1. **搭建联邦学习系统，复现FedAvg算法。要求：**
	1. 参考论文“Communication-Efficient Learning of Deep Networks from Decentralized Data”，根据其中的Algorithm 1实现FederatedAveraging算法。各个客户机的训练用循环实现即可。
	2. 推荐使用python+pytorch进行编程，其中python与pytorch推荐使用conda（<https://docs.conda.io/en/latest/miniconda.html>）进行安装。
	3. 服务器端和客户端分别写到两个文件中，如client.py和server.py。
	4. 服务器与客户端之间的传输可直接通过内存完成。
2. **手动实现Non-IID数据集（pathological and real-world）。要求**
	1. pathological non-IID 数据集划分方式，可参考上述论文；real-world non-IID数据集划分方式可以参考<https://towardsdatascience.com/preserving-data-privacy-in-deep-learning-part-2-6c2e9494398b>
	2. 建议为每个client上的数据集生成单个文件，可以减少每次启动代码时的I/O时间。
3. **探索不同数量的客户机对算法准确率的影响。要求：**
	1. 例如分别设置10， 20， 100， 1000个客户机，观察结果并进行分析。实际部署的情况下，手机用户的数量甚至更多，学生也可以模拟实际环境进行实验。
4. **使用多进程模拟多客户机并行，并处理好通讯问题。要求：**
	1. 为了进一步实现真实的联邦学习系统，需要实现客户机并行计算。通过诸如为不同客户机分配数量不同的数据集等方法，使得客户机在计算能力和通讯能力上有所差异，模拟实际部署情况。
5. **完成报告**