**Task 2 a)**

We ran a simulation for federated learning with 3 clients, we used a simple CNN implementation consisting of 4 convolution layers and one full connected layer.

We created a load dataset function to load the ChestMnist dataset in a way to create a list of *trainloaders* and *valloaders* with *length = number of clients*, the function also returns the testloader which is used at the server-side to evaluate the training

The metrics were aggregated using a simple weighted average function.

The evaluation was done at the client side with a small portion of the validation dataset (referred to as distributed in the table) and on the server-side with the test dataset (referred to as centralized in the table)

We tried this setup and documented the results while varying the batch size and learning rate as follows

* Batch size = 32 and learning rate = 1e-3
* Batch size = 8 and learning rate = 1e-3
* Batch size = 64 and learning rate = 1e-3
* Batch size = 32 and learning rate = 1e-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Round | Batch\_size=32  L\_R = 1e-3 | Batch\_size=8  L\_R = 1e-3 | Batch\_size=64  L\_R = 1e-3 | Batch\_size=32  L\_R = 1e-1 |
| Loss ↓ Distributed | 1 | 0.052 | 0.211 | **0.0261** | 0.0504 |
| 2 | 0.048 | 0.202 | **0.0243** | 0.0504 |
| 3 | 0.047 | 0.195 | **0.0241** | 0.0506 |
| Loss ↓ Centralized | *0* | *0.060* | *0.241* | *0.030* | *0.061* |
| 1 | 0.054 | 0.219 | **0.026** | 0.0524 |
| 2 | 0.050 | 0.210 | **0.025** | 0.0524 |
| 3 | 0.049 | 0.202 | **0.024** | 0.0525 |
| Accuracy ↑ Distributed | 1 | 12.59 | 12.594 | **12.6** | 12.59 |
| 2 | 13.96 | 12.594 | **20.3** | 12.59 |
| 3 | 24.28 | 15.874 | **24.8** | 12.59 |
| Accuracy ↑ Centralized | *0* | *3.25* | *12.035* | *0.51* | *2.23* |
| 1 | 12.03 | 12.035 | **12.04** | 12.035 |
| 2 | 13.40 | 12.035 | **19.74** | 12.035 |
| 3 | 23.81 | 15.517 | **24.54** | 12.035 |

Results, as expected, are better with larger batch size, the model is capable of converging faster, Comparing batch size of 64 batch size of 8, the accuracy with the first one was capable to jump from 12% to 19.8% in one round, while with batch size 8 accuracy remained 12% for 2 rounds.

The same is true for the learning rate with a high learning rate (last column); the model didn’t converge at all, and as shown, loss and accuracy are not changing round after the other.