Data Science – HW4

Student ID: 109062211 Name: 張惇媛

1. Code

userbase.py

```
def set_parameters(self, model, beta=1):
    # replace user's local model with global model
    # user's model parameters = beta*global model parameters + (1-beta)*user's model parameters
    # state_dict - get model parameters
    # load_state_dict - update model parameters
    user_model_dict = self.model.state_dict()
    for key, parameters in user_model_dict.items():
        user_model_dict[key] = beta*model.state_dict()[key] \
                               + (1-beta)*self.model.state_dict()[key]
    self.model.load_state_dict(user_model_dict)
```

serverbase.py

```
def aggregate_parameters(self):
    n_samples = []
    states = []
    # pick selected users' number of samples and model parameters
    for user in self.selected_users:
        n_samples.append(user.train_samples)
        states.append(user.model.state_dict())
    # calculate weight for each selected user
    N = sum(n_samples)
    weights = [n/N for n in n_samples]
    # weighted sum all selected users' model parameters and update global model
    global_model_dict = self.model.state_dict()
    for key, parameters in global_model_dict.items():
        for i in range(len(self.selected_users)):
            if i == 0:
                global_model_dict[key] = weights[i] * states[i][key]
            else:
                global_model_dict[key] += weights[i] * states[i][key]
    self.model.load_state_dict(global_model_dict)
```

```
def select_users(self, round, num_users):
    import random
    # guarantee num_users <= len(self.users)
    num_users = min(num_users, len(self.users))
    # random selection
    selected_idx = random.sample(range(len(self.users)), num_users)

    selected_objects = []
    for i in range(num_users):
        selected_objects.append(self.users[selected_idx[i]])
    return selected_objects
```

2. Discussion
    a. Data distribution
        - alpha = 0.1
            - user 資料分布

```
TRAIN #sample by user: [7517, 5817, 4245, 2533, 4726, 5121, 5122, 1664, 8721, 4534]
7517 samples in total
c=2,n=1015| c=4,n=2966| c=5,n=4| c=6,n=47| c=8,n=3485|
5 Labels/ 7517 Number of training samples for user [0]:
5817 samples in total
c=2,n=240| c=5,n=4559| c=7,n=24| c=8,n=994|
4 Labels/ 5817 Number of training samples for user [1]:
4245 samples in total
c=2,n=2539| c=3,n=412| c=5,n=390| c=8,n=492| c=9,n=412|
5 Labels/ 4245 Number of training samples for user [2]:
2533 samples in total
c=1,n=817| c=2,n=58| c=3,n=52| c=7,n=1495| c=9,n=111|
5 Labels/ 2533 Number of training samples for user [3]:
4726 samples in total
c=1,n=284| c=2,n=209| c=6,n=753| c=7,n=3479| c=9,n=1|
5 Labels/ 4726 Number of training samples for user [4]:
5121 samples in total
c=0,n=4973| c=2,n=100| c=3,n=9| c=5,n=29| c=6,n=3| c=8,n=7|
6 Labels/ 5121 Number of training samples for user [5]:
5122 samples in total
c=4,n=2018| c=6,n=3102| c=7,n=1| c=9,n=1|
4 Labels/ 5122 Number of training samples for user [6]:
1664 samples in total
c=1,n=15| c=3,n=1489| c=4,n=15| c=6,n=1| c=8,n=21| c=9,n=123|
6 Labels/ 1664 Number of training samples for user [7]:
8721 samples in total
c=2,n=838| c=3,n=3034| c=6,n=498| c=9,n=4351|
4 Labels/ 8721 Number of training samples for user [8]:
4534 samples in total
c=0,n=27| c=1,n=3884| c=2,n=1| c=3,n=4| c=4,n=1| c=5,n=18| c=6,n=596| c=7,n=1| c=8,n=1| c=9,n=1|
10 Labels/ 4534 Number of training samples for user [9]:

TEST #sample by user: [500, 400, 500, 500, 500, 600, 400, 600, 400, 1000]
```

            - global model accuracy

```
-------------Round number: 149 -------------


Average Global Accurancy = 0.4315, Loss = 1.69.
Best Global Accurancy = 0.4780, Loss = 1.67, Iter = 148.
Finished training.
```

        - alpha = 50.0
            - user 資料分布

```
TRAIN #sample by user: [4901, 5481, 4907, 4629, 5449, 4889, 4770, 4601, 5073, 5300]
4901 samples in total
c=0,n=504| c=1,n=489| c=2,n=491| c=3,n=510| c=4,n=415| c=5,n=500| c=6,n=489| c=7,n=606| c=8,n=447| c=9,n=450|
10 Labels/ 4901 Number of training samples for user [0]:
5481 samples in total
c=0,n=483| c=1,n=538| c=2,n=696| c=3,n=386| c=4,n=454| c=5,n=568| c=6,n=684| c=7,n=528| c=8,n=573| c=9,n=571|
10 Labels/ 5481 Number of training samples for user [1]:
4907 samples in total
c=0,n=554| c=1,n=456| c=2,n=452| c=3,n=513| c=4,n=458| c=5,n=439| c=6,n=552| c=7,n=482| c=8,n=576| c=9,n=425|
10 Labels/ 4907 Number of training samples for user [2]:
4629 samples in total
c=0,n=453| c=1,n=457| c=2,n=444| c=3,n=525| c=4,n=486| c=5,n=471| c=6,n=485| c=7,n=482| c=8,n=378| c=9,n=448|
10 Labels/ 4629 Number of training samples for user [3]:
5449 samples in total
c=0,n=502| c=1,n=569| c=2,n=544| c=3,n=483| c=4,n=482| c=5,n=599| c=6,n=510| c=7,n=547| c=8,n=661| c=9,n=552|
10 Labels/ 5449 Number of training samples for user [4]:
4889 samples in total
c=0,n=509| c=1,n=502| c=2,n=457| c=3,n=416| c=4,n=560| c=5,n=464| c=6,n=472| c=7,n=415| c=8,n=589| c=9,n=505|
10 Labels/ 4889 Number of training samples for user [5]:
4770 samples in total
c=0,n=522| c=1,n=484| c=2,n=430| c=3,n=564| c=4,n=505| c=5,n=454| c=6,n=399| c=7,n=429| c=8,n=405| c=9,n=578|
10 Labels/ 4770 Number of training samples for user [6]:
4601 samples in total
c=0,n=477| c=1,n=460| c=2,n=479| c=3,n=475| c=4,n=526| c=5,n=449| c=6,n=479| c=7,n=367| c=8,n=494| c=9,n=395|
10 Labels/ 4601 Number of training samples for user [7]:
5073 samples in total
c=0,n=450| c=1,n=525| c=2,n=457| c=3,n=582| c=4,n=507| c=5,n=551| c=6,n=466| c=7,n=553| c=8,n=392| c=9,n=590|
10 Labels/ 5073 Number of training samples for user [8]:
5300 samples in total
c=0,n=546| c=1,n=520| c=2,n=550| c=3,n=546| c=4,n=607| c=5,n=505| c=6,n=464| c=7,n=591| c=8,n=485| c=9,n=486|
10 Labels/ 5300 Number of training samples for user [9]:

TEST #sample by user: [1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
```

- global model accuracy

```
-------------Round number: 149 -------------


Average Global Accurancy = 0.8069, Loss = 0.74.
Best Global Accurancy = 0.8116, Loss = 0.69, Iter = 144.
Finished training.
```

- Summary

  alpha 較小時，各 user 取得的 samples 數量較不平均，且每個 user 中的各 label 中 samples 數量也不平均，出現 imbalanced data 的問題；而 alpha 較大時，user 的資料分布平均。因此，對於 global model accuracy 來說，alpha 較大的 dataset 在 model 的 performance 較好。

b. Number of users in a round

- num_users = 2
  - global model accuracy

```
-------------Round number: 149 -------------


Average Global Accurancy = 0.7326, Loss = 0.77.
Best Global Accurancy = 0.7456, Loss = 0.73, Iter = 148.
Finished training.
```

  - 模型收斂速度

```
-------------Round number: 49 -------------


Average Global Accurancy = 0.5051, Loss = 1.35.
Best Global Accurancy = 0.5057, Loss = 1.36, Iter = 46.

-------------Round number: 99 -------------


Average Global Accurancy = 0.6521, Loss = 0.97.
Best Global Accurancy = 0.6684, Loss = 0.94, Iter = 95.
```

- num_users = 10
  - global model accuracy

```
-------------Round number: 149 -------------


Average Global Accurancy = 0.8201, Loss = 0.70.
Best Global Accurancy = 0.8201, Loss = 0.70, Iter = 149.
Finished training.
```

■ 模型收斂速度

```
------------Round number: 49 -------------


Average Global Accurancy = 0.6964, Loss = 0.86.
Best Global Accurancy = 0.7044, Loss = 0.83, Iter = 48.

------------Round number: 99 -------------


Average Global Accurancy = 0.8059, Loss = 0.65.
Best Global Accurancy = 0.8087, Loss = 0.62, Iter = 95.
```

- Summary
  num_users 較小時，在每次的更新，global model 從 user 端能接收到的資訊較少，較無法反應出整體 user 端的資訊，導致 global model accuracy 較低、收斂速度較慢。

3. Output

```
------------Round number: 149 -------------


Average Global Accurancy = 0.8143, Loss = 0.70.
Best Global Accurancy = 0.8217, Loss = 0.64, Iter = 123.
Finished training.
```

4. Summary

在這次作業中，實作、分析的主題是為聯邦學習(Federated learning)，在實作的部分中，我學習到了如何進行聯邦學習模型參數的合併，包含 user 中合併 local model 與 global model 以及 server 中合併 selected users' model。而在分析的部分，實際比較不同情形下，參數選擇對於正確性的差異，我學習到了影響聯邦學習模型正確性的因素，能夠運用於實務上，作為模型調整的方向。