

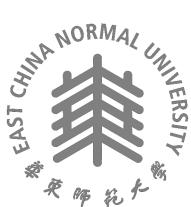
2022 届硕士专业学位研究生学位论文

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: \_\_\_\_\_

论文作者: \_\_\_\_\_

2021 年 09 月 10 日

Thesis (Professional) for Master's Degree in 2022

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:	Cryptography and Network Security
Supervisor:	
Candidate:	

Nov 9, 2021



## 摘要

随着机器学习成为一种实践和商品，提供了大量基于云的服务和框架来帮助客户开发和部署机器学习应用程序。虽然在云中外包模型训练和服务任务很普遍，但保护训练数据集中敏感样本的隐私并防止信息泄露给不受信任的第三方也很重要。过去的工作表明，恶意机器学习服务提供商或最终用户可以轻松地从模型参数甚至模型输出中提取有关训练样本的关键信息。

在大数据时代，数据隐私已成为最重要的问题之一。迄今为止，存在大量安全策略和加密算法，试图确保敏感数据不会受到损害。此外，其中大部分安全策略都假设只有拥有密钥的人才能访问机密数据。然而，随着机器学习的广泛使用，特别是集中式机器学习，为了训练有用的模型，数据应该被收集并转移到一个中心点。因此，对于那些隐私敏感的数据，难免会面临数据泄露的风险。因此，如何在不泄漏数据的情况下对私有数据集进行机器学习是共享智能的关键问题。基于隐私保护，具有多方隐私保护的机器学习可以帮助各方用户在保证自身数据安全的前提下，共同学习彼此的数据。其中，联邦学习是一种典型的有助于解决多方计算下的隐私问题的学习方法。

随着数据孤岛的出现和隐私意识的普及，联邦学习作为一种新兴的数据共享和交换模型，可以在保护数据隐私和安全的前提下实现多方协作，因为分布在多个设备上的数据无法发送当地。为实现各方利益，在金融、医疗、教育等诸多领域得到广泛应用。但是，联邦学习也存在各种安全和隐私问题。本文从联邦学习的概述出发，详细描述了威胁模型和存在的安全问题，包括模型重建攻击、中毒攻击、推理攻击等，然后对联邦学习隐私保护安全技术进行了一定的分析。与安全多方

计算和同态加密相比，差分隐私在效率方面非常出色。

在本文中，我们开发了一种基于自适应差分隐私混洗的联邦学习算法。该模型是在多方隐私保护下通过梯度学习联合训练的。具体来说，该模型在每次迭代中通过梯度下降进行优化，并且可以通过传递梯度来从其他用户的数据中学习。

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。之后我们设计了动量组合机制分析加噪累积产生的隐私预算，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大大减少了噪声对模型输出结果的影响，提高了模型的准确性。
- (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，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, Differential privacy , Security shuffle*



# 目录

第一章 绪 论 . . . . .	1
1.1 研究背景及意义 . . . . .	1
1.2 安全性和隐私威胁 . . . . .	4
1.3 国内外研究现状 . . . . .	6
1.4 本文工作与主要贡献 . . . . .	7
1.5 本文组织结构 . . . . .	8
1.6 本章小结 . . . . .	9
第二章 基础知识 . . . . .	10
2.1 神经网络 . . . . .	10
2.1.1 基本结构 . . . . .	10
2.1.2 随机梯度下降算法 . . . . .	11
2.1.3 经验风险最小化 . . . . .	12
2.2 联邦学习 . . . . .	13
2.3 差分隐私 . . . . .	13
2.3.1 基本定义 . . . . .	13
2.3.2 实现机制 . . . . .	15
2.3.3 相关定理 . . . . .	16
2.3.4 联邦学习中的差分隐私 . . . . .	17
2.4 本章小结 . . . . .	18
第三章 联邦学习中的本地自适应差分隐私机制 . . . . .	19
3.1 引言 . . . . .	19

3.2	相关理论 . . . . .	22
3.2.1	自适应噪声添加算法 . . . . .	22
3.2.2	拉普拉斯平滑机制 . . . . .	25
3.2.3	自适应梯度裁剪 . . . . .	26
3.3	自适应差分隐私算法 . . . . .	26
3.4	隐私性证明 . . . . .	28
3.5	模型效用分析 . . . . .	30
3.6	隐私预算分析 . . . . .	30
3.7	本章总结 . . . . .	32
<b>第四章</b>	<b>联邦学习的安全混洗模型 . . . . .</b>	<b>33</b>
4.1	引言 . . . . .	33
4.2	安全混洗模型 . . . . .	34
4.2.1	客户端抽样 . . . . .	36
4.2.2	混洗器 . . . . .	36
4.3	隐私放大效应 . . . . .	38
4.4	模型收敛性分析 . . . . .	41
4.5	实验评估 . . . . .	42
4.5.1	实验准备 . . . . .	42
4.5.2	实验设计 . . . . .	42
4.5.3	实验分析 . . . . .	42
4.6	本章总结 . . . . .	42
<b>第五章</b>	<b>实验与评估 . . . . .</b>	<b>44</b>
5.1	基准数据集介绍 . . . . .	44
5.2	实验环境与配置 . . . . .	45
5.3	实验设计 . . . . .	45
5.3.1	联邦学习模型 . . . . .	45
5.3.2	神经网络模型 . . . . .	46
5.4	自适应扰动方案的实验评估 . . . . .	47

5.5	安全混洗算法的实验评估 . . . . .	49
5.6	结果分析 . . . . .	51
5.7	本章小结 . . . . .	53
<b>第六章</b>	<b>总结与展望 . . . . .</b>	<b>54</b>
6.1	论文总结 . . . . .	54
6.2	论文展望 . . . . .	55
	<b>参考文献 . . . . .</b>	<b>57</b>
	<b>致谢 . . . . .</b>	<b>65</b>
	<b>发表论文和科研情况 . . . . .</b>	<b>67</b>

# 插图

1.1	联邦学习模型概况 . . . . .	3
2.1	神经网络结构图 . . . . .	11
2.2	联邦学习模型工作流程 . . . . .	14
2.3	差分隐私的相邻数据集示意图 . . . . .	14
3.1	神经网络的前向传播和反向传播流程图 . . . . .	22
4.1	联邦学习安全模型框架 . . . . .	34
4.2	联邦学习安全模型与原联邦学习模型的信任域对比 . . . . .	35
4.3	联邦学习安全混洗模型中执行参数拆分混洗的混洗器 . . . . .	38
5.1	卷积神经网络结构图 . . . . .	46
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 . . . . .	48
5.3	不同隐私预算的自适应干扰机制在 MINIST 数据集上的准确率 . . . . .	48
5.4	DP-SGD、DLPP、ACDP-SGD 在模型准确率和隐私预算上的对比 . . . . .	49
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 . . . . .	50
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 . . . . .	51
5.7	自适应差分混洗模型和其他联邦学习隐私保护模型的比较 . . . . .	52

# List of Algorithms

1	随机梯度下降算法 . . . . .	12
2	自适应高斯噪声算法 . . . . .	25
3	差分隐私随机动量优化算法 . . . . .	27
4	联邦学习中的安全模型算法: $\mathcal{A}_{\text{csdp}}$ . . . . .	36
5	混淆器中的拆分混淆算法 . . . . .	37

# 第一章 緒論

## 1.1 研究背景及意义

在过去的近十年，人工智能（Artificial Intelligence, AI）取得了令人难以置信的进步，机器学习也越来越广泛地应用于各种领域，包括医疗健康、自动驾驶、金融贸易等。为了进一步提高模型的训练精度和学习能力，新兴的深度神经网络，也称为深度学习 (Deep Learning, DL) 随之提出。深度学习算法的目标是通过从数据中泛化来学习如何执行某些任务，比如说图像分类、语音识别、自然语言翻译等。深度学习作为最有前景的技术之一，已广泛应用于图像分类、自动驾驶、智慧医疗等各个方面。例如，智能图像识别系统已广泛部署在机场、火车站等公共场所。在识别可疑恐怖分子和检测违禁物品方面，它已被证明比人类更精确。基于患者的医疗数据，基于深度学习的回归技术可以帮助诊断和预防某些疾病（例如，遗传性和传染病）。很明显，基于深度学习的服务正在从旅行、社交、经济等诸多方面慢慢改变我们的生活。

深度学习算法的输入数据通常表示为一组样本。每个样本将包含一组特征值。例如，考虑一张 100x100 像素的照片，其中每个像素由一个数字（0-255 灰度）表示。我们可以用这些像素值组成一个长度为 10,000 的向量，通常称为特征向量。每张照片，表示为一个特征向量，可以与一个标签（例如，照片中人物的名字）相关联。深度学习算法将使用由多个特征向量及其相关标签组成的训练集来构建深度学习模型，这个过程称为模型的训练。当呈现一个新的测试样本时，深度学习模型应该给出预测的标签。模型准确预测标签的能力是衡量该模型对未知的数据的泛化程度的标准。它是通过测试误差（泛化误差）凭经验衡量的，它可以取决于用于

训练模型的数据的质量和数量、使用什么深度学习算法来构建模型、深度学习算法超参数的选择（例如使用交叉验证），甚至是特征的提取方法。

一般来说，训练数据量在某种程度上决定了模型的性能。为了支持基于机器学习/深度学习的开发不断增长的需求，许多互联网云提供商推动机器学习即服务（DLaaS），为机器学习/深度学习模型训练和服务提供计算平台和学习框架。典型的机器学习即服务平台包括 Amazon Sagemaker、Google Cloud ML Engine、Microsoft Azure ML Studio。要使用 MLaaS，客户需要向云提供商提供训练数据集和机器学习/深度学习算法。云提供商搭建深度学习环境，分配一定的计算资源，自动运行模型训练任务。云提供商还可以提供模型服务，将在云侧或客户侧训练的模型存储在云平台中。最后，云提供商向用户发布查询的 API 接口，使他们能够使用模型进行预测或分类。

训练数据集是训练和生成机器学习模型所必需的。数据集可能包含敏感样本，例如个人医疗记录、员工信息、财务数据等。我们的搜索查询、浏览历史、购买交易、我们观看的视频以及我们的电影的偏好都有可能被收集在使用的移动设备和计算机中。在街道上，以及即使在我们自己的办公室和家里。这种私人数据被用于各种深度学习应用。一些深度学习应用程序需要私人数据，此类私人数据以明文形式上传到集中的服务器，供深度学习算法学习数据中的规律，并从中构建模型。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。问题不仅限于与将所有这些私人数据暴露在这些公司的内部威胁中，或外部威胁，如果持有这些数据集的公司遭到黑客攻击，那将会导致千万甚至亿万级的用户数据泄漏。

如何保护这些样本的隐私已成为机器学习中一个新的安全问题。这种隐私威胁在机器学习即服务中尤为严重。首先，在模型训练服务中，客户需要将训练数据上传到云提供商，提供商拥有数据的完全访问权限。过去的工作表明，恶意提供者

可以轻松窃取敏感数据，将它们嵌入模型中实现隐私的泄漏。其次，在模型训练服务中，客户需要将预训练好的模型上传到云提供商，并设置端点供远程用户使用模型。成功的深度学习模型包含有关训练集的基本信息。因此，即使恶意提供者无法直接访问数据集，他也可以从模型参数中提取有关训练数据的敏感信息。第三，即使云提供商是可信的，只有黑盒访问模型输出结果的恶意远程用户仍然能够通过使用精心设计的输入查询模型来检索有关训练数据的信息。

针对这些问题，Google 在 2016 年提出了联邦学习（Federated Learning, FL）的概念，它是一种典型的有助于解决多方计算下的隐私问题的学习方法。如图1.1所示，联邦学习的基本框架包含多个本地设备和一个中央服务器，所有训练数据保留在本地设备，所有设备共同协作训练一个全局模型。联邦学习实现了数据存储和模型训练的需求分离，使所有本地设备可以在不共享训练数据的情况下参与全局模型的训练。由于这种性质，联邦深度学习受到了工业界和学术界的广泛关注，并且已经提出了各种分布式学习架构 [3]、[4]、[5] 来服务于特定场景。

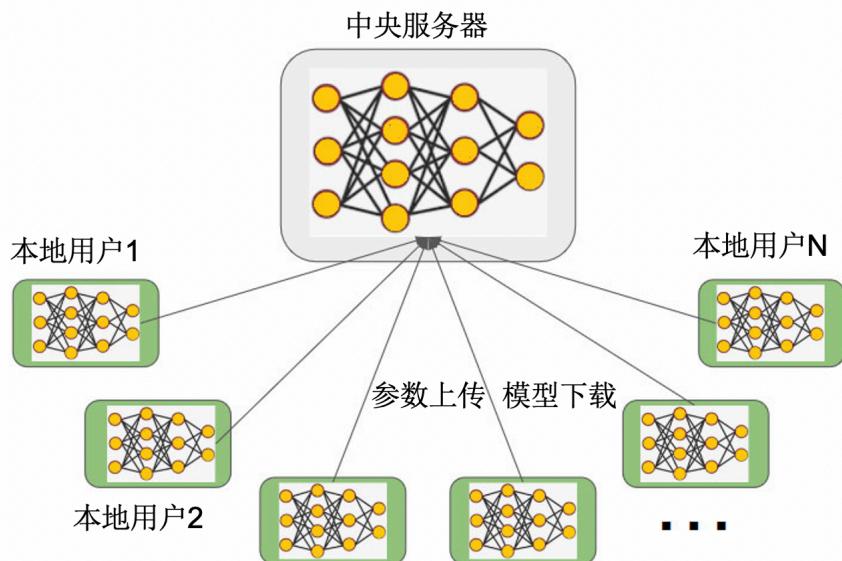


图 1.1: 联邦学习模型概况

联邦学习允许更智能的模型、更低的延迟和更低的功耗。在联邦学习框架中，由于数据通常分布和存储在不同的位置（例如，数据中心和医院），无需集中联合

学习使手机能够协作学习共享的预测模型，同时将所有训练数据保存在设备上，将深度学习的能力与数据存储在云中的需求分离开来，使用户可以在不共享本地数据的情况下在本地设备上进行预测。

## 1.2 安全性和隐私威胁

联邦学习的一个突出优点是它可以在服务器和客户端之间无需任何个人数据交换的情况下进行本地训练，从而防止客户端的数据被隐藏的对手窃听。尽管联邦学习的优势明显，而且与时俱进，但在实际应用之前，还需要对其安全性进行测试。近年来，大量的研究结果表明，联邦学习机制仍然存在安全问题，在训练过程中，本地设备与中央服务器之间的通信信道和传递的模型参数都有可能成为第三方窃取敏感信息的途径，联邦学习的框架仍然存在本地训练数据泄漏等隐私威胁。作为一种新的神经网络训练模型，攻击者可以从共享梯度中跟踪和获取参与者的隐私，联邦学习仍然面临各种安全和隐私威胁。

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来生成样本原型。针对联邦学习中用户本地训练数据的攻击方式包括投毒攻击、模型重建攻击、模型反演攻击、成员推理攻击等。

投毒攻击：在联邦学习中，本地客户端在各自的设备上进行模型训练，将得到的训练参数上传给中央服务器。因为训练参数不需要通过可信机构的检查，所以有一些攻击者将恶意的训练样本注入自己的本地模型中，影响全局模型的更新结果，导致最终的模型预测结果错误甚至全局模型不可用。投毒攻击的影响对于许多企业和行业来说可能是致命的，在医疗部门、航空部门或道路安全方面甚至会危及生命。Marcus Comiter<sup>[56]</sup> 曾使用投毒攻击进行实验，通过对熊猫的图像样本注入微小的恶意数据，导致算法预测结果发生重大变化，将熊猫识别为长臂猿。

模型重建攻击：在这种情况下，敌手的目标是窃取用户的原始训练数据。模型重建攻击需要白盒访问模型的权限，即模型中的特征向量对于敌手必须是已知的，敌手通过对特征向量的知识来重建用户的原始训练数据。对于一些机器学习算法，比如支持向量机（Support Vector Machine, SVM）或 K 最近邻算法（K-Nearest Neighbors, KNN），它们将特征向量存储在模型本身，容易受到重建攻击。攻击者通过解码用户上传的参数更新，反推出用户本地训练集中某条目标数据和其属性值。

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，在模型上添加了一定的隐私保护措施后，攻击也基本失效。

成员推理攻击<sup>[13]</sup>：给定一个深度学习模型和一条数据样本（敌手的知识），成员推理攻击旨在确定样本是否为用于构建此深度学习模型的训练集成员（对手的目标）。这种攻击可能是被对手用来了解某个人的记录是否用于训练深度学习模型，此类攻击利用深度学习模型对训练集中使用的样本与未包含的样本的预测差

异。Shokri 等人采用样本正确标签和目标模型预测的结果作为输入，训练影子模型作为攻击模型，达到推断任意样本是否在训练集中的目的。

### 1.3 国内外研究现状

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联邦学习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噪音以实现隐私保护。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。

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，利用隐私数据参与安全计算，共同完成某项计算任务。安全多方计算是解决协同计算问题的一种解决方案，它必须保证计算中各方信息的保密性、独立性和准确性。SMC 安全模型自然涉及多方，各方除了自己的输入和输出一无所知，以确保完整的零知识安全证明。当前，安全多方计算领域常见的技术主要包括混淆电路、零知识证明、不经意传输和秘密共享等。

同态加密是一种加密形式，它允许用户对其加密数据执行计算，这些结果计算以加密形式保留，解密后的输出与未加密时进行相同计算操作产生的结果相同。同态加密可以分为加法同态加密、乘法同态加密和全同态加密。同态加密在联邦学习中的应用主要通过防止服务器对本地客户端上传的权重进行逆向工程从而反推出训练数据，确保每个客户端对全局模型的更改都保持隐藏状态。因为服务端接收的是本地客户端通过同态加密算法处理后的数据，这种模型的安全性是以服务器上的计算成本为代价的<sup>[23]</sup>，这种加密场景的高计算复杂度会严重危害分布式机器学习设置的性能。

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本地模型的局部梯度，将差分隐私机制与模型训练中的随机梯度下降算法（Stochastic Gradient Descent，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准确性，当使用较低的隐私预算达到较强的隐私保护的效果，可能使得模型难以收敛，可用性大幅下降；当隐私保护强度太低时，可能无法防御诸如 GAN 攻击等大规模的生成对抗网络攻击。

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。之后我们设计了动量组合机制分析加噪累积产生的隐私预算，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大大减少了噪声对模型输出结果的影响，提高了模型的准确性。

- (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攻击模型和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，分析属性值的贡献率，根据贡献比率添加对应的高斯噪声，然后采用拉普拉斯平滑机制提升模型的快速收敛，之后采用动量组合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联邦学习系统中，提高系统学习效果，最后证明了安全混淆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。

第六章是对文本的工作内容的总结和未来研究方向的展望。首先对本文的研究内容进行了概括，并总结了现有方案的不足之处，之后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，首先介绍了本文章的研究背景和意义，总结了当前联邦学习发展过程中的挑战和难点，并具体针对联邦学习中的隐私威胁和隐私保护的研究现状做了具体的阐述，最后介绍了文章的主要研究、贡献和组织结构。

## 第二章 基础知识

在本章节中我们将介绍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 神经网络

#### 2.1.1 基本结构

深度学习模型通常采用神经网络的形式。已经为不同的应用提出了各种神经网络架构，例如多层感知器、卷积神经网络和循环神经网络。神经网络<sup>[34]</sup>最初的设计灵感来源于人脑的结构。我们知道，人类的大脑是处理信息的主要部分，也是人中枢神经系统中的重要部分。人脑中含有大量的神经元，它们像网状物一样复杂的相互连接。当人脑接收到外部环境或者感觉器官传入的刺激（兴奋），它随着神经元一层一层的将刺激（兴奋）传导到神经中枢（大脑或脊髓），神经中枢根据接收到的信号，作出不同的判断，最后传递到输出神经。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。

神经网络的基本组成单元是神经元，一个神经网络可能包含数百亿个简单的神经元，它们按层排列，密集而复杂的相互连接着。神经网络中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。每一层的神经元只可能与其前一层和后一层的神经元相连接，不存在跨层连接。

如图2.1所示，一个神经网络由一个输入层、一个输出层以及输入和输出之间

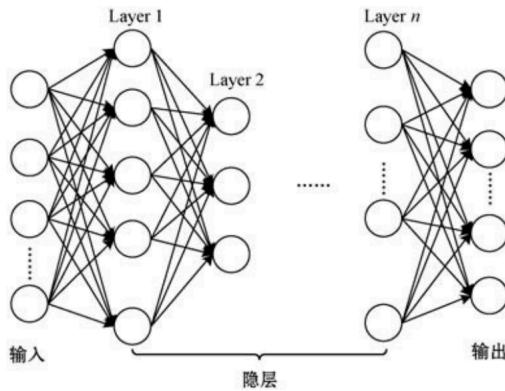


图 2.1: 神经网络结构图

的一系列隐藏层组成。每一层都是一组称为神经元的单元，它们连接到上一层和下一层的其他神经元。输入层用于接收信息，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，然后训练数据被送入输入层；之后传入隐藏层进行特征的提取、网络权重的调整使得隐藏层的神经单元对某种模式形成反应；最后传导到输出层，输出模型判断的结果。神经元之间的每个连接都可以通过应用线性函数和元素级非线性激活函数（例如 sigmoid 或 ReLU）将信号传输到下一层的另一个神经元。通过这种方式，神经网络通过隐藏层转换输入，然后转换输出。

反向传播和随机梯度下降是训练深度学习模型和寻找最佳参数的常用方法。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。

### 2.1.2 随机梯度下降算法

随机梯度下降算法（Stochastic Gradient Descent, SGD）是一种主流的用于机器学习和深度学习模型优化的迭代方法，从随机的权重系数开始，迭代运行梯度下降算法，使损失函数收敛到局部最小值，以找到使模型达到全局最优的权重系数，具体的算法如所示。

**Algorithm 1** 随机梯度下降算法

- 
- 1: 输入: 学习率  $\alpha$
  - 2: 输出: 初始参数  $\theta$
  - 3: 初始化模型权重  $\theta$ , 作为梯度下降的起始点
  - 4: **while** 模型未达到全局最优点 **do**
  - 5: 从训练集中均匀抽出一小批量 (minibatch) 样本:  $\mathbb{B} = \{x^{(1)}, \dots, x^{(m')}\}$
  - 6: 计算梯度估计:

$$g = \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta)$$

- 7: 梯度下降:
- $$\theta = \theta - \epsilon g$$

- 8: **end while**
- 

**2.1.3 经验风险最小化**

在神经网络中, 模型通过不断的学习数据集中的特征得到预测值, 通过损失函数计算预测值与真实值之间的误差, 之后再采用反向传播算法调整权重系数使得最终的损失函数的值最小。整个模型训练的过程可以理解为经验风险最小化 (Empirical risk minimization, ERM) 问题:

$$F(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n f_i(\boldsymbol{\theta}) \quad (2.1)$$

模型在训练集  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$  上进行训练, 其中,  $F(\boldsymbol{\theta})$  表示经验损失函数;  $f_i(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}; \mathbf{x}_i, y_i)$  表示在第  $i$  个训练样本  $(\mathbf{x}_i, y_i)$  上定义的损失函数;  $\boldsymbol{\theta} \in \mathbb{R}^d$  表示模型最终训练得到的权重参数。模型的训练目标是找到最终的权重参数  $\hat{\boldsymbol{\theta}} \in \mathbb{R}^d$ , 使得公式3.1所计算得到的经验风险值最小。

**定义 2.1.1.** 当  $\|\nabla f(\boldsymbol{\theta})\|_2 \leq \zeta$  时,  $\boldsymbol{\theta} \in \mathbb{R}^d$  是使得 ERM 函数达到全局最优解的参数

**定义 2.1.2.** 对于函数  $f: \mathbb{R}^d \rightarrow \mathbb{R}$ , 如果对于任意的  $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$ , 都有  $|f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2)| \leq G \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2$ , 则称函数  $f$  满足  $G$ -Lipschitz ( $G$ -利普希茨连续条件)。

**定义 2.1.3.** 对于函数  $f: \mathbb{R}^d \rightarrow \mathbb{R}$ , 如果对于任意的  $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$ , 都有  $\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\|_2 \leq L \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2$ , 则称函数  $f$  满足  $L$ -Lipschitz ( $L$ -利普希茨连续条件)。

## 2.2 联邦学习

传统的集中式深度学习需要将训练数据放在一起到数据中心。该模型以集中方式进行训练。而联邦学习允许数据所有者拥有一个私人学习网络，该网络使用本地数据集进行训练。之后，每个参与者将本地模型的梯度上传到云服务器。通过使用云服务器收集的全局梯度进行更新，可以避免局部模型过度拟合。此外，它还保护本地数据不被其他参与者或云服务器直接知道。联邦学习的基本工作流程如下：

- **初始化：**所有用户在各自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，然后在各自的本地数据集  $D_i$  上进行模型训练，更新模型参数： $\omega_i^{r+1} \leftarrow \omega_i^r - \eta_i \nabla g(D_i^t, \omega_i^r)$
- **中央参数聚合：**中央服务器等待所有本地客户端将更新后的模型参数  $M1, M2 \dots M_n$  上传，聚合得到全局模型的参数，之后更新全局模型： $\omega^{r+1} \leftarrow \omega^r - \eta \frac{\sum_{U_i \in U^t} \omega_i}{\sum_{U_i \in U^t} |D_i^t|}$
- **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。

## 2.3 差分隐私

### 2.3.1 基本定义

**定义 2.3.1** (邻近数据集). 现有两个属性相近的数据集  $D$  和  $D'$ ，他们的数据记录差为  $D \Delta D'$ ，如果  $|D \Delta D'| = 1$ ，则称数据集  $D$  和  $D'$  为邻近数据集 (*Adjacent Dataset*)。

2016 年，Dwork<sup>[14]</sup> 等人首次提出了差分隐私的概念，它在针对数据隐私泄漏的新型隐私定义，目的是使数据库的查询函数对数据集中单条记录的变化不敏感。其思想是添加一定量的噪音来随机化给定算法的输出，从而使攻击者无法区分任何两个相邻的输入数据集的输出。具体的定义如下：

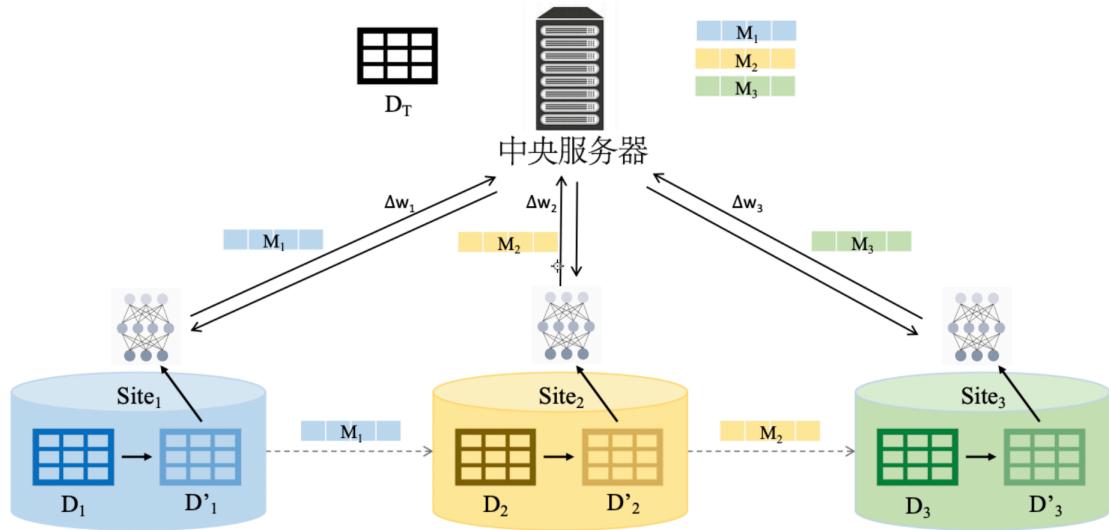


图 2.2: 联邦学习模型工作流程

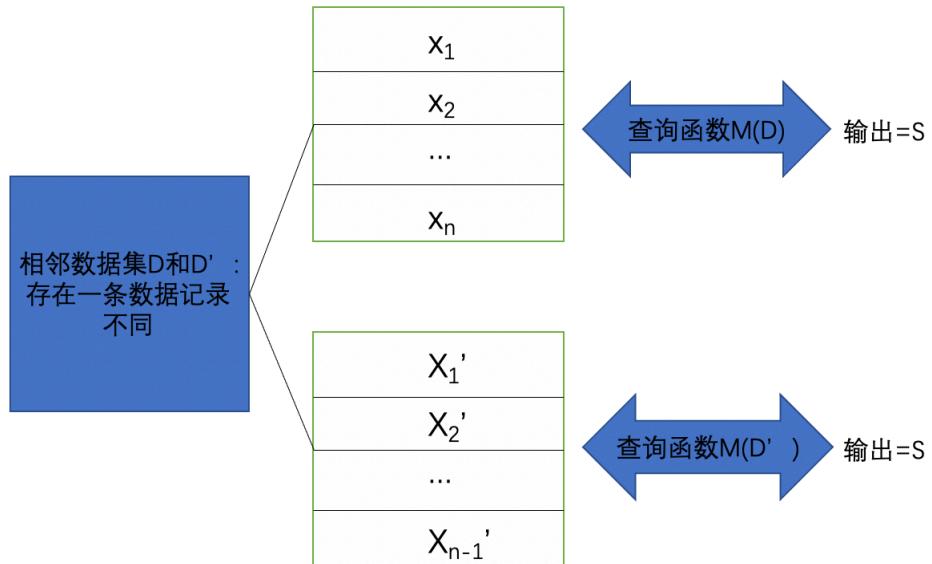


图 2.3: 差分隐私的相邻数据集示意图

**定义 2.3.2 (( $\varepsilon, \delta$ )-差分隐私).**  $\mathcal{D}$  表示数据集合,  $D$  和  $D'$  为邻近数据集。现有随机算法  $M : D \rightarrow R$ ,  $D$  表示定义域,  $R$  表示值域。如果对于任意两个邻近数据集  $S, S' \in \mathcal{S}^n$  和输出子集  $O \subseteq \mathcal{R}$  时, 总有

$$\Pr[\mathcal{M}(D) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(D') \in S] + \delta$$

, 则称该随机算法满足  $(\varepsilon, \delta)$ -差分隐私。

添加项  $\delta \in [0, 1]$  表示以某种概率打破  $(\epsilon, 0)$ -差分隐私。当  $\delta = 0$  时，则将  $M$  称为  $\epsilon$ -差分隐私。 $\epsilon$  和  $\delta$  表示隐私预算参数， $\epsilon$  和  $\delta$  越小，算法能提供的隐私保证程度越强。

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。对于相邻数据集  $D$  和  $D'$ ，他们的敏感度代表某一个查询函数在这两个相邻数据集上输出的最大不同。查询函数的类型决定了敏感度，也为噪声的添加提供了依据。

**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。

### 2.3.2 实现机制

在差分隐私的实际应用中，如何针对不同的场景和问题设计添加噪声的机制使算法能满足差分隐私保护的要求呢？差分隐私的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>。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隐私保护<sup>[35]</sup>。

**定理 2.3.4 (拉普拉斯机制).** 给定一个基于数据集  $D$  的查询函数  $f(D)$ ，算法  $\ddot{f}(D)$  满足  $\epsilon$ -差分隐私，当：

$$\ddot{f}(D) = f(D) + \text{Lap}\left(\frac{GS}{\epsilon}\right)$$

其中，噪声参数满足  $\text{Lap}\left(\frac{GS}{\epsilon}\right)$  的 Laplace 分布， $GS$  表示数据集的敏感度。

与拉普拉斯机制类似，高斯机制对输入数据的所有维度添加满足高斯分布的噪声。

**定理 2.3.5 (高斯机制).** 一个查询函数  $f : D \rightarrow R$ , 该算法的敏感度表示为  $S_f$ , 算法  $M$  满足  $\epsilon$ -差分隐私, 当:

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, S_f^2 \cdot \sigma^2)$$

其中,  $\mathcal{N}(0, S_f^2 \cdot \sigma^2)$  是满足正态(高斯)分布的, 均值为 0, 标准差为  $S_f\sigma$ 。当  $\epsilon \in (0, 1]$ ,  $\sigma \geq \sqrt{2 \ln(1.25/\delta)} \Delta_A / \epsilon$ , 算法  $M$  满足  $(\epsilon, \delta)$ -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。

**定理 2.3.6 (指数机制).** 指数机制满足差分隐私, 如果:

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

其中  $u(D, p)$  为评分函数, 评分越高, 则输出的概率越大<sup>[53]</sup>,  $\Delta u$  表示  $u(D, p)$  的全局敏感度。

### 2.3.3 相关定理

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。组合性是指将独立的满足差分隐私的算法进行串行组合或者并行组合之后得到的算法依然满足差分隐私。

**定理 2.3.7.** 对于任意满足  $(\varepsilon, \delta)$ -差分隐私的算法  $\mathcal{M}_1$  和  $\mathcal{M}_2$ , 算法  $\mathcal{M}_3$ :  $\mathcal{M}_3(\vec{x}) = (\mathcal{M}_1(\vec{x}), \mathcal{M}_2(\vec{x}))$  也满足  $(\varepsilon, \delta)$ -差分隐私。

**定理 2.3.8.** 对于任意满足  $(\varepsilon, \delta)$ -差分隐私的算法  $\mathcal{M}_1, \dots, \mathcal{M}_d$ , 算法  $\overline{\mathcal{M}}$ :  $\overline{\mathcal{M}}(\vec{x}) = (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$  也满足  $\left(\varepsilon \cdot \left(\sqrt{2d \ln(1/\delta)} + (e^\varepsilon - 1) \cdot d\right), \delta \cdot (d+1)\right)$ -差分隐私。

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。在差分隐私的应用程序中, 通常结合串并行组合定理分析算法累积的总体隐私预算和隐私成本。

对于一个随机算法设计满足差分隐私的方案通常包括以下步骤:

- (1) 通过敏感度有界函数的组合来设计逼近的系统
- (2) 选择合适的噪声机制和参数实现差分隐私
- (3) 结合串并行组合定理分析算法累积的总体隐私预算和隐私成本

## 2.4 本章小结

本章节为基础知识, 对于论文的研究所涉及的基础知识定理进行了讲解。本章主要介绍了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 联邦学习中的本地自适应差分隐私机制

### 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，在训练过程中，本地设备与中央服务器之间的通信信道和传递的模型参数都有可能成为第三方窃取敏感信息的途径，联邦学习的框架仍然存在本地训练数据泄漏等隐私威胁<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。

在第二章的基础知识中曾讲到，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躁后的训练数据<sup>[37][38][39]</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  $\operatorname{argmin}$  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。在本地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  $\mathcal{D}_i$  和接受的服务器的全局模型  $w_G^t$  作为本地的参数, 即  $w_i^t = w_G^t$ , 采用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,  $n$  表示对矩阵的每一个元素添加噪音)。
- **模型聚合:** 服务器使用参数聚合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。
- **全局收敛:** 重复步骤 (1) - (4) 直至全局模型收敛。

与 SGD 相比，差分隐私随机梯度下降 (DP-SGD) 严重降低了训练模型的效用。如图所示，当差分隐私提供的隐私强度增加时，MNIST 数据集上逻辑回归的训练和验证的损失率迅速增加。由 DP-SGD 训练的卷积神经网络 (CNN) 的测试精度比 MNIST 上的非差分隐私网络低得多。Goodfellow<sup>[64]</sup> 提出了  $\ell_2$  范式梯度裁剪的方式以限制函数敏感度，并设计了“Moments Accountant”(MA) 来计算更准确的隐私预算估计。

然而，在传统的基于差分隐私的联邦学习框架中，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。而基于梯度扰动的方法的问题在于它们的迭代性质会导致隐私预算的飙升。因此，当前的主要挑战是设计一种新型的满足差分隐私的扰动算法，既能保证模型的效用性，并且维持较高的计算效率。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，并在梯度下降的过程中采用拉普拉斯平滑机制保证模型的快速收敛。拉普拉斯平滑 (Laplacian Smoothing, LS) 可以看作是一种对高斯噪声注入的随机梯度进行后处理的去噪技术。

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，通过利用通信信道访问用户梯度的便利，它也试图在训练过程中反推出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本地训练样本信息，并且尽量维持原有模型的精度。为实现相同的效用保证，我们算法的梯度复杂度 (即计算的随机梯度总数) 为  $O(n^{3/2})$ ，比之前的最佳结果高出  $\Theta(n^{1/2})$ 。之后我们在凸 ERM 和非

凸 ERM（逻辑回归和卷积神经网络）上进行实验，评估我们提出的方法，发现我们的方法不仅产生了在模型精度方面最接近非差分隐私的模型，而且还降低了计算成本。

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，从以下三个方面展开研究：

- (1) 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高斯噪声。
- (2) 对于添加高斯噪声的梯度添加拉普拉斯平滑机制。
- (3) 根据训练轮数和层间贡献率对梯度进行自适应裁剪。

## 3.2 相关理论

### 3.2.1 自适应噪声添加算法

在第二章我们详细介绍了联邦学习的流程和神经网络的结构，每个用户在本地设备在私有数据集上进行训练，得到本地模型的输出。本地训练的流程如图3.1所示，神经网络中前向传播算法的第一步在输入层。我们使用前馈神经网络接收输入的  $x$  运行前向传播算法，得到预测值，然后通过反向传播算法不断调整参数使与预测值和真实值之间的误差降低。Bach 等人提出了针对神经网络的逐层关联传播算法<sup>[65]</sup>，它允许分解深度神经网络的预测值，我们利用该算法将神经网络的输出值按层进行分解，得到每层的属性值对于模型输出的贡献比，然后根据属性的贡献率，在梯度下降的过程中添加对应比的高斯噪声。

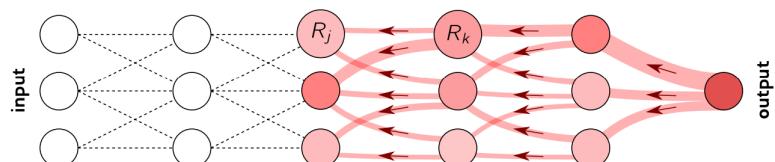


图 3.1: 神经网络的前向传播和反向传播流程图

在逐层关联传播算法中，根据神经网络的结构自后向前计算归因值，也称为相关性分数。对于卷积神经网络等网络结构，从输出层开始根据反向传播算法计算归因分数，输出层的归因分数通常作为输出层的预激活值，在反向传播过程中，每一层的所有神经元的归因分数总和是恒定。

在卷积神经网络结构中，每个隐藏神经元的转化过程表示为  $y = a(\mathbf{x} * \omega + b)$ ，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。

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个隐藏层的线性变换所利用。直白地说，为了得到一个具有隐私保护的学习模型，我们可以在第一个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，根据属性对于模型输出的贡献率添加对应比例的高斯噪声。

令  $C_{a_i}^{l_k}(x_i)$  表示第  $k$  层的神经元  $a_i$ ，” $\leftarrow$ ” 表示神经网络每一层之间的连接关系。” $l_k \leftarrow l_{k+1}$ ” 是指深度神经网络中第  $k$  层和第  $k+1$  层之间相邻层的连接关系。对于模型输出的贡献率，根据神经网络相邻层间的线形关系，那么神经元  $a_i$  的贡献率即为与之相邻的第  $k+1$  层的神经元的贡献率之和：

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

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：

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

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

$$C_{a_i \leftarrow a_j}^{l_{k-1} \leftarrow l_k}(x_i) = \begin{cases} \frac{a_i w_{i,j}}{\sum_{a_i \in l_{k-1}} a_i w_{i,j}} C_{a_j}^{l_k}(x_i) & \sum_{a_i \in l_{k-1}} a_i w_{i,j} \neq 0 \\ \mu & \sum_{a_i \in l_{k-1}} a_i w_{i,j} = 0 \end{cases} \quad (3.4)$$

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。那么模型的输出可以表示为层神经元贡献值的累加：

$$\begin{aligned} \sum f(x_i, \omega_i^r) &= C_{a_8}^{l_3}(x_i) + C_{a_9}^{l_3}(x_i) \\ &+ C_{a_4}^{l_2}(x_i) + C_{a_5}^{l_2}(x_i) + C_{a_6}^{l_2}(x_i) + C_{a_7}^{l_2}(x_i) \\ &+ C_{a_1}^{l_1}(x_i) + C_{a_2}^{l_1}(x_i) + C_{a_3}^{l_1}(x_i) \end{aligned}$$

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：

$$C_j(x_i) = \frac{1}{n} \sum_{i=1}^n C_{x_{i,j}}(x_i), j \in [1, u] \quad (3.5)$$

在原始的参数上计算神经网络中每个梯度对于模型输出的贡献后，按照公式3.6采用高斯机制根据属性类的贡献率在属性值中注入噪音以保护原始的参数。

$$\tilde{C}_j(x_i) = C_j(x_i) + \mathcal{N}(0, S_f^2 \cdot \sigma^2), j \in [1, u] \quad (3.6)$$

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

然后，我们引入了两个调整参数  $f$  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归因分数是高还是低，其值由联邦学习中的本地用户自己定义，即贡献超过阈值  $f$  的梯度对输出的贡献更大，我们向所有这些梯度注入自适应拉普拉斯噪音。当归隐分数低于阈值  $f$  时，对这些梯度进行概率选择。也就是说，我们抛弃概率为  $1 - p$  的原始数据，并对一些概率为  $p$  的梯度注入自适应拉普

拉斯噪声。该公式如下：

$$\tilde{x}_{i,j} = \begin{cases} \ddot{x}_{i,j} & \beta \geq f \\ \bar{x}_{i,j} & \beta < f \end{cases} \quad (3.7)$$

其中  $\beta$  代表该层神经元的归因分数： $\beta = \frac{|\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|}$ ，当  $\beta < f$  时，我们有：

$$\bar{x}_{i,j} = \begin{cases} \ddot{x}_{i,j} \text{ with probability } p \\ x_{i,j} \text{ with probability } 1 - p \end{cases} \quad (3.8)$$

在此方案中，隐私预算  $\epsilon_l$  是根据各自的归因分数按比例分配给每个梯度： $\sigma = \sigma_j = \frac{u * |\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|} * \sigma_l$ ，自适应高斯噪声按以下方式注入梯度中：

$$x'_{i,j} = x_{i,j} + \frac{1}{|D_i^t|} \mathcal{N}(0, S_f^2 \cdot \sigma^2) \quad (3.9)$$

在不失一般性的情况下，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，代表越高的隐私保护水平，模型准确性越低，反之亦然。算法 ref 自适应高斯噪声算法详细描述了自适应高斯噪声算法，我们将层间依赖算法于随机梯度下降算法相结合，关键的步骤包括：计算梯度以及归因分数，根据归因分数分配相应的隐私预算，在梯度上添加高斯噪声，最后进行梯度下降。

### 3.2.2 拉普拉斯平滑机制

在本章的开头曾介绍了联邦学习中的 ERM 问题，它是模型训练的关键问题：在梯度下降的过程中，如何快速使 ERM 函数收敛，达到全局的最优点，使模型的输出值与真实值最接近，提高模型精度。因为我们已经在梯度中添加了高斯噪声，对模型的收敛速度也会产生影响，当 ERM 非凸时，可能导致函数无法达到全局最优点。因此，在本节中，我们提出了拉普拉斯平滑机制，针对加噪后的梯度，采用拉普拉斯平滑机制使随机梯度下降算法在非凸 ERM 和凸 ERM 的情况都能达到全局收敛。

### 3.2.3 自适应梯度裁剪

在传统的差分隐私随机梯度下降算法中，

**Algorithm 2** 自适应高斯噪声算法

---

```

1: 输入: 数据集  $\{x_1, \dots, x_N\}$ , 损失函数  $\mathcal{L}(\theta) = \frac{1}{N} \sum_i \mathcal{L}(\theta, x_i)$ , 学习率  $\eta_t$ , 隐私预算  $\sigma$ , 批大小  $L$ 
2: 初始化: 模型权重  $\theta_0$ 
3: for  $t \in [T]$  do
4:   以概率  $L/N$  随机采样一批数据集  $L_t$ 
5:   计算梯度
6:    $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$ 
7:   计算梯度的贡献率

$$\beta = \frac{|\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|}$$

8:   根据贡献率分配相应的隐私预算

$$\sigma == \frac{u * |\tilde{C}_j|}{\sum_{j=1}^u |\tilde{C}_j|} * \sigma_l$$

9:   添加高斯噪声

$$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \sum_i \left( \bar{\mathbf{g}}_t(x_i) + \frac{1}{|D_i^t|} \mathcal{N}(0, S_f^2 \cdot \sigma^2) \right)$$

9:   梯度下降

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$$

10: end for
11: 输出:  $\theta_t$ 

```

---

### 3.3 自适应差分隐私算法

算法的主要思想是基于从先前更新中获得的信息迭代构建差分隐私梯度估计器  $\mathbf{v}_p^t$ 。算法3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，并使用差分隐私组合定理衡量所添加的噪声大小。首先，我们采用先验组合机制计算  $eps_{iter}$  和  $\delta_{iter}$ （算法第1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11-13行）。然后，我们根据  $\mathbf{v}^t = \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t) + (1 - \gamma)(\mathbf{v}_p^{t-1} - \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1}))$  等式不断的迭代更新  $\mathbf{v}^t$ ， $\nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t), \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1})$  都表示小批量的随机梯度， $\mathbf{v}_p^{t-1}$  是在最后一次迭代中计算得到的差分梯度估计器。 $\gamma$  表示动量参数，用于控制先验参数  $\mathbf{v}_p^{t-1} - \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1})$  的衰减率。这样的做法可以让早期的梯度对当前梯度的影响越来越小，如果没有衰减值，模型往往会震荡难以收敛，甚至发散。之后，对梯度进行范数裁剪（算法第17行），在更新后的权重系数  $\mathbf{v}^t$  上添加噪声矩阵为  $\sigma_2 \mathbf{I}_d$  的高斯噪声  $\mathbf{u}^t$ ，以使加噪后的梯度满足差分隐私。高斯随机向量的方差  $\sigma_0^2, \sigma^2$  由我们基于RDP的分析确定，下文将仔细阐述。最后，服务器对用户上传的梯度进

行聚合，并更新模型参数  $w$ 。该算法有三个主要部分：带有动量的梯度估计，满足 RDP 的高斯噪声，自适应梯度裁剪。

---

**Algorithm 3** 差分隐私随机动量优化算法

---

```

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  $l$ , 隐私参数  $\epsilon, \delta$ ,
2: 输出: 模型梯度
3: 初始化模型权重  $\theta^0$ 
4: 初始化动量梯度估计:  $\mathbf{v}_p^0 = \mathbf{v}^0 + \mathbf{u}^0$ 
5: while  $\exists \delta_u < \delta$  do
6:    $n=0$ 
7:    $grad=0$ 
8:   计算  $eps_{iter}$ ,  $\delta_{iter}$ 
9:   for each  $u \in \text{Users}$  do
10:    计算  $\delta_u$ 
11:    if  $\delta_u > \delta$  then
12:      continue
13:    end if
14:    从客户端数据集中随机采样
15:     $gt_u = \nabla l(\theta^t, x)$ 
16:    自适应梯度裁剪
17:     $\theta^{t+1} = \theta^t - \eta_t \mathbf{v}_p^t$ , 其中  $\eta_t = \min \left\{ \zeta / \left( n_0 L \|\mathbf{v}_p^t\|_2 \right), 1 / (2n_0 L) \right\}$ 
18:    添加高斯噪声
19:     $\mathbf{v}^{t+1} = \nabla F_{\mathcal{B}_{t+1}}(\theta^{t+1}) + (1 - \gamma) (\mathbf{v}_p^t - \nabla F_{\mathcal{B}_{t+1}}(\theta^t))$ , 其中噪声满足  $\mathbf{u}^{t+1} \sim N(0, \sigma^2 \mathbf{I}_d)$ 
20:    更新动量梯度估计
21:     $\mathbf{v}_p^{t+1} = \mathbf{v}^{t+1} + \mathbf{u}^{t+1}$ 
22:     $n++$ 
23:  end for
24:   $\theta^0 = \theta^0 - \alpha * grad/n$ 
25: end while

```

---

在算法3中，假使其中的每个优化函数都满足  $G$ -Lipschitz，并且有  $L$ -Lipschitz 的梯度。给定总迭代次数  $T$ ，动量参数  $\alpha$  和一阶驻点的准确率  $\zeta$ 。对于任意的  $\delta > 0$  和隐私预算  $\epsilon$ 。 $b_0$  和  $b$  表示批大小，当噪声参数  $\sigma_0^2 = 14TG^2\alpha / (\beta n^2\epsilon)$  和  $\sigma^2 = 14T((1 - \gamma)\zeta/n_0 + \gamma G)^2 \alpha / (\beta n^2\epsilon)$  时，基于凸 ERM 的自适应差分私有随机优化算法满足  $(\epsilon, \delta)$ -差分隐私。

我们的算法要求每个分量函数  $f_i$  是  $G$ -Lipschitz，并且具有  $L$ -Lipschitz 的梯度。该梯度将用于推导底层查询函数（即算法 3 中的梯度动量  $\mathbf{v}^t$ ）的敏感性，从而确定高

斯噪声。同时，我们采用梯度裁剪技术保证在每轮迭代过程中满足  $\|\nabla f_i(\boldsymbol{\theta}^t)\|_2 \leq C_1$  and  $\|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_i(\boldsymbol{\theta}^{t-1})\|_2 \leq C_2$ ，其中  $C_1$  和  $C_2$  由本地用户预先定义。梯度动量  $\mathbf{v}^t$  的敏感度上界为  $2((1-\gamma)C_2 + \gamma C_1)/b$ 。

在模型的每一轮迭代过程中，算法将计算添加了高斯噪声的梯度  $\mathbf{v}^{t+1} = \nabla F_{\mathcal{B}_{t+1}}(\boldsymbol{\theta}^{t+1}) + (1-\gamma)(\mathbf{v}_p^t - \nabla F_{\mathcal{B}_{t+1}}(\boldsymbol{\theta}^t))$ ，其中噪声满足  $\mathbf{u}^{t+1} \sim N(0, \sigma^2 \mathbf{I}_d)$ 。对梯度注入的噪声量为  $\frac{1}{|D_i^t|} \text{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$ ，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拉普拉斯噪声，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  $S_f\sigma$  的高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  $S_f\sigma$  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裁剪。因此，梯度向量  $g$  被  $gt_u = gt_u / \max\left(1, \frac{\|gt_u\|}{C}\right)$  取代，以达到裁剪阈值  $C$ 。这种裁剪保证了如果  $|gt_u| \leq C$ ，那么  $gt_u$  将被保留，而如果  $|g| > C$ ，它将被裁减为梯度范数  $C$ 。

在本章接下来的两节，我们将给出基于凸 ERM 的自适应差分隐私随机优化算法关于隐私保证和模型效用性的证明，并与前人的方案进行对比。

### 3.4 隐私性证明

根据算法3，在第  $t$  轮迭代过程采用的算法为  $\mathcal{M}_t$ ，由  $0 \sim t$  轮的高斯噪声组成： $\mathcal{G}_0, \dots, \mathcal{G}_t$ ，其中  $\mathcal{G}_0 = \nabla F_{\mathcal{B}_0}(\boldsymbol{\theta}^0) + \mathbf{u}^0$ ， $\mathcal{G}_t = \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t) - (1 - \gamma)\nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1}) + \mathbf{u}^t$ 。因此本节证明  $\mathcal{M}_t$  是满足差分隐私的。假设模型的训练集为  $S$ ， $S'$  表示与  $S$  第  $i'$  个数据记录不同的相邻数据集。

对于算法  $\mathcal{M}_t$  给出严格的隐私证明存在两个难点：1. 算法中的子采样机制  $\{\mathcal{G}_i\}_{i=0}^{T-1}$ ；2. 当  $t > 0$ ，如何控制  $\mathcal{G}_t$  的敏感度。第一个难点可以通过我们的子采样定理??的隐私放大来解决，这为我们提供了隐私保证的紧密封闭形式。对于第二个难点，我们可以通过使用自适应步长，使用更少量的随机噪声来实现差分隐私。

根据算法3， $\mathcal{G}_t$  表示在从训练集  $S$  中均匀采样的样本集  $\mathcal{B}_t$  上应用高斯机制  $\tilde{\mathcal{G}}_t$ ：

$$\tilde{\mathcal{G}}_t = \begin{cases} \frac{1}{b} \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^0) + \mathbf{u}^0, & t = 0 \\ \frac{1}{b} \sum_{i=1}^n (\nabla f_i(\boldsymbol{\theta}^t) - \phi \nabla f_i(\boldsymbol{\theta}^{t-1})) + \mathbf{u}^t, & t > 0 \end{cases}$$

其中， $\phi = 1 - \gamma$ 。对于  $\tilde{\mathcal{G}}_0$  中的  $\tilde{\mathbf{q}}_0 = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^0) / b_0$ ， $\Delta(\tilde{\mathbf{q}}_0)$  的敏感度由下式决定：

$$\|\tilde{\mathbf{q}}_0(S) - \tilde{\mathbf{q}}_0(S')\|_2 \leq \frac{1}{b} \|\nabla f_i(\boldsymbol{\theta}^0) - \nabla f_{i'}(\boldsymbol{\theta}^0)\|_2 \leq \frac{2G}{b_0}$$

该式的最后一个不等式由算法3中的每个子函数的  $G$ -Lipschitz 决定，对于  $\tilde{\mathcal{G}}_t$  中的  $\tilde{\mathbf{q}}_t = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^t) / b - (1 - \gamma) \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^{t-1}) / b$ ，当  $t > 0$  时，函数  $\Delta(\tilde{\mathbf{q}}_t) = \|\tilde{\mathbf{q}}_t(S) - \tilde{\mathbf{q}}_t(S')\|_2$  的敏感度由下式决定：

$$\frac{1 - \gamma}{b} \|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_i(\boldsymbol{\theta}^{t-1}) + \nabla f_{i'}(\boldsymbol{\theta}^t) - \nabla f_{i'}(\boldsymbol{\theta}^{t-1})\|_2 + \frac{\gamma}{b} \|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_{i'}(\boldsymbol{\theta}^t)\|_2$$

因此，可以推理出：

$$\begin{aligned} \|\mathbf{q}_t(S) - \mathbf{q}_t(S')\|_2 &\leq \frac{2L(1 - \gamma)}{b} \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\|_2 + \frac{2\gamma G}{b} \\ &= \frac{2L(1 - \gamma)}{b} \eta_{t-1} \|\mathbf{v}_p^{t-1}\|_2 + \frac{2\gamma G}{b} \\ &\leq \frac{2(1 - \gamma)\zeta}{n_0 b} + \frac{2\gamma G}{b} \end{aligned}$$

该式的第一个不等子式由  $L$ -Lipschitz 的连续梯度和每个子函数的  $G$ -Lipschitz 决定。最后一个不等子式由算法中选择的自适应步长  $\eta_t = \min \left\{ \zeta / \left( n_0 L \|\mathbf{v}_p^t\|_2 \right), 1 / (2n_0 L) \right\}$  决定。 $\eta_t$  自适应步长是控制函数  $\tilde{\mathbf{q}}_t$  敏感度有界的关键，如果我们选择一个固定的步长  $\eta_t = 1/(2L)$ ，函数  $\tilde{\mathbf{q}}_t$  敏感度会按照  $O(G^2/b)$  的顺序。这将导致算法需要添加更大的随机噪声来实现差分隐私，从而降低了模型的效用。

根据定理??，如果添加高斯噪声的参数满足  $\sigma_0^2 = 14T\alpha G^2 / (\beta n^2 \epsilon)$  和  $\sigma^2 = 14T\alpha ((1 - \gamma)\zeta/n_0 + \gamma G)^2 / (\beta n^2 \epsilon)$ ，高斯机制  $\tilde{\mathcal{G}}_t$  满足  $(\alpha, \beta\epsilon n^2 / (7b_0^2 T))$ -RDP，子采样造成的隐私放大效应显示  $\mathcal{G}_t$  满足  $(\alpha, \beta\epsilon/T)$ -RDP。因此结合高斯机制和子采样机制的算法  $\mathcal{G}_t$  满足  $(\alpha, \beta\epsilon/T)$ -RDP。根据 RDP 的组合性质，在  $T'$  轮迭代之后，算法3满足  $\alpha = \log(1/\delta)/((1 - \beta)\epsilon) + 1$ -RDP。根据定理??，当  $\alpha = \log(1/\delta)/((1 - \beta)\epsilon) + 1$  时，算法3满足  $(T'\epsilon/T, \delta)$ -差分隐私。

### 3.5 模型效用分析

根据  $\tilde{\boldsymbol{\theta}}$  的定义，可以得到：

$$\mathbb{E} \|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 = \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla F(\boldsymbol{\theta}^t)\|_2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\mathbf{v}_p^t\|_2 + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\|_2$$

其中期望值覆盖算法的所有随机性。对于算法3建立严格的效果保证的关键挑战是如何在自适应步长  $\eta_t$  和梯度  $\mathbf{v}_p^t$  上的随机噪声  $\mathbf{u}^t$  推导出  $\sum_{t=0}^{T-1} \mathbb{E} \|\mathbf{v}_p^t\|_2 / T$  and  $\sum_{t=0}^{T-1} \mathbb{E} \|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\|_2 / T$  的严格上界。

首先，考虑到自适应步长  $\eta_t$ ，推导出  $\sum_{t=0}^{T-1} \mathbb{E} \|\mathbf{v}_p^t\|_2 / T$  的上界：

$$\frac{4n_0 L D_F}{T\zeta} + \frac{1}{T\zeta} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\|_2^2 + 2\zeta$$

其中， $D_F = F(\boldsymbol{\theta}^0) - F(\boldsymbol{\theta}^*)$ 。然后，可以推导出项  $\sum_{t=0}^{T-1} \mathbb{E} \|\mathbf{v}_p^t - \nabla F(\boldsymbol{\theta}^t)\|_2^2 / T$  的上界：

$$\frac{2(1 - \gamma)^2 \zeta^2}{n_0^2 \gamma b} + \frac{2\gamma G^2}{b} + \frac{G^2}{T\gamma b_0} + \frac{Td\sigma^2 + d\sigma_0^2}{T\gamma}$$

该多项式的第一项由自适应步长  $\eta_t$  决定，最后一项由在梯度  $\mathbf{v}_p^t$  上的随机噪声  $\mathbf{u}^t$

决定。该边界的最后一项由  $d\sigma^2/\gamma$  决定，从而验证了通过自适应步长能有效控制  $\mathbf{v}^t$  的敏感度，添加噪声的参数  $\sigma^2$  越小，即能保证模型的效用越高。

最后，结合上述给出的两个上界，可以得到：

$$\mathbb{E}\|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 \leq C_1\zeta + C_2 \frac{\sqrt{LD_F d \log(1/\delta)} G}{n\epsilon\zeta}$$

通过控制  $\zeta$  可以得到  $\zeta = (LD_F d \log(1/\delta))^{1/4} (C_2 G)^{1/2} / (C_1 n \epsilon)^{1/2}$ 。因此， $\mathbb{E}\|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 \leq C_3 \zeta$ ，其中  $C_1, C_2, C_3$  是常数。

### 3.6 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，我们提出动量组合的概念，去计算算法迭代过程中添加噪声所累积的隐私预算成本。

根据差分隐私的并串行组合定理，被查询  $n$  次的数据的隐私预算将增加  $n$  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，会发生预拟合，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>。

根据上文提出的差分隐私的并串行组合定理，我们设计了一个动量组合定理：

**定理 3.6.1 (动量组合定理).** 假使存在算法  $M_i$  满足  $(\epsilon_i, \delta_i)$ -差分隐私，那么对于  $M_{[k]} = (M_1, M_2, \dots, M_k)$ ，有  $M_{[k]}$  也是满足  $(\epsilon, \delta)$ -差分隐私的，其中

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

我们采用朴素贝叶斯机制计算  $\delta$  可以得到：

$$\delta = \sum_{i=1}^k B(i, k, p) [\Pr [L_{d,d'} * i > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * i < -\varepsilon]] \quad (3.10)$$

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  $\eta = H^2/2\sigma^2$ 。因此可以采用动量组合定理这样表达隐私预算损失：

$$\begin{aligned} \Pr [L_{d,d'} * i > \varepsilon] &= \Pr [N(\eta i, 2\eta i) > \varepsilon] \\ &= \Pr \left[ N(0, 1) > \frac{-\eta i + \varepsilon}{\sqrt{2\eta i}} \right] = \Pr \left[ N(0, 1) < \frac{\eta i - \varepsilon}{\sqrt{2\eta i}} \right] \\ &= \Pr \left[ N(0, 1) < \sqrt{\frac{\eta i}{2}} - \frac{\varepsilon\sigma}{\sqrt{2\eta i}} \right] = \Phi \left( \frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) \end{aligned} \quad (3.11)$$

然后，可以计算得到隐私预算：

$$\delta = \sum_{i=1}^k B(i, k, p) \left[ \Phi \left( \frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) - e^\varepsilon \Phi \left( -\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}} \right) \right] \quad (3.12)$$

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：

$$\begin{aligned} &\sum_{i=1}^T B(i, T, p) [\Pr [L_{d,d'} * i > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * i < -\varepsilon]] \\ &\leq \sum_{i=1}^T B(i, T, p) [\Pr [L_{d,d'} * T > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * T < -\varepsilon]] \\ &\leq \Pr [L_{d,d'} * T > \varepsilon] - e^\varepsilon \Pr [L_{d',d} * T < -\varepsilon] \end{aligned} \quad (3.13)$$

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

$$\Phi(\sqrt{\epsilon/2}(1/\alpha - \alpha)) - e^\varepsilon \Phi(-\sqrt{\epsilon/2}(1/\alpha + \alpha)) \leq \delta \quad (3.14)$$

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。

### 3.7 本章总结

## 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。Truex 等人<sup>[49]</sup> 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  $\epsilon_2$  的本地差分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，隐私预算暴增<sup>[43]</sup>。

在本章节，我们在联邦学习框架中，设计了一个全新的安全混洗器，与本地自适应差分隐私相结合，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。本地客户端使用自适应差分隐私在模型训练的梯度下降算法过程中加躁，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，再发送给中央服务器进行聚合。安全混洗器独立于服务器并专门用于本地客户端梯度的子采样、拆分混洗、上传。这个模型通过子采样和拆分混洗两者的结合达到隐私放大效应，降低了

隐私预算，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 安全混淆模型

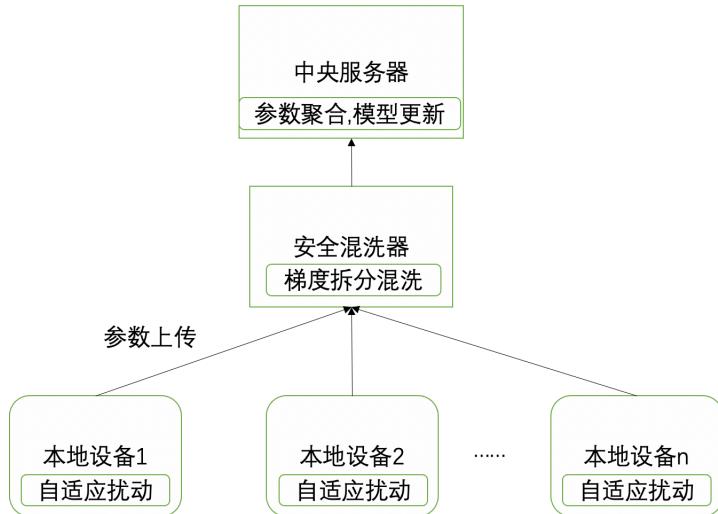


图 4.1: 联邦学习安全模型框架

如图4.1所示，该框架主要由本地客户端、混淆器和中央服务器 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。
- 混淆器：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淆协议在对数据一无所知的情况下，对采样后的梯度完成安全的拆分混淆操作，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，最后将混淆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。服务器接受混淆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  $\epsilon_l$ -差分隐私。在算法4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淆器。混淆器对收到的梯度进行拆分混淆，然后发送给中央服务器。最后，中央服务器对混淆后的梯度进行聚合求均值，更新全局模型。

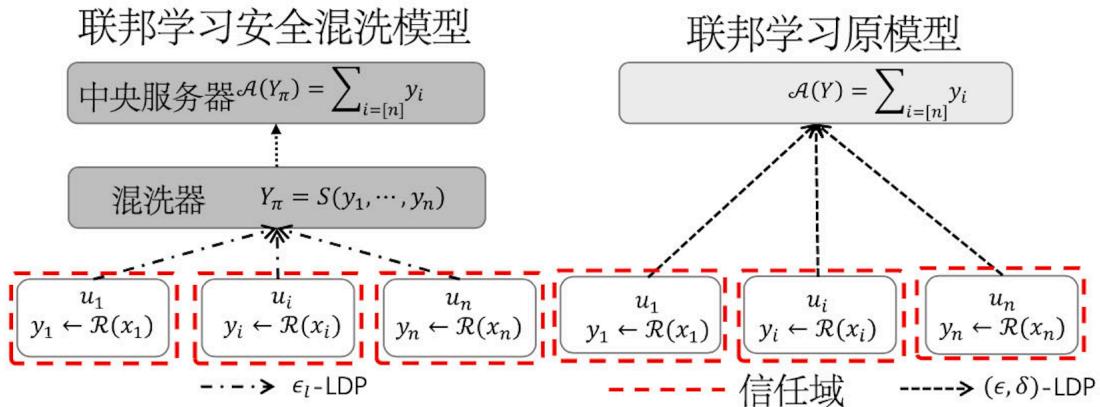


图 4.2: 联邦学习安全模型与原联邦学习模型的信任域对比

**Algorithm 4** 联邦学习中的安全模型算法:  $\mathcal{A}_{\text{csdp}}$ 


---

```

1: 输入: 数据集  $\mathcal{D} = \bigcup_{i \in [m]} \mathcal{D}_i$ ,  $\mathcal{D}_i = \{d_{i1}, \dots, d_{ir}\}$ , 损失函数  $F(\theta) = \frac{1}{mr} \sum_{i=1}^m \sum_{j=1}^r f(\theta; d_{ij})$ ,
   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:  $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}(\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 
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  $q = \frac{r_2}{r_1}$ 。

**4.2.2 混洗器**

McMahan 等人先前的研究工作<sup>[52]</sup> 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，如果使每个客户对服务器产生一定程度的匿名性，就能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、公共 WiFi 接入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，使中央服务器很难结合多个客户端的同步更新来推断任何本地设备的更多信息，具体算法如5所示。

---

#### **Algorithm 5** 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
- 

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

元素分配一个元数据，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。

如图4.3所示，假使现有本地模型  $M_1, M_2, M_3, M_4, M_5$ ，每个模型都有相同的结构，但权重值不同。原始的联邦学习框架是将模型在本地训练后得到的参数直接发送到中央服务器，如图4.3 上半部分所示。

图4.3中的下半部分展示了我们的方案中，首先，对于每个模型，我们分割每个本地模型经过本地训练后所产生的权重。然后，对于每个权重，我们通过随机混洗机制对其进行混洗，并将每个权重及其元数据发送到中央服务器，其中元数据表示该权重值在网络结构中的位置。

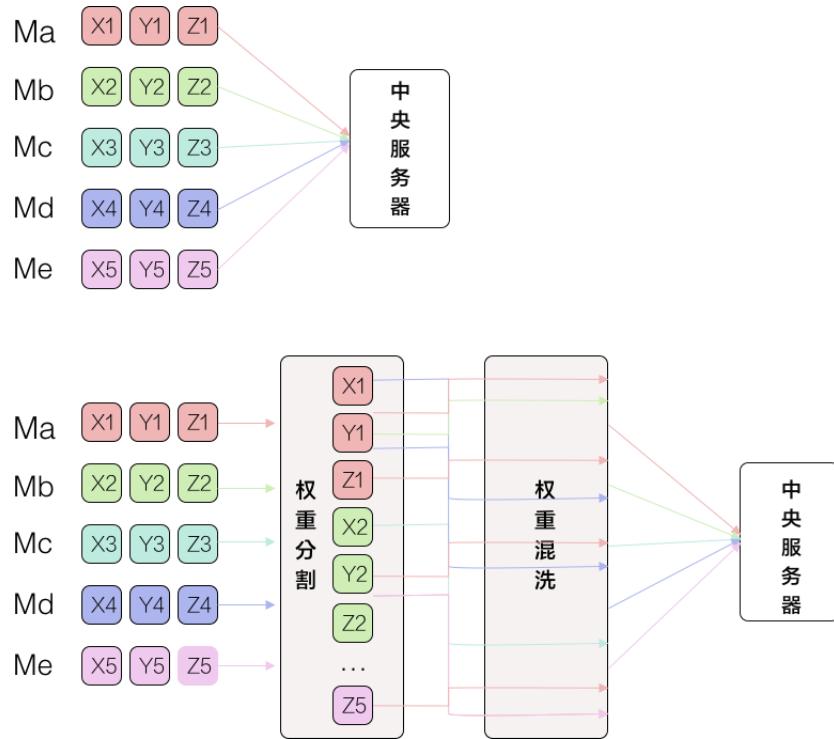


图 4.3: 联邦学习安全混洗模型中执行参数拆分混洗的混洗器

### 4.3 隐私放大力效

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。

在算法4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  $\varepsilon_c - \text{DP}$ 。从  $(\epsilon_c + \epsilon_l)$  到  $\varepsilon_c$  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； $\varepsilon_c$  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  $\mathcal{A}_{\text{csdp}}$  在每次迭代中对每个样本  $d_{ij}$  都能保证  $\epsilon_0$ -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4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洗后的权重。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  $\tilde{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{\log(1/\delta)}{ks}}\right)$ 。

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

在本节中，我们分析采用采样和混洗算法后模型的收敛性。

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，计算本地模型的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/平均数并更新全局模型。

在算法4中，在每一轮迭代过程中，中央服务器聚合上传的  $ks$  个加噪后的梯度，如算法4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： $\theta_{t+1} \leftarrow \prod_C (\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 。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： $\theta_{t+1} \leftarrow \prod_C (\theta_t - \eta_t \mathbf{g}_t)$ ， $\mathbf{g}_t$  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4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  $\sqrt{\frac{cd}{qn} \left( \frac{e^{\epsilon_0} + 1}{e^{\epsilon_0} - 1} \right)} \geq \Omega(1)$  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4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  $C$ ，在模型训练过程中采用随机梯度下降算法  $\theta_{t+1} \leftarrow \Pi_C(\theta_t - \eta_t \mathbf{g}_t)$ ，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<sup>[50]</sup> 中已有的标准随机梯度下降算法收敛结果中使用的定理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 实验评估

### 4.5.1 实验准备

### 4.5.2 实验设计

### 4.5.3 实验分析

## 4.6 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证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洗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。本章的实验主要针对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洗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 个尺寸为 28 x 28 像素的手写数字图像，每个像素点用灰度值表示，灰度值范围为 0 到 255，图像包含十个类别。
- (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。

(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） $32 \times 32$  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。

## 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 Pytorch 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

实验设置了 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神经网络前向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算所有参数的灵敏度  $\Delta f$ ，不同的参数可能导致函数的灵敏度不同。

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能达到的全局模型准确

率，并且将添加隐私保护后的系统精度与未添加隐私保护的模型精度相比较。

### 5.3.2 神经网络模型

Shokri<sup>[51]</sup>等人在论文中公开了他们的源代码，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：

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

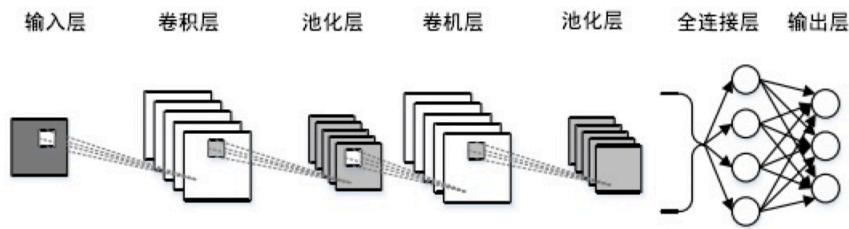


图 5.1: 卷积神经网络结构图

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模型的损失，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。为了展现学习模型的随机性，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CNN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。

实验中使用 60000 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：在公共数据集上进行训练，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  $\delta = 10^{-5}$ ，固定的梯度裁剪阈值为 0.001。如图5.2所示，隐私预算参数  $\epsilon$  为研究变量。隐私预算代表着隐私保护强度，两者呈负相关的关系。隐私预算越大意味着添加的噪声量越小，隐私保护的强度越低，模型的精度越高，符合上文的理论分析。当隐私预算  $\frac{\epsilon}{c} \geq 5$  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，与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：之后我们比较了在不同隐私预算下的自适应干扰模型的准确性，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超参数设置 (a):  $f = 0.15, p = 0.85$ , (b):  $f = 0.10, p = 0.90$  (c):  $f = 0.05, p = 0.95$ 。在实验中，隐私预算  $\epsilon$  的值是  $\epsilon_c$ 、 $\epsilon_l$  和  $\epsilon_f$  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

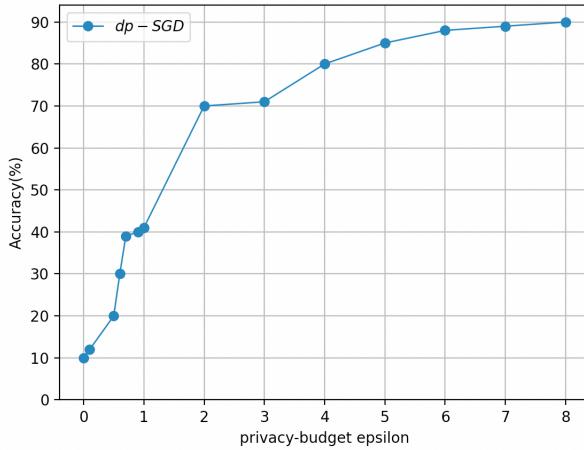


图 5.2: 梯度固定加噪方法下模型准确率随隐私预算变化情况

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，自适应差分隐私联邦学习模型的准确率高达 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%。

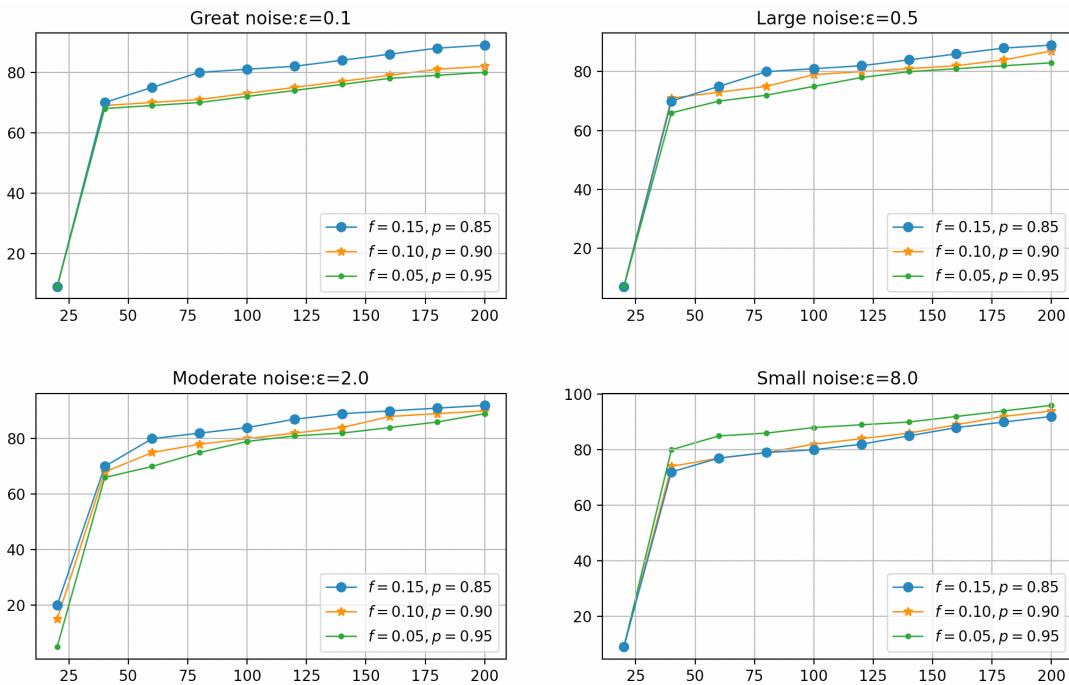


图 5.3: 不同隐私预算的自适应干扰机制在 MINIST 数据集上的准确率

综上，自适应隐私预算分配可以根据模型训练的收敛规律，合理地分配隐私参数，从而提高模型表现，但隐私预算参数需要小心选取，过大的隐私预算参数会

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差分隐私机制保护深度学习模型隐私的工作（DLPP 和 DP-SGD）进行了比较。从图5.4不难发现我们的方案（ACDP-SGD）即使在强隐私预算下 ( $\epsilon=0.1$ ) 也表现良好。当调整因素设置为  $f = 0.15$  和  $p = 0.85$  时，模型的准确率在 200 个通信回合后还能达到 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%。

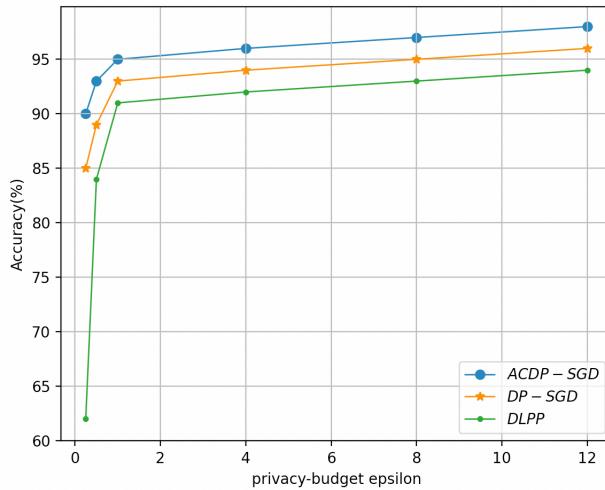


图 5.4: DP-SGD、DLPP、ACDP-SGD 在模型准确率和隐私预算上的对比

## 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。首先评估参数：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 能达到的模型精度与不添加噪声的联邦学习几乎接近。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。

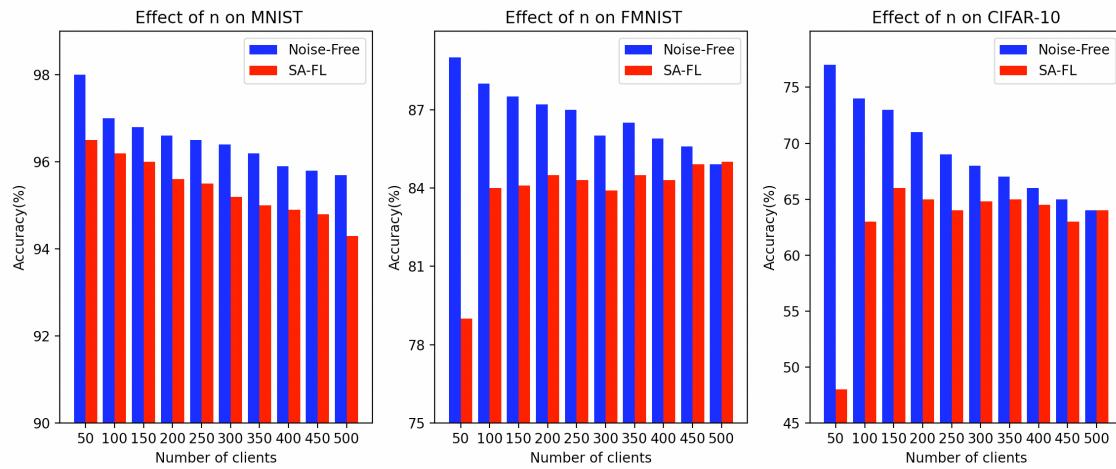


图 5.5: 安全混洗模型中参与混洗的本地客户端数量对联合模型精度的影响

接着，我们分别在 MNIST, FMNIST 和 CIFAR-10 数据集上评估了客户端采样比  $f_r$  和通信回合  $m$  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 上达到与不添加噪声的联邦学习模型几乎相近的性能。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。

最后，我们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<sup>[53]</sup> 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  $(8, 412)$

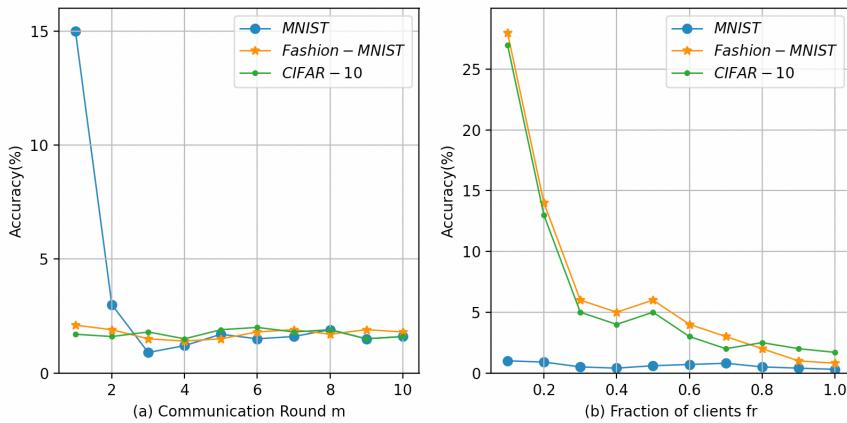


图 5.6: 安全混洗模型中通信轮数和客户端采样比对联合模型精度的影响

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<sup>[54]</sup> 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 轮的通信回合和更高的隐私预算才能使模型收敛。最近, Truex 等人<sup>[55]</sup> 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  $\epsilon = \alpha - 2c - 10\rho$  (例如,  $\alpha = 1, c = 1, \rho = 10$ ) 来实现模型的收敛, 这导致了方案的隐私保证程度太低。与以往的工作相比, 我们的方案大大减少了客户端和中央服务器之间需要的通信回合 (例如, MNIST 为 10, FMNIST 和 CIFAR-10 为 15 就能达到全局模型收敛), 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。

## 5.6 结果分析

为了验证自适应扰动算法在隐私保护的同时, 也能使模型训练的精度维持在较优的水平, 我们进行了对比实验, 使用自适应扰动算法和使用固定加躁方法在不同的隐私参数  $\epsilon$  下进行对比实验。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参数聚合时个体信噪比不变,

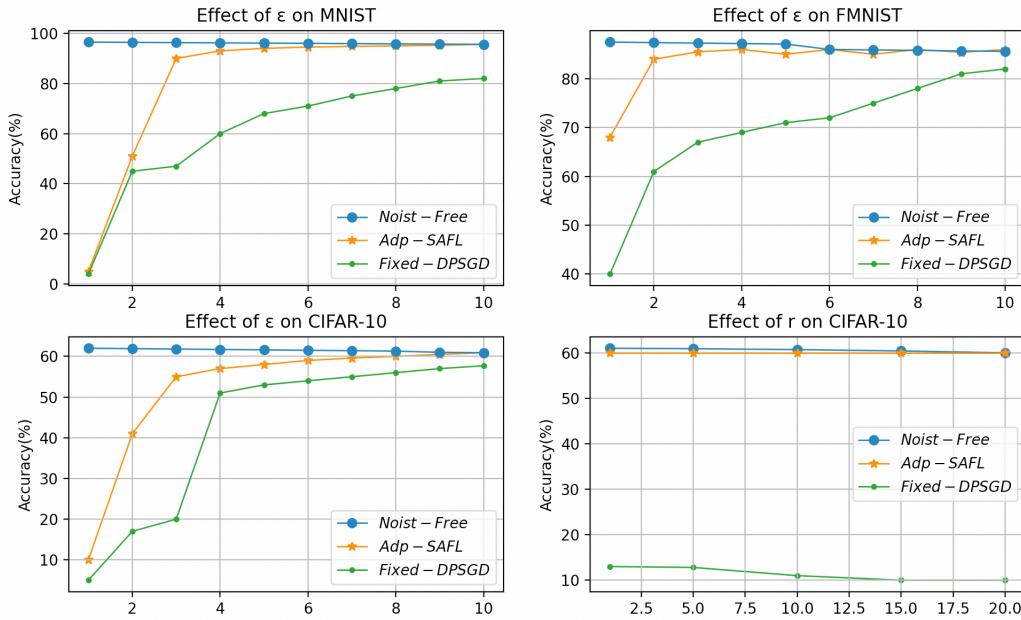


图 5.7: 自适应差分混淆模型和其他联邦学习隐私保护模型的比较

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。

联邦学习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权重贡献率的计算和梯度的扰动。我们使用 20 个通信回合来训练中央服务器的初始化模型，这平均需要 68.22 秒。在本地模型的训练开始之前，用户需要使用前向传播算法计算权重。这个过程只需要训练神经网络前向传播算法，而不需要训练反向传播来计算损失函数，进行梯度下降，其平均耗时为 4.35 毫秒。为了减轻隐私威胁，我们提出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权重的贡献率同步进行，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。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，使得用户数据与用户所使用的编码器之间的关联性消失，使得中央服务器的计算代价增大。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方案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

随着人工智能深度学习的快速发展，出现了越来越多的复杂模型和算法，能够有效的解决各类难题。基于人工智能的产品给医疗、教育、金融、工业等各个领域带来了新一波的发展热潮，也使人们的生活水平大大提高。然而，用户在享受深度学习模型所带来的便利时，也带了许多隐私问题。由于人工智能的基础是基于大数据，许多模型和算法的学习必须基于真实的用户行为数据。而很多深度学习服务的提供方不能有效的保护用户的数据隐私。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。之后，产生了联邦学习框架，它解决了分布式终端用户在本地更新模型的问题，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。它不是将数据上传到中央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，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。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。之后，我们采用动量组合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。最后通过多组真实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固定差分隐私随机梯度下降算法。
- (2) 本文提出了安全混洗框架，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，实现了数据隐私性与模型可用性之间的更好平衡。最后，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，证明了方案的有效性和可行性。

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。在此我将对未来的研方向做出几点展望：

- (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联邦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，推动联邦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.

- [52] McMahan H B, Ramage D, Talwar K, et al. Learning differentially private recurrent language models[J]. arXiv preprint arXiv:1710.06963, 2017.
- [53] Geyer R C, Klein T, Nabi M. Differentially private federated learning: A client level perspective[J]. arXiv preprint arXiv:1712.07557, 2017.
- [54] Bhowmick A, Duchi J, Freudiger J, et al. Protection against reconstruction and its applications in private federated learning[J]. arXiv preprint arXiv:1812.00984, 2018.
- [55] Truex S, Liu L, Chow K H, et al. LDP-Fed: Federated learning with local differential privacy[C]//Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking. 2020: 61-66.
- [56] Comiter M. Attacking artificial intelligence[J]. Belfer Center Paper, 2019: 2019-08.
- [57] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016: 308-318.
- [58] 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.
- [59] Xie L, Lin K, Wang S, et al. Differentially private generative adversarial network[J]. arXiv preprint arXiv:1802.06739, 2018.
- [60] Jordon J, Yoon J, Van Der Schaar M. PATE-GAN: Generating synthetic data with differential privacy guarantees[C]//International conference on learning representations. 2018.
- [61] Zhang J, Zheng K, Mou W, et al. Efficient private ERM for smooth objectives[J]. arXiv preprint arXiv:1703.09947, 2017.

- [62] Wang D, Ye M, Xu J. Differentially private empirical risk minimization revisited: Faster and more general[J]. arXiv preprint arXiv:1802.05251, 2018.
- [63] Wang D, Chen C, Xu J. Differentially private empirical risk minimization with non-convex loss functions[C]//International Conference on Machine Learning. PMLR, 2019: 6526-6535.
- [64] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016: 308-318.
- [65] Bach S, Binder A, Montavon G, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation[J]. PloS one, 2015, 10(7): e0130140.

## 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，非常感谢我的父母和家人一直以来对我的鼓励与陪伴。在研究生生涯的这两年，我更加深刻体会到未来自己身上所担负的责任，希望我在未来的工作中能兢兢业业，踏实负责，实现我的社会价值；在未来的生活中，希望我能多多陪伴我的父母以回报养育之恩。

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，也感谢各位评审和答辩的专家在百忙之中对我论文的指导，谢谢你们。

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] 第一作者, 第二作者. 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]