联邦学习实验报告

廖凡　521021910346

2024 年 5 月 12 日

目录

1 实验设计

训练网络模型是由两个卷积层构成，它的实现比较简单，且训练成本较低，训练得较快，但是效果会不是那么出色，但是本实验的主要目的是实现联邦学习算法，对于正确率容忍度较大，所以也是可以接受。

实现联邦学习的主要算法是 FedAvg，它是将所有客户端模型参数相加后平均，实现起来较容易，伪代码如下，

Algorithm 1: FedAvg

```
global_params = sum(client_params);
global_params /= N;
return global_params;
```

实现 server-client socket 编程时，采用 TCP 协议，服务端建立连接后时刻监听端口并接收来自客户端的模型参数，并需要向 20 个客户端发送全局模型参数。而客户端需要接受来自服务端的通信并向服务端发送模型。

1.1 Stage1

有 N 个客户端的联邦学习系统框架如图5所示。服务端和客户端以文件夹的方式共享 pth 文件，共享文件夹存放着 global model 和 20 个 client model，客户端在每个训练回合结束后，客户端将其本地模型保存在共享文件夹中，当所有本地模型训练好时，服务器进程加载所有客户端的本地模型并使用 FedAVG 算法聚合参数。服务器进程完成聚合并保存更新的全局模型时，客户端进程将新更新的全局模型加载为新的本地模型，并开始新一轮的训练。

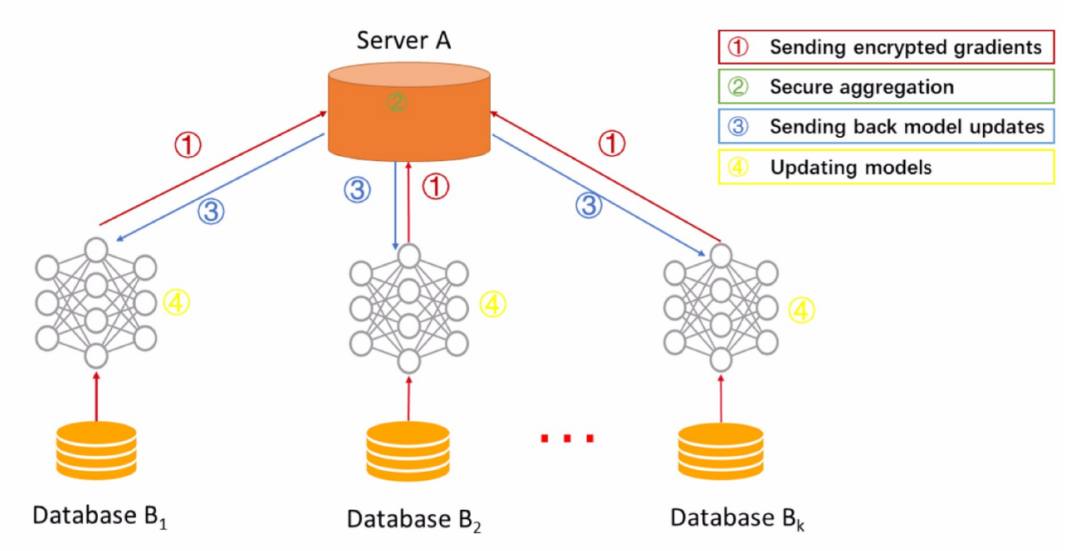

图 1: FL 系统框架

1.2 Stage2

与 Stage1 框架极为类似，除了从 20 个客户端中只随机选择 16 个客户端进行更新，其他更新训练步骤和 Stage1 一模一样。

1.3 Stage3

有 N 个客户端的联邦学习系统框架如图2所示。服务端和客户端使用 socket 方式同时传输模型参数。global model 和 20 个 client model 各自由服务端和 20 个客户端私有，客户端在每个训练回合结束后，客户端将其本地模型通过 socket 发送给服务端，当服务端收到所有客户端的本地模型后，使用 FedAVG 算法聚合参数更新全局模型并通过 socket 分发到每个客户端。客户端收到模型后开始新一轮的训练。

2 实验结果

2.1 Stage1

在经过 20 个客户端的 10 轮，每轮 30 个 epoch 训练后，最终服务端在测试集上的正确率如下所示，

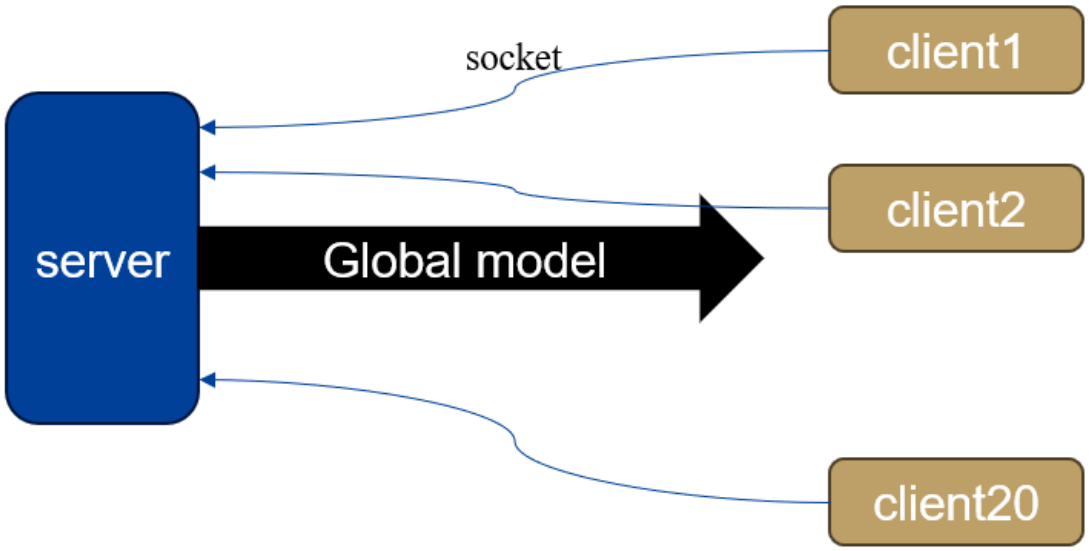

图 2: FL 系统框架

2.2 Stage2

在 Stage2 中，我们随机选择 16 个客户端参与到全局模型更新中，同样的，客户端与服务端之间更新 10 轮，每轮 30 个 epoch 训练，最终服务端在测试集上的正确率如下所示，可以看到 stage1 和 stage2 之间的正确率差别不大，并且收敛速度也十分相似，收敛到 0.56 附近，二者性能差距不大。

2.3 Stage3

Stage3 实现了通过 socket 通信完成客户端与服务端的模型交换，由于本身电脑的原因，最多可以实现 6 个客户端同时通信，因为内存仅能装载 6 个训练集的数据，所以设定参与训练的客户端数为 6。同样的经过 10 轮，每轮 30 个 epoch 训练，最终服务端在测试集上的正确率如下所示，最终的正确率也是收敛在 0.6 左右。

3 总结

最终我们成功实现了联邦学习的三种实现方法，并且在 CIFAR10 数据集上的正确率在 0.6 左右，考虑到模型的简单和较少的训练次数，该结果也可以接受。

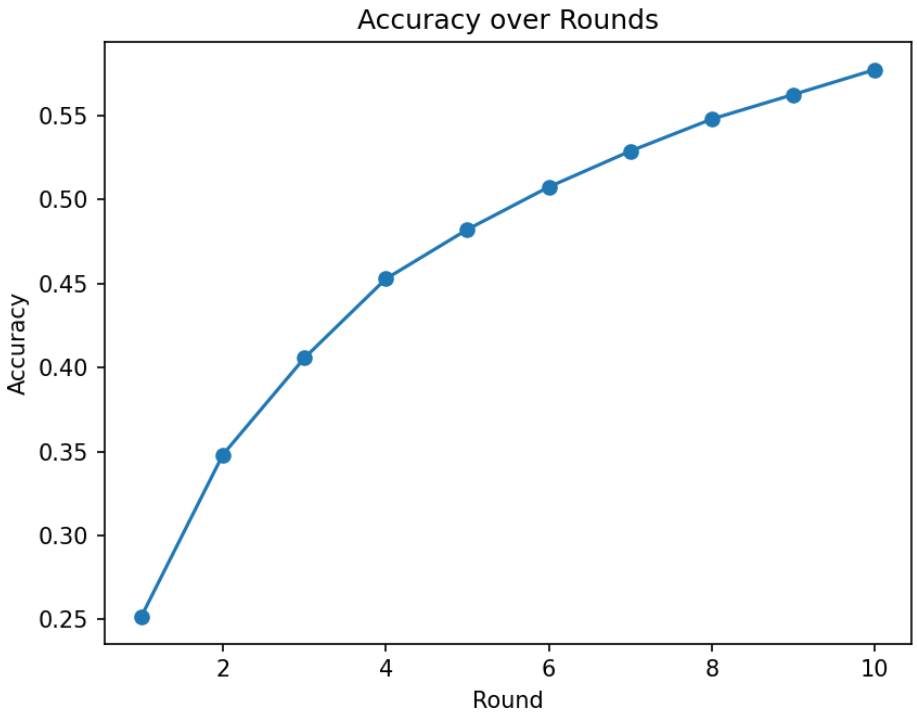

图 3: Stage1

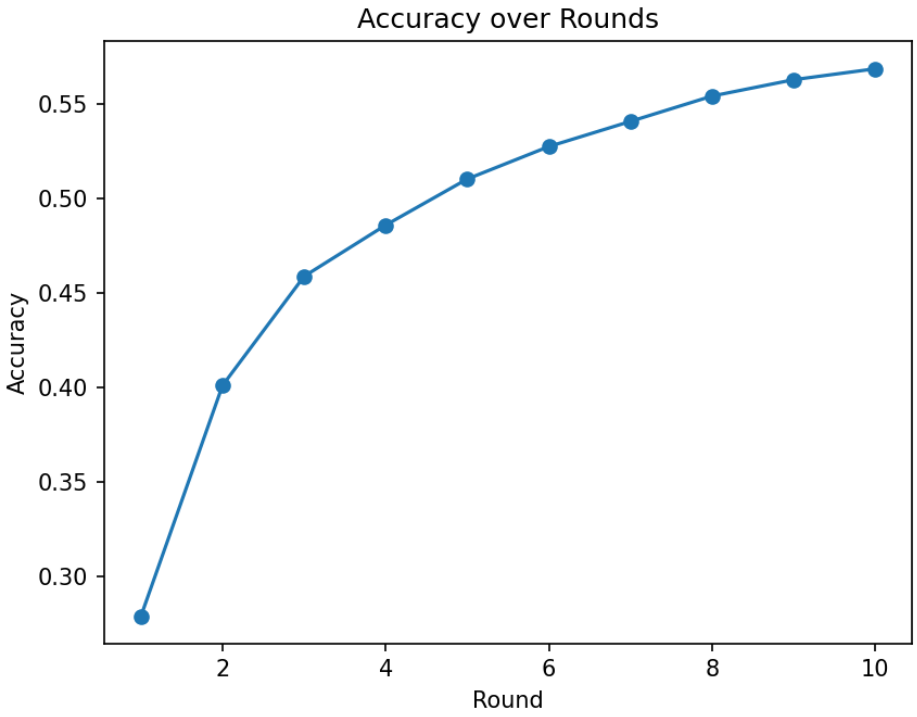

图 4: stage2

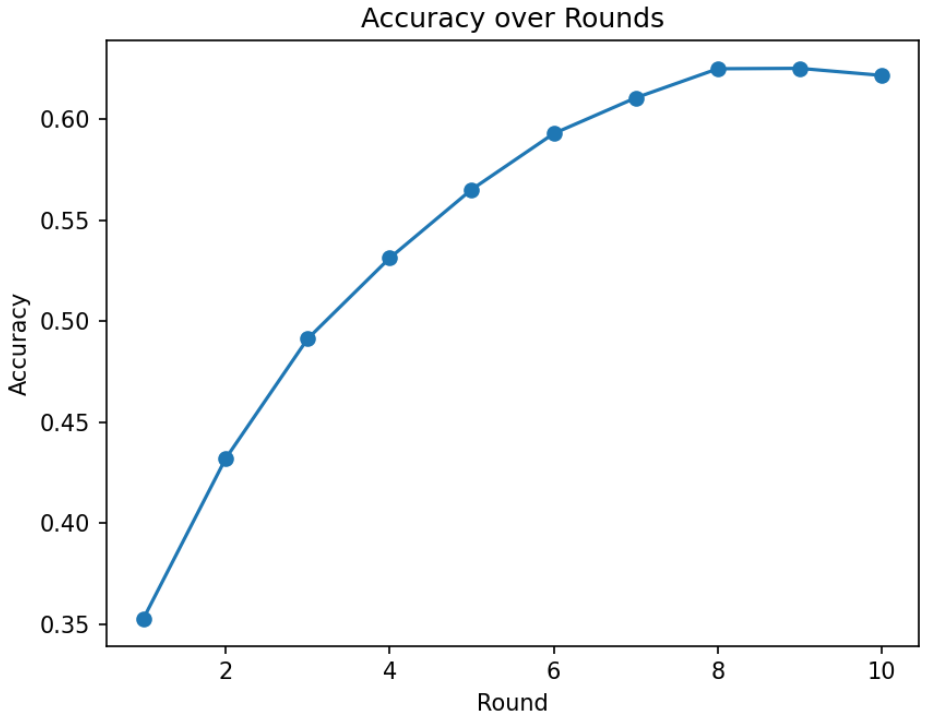
Accuracy over Rounds
Accuracy
0.60
0.55
0.50
0.45
0.40
0.35
2
4
6
8
10
Round

图 5: Stage3