早療字跡辨識系統

需求分析

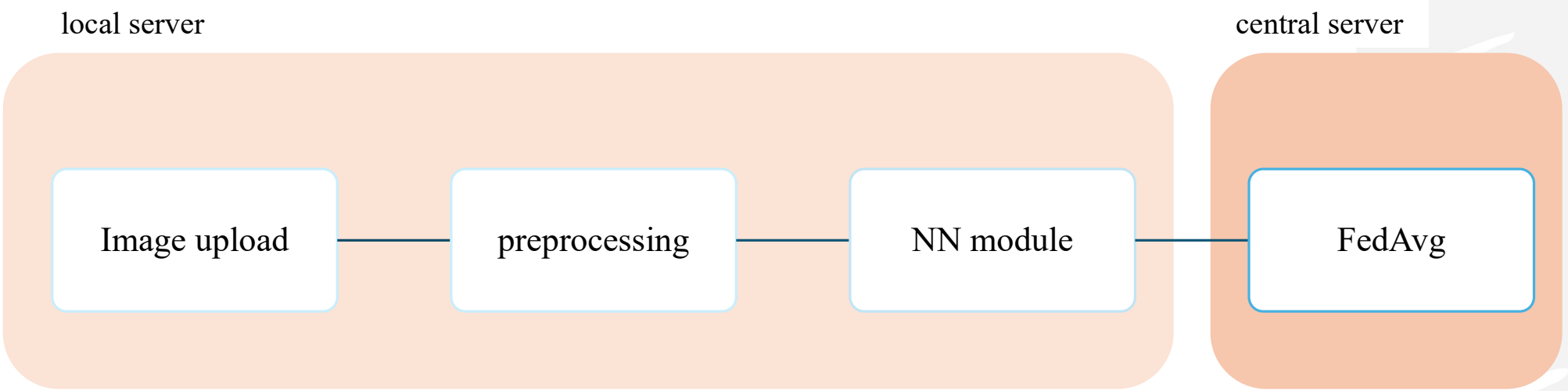
動機與範圍

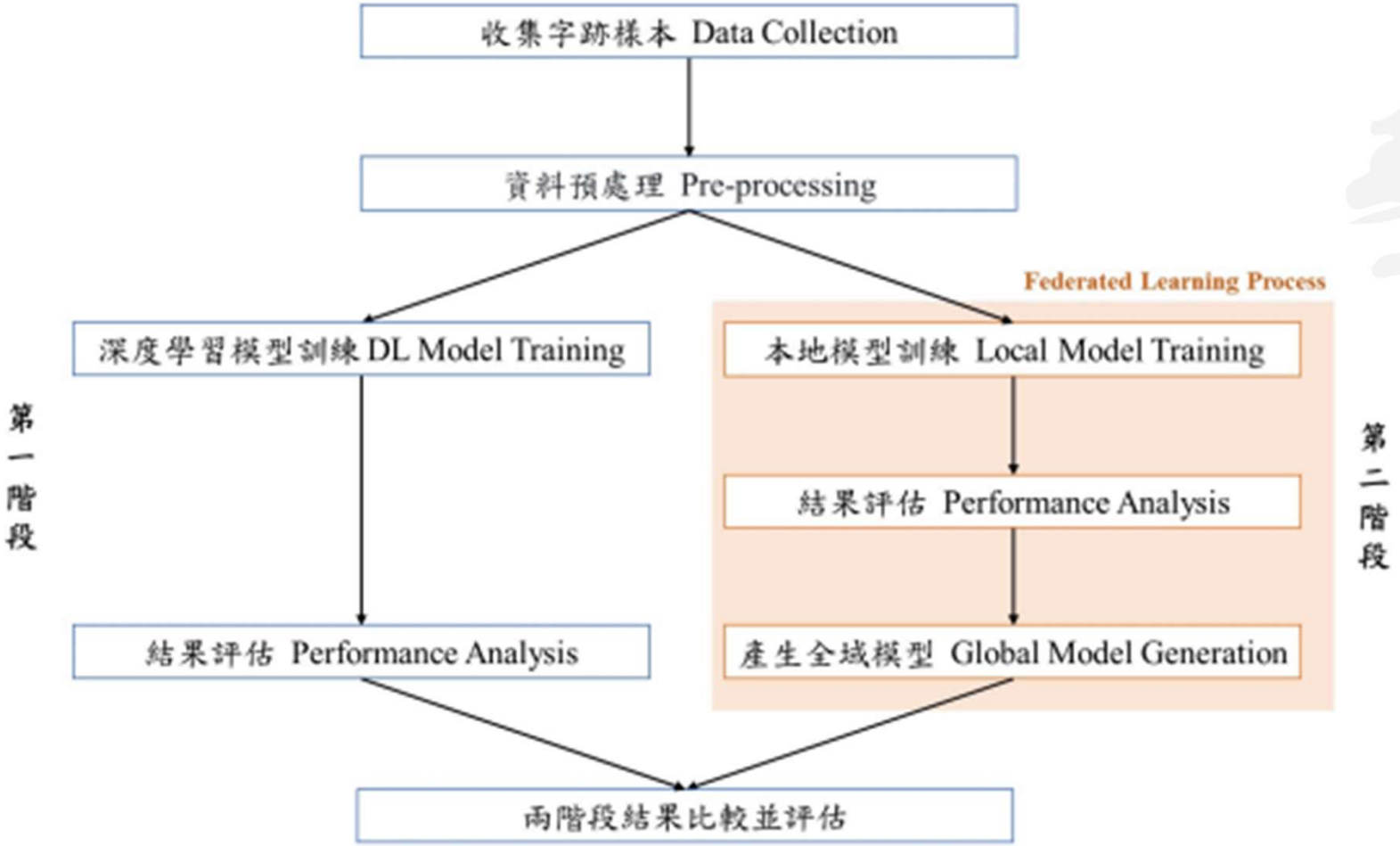
- ADHD與ASD等神經發展障礙難以察覺，未持續治療可能影響終身
- 神經發展障礙會影響精細動作控制，常伴隨書寫障礙出現
- 偏鄉兒童因經濟或交通上的不便，若有早療需求易錯過就醫黃金時期
- 手寫字跡檢測為相對簡單和易操作的早期識別書寫障礙管道
- 現代注重個人隱私，不同機構間難以集中收集大量資料做訓練，偏鄉地區更易因樣本數少，造成訓練困難

目標與價值

- 利用深度學習輔助辨識書寫障礙，協助找出需多加關注的兒童
- 確保使用者上傳資料不會離開本地，也能達成機構間的資訊共享
- 模型訓練去中心化，和中央伺服器連線失敗仍不影響訓練進度
- 樣本數較少的機構，可透過聯邦式學習共享，獲得以大量樣本訓練預測更精準的模型

實作方法


local server
central server
Image upload
preprocessing
NN module
FedAvg


收集字跡樣本 Data Collection
資料預處理 Pre-processing
Federated Learning Process
深度學習模型訓練 DL Model Training
本地模型訓練 Local Model Training
結果評估 Performance Analysis
結果評估 Performance Analysis
產生全域模型 Global Model Generation
兩階段結果比較並評估
第一階段
第二階段

預處理 – 增加樣本數

```python
# Data transformations
transform = transforms.Compose([
    transforms.Resize((400, 400)),  # Resize image
    transforms.RandomCrop((224, 224)),  # Randomly crop image
    transforms.RandomRotation(3),  # Randomly rotate image
    transforms.ToTensor(),  # Convert image to tensor
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])  # Normalize image
])
```

本地模型-書寫障礙辨識

```python
class SimpleNN(nn.Module):
    Codiumate: Options | Test this method
    def __init__(self):
        super(SimpleNN, self).__init__()
        self.conv_stack = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        )
        self.fc_stack = nn.Sequential(
            nn.Linear(64 * 112 * 112, 1000),
            nn.ReLU(),
            nn.Linear(1000, 2)
        )

    Codiumate: Options | Test this method
    def forward(self, x):
        x = self.conv_stack(x)
        x = torch.flatten(x, 1)
        logits = self.fc_stack(x)
        return logits
```

```python
class CNN(nn.Module):
    Codiumate: Options | Test this method
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(64 * 56 * 56, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()

    Codiumate: Options | Test this method
    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(x)
        x = self.relu(self.conv2(x))
        x = self.pool(x)
        x = self.flatten(x)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

client

1. Pickle 序列化模型訓練後的參數
2. gzip 壓縮後的參數和客戶 ID 存入文件
3. request.post 至伺服器

server

1. 檢查 request中是否有 model 文件
2. 接收和以 gzip 解壓縮檔案，並以 pickle 反序列化讀取參數
3. 將參數加入列表中，並檢查是否接收至少兩個模型參數
4. 條件達成則計算平均參數和更新 epoch，之後清空列表

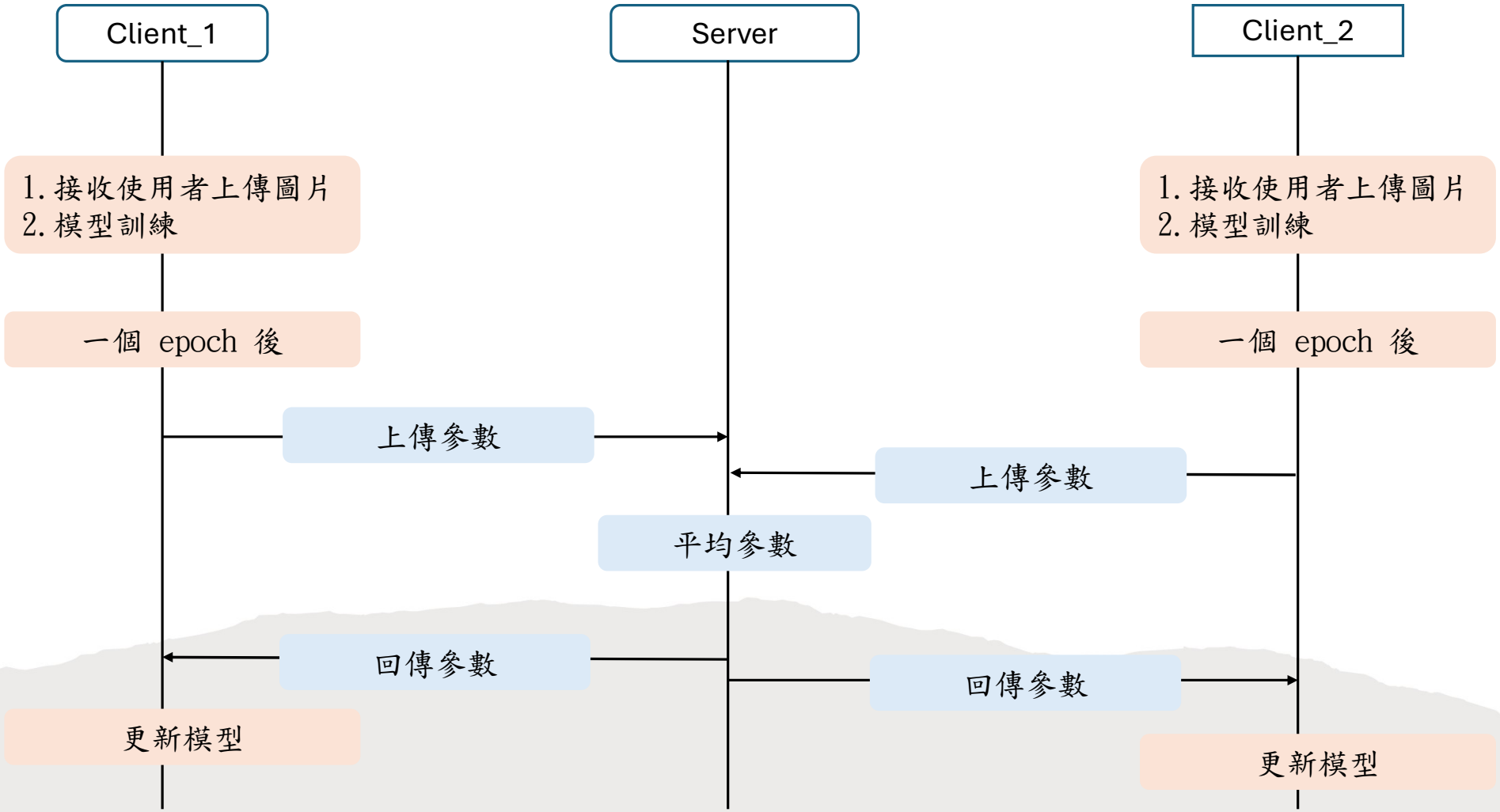
server

1. 以空字典儲存平均參數
2. 讀取客戶端請求的 epoch，若匹配則回傳 404
3. gzip 壓縮存有平均參數的字典，寫入內存文件
4. 回傳文件給客戶端

client

1. request.get 伺服器的模型參數，成功則寫入文件，失敗回傳 Fslse
2. gzip 讀取壓縮文件並加載至模型，pickle 讀取反序列化參數
3. 指定時間間隔多次嘗試從伺服器取得平均參數

系統架構


Client_1
Server
Client_2
1. 接收使用者上傳圖片
2. 模型訓練
一個 epoch 後
上傳參數
上傳參數
平均參數
回傳參數
回傳參數
更新模型
1. 接收使用者上傳圖片
2. 模型訓練
一個 epoch 後
更新模型

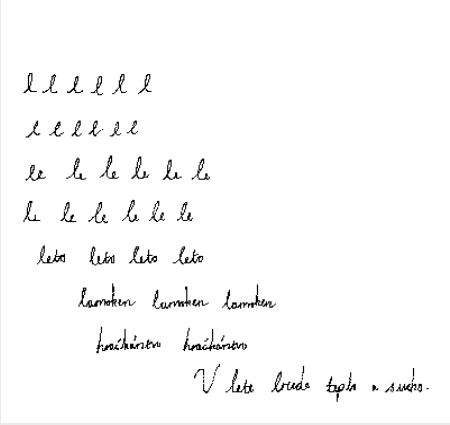
運作環境

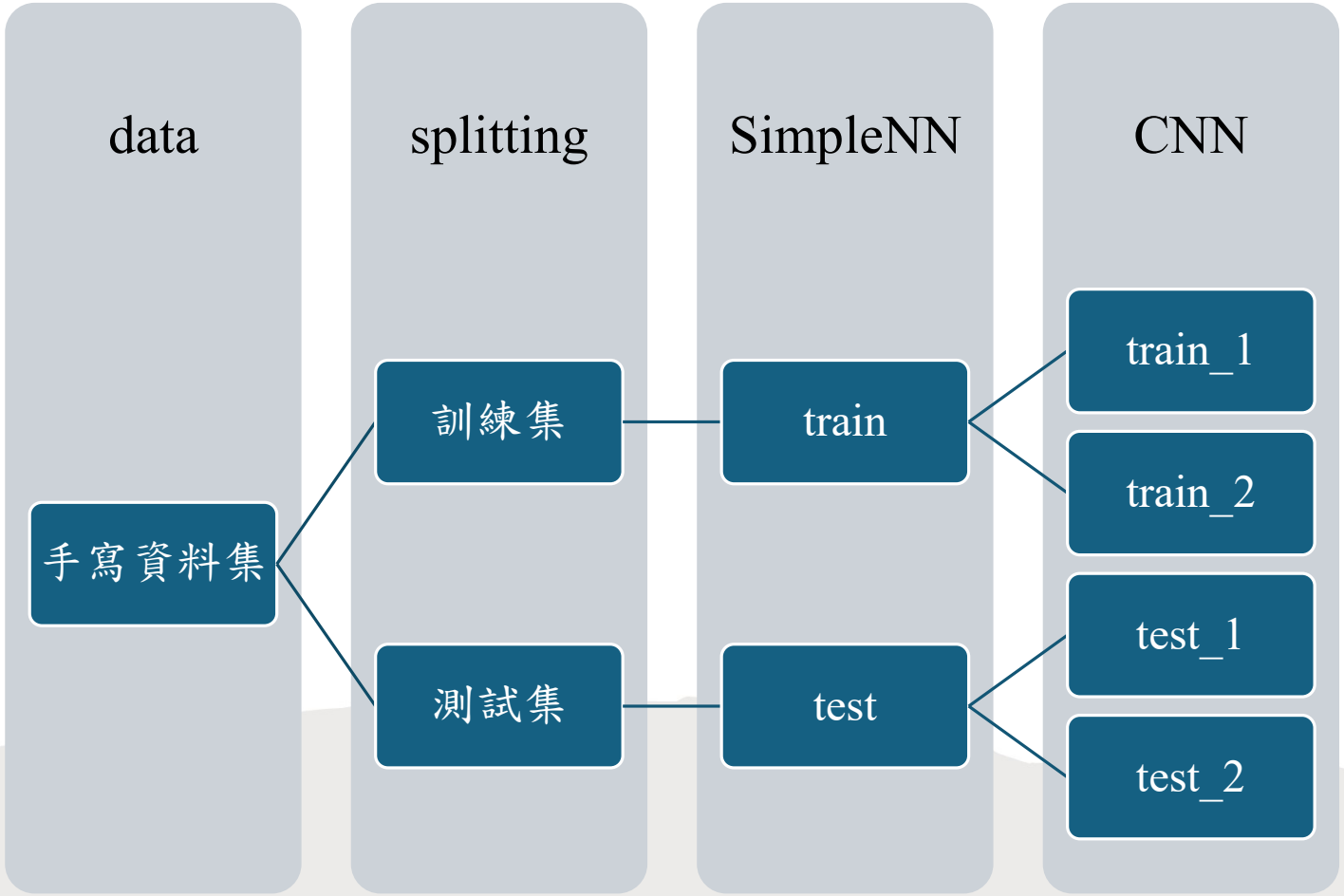
伺服器環境規格	
CPU	12th Gen Intel(R) Core(TM) i5-1235U　1.30 GHz
作業系統	Ubuntu
Web service	flask
程式語言	python

訓練資料來源

斯洛伐克文手寫集：https://github.com/Alar-q/Dysgraphia-Diagnosis

	Train_1	Train_2	Test_1	Test_2
SimpleNN	80	同train_1	20	同test_1
CNN	68	12	10	10




data
splitting
SimpleNN
CNN
手寫資料集
訓練集
測試集
train
test
train_1
train_2
test_1
test_2

主要功能

本地模型訓練

- 模擬圖片上傳
- 字跡辨識結果
- 參數收發更新

中央伺服器聚合

- 分散式訓練
- 非同步更新

測試影片

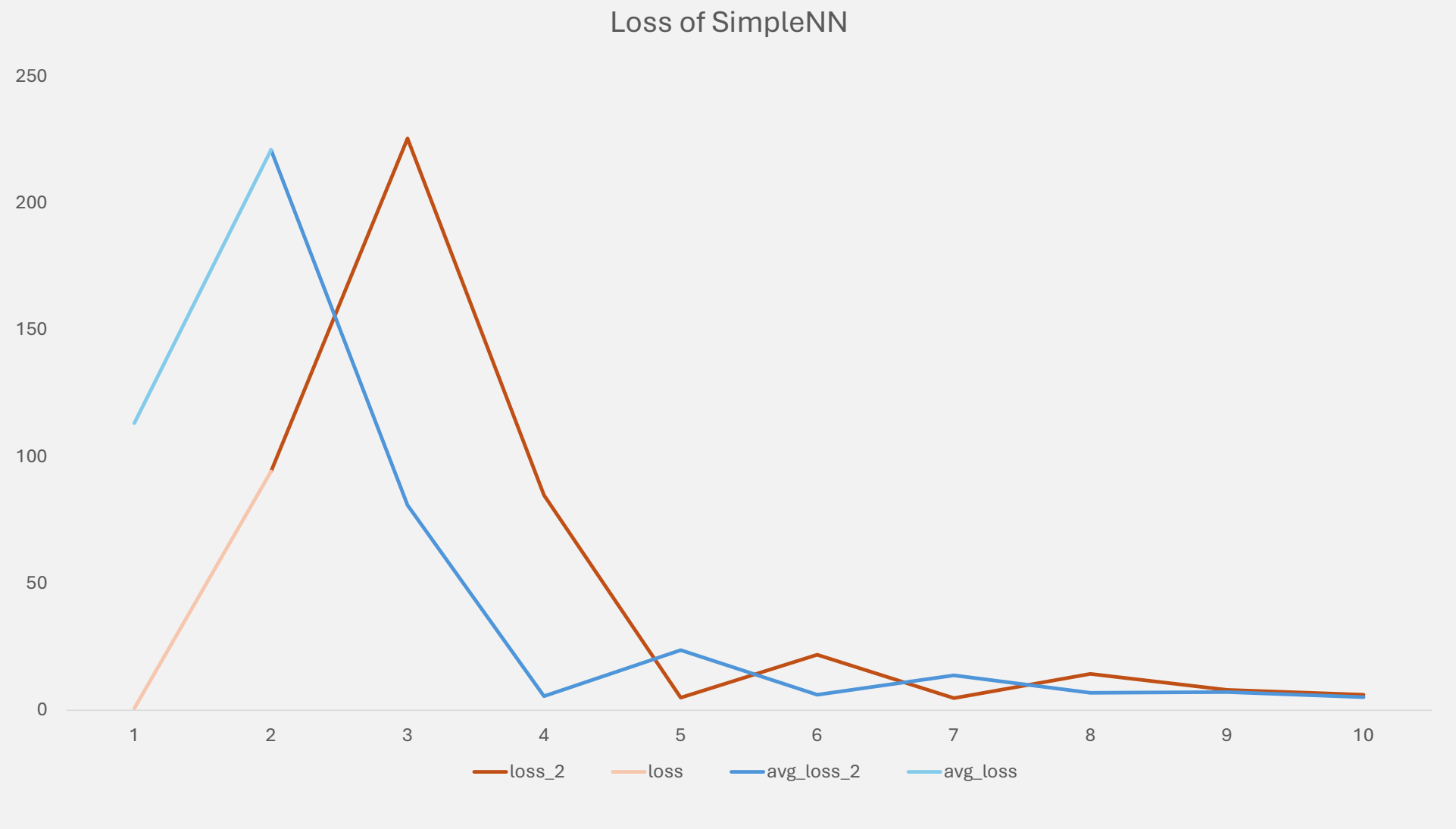
- 模擬影片(最末頁)： SimpleNN，2個 epoch，總時長6分57秒
- 測試結果:

```
Epoch 1
-------------------------------
/local-pool-83/cheng-yao/.local/lib/python3.10/site-packages/torch/autograd/__init__.py:266: UserWarning: CUDA initialization: Unexpected error from cudaGetDeviceCount(). Did
you run some cuda functions before calling NumCudaDevices() that might have already set an error? Error 804: forward compatibility was attempted on non supported HW (Triggered
 internally at ../c10/cuda/CUDAFunctions.cpp:108.)
  Variable._execution_engine.run_backward(  # Calls into the C++ engine to run the backward pass
loss: 0.740931  [    0/  100]
Parameters uploaded successfully.
Test Error:
 Accuracy: 50.0%, Avg loss: 113.104294

Epoch 2
-------------------------------
loss: 107.712555  [    0/  100]
Parameters uploaded successfully.
```

Client_1

```
Epoch 1
-------------------------------
/local-pool-83/cheng-yao/.local/lib/python3.10/site-packages/torch/autograd/__init__.py:266: UserWarning: CUDA initialization: Unexpected error from cudaGetDeviceCount(). Did
you run some cuda functions before calling NumCudaDevices() that might have already set an error? Error 804: forward compatibility was attempted on non supported HW (Triggered
 internally at ../c10/cuda/CUDAFunctions.cpp:108.)
  Variable._execution_engine.run_backward(  # Calls into the C++ engine to run the backward pass
loss: 0.767739  [    0/  100]
Parameters uploaded successfully.
Test Error:
 Accuracy: 50.0%, Avg loss: 113.175430

Epoch 2
-------------------------------
loss: 94.192284  [    0/  100]
 Parameters uploaded successfully.
```

Client_2


Loss of SimpleNN
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
loss_2
loss
avg_loss_2
avg_loss

```
INFO:werkzeug:127.0.0.1 - - [11/Jun/2024 16:36:35] "POST /upload?client_id=client_1 HTTP/1.1" 200 -
INFO:root:Sending averaged parameters for epoch 0
INFO:werkzeug:127.0.0.1 - - [11/Jun/2024 16:36:37] "GET /model?epoch=0 HTTP/1.1" 200 -
INFO:root:Received parameters from client. Total received: 1
INFO:werkzeug:127.0.0.1 - - [11/Jun/2024 16:36:42] "POST /upload?client_id=client_1 HTTP/1.1" 200 -
INFO:root:Received parameters from client. Total received: 2
INFO:root:Parameters averaged for epoch 1
```

CNN + FedAvg

- 執行10個epoch ： 55分
- 設定共2個客戶端，皆上傳參數後更新模型

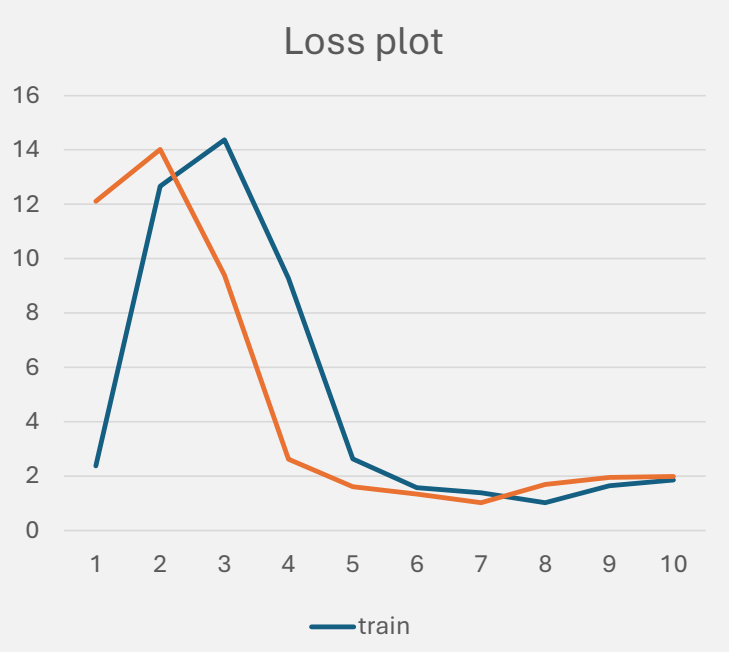
CNN


Loss plot
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
train


Loss plot
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
train
test

CNN + FedAvg

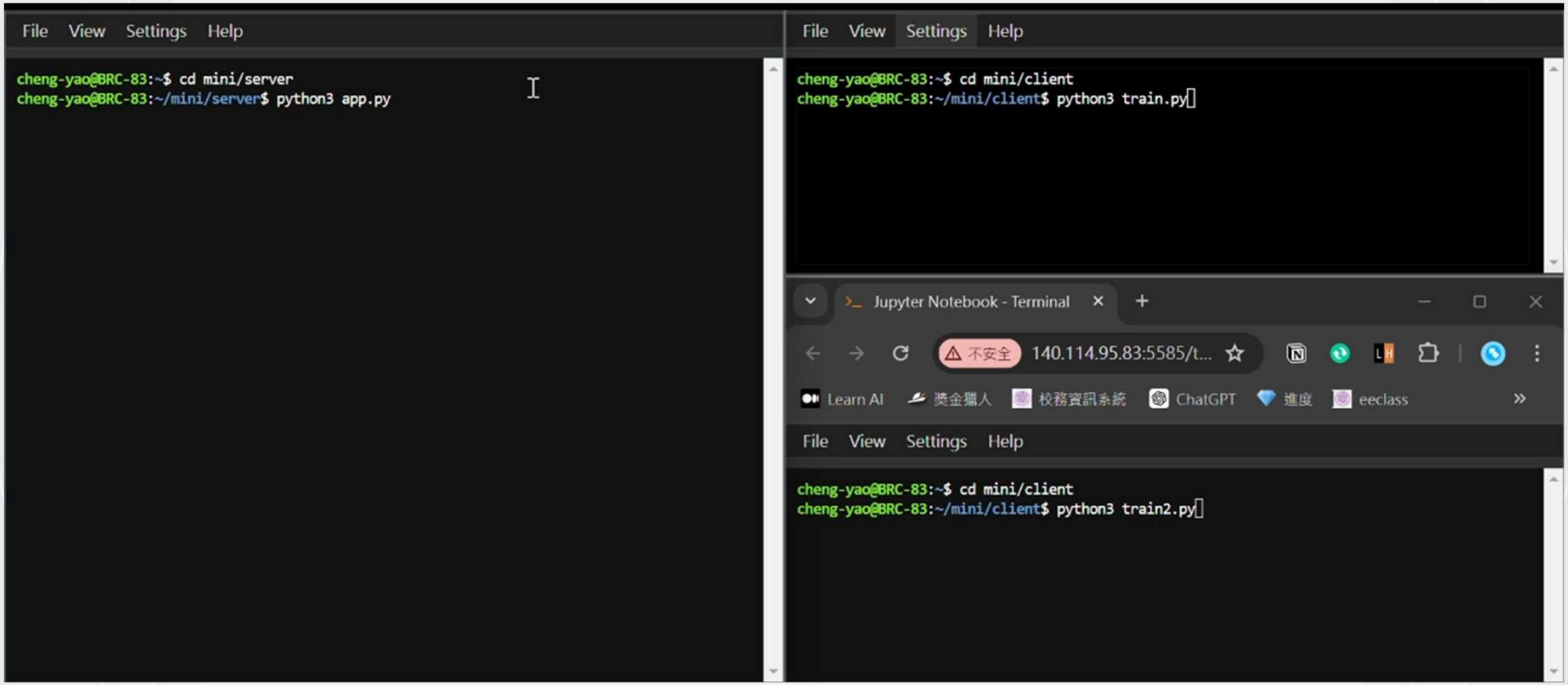

Loss plot
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1 2 3 4 5 6 7 8 9 10
train
test


Loss plot
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10
train

結果比較

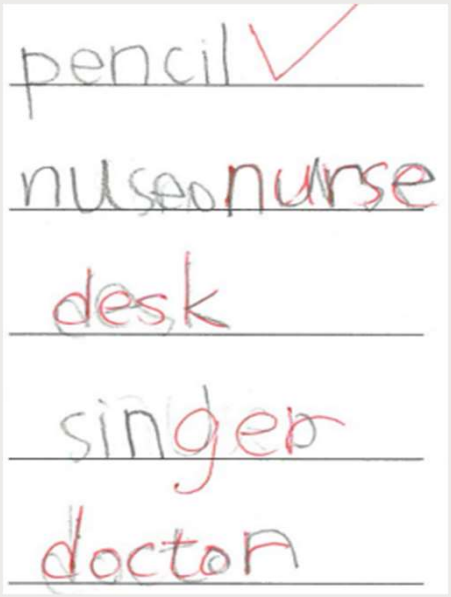

Loss plot
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
train
test


Loss plot
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
train
test


File View Settings Help
cheng-yao@BRC-83:~$ cd mini/server
cheng-yao@BRC-83:~/mini/server$ python3 app.py
File View Settings Help
cheng-yao@BRC-83:~$ cd mini/client
cheng-yao@BRC-83:~/mini/client$ python3 train.py
Jupyter Notebook - Terminal
不安全
140.114.95.83:5585/t...
Learn AI
獎金獵人
校務資訊系統
ChatGPT
進度
eeclass
File View Settings Help
cheng-yao@BRC-83:~$ cd mini/client
cheng-yao@BRC-83:~/mini/client$ python3 train2.py

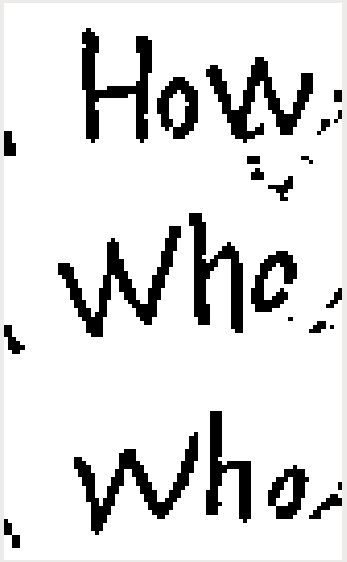
第二階段–英文手寫辨識

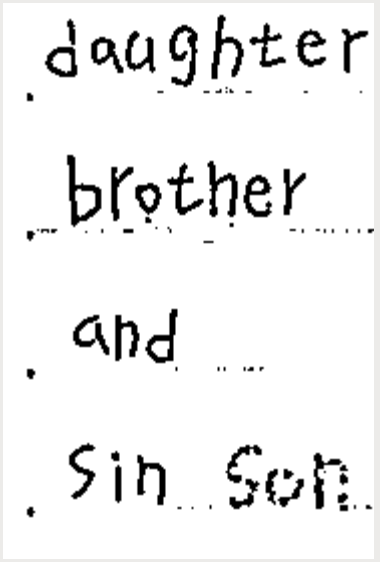
- 由博幼基金會提供學童手寫英文考卷擷取訓練
- 增加預處理程序 - 去除鉛筆筆跡以外的劃記
- 資料標記 - 劃分為特教生1和一般生0
- 其餘步驟皆和第一階段 CNN + FedAvg 相同



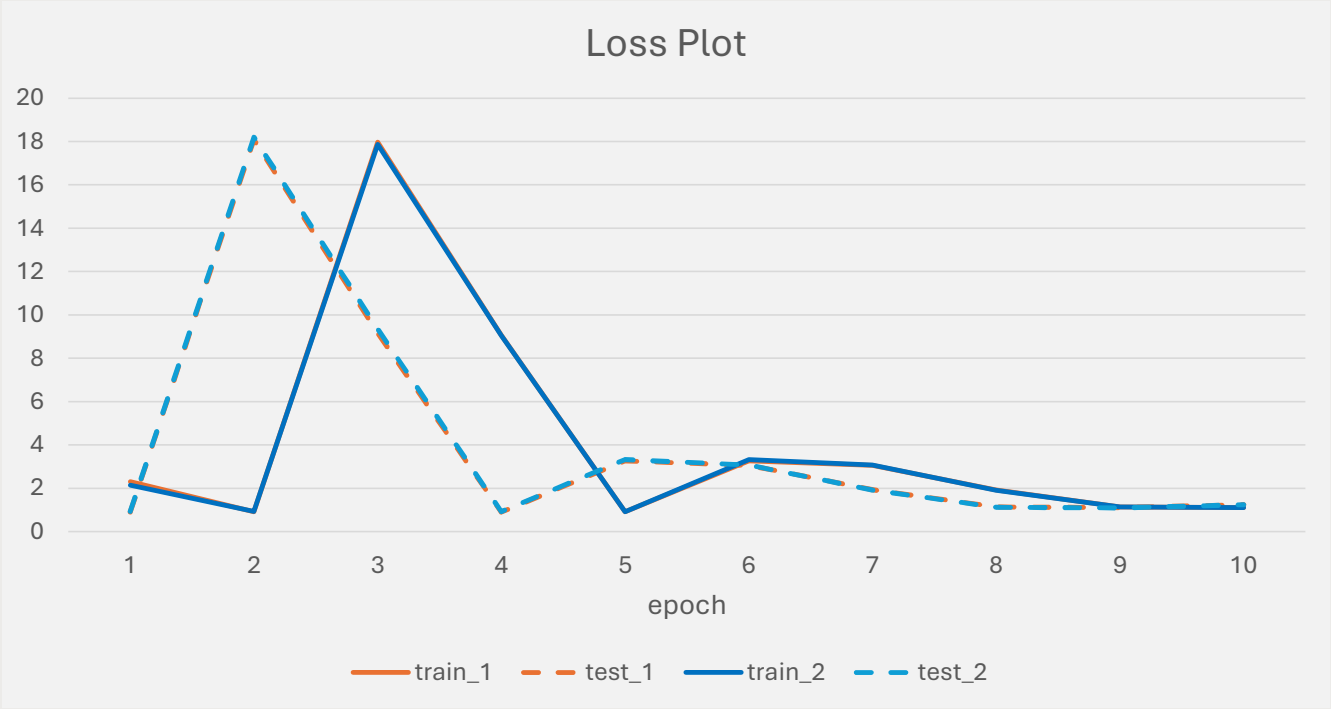
預處理

- 去除印刷底線、題目
- 去除紅、藍筆劃記
- 過濾躁點
- 增強鉛筆手寫訊號和對比度





結果


Loss Plot
20
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
epoch
train_1
test_1
train_2
test_2

結論

- 相同資料量，使用聯邦式學習相較未使用，訓練過程 loss 較少
- 聯邦式學習可以作為不同機構間保有隱私溝通的管道
- 非同步更新使個別模型能獨立訓練
- 英文手寫的分類 1 資料量不足