### serverbase.py

```
def select_users(self, round, num_users):
    return np.random.choice(self.users, num_users) #Use numpy module random choice function to choice num_users of users

## TODO
....
Randomly select {num_users} users from all users
Args:
    round: current round
    num_users: number of users to select
Return:
    List of selected clients objects

Hints:
    1. Default 10 users to select, you can modify the args {--num_users} to change this hyper-parameter
    2. Note that {num_users} can not be larger than total users (i.e., num_users <= len(self.user))
....</pre>
```

使用 numpy module 內建的 random choice function 去挑選 users

```
def aggregate_parameters(self):
   total_train_samples = 0
   new parameters = []
   for i in self.selected_users[0].get_parameters():
       new_parameters.append(0)
   for users in self.selected_users:
       for i, parameter in enumerate(users.get_parameters()):
           new_parameters[i] += parameter * users.train_samples
       total_train_samples += users.train_samples
   for parameter in new_parameters:
       parameter /= total train samples
   for parameter, new_parameter in zip(self.model.parameters(), new_parameters):
       parameter.data = new_parameter.data.clone()
   Weighted sum all the selected users' model parameters by number of samples
   Args: None

    Use self.selected_users, user.train_samples.

       2. Replace the global model (self.model) with the aggregated model.
```

將 selected 的 user model 的 parameter 做加權平均並更新 global model 的 parameter,程式中共有四個 block(for 迴圈),第一個 block 是做初始化,第二個 block 做 user model 的加權(權重是 user 的 train\_samples),第三個 block 對家權 過的每個 parameter 做平均,第四個 block 更新 global model 的 parameter

# userbase.py

```
def set_parameters(self, model, beta=1):
    new_parameter = []
    for global_parameter, local_parameter in zip(model.parameters(), self.get_parameters()):
        new_parameter.append(beta * global_parameter + (1 - beta) * local_parameter)
    self.update_parameters(new_parameter)
    ## TODO
    ...

Replace the user's local model with the global model
Args:
        model: the global model parameters
        beta: moving average model,
              i.e., user's model parameters = beta * global model parameters + (1 - beta) * user's model parameters
Return:
        None

Hint:
        1. You can use self.model (the user's model), model (global model parameters).
        ...
```

利用 global model 的 parameter 與 user model 的 parameter 以特定比例(beta,1 - beta)更新 user model 的 parameter

### Data Distribution:

當 alpha 值為 0.1 時:

用戶樣本數量相對較少,對於模型聚合的貢獻較小。

擁有更多樣本的用戶(alpha 值較高)在模型聚合中的影響更大,其模型參數对於全局模型的更新具有更大的權重。

参與訓練的用戶之見的數據分布可能存在較大差異,由於樣本數量不均衡,部 分用戶的數據特徵可能無法很好地被捕捉到,可能導致全局模型的準確性下 降。

|    |   | T V | 9 =    | AA EJ |   |
|----|---|-----|--------|-------|---|
| 0  | <pre>import os os.chdir('<u>content/drive/NyDrive/NEL</u>') ####################################</pre>  |     |        |       |   |
|    | python main.pydataset CIFAR10-alpha0.1-ratio1.0-users10algorithm Fedävgnum_glob_iters 150local_epochs 10num_users 10learning_rate 0.1model resnet | 18  | device | cuda  |   |
| C+ | Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       | ^ |
|    | Round number: 142   |     |        |       |   |
|    | Average Global Accurancy = 0.2531, Loss = 2.14. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    | Round number: 143   |     |        |       |   |
|    | Average Global Accurancy = 0.2067, Loss = 3.61. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    | Round number: 144   |     |        |       |   |
|    | Average Global Accurancy = 0.1448, Loss = 2.87. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    | Round number: 145   |     |        |       |   |
|    | Average Global Accurancy = 0.1581, Loss = 2.75, Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139,  |     |        |       |   |
|    | Round number: 146   |     |        |       |   |
|    | Average Global Accurancy = 0.2165, Loss = 2.26. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    |   |     |        |       |   |
|    | Average Global Accurancy = 0.2500, Loss = 2.25. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    | Round number: 148   |     |        |       |   |
|    | Average Global Accurancy = 0.2128, Loss = 2.15. Best Global Accurancy = 0.2885, Loss = 2.45, Iter = 139.  |     |        |       |   |
|    | Round number: 149   |     |        |       |   |
|    | Average Global Accurancy = 0.2313, Loss = 2.51. Best Global Accurancy = 0.2865, Loss = 2.45, Iter = 139. Finished training.                       |     |        |       | İ |

#### 當 alpha 值為 50.0 時:

用戶樣本數量相對較多,對於模型聚合的貢獻較大。 用戶的數據更具代表性,對於全局模型的更新具有更大的影響力。 數據分布更均衡,不同用戶之間的數據特徵可能更能夠相互補充和提升,有助 於提高全局模型的準確性。

| 0 | import os os. cháir ("Content/drive/MyGrive/MFL") fisiphs 50.0 fisiphs 50.0 fyrthon main.py —dataset CIFAR10-alpha50.0-ratiol.0-users10 —algoritha Feddyg —num_glob_iters 150 —local_epochs 10 —num_users 10 —learning_rate 0.1 —model r | ↑ ↓ ⇔ □ ‡ ☐ : |   |
|---|--|---------------|---|
|   | Best Global Accurancy = 0.8009. Loss = 0.77. Iter = 135.   | 4             | Δ |
|   | Round number: 142  |               |   |
|   | Average Global Accurancy = 0.7856, Loss = 0.81. Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 135.   |               |   |
|   | Round mumber: 143  |               |   |
|   | Average Global Accurancy = 0.7668, Loss = 0.97.  Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 135.  |               |   |
|   | Round number: 144  |               |   |
|   | Average Global Accurancy = 0.7725, Loss = 0.85. Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 138.   |               |   |
|   | Round number: 146  |               |   |
|   | Average Global Accurancy = 0,7824, Loss = 0,80. Best Global Accurancy = 0,8009, Loss = 0,77, Iter = 138.   |               |   |
|   | Round number: 146  |               |   |
|   | Average Global Accurancy = 0.7879, Loss = 0.94. Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 135.   |               |   |
|   |  |               |   |
|   | Round masher: 147  |               |   |
|   | Average Global Accurancy = 0.7821, Loss = 0.79. Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 135.   |               |   |
|   | Round number: 148  |               |   |
|   | Average Global Accurancy = 0.7832, Loss = 0.80. Best Global Accurancy = 0.8009, Loss = 0.77, Iter = 135.   |               |   |
|   | Round number: 149  |               |   |
|   | Average Global Accurancy = 0.7749. Loss = 0.84. Best Global Accurancy = 0.8009. Loss = 0.77. Iter = 135. Finished training.  |               |   |

# Number of users in a round:

當 num\_users 值為 2 時:

用戶多樣性較低,較少數量的用戶參與訓練可能導致數據分佈不平衡,無法充分利用數據的多樣性,從而影響準確性。

較少用戶同時參與訓練雖然減少並行計算效率,降低模型訓練的速度,但由於 通信開銷降低導致通信效率的上升,導致訓練速度可能升高。

|   | 1 V V 4 4 6   | / = | i |
|---|---|-----|---|
| Ī | os.chdir('/content/drive/Ngrive/NEL') !python main.py —dataset CIFAR10-alpha50.0-ratiol.0-users10algorithm Feddwgnum_glob_iters 150local_epochs 10num_users 2learning_rate 0.1model resmet18device cuda |     |   |
| D | Best Global Accurancy = 0.6493, Less = 1.01, Iter = 118.  |     | Δ |
|   |   |     |   |
|   | Average Global Accurancy = 0.5986, Loss = 1.19. Eest Global Accurancy = 0.6493, Loss = 1.01, Iter = 118.  |     |   |
|   |   |     |   |
|   | Average Global Accurancy = 0.5308, Loss = 1.48. Eest Global Accurancy = 0.6403, Loss = 1.01, Iter = 118.  |     |   |
|   | Round number: 144   |     |   |
|   | Average Global Accurancy = 0.6211, Loss = 1.10. Est Global Accurancy = 0.6493, Loss = 1.01, Iter = 118.   |     |   |
|   | Round number: 145   |     |   |
|   | Average Global Accurancy = 0.6399, Loss = 1.01. Eest Global Accurancy = 0.6493, Loss = 1.01, Iter = 118.  |     |   |
|   | Round number: 146   |     |   |
|   | Average Global Accurancy = 0.6251, Loss = 1.09. Best Global Accurancy = 0.6493, Loss = 1.01. Iter = 118.  |     |   |
|   |   |     |   |
|   |   |     |   |
|   | Average Global Accurancy = 0.5100, Loss = 1.66. Best Global Accurancy = 0.6493, Loss = 1.01, Iter = 118.  |     |   |
|   |   |     |   |
|   | Average Global Accurancy = 0.6598, Loss = 1.00.  Best Global Accurancy = 0.6598, Loss = 1.00. Iter = 148.   |     |   |
|   |   |     |   |
|   | Average Global Accurancy = 0.6468, Loss = 1.03. Best Global Accurancy = 0.6598, Loss = 1.00. Teer = 148. Finished textining.  |     | į |

#### 當 num\_users 值為 10 時:

用戶多樣性較高,較多數量的用戶參與訓練可以提供更多樣化的數據樣本,有助於全局模型學習更廣泛的特徵和模式,從而提高準確性。

較多用戶同時參與訓練可以提高並行計算效率,加快模型訓練的速度,但由於 通信開銷上升導致通信效率的下降,導致訓練速度可能下降。

| [] | import os os.chdir(' <u>content/drive/NyOrive/NFL'</u> )   Syrthon anin.pydataset CIFARIO-alpha100.0-ratiol.0-users10algoritha Fedavgnum_glob_iters 150local_epochs 10num_users 10learning_rate 0.1aodel resnet18device cuda |   |
|----|--|---|
|    | Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139.   |   |
|    |  |   |
|    | Average Global Accurancy = 0.8095, Loss = 0.70. Best Global Accurancy = 0.8153, Loss = 0.67. Iter = 139.   |   |
|    | Round number: 143  |   |
|    | Average Global Accurancy = 0.8140, Loss = 0.68. Best Global Accurancy = 0.8153, Loss = 0.67. Iter = 139.   |   |
|    | Round number: 144  |   |
|    | Average Global Accurancy = 0.7985, Loss = 0.74. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139.   |   |
|    | Round number: 145  |   |
|    | Average Global Accurancy = 0.8099, Loss = 0.69. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139.   |   |
|    | Round number: 146  |   |
|    | Average Global Accurancy = 0.8041, Losz = 0.71.  Best Global Accurancy = 0.8163, Loss = 0.67, Iter = 139.  |   |
|    |  |   |
|    | Average Global Accurancy = 0.8080, Loss = 0.69. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139.   |   |
|    |  |   |
|    | Average Global Accurancy = 0.7983, Loss = 0.67, Iter = 139. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139.   |   |
|    |  |   |
|    | Average Global Accurancy = 0.8103, Loss = 0.68. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 139. Finished training.  | ĺ |

# 最終 acc 的前十筆及後十筆輸出 (輸出筆數過多)

| _ | ↑ ↓ © 目 ‡ [ ]   ↑ ↓ © 目 ‡ [ ]  |
|---|--|
| ) | isport os or. chdir ("content/drive/MpOrive/MFL")   python main.py   |
| 3 | Sumany of normal model training process: Dataset: CIFARIO-alphalOO.O-ratiol.O-usersIO model: rementIB            |
|   | Devise: cuda Number of global rounds: 150 Number of local rounds: 10 Clents' learning rate: 0.1                  |
|   |  |
|   | [Start training iteration 0]   |
|   | Users in total: 10 Number of users per round / total users: 10 / 10 Finished creating FedAvg serves.             |
|   | ——————————————————————————————————————   |
|   | Average Global Accurancy = 0.1068, Loss = 2.30. Best Global Accurancy = 0.1068, Loss = 2.30, Iter = 0.           |
|   | Round number: 1  |
|   | Average Global Accurancy = 0.1039, Loss = 19565.48.  Best Global Accurancy = 0.1039, Loss = 2.30, Iter = 0.      |
|   | Round number: 2  |
|   | Average Global Accurancy = 0.1367, Loss = 5839.31.  Best Global Accurancy = 0.1367, Loss = 5839.31, Ites = 2.    |
|   | Round number: 3  |
|   | Average Global Accurancy = 0.2087, Loss = 8.96. Best Global Accurancy = 0.2087, Loss = 8.96, Iter = 3.           |
|   | Round number: 4  |
|   | Average Global Accurancy = 0.2338, Loss = 2.02.  Best Global Accurancy = 0.2338, Loss = 2.02, Iter = 4.          |
|   | Round number: 5  |
|   | Average Global Accurancy = 0.2800, Loss = 2.11. Best Global Accurancy = 0.2800, Loss = 2.11, Iter = 5.           |
|   |  |
|   | Average Global Accurancy = 0.3297, Loss = 1.81.  Best Global Accurancy = 0.3297, Loss = 1.81, Iter = 6.          |
|   | Round number: 7  Average Global Accurancy = 0.3545, Loss = 1.74.   |
|   | Average 0.0001 Accurancy = 0.3040, Loss = 1.74, Tex = 7.  Best Global Accurancy = 0.3046, Loss = 1.74, Itex = 7. |
|   |  |
|   | Average Global Accurancy = 0.3468, Loss = 1.84. Best Global Accurancy = 0.3845, Loss = 1.74, Iter = 7.           |
|   | ——————————————————————————————————————   |
|   | Average Clobal Accurancy = 0.4054, Loss = 1.60.  Best Global Accurancy = 0.4054, Loss = 1.60, Iter = 9.          |
|   |  |
|   | Average Global Accurancy = 0.4345, Loss = 1.54.  Best Global Accurancy = 0.4345, Loss = 1.54, Iter = 10.         |

| 0 | Round number: 140   |     | ↑ ↓ © <b>目 ‡</b> 🖟 🖺 : |
|---|---|-----|------------------------|
|   | Average Global Accurancy = 0.8074, Loss = 0.72. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 1                           | 39. |                        |
|   | Round number: 141   |     |                        |
|   | Average Global Accurancy = $0.8033$ , Loss = $0.73$ .<br>Best Global Accurancy = $0.8153$ , Loss = $0.67$ , Iter = $1$          | 39. |                        |
|   | Round number: 142   |     |                        |
|   | Average Global Accurancy = $0.8095$ , Loss = $0.70$ .<br>Best Global Accurancy = $0.8153$ , Loss = $0.67$ , Item = $1$          | 39. |                        |
|   | Round number: 143   |     |                        |
|   | Average Global Accurancy = $0.8140$ , Loss = $0.68$ . Best Global Accurancy = $0.8153$ , Loss = $0.67$ , Iter = $1$             | 39. |                        |
|   | Round number: 144   |     |                        |
|   | Average Global Accurancy = $0.7985$ , Loss = $0.74$ .<br>Best Global Accurancy = $0.8153$ , Loss = $0.67$ , Item = $1$          | 39. |                        |
|   | Round number: 145   |     |                        |
|   | Average Global Accurancy = 0.8099, Loss = 0.69. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 1                           | 39. |                        |
|   | Round number: 140   |     |                        |
|   | Average Global Accurancy = $0.8041$ , Loss = $0.71$ .<br>Best Global Accurancy = $0.8153$ , Loss = $0.67$ , Iter = $10.000$     | 99. |                        |
|   |   |     |                        |
|   | Average Global Accurancy = 0.8080, Loss = 0.69. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 13                          | 99. |                        |
|   |   |     |                        |
|   | Average Global Accurancy = 0.7983, Loss = 0.72. Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 1:                          | 39. |                        |
|   |   |     |                        |
|   | Average Global Accurancy = 0.8103, Loss = 0.68.<br>Best Global Accurancy = 0.8153, Loss = 0.67, Iter = 1:<br>Finished training. | 39. | Î                      |

這次作業我學習到聯邦平均算法的實作方法,透過實作 selected\_users(), aggregate\_parameters(), set\_parameters()這三個 function,認識此算法中 Server 端是如何與 client 端互動,透過隨機挑選固定數量用戶上傳模型修改全局模型的參數,再由用戶下載全局模型參數,來避免資料外洩的問題。