关于 FedAvg 在 Non-IID 数据上的收敛性

摘要

随着数据隐私保护需求的增加和物联网的迅速发展，联邦学习能够让多个客户端在不泄露自己数据隐私的前提下协同训练机器学习模型，在数据安全和隐私保护方面展现出了独特的优势。同时，在联邦学习的训练过程中，FedAvg 是其中最常用的算法，因此，本文就该算法进行研究，研究该算法的收敛性与每轮本地局部更新次数 E、参与设备数 K 以及采样方案之间的关系。

关键词：联邦学习；FedAvg；收敛性

1 引言

随着互联网技术的迅速发展和人民生活水平的不断提高，手机、平板电脑以及笔记本电脑等电子设备数量激增 [1]。随着电子设备的智能化和高效化，人们可以运用电子设备进行各种各样的活动，因此电子设备每时每刻都会产生海量的数据。而数据量是人工智能发展的关键因素之一。人工智能模型需要大量的数据来训练，数据量的大小直接影响到人工智能技术的性能和发展，数据量越大，训练出来的模型越能反映真实情况，才能更好地进行预测。例如，在自然语言处理任务中，文本语料库越多，训练出来语言模型能更准确地理解和生成语言 [2]。在图像识别任务中，图片数据集越多，分类模型识别的准确率越高 [7]。因此，在海量数据的支撑下，人工智能领域能够快速地发展 [4]。

然而，人工智能领域也存在一些亟待解决的问题，例如，如何整合数据来进行机器学习模型训练。一方面是研究者们不能直接从用户设备上获取数据。用户设备的数据涉及到用户的个人信息，例如身份信息、位置和喜好等。这些数据的泄露可能会导致严重的后果，例如身份被盗用导致财产损失。国内外政府机构出台了越来越多的法律法规来保护人们的数据安全和隐私权，国外的立法有欧盟的《通用数据保护条例》 [5]，国内的有《中华人民共和国个人信息保护法》。另一方面是企业和机构之间的数据不流通。由于行业竞争、法律规定以及企业名声等因素，不同企业与机构难以直接共享数据来训练出性能优异的机器学习模型。这两方面的原因导致分布在小型用户设备、企业和机构中的数据无法进行整合，造成了“数据孤岛”的问题。

McMahan 团队在 2017 年提出了联邦学习 [3] 这一新的机器学习范式。集中式机器学习先收集用户设备的数据，再由服务器上进行模型的训练。联邦学习不收集用户设备的数据，而是聚合用户设备训练的模型。联邦学习的模型训练流程为：第一步，用户设备使用自身数据来训练模型。第二步，用户设备将训练好的模型发送给服务器。第三步，服务器对多个用户设备发送的模型进行聚合。第四步，服务器将聚合后的模型下发给用户设备。重复执行第一

步到第四步，直到模型收敛或达到指定迭代轮数。从训练流程可以看出，用户设备与服务器只进行模型参数的传输，每个用户的数据一直都保留在用户自己的设备中，并没有与其他设备进行数据共享，从而保证了本地数据的隐私性和安全性。

然而对于联邦学习来说，其中最广泛使用的算法便是联邦平均算法（FedAvg），因此对于该算法进行进一步的分析，探究其收敛性与各方面参数的关系，同时由于实际上的数据大部分都是非独立同分布的（Non-IID），因此本文分析时也会对数据的分布进行划分。

2 相关工作

2.1 联邦学习

联邦学习（Federated Learning，FL）是一种新型分布式机器学习范式。相较于一般的分布式机器学习，联邦学习中客户端的数据是自己生成和拥有的。不同客户端之间、客户端与服务器之间不进行原始数据的传输，从而保证了每个客户端的数据隐私安全。联邦学习架构 [6] 可以分为两种，分别是客户端-服务器架构和对等网络架构。客户端-服务器架构包含多个客户端和一个服务器。服务器先接收多个客户端的模型更新信息（客户端训练出来的模型与训练前模型之间的差值），然后将模型更新信息进行聚合得到全局模型更新信息，最后将全局模型更新信息下发给客户端。客户端的作用是训练模型、发送模型更新信息给服务器、接受服务器下发的全局模型更新信息和更新模型。服务器和客户端重复以上过程，直至模型收敛。对等网络架构仅包含多个客户端。客户端训练模型后将模型更新信息发送给其他客户端，并且接收其他客户端的模型更新信息，最后客户端将所有模型更新信息聚合为一个全局模型更新信息，并使用它来更新模型。重复以上过程，直至模型收敛。当前的研究更多地关注于客户端-服务器架构，因此本文也是基于客户端-服务器架构进行研究。

2.2 国内外研究现状

McMahan 团队在提出联邦学习框架 [3] 时，还提出了联邦平均算法（FedAvg），该算法让用户设备进行多轮模型训练后再与中心设备进行通信，以此来减少通信次数。FedAvg 在数据 IID 时表现良好，但在数据 Non-IID 时，该算法的性能表现一般，并且 McMahan 团队并未给出 FedAvg 的收敛性证明。大部分的证明基于以下两个假设（1）数据是 IID 的（2）所有设备都是活动的，在这基础上已经做出了很多努力来开发联邦学习算法的收敛保证。Khaled (2019)，Yu (2019)；Wang (2019) 提出了后一种假设，而 Zhou 和 Cong (2017)；Stich (2018)；Wang 和 Joshi (2018)；Woodworth (2018) 做出了这两种假设。这两个假设并不符合大部分实际情况，实际上数据更多的情况下是 Non-IID 的，并且没法保证所有的设备都能按时参与模型的训练，然而长时间的等待未响应的设备显然也是不合理的。先前 Sahu (2018) 提出的 Fedprox 算法不需要上述两个假设，并在添加的近端项消失时将 FedAvg 作为特例。然而，他们的理论并没有涵盖 FedAvg。因此本文分析了 FedAvg 算法在 Non-IID 数据且部分设备参与的情况下的收敛性的分析证明。

2.3 数据集介绍

MNIST 数据集是一个用来训练各种图像处理系统的二进制图像数据集，广泛应用于机器学习中的训练和测试。作为一个入门级的计算机视觉数据集，发布 20 多年来，它已经被无数机器学习入门者 “咀嚼” 千万遍，是最受欢迎的深度学习数据集之一。该数据集的论文想要证明在模式识别问题上，基于 CNN 的方法可以取代之前的基于手工特征的方法，所以作者创建了一个手写数字的数据集，以手写数字识别作为例子证明 CNN 在模式识别问题上的优越性。MNIST 数据集是从 NIST 的两个手写数字数据集:Special Database 3 和 Special Database 1 中分别取出部分图像，并经过一些图像处理后得到的。MNIST 数据集共有 70000 张图像，其中训练集 60000 张，测试集 10000 张。所有图像都是 28×28 的灰度图像，每张图像包含一个手写数字。而我们采用的 MNIST 数据集是 LeCun（1998）以 Non-IID 的方式在 N=100 个设备中分布的，这样就会使得每个设备中包含一个两位数的样本，这样的分配方式获得了两种 MNIST 数据集，分别为 MNIST 平衡数据集与 MNIST 不平衡数据集。前者平衡使得每个设备中的样本数量相同，而后者不平衡则使设备的样本数量遵循幂律。为了更精确地处理异质性，以 Sahu（2018）的方法将其表示为合成的 $(\alpha,\ \beta)$，其中 α 控制局部模型彼此之间的差异程度，β 控制每个设备的局部数据与其他设备的本地数据的差异程度。这样就又获得了合成（0,0）与合成（1,1）数据集。

3 本文方法

3.1 本文方法概述

本文采用的模型方法就是最常见的 FedAvg 算法，因为是联邦学习，所以分为中央服务器和其余设备，开始时中央服务器由我们初始化一个模型，然后将该模型广播至各个设备中，各个设备再根据自身的数据进行局部更新，更新的方法 $\mathbf{w}_{t+i+1}^{k} \longleftarrow \mathbf{w}_{t+i}^{k} - \eta_{t+i}\nabla F_k(\mathbf{w}_{t+i}^{k}, \xi_{t+i}^{k}), i = 0, 1, \cdots, E-1$，其中 $\mathbf{w}_{t+i}^{k}$ 为第 k 个设备第 t+i 轮时的模型，η_{t+i} 为第 t+i 轮时的学习率，ξ_{t+i}^{k} 是第 k 个设备第 t+i 轮时均匀选择的样本，这就是边缘设备的更新方法，然后设备在接收到中央服务器的模型进行局部更新 E 轮后再将模型传输至中央服务器，中央服务器再将收到的多个模型进行聚合得到新的模型，按照这样的方法进行一定轮数后或者达到一定精度后最终聚合得到的模型就是最终的结果了。

3.2 部分设备参与

其中中央服务器在聚合的过程中按照之前的说明，并非所有的设备都能按时返回相应的更新后的模型，假设全部的设备都能返回相应的结果，则中央服务器聚合的方法采用 $\mathbf{w}_{t+E} \leftarrow \sum_{k=1}^{N} p_k \mathbf{w}_{t+E}^{k}$ 其中 $\mathbf{p}_k$ 是每个设备的占比且 $\sum_{k=1}^{N} p_k = 1$。然而实际上，并非所有设备都能按时反馈更新后模型，因此等待全部设备的反馈显然是不现实的，所以模型的更新应该采用部分设备参与的前提下来进行考虑，所以我们设置一个阈值 K，当中央服务器收到的设备模型数量达到阈值 K 时就不再接收之后的模型，只使用接收到的前 K 个模型进行更新，因此中央服务器聚合的方法便改为 $\mathbf{w}_{t+E} \longleftarrow \frac{N}{K}\sum_{k\in\mathcal{S}_t} p_k \mathbf{w}_{t+E}^{k}$ 其中 $\mathbf{S}_t$ 表示的为被选中的 K 个设备，亦可证明 $\frac{N}{K}\sum_{k\in\mathcal{S}_t} p_k$ 的结果也是 1。

3.3 抽样方案

进一步对这 K 个设备的抽取进行细分，K 个设备如何进行采样分为两个方案，方案一：假设 St 是一个包含 K 个索引的子集，这些索引是根据概率 $\mathbf{p}_1 \cdots$，$\mathbf{p}_N$ 进行有放回随机选择的。这意味着可能会有重复抽中的可能。聚合 FedAvg 的更新则为 $\mathbf{w}_t \longleftarrow \frac{1}{K}\sum_{k\in\mathcal{S}_t}\mathbf{w}_t^k$；方案二：假设 St 是一个包含 K 个索引的子集，这些索引是从 [N] 中均匀且无重复地进行抽样得到的。假设数据是平衡的，也就是说每个索引的采样概率都是相等的，即 $\mathbf{p}_1 = ... = \mathbf{p}_N = \frac{1}{N}$。聚合 fedavg 的更新则为 $\mathbf{w}_t \longleftarrow \frac{N}{K}\sum_{k\in\mathcal{S}_t}\mathbf{p}_k\mathbf{w}_t^k$，根据两种不同的抽样方案分别设计程序进行研究收敛结果与抽样方案之间的关系。

4 复现细节

4.1 与已有开源代码对比

因为本文主要目的的是公式的推导与实验的验证，因此代码使用的是作者的开源代码进行的实验。在作者开源代码的基础上，本文还增加了一个数据集的生成代码，原先的代码生成包括生成下载 MNIST 数据集，然后生成平衡数据集，因为本文的实验中还需用到非平衡的数据集，因此在这基础上进行了非平衡数据集的生成以满足之后的验证，除了上面两种数据集，还有生产合成数据集，根据参数不同来控制局部模型彼此之间的差异程度。

4.2 实验环境搭建

实验环境的搭建较为简单，安装 python 环境后，再安装 pytorch 库，最后使用 python 编译器即可。

4.3 使用说明

本文的主要程序还是利用联邦学习来进行训练模型，从而观测其收敛性的情况。因此在训练前，第一步就是需要具备训练所需的数据集，分别在 data 文件夹中的不同数据集目录下有各自的数据集生产代码，运行后即可得到所需的数据集文件。之后在开始训练前还分别有几个参数需要进行设置，algo 用来选择训练的模型名称，像本文中使用的就有 fedavg4、fedavg5、fedavg9；dataset 用来选择训练所需的数据集；$\mathbf{num}_{round}$ 表示的是训练的总轮数，之后便可运行得到相应的数据集在相应的参数设置下的结果了。

5 实验结果分析

5.1 每轮本地局部更新次数 E 的影响

根据文章的公式推导分析，模型要想达到给定的精度的通信轮数应满足下面的关系：$\frac{T_\epsilon}{E} \propto \left(1+\frac{1}{K}\right)EG^2 + \frac{\sum_{k=1}^N p_k^2\sigma_k^2 + L\Gamma + \kappa G^2}{E}$ 也可看出来这个通信的轮数应该是每轮局部更新次数 E 的一个双曲线函数，根据这个关系来看这个 E 的设置过大或者过小都会使通信成本增加，理应存在最优的 E 值使其达到最优效果。直观上，E 小意味着通信负担重，E 大意味着收敛速度低。人们需要在通信效率和快速收敛之间进行权衡。我们的实验结果图 1。

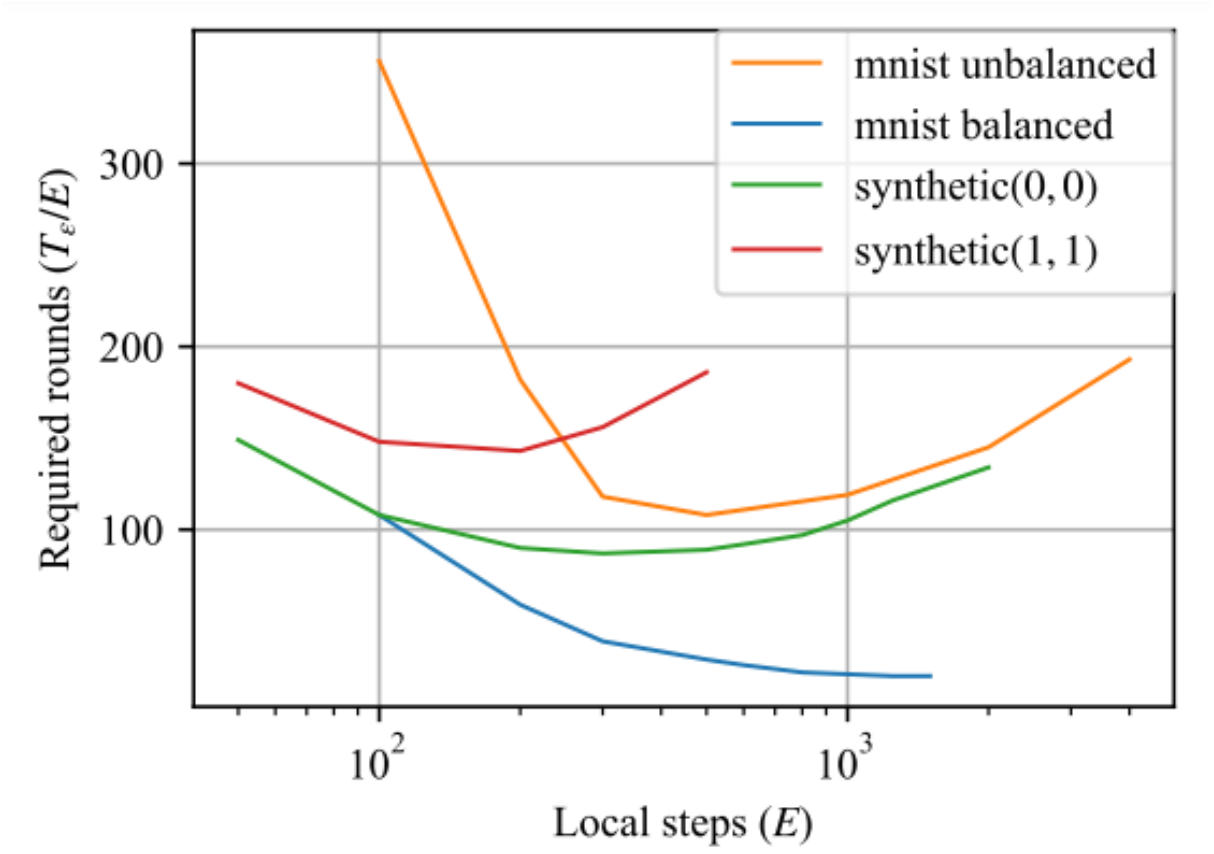

图 1. 实验结果示意

我们在图 1中的非平衡数据集上观察到这种现象。和上述讲述的双曲线关系吻合，随着 E 的增加其通信轮数是先减小后增加的，但是这种现象并没有出现在 mnist 平衡数据集中，这原因仍需进一步的研究。

5.2 参与设备数 K 的影响

上文中也说明了实际情况下并非所有的设备都能按时进行反馈更新好的模型，因此采用使用部分设备进行聚合的方法，就是设置阈值 K 来仅接受前 K 个设备的模型来进行聚合，然后我们对这个 K 值的设置也进行了研究。按照文中的公式分析，较大的 K 可能会稍微加速收敛，当数据为 IID 时，随着 K 的增加，收敛速度显著提高。然而，在 Non-IID 设置下，收敛速率对 K 的依赖性较弱。这同时意味着 FedAvg 无法实现线性加速。我们的实验结果如图 2。

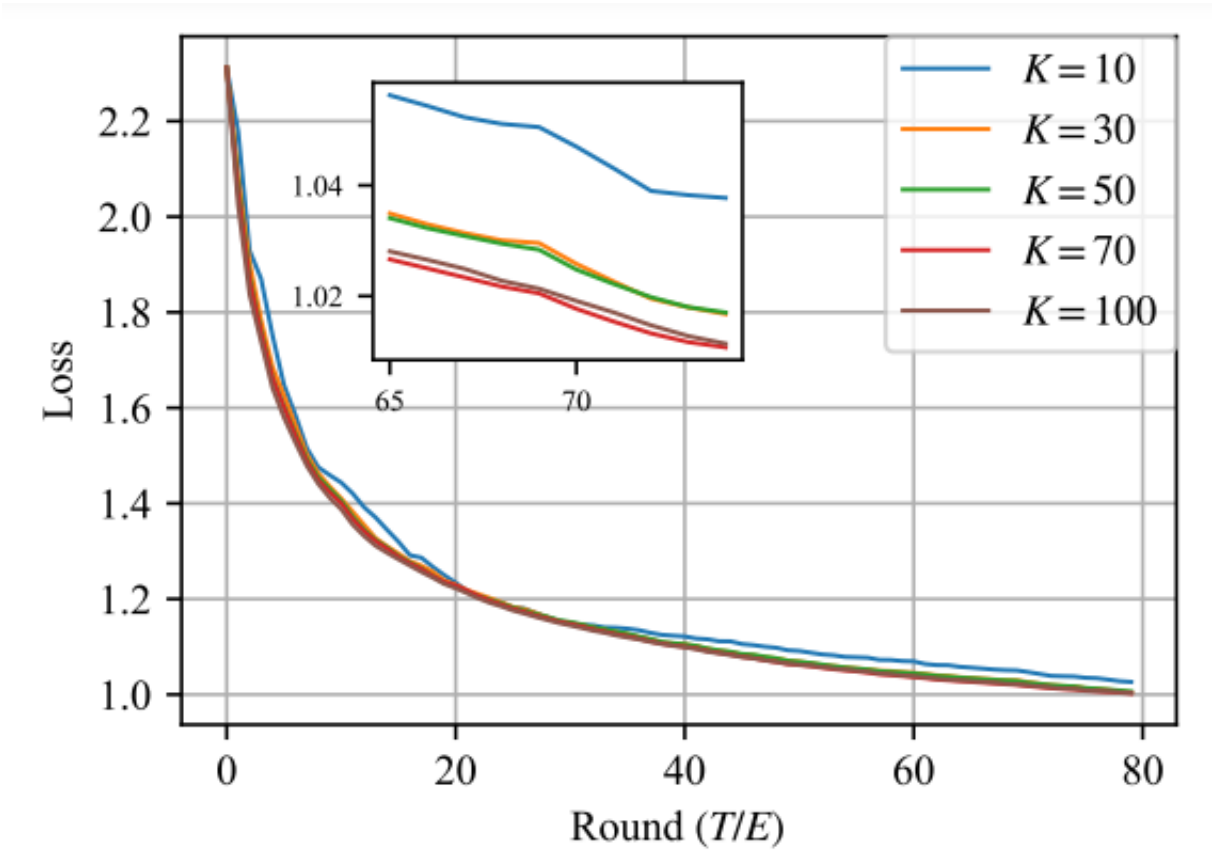

图 2. 实验结果示意

可见在合成 (0,0) 数据集中，K 对 FedAvg 的收敛性影响有限。它揭示了足够大的 K 曲线略好。这证明，当导致采样的方差不是太大，可以使用少量设备而不会严重损害训练过程，这也消除了在凸联邦优化中对尽可能多的设备进行采样的需要。

5.3 采样方案的影响

非均匀采样比均匀采样收敛更快，特别是当 $\mathbf{p}_1 \ldots$，$\mathbf{p}_N$ 是高度不均匀的。如果系统可以随时选择激活 N 个设备中的任何一个，则应使用方案一。然而，通常系统无法控制采样；相反，服务器只使用返回的前 K 个结果进行更新。在这种情况下，我们可以假设 K 个设备从所有 N 个设备中均匀采样，从而应用公式分析来保证收敛性。我们的实验结果如图 3。

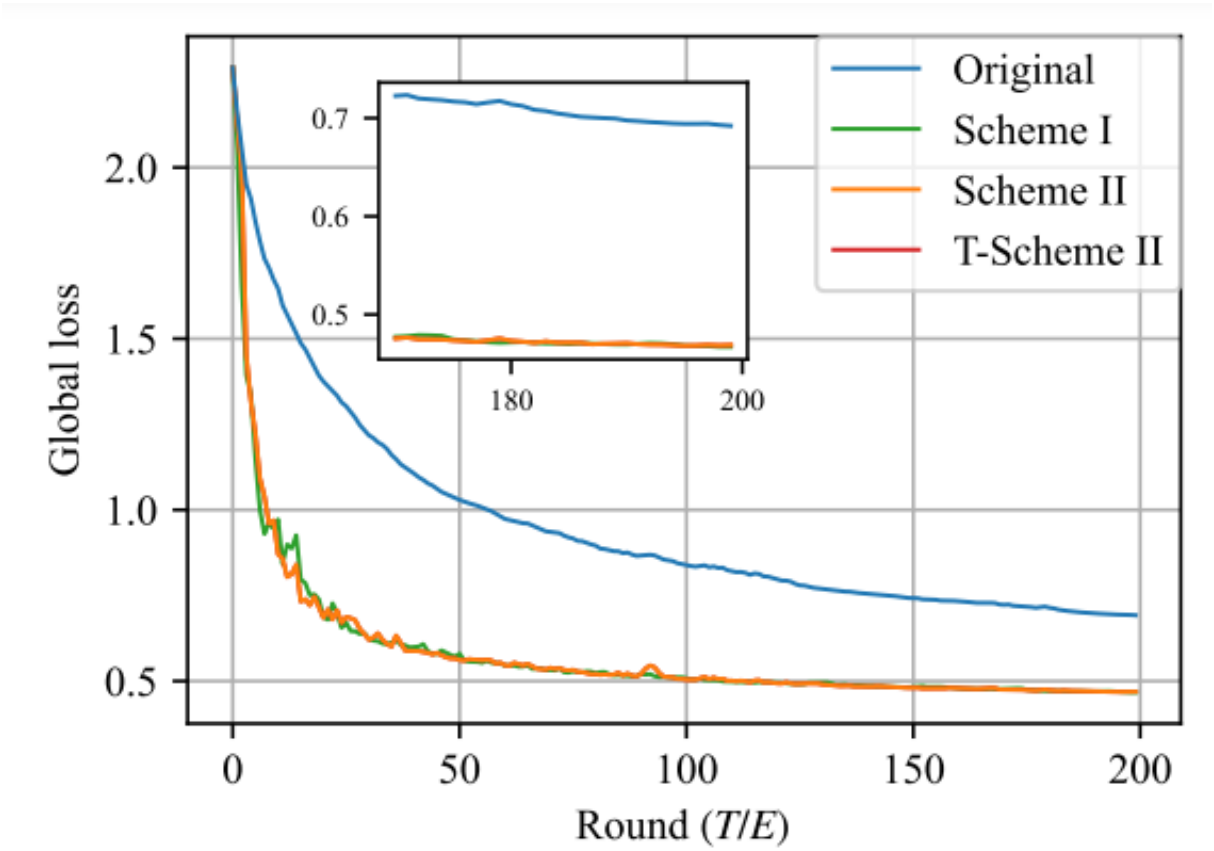

图 3. 实验结果示意

可见在合成 (0,0) 数据集中，适当的采样和平均方案对 FedAvg 至关重要，两种抽样方案都相比原方案的收敛速度更快。

6 总结与展望

本文对于联邦学习中最常用的算法 FedAvg 进行了理论分析，验证了其收敛性与其中各个参数之间的关系，尤其是对于实际中更常出现的部分设备参与以及数据不平衡等情况下的分析，这也为之后的联邦学习算法在这两种情况下的发展奠定了理论基础，以便开发出准确度更高、速度更快、开销更小的联邦学习算法。

参考文献

[1] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 9224–9232, 2018.

[2] Ziyue Jiang, Yi Ren, Ming Lei, and Zhou Zhao. Fedspeech: Federated text-to-speech with continual learning. *arXiv preprint arXiv:2110.07216*, 2021.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[4] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[5] General Data Protection Regulation. General data protection regulation (gdpr). *Intersoft Consulting, Accessed in October*, 24(1), 2018.

[6] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[7] Songyang Zhang, Jiale Zhou, and Xuming He. Learning implicit temporal alignment for few-shot video classification. *arXiv preprint arXiv:2105.04823*, 2021.