Dynamic Programming
IEMS469 - 23 Fall
Assignment 3 - Part I

Dinglin Xia

November 15th, 2023

1 Implement FedAvg for Federated EMNIST dataset

1.1 Data Description

We plot the histogram of the whole training data in figure 1 (upper). At the same time, we sample 5 clients out of the 100 and plot their individual training dataset. From the plot, we may witness that each customer's distribution is very different (e.g., some clients have more "numbers" while some have more letters).

1.2 Training Result

We first trained data with parameter $C = 10\%$ (i.e., in each communication round, we only sample 10% of all clients to make an update). From the plot we see both training accuracy and validation accuracy is steadily increasing to the communication rounds. We also tried different combinations of hyperparameters: $C \in [10\%, 5\%, 4\%, 2\%]$, and $E \in [1, 5]$ (i.e., update epochs for each client in each communication round). The full experiment record is shown in Table 1 (where we also show the result of centralized learning as a baseline). More training plots are attached in the Appendix. Together with the table 1 and the training plot 4 we see a general pattern that higher C and E can lead to a faster convergence rate. However, they may not lead to a better final convergence result (might due to overfitting).

1.3 Testing Result

From the validation result shown in Table 1, we see the best combination of hyperparameters is $C = 5\%, E = 1$. We test the model on the dataset and the corresponding training result is:

Testing Loss: 0.714

Testing Accuracy: 80.9%

This is slightly worse than the validation results.

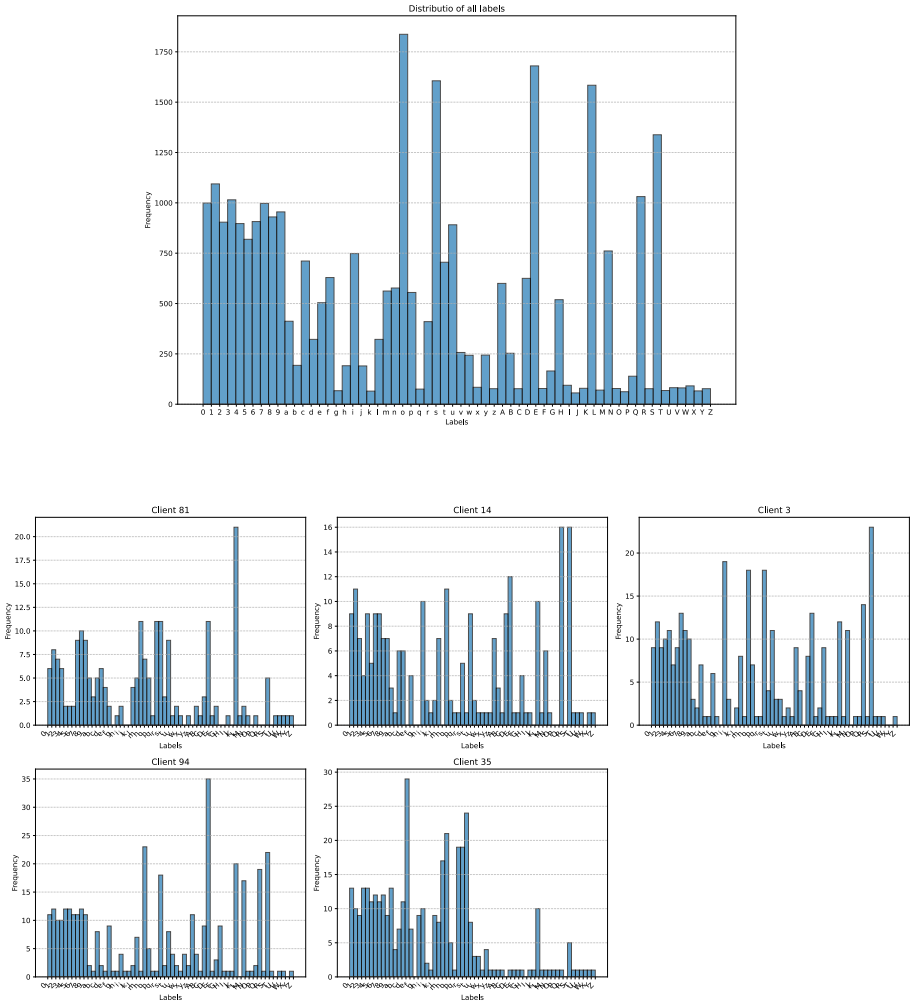


Figure 1: The distribution of all training data (upper) and 5 sample clients. An unbalanced distribution among different clients can be witnessed.

Hyperparameters	Training Loss	Training Acc.	Validation Loss	Validation Acc.
Centralized	0.119	95.6%	1.285	80.3%
C = 10%, E = 1	0.510	87.2%	0.725	78.9%
C = 10%, E = 5	0.057	97.6%	0.996	82.9%
C = 5%, E = 1	0.229	89.6%	**0.728**	**85.0%**
C = 5%, E = 5	0.121	96.4%	1.117	81.2%
C = 4%, E = 1	0.430	86.5%	0.734	78.2%
C = 4%, E = 5	0.010	96.8%	0.850	82.5%
C = 2%, E = 1	0.491	88.3%	0.579	84.9%
C = 2%, E = 5	0.210	89.6%	1.537	77.3%

Table 1: Training/Validation Loss/Accuracy of different combinations of hyperparameters. The best combination of hyperparameters is $C = 5\%, E = 1$.

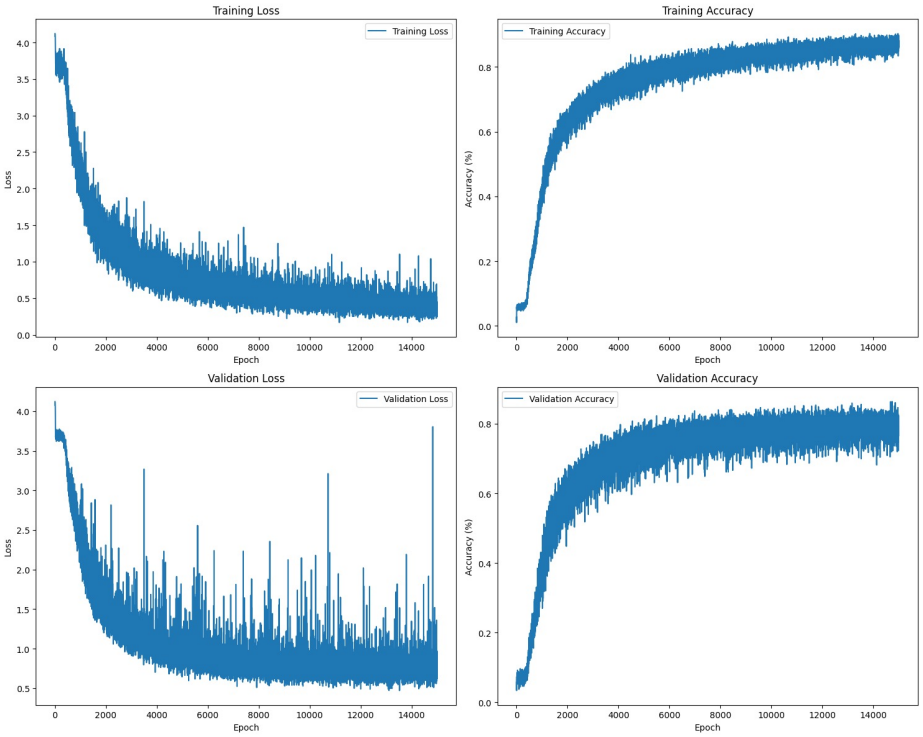


Figure 2: The training plot of FedAvg with hyperparameters $C = 10\%, E = 1$. We see both training and validation accuracy are steadily increasing, though validation shows higher variance and finally converges to a point slightly lower than training results.

2 Parallel Clients

In this part, we use Ray to parallelize the training process. In each communication round, we only sample 4 out of 100 clients for updates. The training plot is shown in the figure 3. The ultimate performance of the trained model on test data achieves the following loss and accuracy in Table 2. The corresponding Test loss and accuracy are:

- **Testing Loss**: 0.822
- **Testing Accuracy**: 78.2%

This is slightly worse than the validation results.

Hyperparameters	Training Loss	Training Acc.	Validation Loss	Validation Acc.
Parallel Training	0.153	87%	0.902	77.2%

Table 2: Training/Validation Loss/Accuracy of parallel training.

3 Appendix

This part gives training plots of more combinations of hyperparameters.

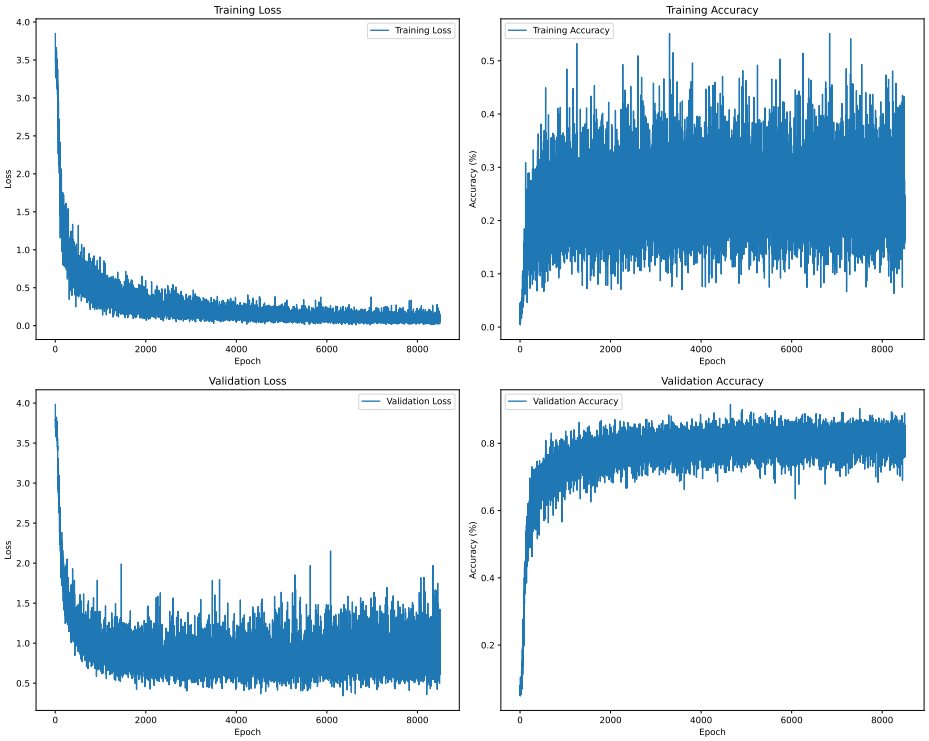
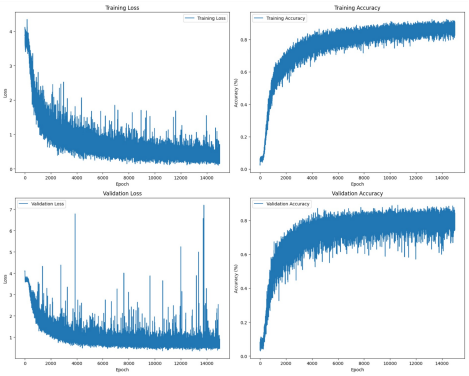
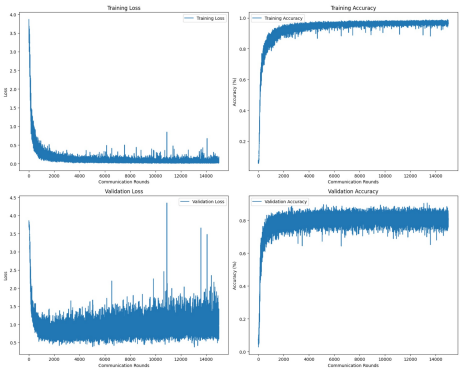


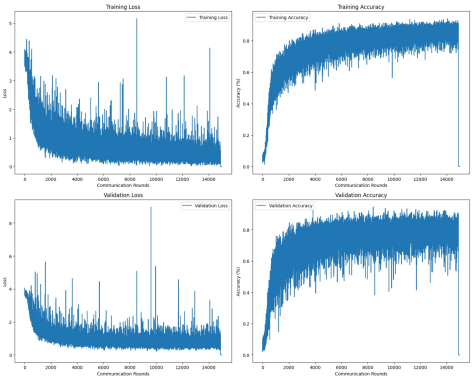
Figure 3: The training plot of FedAvg with hyperparameters $C = 4\%, E = 1$ (parallel training with implementation by Ray).



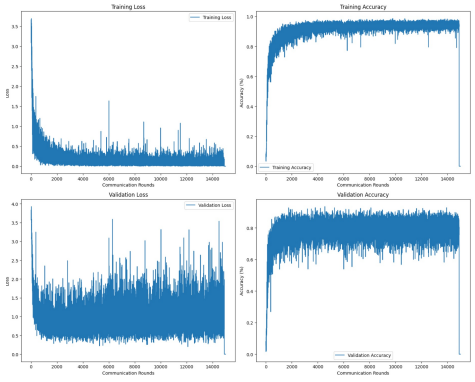
(a) C = 0.05, E = 1



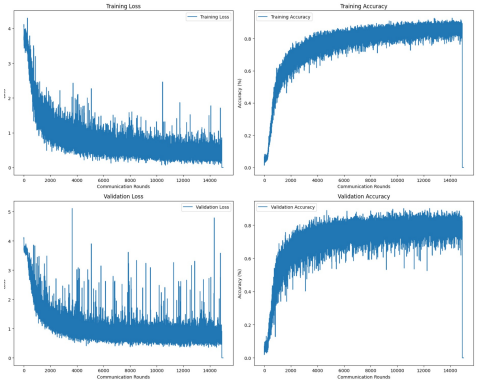
(a) C = 0.05, E = 5



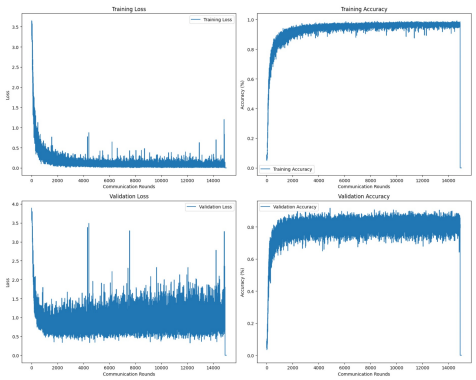
(c) C = 0.02, E = 1



(d) C = 0.02, E = 5



(e) C = 0.04, E = 1



(f) C = 0.04, E = 1

Figure 4: The training plot of FedAvg with several different hyperparameter settings.