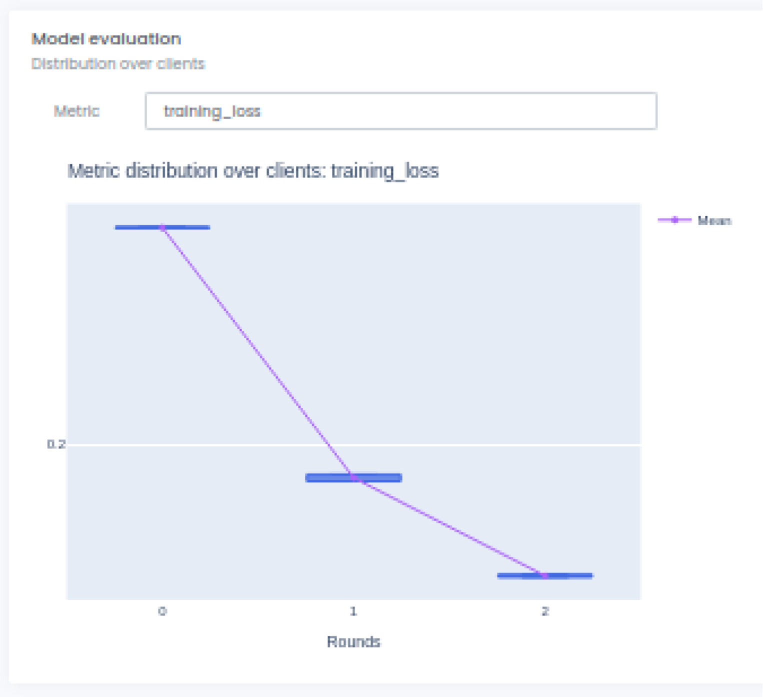
We have a working setup of FEDn framework. In this task, we will train a test model in federated machine learning settings. Press the *start run* button in the *Control* tab and initiate the training process. The process will take some time and you can monitor the process from the *Event* tab.



Events

Timestamp	Sender	Role	Status
2022-07-17 10:03:17.679744	client51f70e92	WORKER	Model validation completed.
2022-07-17 10:03:17.659160	clientaba344bc	WORKER	Model validation completed.
2022-07-17 10:03:05.200817	clientaba344bc	WORKER	Processing validation request for model_id 8121b920-5c55-4a8d-bb49-19d521562a89

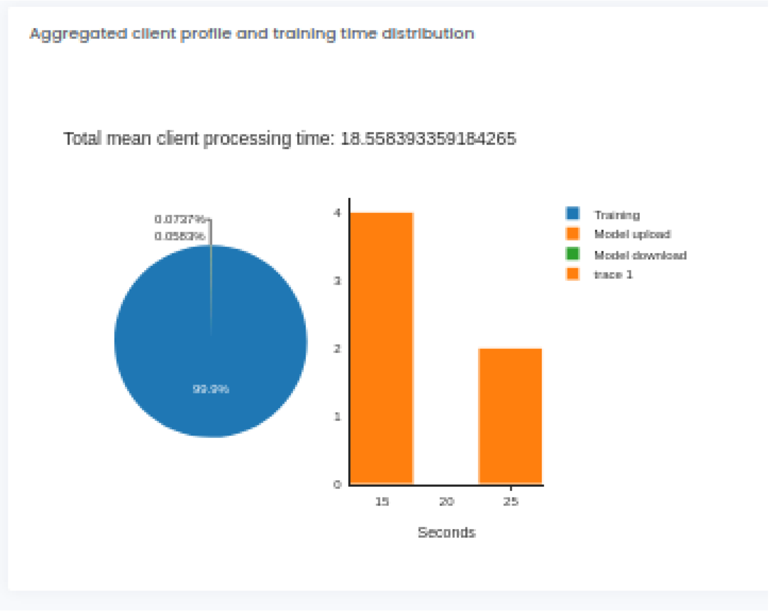
Once all the rounds finish, the final status can be viewed from the *Monitor* tab.



Model evaluation

Distribution over clients

Metric: training_loss



Aggregated client profile and training time distribution

Total mean client processing time: 18.558393359184265

Task 3: Setup and connect a client to to UEssex FEDn deployment

1 - To understand the distributed deployment, please follow the link: https://scaleoutsystems.github.io/fedn/deployment.html

2 - The University of Essex has a working distributed deployment of FEDn framework. You will connect your client to the FEDn framework in the following steps:

- Double click on the ‘iads_tap_udp_7035_XXX
- MAC users (Tunnelblic): click ‘All users’ in the screen
- Select the new configuration named ‘iads_tap_udp_7035_XXX’ in your vpn list of connections and click **‘connect’** in order to connect to NCL VPN and reach FEDn server
- Once connected, verify you can reach FEDn server via the vpn by running:
  - Windows users: open cmd
  - Linux/Mac users: open terminal
  - `Ping 192.168.177.177`

3 - Next step is to setup a new client and joins the UEssex federation.

```
# docker run -v /home/stoor/client.yaml:/app/client.yaml -v
$PWD/data:/var/data --add-host combiner:192.168.177.177 -e
ENTRYPOINT_OPTS=--data_path=/var/data/mnist.npz
ghcr.io/scaleoutsystems/fedn/fedn:develop-mnist-keras run client -in
client.yaml
```

Note: Ask instructors to give you details about `client.yaml` file.

Task 4 (Optional): Prepare a new compute package with your own custom dataset

In this task, you need to prepare a new **compute-package** and run the whole setup again. First, read this explanation of the **compute-package**:
https://scaleoutsystems.github.io/fedn/tutorial.html

You have a **compute-package** from Task0, unpack the archive file, and read the following files:

- train.py
- validate.py
- initial_model.py
- data/read_data.py
- settings.yaml
- fedn.yaml

1- Start by making minimal changes in the MNIST model architecture. For example, change the number of layers or number of neurons in each layer and run the setup. Instructions for how to repackage the **compute-package** can be found here:
https://github.com/scaleoutsystems/fedn/tree/develop/test/mnist-keras

2- Use your own neural network model and prepare a new **compute-package**.