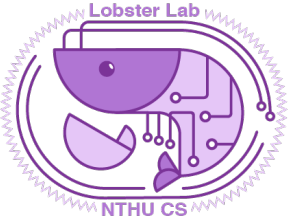


Horizontal Federated Learning



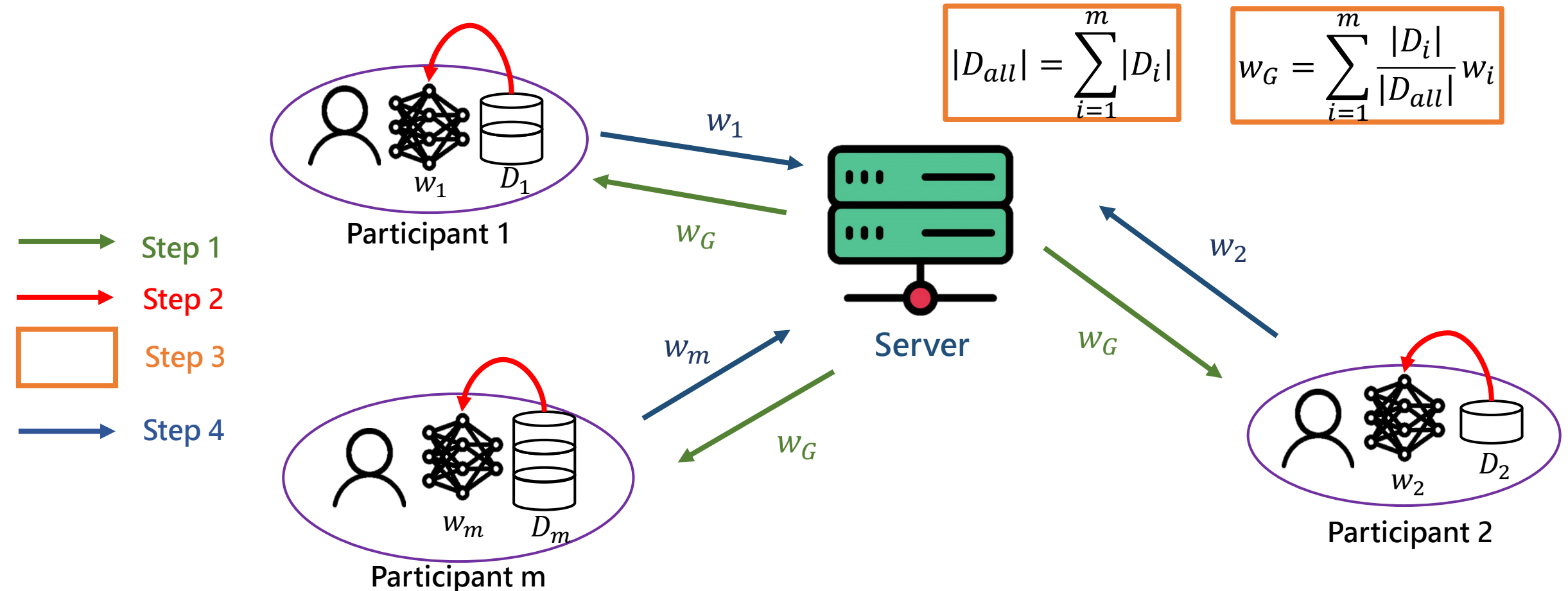
大綱

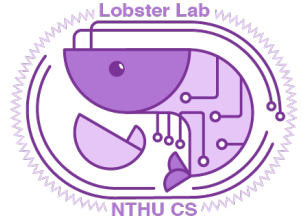
- 作業介紹
- 提供的 code
- 繳交內容
- 評分標準

作業介紹



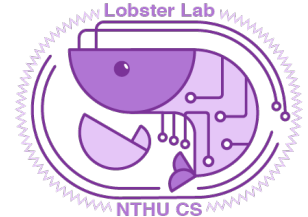
Horizontal Federated Learning





提供的完整檔案

- 訓練主程式：main.py
- server：FLAlgorithms/servers/serverbase.py, serveravg.py
- user: FLAlgorithms/users/ userbase.py, useravg.py
- 訓練資料：data/CIFAR10
- 切割資料：data/CIFAR10/generate_niid_dirichlet.py
- 執行指令：run.sh

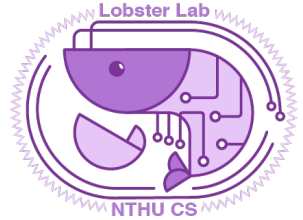


提供的完整檔案

```
.
├── data
│   └── CIFAR10
│       └── generate_niid_dirichlet.py
├── FAlgorithms
│   ├── servers
│   │   ├── serveravg.py
│   │   └── serverbase.py
│   ├── trainmodel
│   │   └── resnet.py
│   └── users
│       ├── useravg.py
│       └── userbase.py
├── main.py
├── run.sh
└── utils
    ├── log_utils.py
    ├── model_config.py
    └── model_utils.py
```

---- HFL

```
|---- data
|    |---- CIFAR10
|        |---- generate_niid_dirichlet.py
|---- FAlgorithms
|    |---- servers
|        |---- serverbase.py
|        |---- serveravg.py
|    |---- trainmodel
|        |---- resnet.py
|    |---- users
|        |---- userbase.py
|        |---- useravg.py
|---- utils
|    |---- log_utils.py
|    |---- model_config.py
|    |---- model_utils.py
|---- main.py
|---- run.sh
```



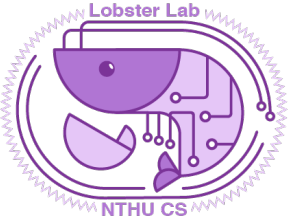
需要由同學實作的兩個部分

1. Server : serverbase.py

- 請完成 class Server 中 `select_users(self, model, beta)`、`aggregate_parameters(self)` 兩個 functions。
- `select_users`: server 去選擇哪些 users 參與這一輪的訓練
- `aggregate_parameters`: server 根據 users 所擁有的訓練資料比例，將 users 的模型加權平均，形成新的 global model

2. User : userbase.py

- 請完成 class User 中 `set_parameters(self, model, beta)` 這個 function。
- `set_parameters`: 將 global model 分配給 users，用 global model 初始化自己的 local model



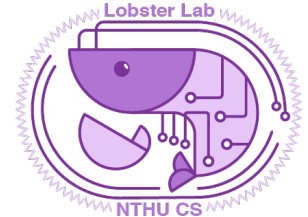
需要由同學探討的部分

1. Data distribution

- `python generate_niid_dirichlet.py --n_class 10 --sampling_ratio 1.0 --alpha {50.0} --n_user 10`
- 請同學探討 α 在 $\{0.1, 50.0\}$ 兩種情況下，每個 user 資料分布的情況，以及對訓練出來的 global model accuracy 會有什麼差異
- 附兩個對應的 global model accuracy 截圖在報告中

2. Number of users in a round

- `python main.py --dataset CIFAR10-alpha100.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 200 --local_epochs 20 --num_users {10} --learning_rate 0.1 --model resnet18 --device cuda`
- 請同學探討 num_users 在 $\{2, 10\}$ 兩種情況下，訓練出來的 global model accuracy 與模型收斂速度的差異



指令

1. Data distribution

```
# split CIFAR10
cd ./data/CIFAR10
python generate_niid_dirichlet.py --n_class 10 --sampling_ratio 1.0 --alpha 0.1 --n_user 10
python generate_niid_dirichlet.py --n_class 10 --sampling_ratio 1.0 --alpha 50.0 --n_user 10
python generate_niid_dirichlet.py --n_class 10 --sampling_ratio 1.0 --alpha 100.0 --n_user 10
```

2. Train

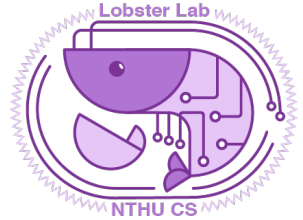
```
# run (pwd: ./)
1. Data distribution
alpha50.0: python main.py --dataset CIFAR10-alpha50.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10
alpha0.1: python main.py --dataset CIFAR10-alpha0.1-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10
```

2. Number of users in a round:

```
2 users: python main.py --dataset CIFAR10-alpha50.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 2
10 users: python main.py --dataset CIFAR10-alpha50.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10
```

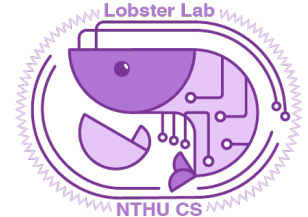
3. Model Accuracy

```
python main.py --dataset CIFAR10-alpha100.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10
```



需要使用到的套件

- pytorch – 1.10.1
 - torchaudio – 0.10.1
 - torchvision – 0.11.2
 - cudatoolkit – 11.3.1
 - numpy – 1.21.5
 - python – 3.9.12
- OS: Ubuntu 20.04.4 LTS
 - CPU: AMD Ryzen 9 5900X
12-Core Processor
 - GPU: GeForce RTX 3080 Ti
 - You can also run the codes
in *Google Colab*.



繳交內容

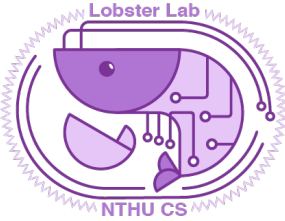
- 提供的 code 以及完成 TODO 的 serverbase.py 以及 userbase.py
- 報告 60% : 請在其中敘述同學在此次作業中 (以下列的為 “必須” 包含內容)
 - 10%: 解釋你寫的程式碼 (可以使用comment方式簡短解釋即可)
 - 20%: 探討問題的原因
 - 5%: 最終acc的輸出截圖 (--num_users 10, --alpha 100.0)
 - 10%: 請簡短說明此次作業中學到的重點
 - 5% : 檔名 : {學號}_report.pdf , 例如: 110123456_report.pdf
 - 最後請轉為 pdf 檔
 - 10% : 請用螢幕錄影錄下
 - 最後 10 個 round 的輸出結果
 - 檔名 : {學號}_result.mp4 , 例如: 110123456_result.mp4
- 請將所有檔案 (包含程式、報告、螢幕錄影) 打包成 .zip 檔
 - 檔名 : {學號}_hw5.zip , 例如: 110123456_hw5.zip
- 注意 : 繳交的code必須要是使用提供的script 檔 run.sh可執行的
 - 否則程式部分不予計分 , 只會以報告內容中同學的嘗試、分析以及心得給予報告分數

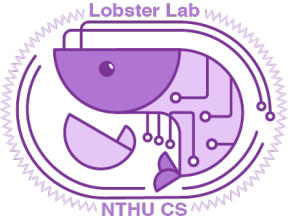


繳交內容

```
---- HFL
|---- data
|    |---- CIFAR10
|    |    |---- generate_niid_dirichlet.py
|---- FLAlgorithms
|    |---- servers
|    |    |---- serverbase.py
|    |    |---- serveravg.py
|    |---- trainmodel
|    |    |---- resnet.py
|    |---- users
|    |    |---- userbase.py
|    |    |---- useravg.py
|---- utils
|    |---- log_utils.py
|    |---- model_config.py
|    |---- model_utils.py
|---- main.py
|---- run.sh

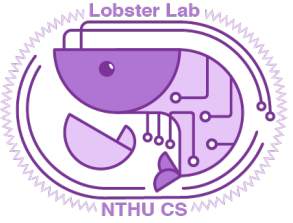
---- 110123456_report.pdf
---- 110123456_result.mp4
```





評分標準

- 60 % report
- 20 % 完成 serverbase.py 中的 TODO
- 10 % 完成 userbase.py 中的 TODO
- 10 % 最終 accuracy
 - 前10%: 10分
 - 前11%-50%: 7分
 - 前51%-90%: 4分
 - 最後10%: 1分
- 鼓勵討論，請勿抄襲同學作業



繳交期限

- 6 / 18 (二) 23:59 前上傳至作業區
- 若有其他問題，同學可以再來信詢問，
- Email: lobsterlab.cs.nthu@gmail.com
- 信件主旨請以 **[DS HW5]** 開頭，並於信中註明學號和姓名
- Example: [DS HW5] 這是範例