README FILE

ROHAN SINGHVI (IIB2020004)
AKASH BISWAS (IIT2020001)
SHUBHAM (IIT2020007)
ASHISH AGARWAL (IIT2020009)
RAHUL MAHTO (IIT2020022)

Federated Learning using MNIST Data

This is a Python implementation of Federated Learning using MNIST Data. The goal of this project is to showcase how Federated Learning can be applied to the classic MNIST dataset.

About Federated Learning

Federated Learning is a type of machine learning where the training data is decentralized and stored on different devices, and the model is trained locally on each device. The updated model is then sent to a central server where it is aggregated to create a global model. Federated Learning is particularly useful in scenarios where the data cannot be centralized due to privacy concerns, regulatory restrictions, or technical limitations.

About the MNIST Dataset

The MNIST dataset is a collection of handwritten digits used for training image processing systems. It contains 60,000 training images and 10,000 test images. Each image is a 28x28 grayscale image of a digit from 0 to 9.

Requirements

- Python 3.6 or higher
- TensorFlow 2.0 or higher
- NumPy
- matplotlib

Commands

To run the Federated Learning using MNIST Data, you can follow these steps:

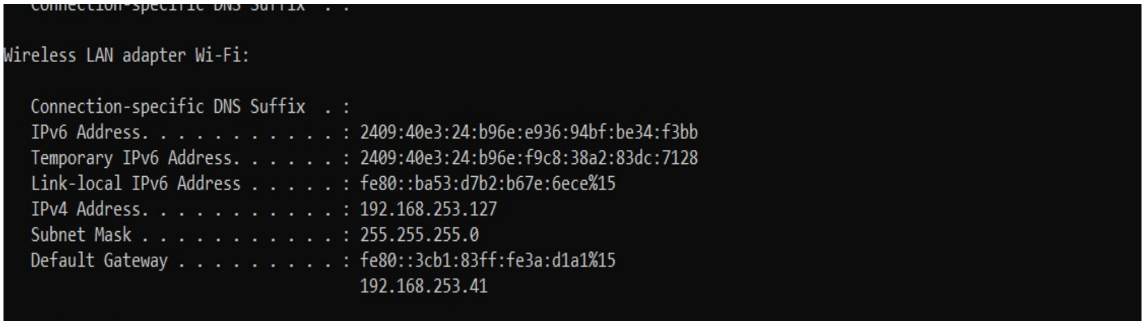
1. Download the Zip and Unzip the file.

2. Install the required dependencies:

Command: pip install -r requirements.txt

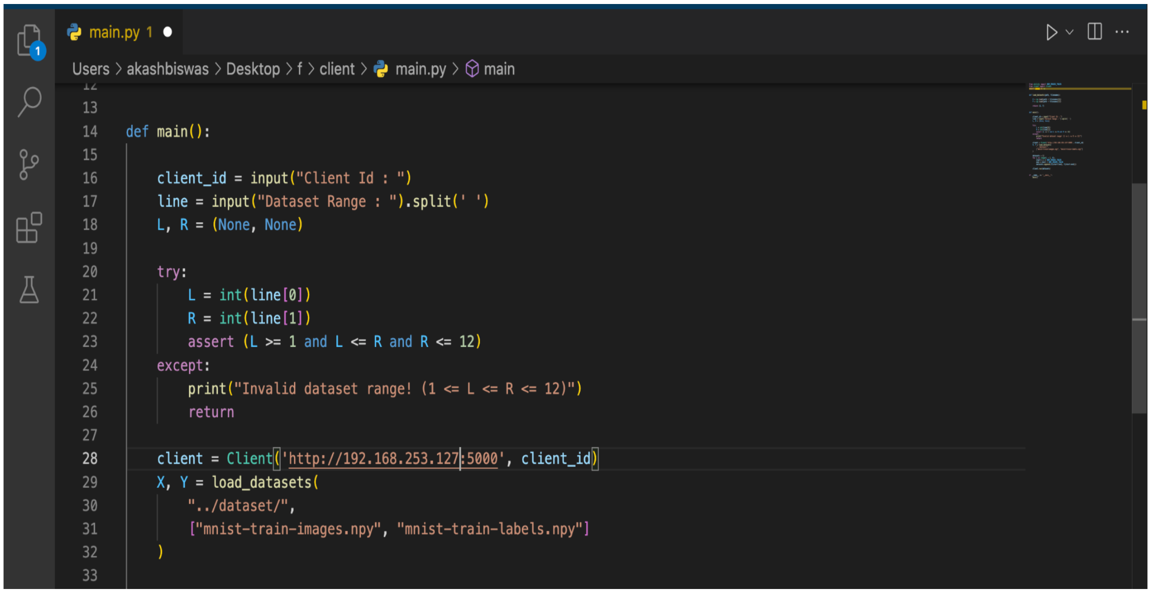
3. Check the IP Address of the Server:

Command: ifconfig



```
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2409:40e3:24:b96e:e936:94bf:be34:f3bb
   Temporary IPv6 Address. . . . . . : 2409:40e3:24:b96e:f9c8:38a2:83dc:7128
   Link-local IPv6 Address . . . . . : fe80::ba53:d7b2:b67e:6ece%15
   IPv4 Address. . . . . . . . . . . : 192.168.253.127
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::3cb1:83ff:fe3a:d1a1%15
                                       192.168.253.41
```

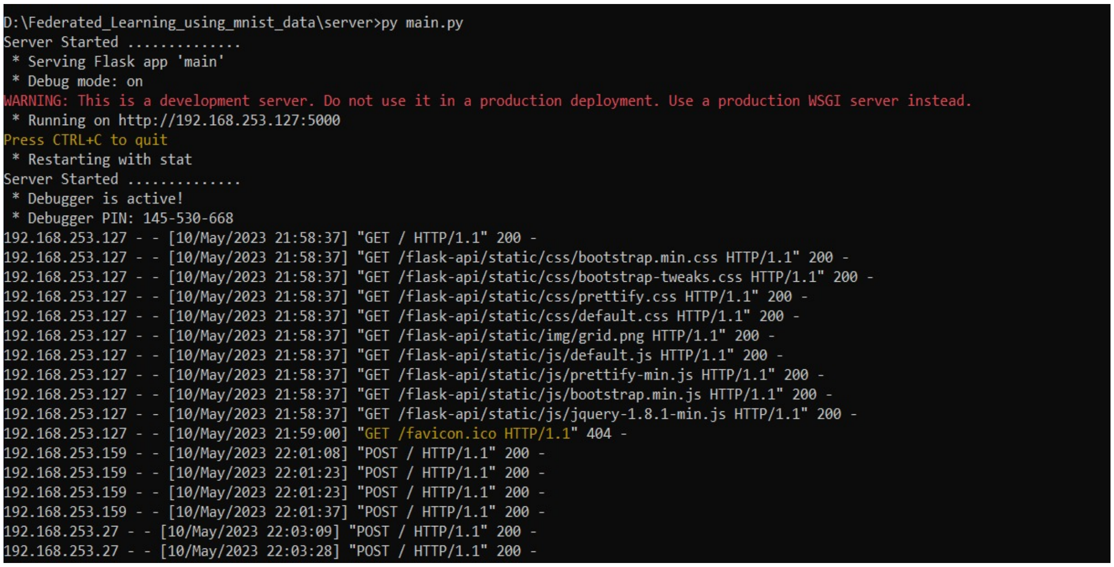
4. Change the IP Address in the Client file of Main.py to create a socket connection:



```python

def main():

    client_id = input("Client Id : ")
    line = input("Dataset Range : ").split(' ')
    L, R = (None, None)

    try:
        L = int(line[0])
        R = int(line[1])
        assert (L >= 1 and L <= R and R <= 12)
    except:
        print("Invalid dataset range! (1 <= L <= R <= 12)")
        return

    client = Client('http://192.168.253.127:5000', client_id)
    X, Y = load_datasets(
        "../dataset/",
        ["mnist-train-images.npy", "mnist-train-labels.npy"]
    )

```

5. Start the server:

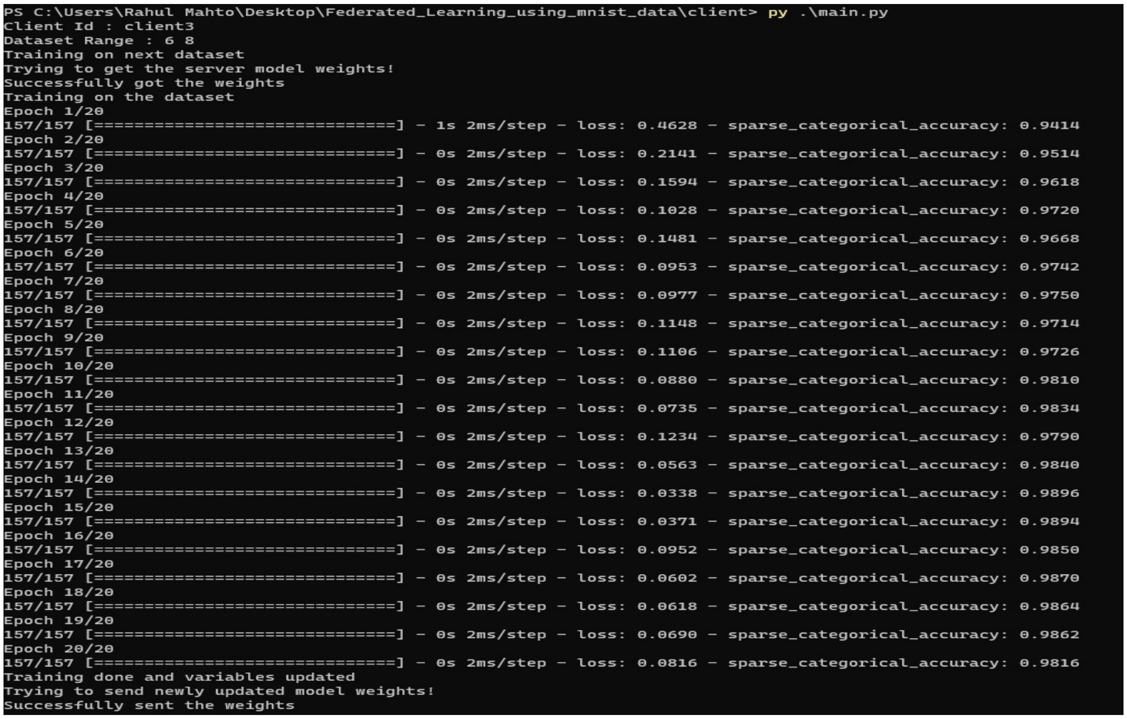
Command: cd server
Command: py main.py



```
D:\Federated_Learning_using_mnist_data\server>py main.py
Server Started ..............
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://192.168.253.127:5000
Press CTRL+C to quit
 * Restarting with stat
Server Started ..............
 * Debugger is active!
 * Debugger PIN: 145-530-668
192.168.253.127 - - [10/May/2023 21:58:37] "GET / HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/css/bootstrap.min.css HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/css/bootstrap-tweaks.css HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/css/prettify.css HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/css/default.css HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/img/grid.png HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/js/default.js HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/js/prettify-min.js HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/js/bootstrap.min.js HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:58:37] "GET /flask-api/static/js/jquery-1.8.1-min.js HTTP/1.1" 200 -
192.168.253.127 - - [10/May/2023 21:59:00] "GET /favicon.ico HTTP/1.1" 404 -
192.168.253.159 - - [10/May/2023 22:01:08] "POST / HTTP/1.1" 200 -
192.168.253.159 - - [10/May/2023 22:01:23] "POST / HTTP/1.1" 200 -
192.168.253.159 - - [10/May/2023 22:01:23] "POST / HTTP/1.1" 200 -
192.168.253.159 - - [10/May/2023 22:01:37] "POST / HTTP/1.1" 200 -
192.168.253.27 - - [10/May/2023 22:03:09] "POST / HTTP/1.1" 200 -
192.168.253.27 - - [10/May/2023 22:03:28] "POST / HTTP/1.1" 200 -
```

4. Start the clients:

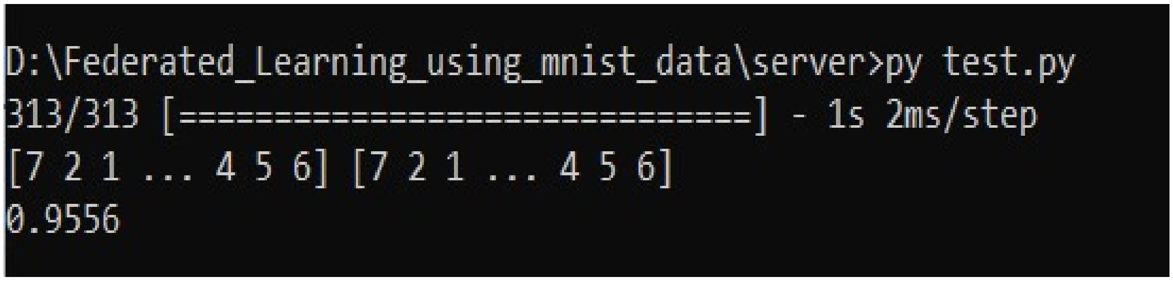
Command: cd client
Command: py main.py



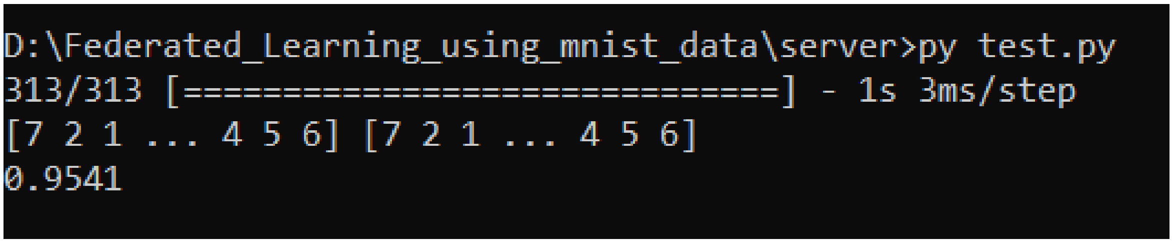
```
PS C:\Users\Rahul Mahto\Desktop\Federated_Learning_using_mnist_data\client> py .\main.py
Client Id : client3
Dataset Range : 6 8
Training on next dataset
Trying to get the server model weights!
Successfully got the weights
Training on the dataset
Epoch 1/20
157/157 [==============================] - 1s 2ms/step - loss: 0.4628 - sparse_categorical_accuracy: 0.9414
Epoch 2/20
157/157 [==============================] - 0s 2ms/step - loss: 0.2141 - sparse_categorical_accuracy: 0.9514
Epoch 3/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1594 - sparse_categorical_accuracy: 0.9618
Epoch 4/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1028 - sparse_categorical_accuracy: 0.9720
Epoch 5/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1481 - sparse_categorical_accuracy: 0.9668
Epoch 6/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0953 - sparse_categorical_accuracy: 0.9742
Epoch 7/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0977 - sparse_categorical_accuracy: 0.9750
Epoch 8/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1148 - sparse_categorical_accuracy: 0.9714
Epoch 9/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1106 - sparse_categorical_accuracy: 0.9726
Epoch 10/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0880 - sparse_categorical_accuracy: 0.9810
Epoch 11/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0735 - sparse_categorical_accuracy: 0.9834
Epoch 12/20
157/157 [==============================] - 0s 2ms/step - loss: 0.1234 - sparse_categorical_accuracy: 0.9790
Epoch 13/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0563 - sparse_categorical_accuracy: 0.9840
Epoch 14/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0338 - sparse_categorical_accuracy: 0.9896
Epoch 15/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0371 - sparse_categorical_accuracy: 0.9894
Epoch 16/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0952 - sparse_categorical_accuracy: 0.9850
Epoch 17/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0602 - sparse_categorical_accuracy: 0.9870
Epoch 18/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0618 - sparse_categorical_accuracy: 0.9864
Epoch 19/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0690 - sparse_categorical_accuracy: 0.9862
Epoch 20/20
157/157 [==============================] - 0s 2ms/step - loss: 0.0816 - sparse_categorical_accuracy: 0.9816
Training done and variables updated
Trying to send newly updated model weights!
Successfully sent the weights
```

5. Test the Model:

Command: cd server
Command: python3 test.py



```
D:\Federated_Learning_using_mnist_data\server>py test.py
313/313 [==============================] - 1s 2ms/step
[7 2 1 ... 4 5 6] [7 2 1 ... 4 5 6]
0.9556
```



```
D:\Federated_Learning_using_mnist_data\server>py test.py
313/313 [==============================] - 1s 3ms/step
[7 2 1 ... 4 5 6] [7 2 1 ... 4 5 6]
0.9541
```