

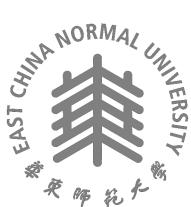
2022 届硕士专业学位研究生学位论文

分类号: \_\_\_\_\_

学校代码: 10269

密 级: \_\_\_\_\_

学 号: 51194501126



東華師範大學

**East China Normal University**

**硕士专业学位论文**

**MASTER'S DISSERTATION (Professional)**

**论文题目：基于联邦学习的隐私保护的技术  
研究**

院 系: 软件工程学院

专业学位类别: 工程硕士

专业学位领域: 软件工程

论文指导教师: 曹珍富 教授

论文作者: 何慧娴

2021 年 09 月 10 日

Thesis (Professional) for Master's Degree in 2021

School Code: 10269

Student Number:51194501126

# EAST CHINA NORMAL UNIVERSITY

## **TITLE: TECHNOLOGIES RESEARCH FOR PRIVACY PRESERVING BASED ON FEDERATED LEARNING**

Department:	Software Engineering Institute
Major:	Software Engineering
Research Direction:	Privacy Preserving
Supervisor:	Professor ZhenFu Cao
Candidate:	HuiXian He

Nov 9, 2021



# 摘 要

随着人工智能的快速发展与移动设备的普及，需要多个参与方协作的应用场景不断涌现，分布式数据处理和分布式机器学习的作用日益凸显。比如分散在多个银行的金融数据、不同医院里的医疗记录、大平台下的每个用户的行为记录，以及智能电表、传感器或移动设备等产生的数据都需要分布式处理与挖掘。数据孤岛是分布式数据处理和分布式机器学习面临的重要挑战之一，作为解决数据孤岛的解决方案，联邦学习是一种很有前景的分布式计算框架，可以在多个分散的边缘设备上本地训练模型，而无需将其数据传输到服务器。随着公民隐私意识的提高和相关法律的完善，联邦学习中的隐私安全问题也日益受到人们的关注，且最新的研究工作表明已经能通过对模型的梯度参数进行攻击，还原用户的隐私数据，即仅通过保持数据的局部性来保护隐私是不够的，并且隐私保护技术在保护隐私的同时，还会牺牲模型精度。为此，本文使用差分隐私技术来保护联邦学习中用户的隐私，并针对分布式场景，分析模型训练过程中针对梯度下降算法的自适应干扰机制，实现提高模型精度的目的，并提出安全混洗模型，防止恶意服务器的攻击。本文主要工作包括如下几个方面：

本文主要的工作和贡献如下：

- (1) 在联邦学习差分隐私的场景下，本文提出了一种新型的、基于本地差分隐私的权重分配自适应干扰算法。在客户端本地训练的神经网络模型中，通过分析前向传播算法，计算每个属性类对于模型输出的贡献比，然后，我们设计了一个自适应噪声添加的方案，根据贡献率注入不同隐私预算的噪声。之后我们设计了解析高斯机制分析加噪累积的隐私预算，并证明了算法满足

$(\epsilon_c + \epsilon_l)$ -差分隐私。与传统的注入噪声的方法相比，我们在相同的隐私保护程度下最大限度地提高了模型的准确性，减少噪声对模型输出结果的影响，提高模型精度。

- (2) 考虑到联邦学习中参数聚合器的攻击和针对参数传播信道的攻击，本文提出了一种新的安全聚合机制，在本地客户端和中心服务器之间新增混洗器，在用户将参数上传到云服务器之前，先对参数进行混洗，模型参数的更新被匿名的发送到混洗器，通过对模型参数的拆分和混洗实现客户端匿名，并且证明了安全混洗模型的可行性。
- (3) 本文在三种数据集上进行实验，并与前人的方案进行对比，证明了自适应本地差分隐私方案和安全混洗框架的结合，在较低的隐私预算下还能使联邦学习模型维持较高的精度。

**关键词：** 联邦学习，隐私保护，本地差分隐私，安全混洗

## ABSTRACT

With the rapid development of artificial intelligence and the proliferation of mobile devices, application scenarios that require the collaboration of multiple participants are emerging. The role of distributed data processing and distributed machine learning is becoming increasingly prominent. For example, financial data scattered across multiple banks, medical records in different hospitals, behavioural records of each user under a large platform, as well as data generated by smart meters, sensors or mobile devices all need to be processed and mined in a distributed manner.

Data silos are one of the key challenges that distributed data processing and distributed machine learning facing. As a solution to address data silos, Federated Learning is a promising distributed computing framework that can train models locally on multiple decentralised edge devices without transferring their data to servers. With the increasing awareness of privacy among citizens and the improvement of related laws, privacy security in federation learning is also a growing concern, and recent research work has shown that it has been possible to restore users' private data by attacking the gradient parameters of the model, i.e. it is not enough to protect privacy by keeping the data local, and privacy-preserving techniques just protect privacy at the huge expense of model accuracy.

To this end, this paper uses differential privacy techniques to protect the privacy of users in federated learning, and analyses the adaptive interference mechanism against the gradient descent algorithm during model training for distributed scenarios. In order to achieve the goal of improving model accuracy, we propose a secure split-shuffle

model to prevent attacks by malicious servers. The main work of this paper includes the following aspects:

1. In a federated learning differential privacy scenario, this paper presents a novel, local adaptive differential privacy interference algorithm. In a client-side locally trained neural network model, the contribution ratio of each attribute class to the model output is calculated by analysing the forward propagation algorithm, and then we develop an adaptive noise addition scheme that injects noise with different privacy budgets according to the contribution ratio. Compared with the traditional method of injecting noise, we maximise the accuracy of the model with the same degree of privacy protection, reduce the impact of noise on the model output results and improve the model accuracy.
2. Considering the attacks on parameter aggregators in federated learning, this paper proposes a new secure aggregation mechanism by adding a new shuffler between the local client and the central server, where parameters are splitted and shuffled before users upload them to the cloud server. The updates to model parameters are sent anonymously to the shuffler, achieving client anonymity through splitting and shuffling of model parameters.Finally, we demonstrate the feasibility of the shuffle model.
3. In this paper, we do experiments on three datasets , then demonstrate the combination of the local adaptive differential privacy algorithm and the secure shuffle framework can reach the balance between model accuracy and privacy in the federated learning model.

**Keywords:** *Federated learning, Privacy preserving, Local differential privacy , Security shuffle*



# 目录

<b>第一章 绪 论 . . . . .</b>	<b>1</b>
1.1 研究背景及意义 . . . . .	1
1.2 问题和挑战 . . . . .	4
1.2.1 数据异构 . . . . .	4
1.2.2 高昂的通信代价 . . . . .	4
1.2.3 安全性和隐私威胁 . . . . .	5
1.3 国内外研究现状 . . . . .	5
1.3.1 攻击模型的研究现状 . . . . .	6
1.3.2 隐私保护的研究现状 . . . . .	7
1.4 本文工作与主要贡献 . . . . .	9
1.5 本文组织结构 . . . . .	9
1.6 本章小结 . . . . .	10
<b>第二章 基础知识 . . . . .</b>	<b>11</b>
2.1 联邦学习 . . . . .	11
2.1.1 基本概念 . . . . .	11
2.1.2 联邦学习的分类 . . . . .	12
2.1.3 模型框架 . . . . .	13
2.2 深度神经网络 . . . . .	14
2.2.1 基本结构 . . . . .	14
2.2.2 前向传播算法 . . . . .	15
2.2.3 反向传播算法 . . . . .	16

2.3	差分隐私 . . . . .	17
2.3.1	基本定义 . . . . .	17
2.3.2	相关概念 . . . . .	18
2.3.3	实现机制 . . . . .	20
2.4	本章小结 . . . . .	21
<b>第三章</b>	<b>联邦学习中的自适应本地差分机制 . . . . .</b>	<b>22</b>
3.1	引言 . . . . .	22
3.2	基于自适应差分隐私的随机梯度下降算法 . . . . .	24
3.3	详细设计 . . . . .	25
3.3.1	自适应噪声添加 . . . . .	26
3.3.2	梯度范数裁剪 . . . . .	29
3.3.3	解析高斯机制 . . . . .	29
3.4	隐私性证明 . . . . .	30
3.5	隐私预算分析 . . . . .	33
3.6	本章总结 . . . . .	35
<b>第四章</b>	<b>联邦学习的安全混洗模型 . . . . .</b>	<b>37</b>
4.1	引言 . . . . .	37
4.2	安全混洗模型 . . . . .	38
4.2.1	客户端抽样 . . . . .	39
4.2.2	混洗器 . . . . .	40
4.3	隐私放大效应 . . . . .	41
4.4	模型收敛性分析 . . . . .	44
4.5	本章总结 . . . . .	46
<b>第五章</b>	<b>实验与评估 . . . . .</b>	<b>47</b>
5.1	基准数据集介绍 . . . . .	47
5.2	实验环境与配置 . . . . .	48
5.3	实验设计 . . . . .	48
5.3.1	联邦学习模型 . . . . .	48

5.3.2	神经网络模型 . . . . .	49
5.4	自适应扰动方案的实验评估 . . . . .	50
5.5	安全混洗算法的实验评估 . . . . .	53
5.6	结果分析 . . . . .	55
5.7	本章小结 . . . . .	56
<b>第六章</b>	<b>总结与展望 . . . . .</b>	<b>57</b>
6.1	论文总结 . . . . .	57
6.2	论文展望 . . . . .	58
	<b>参考文献 . . . . .</b>	<b>60</b>
	<b>致谢 . . . . .</b>	<b>66</b>
	<b>发表论文和科研情况 . . . . .</b>	<b>68</b>

# 插图

1.1	联邦学习模型概况 . . . . .	3
2.1	联邦学习模型工作流程 . . . . .	13
2.2	深度神经网络结构图 . . . . .	15
2.3	前馈神经网络结构图 . . . . .	16
2.4	差分隐私的相邻数据集示意图 . . . . .	18
3.1	层间依赖传播算法 . . . . .	26
4.1	联邦学习中的安全模型框架 . . . . .	40
4.2	联邦学习安全模型中执行参数拆分混淆的混淆器 . . . . .	41
5.1	卷积神经网络结构图 . . . . .	49
5.2	梯度固定加噪方法下模型准确率随隐私预算变化情况 . . . . .	51
5.3	不同隐私预算的自适应干扰模型的准确率 . . . . .	52
5.4	DP-SGD、DLPP、ACDP 在模型准确率和隐私预算上的对比 . . . . .	52
5.5	安全混淆模型中参与混淆的本地客户端数量对联合模型精度的影响 .	53
5.6	安全混淆模型中通信轮数和客户端采样比对联合模型精度的影响 .	54
5.7	自适应差分混淆模型和其他联邦学习隐私保护模型的比较 . . . . .	55

# List of Algorithms

1	基于自适应差分隐私的随机梯度下降算法 . . . . .	25
2	解析高斯算法 . . . . .	31
3	联邦学习中的安全模型算法: $\mathcal{A}_{\text{csdp}}$ . . . . .	39
4	混淆器中的拆分混淆算法 . . . . .	42

# 第一章 緒論

## 1.1 研究背景及意义

随着机器学习的不断发展和壮大，我们一方面惊叹于它的成就，比如 Alpha GO 击败了围棋世界冠军——柯洁、面部识别技术帮助我们抓住了躲藏多年的逃犯、大型工业企业也大力应用机器学习技术推动生产力的快速发展；另一方面，我们也认识到，机器学习还有巨大的发展潜力，例如：在医疗建设方面，构建基于大量病例的医疗救助诊断系统<sup>[7]</sup>；在金融建设方面，运行基于大量商业行为数据的信用风险控制模型，帮助高价值的企业融资，并基于整个产业链的数据提供个性化的产品分配和营销策略。我们在各行各业真正见证了人工智能（AI）的巨大潜力，以及已经开始期待在许多应用中使用更复杂、更尖端的人工智能技术，包括无人驾驶、医疗、金融等<sup>[1]</sup>。今天，人工智能技术几乎在各方面都大显身手。传统的人工智能系统依赖于集中管理的训练数据集，建立在大量数据上，从数据中学习特征，从而完成复杂的任务，甚至是人类也难以完成的操作。

人工智能的基础是大数据，而大多数训练数据是来自于不同组织的个人或机构。在一个深度学习的项目中，可能涉及到多个领域，需要采集不同公司、不同机构、不同部门的数据进行融合。（比如研究用户的消费爱好和水平，可能需要采集各个消费平台、银行、商店等多个机构的数据），然而在现实生活中，数据是分散在各地的，很难进行整合。于是诞生了集中式深度学习，它是通过收集数据并将其发送到一个能看到并控制所有数据的中央服务器，完成所有训练数据的整合。这个中心位置不仅要有强大的计算机集群来训练和创建深度学习模型，还要处理敏感数据并防止数据被用于其他目的。此外，敏感数据的处理方式必须不损害用户

的隐私。集中式的深度学习需要大量的数据去训练模型，达到较好的训练效果。

近些年来，大量的互联网公司从数百万的用户那里收集数据，然后利用这些数据进行深度学习，实现智能推荐<sup>[6]</sup>、语音识别<sup>[5]</sup>、面部识别<sup>[3][4]</sup>等。然而在集中式深度学习中，中央服务器是半可信的，半可信是指服务器是诚实但好奇的（Honest but Curious），在处理一些敏感数据时，用户也不知道他们的数据将被用于何处，而且这些数据的采集很可能威胁到用户的隐私安全。在 2018 年，中国互联网协会收到用户举报发现，腾讯音乐等多家应用软件以“通过深度学习向用户提供更好的服务”为由，长期收集并保存大量的用户个人数据，如照片、地址、电话等，甚至将这些包含了用户大量个人隐私的数据用作其他途径，为企业谋取更多利益。资料显示，许多应用软件在数据收集方面存在大量的安全漏洞，比如，未经授权访问用户手机中的“通讯录”、“位置”、“麦克风”等信息，导致千万级的用户资料泄露。

随着越来越多的涉及数据泄漏和隐私侵权事件的发酵，人们的隐私意识的普遍提高，越来越的用户关注自己的隐私信息是否在未经个人许可，或者出于商业和政治目的被他人或机构利用。更多的用户拒绝向互联网企业提供“收集数据”的权限，关闭了“通讯录”、“短信”、“位置”等访问权限。同时，相关的隐私法律法规不断完善，2018 年欧洲联盟会出台的《通用数据保护条例》<sup>[2]</sup> 强调保护用户的隐私和数据的使用安全性，2020 年中国出台的《网络安全与数据合规》白皮书中明确要求加强用户个人信息保护。随着个人意识和国家政策的关注，在大数据和人工智能领域数据采集和使用的过程中，保护用户隐私和数据的安全使用显得越来越重要。

人工智能的力量是基于大数据的，在数据监管和隐私保护的要求下，传统的集中式深度学习系统难以收集到模型训练所需要的数据，进而无法提供更专业的网络服务。大数据的基础没有了，人工智能的未来也就岌岌可危。那么能否创建一个深度学习框架，使人工智能系统能够更有效和准确地集体使用数据，同时满足隐私性、安全性和监管要求，并解决数据孤岛的问题？

为了解决这个问题，Google 在 2016 年率先提出了联邦学习的概念<sup>[21]</sup>，它是一个分布式的深度学习框架。如图1.1所示，在联邦学习框架中存在多个参与者和一

个中央服务器。多个参与者根据自己的本地训练数据集，训练网络模型，而无需共享这些训练集数据。每个参与者在本地的设备上训练本地模型，并与其他参与者共享联合模型的训练参数。中央服务器通过收集和交换本地用户上传的训练参数，训练联合模型，使之能达到与集中式深度学习训练的模型几乎相近的精度。由于参与者之间无需共享本地训练数据，分布式联邦学习系统被认为是有利保护用户隐私的。

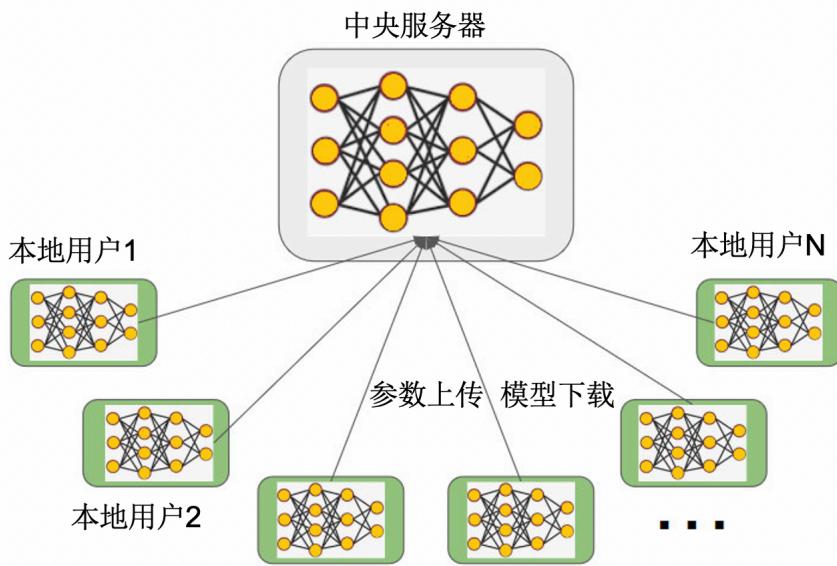


图 1.1: 联邦学习模型概况

联邦学习在隐私敏感的场景（包括金融、工业和许多其他与数据相关的场景）中展现出巨大的前景，这是因为它具有独特的优势，多个参与者无需共享本地数据却能训练出统一的全局模型，保护了本地数据的隐私性。联邦学习解决了数据聚合的问题，并允许一些机器学习模型和算法在各机构和部门之间进行独立设计和训练。在一些移动设备上的机器学习模型应用中，联邦学习展示出良好的性能和稳健性。此外，对于一些没有足够的私人数据来训练准确的本地模型的用户（客户）来说，深度学习模型和算法的性能可以通过联邦学习得到显著改善<sup>[18]</sup>。

## 1.2 问题和挑战

### 1.2.1 数据异构

由于联邦学习的重点是通过以分布式方式从所有参与的客户端设备中学习本地数据来获得高质量的全局模型，所以它无法捕捉每个设备的个人信息，导致推理或分类性能下降。此外，传统的联邦学习要求所有参与的设备同意使用一个共同的模型来共同训练，这在复杂的现实世界物联网应用中是不现实的。研究人员对联邦学习在实际应用中面临的问题总结如下<sup>[19]</sup>：

(1) 设备的异质性：由于客户端设备的硬件条件 (CPU、内存)、网络连接 (3G、4G、5G、WiFi) 和电源 (电池) 的变化，联邦学习网络上每个设备的通信、存储和计算能力都可能有差异。受限于网络和设备，不能保证在任何时候所有设备都能参与学习。此外，设备可能会受到意外事件的影响，如断电或断网。这种异质性的系统结构影响了联邦模型的整体学习战略<sup>[20]</sup>。

(2) 统计的异质性：在整个网络中，设备通常以不同的方式产生和收集数据，而且不同设备的数据量、特征等会有很大的不同，所以联邦学习网络中的数据不是独立和相同的分布 (Non Independent and Identically Distributed, Non-IID)。目前的深度学习算法主要是基于 IID 数据。因此，非 IID 数据的异质属性给建模、分析和评估带来了重大挑战。联邦学习的参数聚合 (Federated Average, FedAvg) 方法可以解决非均匀同分布数据的问题，但是当数据分布偏态很严重的时候 FedAvg 的性能退化严重，一方面其性能比中心化的方法差好多，另一方面它只能学习到设备粗粒度的特征而无法学习到细粒度的特征。

(3) 模型的异质性：每个客户根据其应用场景要求定制不同模型。

### 1.2.2 高昂的通信代价

在联邦学习过程中，根据存储在几十甚至几百万个远程客户端设备上的数据来学习一个全局模型。原始数据被储存在本地的客户端设备上，在训练过程中这些远程设备必须不断地与中央服务器互动，以完成全局模型的构建<sup>[17]</sup>。通常情况下，

整个联盟学习网络可能涉及大量的设备，而网络通信可能比本地计算慢几个数量级，因此高通信成本成为联邦学习的关键瓶颈。

### 1.2.3 安全性和隐私威胁

联邦学习解决了传统集中式深度学习所面临的大规模数据收集等问题，减少了数据在收集过程中所遇到的隐私泄露风险，节省了传输数据所占用的通信资源。然而，在训练过程中传递模型的更新信息仍然存在向第三方或中央服务器暴露敏感信息的风险。隐私保护成为联邦学习需要重点考虑的问题。

在联邦学习系统中，攻击方可能是内部攻击者，比如中央服务器、本地客户端；也有可能是外部攻击者。他们试图影响、破坏联邦学习模型的准确性，通过客户端上传的参数恶意的窃取用户的训练数据。

有一些恶意参与者会发送无效的模型参数更新到中央服务器，破坏全局模型的训练。比如，这些恶意参与方作为本地客户端参加训练，修改本地的训练数据，对本地数据注入一些有毒的数据，进行投毒攻击，从而损害全局模型的准确性，操纵模型的预测结果。

外部攻击主要通过本地客户端与中央服务器之间的通信信道发起。在训练过程中，局部模型更新和全局模型参数的结合过程，提供了关于训练数据的隐藏知识，用户的个人信息很有可能泄露给不受信任的服务器或其他恶意第三方。例如，白盒推理攻击和黑盒推理攻击<sup>[21]</sup> 通过客户端上传的参数恶意的窃取用户的训练数据生成的样本原型。

## 1.3 国内外研究现状

尽管联邦学习框架提供了隐私保护的机制，还是有各种类型的攻击方式可以攻击联邦学习系统，从而破坏联邦学习的系统安全和参与方的隐私。本节将重点讨论关于联邦学习的攻击模型和隐私保护的研究现状。

### 1.3.1 攻击模型的研究现状

各类攻击模型阻碍了深度学习技术的发展，也会极大地威胁到人们的隐私敏感信息。无论是模型并行化还是数据并行化，分布式学习系统在用户数据隐私性方面相对于集中式学习存在一定的优势。但是在分布式联邦学习系统中，参与者需要多次的联合迭代过程才能完成全局模型的收敛，参与者的参数也需要多次的训练、上传和共享，这些参数中包含的参与者训练集的相关信息，用户的信息可以通过计算用户上传的多个参数得到。因此，有许多外部攻击者或者恶意服务器通过用户上传的参数恢复出原始的样本试例。

模型反演攻击<sup>[11]</sup> 利用用户上传的参数信息，以一种很简单的方式攻击用户数据：一旦用户的网络模型经过训练并达到收敛，攻击者就可以通过调整网络模型权重的梯度，获得网络模型中所有表示类的逆向工程试例。在模型反演攻击中，攻击者无需接触目标信息的标签类，攻击模型仍然能够恢复原始样本试例。这一攻击模型表明，任何经过精确训练的深度学习网络，无论是以何种方式进行训练收敛，都可以透露深度网络中区分不同标签类的信息。但是参数中包含的信息有限，模型反演攻击方式很难攻击卷积神经网络等复杂深度网络模型，在模型进行了一定的隐私保护后，攻击也基本失效。

生成对抗网络攻击（GAN 攻击）<sup>[12]</sup>：目前研究人员也利用诸多安全模型对深度学习网络的训练数据集进行保护，但 Hitaj 等人<sup>[8]</sup>发现，一个联邦学习框架非常容易受到系统内参与者发起的主动攻击。他们首次提出了基于 GAN 的模型<sup>[7] [8] [9] [10]</sup>重建攻击，攻击者为本地客户端。在训练阶段，攻击者冒充为本地的无害用户，训练 GAN 模型，模拟产生其他用户的训练数据产生的原型样本，之后通过不断添加假的训练样本，攻击可以逐渐影响整个学习过程，使受害者暴露出更多关于被攻击者的目标类的敏感信息。除了客户端发起的 GAN 攻击，服务器也能通过 GAN 攻击。恶意服务器最初假装是一个为用户提供联邦学习服务的正常服务器，其主要目标是重建被攻击用户的训练样本。

投毒攻击<sup>[15]</sup>：在联邦学习框架中，攻击者可能试图修改、删除或插入恶意信息

到训练数据中，以破坏原始数据分布，改变学习算法的逻辑。两种常见的中毒攻击的例子包括标签反转攻击<sup>[9]</sup> 和后门攻击<sup>[10]</sup>。标签反转攻击是指恶意用户反转样本标签，并在训练数据中加入预定义的攻击点，导致训练后的模型偏离预测的界限。与标签反转攻击不同，后门攻击要求攻击者用精心设计的训练数据，利用特定的隐藏模式来训练目标的深度神经网络（DNN）模型。这些模型被称为”反馈回路”，可以干扰学习模型，并在预测阶段产生与真实情况截然不同的结果。

成员推理攻击<sup>[13]</sup>：给定一个数据点和一个预训练过的模型，判断该数据点是否被用于训练该模型。在联邦学习中，每轮迭代的梯度都被发送给了服务器，在成员推理攻击中，中央服务器有能力推断一个特定数据点是否在本地训练集中。在一些情况下，它可以直接导致隐私泄露。例如，发现特定患者的临床记录用于训练与疾病相关的模型会泄露该患者患有疾病的事。在实践中，Fredrikson 等<sup>[51]</sup> 在 2014 年发现通过机器学习模型针对医院用药系统，结合病人的统计信息可以回推出该病患的个人资料。

### 1.3.2 隐私保护的研究现状

随着针对联邦学习框架的攻击模型增多，研究人员开始关注训练网络模型时存在的隐私安全问题。关于联邦学习的隐私定义主要分为全局隐私和局部隐私。在本地局部隐私中，每个客户端发送一个不同的隐私值，该值被安全的加密的上传到中央服务器。在全局隐私中，服务器在最终输出中添加不同的隐私噪音。安全多方计算、同态加密<sup>[10]</sup> 和差分隐私<sup>[9]</sup> 是最常见的保证联邦学习中的安全和隐私的技术。在分布式环境下，常用密码学中的同态加密（Homomorphic Encryption, HE）和本地差分隐私（Local Differential Privacy, LDP）技术来解决分布式数据收集中的隐私保护问题，保证数据收集者不能拥有任何个体用户数据的准确值，但是仍能获取用户数据的一些基本统计信息。

安全多方计算（Secure Multi-Party Computation, SMC）是由姚期智在 1982 年提出的<sup>[16]</sup>，多个参与者在不泄露各自隐私数据情况下，利用隐私数据参与保密计算，共同完成某项计算任务。目前，在安全多方计算领域，主要用到的是技术是秘

密共享、不经意传输、混淆电路、同态加密、零知识证明等关键技术。

同态加密是一种加密形式，允许在加密之后的密文上直接进行计算，且计算结果解密后和明文的计算结果一致。利用同态加密技术可以实现让解密方只能获知最后的结果，而无法获得每一个密文的消息，提高信息的安全性。。如果对密文进行加法（或乘法）运算后解密，与明文进行加法（或乘法）运算的结果相等，则称这种加密算法为加法（乘法）同态。如果同时满足加法和乘法同态，则称为全同态加密。在联邦学习中，因为只需要对中间结果或模型进行聚合，一般使用的同态加密算法为 PHE（加法同态加密算法），在加密机制下进行本地客户端和云服务器的参数交换，保护用户数据隐私，例如在 FATE 中使用的 Paillier 即为加法同态加密算法<sup>[23]</sup>。

差分隐私（Differential Privacy, DP）方法的主要原理是向数据添加噪音，或使用概括方法来掩盖某些敏感属性<sup>[14]</sup>，使至多相差 1 条数据的 2 个数据集的查询结果概率不可区分，以保护用户的隐私。在联邦学习框架中，通过在本地模型和全局模型中对相关训练参数添加噪声，进行扰动，使敌手无法获得真实的模型参数，进而防御模型反演攻击、成员推理攻击等。在深度学习中，差分隐私可以作为一种局部隐私保护方案来保护用户梯度的隐私，Ding M 等人<sup>[24]</sup> 提出了一种隐私保护的深度学习方法，主要通过添加噪声来扰乱少量步骤后的局部梯度，将差分隐私机制与模型训练中的随机梯度下降算法（SGD）相结合。令人担忧的是，现有的差分隐私保护方案很难权衡隐私保护预算的成本和联邦学习模型的有效性，因为较高的隐私保护预算可能对一些大规模的攻击（如基于 GAN 的攻击）不是很有用，而较低的隐私保护预算可能阻碍模型的局部收敛。而且与安全多方计算等密码学技术相比，差分隐私无法保证参数传递过程中的机密性。

总的来说，安全多方计算基于复杂的计算协议，同态加密的运算成本非常高，而差分隐私破坏了数据的可用性，很难在模型性能和隐私成本上达到平衡，当前的研究方向主要集中在对数据集和神经网络中的参数的加密和隐私保护机制上，较少关注到模型整体框架等过程。目前的联邦学习中的隐私保护方法还有许多不足，

不能在隐私性和模型可用性上都达到一个相对满意的效果，此外，大部分方法是基于统一的、固定的参数设置，会导致模型迭代过程中累积大量隐私损失，使模型性能大幅下降。因此，在联邦学习场景下，保护用户隐私的同时维持模型准确性仍需大量的研究。

## 1.4 本文工作与主要贡献

针对联邦学习中隐私性和模型精度的双重指标，本文提出了本地自适应差分隐私算法和安全混洗框架，主要的工作和贡献包含以下三个方面：

- (1) 在联邦学习差分隐私的场景下，本文提出了一种新型的、基于本地差分隐私的权重分配自适应干扰算法。在客户端本地训练的神经网络模型中，通过分析前向传播算法，计算每个属性类对于模型输出的贡献比，然后，我们设计了一个自适应噪声添加的方案，根据贡献率注入不同隐私预算的噪声。与传统的注入噪声的方法相比，我们在相同的隐私保护程度下最大限度地提高了模型的准确性，减少噪声对模型输出结果的影响，提高模型精度。
- (2) 考虑到联邦学习中参数聚合器的攻击和针对参数传播信道的攻击，本文提出了一种新的安全聚合机制，在本地客户端和中心服务器之间新增混洗器，在用户将参数上传到云服务器之前，先对参数进行混洗，模型参数的更新被匿名的发送到混洗器，通过对模型参数的拆分和混洗实现客户端匿名，并且证明了安全混洗模型的可行性。
- (3) 本文在三种数据集上进行实验，并与前人的方案进行对比，证明了自适应本地差分隐私方案和安全混洗框架的结合，在较低的隐私预算下还能使联邦学习模型维持较高的精度。

## 1.5 本文组织结构

本文一共六章，主要内容的组织安排如下：

第一章对本文研究内容：联邦学习的研究背景、国内外研究现状进行了阐述，介绍了目前联邦学习中的隐私保护的研究现状和发展方向。

第二章详细介绍本文研究内容所涉及的一些理论基础与背景知识，包含了联邦学习的相关概念，差分隐私的基础知识和神经网络的基本结构。

第三章描述了本文所提出的本地自适应差分隐私算法的设计和实现，根据神经网络前向传播算法，分析属性值的贡献度，根据属性值自适应添加高斯噪声，然后采用解析高斯机制分析添加的噪声大小，并证明了在自适应差分隐私机制下的联邦学习算法的隐私性。

第四章在上一章的基础之上，提出了一种联邦学习安全混洗模型，混洗器对客户端上传的梯度进行采样后，然后拆分混洗，再将混洗模型和自适应本地差分隐私保护方法结合在分布式系统中，提高系统学习效果。并且证明了安全混洗模型的隐私性和收敛性。

第五章为实验部分，基于本文提出的隐私保护框架，我们在三个基准数据集的进行了实验和讨论，并与之前的差分隐私联邦学习框架进行对比实验。

第六章是对本文的一个内容总结和展望，首先对本文的研究内容进行了概括，并对现有的不足进行总结，对未来的研究和改进方向进行了展望。

## 1.6 本章小结

这一章节为绪论，主要介绍的是本文章的研究背景以及意义，对当下联邦学习中的应用以及存在的问题与挑战行了介绍和总结、讨论了联邦学习中隐私威胁和隐私保护的国内外研究现状，并对文章的主要工作和文章的章节进行了介绍。

## 第二章 基础知识

在本章节中我们将介了本文研究所需要的一些基本知识，有助于更好的理解之后章节的内容。

### 2.1 联邦学习

#### 2.1.1 基本概念

深度学习的成功应用需要建立在大量数据的基础之上，才能完成人们指派的学习任务。然而，近年来数据泄漏和隐私侵权事件不断发生，用户开始更加关注他们的隐私信息是否未经自己的许可，或被他人出于商业或者政治目的而被利用。人们逐渐地意识到，在人工智能的构建与使用的过程中保护用户隐私和数据机密的重要性。

大部分拥有的训练数据是由不同组织的个人、部门产生并拥有的，传统机器学习的做法是收集数据并传输到一个中心服务器，服务器可以看见并控制所有的数据，因此这个中心点不仅需要拥有高性能的计算集群来训练和建立机器学习模型，而且还需要处理敏感数据，避免泄漏用户隐私。然而，这种方法需要用户对服务器的完全信任，这已经不再有效或适用了。在这样的情况下，数据拥有者倾向于将自己的数据保留在自己的手中，进而会形成各自孤立的数据孤岛，至此大量数据的基础已经消失，人工智能的未来将面临绝境。作为回应，2016 年谷歌<sup>[25]</sup>率先提出联邦学习概念，旨在建立高质量分布式学习的框架。在联邦学习系统中，数据所有者（参与者）不需要彼此共享原始数据，也不需要依赖单个可信实体（中心服务器）来进行机器学习模型的分布式训练。相反，参与者通过在自己的本地数据上执

行本地训练算法，并且只与参数服务器共享模型参数，来共同协作训练联邦模型。在每轮训练中，参数聚合节点会随机选择合适的节点加入到训练池中。那些被选中的本地节点通常是保持充电且无线网络可用。然后参数聚合节点平均所有已提交者的权重并作为下一轮回合的初始化模型。重复此过程直至终止条件。

### 2.1.2 联邦学习的分类

根据用户维度和模型特征维度的重合去分类，将联合学习分为水平联邦学习、纵向联邦学习和联合迁移学习<sup>[26]</sup>。

- **水平联邦学习：**当两个数据集的用户属性重叠较多而用户重叠较少的情况下，我们对数据集进行横向切割（即按用户维度切割），取出两边用户属性相同但用户不完全相同的那部分数据用于训练。这种方法被称为横向联合学习。例如，两家银行位于不同的地区，有来自各自地区的用户群，而且它们之间的联系非常少。然而，他们的业务活动非常相似，因此他们的用户特征也是一样的。在这个阶段，我们可以使用跨部门的联邦学习来建立一个联合模型。2016年，谷歌提出了一个在安卓手机上更新模型的联合数据建模系统：模型参数在本地不断更新，并在各个用户使用安卓手机时上传到安卓云端，使拥有数据的每一方都能建立一个具有相同特征维度的联合模型。
- **纵向联邦学习：**在两个数据集中用户重叠较多，而用户属性重叠较少的情况下，我们将数据集纵向切开（即按特征维度），选择数据集中两边用户相同但用户属性不完全相同的部分进行训练。这种方法被称为纵向的联邦学习。例如，有两个不同的组织，一个是在一个地方的银行，另一个是在同一个地方的电子商务公司。他们的用户群很可能包括该地的大部分人口，所以有很大的用户交集。然而，由于银行储存的是用户的收入和支出以及信用评分的数据，而电子商务公司储存的是用户的浏览和购买历史的数据，他们的用户档案并没有那么紧密的联系。纵向的联邦学习是在一个加密的空间里将这些不同的功能结合起来，以提高模型的性能。

- **联合迁移学习：**联合迁移学习是通过使用迁移学习模型来弥补数据或标签的差距，而不是对数据进行切分。当两个数据集中的用户和用户属性几乎没有重叠。这种方法被称为联合迁移学习。这里举一个例子，考虑两个不同的组织，一个是中国的银行，另一个是美国的电子商务公司。由于地理上的限制，这两个机构的用户群重叠的地方很少。由于它们是不同类型的组织，数据的特点也没有太多的重叠。在这种情况下，为了保证有效的联邦学习，可以引入联合迁移学习，以克服单变量数据量小和标注样本小的问题，提高模型的效率。

### 2.1.3 模型框架

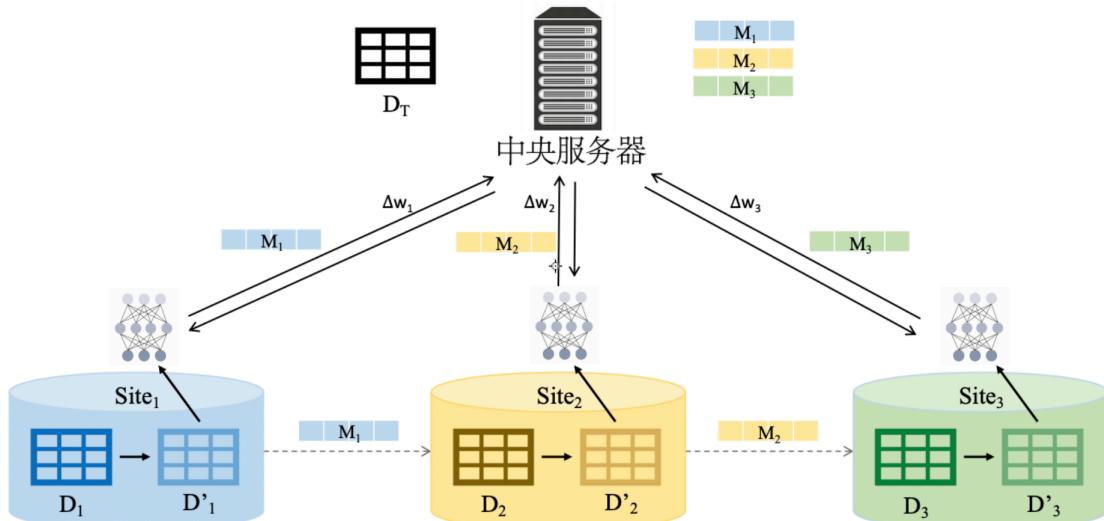


图 2.1: 联邦学习模型工作流程

本文我们提出的方案是基于典型的分布式横向联邦学习系统架构，即各个参与者的本地数据集特征空间相同，但样本不同。通过中心服务器，各个参与者相互协作，在保护个人本地敏感数据的同时，有效地提高本地学习效果。如图2.1所示，分布式横向联邦学习的基本工作流程如下：

- **初始化：**所有用户在他们的设备上都有一个预先分配的神经网络模型，并且可以自愿加入联邦学习协议，指定相同的深度学习和模型训练目标。

- **本地训练:** 在一个给定的通信回合中, 联邦学习参与者首先从中央服务器下载全局模型参数, 然后使用他们的私人训练模式训练模型, 更新本地模型(即模型参数), 并将这些更新发送到中央服务器。
- **中央参数聚合:** 中央服务器汇总此次通信回合中所有参与者上传的模型参数, 并对其进行聚合求得全局模型的参数, 然后更新全局模型。
- **迭代更新:** 迭代地执行上述步骤直至全局模型参数满足收敛条件, 最终得到最优的全局模型。

## 2.2 深度神经网络

### 2.2.1 基本结构

神经网络<sup>[34]</sup>的设计来源于人脑的结构, 是人脑处理信息方式的一个简化模型。人类的大脑是人中枢神经系统中的主要部分, 这些神经元像网状物一样相互连接。来自外部环境的刺激或来自感觉器官的输入通过感受器之后进入传入神经, 神经元一层一层兴奋(激活)后, 传导到神经中枢(大脑或脊髓), 神经中枢(相当于输出层)根据信号的类型做出不同的判断(分类), 然后再下达命令, 将信号传递到输出神经。不同的信号, 大脑都可以进行学习和分辨, 而这一通用的模型, 就是神经网络。

神经网络的基本单元是神经元, 由数百万个简单的神经元组成, 这些神经元密集地相互连接, 神经元按层排列。每一层有多个神经元, 层与层之间是“前馈传播”的, 也就是说, 网络中的数据只在一个方向上移动。一个单独的神经元可能与它前面一层的几个神经元相连, 它从这些神经元接收数据; 与它后面一层的几个神经元相连, 它向这些神经元发送数据。神经元之间不存在同层连接, 也不存在跨层连接。

如图2.2所示, 神经网络通常有三个部分: 一个输入层, 主要用于获取输入的信息; 一个或多个隐藏层, 主要进行特征提取, 调整权重让隐藏层的神经单元对某种

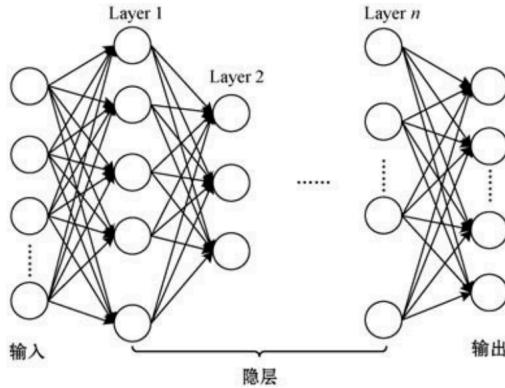


图 2.2: 深度神经网络结构图

模式形成反应；以及一个输出层，对接隐藏层并输出模型结果，调整权重以对不同的隐藏层神经元刺激形成正确的反应。当一个神经网络被训练时，其所有的权重和阈值最初都被设置为随机值。训练数据被送入输入层，并通过后续隐藏层，以复杂的方式相乘和相加，直到最后到达输出层，从根本上改变了数据。在训练过程中，不断调整网络的权重和阈值，直到具有相同标签的训练数据持续产生类似的输出。

### 2.2.2 前向传播算法

神经网络中层与层之间的“前馈传播”的算法简称为前向传播算法：网络中上一层的输出作为下一层的输入，并计算下一层的输出，一直到运算到输出层为止。如图2.3所示，假设现有输入层的训练数据为  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ,  $x_i \in R^d$ ,  $y_i \in R^l$ ，即输入样本由  $d$  个属性描述，输出是  $l$  维实值向量。

假设隐藏层神经元个数为  $q$  个， $\theta_j$  表示输出层神经元的阈值。隐藏层第  $h$  个神经元的阈值用  $\gamma_h$  表示。输入层第  $i$  个神经元与隐藏层第  $h$  个神经元之间的连接权为  $v_{ih}$ 。隐藏层第  $h$  个神经元与输出层第  $j$  个神经元之间的连接权为  $w_{hj}$ 。隐藏层第  $h$  个神经元接收到的输入为  $\alpha_h = \sum_{i=1}^d v_{ih}x_i$ ，输出层第  $j$  个神经元接收到的输入为  $\beta_j = \sum_{h=1}^q w_{hj}b_h$ 。其中  $b_h$  为隐藏层层第  $h$  个神经元的输出。假设隐层和输出层神经元都使用 Sigmoid 激活函数。对训练数据  $(x_k, y_k)$ ，假定神经网络的输出为

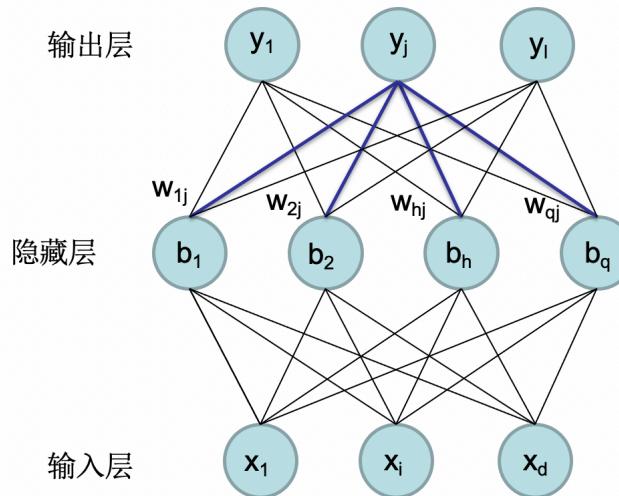


图 2.3: 前馈神经网络结构图

$\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$ , 即神经网络的预测输出表达式为:

$$\hat{y}_j^k = f(\beta_j - \theta_j) \quad (2.1)$$

那么如何评估神经网络输出的预测值与真实值之间的差异程度呢? 这里提出损失函数  $L$ , 本文采用均方差损失函数, 这种损失是通过计算实际(目标)值和预测值之间的平方差的平均值, 网络在样本集  $(x_k, y_k)$  上的均方误差为:

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2 \quad (2.2)$$

### 2.2.3 反向传播算法

深度神经网络算法训练的目的就是使得损失函数  $L$  最小, 通常采用随机梯度下降 (Stochastic Gradient Descent, SGD) 算法, 即在每次迭代过程中批量随机抽取训练样本  $(B)$ , 并计算损失函数  $L$  的偏导数  $g_B = \frac{1}{|B|} \sum_{x \in B} \nabla_{\theta} L(\theta, x)$ , 然后沿着负梯度方向  $-g_B$  朝着局部最小值更新权重系数  $\theta$ 。<sup>[29]</sup> 反向传播算法 (gradient descent)<sup>[30]</sup> 策略, 依据输出层的输出结果计算误差, 再将误差反向传播到隐藏层神经元, 最后依据隐层神经元的误差来对连接权和阈值进行调整<sup>[31]</sup>。

总的来说, 在深度神经网络中, 对每个训练样本, 通过前向传播算法从输入层、隐藏层到输出层依次训练, 在输出层得到预测的结果, 然后根据损失函数计算

预测值与真实值之间的差异程度，之后根据反向传播算法调整权重系数，更新网络参数，使得损失函数的值最小，模型达到全局最优。

## 2.3 差分隐私

差分隐私最初是由微软研究院在 2006 年<sup>[9]</sup> 针对统计数据库的隐私泄露问题提出的一种新的隐私定义，目的是使得数据库查询结果对于数据集中单个记录的变化不敏感。简单来说，就是单个记录在或者不在数据集中，对于查询结果的影响微乎其微。那么攻击者就无法通过加入或减少一个记录，观察查询结果的变化来推测个体的具体信息。

差分隐私首先被应用于数据查询，为了更好地说明数据集之间的差异，定义了相邻数据集的概念：两个数据集只差一个信息或只差一个数值不同的记录<sup>[28]</sup>。因此，查询数据库相关信息的攻击者将无法以任何概率确定  $X_n$  是否存在于数据集中，而成员  $X_n$  被认为是相对安全的。

### 2.3.1 基本定义

**定义 2.3.1** (邻近数据集). 现有数据集  $D$  和  $D'$ ，两者具有相同的属性结构，他们的对称差为  $D\Delta D'$ ,  $|D\Delta D'|$  表示  $D\Delta D'$  中记录的数量。若  $|D\Delta D'| = 1$ ，那么  $D$  和  $D'$  就是邻近数据集 (*Adjacent Dataset*)。

假设存在有限域  $Z$ ,  $z \in Z$  为  $Z$  中的元素，从有限域  $Z$  中抽样所得  $z$  组成数据集  $D$ ，数据的属性个数为维度  $d$ ，样本量为  $n$ 。对数据集  $D$  的查询即映射函数，用  $F = \{f_1, f_2, \dots\}$  来表示一组查询，算法  $M$  表示满足差分隐私的查询机制，它通过对查询  $F$  的结果进行处理，使之满足隐私保护的条件。

**定义 2.3.2** (差分隐私成立条件). 若随机算法  $M : D \rightarrow R$  满足  $(\varepsilon, \delta) - DP$ ，当且仅当相邻数据集  $d, d'$  对于算法  $M$  的所有可能输出子集  $S \in R$  满足不等式<sup>[40]</sup>：

$$\Pr[M(d) \in S] \leq e^\varepsilon \Pr[M(d') \in S] + \delta$$

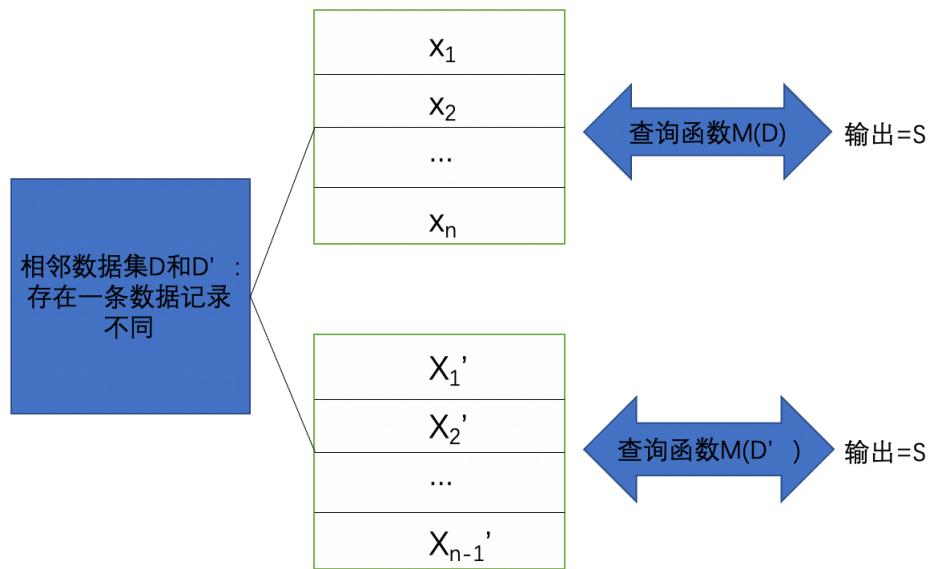


图 2.4: 差分隐私的相邻数据集示意图

其中,  $\varepsilon$  表示隐私预算参数,  $\varepsilon$  越接近 0 代表着数据集  $D$ ,  $D'$  上输出的数据分布越接近, 输出结果越不可区分, 隐私预算越低, 隐私保护的强度越高。添加值  $\delta$  代表允许以概率  $\delta$  打破  $\varepsilon - \text{DP}$  的可能性, 是用于限制模型行为任意改变的概率, 值通常选择小于  $1/|D|$ 。当  $\delta = 0$  时, 定义转化为  $\varepsilon - \text{DP}$ , 这时机制提供了更加严格的隐私保护。隐私保护强度取决于隐私预算参数, 在传统的数据保护领域, 如果  $\varepsilon \in (0, 1)$ , 那么此时隐私保护强度是有用的, 但是在深度学习方面, 当  $\varepsilon \in (0, 10)$  是才认为隐私保护强度是有效的。

### 2.3.2 相关概念

差分隐私保护的实现是在查询函数的返回值中注入一定量的干扰噪声, 但是注入的噪声量太大会影响最终结果的准确性, 太少则无法保障数据的隐私性。那么如何衡量添加的噪声量, 既能保障数据的安全, 又能维持数据的可用性呢? 这里针对数据集提出敏感度的概念, 加入的噪声量大小与数据集的敏感度息息相关。

对于数据集  $D$ , 它的敏感度是指在数据集  $D$  中任意删除一条记录, 对最终的查询结果产生的最大影响。数据集的敏感度决定了注入噪声量的大小。在差分隐

私中有两种敏感度，分别是全局敏感度和局部敏感度。

**定义 2.3.3 (全局敏感度).** 假设存在函数  $f : D \rightarrow R^d$ , 输入为一数据集, 输出为  $d$  维的实数向量。对于任意的邻近数据集  $D$  和  $D'$ ,

$$GS_f = \max_{D, D'} \|f(D) - f(D')\|_1$$

称为函数  $f$  的全局敏感度。

全局敏感度由函数本身决定, 反映了一个查询函数在一对相邻数据集上进行查询时变化的最大范围, 它取决于查询函数本身, 与数据集无关。不同的函数会有不同的全局敏感度。但是如果全局敏感度太大, 根据全局敏感度生成的噪声可能会过度保护数据, 因此提出了局部敏感性的定义。

**定义 2.3.4 (局部敏感度).** 对于一个查询函数  $f : D \rightarrow R^d$ , 其中  $D$  为一个数据集,  $R^d$  为  $d$  维实数向量, 是查询的返回结果。假使给定数据集  $D$ , 它有任意邻近数据集  $D'$ , 查询函数  $f$  在  $D$  上的局部敏感度定义为:  $LS_f(D) = \max_{D'} \|f(D) - f(D')\|_1$ 。

与全局敏感度不同, 局部敏感度是由给定的数据集和查询函数共同决定的。由于局部敏感度只是对于一个数据集做变化, 利用了数据集的数据分布特征, 局部敏感度通常要比全局敏感度小得多。但是, 也正是因为局部敏感度在某种程度上反映了数据集的分布特征, 所以直接应用局部敏感度生成噪声可能会泄漏数据的隐私信息。

敏感度代表了查询函数针对相邻数据集的输出的最大不同, 或者说量化评估了最坏情况下单个样本对整体数据带来的不确定性大小。敏感度函数仅与查询函数的类型有关, 给噪声的添加提供了依据。

在解决一个复杂的差分隐私保护问题时, 可能在多个场景, 多个步骤多次应用差分隐私技术, 在这种情况下, 如何保证最终结果的差分隐私性, 以及隐私保护的程度该如何去度量呢? 这里引出差分隐私的三个最重要的性质: 可量化性、可组合性和后处理不变性<sup>[35]</sup>。

可量化性指的是差分隐私算法在计算特定随机化过程时，可以透明化、精准量化所施加的噪声大小，即上文提及的隐私预算。这样使用者就可以清楚地知道算法的隐私保护力度；差分隐私的后处理不变性，确保了即使对算法的结果进行进一步处理，只要不引入额外信息，后续的处理就并不会削弱算法的隐私保护力度。组合性是指将相互独立的差分隐私算法进行组合<sup>[41]</sup> 后依然满足差分隐私，具体可分为并行组合和串行组合。

**定理 2.3.5 (串行组合).** 给定  $n$  个随机算法  $M_i (1 \leq i \leq n)$  满足  $\varepsilon_i - DP$ ，那么针对一个数据库  $D$  而言，在  $D$  上的算法串行序列组合满足  $\varepsilon - DP$ ，其中  $\sum_{i=1}^n \varepsilon_i = \varepsilon$ 。

**定理 2.3.6 (并行组合).** 如果数据库  $D$ ，划分成  $n$  个不相交的子集  $\{D_1, D_2, \dots, D_n\}$ ，在每个子集上应用算法  $M_i$ ，每个算法提供  $\varepsilon_i - DP$ ，则在序列  $\{D_1, D_2, \dots, D_n\}$  上整体满足  $(\max \{\varepsilon_1, \dots, \varepsilon_n\}) - DP$ 。

通过差分隐私的串并行组合定理，人们可以利用基础的差分隐私算法设计出复杂的满足差分隐私保证的系统，只要算法中的每一个步骤都满足差分隐私要求，那么这个算法的最终结果将满足差分隐私特性，这也是差分隐私的重要优势之一。

### 2.3.3 实现机制

在差分隐私的实际应用中，针对一个算法如何添加噪声使其能满足差分隐私保护的要求，在不同的场景和问题下有不同的差分隐私实现机制，主要分为拉普拉斯机制 (Laplace Mechanism)<sup>[9]</sup>、指数机制 (Exponential Mechanism)<sup>[32]</sup> 与高斯机制 (Gaussian Mechanism)<sup>[33]</sup>，这三种是最基础的差分隐私保护的实现机制。其中，指数机制适用于非数值型结果的隐私保护，拉普拉斯机制和高斯机制适用于对数值型结果的保护<sup>[35]</sup>。

首先我们介绍拉普拉斯机制如何实现差分隐私。Laplace 分布是统计学中的概念，是一种连续的概率分布。

**定理 2.3.7 (拉普拉斯机制).** 一个函数  $f : D \rightarrow R$ , 机制  $M$  满足  $\varepsilon - DP$ , 当:

$$M(D) = f(D) + \text{Lap} \left( \frac{\Delta f}{\varepsilon} \right)$$

其中, 噪声服从尺度参数满足  $\frac{\Delta f}{\varepsilon}$  的 Laplace 分布。

与拉普拉斯机制类似, 高斯机制对输入的所有维度添加高斯噪声干扰  $N(0, \sigma^2)$ 。

**定理 2.3.8 (高斯机制).** 对任意  $\epsilon \in (0, 1), \delta > (\sqrt{2 \ln(1.25/\delta)} f / \epsilon)$ , 有噪音  $Y \sim N(0, \delta^2)$ , 则满足  $(\epsilon, \delta)$ -差分隐私。

$$\Pr \left[ \mathcal{A}(D_1) = \tilde{D} \right] \leq e^\epsilon \times \Pr \left[ \mathcal{A}(D_2) = \tilde{D} \right] + \delta$$

其中,  $\epsilon$  表示隐私保护预算,  $\delta$  表示隐私保护的水平误差, 是一个较小的常数。当  $\delta = 0$  时, 称为  $(\epsilon, 0)$ -差分隐私。

但是对于离散型的查询结果或数据要如何处理呢? 这就产生了指数机制, 通常使用指数机制来随机选择离散的输出结果来满足差分隐私。指数机制整体的思想就是, 对于一个查询函数, 不是确定性的输出一个  $R_i$  结果, 而是以一定的概率值返回结果, 从而实现差分隐私。而这个概率值则是由打分函数确定, 得分高的输出概率高, 得分低的输出概率低。

**定理 2.3.9 (指数机制).** 指数机制满足差分隐私, 如果:

$$A(D, u) = \left\{ p : | \Pr[p \in O] \propto \exp \left( \frac{\varepsilon u(D, p)}{2\Delta u} \right) \right\}$$

其中  $\Delta u$  为评分函数  $u(D, p)$  的全局敏感性。由定理2.3.9可知, 评分越高, 则输出的概率越大。<sup>[?]</sup>。

## 2.4 本章小结

本章对论文需要使用的一些基础理论知识进行了讨论。主要介绍了深度神经网络的结构和算法、联邦学习系统的学习协议以及差分隐私的基本概念、定义和定理。分布式联邦学习系统是本论文主要使用的系统架构, 本文所针对的攻击模型和隐私保护方案都是基于该分布式联邦学习系统。

## 第三章 联邦学习中的自适应本地差分机制

### 3.1 引言

与传统的集中式深度学习相比，联邦学习通过分布式训练在一定程度上缓解了隐私泄漏的问题。然而，许多研究表明，攻击者仍然可以通过模型训练的梯度损害用户的隐私<sup>[48]</sup>。深度学习技术可以“记忆”模型中的训练数据信息，在这种情况下，敌方一旦通过白盒推理攻击或者黑盒推理攻击访问模型，就可以推演出客户端本地的训练数据。

在传统的集中式隐私保护方案中，数据管理者倾向于给每个用户的数据以相同的隐私预算。同样的隐私预算忽略了用户之间的差异。有些用户希望有更好的隐私保护。而有些用户对某些数据的隐私不敏感。在这种情况下，由于联邦学习模型是分布式结构，从一个大数据库到许多小数据库，所以对于每个用户来说。他们只需要关心他们自己的隐私。他们可以设置不同的隐私预算方案，而不是传统的统一分配，然后在最坏的情况下注入噪音。

联邦学习模型的优化问题可以概括为 ERM（经验风险最小化）问题<sup>[40]</sup>：

$$\arg \min_{\theta \in \mathcal{C}} \left( F(\theta) := \frac{1}{m} \sum_{i=1}^m F_i(\theta) \right) \quad (3.1)$$

从隐私保护的角度讲，我们只要截断了从原始输入到输出，在其中加入一道隐私保护屏障，具体在哪一步截断则对应于不同的方法。差分隐私保护机器学习的方法具体有以下几种：

- **输入扰动：**输入扰动是在获取的训练数据上直接添加噪声，之后的模型训练和优化都是基于加噪后的训练数据<sup>[?][?][?]</sup>。

- **输出扰动:** 输出扰动沿袭了拉普拉斯机制最简单的思路, 即考虑函数输出的敏感度来添加噪声, 那么在 ERM 公式中我们只需要考虑  $\text{argmin}$  函数输出的敏感度, 基于这个敏感度来添加拉普拉斯噪声即可得到一个简单的满足差分隐私的 ERM 方法<sup>[36]</sup>。
- **梯度扰动:** 梯度扰动是在执行最小化损失函数的过程中, 设计满足差分隐私的算法。
- **目标扰动:** 目标扰动是在模型的目标函数中添加一个随机量, 以使得最终模型的输出满足随机性。

基于输入的扰动和输出的扰动基本可以视为一个黑匣子模型, 简单直接。但是这种添加噪声的方式无法对训练过程中数据的相互依赖性和输出有效性作出有用的、紧密的描述。在输入数据中加入过多的噪声, 可能会影响模型训练的收敛性。在输出参数中加入过于保守的噪声, 也就是根据最坏的攻击情况去添加噪声, 可能会影响模型的实用性。

当前在深度学习模型中应用差分隐私的主流方案是在模型的梯度上添加噪声, 方案的目标是在满足差分隐私的条件下, 实现整体模型的最优可用性。Song 等人<sup>[47]</sup>提出了一个  $(\epsilon_c + \epsilon_d)$ -差分隐私版本的随机梯度下降算法。在模型的每一次迭代过程中, 对梯度添加高斯噪声, 并通过差分隐私的组合性和隐私放大效果, 得到完全隐私损失的上界。传统的联邦学习中使用差分隐私的主要流程如下所示:

- **本地计算:** 客户端  $i$  根据本地数据库  $D_i$  和接受的服务器的全局模型  $w_G^t$  作为本地的参数, 即  $w_i^t = w_G^t$ , 进行梯度下降策略进行本地模型训练得到  $w_i^{t+1}$  ( $t$  表示当前通信回合)。
- **模型扰动:** 每个客户端产生一个随机噪音  $n$ ,  $n$  是符合高斯分布的, 使用  $\bar{w}_i^{t+1} = w_i^{t+1} + n$  扰动本地模型 (这里注意  $w$  是一个矩阵, 那么  $n$  就对矩阵的每一个元素产生噪音)。

- 模型聚合: 服务器使用 FedAVG 算法聚合从客户端收到的  $\bar{w}_i t + 1$  得到新的全局模型参数  $w_G^{t+1}$ , 也就是扰动过的模型参数。
- 模型广播: 服务器将新的模型参数广播给每个客户端。
- 本地模型更新: 每个客户端接受新的模型参数, 重新进行本地计算。

然而, 传统的基于差分隐私的联邦学习框架对所有本地数据添加相同的干扰, 难免降低了全局模型的精度。因此本文采用一种更加复杂的方法来分析训练过程中训练数据对模型输出的贡献比率, 然后根据每一层神经网络对模型输出的贡献率, 在梯度上自适应添加噪声。

在本文中, 我们认为中央参数服务器是半可信的 (Honest but Curious, HbC), 一个“诚实但好奇”的实体。也就是说, 服务器将遵循与所有用户的协议。然而, 通过利用完全访问用户梯度的便利, 它也试图在训练过程中获得关于客户端的额外的信息。出于这个原因, 我们提出的自适应加噪机制目的是保护发送到服务器的本地梯度不被推断出任何关于用户的额外信息, 并且尽量维持原有模型的精度。

总的来说, 本章提出的隐私保护方案是基于本地客户端的本地数据维度的, 从以下三个方面展开研究: 第一, 通过在本地模型训练的梯度下降算法过程中针对不同层的贡献比自适应添加噪声; 第二, 采用解析高斯机制, 计算对其梯度施加的噪声大小; 第三, 使用差分隐私的组合定理和后处理定理分析模型整体的隐私性。

### 3.2 基于自适应差分隐私的随机梯度下降算法

算法1详细描述了在本地客户端训练过程中, 在 SGD 算法中添加自适应差分隐私, 并使用解析高斯机制衡量所添加的噪声大小。首先, 我们采用先验组合机制计算  $\epsilon_{iter}$  和  $\delta_{iter}$  (算法第 7 行)。每个客户端对训练数据进行采样, 并计算他们的隐私预算  $\delta_u$ 。如果  $\delta_u > \delta$ , 用户将终止采样和训练, 并且不上传其梯度信息 (算法第 9-12 行)。否则, 用户将计算梯度的贡献率 (算法第 16 行), 接着使用拉普拉斯机制注入自适应的噪声量 (算法第 18 行), 然后对梯度进行范数裁剪 (算法第 19

行)。最后,服务器对用户的梯度进行平均,并更新模型参数  $w$ 。该算法有三个主要部分:自适应差分隐私,梯度范数裁剪,以及采用解析高斯计算累积添加的噪声量。

---

**Algorithm 1** 基于自适应差分隐私的随机梯度下降算法
 

---

```

1: 输入: 预估迭代次数  $T$ , 学习率  $\alpha$ , 梯度裁剪阈值  $C$ , 目标损失函数  $l$ , 解析高斯机制噪声
    $(\Delta, \varepsilon, \delta)$ ,  $f$  贡献率阈值,  $p$  注入噪声的概率
2: 输出: 模型梯度
3: 初始化模型权重  $w$ 
4: while  $\exists \delta_u < \delta$  do
5:    $n=0$ 
6:    $grad=0$ 
7:   计算  $eps_{iter}$ ,  $\delta_{iter}$ 
8:   for each  $u \in Users$  do
9:     计算  $\delta_u$ 
10:    if  $\delta_u > \delta$  then
11:      continue
12:    end if
13:    从客户端数据集中随机采样
14:     $gt_u = \nabla l(w, x)$ 
15:    计算梯度的平均贡献率
16:     $C_j(x_i) = \frac{1}{n} \sum_{i=1}^n C_{x_{i,j}}(x_i), j \in [1, u]$ 
17:    添加自适应噪声
18:     $gt'_u = gt_u + \frac{1}{|D_i^t|} \text{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$ 
19:     $gt_u = gt_u / \max\left(1, \frac{\|gt_u\|}{C}\right)$ 
20:     $n++$ 
21:  end for
22:   $w = w - \alpha * grad/n$ 
23: end while
  
```

---

### 3.3 详细设计

在本节接下来的三个部分,我们将详细描述如何在神经网络的随机梯度下降算法中自适应添加噪声、梯度剪裁以及使用解析高斯机制衡量添加的噪声大小。

### 3.3.1 自适应噪声添加

在第二章我们详细介绍了深度神经网络的结构，每个用户在本地用原始数据进行训练，在神经网络中进行前向传播操作，得到本地模型的输出。输入层的前向传播是神经网络中前向传播算法的第一步。

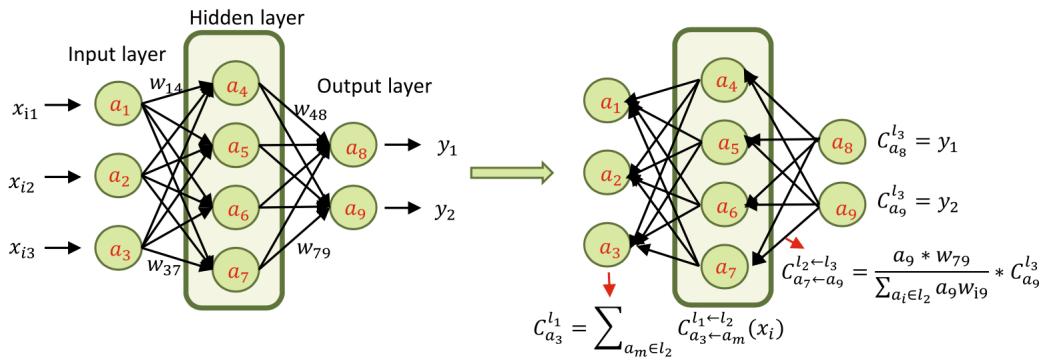


图 3.1: 层间依赖传播算法

我们采用层间相关性传播算法计算每一层对模型输出的贡献比率。每个用户都在本地对原始数据在神经网络中进行前向传播的训练，这可以获得一个新的数据操作，从而获得本地模型的输出。

” $\leftarrow$ ” 表示两部分之间的连接关系。” $l_2 \leftarrow l_3$ ” 是指深度神经网络中第 2 层和第 3 层之间相邻层的连接关系。当第 k 层为输出层时，我们有：

$$C_{a_i}^{l_k}(x_i) = f(x_i, \omega_i^r) \quad (3.2)$$

根据矩阵层之间的线性相关性，神经元  $a_i$  在第 k 层的贡献  $C_{a_i}^{l_k}(x_i)$  等于连接到神经元  $a_i$  的相邻层的贡献之和：

$$C_{a_i}^{l_k}(x_i) = \sum_{a_j \in l_{k+1}} C_{a_i \leftarrow a_j}^{l_k \leftarrow l_{k+1}}(x_i) \quad (3.3)$$

比如，在图3.1中，存在：

$$C_{a_7}^{l_2}(x_i) = \sum_{a_j \in l_3} C_{a_7 \leftarrow a_j}^{l_2 \leftarrow l_3}(x_i) = C_{a_7 \leftarrow a_8}^{l_2 \leftarrow l_3}(x_i) + C_{a_7 \leftarrow a_9}^{l_2 \leftarrow l_3}(x_i) \quad (3.4)$$

因此，神经元  $a_j$  对于输出层的贡献等于模型的输出。第  $k$  层的神经元  $a_j$  对于第  $k-1$  层的神经元  $C_{a_i \leftarrow a_j}^{l_{k-1} \leftarrow l_k}(x_i)$  等于：

$$C_{a_i \leftarrow a_j}^{l_{k-1} \leftarrow l_k}(x_i) = \begin{cases} \frac{a_i w_{i,j}}{\sum_{a_i \in l_{k-1}} a_i w_{i,j}} C_{a_j}^{l_k}(x_i) & \sum_{a_i \in l_{k-1}} a_i w_{i,j} \neq 0 \\ \mu & \sum_{a_i \in l_{k-1}} a_i w_{i,j} = 0 \end{cases} \quad (3.5)$$

其中  $\mu$  是一个无限接近于零，但大于零的数字。从上述公式中，我们可以认为每一层的贡献是相等的，而且贡献是逐层传递的。根据以上公式的推导，我们能得到神经网络模型中每一层以及每个神经元的贡献值。

通过从数据元组中提取同一属性的贡献，我们可以计算出每个属性类对模型输出的平均贡献：

$$C_j(x_i) = \frac{1}{n} \sum_{i=1}^n C_{x_{i,j}}(x_i), j \in [1, u] \quad (3.6)$$

在原始的参数上计算神经网络中每个属性类对于模型输出的贡献后，按照公式3.7采用拉普拉斯机制在属性类的贡献率中注入噪音以保护原始的参数。

$$\tilde{C}_j(x_i) = C_j(x_i) + \text{Lap}\left(\frac{GS_c}{\epsilon_c}\right), j \in [1, u] \quad (3.7)$$

其中，函数的局部敏感度为  $GS_c = \frac{2u}{|D|}$ ， $u, |D|$  分别代表了属性和数据元组的最大数量。

在第二章中我们介绍了关于神经网络的结构，

$$y = a(\mathbf{x} * \omega + b) \quad (3.8)$$

公式3.8表示学习模型中每个隐藏神经元的转化过程。其中  $\mathbf{x}$  代表输入向量， $y$  是输出， $b$  和  $\omega$  分别代表偏置项和权重矩阵。 $a()$  是一个激活函数，用于结合线性变换和非线性变换。 $y = a(\mathbf{x} * \omega + b)$  是线性变换部分。

由于神经网络的结构，上一层的输出是下一层的输入，由此我们可以得出，原始的训练数据只被第一隐层的线性变换所利用。直观地说，为了得到一个具有隐私保护的学习模型，我们可以在第一层隐藏层的数据中注入噪声。正如 Phan 等人<sup>[36]</sup>提到的，对于线性变换有一种传统的方法，即向原始数据注入具有相同隐私预算

的噪声，但是这容易导致隐私预算增加，并且使原始数据失真过多。因此，本文提出一种自适应噪声添加算法，针对每个梯度计算其贡献值，根据贡献值进行梯度裁剪并添加噪声。

首先，我们引入了两个调整因素  $f$  和  $p$ 。其中， $f$  代表一个阈值，用于决定属性对模型结果输出的贡献是高还是低，其值由用户定义，即贡献超过阈值  $f$  的属性类对输出的贡献更大。然后，我们向所有这些属性注入自适应拉普拉斯噪声。当贡献率低于阈值  $f$  时，对这些属性进行概率选择。也就是说，我们选择概率为  $1 - p$  的原始数据，并对一些概率为  $p$  的属性注入自适应拉普拉斯噪声。该公式如下：

$$\tilde{x}_{i,j} = \begin{cases} \ddot{x}_{i,j} & \beta \geq f \\ \bar{x}_{i,j} & \beta < f \end{cases} \quad (3.9)$$

其中  $\beta$  代表贡献率： $\beta = \frac{|\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|}$ ，当  $\beta < f$  时，我们有：

$$\bar{x}_{i,j} = \begin{cases} \ddot{x}_{i,j} \text{ with probability } p \\ x_{i,j} \text{ with probability } 1 - p \end{cases} \quad (3.10)$$

$f$  和  $p$  是超参数，用户可以根据自己的情况来调整。

也就是说，隐私预算  $\epsilon_l$  是根据贡献率： $\epsilon_j = \frac{u * |\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|} * \epsilon_l$  按比例分配给每个属性类，自适应噪声按以下方式注入属性中：

$$x'_{i,j} = x_{i,j} + \frac{1}{|D_i^t|} \text{Lap} \left( \frac{G S_l}{\epsilon_j} \right) \quad (3.11)$$

在不丧失一般性的情况下，调整因子  $f$  和  $p$  的值与系统的准确性和隐私水平有关。即  $f$  越小， $p$  越大。越高的秘密水平，准确性越低，反之亦然。当  $f$  值为 0.15， $p$  设置为 0.85 时，使用自适应的噪声分布与拉普拉斯机制基本吻合。我们可以得出一个令人信服的结论，在相同的噪声水平下，自适应噪声添加的隐私预算接近于原始的拉普拉斯机制，因此我们的方案在缩小调整系数范围的情况下能达到相近的隐私保护效果。

### 3.3.2 梯度范数裁剪

在模型的每一轮迭代过程中，算法将计算添加了拉普拉斯噪声的梯度  $gt'_u = gt_u + \frac{1}{|D_i^t|} \text{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$ ，方差是  $2b^2$ 。对梯度注入的噪声量  $\frac{1}{|D_i^t|} \text{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$ ，决定于用户个体对于梯度  $g$  在二范数下的最大全局敏感度，即  $\delta$ 。由于梯度的大小没有一个先验的界限，我们采用二范数的固定值对每个梯度进行裁剪。

用户上传的梯度向量将可以改写为  $gt_u = gt_u / \max\left(1, \frac{\|gt_u\|}{C}\right)$ ，其中  $C$  是裁剪阈值。对于梯度的裁剪能保证梯度值小于设定的阈值  $i$ 。也就是当  $\|g\| \leq C$ ， $g$  保持不变；当  $\|g\| > C$  时，它会按照裁剪比例缩小为  $C$ 。

但是如果裁剪阈值  $C$  的值如果太小，那么裁剪后的噪声会较小，算法添加的噪声较小时可能会破坏梯度估计的无偏性；可是如果不对梯度进行裁剪，大量的噪声添加到每个梯度会导致模型的可用性大大降低。在模型训练前期，梯度所包含的数据信息更多，因此可以对应添加更多的高丝噪声，使用较大的  $C$  的值，使得梯度裁剪后的模型偏差更小；而在模型训练后期，梯度所包含的数据信息相对较小了，如果还使用相同的  $C$ ，会引入很多不必要的噪声。

因此我们根据训练轮数和层间贡献率动态调整梯度裁剪阈值  $C$ ：在每次迭代中，该算法使用方差为  $2b^2$  的拉普拉斯机制来计算噪声梯度  $gt'_u = gt_u + \frac{1}{|D_i^t|} \text{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$ 。噪声  $2b^2$  的大小取决于一个个体在  $l_2$  规范下对  $g$  的最大影响，即  $\delta$ 。由于对梯度的大小没有先验的约束，我们以  $l_2$  规范对每个梯度进行剪辑。因此，梯度向量  $g$  被  $gt_u = gt_u / \max\left(1, \frac{\|gt_u\|}{C}\right)$  取代，以达到剪裁阈值  $C$ 。这种剪裁保证了如果  $\|gt_u\| \leq C$ ，那么  $gt_u$  将被保留，而如果  $\|g\| > C$ ，它将被裁减为梯度准则  $C$ 。

### 3.3.3 解析高斯机制

在第二章的基础知识中，我们简要介绍了传统的高斯机制。它的定义如下：

**定义 3.3.1.** 对于任意  $\varepsilon \in (0, 1)$  与  $c^2 > 2 \ln(1.25/\delta)$ ，高斯噪声参数满足  $\sigma \geq c\Delta_2 f/\varepsilon$  的高斯干扰机制为  $(\varepsilon, \delta)$ -差分隐私。

在此基础上我们自然的有两个疑问：一是是否定义中的参数  $\sigma$  是使算法满足

$(\varepsilon, \delta)$ -差分隐私的最小值, 即是否可以施加更小的干扰来达到相同的差分隐私保护效果; 二是如果隐私预算  $\varepsilon$  大于 1 会发生什么。

Balle<sup>[44]</sup> 等人分析了传统的高斯机制在高隐私保护力度、隐私损失分析、低隐私保护力度三个方面中存在的问题, 提出了一种改进的解析高斯机制 (Analytic Gaussian Mechanism)。传统的高斯机制施加的噪声干扰过大, 方差公式在高隐私预算下过紧, 并且无法适用于隐私预算大于 1 的低隐私保护力度场景。解析高斯机制针对这些局限性提出了解决方案。

解析高斯机制的核心思想是使用高斯累积分布函数的计算, 来代替传统高斯机制的尾部约束近似。它的定义如下:

**定义 3.3.2.** 令  $f : \mathbb{N}^{\mathbb{N}^d} \rightarrow \mathbb{R}^d$  表示一个  $\ell_2$  敏感度为  $\Delta_2$  的函数, 对任意  $\varepsilon \geq 0$  与  $\delta \in [0, 1]$ , 当且仅当  $\sigma$  满足下列不等式时, 含参数  $\sigma$  的高斯机制满足  $(\varepsilon, \delta)$ -差分隐私:

$$\Phi\left(\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}\right) - e^\varepsilon \Phi\left(-\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}\right) \leq \delta \quad (3.12)$$

其中,  $\Phi(t) = (1 + \text{erf}(t/\sqrt{2}))/2$ , erf 是高斯误差函数, 即  $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\eta^2} d\eta$ 。传统的差分隐私高斯机制的干扰大小  $\sigma$  可以直接计算, 解析高斯机制的实现如下算法:

### 3.4 隐私性证明

自适应差分 SGD 算法对线性变换函数进行了扰动, 该函数满足  $(\epsilon_c + \epsilon_l)$  差分隐私。证明如下:

在贡献中添加的扰动为:

$$\ddot{C}_j(x_i) = C_j(x_i) + \text{Lap}\left(\frac{GS_c}{\epsilon_c}\right), j \in [1, u] \quad (3.13)$$

它是满足  $\epsilon_c$ -差分隐私的。

**Algorithm 2** 解析高斯算法

---

```

1: 输入:  $f, x, \Delta, \varepsilon, \delta$ 
2: 输出: 干扰后的  $f$ 
3: 令  $\delta_0 = \Phi(0) - e^\varepsilon \Phi(-\sqrt{2\varepsilon})$ 
4: if 如果  $\delta \geq \delta_0$  then
5:   定义  $B_\varepsilon^+(v) = \Phi(\sqrt{\varepsilon v}) - e^\varepsilon \Phi(-\sqrt{\varepsilon(v+2)})$ 
6:   计算  $v^* = \sup \{v \in \mathbb{R}_{\geq 0} : B_\varepsilon^+(v) \leq \delta\}$ 
7:   令  $\alpha = \sqrt{1+v^*/2} - \sqrt{v^*/2}$ 
8: else
9:   定义  $B_\varepsilon^-(u) = \Phi(-\sqrt{\varepsilon u}) - e^\varepsilon \Phi(-\sqrt{\varepsilon(u+2)})$ 
10:  计算  $u^* = \inf \{u \in \mathbb{R}_{\geq 0} : B_\varepsilon^-(u) \leq \delta\}$ 
11:  令  $\alpha = \sqrt{1+u^*/2} + \sqrt{u^*/2}$ 
12: end if
13: 令  $\sigma = \alpha \Delta / \sqrt{2\varepsilon}$ 
14: 输出:  $f(x) + \mathcal{N}(0, \sigma^2 I)$ 

```

---

贡献  $GS_c$  的敏感度为:

$$\begin{aligned}
GS_c &= \frac{1}{|D|} \sum_{j=1}^u \left\| \sum_{x_i \in D} C_{x_{i,j}}(x_i) - \sum_{x'_i \in D'} C_{x'_{i,j}}(x'_i) \right\|_1 \\
&= \frac{1}{|D|} \sum_{j=1}^u \left\| C_{x_{n,j}}(x_n) - C_{x'_{n,j}}(x'_n) \right\|_1 \\
&\leq \frac{2}{|D|} \max \sum_{j=1}^u \left\| C_{x_{i,j}}(x_i) \right\|_1 \\
&\leq \frac{2u}{|D|}
\end{aligned} \tag{3.14}$$

其中,  $u$  和  $|D|$  分别表示贡献的数量和元组, 然后可以得到:

$$\begin{aligned}
 \frac{\Pr(\tilde{C}(D))}{\Pr(\tilde{C}(D'))} &= \frac{\prod_{j=1}^u \exp\left(\frac{\epsilon_c \left\|\frac{1}{|D|} \sum_{x_i \in D} C_j(x_i) - \tilde{C}_j(x_i)\right\|_1}{GS_c}\right)}{\prod_{j=1}^u \exp\left(\frac{\epsilon_c \left\|\frac{1}{|D'|} \sum_{x'_i \in D'} C_j(x'_i) - \tilde{C}_j(x'_i)\right\|_1}{GS_c}\right)} \\
 &= \prod_{j=1}^u \exp\left(\frac{\epsilon_c}{|D|GS_c} \|C_j(x_n) - C_j(x'_n)\|_1\right) \\
 &\leq \prod_{j=1}^u \exp\left(\frac{\epsilon_c}{|D|GS_c} \max \|C_j(x_n)\|_1\right) \\
 &= \exp\left(\epsilon_c \frac{\max_{x_i \in D} \sum_{j=1}^u \|C_j(x_n)\|_1}{|D|GS_c}\right) \\
 &\leq \exp(\epsilon_c)
 \end{aligned} \tag{3.15}$$

因此, 添加噪声后的贡献值是满足  $\epsilon_c$ -差分隐私的。

假设两个相邻的批次  $D_i^t$  和  $D_i^{t'}$ , 其最后一个元组  $x_n$  和  $x'_n$  不同,  $z(D_i^t)$  和  $z(D_i^{t'})$  分别为线性变换函数。

一般来说, 我们把偏置项视为第一类数据属性, 即:  $x_{i,0} = b_i$ 。线性转换可以改写为:  $\ddot{\mathbf{z}}_{x \in D_i^t}(\omega) = \ddot{\mathbf{x}} * \omega$ 。线性变换的敏感性  $GS_l$  如下:

$$\begin{aligned}
 GS_l &= \sum_{a_i \in l_1} \sum_{j=1}^u \left\| \sum_{x_i \in D_i^t} x_{i,j} - \sum_{x'_i \in D_i^{t'}} x'_{i,j} \right\|_1 \\
 &= \sum_{a_i \in l_1} \sum_{j=1}^u \|x_{n,j} - x'_{n,j}\|_1 \\
 &\leq \sum_{a_i \in l_1} \sum_{j=1}^u \max_{x_i \in D_i^t} \|x_{n,j}\|_1 \\
 &\leq \sum_{a_i \in l_1} u
 \end{aligned} \tag{3.16}$$

其中,  $a_i \in l_1$  是指第一隐藏层  $l_1$  中的神经元  $a_i$ ,  $u$  是数据元组  $x_i \in D_i^t$  中的属性数。它包括两个调整因素:  $f$  和  $p$ , 它们可以过滤多余的噪声。之后的属性的一

般表达式如下：

$$\begin{aligned}
 \tilde{x}_{i,j} &= [(1-f) + f * p] * \ddot{x}_{i,j} + f * (1-p) * x_{i,j} \\
 &= [(1-f) + f * p] \left[ x_{i,j} + \text{Lap} \left( \frac{GS_l}{\epsilon_j} \right) \right] + [f * (1-p)] x_{i,j} \\
 &= x_{i,j} + [(1-f) + f * p] \left[ \text{Lap} \left( \frac{GS_l}{\epsilon_j} \right) \right]
 \end{aligned} \quad (3.17)$$

然后我们可以得到：

$$\begin{aligned}
 \frac{\Pr(\ddot{\mathbf{z}}_{D_i^t}(\omega))}{\Pr(\ddot{\mathbf{z}}_{D_i^{t'}}(\omega))} &= \frac{\prod_{a_i \in l_1} \prod_{j=1}^u \exp \left( \frac{\epsilon_j \left\| \sum_{x_i \in D_i^t} x_{i,j} - \sum_{x_i \in D_i^{t'}} \tilde{x}_{i,j} \right\|_1}{GS_l} \right)}{\prod_{a_i \in l_1} \prod_{j=1}^u \exp \left( \frac{\epsilon_j \left\| \sum_{x'_i \in D_i^{t'}} x'_{i,j} - \sum_{x'_i \in D_i^t} \tilde{x}'_{i,j} \right\|_1}{GS_l} \right)} \\
 &\leq \prod_{a_i \in l_1} \prod_{j=0}^u \exp \left( \frac{\epsilon_j}{GS_l} \left\| \sum_{x_i \in D_i^t} x_{i,j} - \sum_{x'_i \in D_i^{t'}} x'_{i,j} \right\|_1 \right) \\
 &\leq \prod_{a_i \in l_1} \prod_{j=0}^u \exp \left( \frac{\epsilon_j}{GS_l} \max_{x_i \in D_i^t} \|x_{i,j}\|_1 \right) \\
 &\leq \exp \left( \epsilon_l \frac{\sum_{a_i \in l_1} u \left[ \sum_{j=1}^u \frac{|\ddot{C}_j|}{\sum_{j=1}^u |\ddot{C}_j|} \right]}{GS_l} \right) \\
 &= \exp(\epsilon_l)
 \end{aligned} \quad (3.18)$$

根据上述推倒证明可知，在联邦学习的神经网络中添加自适应噪声后，所上传的梯度是满足  $(\epsilon_c + \epsilon_l)$  差分隐私的。在满足差分隐私的基础上，在下一节我们会给出隐私损失累积函数计算隐私成本。

### 3.5 隐私预算分析

本章所提出的自适应差分隐私保护方案是通过在随机梯度下降算法上添加自适应的拉普拉斯扰动，保护数据的隐私性。在上一节我们已经证明了此算法满足  $(\epsilon_c + \epsilon_l)$  差分隐私，那另外一个非常重要的问题就是评估在训练过程中添加噪声所累积的隐私预算成本。在本节中，我们提出隐私损失累积函数的概念，去计算算法

迭代过程中添加噪声所累积的隐私预算成本。

根据差分隐私的组成定理，被查询  $n$  次的数据的隐私风险将增加  $n$  倍。因此，我们希望查询次数越少越好，至少要有一个界限。在实验环境中，我们可以通过几次尝试确定一个相对理想的迭代次数，然后在每次迭代中平均分配隐私预算。然而，在实践中很难选择迭代的数量，因为任何尝试都会增加额外的隐私风险。当数量太小时，会发生拟合，导致性能不佳；如果数量太大，注入的噪声会过大，这将影响模型的准确性。另一种尝试是使用等比例递增的注入噪声序列，这样无论我们有多少次迭代，我们都能找到有限的隐私预算<sup>[32]</sup>。

根据上文提出的解析高斯机制，结合差分隐私的组成定理，我们设计了一个解析高斯组成定理：

**定理 3.5.1**(解析高斯组成定理). 假使存在算法  $M_i$  满足  $(\varepsilon_i, \delta_i)$ -差分隐私，那么对于  $M_{[k]} = (M_1, M_2, \dots, M_k)$ ，有  $M_{[k]}$  也是满足  $(\varepsilon, \delta)$ -差分隐私的，其中

$$\delta = \sum_{i=1}^k B(i, k, p) \left[ \Phi \left( \frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) - e^\varepsilon \Phi \left( -\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) \right]$$

我们采用朴素贝叶斯机制计算  $\delta$  可以得到：

$$\delta = \sum_{i=1}^k B(i, k, p) [\Pr [L_{d,d'} * i > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * i < -\varepsilon]] \quad (3.19)$$

在此情况下，隐私损失变量  $L_{M,d,d'}$  和  $L_{M,d',d}$  同时满足  $N(\eta, 2\eta)$  分布，并且  $\eta = H^2/2\sigma^2$ 。因此可以采用解析高斯机制这样表达隐私预算损失：

$$\begin{aligned} \Pr [L_{d,d'} * i > \varepsilon] &= \Pr [N(\eta i, 2\eta i) > \varepsilon] \\ &= \Pr \left[ N(0, 1) > \frac{-\eta i + \varepsilon}{\sqrt{2\eta i}} \right] = \Pr \left[ N(0, 1) < \frac{\eta i - \varepsilon}{\sqrt{2\eta i}} \right] \\ &= \Pr \left[ N(0, 1) < \sqrt{\frac{\eta i}{2}} - \frac{\varepsilon\sigma}{\sqrt{2\eta i}} \right] = \Phi \left( \frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) \end{aligned} \quad (3.20)$$

然后，可以计算得到隐私预算：

$$\delta = \sum_{i=1}^k B(i, k, p) \left[ \Phi \left( \frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) - e^\varepsilon \Phi \left( -\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) \right] \quad (3.21)$$

因为随着算法迭代次数  $T$  的增加,  $\Phi\left(\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}}\right) - e^\varepsilon \Phi\left(-\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}}\right)$  也在增加, 因此可以计算得到:

$$\begin{aligned} & \sum_{i=1}^T B(i, T, p) [\Pr [L_{d,d'} * i > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * i < -\varepsilon]] \\ & \leq \sum_{i=1}^T B(i, T, p) [\Pr [L_{d,d'} * T > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * T < -\varepsilon]] \\ & \leq \Pr [L_{d,d'} * T > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * T < -\varepsilon] \end{aligned} \quad (3.22)$$

当隐私预算  $\delta$  相同时,  $\sigma_1 \leq \sigma_2 \leq \sigma_3$ 。替换  $\sigma = \alpha H \sqrt{T} / \sqrt{2\epsilon}$ , 则有

$$\Phi(\sqrt{\epsilon/2}(1/\alpha - \alpha)) - e^\varepsilon \Phi(-\sqrt{\epsilon/2}(1/\alpha + \alpha)) \leq \delta \quad (3.23)$$

因此,  $\sigma_1 \leq \sigma_3 = O(\sqrt{T})$ 。与之前的工作相比, 我们的隐私预算能够在相同的迭代次数  $T$ , 更低的上界, 达到满足  $(\epsilon_c + \epsilon_l)$  的差分隐私。

### 3.6 本章总结

联邦学习以分布式学习技术为基础, 使参与者彼此通过一定的方式(如中心服务器)联合起来训练一个神经网络。在这个过程中, 参与者不需要将自己的隐私数据暴露出来便可以参与协作训练, 可以克服参与者本地数据集较小、数据样本比较单一、隐私泄露等缺点。虽然基本的分布式协作深度学习没有直接暴露参与者的隐私数据集, 但是恶意攻击者仍然可以通过共享的参数等信息获得一定的隐私信息。

本章详细介绍了如何在深度学习模型的随机梯度下降算法中添加自适应的拉普拉斯噪声。其中梯度下降作为一种常见的深度学习优化方法, 在梯度上添加扰动是最早被提出、也是当前比较主流的差分隐私加噪方案之一。我们设计了一个自适应噪声添加的方案, 在神经网络前向传播算法中, 根据属性对于模型输出的贡献率注入不同隐私预算的噪声。与传统的注入噪声的方法相比, 我们在相同的隐私保护程度下最大限度地提高了模型的准确性, 并且证明了算法能满足  $(\epsilon_c + \epsilon_l)$  差分隐私。然后我们采用解析高斯机制计算对梯度施加的噪声量大小, 分析了模型整体的隐私预算。

然而，在客户端的本地数据集添加的噪声只能保证本地数据的匿名性，不能够防止外部攻击者针对通信信道的攻击。如果客户端在每次迭代中同时上传了大量的权重更新，中央云服务器仍然可以将它们链接在一起，推导出参数信息。而且，当参与一次迭代的客户端数量达到上千人时，会导致聚合任务升级成一个高维任务，隐私预算暴增。因此，下一章我们对联邦学习模型框架进行了改进，在现有的联邦学习模型上新增混洗器，实现联邦学习框架的隐私安全，提高整体联邦学习模型的精度。

## 第四章 联邦学习的安全混洗模型

### 4.1 引言

上一章节中所提出的本地自适应差分隐私方案是通过在客户端将梯度上传至参数服务器前，对梯度添加自适应噪声，尽管方案采用了本地差分技术减少一定程度的隐私预算，但不可避免的会降低联邦学习模型的准确性以及学习效率。正如<sup>[49]</sup>所指出的，一个复杂的隐私保护系统将多个本地差分隐私的算法进行组合，从而导致这些算法的隐私成本增长。也就是说，隐私预算为  $\epsilon_1$  和  $\epsilon_2$  的局部差异化算法的组合会消耗的隐私预算总和为  $\epsilon_1+\epsilon_2$ 。使用联邦学习训练的联合模型需要客户在多次迭代中向中央服务器上传梯度更新。如果在迭代训练过程中的每一次迭代都应用本地自适应差分隐私，隐私预算就会累积起来，从而导致总隐私预算的爆炸。现有的本地差分隐私协议对于多维聚集的联邦学习框架可能是不可行的，局部噪声带来的误差会随着维度系数的增加而加剧，从而大大降低模型的精度。而且，当参与一次迭代的客户端数量达到上千人时，会导致聚合任务升级成一个高维任务，隐私预算暴<sup>[43]</sup>。而且，值得关注的是，不同的用户有不同的隐私需求，不同的用户上传的梯度对于联合模型的贡献比也有差异，因此本章将提出混合差分隐私技术，构造一个全新的可信第三方——混洗器，与本地差分隐私相结合，实现的方案能提高全局模型的精度，也保证在更低的隐私成本下达到相同的隐私预算。

在本章节中我们提出了一个在联邦学习中的安全混洗器，本地客户端使用自适应差分隐私对于模型的输出进行加噪，然后安全混洗器从客户端上传的样本中随机采样，将收集到的梯度以维度进行拆分，打乱次序，达到隐私放大效果，再采用梯度稀疏化的技术筛选对联合模型贡献较高的梯度，发送给中央服务器进行聚

合。安全混洗器作为一个可信第三方，独立于服务器并专门用于本地客户端梯度的子采样、混洗、上传。这个模型通过子采样和混洗两者的结合达到隐私放大效应，从而提高了整体联邦学习模型的精度。当本地差分隐私添加更少的噪音时，对于同样的中央服务器能达到相同水平的隐私预算。

我们将在本章节详细的描述该框架中各个模块的设计和实现过程。

## 4.2 安全混洗模型

如图4.1所示，该框架主要由本地客户端、混洗器和中央服务器 3 部分组成：

- 本地客户端：基于第三章的本地自适应差分隐私方案，在模型训练的梯度下降算法中对梯度进行自适应的扰动，得到满足  $(\epsilon_c + \epsilon_l)$  差分隐私的梯度。
- 混洗器：一个半诚信的第三方。首先动态采样本地客户端上传的梯度，然后借助现有的安全混洗协议在对数据一无所知的情况下，对子采样后的梯度完成安全的拆分混洗操作，通过隐私放大效应使得算法满足  $\epsilon_0$ -差分隐私，达到梯度匿名机制，最后将混洗后的结果发送至中央服务器。
- 中央服务器：一个诚实但好奇的第三方。服务器接受混洗器上传的梯度并进行聚合，然后更新全局模型。

假设现在有  $m$  个本地客户端，每个客户端表示为  $i \in [m]$ ，有本地数据集  $\mathcal{D}_i = \{d_{i1}, \dots, d_{ir}\} \in \mathbb{S}^r$ ，由  $r$  个数据集合构成。 $F_i(\theta)$  表示在客户端  $i$  的本地数据集  $\mathcal{D}_i$  上进行训练，对于模型梯度  $\theta \in \mathbb{R}^d$  进行衡量的损失函数，其中  $F_i(\theta) = \frac{1}{r} \sum_{j=1}^r f(\theta; d_{ij})$ ， $f(\theta; \cdot) : \mathcal{C} \rightarrow \mathbb{R}$  是凸函数。中央服务器的目标是找到一个最佳的模型参数向量  $\theta^* \in \mathcal{C}$  使得损失函数  $\min_{\theta \in \mathcal{C}} (F(\theta) = \frac{1}{m} \sum_{i=1}^m F_i(\theta))$  最小，其中隐私性满足单个客户端的隐私预算，也就是满足  $\epsilon_0$ -LDP。在算法3中，首先我们从  $m$  个客户端中随机挑选  $k$  个客户端，表示为集合  $\mathcal{U}_t$ ，其中  $k \leq m$ 。每个客户端  $i \in \mathcal{U}_t$  从本地数据集中抽样  $\mathcal{S}_{it}$  个样本训练模型，计算梯度  $\nabla_{\theta_t} f(\theta_t; d_{ij})$ 。第  $i$  个客户端采用基于第三章的自适应本地差分隐私方案，添加噪声、裁剪梯度，然后将梯度发

送给混洗器。混洗器对收到的梯度进行拆分混洗，然后发送给中央服务器。最后，中央服务器对混洗后的梯度进行聚合求均值，更新全局模型。

---

**Algorithm 3** 联邦学习中的安全模型算法:  $\mathcal{A}_{\text{csdp}}$ 


---

```

1: 输入: 数据集  $\mathcal{D} = \bigcup_{i \in [m]} \mathcal{D}_i$ ,  $\mathcal{D}_i = \{d_{i1}, \dots, d_{ir}\}$ , loss function  $F(\theta) = \frac{1}{mr} \sum_{i=1}^m \sum_{j=1}^r f(\theta; d_{ij})$ , 本地差分隐私预算  $\epsilon_0$ , 梯度范数阈值  $C$ , 模型学习率  $\eta_t$ 
2: 初始化:  $\theta_0 \in \mathcal{C}$ 
3: for  $t \in [T]$  do
4:   客户端采样: 混洗器从  $k$  个客户端中随机采样  $i \in \mathcal{U}_t$  个客户端
5:   for 客户端  $i \in \mathcal{U}_t$  do
6:     梯度选择: 客户端 i 从  $s$  个样本空间中随机采样  $\mathcal{S}_{it}$  个梯度
7:     for 样本  $j \in \mathcal{S}_{it}$  do
8:        $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$ 
9:        $\mathbf{g}_t(d_{ij}) \leftarrow \mathbf{g}_t(d_{ij}) / \max \left\{ 1, \frac{\|\mathbf{g}_t(d_{ij})\|_p}{C} \right\}^3$ 
10:       $\mathbf{q}_t(d_{ij}) \leftarrow \mathcal{R}_p(\tilde{\mathbf{g}}_t(d_{ij}))$ 
11:    end for
12:    客户端 i 将  $\{\mathbf{q}_t(d_{ij})\}_{j \in \mathcal{S}_{it}}$  发送给混洗器
13:  end for
14:  混洗器: 混洗器对于  $\{\mathbf{q}_t(d_{ij}) : i \in \mathcal{U}_t, j \in \mathcal{S}_{it}\}$  中的权重进行拆分混洗, 然后上传给中央服务器
15:  中央服务器聚合梯度:  $\bar{\mathbf{g}}_t \leftarrow \frac{1}{ks} \sum_{i \in \mathcal{U}_t, j \in \mathcal{S}_{it}} \mathbf{q}_t(d_{ij})$ 
16:  梯度下降:  $\theta_{t+1} \leftarrow \prod_{\mathcal{C}} (\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 
17: end for
18: 输出: 最终全局模型参数  $\theta_T$ 

```

---

#### 4.2.1 客户端抽样

假设在空间  $\mathcal{U}$  中我们有一个数据集  $\mathcal{D}' = \{U_1, \dots, U_{r_1}\} \in \mathcal{U}^{r_1}$ , 其中包含  $r_1$  个样本元素。如定义4.2.1所示, 本文定义一个子采样程序: 首先采样一个客户端数据集  $\mathcal{D}' \in \mathcal{U}^{r_1}$ , 再从中采样一个子集作为客户端的本地训练数据。

**定义 4.2.1** (子采样). 定义一个抽样程序  $\text{samp}_{r_1, r_2} : \mathcal{U}^{r_1} \rightarrow \mathcal{U}^{r_2}$ , 其中  $r_2 \leq r_1$ : 从输入的数据集  $\mathcal{D}' \in \mathcal{U}^{r_1}$  中以随机概率抽选一个子数据集  $\mathcal{D}''$ , 数据集  $\mathcal{D}'$  中的每个元素在数据集  $\mathcal{D}''$  中出现的概率为  $q = \frac{r_2}{r_1}$ 。

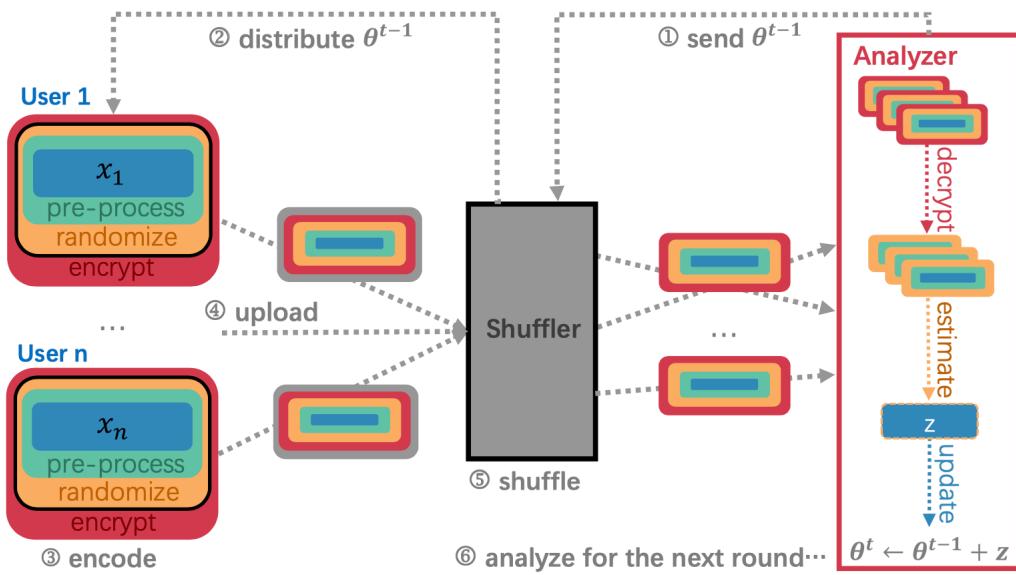


图 4.1: 联邦学习中的安全模型框架

#### 4.2.2 混洗器

先前的研究工作 (H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963, 2017.) 表明，在联邦学习模型中，假如在某个时间段数据是被适当的匿名化，并将数据之间的耦合信息拆分后，模型整体的隐私保障可以得到极大的改善。在第三章中的隐私保护方案是基于本地客户端训练数据的，而面对恶意的中央服务器甚至是恶意的第三方攻击者时，无法保障每个客户端的隐私。

因此在本章中，我们针对客户端上传的梯度，进行参数的拆分混洗，通过混洗器达到客户端的匿名性，打破从中央服务器接收的数据与特定客户端之间的联系，并在每次迭代中从同一客户端发送的梯度更新中将信息解耦。

客户端的匿名性可以通过现有的多种机制来实现，这取决于中央服务器在特定场景下如何跟踪客户端。作为一个典型的保护隐私的最佳做法，每个客户对服务器有一定程度的匿名性，以使客户的个人身份识别与他们的权重更新无法关联。例如，如果服务器通过 IP 地址追踪客户，每个客户可以通过使用网络代理、VPN 服务 [Belesi, 2016]、公共 WiFi 接入 [Dingledine 等人, 2004] 产生一个无法追踪的 IP

地址。再比如，如果服务器通过软件生成的元数据（如 ID）来追踪客户，每个客户可以在向服务器发送元数据之前将其随机化。

但是，我们认为，客户端的匿名性不足以防止通信链道的攻击。例如，如果客户端在每次迭代中同时上传了大量的权重更新，中央服务器仍然可以将它们连接在一起。因此，我们设计了混洗器，以打破来自相同客户的模型权重更新之间的联系，并将其放置于客户端上传梯度更新至中央服务器之间，使中央服务器更难结合多个客户端的同步更新来推断任何客户的更多信息。

如下图所示，我们的混洗器通过以下步骤对客户端上传的梯度参数进行混洗，然后上传给中央服务器：

- 权重分割：每个客户端都对其本地模型的权重进行分割，但给每个分割后的元素贴上一个 id，以表明其在网络结构中的权重位置。
- 权重混洗：对于所有客户端分割后的权重采用随机扰动机制进行混洗。

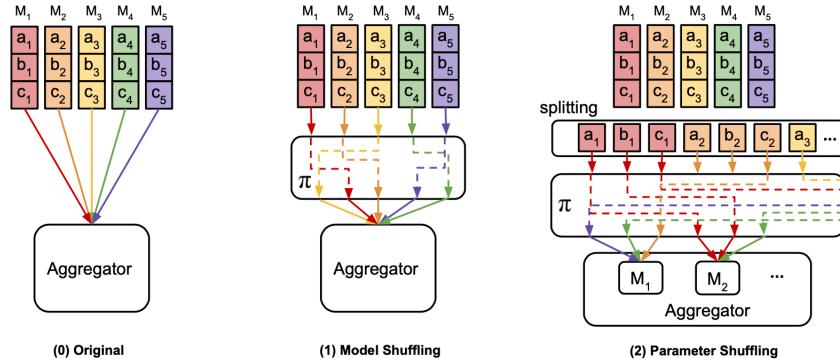


图 4.2: 联邦学习安全模型中执行参数拆分混洗的混洗器

### 4.3 隐私放大效应

隐私放大（privacy amplification）是本章所提出的安全框架中混洗器对隐私效果增强的理论分析，基于该理论，可将现有的本地化差分隐私方法直接应用在安全框架上。

**Algorithm 4** 混洗器中的拆分混洗算法

- 
- 1: **Input:** 本地客户端添加自适应扰动后的权重  $W_{l+1}^s$
  - 2: 对权重  $W_{l+1}^s$  进行分割，给每个元素分配 id
  - 3: **for**  $w^s \in W$  **do**
  - 4:     用一个唯一的 id 标记元素的位置
  - 5:     在通信时刻  $(0, T)$  期间随机采样  $t_{id}^s \leftarrow U(0, T)\%$
  - 6: **end for**
  - 7: 在时刻  $t_{id}^s$  将梯度  $(id, w_{id})$  发送给中央服务器
- 

在算法3中，每个本地客户端采用第三章的满足  $(\epsilon_c + \epsilon_l)$  的自适应本地差分隐私算法，将参数上传至混洗器进行拆分混洗后，所获取的数据满足  $\epsilon_c - DP$ 。从  $(\epsilon_c + \epsilon_l)$  到  $\epsilon_c$  的转变可通过隐私放大理论证明。 $(\epsilon_c + \epsilon_l)$  对应于较大的数值，表示较低的隐私性； $\epsilon_c$  对应于较小的数值，表示较高的隐私性。因此经过混洗器后，隐私性得到了增强。由差分隐私的强组合性可保证算法  $\mathcal{A}_{csdp}$  在每次迭代中对每个样本  $d_{ij}$  都能保证  $\epsilon_0$  的本地差异隐私。因此本节只需要分析采样和混洗操作的隐私放大性。

**定理 4.3.1.** 算法3是满足  $(\epsilon, \delta)$ - 差分隐私的，当对于任意  $\delta, \delta > 0$ ，并且有：

$$\epsilon = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{qT \log(2qT/\delta) \log(2/\delta)}{n}}\right)$$

假设在联邦学习模型中，需要迭代的次数为  $t \in [T]$ 。 $\mathcal{M}_t(\theta_t, \mathcal{D})$  表示在时刻  $t$  对于数据集  $\mathcal{D}$  和模型参数为  $\theta_t$  的差分隐私机制， $\theta_{t+1}$  表示模型的输出。因此，在数据集  $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i \in \mathfrak{S}^n$  上的差分隐私机制定义如下：

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{ks} \circ \text{samp}_{m,k}(\mathcal{G}_1, \dots, \mathcal{G}_m) \quad (4.1)$$

其中， $\mathcal{G}_i = \text{samp}_{r,s}(\mathcal{R}(\mathbf{x}_{i1}^t), \dots, \mathcal{R}(\mathbf{x}_{ir}^t))$  并且  $\mathbf{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [m], j \in [r]$ 。 $\mathcal{H}_{ks}$  表示在  $ks$  个数据样本上进行混洗操作， $\text{samp}_{a,b}$  表示从有  $a$  个元素的集合中随机抽样  $b$  个元素的操作。

接下来我们给出  $\mathcal{M}_t$  的隐私性证明：

假设客户端  $i \in [m]$  的本地数据集为  $\mathcal{D}_i = \{d_{i1}, d_{i2}, \dots, d_{ir}\} \in \mathfrak{S}^r$ ， $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i$

表示总体数据集。根据公式4.1,  $\mathcal{Z}(\mathcal{D}^{(t)}) = \mathcal{H}_{ks}(\mathcal{R}(\mathbf{x}_1^t), \dots, \mathcal{R}(\mathbf{x}_{ks}^t))$  表示在本地客户端进行本地差分隐私后输出的  $ks$  个权重集合上进行混淆后的权重。任取  $\tilde{\delta} > 0$ , 当  $\epsilon_0 \leq \frac{\log(ks/\log(1/\tilde{\delta}))}{2}$  时, 算法  $\mathcal{Z}$  满足  $(\tilde{\epsilon}, \tilde{\delta}) - \text{DP}$  差分隐私, 可得:

$$\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\} e^{\epsilon_0} \sqrt{\frac{\log(1/\tilde{\delta})}{ks}}\right) \quad (4.2)$$

当  $\epsilon_0 = \mathcal{O}(1)$  时, 有  $\tilde{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{\log(1/\tilde{\delta})}{ks}}\right)$ 。

令  $\mathcal{T} \subseteq \{1, \dots, m\}$  表示在时刻  $t$  选取的  $k$  个客户端。对于  $i \in \mathcal{T}$ ,  $\mathcal{T}_i \subseteq \{1, \dots, r\}$  表示在时刻  $t$  客户端  $i$  所抽样的  $s$  条数据样本。对于任意的  $\mathcal{T} \in \binom{[m]}{k}$  和  $\mathcal{T}_i \in \binom{[r]}{s}$ ,  $i \in \mathcal{T}$ , 有  $\bar{\mathcal{T}} = (\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T})$ ,  $\mathcal{D}^{\mathcal{T}_i} = \{d_j : j \in \mathcal{T}_i\}$  for  $i \in \mathcal{T}$ , and  $\mathcal{D}^{\bar{\mathcal{T}}} = \{\mathcal{D}^{\mathcal{T}_i} : i \in \mathcal{T}\}$ 。 $\mathcal{T}$  和  $\mathcal{T}_i, i \in \mathcal{T}$  为抽样产生的任意子集, 其中的随机性由客户端抽样和数据集抽样所决定。算法  $\mathcal{M}_t$  可以等价的表示为  $\mathcal{M}_t = \mathcal{Z}(\mathcal{D}^{\bar{\mathcal{T}}})$ 。

假设现有数据集:  $\mathcal{D}' = (\mathcal{D}'_1) \cup (\cup_{i=2}^m \mathcal{D}_i) \in \mathfrak{S}^n$ , 其中数据集  $\mathcal{D}'_1 = \{d'_{11}, d_{12}, \dots, d_{1r}\}$  和  $\mathcal{D}_1$  为相邻数据集, 它们的第  $d_{11}$  条和第  $d'_{11}$  条数据样本不同。如果  $\mathcal{M}_t$  是满足  $(\bar{\epsilon}, \bar{\delta}) - \text{DP}$  差分隐私的, 那么对于算法  $\mathcal{M}_t$  所选的任意子集  $\mathcal{S}$  都应该满足:

$$\Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] \leq e^{\bar{\epsilon}} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] + \bar{\delta} \quad (4.3)$$

$$\Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] \leq e^{\bar{\epsilon}} \Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] + \bar{\delta} \quad (4.4)$$

由于式4.3和4.4是对称的, 因此只需要证明其中一条。下文给出式4.3的证明:

令  $q = \frac{ks}{mr}$ , 我们给出条件概率的定义:

$$\begin{aligned} A_{11} &= \Pr[\mathcal{Z}(\mathcal{D}^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1] \\ A'_{11} &= \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1] \\ A_{10} &= \Pr[\mathcal{Z}(\mathcal{D}^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1] = \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1] \\ A_0 &= \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \notin \mathcal{T}] = \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{\mathcal{T}}}) \in \mathcal{S} \mid 1 \notin \mathcal{T}] \end{aligned} \quad (4.5)$$

令  $q_1 = \frac{k}{m}$ ,  $q_2 = \frac{s}{r}$ , 那么  $q = q_1 q_2$ , 然后可以得到:

$$\Pr [\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] = q A_{11} + q_1 (1 - q_2) A_{10} + (1 - q_1) A_0 \quad (4.6)$$

$$\Pr [\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] = q A'_{11} + q_1 (1 - q_2) A_{10} + (1 - q_1) A_0 \quad (4.7)$$

因此, 我们可以得到:

$$A_{11} \leq e^{\tilde{\epsilon}} A'_{11} + \tilde{\delta} \quad (4.8)$$

$$A_{11} \leq e^{\tilde{\epsilon}} A_{10} + \tilde{\delta} \quad (4.9)$$

式4.7成立, 因此混洗器  $\mathcal{M}_t$  是满足  $\varepsilon_c$ -差分隐私的。

#### 4.4 模型收敛性分析

回顾第二章的基础知识, 在随机梯度下降算法的每次迭代中, 中央服务器将当前的参数向量发送给所有本地客户端, 客户端收到后在本地数据集上进行模型训练, 计算随机梯度并上传给中央服务器, 然后中央服务器计算收到的梯度的平均值/平均数并更新参数向量。因此在本节中, 我们分析采用采样和混洗算法后模型的收敛性。

在算法3中, 在每一轮迭代过程中, 中央服务器聚合上传的  $ks$  个加躁后的梯度, 如算法3的第 15 行所示, 中央服务器进行聚合后得到结果:  $\bar{\mathbf{g}}_t \leftarrow \frac{1}{ks} \sum_{i \in \mathcal{U}_t, j \in \mathcal{S}_{it}} \mathbf{q}_t(d_{ij})$ , 然后通过随机梯度下降算法更新全局模型参数:  $\theta_{t+1} \leftarrow \prod_{\mathcal{C}} (\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 。其中,  $\mathbf{q}_t(d_{ij}) = \mathcal{R}_p(\nabla_{\theta_t} f(\theta_t; d_{ij}))$ 。

既然随机机制  $\mathcal{R}_p$  是无偏的, 那么平均梯度  $\bar{\mathbf{g}}_t$  也是无偏的, 也就是说, 我们有  $\mathbb{E}[\bar{\mathbf{g}}_t] = \nabla_{\theta_t} F(\theta_t)$ , 其中期望是相对于客户端和数据点的随机抽样以及机制  $\mathcal{R}_p$  的随机性而言的。

令  $F(\theta)$  为凸函数, 考虑这样一个随机梯度下降算法:  $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}(\theta_t - \eta_t \mathbf{g}_t)$ ,  $\mathbf{g}_t$  满足  $\mathbb{E}[\mathbf{g}_t] = \nabla_{\theta_t} F(\theta_t)$  并且  $\mathbb{E}\|\mathbf{g}_t\|_2^2 \leq G^2$ 。当确定  $\eta_t = \frac{D}{G\sqrt{t}}$ , 可以得到:

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq 2DG \frac{2 + \log(T)}{\sqrt{T}} = \mathcal{O}\left(DG \frac{\log(T)}{\sqrt{T}}\right) \quad (4.10)$$

由 Nesterov 等人在文献<sup>[50]</sup> 中的证明可知, 算法3的输出  $\theta_T$  满足:

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD \log(T) \max\left\{d^{\frac{1}{2}-\frac{1}{p}}, 1\right\}}{\sqrt{T}} \left(1 + \sqrt{\frac{cd}{qn}} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)\right)\right) \quad (4.11)$$

其中, 存在  $\sqrt{1 + \frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)^2} \leq \left(1 + \sqrt{\frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)}\right)$ 。

当  $\sqrt{\frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)} \leq \mathcal{O}(1)$  时, 我们恢复了没有隐私性的虚构 SGD 的收敛率。而当  $\sqrt{\frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)} \geq \Omega(1)$  时, 可以推导出:

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD \log(T) \max\left\{d^{\frac{1}{2}-\frac{1}{p}}, 1\right\}}{\sqrt{T}} \sqrt{\frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)}\right) \quad (4.12)$$

如果我们在算法3中设置学习率为  $\eta_t = \frac{D}{G\sqrt{t}}$ , 其中

$G^2 = L^2 \max\left\{d^{1-\frac{2}{p}}, 1\right\} \left(1 + \frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)^2\right)$ 。那么:

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD \log(T) \max\left\{d^{\frac{1}{2}-\frac{1}{p}}, 1\right\}}{\sqrt{T}} \sqrt{\frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)}\right) \quad (4.13)$$

其中, 当  $p \in \{1, \infty\}$  时,  $c = 4$  否则  $c = 14$ 。

**定理 4.4.1** (随机梯度下降算法的收敛性). 假使有凸函数  $F(\theta)$ , 数据集  $D$  的维度为  $\mathcal{C}$ , 在模型训练过程中采用随机梯度下降算法  $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}(\theta_t - \eta_t \mathbf{g}_t)$ , 其中  $\mathbf{g}_t$  满足  $\mathbb{E}[\mathbf{g}_t] = \nabla_{\theta_t} F(\theta_t)$  并且  $\mathbb{E}\|\mathbf{g}_t\|_2^2 \leq G^2$ 。当  $\eta_t = \frac{D}{G\sqrt{t}}$ ,  $\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq 2DG \left(\frac{2+\log(T)}{\sqrt{T}}\right)$  成立。

根据文献中已有的标准随机梯度下降算法收敛结果中使用的4.4.1对  $G^2$  的约束条件, 证明了混洗算法可在  $G^2 = L^2 \max\left\{d^{1-\frac{2}{p}}, 1\right\} \left(1 + \frac{cd}{qn} \left(\frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1}\right)^2\right)$  时达到全局最优解。

## 4.5 本章总结

本章节我们针对联邦学习模型的整体框架进行了隐私性改进，提出了安全混洗模型，在本地客户端和中央服务器之间加设混洗器，通过对本地客户端进行随机抽样，将上传的梯度进行拆分混洗，增加隐私放大效果。然后发送给中央服务器进行聚合。并对方案进行了隐私性证明，表明此安全混洗算法可以保证  $\varepsilon_c$  的差分隐私，然后对此方案在中央服务器上的随机梯度下降算法进行了收敛性的分析，证明在凸函数上，梯度  $\mathbf{g}_t$  满足  $\mathbb{E}[\mathbf{g}_t] = \nabla_{\theta_t} F(\theta_t)$  时模型能达到全局收敛。本章所提出的方案能在保持模型收敛性的情况下，减少隐私预算。

## 第五章 实验与评估

之前的章节中，我们描述了联邦学习的本地自适应差分隐私和安全混淆模型的设计和实现过程。在本节的内容中，我们选取了一些基准的数据集在该验证框架上进行实验评估。本实验是关于联邦学习系统的隐私保护方案。本章的实验主要针联邦深度学习系统训练样本的攻击模型，保护联邦学习系统中参与者的共享梯度信息，避免梯度参数泄露隐私和恶意服务器获取客户端的信息，进而保护参与者本地训练样本。在实验室环境下，通过多 GPU 虚拟化设置模拟分布式联邦学习系统，并且将差分隐私保护方案和混淆器配置在模拟分布式联邦学习系统中，同时在系统中设置攻击模型，评估满足隐私保护算法的系统学习准确率和隐私保护预算。

### 5.1 基准数据集介绍

我们选用了以下三个数据集评估了我们的联邦学习隐私保护框架：

- (1) 手写体数字识别数据集 (MNIST)<sup>[46]</sup> 是用于分类任务的经典数据集，来源于美国国家标准与技术研究所。总共包含了 70000 个手写数字图像，每个图像的尺寸为 28 x 28 像素，每个像素点用灰度值表示，灰度值范围为 0 到 255，图像分为 10 类别，分别代表 0-9。
- (2) FASHION-MNIST 数据集包含了 70000 个不同商品的正面灰度图像，与 MNIST 数据集一样，每个图像的尺寸为 28x28 像素，灰度值范围同样为 0 到 255。所有的图像分为 10 种类别，如：T 恤，牛仔裤，裙子等。虽然数据集格式与 MNIST 相同，但由于图像内容的差别，使得有些模型或者算法在 MNIST 和

FASHION-MNIST 的表现会有很大不同。因此对于分类任务，我们在这两个数据集上都进行了实验作为对比。

(3) CIFAR-10 数据集包含了 10 类（飞机、汽车、鸟类、蛙类、卡车、船、马、猫、鹿、狗）32x32 的彩色图片，一共有 60000 张，每一类包含 6000 张图片。该数据集按照 5:1 的比例划分成了 5 个训练的 batch 和 1 个测试的 batch，每个 batch 均包含 10000 张图片。

## 5.2 实验环境与配置

本文中的所有的实验是在 Windows 10 系统下，使用 CPU Inter(R) Core i3-7100 @ 3.90GHz，GPU 的型号是 NVIDIA GeForce GTX1050，内存 8GB。在实验中使用了 Facebook 公司的 Pythorch 框架对神经网络模型进行编写，相比于 TensorFlow，PyTorch 网络定义方便，更有利于研究小规模项目快速做出原型。其对于并行化数据的支持更有利于分布式联邦系统的实验等）。在对样本数据预处理的部分，我们使用了 Pandas，Numpy 等第三方库。

## 5.3 实验设计

### 5.3.1 联邦学习模型

实验同样设置 30 名联邦学习的参与者，论文研究在分布式联邦系统中添加噪声达到差分隐私并使得整体模型的精度维持较优。首先考虑了如何设置超参数可以更好的让全局模型能够得到更好的训练。分布式联邦学习梯度选择的准则是选择差值变化最大的，调整梯度上传阈值，将上传比例  $\theta_u$  设置为 0.1，将从参数服务器下载的全局参数的比例  $\theta_d$  设置为 1。

接下来，在联邦系统中实施本文所提出隐私保护方案。实验在设置每个参与者在训练分布式联邦系统时每次迭代的总隐私预算为  $\epsilon$ ，将隐私预算分成  $c$  个部分，其中  $c$  是选择每次迭代满足层间前馈传播算法的梯度总数，即  $c = \theta_u |\Delta w|$ 。我们使用拉普拉斯机制根据分配的隐私预算在选择梯度过程中添加噪声。添加的噪声取

决于隐私预验中所有参数的灵敏度  $\Delta f$  都相同, 但具体情况下, 不同的参数可能具有不同的灵敏度。

在分布式联邦学习模型中, 实验评估了不同  $\frac{\theta_u}{\theta_c}$  值的情况下 ( $\theta_u$  为选择梯度阈值的参数), 使用论文方案满足差分隐私的分布式联邦系统的全局模型准确率, 并且将参数保护后系统精度与未保护的模型精度相比较。虽然与集中式深度学习有差距, 由于参与者较多, 而且当参与者共享很大一部分梯度时, 模型的准确性要优于独立训练的准确性。但是, 模型更好的准确性的效果是较低的隐私保护 (即更大的  $\epsilon$  值) 带来的, 更强的隐私保护效果 (更小的  $\epsilon$  值) 会导致较低的模型精度。

### 5.3.2 神经网络模型

Shokri<sup>[51]</sup> 在论文中公开提供了他们的源代码, 实现了一个完整的分布式联邦学习系统。我们将攻击模型部署在该联邦系统中, 并且使用其中的卷积神经网络 (CNN) 架构, 如图5.1。在 CNN 架构中, 网络的前端是卷积层和池化层, 后端则是使用反向传播算法的全连接层。前端的网络结构是在一个 nn. SpatialConvolution 卷积层连接激活函数 TanH, 后面再接一个 nn.SpatialMaxPooling 最大池化层。之后再连接卷积层、TanH 激活函数和池化层单元。后端的网络架构则是 nn.Linear 线性层加上 TanH 激活函数和分类输出层。CNN 网络结构中的参数个数计算如下:

$$32 \times 5 \times 5 + 32 + 64 \times 32 \times 5 \times 5 + 64 + 200 \times 256 + 200 + 10 \times 200 + 10 = 105506$$

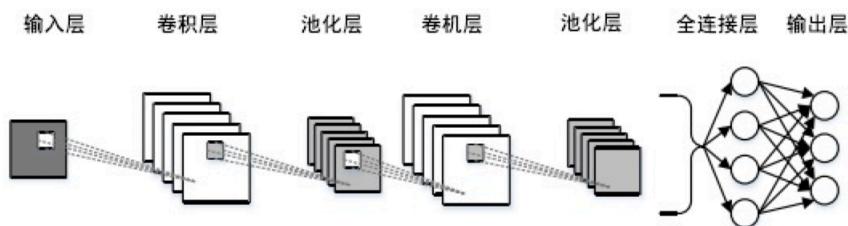


图 5.1: 卷积神经网络结构图

CNN 网络中的损失函数为 nn.CrossEntropyLoss。该函数是 nn.NLLLoss 和 nn.LogSoftmax 的结合, 激活函数使用 Softmax 函数, 损失函数使用交叉熵损失函数评估分类任

务中的损失，也更便于计算反向传播算法。在选择梯度上传的全连接层与传输协议中，部分超参数选择如下：选择参数比例  $\theta_u = 0.01$ ，全局参数  $\theta_d$  下载比例为 1。为了允许在学习中更多的随机性，将学习率设置为  $\alpha = 1 \times 10^{-2}$ ，学习速率衰减值为  $1 \times 10^{-7}$ 。参与者迭代过程使用表 CNN 网络训练本地数据集，攻击者使用基于 CNN 网络的 DCGAN 算法与成员推理攻击的白盒算法。实验在这样的参数设置下搭建一个包含 29 个正常参与者和 1 个攻击者的分布式联邦学习系统，30 个参与者（包含攻击者）都与中央参数服务器进行连接。

我们将与直接增加噪声的情况以及不加噪声的情况进行对比。实验中使用 20000 条数据作为训练数据集，每一个客户端拥有 10 个样本的数据，剩下的样本则作为测试数据集，每种情况分别重复做 5 次并取平均值。实验通信迭代次数为  $T = 200$ ，步长  $\alpha = 1e - 4$ ，衰减系数  $\gamma = 0.99$ 。

## 5.4 自适应扰动方案的实验评估

针对第三章提出的自适应扰动框架，我们从模型预测的准确率和隐私预算参数  $\epsilon$  两个角度评估该方案，隐私预算参数越小，意味着隐私保护的力度越大；模型的准确率越高意味着模型的可用性越高。我们分别使用梯度固定加噪方法和梯度自适应加噪方法进行实验，实验结果如下。

(1) 使用梯度固定加噪方法：使用所有公共训练集数据  $D_{pub}$  进行计算，得到平均梯度 0.001，将其作为固定的梯度裁剪阈值。每轮迭代过程中，在训练批次大小  $L=600$  个样本中添加噪声，因此采样率为  $q = \frac{L}{N} = \frac{600}{60000} = 0.01$ 。在采集的训练样本中添加的噪声量为  $\sigma = 5$ ，隐私参数为  $\delta = 10^{-5}$ 。如图5.2所示，隐私预算参数  $\epsilon$  为研究变量。随着隐私预算参数越大，差分隐私提供的隐私保护强度越小，噪声量越少，模型的准确率越高，符合理论原理。当隐私预算  $\epsilon_c \geq 5$  后，隐私预算参数对于模型准确率影响趋于平稳，综合来看，当  $c \geq 7$  后，部署了差分隐私机制的模型准确率能达到 90% 左右，原始不加噪声的模型相比，准确率下降了 7%。

(2) 使用梯度自适应扰动方法：我们比较了在不同隐私预算下的自适应干扰模

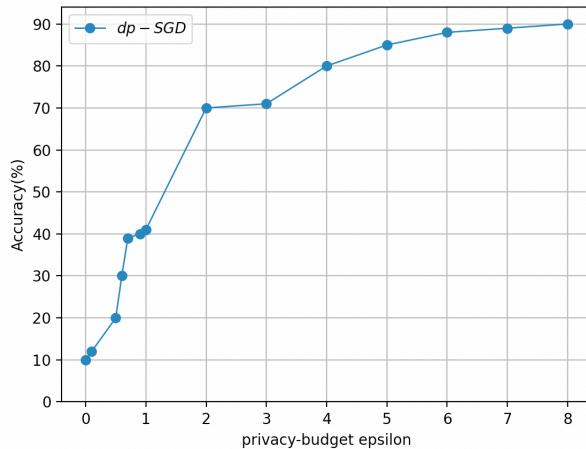


图 5.2: 梯度固定加噪方法下模型准确率随隐私预算变化情况

型的准确性，隐私预算分别为 ( $\epsilon_1 = 0.1, \epsilon_2 = 0.5, \epsilon_3 = 2.0, \epsilon_4 = 8.0$ )。隐私预算  $\epsilon$  越小，噪音就越大。我们还为每个隐私预算选择三种不同的参数取值 ((a):  $f = 0.15, p = 0.85$ , (b):  $f = 0.10, p = 0.90$ (c):  $f = 0.05, p = 0.95$ )。可以肯定的是，设定的 ( $f = 0.15, p = 0.85$ ) 可以保证系统的隐私水平。在实验中，隐私预算  $\epsilon$  的值是  $\epsilon_c$ 、 $\epsilon_l$  和  $\epsilon_c$  的总和。我们将隐私预算的计算分为以下三个步骤：对于贡献的计算、线性转换中的计算和损失函数的计算，即： $\epsilon_c = \epsilon_l = \epsilon_f = \frac{\epsilon}{3}$ 。

正如图5.3所示，随着隐私预算  $\epsilon$  的增加，我们系统的准确性保持稳定的增长趋势。随着调整因子范围的不断缩小，自适应干扰模型的准确率逐渐降低，但仍保持较高的水平。例如，当隐私预算  $\epsilon$  设置为 8.0 时，在  $f=0.15$  和  $p=0.85$  的设置下，APFL 的准确率高达 97.34%，而在  $f=0.10$  和  $p=0.90$  的设置下，准确率为 96.57%，以及在  $f=0.05$  和  $p=0.95$  的设置下，准确率为 96.25%。

综上，自适应隐私预算分配可以根据一般问题的收敛规律，合理地分配隐私参数，从而提高模型表现，但参数  $\gamma$  需要小心选取，过大的  $\gamma$  值会导致训练的初始阶段噪声太大，从而影响模型的可用性。

我们还与近年来使用 DP 机制保护深度学习模型隐私的工作进行了比较，如 [1] 中的 DLPP 和 [18] 中的 DSSGD。在图5.4中，我们可以清楚地得到一个信息，即我们的工作即使在强隐私保证下 ( $\epsilon=0.1$ ) 也表现良好。当调整因素设置为  $f = 0.15$

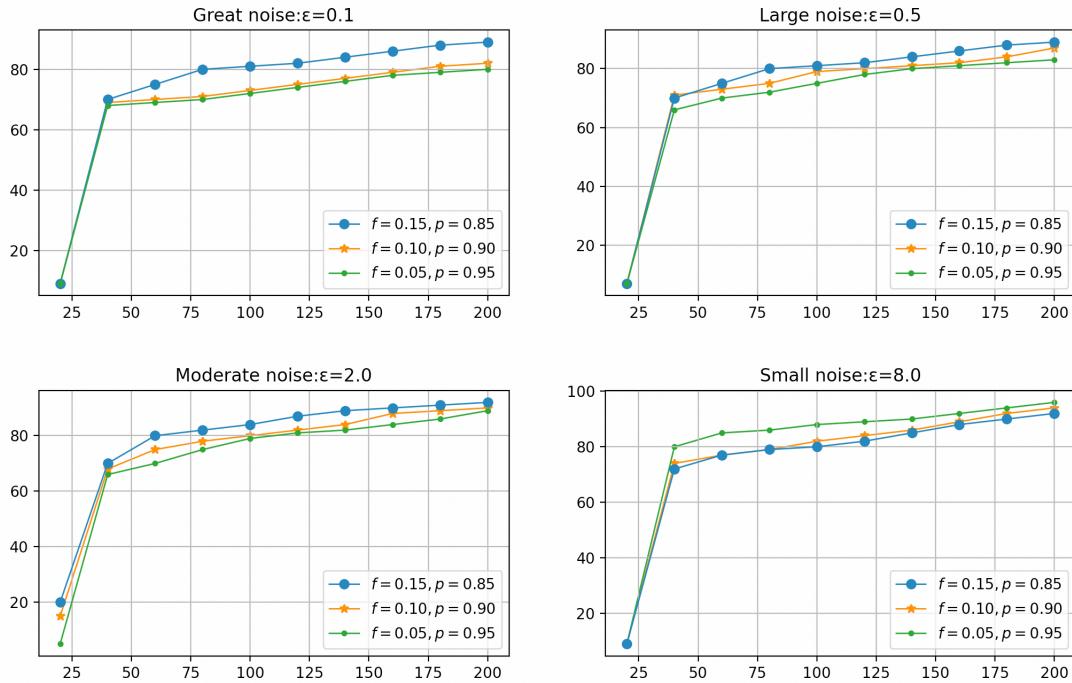


图 5.3: 不同隐私预算的自适应干扰模型的准确率

和  $p = 0.85$  时，模型的准确率在 200 个历时后达到 88.46%。此外，调整因素为  $f=0.05$  和  $p=0.95$ ，自适应干扰模型的准确率为 86.79%。然而，在相同的隐私预算下，差分隐私随机梯度下降算法 (DP-SGD)<sup>[45]</sup> 的准确性仅达到 79.63%，本地差分隐私 DLPP 模型的准确性低于 65.00%。

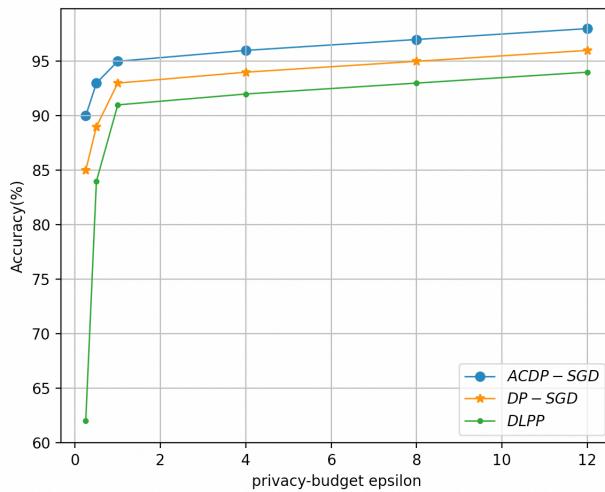


图 5.4: DP-SGD、DLPP、ACDP 在模型准确率和隐私预算上的对比

## 5.5 安全混洗算法的实验评估

我们在 MNIST、FMNIST 和 CIFAR 上评估所提出的安全聚合框架。为了评估参数：客户端数量  $n$  对于隐私预算和模型预测准确率的影响。如图5.5所示，通过客户端采样机制和梯度的拆分混洗算法，我们的安全混洗模型（下文简称 SA-FL）能够以较低的隐私成本实现较高的准确性。在训练中增加客户数量  $n$  的同时，SA-FL 的表现与无噪声的联合学习一样接近。与 MNIST( $n=100, \epsilon=1$ )、FMNIST( $n=200, \epsilon=5$ ) 相比，CIFAR-10( $n=500, \epsilon=10$ ) 需要更多的客户端，这表明对于一个具有较大神经网络模型的更复杂的任务，当在更多的本地数据和更多的客户端上添加扰动之后，需要更多的通信回合才能使联合模型达到更高的精度。

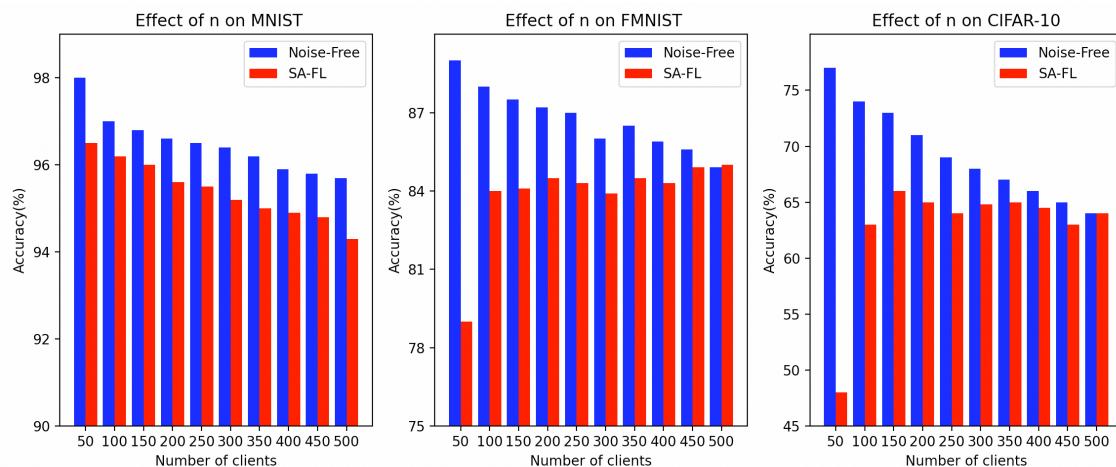


图 5.5: 安全混洗模型中参与混洗的本地客户端数量对联合模型精度的影响

接着，我们分别在 MNIST, FMNIST 和 CIFAR-10 评估了客户端采样比和通信回合对于模型训练准确率的影响。由图5.6可以发现，当  $f_r$  太小的时候，并不影响在 MNIST 上的表现，但对 FASHION-MNIST 和 CIFAR-10 的表现影响很大。当  $f_r$  接近 1 时，安全聚合框架可以在 MNIST、FASHION-MNIST 和 CIFAR-10 上取得与无噪声结果几乎相同的性能。另一个重要的参数是中央参数聚合器和本地客户端之间的通信轮次  $m$ 。不难看出，随着通信次数的增加，我们可以通过所提出的模型在所有数据集上训练出更好的模型。然而，由于数据和任务的复杂性，CIFAR-10 需要更多的通信回合以获得更好的模型。

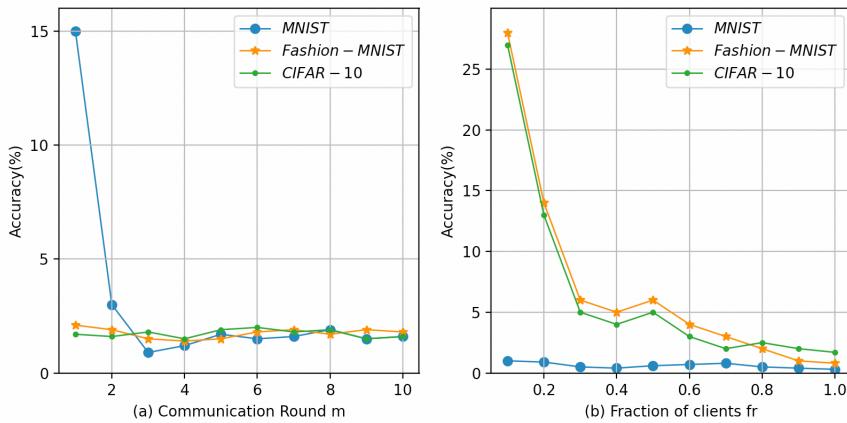


图 5.6: 安全混洗模型中通信轮数和客户端采样比对联合模型精度的影响

最后，我们将统一比较应用了自适应差分隐私算法和安全混洗器的联邦学习模型与其他联邦学习隐私保护模型，在相同隐私预算参数下训练模型能达到的精度。如图5.7(a-c) 中，SA-FL 在  $\epsilon=4$  和  $n=100$  的情况下可以达到 96.24% 的准确率，在  $\epsilon=4$ ,  $n=200$  的情况下可以达到 86.26% 的准确率，在  $\epsilon=10$ ,  $n=500$  的情况下，在 MNIST, FMNIST 和 CIFAR-10 上可以达到 61.4% 的准确率。我们的结果与之前的其他工作相比非常有竞争力。[Geyer 等人,2017] 首次将差分隐私应用于联邦学习，虽然他们只使用了 100 个客户端，但在 MNIST 上，他们只能在  $(\epsilon, m) = (8, 11), (8, 54)$  和  $(8, 412)$  的情况下达到 78%, 92% 和 96% 的准确率，其中  $(\epsilon, m)$  代表隐私预算和通信回合。[Bhowmick 等人，2018] 首次在联合学习中利用本地差分隐私。由于其机制的高变异性，它需要超过 200 轮的通信，并花费更多的隐私预算，即 MNIST ( $\epsilon=500$ ) 和 CIFAR-10 ( $\epsilon=5000$ )。最近的工作 [Truex 等人，2020] 将压缩后的局部差分隐私 ( $\alpha$ -CLDP) 应用到联邦学习中，在 FMNIST 数据集上获得了 86.93% 的准确性。然而， $\alpha$ -CLDP 需要相对较大的隐私预算  $\epsilon = \alpha \cdot 2c \cdot 10\rho$ （例如， $\alpha = 1, c = 1, \rho = 10$ ）来实现该性能，这导致了较弱的隐私保证。与以往的工作相比，我们的方法在客户端和云端之间需要更少的通信回合（例如，MNIST 为 10, FMNIST 和 CIFAR-10 为 15），这使得整个解决方案在实际场景中更加实用。总的来说，SA-FL 在隐私成本、模型精度和通信成本方面都比之前的作品取得了更好的表现。

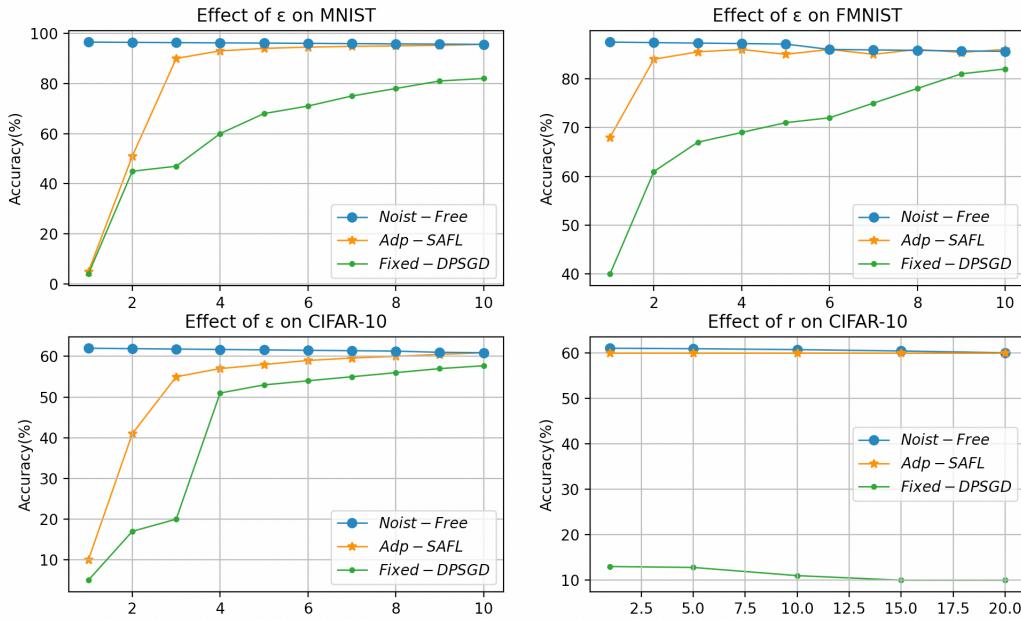


图 5.7: 自适应差分混淆模型和其他联邦学习隐私保护模型的比较

## 5.6 结果分析

为了验证自适应扰动算法对于模型训练的精度也能维持在较优的水平，我们进行了对比实验，使用自适应扰动算法和不使用这种方法在不同的隐私参数  $\epsilon$  下的对比实验。由本章第三节对于自适应差分隐私方案的实验评估，我们可以看到自适应扰动算法基本上占有绝对的优势，尤其是在损失函数值，在隐私参数  $\epsilon=0.1$  时，我们的方法不到 1，而传统的平均算法却在 100 左右。这么大的差距的原因在于，自适应扰动算法的权重分配使得聚合时个体信噪比不变，但整体的聚合结果的信噪比却提高了很多，因此当隐私参数  $\epsilon$  很小，即噪声量很大的时候，表现越好。而当  $\epsilon$  越大时，注入的噪声也就越小，自适应加躁方法的效果就没有噪声大的时候明显。

系统的额外开销主要来自服务器端的预训练过程，以及用户端在开始训练前对贡献的计算和扰动。我们使用 20 个历时来训练云服务器的初始化网络，这平均需要 68.22 秒。在独立和异步的训练过程之前，用户需要用层间依赖传播算法计算权重。这个过程只需要训练正向传播过程，而不需要计算反向传播过程中的梯度

和损失惩罚。其平均时间为 4.35 毫秒。为了减轻隐私威胁，我们的解决方案是向权重、线性变换函数中的原始数据和损失函数的系数注入拉普拉斯噪声。向权重注入噪声的步骤可以与计算贡献同步进行，这需要额外的 2.67 毫秒时间。向线性变换中的原始数据和损失函数的系数注入自适应噪声的操作可以在训练前完成，每一个历时的计算都与扰动的权重相似。因此，在模型效率方面的提升是非常突出的。

从隐私成本和模型精度的总体上看，混淆差分隐私方法在各统计问题的结果可用性上都有着相比本地化差分隐私方法明显更优的结果。但从通信代价和计算代价的角度分析，安全混淆算法中混淆器的引入，一方面使得用户数据与用户所使用的编码器之间的关联性消失，使得分析器端的计算代价增大；另一方面造成了分析器端的通信代价增大。如何兼顾数据的隐私性、可用性、算法的计算代价和通信代价是后续基于 SA-FL 框架构建的隐私保护方法需加以研究的部分。

## 5.7 本章小结

在本章中，我们选取了三个基准数据集对本文提出的自适应本地差分隐私和安全混淆框架进行了一系列的实验来测试其可行性，并且在联邦学习系统上也进行实验和研究。实验结果表明，我们的自适应本地差分隐私可以有效降低隐私预算，并且维持模型精度。安全混淆框架能通过客户端采样算法和梯度的拆分混淆算法，降低隐私保护预算，提高数据的可用性。

## 第六章 总结与展望

### 6.1 论文总结

随着深度学习的兴起，出现了越来越多新的模型和算法，能够更有效的解决各类问题。基于人工智能的产品也在各个领域迎来了一波新的发展热潮，给人民的生活带来了巨大的便利。然而用户在享受深度学习模型带来便利的同时，必须共享自己的数据，随着隐私泄露事件越来越多，数据的安全和隐私问题也逐步引起了人们的关注。

与此同时，各类智能设备也在不断发展，用户产生的数据也越来越多，智能设备的算力不断增强。用户不愿意向商业公司或商业机构提供个人隐私数据。分布式联邦学习系统解决分布式终端用户在本地更新模型的问题，联邦学习的目标是保障大数据共享信息时的数据安全、保护本地数据和个人隐私，在多计算节点之间高效的训练机器学习模型。

分布式联邦学习系统得到了广泛的研究和应用，成为传统集中式机器学习方法的一种改进方法。它不是将数据上传到中心服务器进行集中训练，而是参与者在本地进行模型训练并与参数服务器共享模型更新。参数服务器对来自多个参与者的权重进行聚合，并组合创建一个改进的全局模型，这有助于保障用户的数据隐私和降低通信成本。虽然联邦学习解决了传统集中式深度学习所面临的大规模数据收集等问题，节省了传输数据所占用的通信资源。但是，联邦学习中的共享参数以及传输数据的无线链路仍然可能泄露数据隐私。各类攻击模型阻碍了联邦学习技术的发展，也会极大地威胁到人们的隐私敏感信息。

本文主要研究针对分布式联邦学习系统的隐私安全问题。通过研究神经网络

的前向传播算法和差分隐私的相关性质，提出了一套分布式联邦系统中针对梯度和通信信道攻击的隐私安全方案对策。本文的主要工作和贡献如下：

- (1) 基于本地差分隐私的自适应干扰算法：在客户端本地训练的神经网络模型中，通过分析前向传播算法，计算每个属性类对于模型输出的贡献比，然后，我们开发了一个自适应噪声添加的方案，根据贡献率注入不同隐私预算的噪声。与传统的注入噪声的方法相比，我们在相同的隐私保护程度下最大限度地提高了模型的准确性，减少噪声对模型输出结果的影响，提高模型精度。而且，本文也从本地差分隐私定义的角度，理论证明了提出的方法满足  $\mathcal{E}$ -本地差分隐私。最后通过多组真实数据集以及合成数据集验证了本地自适应扰动机制的性能，证明了其在相同条件下要优于现有的同类方法。
- (2) 本文提出了 SA 安全混洗框架，混洗器对客户端上传的梯度进行采样后，然后拆分混洗，再将混洗模型和自适应本地差分隐私保护方法结合在分布式系统中，提高系统学习效果，在差分隐私的保证下实现了数据隐私度与模型可用性之间的更好平衡。

综上所述，本文的研究充分证明了所提出框架的有效性，可以极大的联邦学习模型的隐私性和可用性，从而进一步推进了联邦学习在安全领域的应用和发展。

## 6.2 论文展望

在可预见的未来，大规模、大数据、分布式的深度学习将得到快速发展。物联网、5G、边缘计算等技术也将迅速普及。人类将彻底步入人工智能时代。在此我将对我未来的研究做出几点展望：

- (1) 本文提出的基于本地自适应混洗差分隐私深度学习算法是一种基础算法，在模型学习的过程中，它的总体隐私预算并不会随着通信回合的增加而大幅上升。因此后续可以研究其在大型数据集与复杂模型结构中的表现。

- (2) 差分隐私对于数据的保护是基于数学证明的，但是缺乏一定的可解释性，如果能够在差分隐私保护深度学习模型上建立更加有效的隐私风险评估和隐私成本评估指标，那么将来应该能更好的应用差分隐私，推动深度学习机制下的差分隐私保护的研究发展。
- (3) 现实生活中，联邦学习的参与方数量可能有百万、千万的级别。当客户端的量级大大增加时，由于本地设备在通信、计算和存储等各个方面的能力大有不同，因此之后关于实际应用中的通信成本、设备异构等方面也需要大量的研究。

## 参考文献

- [1] Pouyanfar S, Sadiq S, Yan Y, et al. A survey on deep learning: Algorithms, techniques, and applications[J]. ACM Computing Surveys (CSUR), 2018, 51(5): 1-36.
- [2] Voigt P, Von dem Bussche A. The eu general data protection regulation (gdpr)[J]. A Practical Guide, 1st Ed., Cham: Springer International Publishing, 2017, 10: 3152676.
- [3] Kendall A, Gal Y, Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7482-7491.
- [4] Hu R, Dollár P, He K, et al. Learning to segment every thing[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4233-4241.
- [5] 张仕良. 基于深度神经网络的语音识别模型研究 [D]. 合肥: 中国科学技术大学, 2017.
- [6] Sardianos C, Tsirakis N, Varlamis I. A survey on the scalability of recommender systems for social networks[M]//Social Networks Science: Design, Implementation, Security, and Challenges. Springer, Cham, 2018: 89-110.
- [7] Shen D, Wu G, Suk H I. Deep learning in medical image analysis[J]. Annual review of biomedical engineering, 2017, 19: 221-248.

- [8] Papernot N, Abadi M, Erlingsson U, et al. Semi-supervised knowledge transfer for deep learning from private training data[J]. arXiv preprint arXiv:1610.05755, 2016.
- [9] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis[C]//Theory of cryptography conference. Springer, Berlin, Heidelberg, 2006: 265-284.
- [10] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms[J]. Foundations of secure computation, 1978, 4(11): 169-180.
- [11] Wu X, Fredrikson M, Jha S, et al. A methodology for formalizing model-inversion attacks[C]//2016 IEEE 29th Computer Security Foundations Symposium (CSF). IEEE, 2016: 355-370.
- [12] Hitaj B, Ateniese G, Perez-Cruz F. Deep models under the GAN: information leakage from collaborative deep learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 603-618.
- [13] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]//2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 3-18.
- [14] Dwork C. Differential privacy[C]//International Colloquium on Automata, Languages, and Programming. Springer, Berlin, Heidelberg, 2006: 1-12.
- [15] Alfeld S, Zhu X, Barford P. Data poisoning attacks against autoregressive models[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016, 30(1).
- [16] Yao A C. Protocols for secure computations[C]//23rd annual symposium on foundations of computer science (sfcs 1982). IEEE, 1982: 160-164.

- [17] Meng X, Bradley J, Yavuz B, et al. Mllib: Machine learning in apache spark[J]. *The Journal of Machine Learning Research*, 2016, 17(1): 1235-1241.
- [18] Wang X, Han Y, Wang C, et al. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning[J]. *IEEE Network*, 2019, 33(5): 156-165.
- [19] Li T, Sahu A K, Talwalkar A, et al. Federated learning: Challenges, methods, and future directions[J]. *IEEE Signal Processing Magazine*, 2020, 37(3): 50-60.
- [20] Tran N H, Bao W, Zomaya A, et al. Federated learning over wireless networks: Optimization model design and analysis[C]//*IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019: 1387-1395.
- [21] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//*Artificial intelligence and statistics*. PMLR, 2017: 1273-1282.
- [22] Zhu L, Han S. Deep leakage from gradients[M]//*Federated learning*. Springer, Cham, 2020: 17-31.
- [23] Aono Y, Hayashi T, Wang L, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 13(5): 1333-1345.
- [24] Ma C, Li J, Ding M, et al. On safeguarding privacy and security in the framework of federated learning[J]. *IEEE network*, 2020, 34(4): 242-248.
- [25] 曹志义, 牛少彰, 张继威. 基于半监督学习生成对抗网络的人脸还原算法研究[J]. *电子与信息学报*, 2018, 40(2): 323-330. Distributed differential privacy via shuffling. In *Eurocrypt*. Springer, 2019.

- [26] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
- [27] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.
- [28] Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training gans[J]. Advances in neural information processing systems, 2016, 29: 2234-2242.
- [29] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016.
- [30] Johnson R, Zhang T. Accelerating stochastic gradient descent using predictive variance reduction[J]. Advances in neural information processing systems, 2013, 26: 315-323.
- [31] Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms[C]//Proceedings of the twenty-first international conference on Machine learning. 2004: 116.
- [32] Dwork C, Kenthapadi K, McSherry F, et al. Our data, ourselves: Privacy via distributed noise generation[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2006: 486-503.
- [33] McSherry F, Talwar K. Mechanism design via differential privacy[C]//48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). IEEE, 2007: 94-103.
- [34] LBengio Y. Learning deep architectures for AI[M]. Now Publishers Inc, 2009.

- [35] Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Found. Trends Theor. Comput. Sci., 2014, 9(3-4): 211-407.
- [36] Bassily R, Smith A, Thakurta A. Private empirical risk minimization: Efficient algorithms and tight error bounds[C]//2014 IEEE 55th Annual Symposium on Foundations of Computer Science. IEEE, 2014: 464-473.
- [37] Acs G, Melis L, Castelluccia C, et al. Differentially private mixture of generative neural networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(6): 1109-1121.
- [38] Su D, Cao J, Li N, et al. Differentially private k-means clustering and a hybrid approach to private optimization[J]. ACM Transactions on Privacy and Security (TOPS), 2017, 20(4): 1-33.
- [39] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. 2007: 791-798.
- [40] Bassily R, Smith A, Thakurta A. Private empirical risk minimization: Efficient algorithms and tight error bounds[C]//2014 IEEE 55th Annual Symposium on Foundations of Computer Science. IEEE, 2014: 464-473.
- [41] McSherry F D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis[C]//Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. 2009: 19-30.
- [42] Thakurta A G. Differentially private convex optimization for empirical risk minimization and high-dimensional regression[M]. The Pennsylvania State University, 2013.

- [43] Lee J, Kifer D. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining. 2018: 1656-1665.
- [44] Balle B, Wang Y X. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising[C]//International Conference on Machine Learning. PMLR, 2018: 394-403.
- [45] Shokri R, Shmatikov V. Privacy-preserving deep learning[C]//Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015: 1310-1321.
- [46] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [47] Song S, Chaudhuri K, Sarwate A D. Stochastic gradient descent with differentially private updates[C]//2013 IEEE Global Conference on Signal and Information Processing. IEEE, 2013: 245-248.
- [48] Geyer R C, Klein T, Nabi M. Differentially private federated learning: A client level perspective[J]. arXiv preprint arXiv:1712.07557, 2017.
- [49] Truex S, Baracaldo N, Anwar A, et al. A hybrid approach to privacy-preserving federated learning[C]//Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. 2019: 1-11.
- [50] Nesterov Y. Introductory lectures on convex optimization: A basic course[M]. Springer Science Business Media, 2003.
- [51] M, Lantz E, Jha S, et al. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing[C]//23rd USENIX Security Symposium (USENIX Security 14). 2014: 17-32.

## 致 谢

时光荏苒，岁月如梭，研究生的日子过得飞快，转眼间我的硕士研究生学习生涯即将接近尾声。在华东师范大学读研的这两年时光，我不但学习到了很多知识，也结识了许多良师益友，此时此刻，我的内心充满了无限的感慨。所谓饮水思源，在此我要向每位陪伴我，鼓励我，教导我的人表示由衷的感谢。

从 2019 年收到华东师范大学的研究生录取通知书，我满怀憧憬和抱负的来到华师大，来到上海可信计算实验室，有幸成为曹珍富老师的学生。感谢实验室的各位老师们，他们不但为我们提供了优质教学环境和资源，还创造了良好的学习氛围，通过一流的科研实力和丰富的科研热情带领我们学习最前沿的科研成果。为了充实我们的研究生生活，学院定期举办各种学术会议和活动，邀请到国内外知名学者给我们做讲座，让我们有机会接触到最新的科研成果。而且，无论是在科研还是生活上遇到问题，老师们都会耐心的给我们提建议，鼓励帮助我们一起克服这些困难。

研究生的时光是轻快而稍纵即逝的，和实验室同学、室友的朝夕相处是我最难忘的回忆。因为有室友高圆圆、陈少敏、冯世玲，宿舍的氛围一直是欢快的，我们早晨共同早起去图书馆自习，下课了去实验室读论文，空闲时间一起在操场打篮球，欢声笑语，常伴我们。三年时光里，我们彻夜未睡，通宵准备数模竞赛；早出晚归，一起在理科楼度过日日夜夜，都将成为我的学生时代美好的回忆。

同门情谊似手足之情，感谢实验室的各位同窗好友，吴楠、汤琦、陆鹏皓、李翔宇、任城东、李明冲等，是有你们的互励互助，我才得以开心努力而充实的度过了这段美好的研究生生活，希望以后仍然

有机会共同努力、共同奋斗。

最后，感谢家人朋友一直以来的坚定支持与热切关怀。学生时代终将落幕，我们经历了与新冠病毒的抗疫战斗，深刻体会到走向社会意味着更多的责任与承担。在未来的工作中，应当当兢兢业业，不负学校美誉；在未来的生活中，应当诚心敬意，报父母养育之恩。

在这篇论文完成之际意味着三年的硕士生涯即将告一段落，而自己也将踏上人生的下一段旅程。回顾硕士三年的时光，非常有幸能成为华东师范大学的学子。非常庆幸能成为曹珍富老师的学生，非常庆幸能和实验室的大家成为朋友，这是人生中可遇而不可求的经历。最后诚挚感谢各位评审专家、老师百忙之中对我的论文所给予的指导和建议。

何慧娴

二零二壹年九月

## 攻读硕士学位期间发表论文、参与科研和获得荣誉情况

### ■ 已完成学术论文

- [1] **Huixian He**, Zhenfu Cao. Adaptive Privacy-preserving and Shuffling Aggregation in Federated-learning[C]. 2021 The 11th International Workshop on Computer Science and Engineering, Shanghai, China.[第一作者]