University of Aberdeen

School of Natural and Computing Sciences

Department of Computing Science

2025 – 2026

Programming Assignment – Individually Assessed (no teamwork)

Title: JC4001 – Distributed Systems

Note: This assignment accounts for 30% of your total mark for the course.

Learning Outcomes

On successful completion of this component, a student will have demonstrated the ability to:

- Understand the principles of federated learning (FL) in distributed systems and how it differs from centralised machine learning.
- Implement a basic federated learning in distributed systems for image classification using the Fashion-MNIST dataset.
- Simulate a federated learning environment in distributed systems where multiple clients independently train models and the server aggregates them.
- Explore the effects of model aggregation and compare with centralised training.
- Evaluate the performance of the FL model under different conditions, such as non-IID data distribution and a varying number of clients.

Introduction

In this assignment, your task is to build a federated learning (FL) algorithm in a distributed system. FL is a distributed approach to train machine learning models, designed to guarantee local data privacy by training learning models without centralised datasets. The FL structure – as shown in Figure 1 – should include two parts: the first part is an edge server for model aggregation, and the second part should consist of several devices, each with a local dataset for local model updating. Then, each device transmits the updated local model to the edge server for model aggregation.

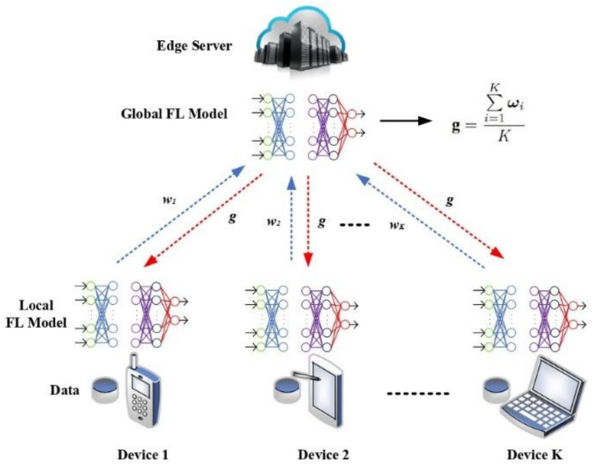

Figure 1: Illustration of the FL structure.

General Guidance and Requirements

Your assignment code and report must conform to the requirements given below and include the required content as outlined in each section. You **must** supply a written report, along with the corresponding code, containing all distinct sections/subtasks that provide a full critical and reflective account of the processes undertaken.

This assignment can be done in Python/PyCharm on your own device. If you work on your own device, be sure to move your files to MyAberdeen so we can run the application and mark it.

Note that it is your responsibility to ensure that your code runs on Python/PyCharm. By default, your code should run by directly clicking the “run” button. If your implementation uses some other command to start the code, it must be mentioned in the report.

Submission Guidelines

After you finish your assignment, please compress all your code files and the report into a compressed file and submit it in MyAberdeen (Course Content -> Course Assessment -> Individual Assignment -> Assignment Submission Link -> View Instructions -> Submission (Attach or Drag and drop files here) – once you are happy with the attachment, click submit). ***The deadline for assignment submission is 12 December 2025 at 23:59 Beijing time.***

Part 1: Understanding Federated Learning [5 points]

A. **Read the Research Paper:** You should read a foundational paper on federated learning, such as Communication-Efficient Learning of Deep Networks from Decentralised Data by McMahan et al. (2017).

B. **Summary Task:** Write a 500-word summary explaining the key components of federated learning (client-server architecture, data privacy, and challenges like non-IID data). **[5 points]**

Part 2: Centralised Learning Baseline [15 points]

A. **Implement Centralised Training:** You should implement a simple neural network using a centralised approach for classifying samples in the Fashion-MNIST dataset. This will serve as a baseline.

- Input: Fashion-MNIST dataset. **[5 points]**
- Model: A basic neural network with several hidden layers. **[5 points]**
- Task: Train the model and evaluate its accuracy. **[5 points]**

Part 3: Federated Learning Implementation [30 points]

A. **Simulate Clients**: Split the Fashion-MNIST dataset into several partitions to represent data stored locally at different clients. Implement a Python class that simulates clients, each holding a subset of the data. **[10 points]**

- Task: Implement a function to partition the data in both IID (independent and identically distributed) and non-IID ways.

B. **Model Training on Clients**: Modify the centralised neural network code so that each client trains its model independently using its local data. **[5 points]**

C. **Server-Side Aggregation**: Implement a simple parameter server that aggregates model updates sent by clients. Use the Federated Averaging (FedAvg) algorithm: **[10 points]**

- Each client sends its model parameters to the server after training on local data.
- The server aggregates these parameters (weighted by the number of samples each client has) and updates the global model.

D. **Communication Rounds**: Implement a loop where clients train their local models and the server aggregates them over multiple communication rounds. **[5 points]**

Part 4: Experimentation and Analysis [20 points]

A. **Experiment 1 - Impact of Number of Clients**: **[10 points]**

- Vary the number of clients (e.g., 5, 10, 20) and evaluate the accuracy of the final federated model.
- Plot the training accuracy and loss over communication rounds for each case.

B. **Experiment 2 - Non-IID Data: [10 points]**

- Modify the data distribution across clients to simulate a non-IID scenario (where clients have biased or skewed subsets of the data).

- Compare the performance of the federated learning model when clients have IID data vs. non-IID data. Plot the accuracy and loss over communication rounds for both cases.

Part 5: Performance Comparison with Centralised Learning [5 points]

A. Compare the federated learning model (both IID and non-IID) to the centralised learning baseline in terms of:

- Final accuracy
- Number of epochs/communication rounds needed to converge

Requirements and Marking Criteria for the Project Report [25 points]

You should write a report. Your report should describe the overall design of federated learning in a distributed system, along with the challenges encountered during implementation.

The marking criteria for the report are the following:

- Structure and completeness (all the aspects are covered) **[5 points]**.
- Clarity and readability (the language is understandable) **[5 points]**.
- Design explained **[5 points]**.
- Challenges discussed **[5 points]**.
- References to the sources **[5 points]**.

Contact

For any questions or clarifications, you can contact the course teachers: (1) Dr Yongchao Huang (yongchao.huang@abdn.ac.uk), (2) Dr Jari Korhonen (jari.korhonen@abdn.ac.uk) and (3) Dr Shahzad Mumtaz (shahzad.mumtaz@abdn.ac.uk)