

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



East China Normal University
硕士专业学位论文
MASTER'S DISSERTATION (Professional)

**论文题目: 基于差分隐私和安全混淆的联邦
学习隐私保护研究**

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

2022 年 02 月 10 日

Thesis (Professional) for Master's Degree in 2022

School Code: 10269

Student Number:51194501126

EAST CHINA NORMAL UNIVERSITY

TITLE: A STUDY OF PRIVACY PRESERVING FEDERATED LEARNING BASED DIFFERENTIAL PRIVACY AND SECURE SHUFFLING

Department:	Software Engineering Institute
Major:	Software Engineering
Research Direction:	Cryptography and Network Security
Supervisor:	Professor ZhenFu Cao
Candidate:	HuiXian He

Feb 10, 2022

摘 要

近年来，人工智能技术在图像识别、语音识别、自动驾驶、智慧医疗等各个领域迎来了一波发展热潮，而深度学习是支撑人工智能快速发展的关键技术。深度学习的应用基于大量的训练数据，其中不乏用户的敏感信息，面临着隐私泄露的风险。此外，由于商业竞争模式和法律法规监管，企业之间无法共享数据以构建一个高质量的模型。联邦深度学习是一种有助于解决多方计算下的数据孤岛问题的学习方法，参与方无需共享本地数据，通过分布式协作训练一个高质量的全局模型，凭借其去中心化、数据隔离、高计算性能等优势成为工业界和学术界的热门研究方向。然而，大量研究表明联邦学习机制存在许多安全漏洞，由于联邦学习的框架并没有对参与方的资质进行校验、没有对模型的访问权加以约束，也并没有考虑到对传递的参数进行保护。这些漏洞可能被内部参与者和外部攻击者所利用，破坏联邦学习系统的安全性。

差分隐私是当前保护联邦学习隐私安全的前沿技术，其通过严格的统计框架提供隐私保证，使得加噪后的梯度不能泄露关于实体数据的敏感信息。然而当前的差分隐私保护联邦深度学习的方案通常面临模型可用性和数据隐私性的博弈。差分隐私的应用可能导致模型的准确率下降，而且随着迭代次数的增多，高维加噪梯度的聚合会导致梯度中所包含的有效信息被噪声所淹没，影响全局模型的收敛性。如何在保护本地数据隐私的同时降低模型的精度损失和通信性能是当前亟需研究的问题。本文设计、实现并评估了一个实用的联邦学习系统，该系统在保护数据隐私的前提下尽可能的维持了模型的可用性和通信效率。本文主要的工作和贡献如下：

- (1) 在差分隐私保护深度学习的场景下，本文设计了一种新型的、基于本地差分隐私的自适应梯度加躁算法。由于不同的神经网络层的神经元对于模型输出的影响不同，本文设计了一种基于神经元的贡献率添加自适应噪声的算法：在客户端本地训练的神经网络模型中，通过运行逐层关联传播算法，计算每个神经元对于模型输出的贡献率。在随机梯度下降过程中，根据贡献率分配动态的隐私预算，在梯度上注入高斯噪声。在设置相同的隐私预算下，相比于固定加躁算法，本方案在相同的隐私保护程度下减少了噪声对模型输出结果的影响。
- (2) 在传统的差分隐私随机梯度下降算法中，通常采用固定的裁剪阈值对梯度进行裁剪以限制函数的敏感度，然而固定的梯度裁剪可能添加额外的噪声。本文设计了一种自适应调整剪裁阈值的方案，通过计算梯度更新的方差和偏差，逐元素地对梯度进行裁剪，根据神经网络各层的均值和统计特征进行梯度裁剪限制敏感度有界，尽可能的保留有效的梯度信息。之后我们利用“*Moments Accountant*”机制分析加噪累积产生的隐私预算，给出更精准的隐私界。
- (3) 由于本地差分隐私并不能有效防御针对联邦学习的生成对抗网络攻击，并且在通信轮数较大的联邦学习模型中，由于差分隐私的强组合性质，噪声量成倍累加，导致整体的隐私成本过高。本文设计了一种新型的联邦学习安全混洗算法，采用指数机制的打分原理挑选出绝对值排名前 k 位的梯度元素，添加拉普拉斯扰动，设计了满足 (ϵ, δ) -差分隐私的 Top-K 梯度选择算法。此外，在联邦学习模型中新增安全混洗器，通过对梯度和索引所构成的矩阵进行置乱，提高了数据的随机性。在每轮通信回合，根据指数衰减机制动态调整客户端采样率，在相同通信回合数下减少通信负荷。通过客户端的采样和梯度索引的混洗达到双重的隐私放大效应，降低系统的整体隐私损失，提高了通信性能，并证明了安全混洗框架的隐私性和全局收敛性。
- (4) 为了验证本文的方案在实际生产环境中的可行性，本文模拟了联邦学习环境，

分别在 MNIST、CIFAR-10、FMNIST 等数据集上进行实验，首先通过控制变量法分析各个参数对于模型精度和通信性能的影响，并与前人的差分隐私方案和安全混淆方案进行对比，通过实验结果证明了自适应本地差分隐私算法和安全混淆算法，在保护数据隐私的前提下尽可能的减小了模型的精度损耗，降低了通信成本。最后，本文模拟了成员推理攻击和生成对抗网络攻击，评估了自适应差分隐私方案和安全混淆方案针对攻击模型的隐私保护效用。

关键词： 联邦学习，隐私保护，差分隐私，安全混淆

ABSTRACT

In recent years, artificial intelligence technology has ushered in a wave of development boom in various fields such as image recognition, voice recognition, autonomous driving, and intelligent medical care, and deep learning is a key technology to support the rapid development of artificial intelligence. The key technology. The application of deep learning is based on a large amount of training data, which is full of users' sensitive information and faces the risk of privacy leakage. In addition, due to competitive business models and legal regulations, data cannot be shared among companies to build a high-quality model. Federated deep learning is a learning method that helps to solve the problem of data silos under multi-party computing, and the participants do not need to share local data to train a high-quality global model through distributed collaboration, which has become a popular research direction in industry and academia with its advantages of decentralization, data isolation, and high computational performance. However, a large number of studies have shown that the federated learning mechanism has many security vulnerabilities due to the fact that the framework of federated learning does not verify the qualifications of the participants, does not impose constraints on the access rights of the model, and does not take into account the protection of the passed parameters. These vulnerabilities can be exploited by both internal participants and external attackers to undermine the security of federal learning systems.

Differential privacy is a current cutting-edge technology to protect the privacy security of federation learning, which provides privacy guarantees through a strict statistical framework so that the post-manicured gradient cannot reveal sensitive information

about the entity data. However, current schemes for differential privacy-preserving federal deep learning usually face a game of model availability and data privacy. The application of differential privacy may lead to a decrease in model accuracy, and the aggregation of high-dimensional agitated gradients as the number of iterations increases can cause the valid information contained in the gradients to be overwhelmed by noise, affecting the convergence of the global model. How to reduce the accuracy loss and communication performance of the model while preserving local data privacy is an urgent research problem at present. In this paper, we design, implement, and evaluate a practical federal learning system that maintains the usability and communication efficiency of the model as much as possible while preserving data privacy. The main work and contributions of this paper are as follows:

1. In the scenario of differential privacy-preserving deep learning, this paper designs a novel, local differential privacy-based adaptive gradient-additive mania algorithm. Since the neurons of different neural network layers have different effects on the model output, this paper designs an algorithm for adding adaptive noise based on the contribution rate of neurons: in the client-side locally trained neural network model, the contribution rate of each neuron to the model output is calculated by running a layer-by-layer associative propagation algorithm. During stochastic gradient descent, a dynamic privacy budget is assigned based on the contribution rate, and Gaussian noise is injected on the gradient. With the same privacy budget set, this scheme reduces the effect of noise on the model output results with the same degree of privacy protection compared to the fixed plus agitation algorithm.
2. In traditional differential privacy stochastic gradient descent algorithms, a fixed clipping threshold is usually used to crop the gradient to limit the sensitivity of the function, however, the fixed gradient crop may add extra noise. In this paper, we design an adaptive adjustment of the clipping threshold scheme to crop

the gradient element by element by calculating the variance and deviation of the gradient update, and to limit the sensitivity of the gradient clipping according to the mean and statistical characteristics of each layer of the neural network to be bounded and retain the effective gradient information as much as possible. After that, we use the "Moments Accountant" mechanism to analyze the cumulative privacy budget generated by noise addition and give more accurate privacy bounds.

3. Since local differential privacy is not an effective defense against generative adversarial network attacks against federation learning, and in federation learning models with a large number of communication rounds, the amount of noise accumulates exponentially due to the strong combinatorial nature of differential privacy, leading to an overall high privacy cost. In this paper, we design a novel federal learning secure mix-and-wash algorithm that uses the scoring principle of the exponential mechanism to select the top k gradient elements in absolute value ranking, add Laplace perturbation, and design the Top-K gradient selection algorithm that satisfies (ϵ, δ) -differential privacy. In addition, a safe mixer is added to the federal learning model to improve the randomness of the data by permuting the matrix composed of gradients and indices. In each communication round, the client sampling rate is dynamically adjusted according to the index decay mechanism to reduce the communication load under the same number of communication rounds. The double privacy amplification effect is achieved by client-side sampling and gradient index scrubbing to reduce the overall privacy loss of the system, improve the communication performance, and demonstrate the privacy and global convergence of the secure scrubbing framework.
4. In order to verify the feasibility of the scheme of this paper in the actual production environment, this paper simulates the federal learning environment and conducts experiments on MNIST, CIFAR-10, and FMNIST datasets respectively. Firstly, we analyze the influence of each parameter on the model accuracy and

communication performance by the control variable method, and compare it with the previous differential privacy scheme and secure mashup scheme, and demonstrate through the experimental results that The adaptive local differential privacy algorithm and the secure mashup algorithm are demonstrated to minimize the accuracy loss of the model and reduce the communication cost while protecting data privacy. Finally, this paper simulates membership inference attacks and generative adversarial network attacks to evaluate the privacy-preserving utility of the adaptive differential privacy scheme and the secure shuffling scheme against the attack model.

Keywords: *Federated learning, Privacy preserving, Differential privacy , Secure shuffle*

目录

第一章 绪 论	1
1.1 研究背景	1
1.2 安全性和隐私威胁	2
1.3 国内外研究现状	5
1.4 研究内容与论文结构	8
1.4.1 研究内容	8
1.4.2 论文结构	10
1.5 本章小结	11
第二章 基础知识	12
2.1 神经网络	12
2.1.1 基本结构	12
2.1.2 随机梯度下降算法	14
2.1.3 经验风险最小化	14
2.2 联邦学习	15
2.3 差分隐私	16
2.3.1 基本定义	16
2.3.2 实现机制	18
2.3.3 相关定理	19
2.4 联邦学习中的差分隐私	20
2.5 本章小结	21

第三章	本地自适应差分隐私算法	22
3.1	引言	22
3.2	模型设计	24
3.2.1	梯度的自适应加躁算法	25
3.2.2	梯度的自适应裁剪算法	29
3.2.3	基于自适应差分隐私的 SGD 算法	32
3.3	隐私参数分析	34
3.4	实验评估	36
3.4.1	实验准备	36
3.4.2	实验设计	38
3.4.3	结果分析	40
3.5	本章总结	47
第四章	基于 Top-K 混洗差分隐私的联邦学习模型	49
4.1	引言	49
4.2	模型设计	51
4.2.1	模型概览	52
4.2.2	Top-K 梯度选择算法	54
4.2.3	客户端动态采样	58
4.2.4	梯度混洗算法	59
4.3	隐私性和收敛性证明	60
4.3.1	隐私性证明	60
4.3.2	模型收敛性分析	63
4.4	实验评估	64
4.4.1	实验准备	64
4.4.2	实验设计	65
4.4.3	结果分析	66
4.5	本章总结	72

第五章 总结与展望	73
5.1 论文总结	73
5.2 论文展望	76
参考文献	77

插图

1.1	联邦学习训练模型概览	3
1.2	针对联邦学习的隐私攻击	4
2.1	神经网络结构图	13
2.2	联邦学习模型工作流程	16
2.3	差分隐私的相邻数据集示意图	16
3.1	自适应差分隐私 SGD 算法流程图	25
3.2	逐层关联传播算法：根据前向传播计算总归因分数	27
3.3	逐层关联传播算法：根据反向传播计算各神经元的归因分数	28
3.4	MNIST 手写数字数据集	37
3.5	模型网络结构	37
3.6	仿真联邦系统模型概览	38
3.7	实现本地自适应差分隐私的伪代码片段	39
3.8	在 MNIST 数据集上噪声大小，裁剪阈值和隐藏层数量这三个参数对于训练准确率的影响	40
3.9	在 MNIST 数据集上不同隐私预算下训练的准确率	42
3.10	不同隐私保护方案在 MNIST 数据集上训练的测试误差变化情况	44
3.11	成员推理攻击过程	46
3.12	在不同模型上进行成员推理攻击的准确率	47
4.1	基于 top-K 安全混洗的联邦学习框架	52
4.2	top-K 梯度选择算法流程图	56

4.3	梯度元素的值及其效用评分	56
4.4	安全混洗模型中本地客户端数量对联邦学习模型训练精度的影响	67
4.5	安全混洗模型中客户端采样比对联邦学习模型训练精度的影响	68
4.6	安全混洗模型中梯度选择比率对联邦学习模型训练精度的影响	68
4.7	不同隐私保护方案在 MNIST 数据集上训练的模型分类准确率变化 情况	70
4.8	联邦学习下的 GAN 模型	71
4.9	在 MNIST 数据集上对不同的隐私保护联邦学习算法下进行 GAN 攻 击所生成的伪样本	72

表格

3.1	本地自适应差分隐私与其他四种基准方案	40
3.2	对比实验在数据集 MNIST 和 CIFAR-10 上的参数设置	43
3.3	本地自适应差分隐私与其他四种基准方案在 100 个训练轮次后模型所能达到的准确率	45
4.1	安全混洗框架实验的模型网络结构	65
4.2	安全混洗框架的比较方案	66
4.3	不同梯度选择比率的通信开销	69

List of Algorithms

1	随机梯度下降算法	14
2	梯度的自适应加躁算法	30
3	梯度的自适应裁剪算法	33
4	基于自适应差分隐私的随机梯度下降算法	34
5	联邦学习中的安全混淆算法： \mathcal{A}_{ssdp}	53
6	Top-K 梯度选择算法	57
7	客户端动态采样算法	59
8	混淆器中的拆分混淆算法	60

第一章 緒論

1.1 研究背景

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，广泛地应用于各种领域。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，深度学习凭借其高效的数据建模、抽象和泛化能力大幅提升了模型的预测准确率。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，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，用于识别可疑恐怖分子和检测违禁物品；基于深度学习的回归技术还可以帮助诊断和预防某些疾病；基于卷积神经网络实现无人驾驶车辆系统的目检测和自动决策等。

当前深度学习的商业应用模式可以概括为：各个信息机构通过其提供的服务平台从用户处收集数据，训练算法模型以提升服务质量，从而获得更多的用户量和数据量。用户的搜索记录、浏览历史、购买交易、观看的视频都有可能被各个商业机构收集，用作模型训练的数据集。在 2018 年，中国互联网协会收到用户举报，发现腾讯等多家应用软件以“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。人们开始担心自己的数据被收集后会被泄露或者是被不正当使用。2018 年欧盟也正式颁布实施了《通用数据保护法案》，旨在保护用户的个人隐私和数据安全。

深度学习提供的服务以大数据算法为基础，然而多个数据源之间也存在着难以打破的壁垒。在大多数行业中，数据是以孤立的岛屿形式存在的。例如，某机构

基于深度学习的算法提供商品推荐的服务，它拥有用户的基本信息数据，却缺失了关于用户消费水平和购买偏好的数据，没有用户购物相关的特征难以训练出精准的推荐数据。除了一些巨头公司，绝大多数的企业存在数据质量差、数据量少的问题，使得他们难以提供优质的人工智能服务。如何在符合法律法规的前提下，采用跨组织的数据进行模型训练，并保护用户的隐私和数据安全是一大难题。

针对数据孤岛问题，Google 在 2016 年提出了联邦学习的框架，它是一种有助于解决多方计算下的数据孤岛问题的学习方法。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，所有训练数据保存在本地设备，不同的参与方按照各自的需求在本地训练模型更新权重。中央服务器接收所有本地设备上传的模型权重，训练一个全局的虚拟模型，通过将各方数据以共享梯度的方式进行聚合，更新全局参数。然后本地设备再从中央服务器下载全局参数，迭代地进行更新，使模型的训练结果最优化。联邦学习本质上是深度学习和分布式计算的结合，将模型训练与在云中存储数据的需求相分离，在合法合规的基础上，使所有本地设备可以在不共享训练数据的情况下联合建模。与集中式深度学习相比，联邦学习系统通过分布式的多方协作学习破解了数据孤岛的壁垒，实现更智能的模型、更低的延迟和更少的功耗，在学术界和工业界受到广泛关注。

1.2 安全性和隐私威胁

尽管联邦学习解决了部分数据孤岛的问题，其本身还存在很多脆弱性和薄弱点。近年来，大量研究表明联邦学习机制仍然存在许多安全漏洞，这些漏洞可能被内部参与者和外部攻击者所利用，破坏联邦学习系统的安全性。由于联邦学习的框架并没有对参与方的资质进行校验、没有对模型的访问权加以约束，恶意的参与方可能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偏离，甚至全局模型不可用。此外，联邦学习也并没有考虑到对传递的参数进行保护，本地设备与中央服务器之间的通信信道有可能成为第三方窃取敏感信息的途径。本地设备上传到中央服务器的梯度本质是由模型和

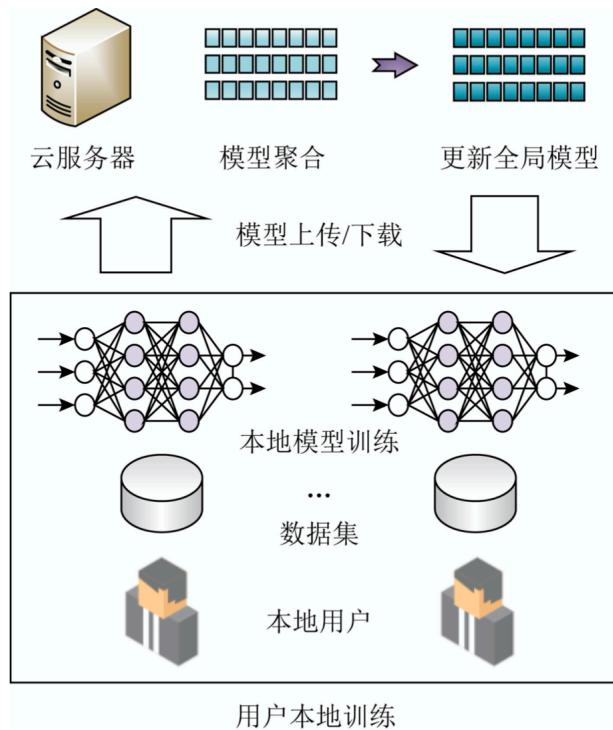


图 1.1: 联邦学习训练模型概览

训练数据计算得到的函数，其中包含了关于本地训练数据的信息。攻击者可以从共享梯度中跟踪和获取参与者的隐私。综上，联邦学习的框架仍然存在本地训练数据泄漏、全局模型不可用等隐私问题。

图1.2大致概括了针对联邦学习隐私攻击的攻击者、攻击内容、攻击类型、攻击方式和攻击发生的时段。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。有一些恶意参与者作为本地客户端参与训练，修改本地的训练数据，注入一些有毒的数据，从而损害全局模型的准确性，操纵模型的预测结果；诚实但好奇的中央服务器通过观察本地客户端上传的梯度更新，篡改训练过程，并控制参与者对全局参数的视图。外部攻击者通过本地客户端与中央服务器之间的通信信道发起攻击，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。内部攻击通常比外部攻击更强，因为敌手拥有关于模型架构和内部参数的信息。

针对联邦学习的攻击方式包括投毒攻击、模型反演攻击、成员推理攻击和生成

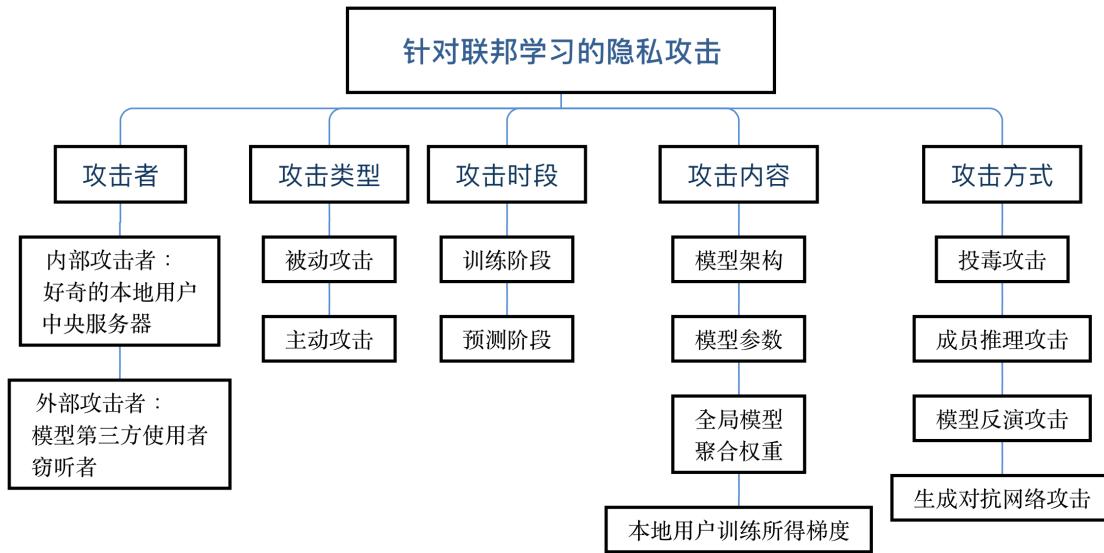


图 1.2: 针对联邦学习的隐私攻击

对抗网络攻击等。

投毒攻击：投毒攻击通常发生在联邦学习的训练阶段。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^[56] 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。

模型反演攻击：当攻击者只能黑盒访问联邦模型时，攻击者仍然可以通过联邦学习系统提供的应用程序接口 (Application program interface, API) 访问系统模型，通过向系统模型发送训练数据，获得所有表示类的预测标签和置信度信息，根据类标签和置信度数据重新建模，还原出目标模型的训练数据^[11]。即使攻击者不清楚联邦模型的内部信息（训练参数、训练数据、模型结构等），依然可以通过 API 逆向反演获得所有预测数据的置信数据，由于置信信息代表了特征向量和标签类的关联性，攻击者将特征向量做为输入信息，对某一标签类进行分类或者回归得到

置信度信息，其中置信度最高的类即为目标模型的目标类之一。

成员推理攻击：给定一个深度学习模型和一条数据样本（攻击者可以获取的知识范围），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。例如，一个医疗机构根据基因型数据提供疾病预测的服务。假设保险公司持有某个客户 A 的基因型数据，通过对该疾病预测服务进行成员推理攻击，推断出 A 是否为某种疾病标签的成员样本，该公司就可以收取更多的费用以谋利。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，通过影子学习的技术构建与目标模型的训练数据相近的数据集^[13]。针对每一种数据样本，攻击者随机初始化一些数据记录作为影子模型的训练集数据。然后将此数据记录喂给目标模型得到预测向量，将所得的预测向量以及样本标签喂入攻击模型得到二分类的结果，达到推断任意样本是否在训练集中的目的。

生成对抗网络攻击：攻击者伪装成正常的用户参与联邦学习的训练，通过生成对抗网络（Generative Adversarial Networks, GAN）模型获得其他本地用户的训练数据。首先，攻击者在本地训练一个生成器网络和判别器网络，两者以零和博弈的思想进行对立训练，形成对抗生成式深度学习网络。生成器用来生成目标模型的伪样本，判别器被训练来区分图像是来自原始数据库还是由 GAN 生成的。攻击者将生成器生成的样本标记为“fake”类上传至全局模型，中央服务器通过聚合所有参与者上传的数据得到全局参数，攻击者通过基于假数据集的正常正向和反向计算获得假梯度，通过最小化假梯度和真实梯度之间的损失，反复优化假样本和标签，获得目标类的隐私数据。由于判别器参与了全局模型的训练，相当于在拥有该样本的用户训练数据上训练判别器，使得生成器有能力构造出与真实样本相似的伪样本。

1.3 国内外研究现状

随着针对联邦学习隐私攻击的模型增多，越来越多的研究者开始关注如何在联邦学习中应用隐私保护机制以防御上述的攻击模型。安全多方计算、同态加密和扰动技术是最常见的提高联邦学习中的安全性和隐私性的技术。

安全多方计算（Secure Multi-Party Computation, SMC）是由姚期智在 1982 年提出的^[16]，多个参与者在不泄露各自隐私数据情况下共同完成某项计算任务。安全多方计算是解决多方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。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。安全多方计算保护联邦学习的重点是如何在没有可信第三方的情况下安全地计算联邦学习中的经验风险最小化函数。同时，每个参与者都不可以获得任何关于其他用户的信息，除了梯度的聚合结果。Bonawitz 等人利用 SMC 设计了一个联邦学习隐私保护框架，通过秘密分享技术安全地汇总用户的梯度，服务器只能看到聚合完成之后的梯度^[3]，不能知道每个用户的私有的真实梯度值，并且对用户的退出具有鲁棒性。然而，他们的模型由于涉及到学习过程中的多轮计算交互，在联邦学习通信回合数很大的情况下会导致通信性能大幅下降。

同态加密（Homomorphic Encryption, HE）是一种加密形式，允许第三方直接对密文进行算术运算^[10]。在密文上进行计算后得到的结果进行解密，与明文上进行相同计算操作得到的结果是一致的。同态加密分为加法同态加密、乘法同态加密和全同态加密。同态加密保护联邦学习的隐私性主要通过将本地用户训练所得的梯度信息进行加密后上传至中央服务器，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，加密场景的高计算复杂度会严重降低联邦学习的计算性能。此外，由于需要传输公钥和秘钥也会增加额外的通信成本。Phong 等人^[23] 提出了一个基于加法同态加密的联邦学习隐私保护方案，中央服务器可以根据同态操作，用加密的本地梯度更新全局模型参数。然而在基于联邦学习的系统中，有许多分布式设备（如智能手机和物联网设备）参与其中，用于解密的同一私钥需要分布到各个客户端，一旦持有相同秘钥的用户相互串通，将无法保证用户的数据隐私。

扰动技术的关键思想是在原始数据中加入噪声，使加躁后的数据与原始数据在统计特征上无法区分。三种广泛使用的扰动技术包括差分隐私、加法性扰动和乘法性扰动。差分隐私 (Differential Privacy, DP) 技术是基于概率统计模型来量化数据集实例的隐私信息被披露的程度^[73]，其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^[14]，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。差分隐私的优势在于其保护数据集所需要的隐私成本可以被量化，在通信效率和计算性能方面与明文计算相差不大。许多公司已经广泛采用差分隐私保护其联邦学习模型，包括谷歌、苹果、Uber、微软和 LinkedIn。

在联邦学习框架应用差分隐私实现隐私保护基本可以分为两种方式：在本地模型训练阶段部署差分隐私；在全局参数聚合阶段部署差分隐私。

联邦学习中的本地用户进行模型训练的过程都可以看作是一次深度学习的过程。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。Song 等人提出了一个 $(\epsilon_c + \epsilon_d)$ -差分隐私版本的随机梯度下降算法 (DP-SGD)，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^[47]。然而，DP-SGD 与 SGD 相比严重降低了模型的准确率。当差分隐私提供的隐私保护强度增加时，在 MNIST 数据集上进行逻辑回归的训练和验证的损失率迅速增加。

Shokri 等人^[45] 提出了一种选择性参数共享的差分隐私保护方案，通过对用户本地梯度绝对值进行排序，选择绝对值排名较大的梯度元素添加拉普拉斯噪声后参与联邦平均，然而随着模型迭代次数的增加，差分隐私的应用会导致模型的可用性下降。Choudhury 等人^[77] 通过在联邦学习框架中应用分布式差分隐私机制，在两个真实的世界健康数据集上分析了不同的隐私预算对联邦模型性能的影响。他们给出的实验结果说明虽然差分隐私提供了一个强大的隐私水平，但是随着模型的通信回合数上升，整体隐私成本增加，模型性能大幅下降。

Geyer 等人^[48] 研究发现与集中式学习相比，联邦学习中的梯度在整个训练过程中对噪声和批量大小具有不同的敏感度。他们的方案通过隐藏客户端的参与信息，

实现客户层面的差分隐私，在分布式训练的过程中根据各个用户的隐私设置动态调整差分隐私的隐私参数，降低全局模型的性能损失。Truex 等人^[49] 将差分隐私和同态加密技术相结合，在每个本地设备训练所得的权重上添加噪声后再使用同态加密技术进行加密，发送给中央服务器。中央服务器根据运算的同态性对加密后的数据进行聚合，更新全局参数。虽然通过差分隐私和同态加密加强了本地数据的隐私性，但是计算成本较高。

然而，Kang Wei 等人^[78] 表示，Geyer^[48] 和 Truex^[49] 的工作没有考虑到本地参数上传阶段的隐私保护，在向服务器上传训练结果时，客户的私人信息有可能被隐藏的敌手所截获。此外，这两篇文章缺乏对隐私性、收敛性能的具体分析。Kang Wei 等人提出了提出了一个基于全局 (ϵ, δ) -DP 概念的新框架，对本地参数上传通道和全局参数下载通道定义不同的差分隐私要求，并在本地客户端和中央服务器采用不同的隐私参数添加高斯噪声。通过分析联邦学习模型损失函数的收敛界限，得出了以下结论：(1) 模型的隐私保护水平越高，收敛性能越差；(2) 在相同的隐私保护水平下，增加客户端的数量可以提高收敛性能；

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，而现有的差分隐私保护联邦学习的方案很难在实现强隐私保护的同时，维持原有模型的精度。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；当隐私保护水平太低时，无法防御生成对抗网络攻击。当前的联邦学习中的隐私保护方案还有许多不足，很难在模型效用、数据隐私保护、通信性能这三个方面都达到满意的效果。

1.4 研究内容与论文结构

1.4.1 研究内容

联邦深度学习通过分布式的协作学习使得各个参与方在无需传递和共享本地的数据资源的情况下训练一个共同的、强大的深度学习模型。与传统的集中式深度学习相比，联邦学习在一定程度上缓解了数据孤岛和隐私泄露的问题。然而许多

研究表明联邦学习机制仍然存在许多安全漏洞。联邦参与方、好奇的中央服务器以及外部的恶意敌手都有可能通过模型权重、共享的梯度和通信信道对联邦模型发起攻击，窃取用户的本地训练数据，破坏模型的可用性。差分隐私是当前保护联邦学习隐私安全的前沿技术，其通过严格的统计框架提供隐私保证和隐私成本计算，使得加噪后的梯度不能泄露关于实体数据的敏感信息。与安全多方计算和同态加密等密码学技术相比，差分隐私保护联邦学习的方案在通信性能和计算性能方面与明文计算相差不大。本文主要针对联邦学习的成员推理攻击和生成对抗网络攻击，设计基于差分隐私和安全混洗的隐私保护方案，对共享的梯度和模型权重进行隐私保护。本文的具体研究内容如下：

- (1) 在联邦学习的本地模型训练过程中，在每一轮随机梯度下降算法中对梯度注入高斯噪声，以防御成员推理攻击。现有的差分隐私保护深度学习方案大多数是基于固定的模型参数和隐私设置，很难在模型可用性和数据隐私性之间达到平衡效果。为了更好的解决现有的差分隐私保护数据隐私的同时，模型可用性下降的问题，本文对梯度加噪和梯度裁剪算法进行了以下优化：第一，根据逐层关联传播算法，在神经网络的前向和反向传播过程中分解神经元对于模型输出的贡献率，根据贡献率分配相应的隐私预算。保持整体的隐私预算不变，对于模型输出影响更大的梯度上添加较少的噪声，以减少梯度加噪对于模型可用性的影响。第二，在随机梯度下降算法的迭代过程中，根据之前训练所得梯度的统计特征，动态调整梯度裁剪阈值，尽可能的保留梯度中的有效信息。第三，本文采用“*Moments Accountant*”机制分析加噪累积产生的隐私预算，使得隐私损失的计算更加精确。
- (2) 当联邦学习中的用户数量达到千万量级时，在本地设备的模型训练上采用差分隐私技术，对于聚合后的梯度平均估计误差能达到 $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{m}}\right)$ 。如果在迭代训练过程中的每一次迭代都应用本地差分隐私，隐私损耗就会成倍累积，从而导致聚合参数上的噪声溢出，影响全局模型的发布结果，增加了通信成本。为了解决这一问题，本文对基于本地差分隐私的联邦学习隐私保护方案

进行了以下优化：第一，基于指数机制的打分函数和稀疏向量的思想对梯度进行采样扰动，相比本地差分隐私而言降低了计算复杂度；第二，在联邦学习中动态的调整客户端采样率，使用指数衰减率来递减训练过程中的采样率，在相同的通信回合下降低整体的通信成本；第三，在中央服务器和本地设备之间引入混洗器。混洗器将所有客户上传的加密数据集合中的向量元素进行拆分和随机置换，得到一个无序的消息集合。混洗和采样的操作达到了双重的隐私放大效应，从 $(\epsilon_c + \epsilon_l)$ 的本地差分隐私放大至 $\bar{\epsilon}$ -中央差分隐私。在本地设备添加更少的噪声，而在中央服务器上达到相同的隐私保护效果，兼顾隐私保护能力与模型可用性。

1.4.2 论文结构

本文一共五个章节，主要内容的组织安排如图所示：

第一章介绍了联邦学习的研究背景和存在的隐私威胁，并具体阐述了针对联邦学习的差分隐私保护的研究现状与发展方向，最后介绍了本文的相关工作和贡献。

第二章详细介绍了关于本文研究内容的基础知识，包括联邦学习的工作流程，差分隐私的基本概念和定理、神经网络的基本结构和训练算法。

第三章是基于本地自适应差分隐私保护联邦学习的共享梯度。首先，在引言部分介绍了差分隐私保护深度学习算法的四种扰动方式，主要针对梯度扰动分析了前人方案的不足之处，接着提出了本文所设计方案的两个创新点，依次详细的描述了梯度的自适应加躁算法和梯度的自适应裁剪算法的设计思路和实现流程。将梯度自适应的加躁和裁剪算法应用在随机梯度下降算法中，得到本地模型训练的核心算法，并结合 MA 机制分析总隐私损失。最后，我们通过三方面的实验对本地自适应差分隐私保护方案进行了全面的评估：其一，分析噪声水平、裁剪阈值、隐藏层数量这些参数对模型分类准确率影响；其二，与非隐私的 SGD、前人提出的差分隐私 SGD 方案，比较各个方案在相同隐私预算的情况下模型分类所能达到的准确率和模型收敛速度；其三，针对部署了本地自适应差分隐私的联邦学习模型

训练成员推理攻击模型，评估隐私保护效用。

第四章针对高维聚合场景下噪声成倍累积的问题，设计了 Top-K 梯度安全混洗算法。首先，定义了威胁模型和隐私表述，描绘了基于安全混洗的联邦学习模型框架，依次详细的描述了本地 top-K 梯度采样算法、客户端动态采样算法和梯度混洗算法的设计思路和实现流程。接着，证明了采样和混洗算法实现了 $(\epsilon_c + \epsilon_l)$ -本地差分隐私到 $\bar{\epsilon}$ -中央差分隐私的隐私放大效用。最后，我们通过三方面的实验对 Top-K 梯度安全混洗算法进行了全面的评估：其一，分析客户端数量 N 、梯度选择的比率、客户端采样比 f_r 和最大聚合次数对于模型分类准确率的影响；其二，与非隐私的 SGD、前人提出的差分隐私 SGD 方案进行对比实验，评估指标为模型分类准确率和通信性能；其三，针对部署了 Top-K 梯度安全混洗算法的联邦学习模型训练生成对抗网络攻击模型，评估隐私保护效用。

第五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。

1.5 本章小结

这一章节为绪论，首先介绍了联邦学习的研究背景和存在的隐私威胁，并具体阐述了针对联邦学习的差分隐私保护的研究现状与发展方向，最后介绍了本文的相关工作、贡献和论文的组织结构。

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。当给出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，是通过测试误差（泛化误差）衡量的。模型的泛化能力取决于训练数据的质量和数量、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。

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^[34]最初的设计灵感来源于人脑的结构。众所周知，人类的大脑是处理信息的重要部分。人脑中含有大量的神经元，当人脑接受到外部环境的刺激时，信号随着神经元一层一层的传入人脑神经中枢，神经中枢根据接受到的信号给出判断，然后再随着输出

神经网络传递，最终作出不同的行为或者判断。神经网络就是模拟人脑处理信息的流程对数据进行学习的。

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。第1层的神经元与第1-1层的所有神经元相连，从这些神经元接收数据，这就是全连接的概念。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。

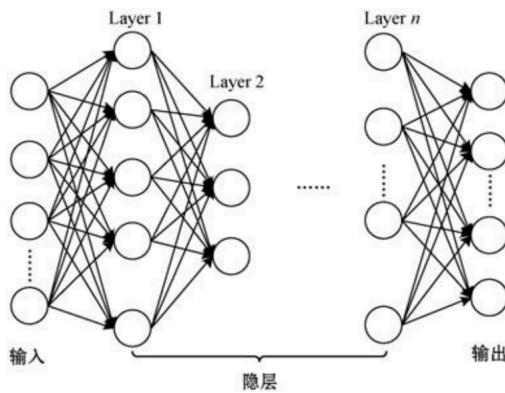


图 2.1: 神经网络结构图

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特定的反应；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。通过类似于人脑处理信息的方式，神经网络在重复训练的过程中调整网络权重，使最终输出的预测结果与真实结果更加接近，但是如何调整网络权重使误差最小呢？

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（如交叉熵损失函数、均方误差损失函数等）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，从数据集中随机采样一批训练样本，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 α
- 2: **输出:** 初始参数 θ
- 3: 初始化模型权重 θ ，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（Em-

pirical 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2.1所计算得到的经验风险值最小。

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:** 所有用户在个字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: $\omega^{r+1} \leftarrow \omega^r - \eta \frac{\sum_{U_i \in U^t} \omega_i^r}{\sum_{U_i \in U^t} |D_i^t|}$
- **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。

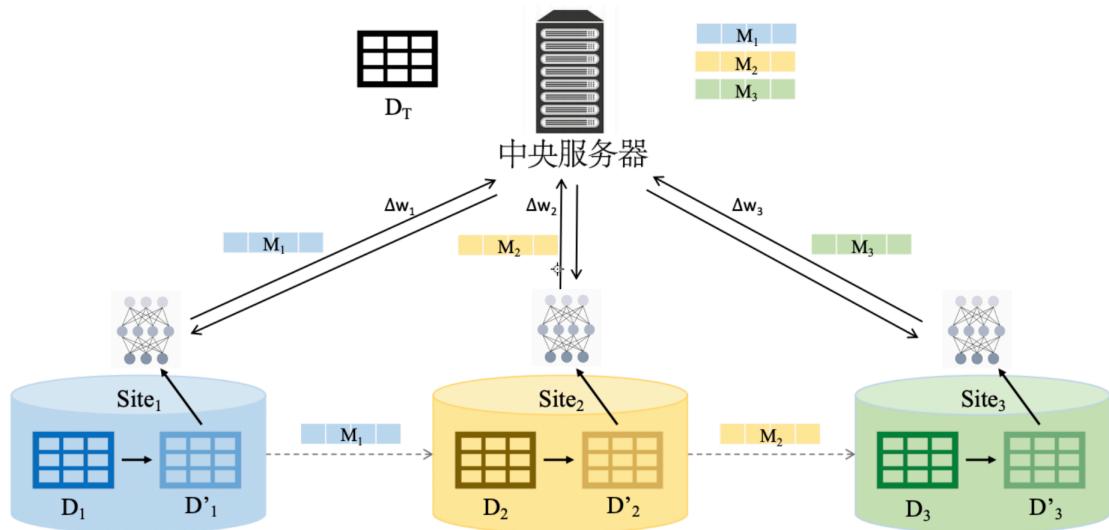


图 2.2: 联邦学习模型工作流程

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*)。

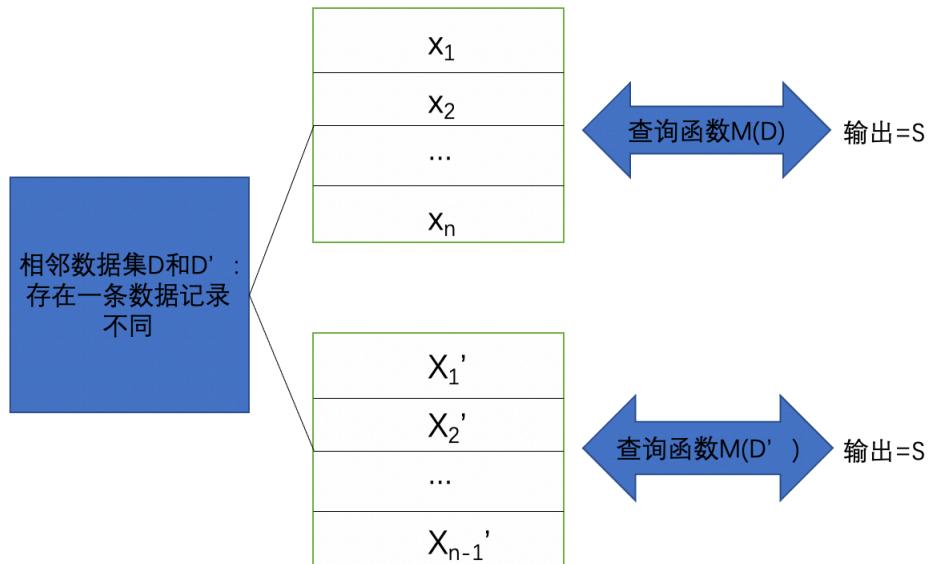


图 2.3: 差分隐私的相邻数据集示意图

2016 年, Dwork^[14] 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:

定义 2.3.2 ((ϵ, δ) -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

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

, 则称该随机算法满足 (ϵ, δ) -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 ϵ -差分隐私。 ϵ 和 δ 表示隐私预算参数, ϵ 和 δ 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, 对于相邻数据集 D 和 D' , 某个查询函数在此相邻数据集上所输出的结果的不同程度代表了此函数的敏感度, 而函数的敏感度决定了需要在函数中添加的噪声量来实现最终输出结果的差分隐私, 因此加入的噪声量与函数的敏感度高度相关的。

定义 2.3.3 (函数敏感度). 假设存在查询函数 $f : D \rightarrow R^d$, 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' , 函数 f 的 L_1 敏感度 (L_2 敏感度) 表示为 $\Delta_1(q)$ ($\Delta_2(q)$), 计算公式如下:

$$\Delta_1(q) = \max_{D \sim D'} \|f(D) - f(D')\|_1, \quad \Delta_2(q) = \max_{D \sim D'} \|f(D) - f(D')\|_2$$

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)^[9]、指数机制 (Exponential Mechanism)^[32] 与高斯机制 (Gaussian Mechanism)^[33]。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^[35]。

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 ϵ -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, 高斯机制通过对数据中的所有维度添加满足高斯分布的噪声实现差分隐私。

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 ϵ -差分隐私, 当:

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

其中, $\mathcal{N}\left(0, S_f^2 \cdot \sigma^2\right)$ 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 (ϵ, δ) -差分隐私。其中 ϵ 代表了隐私保护的程度, 与噪声量呈负相关; δ 是松弛项, 表示允许以多少的概率打破差分隐私。

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:

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

其中 $u(D, p)$ 为评分函数, 评分越高, 则输出的概率越大^[53], Δu 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: 组合性、可量化性和后处理不变性^[35]。组合性指的是将多个差分隐私的算法进行串行组合或者并行组合后得到的算法整体依然满足差分隐私。

定理 2.3.7. 对于任意满足 (ε, δ) -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 (ε, δ) -差分隐私。

定理 2.3.8. 对于任意满足 (ε, δ) -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。

可量化性指的是在算法中添加随机扰动以满足差分隐私时, 可以精准的计算所添加的噪声量, 代表了整体算法满足差分隐私所需要的隐私预算, 这里的隐私预算是可量化的。后处理一致性则是指在对满足差分隐私的算法进行进一步处理时, 只要不引入额外的信息, 其输出依然满足差分隐私, 整体的隐私保护力度也是不变的。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) 将敏感度有界的函数相组合使得整个系统的查询函数敏感度有界
- (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 联邦学习中的差分隐私

现有的在联邦学习模型中应用差分隐私实现隐私保护主要有两种机制：中央差分隐私和本地差分隐私。在中央差分隐私机制中，它要求一个可信的第三方服务器对所有用户的数据分析结果进行隐私化处理。在联邦学习的场景中，实现中央差分隐私需要一个可信的第三方服务器，如果中央服务器是可信的，通过定义全局函数的敏感度，在查询结果上添加噪声，使得全局聚合函数的输出结果是不可区分的；如果中央服务器是诚实但好奇的，在本地客户端和中央服务器的通信信道中间新增一个可信第三方服务器对所有用户的训练结果添加拉普拉斯扰动实现差分隐私。与中央差分隐私机制不同，本地差分隐私机制没有对第三方服务器的任何假设，所有用户的训练结果在本地设备对数据添加随机扰动后上传至中央服务器，它可以为每个用户提供强大的隐私保证。然而，由于每个本地用户都必须在自己的数据中添加高斯噪声，所以模型总体的噪声相比中央差分隐私要大得多，导致统计结果可用性差。

在联邦学习环境中，用户面对的可能是大量的非可信实体，而一个真正可信的数据机构又很难找到。本文主要应用本地差分隐私机制实现联邦学习的隐私保护，同时引入其他技术使整体的噪声量降低，降低全局模型的准确率损失。接下来介绍在联邦学习中应用本地差分隐私的基本流程：

- **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。

2.5 本章小结

本章节介绍了论文研究内容所需了解的基础知识, 包括差分隐私、神经网络和联邦学习。本节还介绍了联邦学习的基本工作流程, 本文所提出的隐私保护方案都是针对联邦学习中的隐私泄露问题。此外, 本节介绍了神经网络中前向传播和反向传播的算法以及具体实现方法——随机梯度下降算法, 第三章所提出的本地自适应梯度加噪方案就是基于神经网络的结构进行设计; 差分隐私是本文所重点关注的实现隐私保护的机制, 论文重点讲解了差分隐私的定义、实现机制和相关定理。最后, 论文介绍了传统的在联邦学习中实现差分隐私的方式, 包括本地差分隐私和中央差分隐私。

第三章 本地自适应差分隐私算法

3.1 引言

与传统的集中式深度学习相比，联邦深度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，许多研究表明深度学习技术可以“记忆”模型中的训练数据信息，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，联邦深度学习的框架仍然存在本地训练数据泄漏等隐私威胁^[48]。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。

我们认为中央参数服务器是一个“诚实但好奇”（Honest but Curious, HbC）的实体。也就是说，服务器将遵循与所有用户的协议。然而，它也试图在训练过程通过通信信道访问用户梯度，反推出关于客户端的训练数据的额外信息。出于这个原因，我们设计的本地自适应加噪算法目的是保护发送到服务器的本地梯度不会被中央服务器推断出任何关于用户的本地训练样本信息，并且尽可能维持原有模型的精度。

由于本地客户端上传至中央服务器的参数可能被敌手作为先验信息来进行成员推理攻击，为了保护本地客户端的训练数据的隐私性，我们考虑在用户的本地模型训练阶段实现隐私保护的方案。从隐私保护的角度讲，我们只要截断了本地训练的原始输入到输出，在其中加入一道隐私保护屏障即可使最终的结果满足差分隐私。根据在哪一步截断将差分隐私保护联邦深度学习的方法分为以下几种：

- **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^{[37][38][39]}。

- **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 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^[36]。
- **梯度扰动：**梯度是通过损失函数对网络模型参数进行计算得到的，其中包含了数据集的信息；在梯度上添加噪声，也能保证整体训练过程满足差分隐私。
- **目标扰动：**目标扰动是在模型的目标函数中添加随机扰动，使得最终输出的结果满足差分隐私。

基于输入的扰动和输出的扰动基本可以视为一个黑盒模型，这种添加扰动的方式简单直接，但无法对模型内部数据的相关性作出细致有效的分析，在获取的训练数据和输出的训练参数上添加过多的噪声可能会影响后续模型的收敛，破坏模型的可用性。

当前在联邦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，使输出的结果满足差分隐私并维持模型的高可用性。Song 等人提出了一个 $(\epsilon_c + \epsilon_d)$ -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^[47]。Goodfellow 提出了 ℓ_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^[64]，在预训练过程中，该方法与 PCA 相结合，形成了一个满足 $(\epsilon_c + \epsilon_d)$ -差分隐私的 PCA。由 Song 等人中的实验数据可知，差分隐私随机梯度下降(DP-SGD)与 SGD 相比严重降低了训练模型的效用。当差分隐私提供的隐私强度增加时，在 MNIST 数据集上进行逻辑回归的训练和验证的损失率迅速增加^[47]。在 MNIST 上数据集上，采用 DP-SGD 训练的卷积神经网络(CNN) 的测试精度比 SGD 低得多。

在神经网络的模型训练中，不同阶段的噪声添加对于模型的准确率和收敛速度有不同的影响。在训练的初始阶段，模型的权重和最优权重的距离还很远，梯度

较大，此时添加噪声对于模型的输出影响较小。随着训练的多次迭代，模型的权重越来越接近最优权重，梯度也越来越小，梯度的敏感度变高，在梯度上添加噪声对于模型输出的影响变大。之前的方案是在训练的每个阶段，在梯度上添加同样大小的噪声，会造成隐私预算的不必要的损失和模型精度的降低。本文主要针对这一问题，对梯度加噪的联邦学习隐私保护方案进行改进，创新点主要有两个方面：

- (1) 通过逐层关联传播算法计算特征对于模型输出的贡献率，根据贡献率动态地分配隐私预算，在梯度上添加自适应的高斯噪声。
- (2) 根据训练轮数和梯度变化的偏差与方差动态地更新梯度裁剪阈值，对梯度进行自适应裁剪。

3.2 模型设计

本地自适应差分隐私算法主要分为两个环节，预训练和正式训练。在预训练阶段，我们采用逐层关联传播算法在神经网络的前向传播和反向传播流程中分解模型输出，计算网络中每个神经元对模型输出的贡献率。在神经网络的正式训练阶段，通过随机梯度下降算法计算损失函数对于每个神经元的偏置和权重的梯度，基于每个神经元的贡献率给梯度分配不同的隐私预算，在梯度上添加高斯噪声。根据神经元的贡献率对梯度自适应加噪，避免了无法量化的噪声，从而提升了隐私保护模型的准确性。此外，我们根据梯度更新的统计信息对梯度裁剪阈值进行自适应调整，保证模型的快速收敛。之后我们在 MNIST 和 FICAR-10 数据集上进行实验，评估我们提出的方法，并与前人的四种差分隐私保护方案进行对比，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，而且还降低了隐私预算。并且，我们针对添加本地自适应差分隐私方案的模型进行成员推理攻击，评估了该方案的隐私保护效用。

本文的模型主要针对本地设备的模型训练阶段，设计自适应的梯度加噪和裁剪算法，保证中央服务器接收到满足 (ϵ, δ) -差分隐私的权重，从而减轻成员推理攻击对联邦学习所构成的隐私威胁。本地客户端进行模型训练的基本流程如图3.1所示

示，通过从数据集 D 中随机采样构造各批次的样本，对于每一批样本通过梯度下降算法进行模型训练，通过梯度自适应裁剪算法加快模型的收敛速度，在梯度上添加自适应的高斯噪声，使整体训练系统满足 (ϵ, δ) -差分隐私。

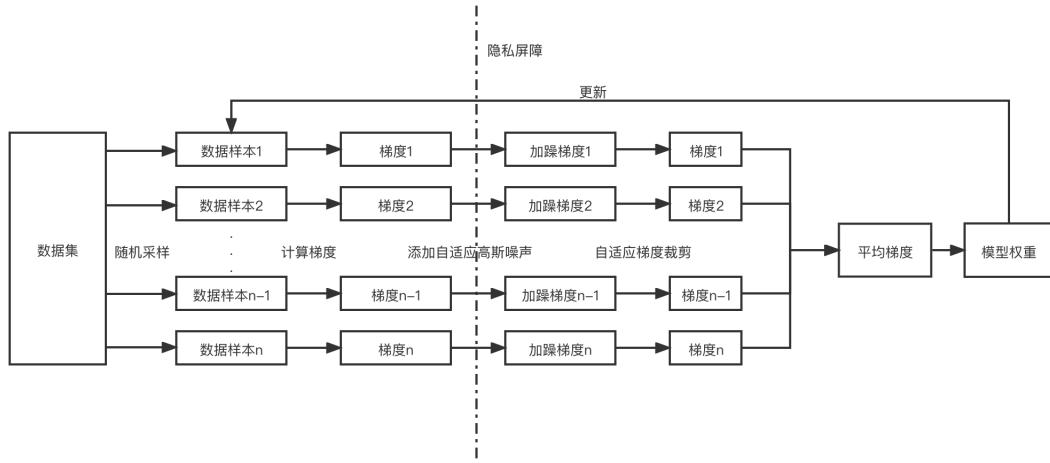


图 3.1: 自适应差分隐私 SGD 算法流程图

3.2.1 梯度的自适应加躁算法

Bach 等人提出了针对神经网络的逐层关联传播算法^[65]，这是一种用于计算特定图像区域对分类器输出结果影响的技术。该技术被认为是非常有效的，可以逐个像素地分解图像的预测值^[74]，可以应用在深度神经网络中对各神经层进行循环重正化^[75]，用来分解深度神经网络的预测值，得到神经网络中各个特征对于模型结果的贡献。经典的反向传播算法是通过链式法则对模型的输出进行反向求导，根据损失函数计算每个神经元对模型总误差的贡献值，然后来调整模型的权重，以降低总误差。根据这一思路，在本文的工作中，创造性地利用逐层关联传播算法将神经网络的输出值按层进行分解，得到每层的神经元对于模型输出的贡献率。为了尽量弥补噪声添加而带来的模型可用性下降的问题，我们根据特征对于模型输出的总体影响，分配隐私预算。在贡献率大的特征上分配较高的隐私预算，添加的噪声量较小；在贡献率较低的特征上分配较低的隐私预算，添加的噪声量较大。

本地用户进行模型训练的主要步骤如下：

- (1) 对模型进行预训练，运行逐层关联传播算法计算每个神经元对于模型输出的归因分数，归因分数代表了神经元对于模型输出的贡献程度；
- (2) 初始化神经网络的参数 θ_0 、模型训练的迭代次数 T 、高斯噪声隐私参数 σ
- (3) 从数据集 D 中随机采样 L 训练样本送入神经网络中，运行前向传播算法，得到模型的预测结果 $f_{X_i}(\theta_t)$
- (4) 通过交叉熵损失函数计算模型的预测值 $f_{X_i}(\theta_t)$ 与真实值 y_i 之间的误差 $\mathcal{L}(\theta_t, X_i) = -\sum_{(X_i, y_i) \in L_i} y_i \log f_{X_i}(\theta_t)$ ，运行反向传播算法，计算损失函数对于所有神经元的权重和偏置的梯度 $\mathbf{g}_t(x_i)$
- (5) 根据步骤 (1) 中计算的归因分数，进行隐私预算的分配，计算每个梯度分配到的隐私预算 $\sigma_i = \frac{\sigma_t}{Cr_j(x_i)}$
- (6) 根据隐私预算在梯度上添加对应的高斯噪声
- (7) 重复步骤 (3) - (7) 直至模型收敛，输出模型权重

神经元归因分数的计算

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 $y = \sigma(\mathbf{x} * \omega + b)$ ，其中 \mathbf{x} 代表输入向量， y 是输出， b 和 ω 分别代表偏置项和权重重矩阵。 $\sigma()$ 是一个激活函数，用于结合线性变换和非线性变换。在前向传播过程中，图像输入以像素的形式进入网络，然后与网络权重 w 相乘，再加上一个偏置 b ，并通过激活函数使其成为非线性计算，这个过程一直持续到输出层得到模型输出。

对于卷积神经网络等网络结构，模型输出表示为 F_θ ，输出层的神经元表示为 z 。在逐层关联传播算法中，根据神经网络的结构自前向后计算总归因分数 Cr_z 。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。运行前向传播算法获得总归因分数后利用 F_θ 和 Cr_z 在反向传播算法中逐层分解，计算各层神经元的归因分数，它代表了每个神经元对于模型输出的贡献程度。

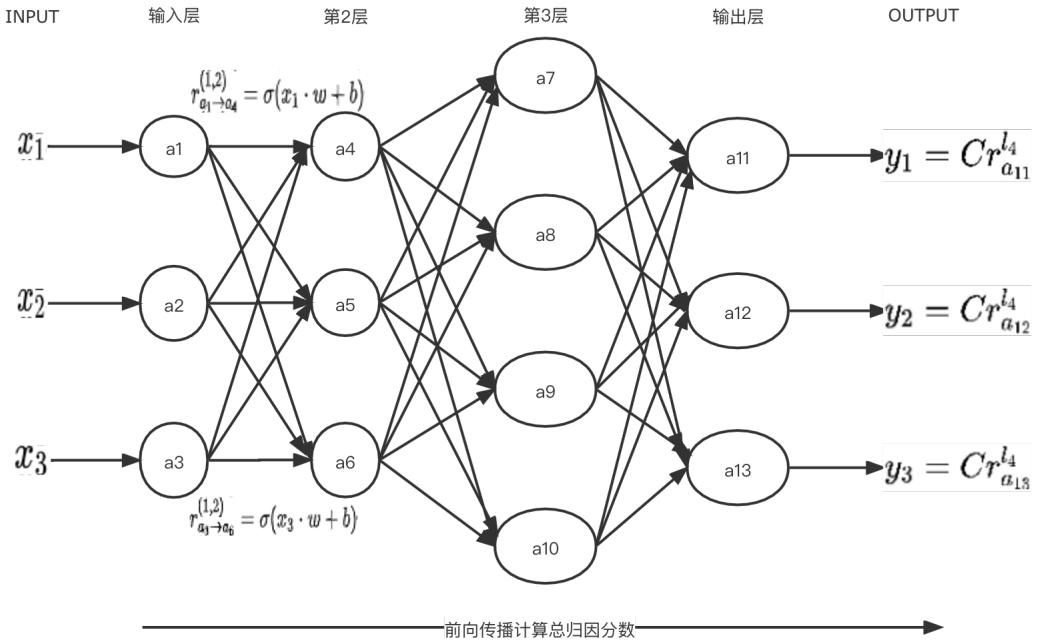


图 3.2: 逐层关联传播算法: 根据前向传播计算总归因分数

我们具体阐述如何计算网络中某一个神经元 a_j 的归因分数。箭头 (\rightarrow) 和 (\leftarrow) 分别代表第 l_m 层和第 l_{m+1} 层的前向连接和反向连接关系, $r_{n \rightarrow z}$ 表示由神经元 n 到神经元 z 的前向传播关系, $R_{n \leftarrow z}^{(l_m, l_{m+1})}$ 表示第 $m+1$ 层的神经元 z 到第 m 层的神经元 n 的反向传播关系。在前向传播算法中, 神经元 a_{i1} 接受上一层输入的像素值 x_1 并计算 $r_{a_{i1} \rightarrow a_j}^{(l_m, l_{m+1})} = \sigma(x_1 * \omega + b)$, 然后将计算结果传输给第 l_{m+1} 层的神经元 a_j ; 同理, 输出层 l_{m+1} 的神经元 a_j 接受来自上一层神经元 a_{i2} 的计算结果 $r_{a_{i2} \rightarrow a_j}^{(l_m, l_{m+1})} = \sigma(x_2 * \omega + b)$, 那么模型输出的结果为 $r_{a_j} = r_{a_{i1} \rightarrow a_j} + r_{a_{i2} \rightarrow a_j} + b_{a_j}$ 。如图3.2所示, 神经元 a_4 接收来自输入层的神经元 a_1 、 a_2 、 a_3 的计算结果, 分别为 $\sigma(x_1 * \omega + b)$ 、 $\sigma(x_2 * \omega + b)$ 、 $\sigma(x_3 * \omega + b)$, 神经元 a_4 接收的总输入值为 $r_{a_1 \rightarrow a_4}^{(l_1, l_2)} + r_{a_2 \rightarrow a_4}^{(l_1, l_2)} + r_{a_3 \rightarrow a_4}^{(l_1, l_2)} + b_{a_4}$ 。

$Cr_{a_i}^{l_m}(x_i)$ 表示第 m 层的神经元 a_i 对于模型输出的归因分数。因为输出层的神经元的归因分数等于模型的输出, 对于图3.2中的神经元 a_{11} , 有 $Cr_{a_{11}}^{l_4}(x_i) = y_1$ 。

根据神经网络相邻层间的线形关系和反向传播算法, 第 m 层神经元和第 $m+1$

层神经元的反向关联性表示为：

$$R_{a_i \leftarrow a_j}^{(l_m, l_{m+1})} = \begin{cases} \frac{r_{a_i \rightarrow a_j}}{r_{a_j} + b_{a_i}} \cdot Cr_{a_j}^{l_m}, r_{a_j} \geq 0 \\ \frac{r_{a_i \rightarrow a_j}}{r_{a_j} - b_{a_i}} \cdot Cr_{a_j}^{l_m}, r_{a_j} < 0 \end{cases}$$

第 m 层的神经元 a_i 的归因分数 $Cr_{a_i}^{l_m}(x_i)$ 即为与之相关联的第 $m+1$ 层的所有神经元 $a_j \in l_{m+1}$ 的归因分数总和：

$$Cr_{a_i}^{l_m}(x_i) = \sum_{a_j \in l_{m+1}} Cr_{a_i \leftarrow a_j}^{l_m \leftarrow l_{m+1}}(x_i)$$

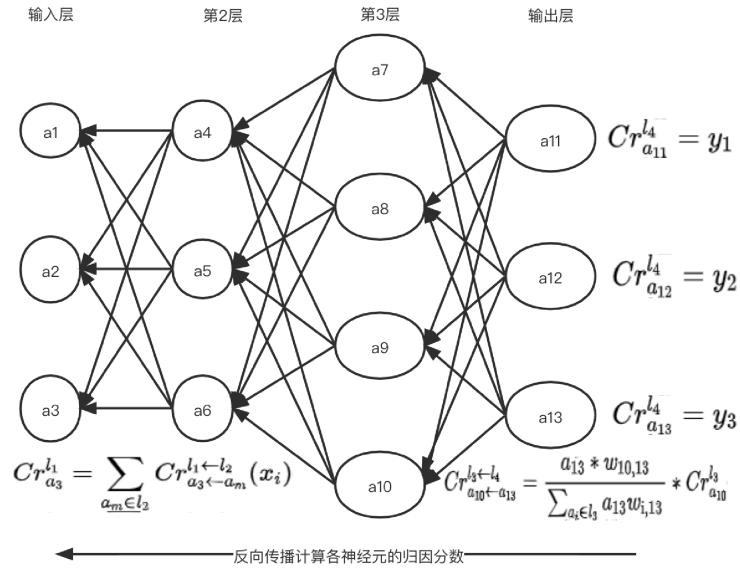


图 3.3: 逐层关联传播算法：根据反向传播计算各神经元的归因分数

如图3.3所示，第 1 层的神经元 a_3 的归因分数表示为：

$$Cr_{a_3}^{l_1} = \sum_{a_m \in l_2} Cr_{a_3 \leftarrow a_m}^{l_1 \leftarrow l_2}(x_i)$$

根据以上公式的推导，我们能得到神经网络中各个神经元的归因分数。那么模型的输出可以表示为各层神经元归因分数的累加和：

$$\begin{aligned} \sum f(x_i, \omega_i^r) &= Cr_{a_{11}}^{l_4}(x_i) + Cr_{a_{12}}^{l_4}(x_i) + Cr_{a_{13}}^{l_4}(x_i) \\ &\quad + Cr_{a_7}^{l_3}(x_i) + Cr_{a_8}^{l_3}(x_i) + Cr_{a_9}^{l_3}(x_i) + Cr_{a_{10}}^{l_3}(x_i) \\ &\quad + Cr_{a_4}^{l_2}(x_i) + Cr_{a_5}^{l_2}(x_i) + Cr_{a_6}^{l_2}(x_i) \\ &\quad + Cr_{a_1}^{l_1}(x_i) + Cr_{a_2}^{l_1}(x_i) + Cr_{a_3}^{l_1}(x_i) \end{aligned}$$

其中， $\sum f(x_i, \omega_i^r)$ 等于模型的总输出。根据以上公式，已经可以得到每个神经元的归因分数和神经网络各层的归因分数。

根据每个神经元的归因分数和模型输出，我们可以计算出每层神经网络对模型输出的平均贡献：

$$Cr_j(x_i) = \frac{1}{n} \sum_{i=1}^n Cr_{x_{i,j}}(x_i), j \in [1, u] \quad (3.1)$$

和每个神经元 a_i 对于模型输出的贡献率：

$$\ddot{Cr}_j(x_i) = \frac{Cr_{a_i}^{l_m}(x_i)}{\sum_{i=1}^n Cr_{x_{i,j}}(x_i)}, j \in [1, u] \quad (3.2)$$

梯度的自适应加躁

所谓梯度的自适应加躁，即根据神经元对模型输出的贡献率分配不同的隐私预算。对于贡献率较大的特征，分配的隐私预算更高，即在梯度上添加的噪声量更小；反之亦然。在上一节中，通过逐层关联传播算法对模型进行预训练，得到了每层神经元对于模型输出结果的贡献率，贡献率决定了给梯度加躁的隐私预算的大小。

在完成模型的预训练之后，进入模型的正式训练环节。重新初始化神经网络的训练参数，根据第二章所介绍的随机梯度下降算法，找到最终的权重参数 $\hat{\theta} \in \mathbb{R}^d$ ，使经验风险值最小。具体的算法如2所示，关键的步骤包括：计算损失函数对于所有权重和偏置的梯度，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，重复上述步骤直至模型收敛，最后输出权重 θ_t 。

。

3.2.2 梯度的自适应裁剪算法

在传统的差分隐私随机梯度下降算法中，提供隐私保护的常用技术是限制函数的敏感度并添加与敏感度界限成比例的高斯噪声。而且，相关研究表明合适的梯度裁剪能加快模型的收敛速度。为此，我们需要在每一轮 SGD 上限制梯度的敏感

Algorithm 2 梯度的自适应加躁算法

```

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  $\eta_t$ , 初始隐私预算  $\sigma_l$ , 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:   for  $x_i \in L_t$  do
6:     计算梯度:  $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$ 
7:     根据神经元对模型输出的贡献率分配相应的隐私预算:  $\sigma_i = \frac{\sigma_l}{C r_j(x_i)}$ 
8:     在梯度上添加高斯噪声:  $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \sum_i (\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, S_f^2 \cdot \sigma_i^2))$ 
9:   end for
10:  计算平均加躁梯度:  $\tilde{w}^t = \frac{1}{L} \sum_{x_i \in L_t} \tilde{w}^t(x_i)$ 
11:  梯度下降:  $\theta^{t+1} = \theta^t - \eta_t \tilde{w}^t$ 
12: end for
13: 输出:  $\theta_t$ 

```

性。Abadi^[65] 等人提出通过梯度裁剪使得梯度保持在 [-C,C] 的范围内, 以保证函数的敏感度有界。如果损失函数是可微的 (如果不可微, 则使用子梯度) 和 Lipschitz 有界的, 用 Lipschitz 界限制梯度范数, 并用它来推导出梯度的敏感度。如果损失函数导数作为输入的函数有界, 可以推导出梯度敏感度。如果损失函数不像深度学习应用中那样具有已知的 Lipschitz 界, 则很难推导出梯度范数的先验界。

在固定的梯度裁剪算法中, 训练初期无从得知梯度范数的先验界限, 所以基本是采用一个固定的梯度范数进行裁剪。假设用户上传的梯度向量为 \mathbf{g}_t , 根据固定梯度范数进行裁剪后, 梯度缩放为 $\mathbf{g}/\max(1, \frac{\|\mathbf{g}\|_2}{C})$, 其中 C 是梯度阈值。对于梯度的裁剪能保证梯度值小于梯度阈值时, 也就是当 $\|\mathbf{g}\|_2 \leq C$, \mathbf{g} 保持不变; 当 $\|\mathbf{g}\|_2 > C$ 时, 它会按照裁剪比例缩小为 C 。在每次训练迭代中, 可以使用经验值来获得梯度范数的近似界限, 并在损失函数近似界限处裁剪梯度。然而, 经验值的可用性是一个强有力的前提, 在没有经验值的情况下如何针对自适应添加的噪声裁剪梯度是一个难题。如果梯度阈值 C 的值太小, 那么裁剪后的梯度会较小, 算法添加的噪声较小时可能会破坏梯度估计的无偏性; 另一方面, 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。神经网络的架构、损失函数本身、数据的缩放都会影响裁剪范数的选择。本章节所设计的方案根

据训练轮数和梯度变化的偏差与方差动态的更新梯度裁剪阈值，对梯度进行自适应裁剪。

在深度学习的模型训练中，模型的泛化能力取决于预测值的方差、偏差和数据的噪声。偏差度量的是模型预测值与真实值之间的偏离程度；方差度量的是训练数据的变动给模型预测结果带来的影响，也就是噪声的添加会影响梯度的方差，而随机梯度的方差决定了 SGD 算法的收敛速度，梯度的裁剪会影响偏差。因此我们更关注梯度更新的方差和偏差来决定如何对梯度进行裁剪。之前的梯度裁剪算法给梯度本身添加了额外的噪声，因此我们考虑根据训练时观察到的历史梯度的统计数据来设置梯度阈值，通过计算梯度更新的偏差和方差在每轮随机梯度下降中更新梯度阈值。

首先，我们对梯度进行裁剪，裁剪后的梯度为 \hat{w}^t ：

$$\hat{w}^t = \text{clip}(w^t, C^t) \triangleq w^t \cdot \min\left(1, \frac{C^t}{\|w^t\|_2}\right) \quad (3.3)$$

然后对保留的梯度 \hat{w}^t 根据神经元的归因分数添加高斯噪声 $\mathcal{N}(0, S_f^2 \cdot \sigma^2)$ ：

$$\tilde{w}^t = \hat{w}^t + N^t \quad N^t \sim \mathcal{N}(0, \sigma^2 I) \quad (3.4)$$

理想情况下，裁剪后的梯度对于模型的收敛的影响应该很小，因此我们希望在每一轮梯度下降算法中找到最佳的裁剪阈值 C^t 使 $\mathbb{E} \|\tilde{w}^t - w^t\|^2$ 最小。

根据三角不等式和 Jensen 不等式，更新前后的梯度方差与偏差可以表示为以下公式：

$$\text{bias}(\tilde{w}^t) \leq \text{bias}(w^t) + 2\mathbb{E} \|\tilde{w}^t - w^t\| \text{ 和 } \text{Var}(\tilde{w}^t) \leq 3\text{Var}(w^t) + 6\mathbb{E} \|\tilde{w}^t - w^t\|^2 \quad (3.5)$$

我们通过约束 $\mathbb{E} \|\tilde{w}^t - w^t\|$ 找到最佳的裁剪阈值 C^t ，将上述公式转换为：

$$\mathbb{E} \|\tilde{w}^t - w^t\|^2 = \|w^t\|^2 \left(1 - \frac{1}{\max(1, \|w^t\|)}\right)^2 + C^{t2} \sigma^2 \quad (3.6)$$

公式3.6中的第一项对应于变换后的梯度 w_t 可能被裁剪的情况，第二项对应于注入到裁剪梯度中的高斯噪声。理想情况下，我们希望能找到使上述表达式3.6最小化的裁剪阈值 C^t 。

为了使预测的梯度值的偏差最小，根据上一轮迭代得到的加躁梯度，通过指数渐进平均估计可得：

$$m^t = \beta_1 m^{t-1} + (1 - \beta_1) \tilde{w}^t \quad (3.7)$$

其中 β_1 是指数移动平均线的衰减参数。

为了使预测的梯度值的方差最小，假使梯度没有被裁剪时，根据 $\tilde{w}^t = w^t + C^t N^t$ ，从 $\mathbb{E}(\tilde{w}_i^t - m_i^t)^2$ 推导出 $\mathbb{E}(w_i^t - m_i^t)^2$ ：

$$\begin{aligned} \mathbb{E}(w_i^t - m_i^t)^2 &= \mathbb{E}(\tilde{w}_i^t - m_i^t)^2 + \mathbb{E}(C^t N_i^t)^2 + 2\mathbb{E}(-C^t N_i^t)(w_i^t + C^t N_i^t - m_i^t) \\ &= \mathbb{E}(\tilde{w}_i^t - m_i^t)^2 - \mathbb{E}(C^t N_i^t)^2 - 2\mathbb{E}(C^t N_i^t)(w_i^t - m_i^t) \\ &= \mathbb{E}(\tilde{w}_i^t - m_i^t)^2 - C^{t2}\sigma^2 \end{aligned}$$

我们需要确保 $(w_i^t - m_i^t)^2$ 满足上限和下限：

$$(w_i^t - m_i^t)^2 \approx \min \left(\max \left((\tilde{w}_i^t - m_i^t)^2 - C^{t2}\sigma^2, h_1 \right), h_2 \right)$$

其中， h_1 和 h_2 为常数，我们使用上式的指数移动平均值来估计方差：

$$\begin{aligned} v_t &= \min \left(\max \left((\tilde{g}_i^t - m_i^t)^2 - C^{t2}\sigma^2, h_1 \right), h_2 \right) \\ (s_i^t)^2 &= \beta_2 (s_i^{t-1})^2 + (1 - \beta_2) v_t \end{aligned} \quad (3.8)$$

梯度自适应裁剪算法在每个训练迭代时刻 t 设置梯度裁剪阈值 C^t ，其中每个迭代对应于一个 minibatch 的处理，接着跟踪训练过程中看到的每个批次的梯度范数。在每一轮的梯度聚合之后，根据公式3.7和3.8计算梯度变化的方差和偏差，更新梯度裁剪阈值 C^t 使 $\mathbb{E}\|\tilde{w}^t - w^t\|^2$ 最小。将自适应梯度裁剪应用到随机梯度下降算法中，具体算法如3所示。梯度自适应裁剪算法的动态性导致了 C^t 的自适应设置，该设置由数据、网络和损失动态决定，而不是由用户在训练初始阶段设置固定的裁剪值，根据神经网络各层的均值和统计特征进行梯度裁剪既能限制敏感度有界，也能保留有效的梯度信息。

3.2.3 基于自适应差分隐私的 SGD 算法

结合前两节所提出的自适应梯度加躁和裁剪算法，我们设计了算法4。在本地客户端训练过程中，在随机梯度下降算法中并使用自适应梯度裁剪算法动态调

Algorithm 3 梯度的自适应裁剪算法

```

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  $L$ , 裁剪阈值  $C^0$ 
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:   for  $x_i \in L_t$  do
6:     计算梯度:  $\mathbf{w}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$ 
7:     梯度裁剪:  $\hat{w}^t = \text{clip}(w^t, C^t) \triangleq w^t \cdot \min\left(1, \frac{C^t}{\|w^t\|_2}\right)$ 
8:     在梯度上添加高斯噪声:  $\tilde{w}^t \leftarrow (\hat{w}^t(x_i) + \mathcal{N}(0, S_f^2 \cdot \sigma^2))$ 
9:   end for
10:  计算平均加躁梯度:  $\tilde{w}^t = \frac{1}{L} \sum_{x_i \in L_t} \tilde{w}^t(x_i)$ 
11:  梯度下降:  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{w}^t$ 
12:  根据公式3.8和3.7计算梯度变化的方差和偏差, 更新  $C^t$ 
13: end for
14: 输出:  $\theta_t$ 

```

整梯度阈值, 添加自适应噪声使算法整体满足 (ϵ, δ) -差分隐私, 最小化目标函数 $f(\theta) = \frac{1}{N} \sum_{k=1}^N f_k(\theta)$ 。首先, 通过逐层传播算法对模型进行预训练, 得到网络中各个神经元的归因分数及其对于模型输出的贡献率。然后运行随机梯度下降算法每一轮用户随机采样小批次的 L 个样本, 对于样本集中的每条数据记录, 计算损失函数对于所有神经元的权重和偏置的梯度 $\mathbf{g}_t(x_i)$, 然后将梯度裁剪到一个固定的范围 $[-C_t, C_t]$, 从而控制个体数据对输出结果的影响, 此时梯度的敏感度为 $\Delta_2(f) = \max_{x_i \in D} \|\hat{g}^t\|_2 \leq C$ 。完成梯度裁剪后, 对梯度添加与贡献率成反比的高斯噪声, 得到满足 (ϵ, δ) -差分隐私的梯度数据。之后计算批次大小为 L 的样本集的平均加躁梯度, 然后进行梯度下降, 计算梯度更新的方差和偏差, 调整梯度裁剪阈值 C^t 。不断的重新采样数据, 迭代地进行梯度下降的训练, 使目标函数最小, 输出模型权重 θ_t 。

当噪声参数满足 $\frac{\sigma_t}{C^0} = \frac{\sqrt{2 \ln(1.25/\delta)}}{\epsilon}$ 时, 添加噪声后的梯度满足 (ϵ, δ) -差分隐私。根据差分隐私的可组合性与后处理不变性, 添加噪声后的数据满足差分隐私, 该性质会传递到后续的处理过程, 最终得到的模型权重也满足 (ϵ, δ) -差分隐私。

在下一节, 我们将给出自适应差分隐私 SGD 算法的整体隐私预算 (ϵ, δ) 分析。

Algorithm 4 基于自适应差分隐私的随机梯度下降算法

```

1: 输入: 目标函数  $f(\theta) = \frac{1}{N} \sum_{k=1}^N f_k(\theta)$ , 学习率  $\eta^t$ , 训练批次大小  $L$ , 高斯噪声参数  $\sigma_l$ , 裁剪
   阈值  $C^0$ 
2: 初始化:  $m^0 = 0 \cdot 1, s^0 = \sqrt{h_1 h_2} \cdot 1$ 
3: 随机初始化模型参数:  $\theta^0$ 
4: for  $t \in [T]$  do
5:   以概率  $L/N$  随机采样一批数据集  $L_t$ 
6:   for  $x_i \in L_t$  do
7:     根据逐层传播算法计算神经元对于模型输出的归因分数:  $Cr_{a_i}^{l_m}(x_i) = \sum_{a_j \in l_{m+1}} Cr_{a_i \leftarrow a_j}^{l_m \leftarrow l_{m+1}}(x_i)$ 
8:     计算神经元对模型输出的贡献率:  $\tilde{Cr}_j(x_i) = \frac{Cr_{a_i}^{l_m}(x_i)}{\sum_{i=1}^n Cr_{x_i,j}(x_i)}, j \in [1, u]$ 
9:     计算梯度:  $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$ 
10:    梯度裁剪:  $\hat{g}^t = \text{clip}(g^t, C^t) \triangleq g^t \cdot \min\left(1, \frac{C^t}{\|g^t\|_2}\right)$ 
11:    根据贡献率分配相应的隐私预算:  $\sigma_i = \frac{\sigma_l}{\tilde{Cr}_j(x_i)}$ 
12:    在梯度上添加高斯噪声:  $\tilde{g}^t \leftarrow (\hat{g}^t(x_i) + \mathcal{N}(0, S_f^2 \cdot \sigma_i^2))$ 
13:  end for
14:  计算平均加躁梯度:  $\tilde{g}^t = \frac{1}{L} \sum_{x_i \in L_t} \tilde{g}^t(x_i)$ 
15:  梯度下降:  $\theta^{t+1} = \theta^t - \eta^t \tilde{g}^t$ 
16:  根据公式3.8和3.7计算梯度变化的方差和偏差, 更新  $C^t$ 
17: end for
18: 输出:  $\theta_t$ 

```

3.3 隐私参数分析

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高斯噪声, 保护数据的隐私性。在上一节我们已经证明了此算法满足 (ϵ, δ) -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成本。

我们向梯度中添加高斯噪声以得到加躁后的数据, 根据第二章所给出的高斯机制的定理可知, 当隐私参数 $\sigma = \sqrt{2 \log \frac{1.25}{\delta}} / \varepsilon$ 时, 每一批次的输出都满足 (ε, δ) -差分隐私。考虑到每一批次的训练样本是以概率 $q = L/N$ 从数据集中随机采样的, 根据隐私放大定理和差分隐私的强组合定理, 隐私性由 (ε, δ) -差分隐私扩大到 $(q\varepsilon, q\delta)$ -差分隐私。

然而, 组合定理并没有考虑到特定的噪声分布, 给出的隐私边界较为松散。在

本节中，我们采用“Moments Accountant”(MA) 机制，去计算算法迭代过程中添加噪声所累积的隐私成本。MA 机制计算隐私成本的思想是将隐私损失视为一个随机变量，通过跟踪隐私损失随时间变化的情况，根据组合定理，计算各轮隐私损失的加和分布得到总隐私损失。

假使算法 \mathcal{M} 是满足 (ϵ, δ) -差分隐私的，那么 \mathcal{M} 中的隐私损失随机变量是存在严格的尾部边界。

对于邻近数据集 D, D' ，算法 \mathcal{M} ，额外输入 aux ，算法的输出 $o \in \mathcal{R}$ 的隐私损失表示为：

$$c(o; \mathcal{M}, \text{aux}, D, D') \triangleq \log \frac{\Pr[\mathcal{M}(\text{aux}, D) = o]}{\Pr[\mathcal{M}(\text{aux}, D') = o]}$$

由于相邻数据集的输出参数的分布是完全一致的，因此 $c(\theta, \mathcal{M}_t, \text{aux}, d, d')$ 的估计趋近于 0，可以通过隐私损失随机变量的矩大小来衡量隐私损失。

由于在随机梯度下降算法 \mathcal{M} 中，通过迭代的梯度下降更新权重，每一轮输出的梯度满足差分隐私，根据差分隐私的串并行组合定理和后处理不变性，最终的模型输出也满足差分隐私。我们定义第 λ^{th} 个时刻的矩生成函数：

$$\alpha_{\mathcal{M}}(\lambda; \text{aux}, D, D') \triangleq \log \mathbb{E}_{o \sim \mathcal{M}(\text{aux}, D)} [\exp(\lambda c(o; \mathcal{M}, \text{aux}, D, D'))]$$

为了证明给定算法 \mathcal{M} 的隐私保障，我们对于每一轮的 $\alpha_{\mathcal{M}}(\lambda; \text{aux}, D, D')$ 给出严格的尾部边界，考虑到所有可能的额外输入 aux 和邻近数据集 D, D' ，整体的矩生成函数为：

$$\alpha_{\mathcal{M}}(\lambda) \triangleq \max_{\text{aux}, D, D'} \alpha_{\mathcal{M}}(\lambda; \text{aux}, D, D')$$

$\alpha_{\mathcal{M}}(\lambda)$ 满足串行组合定理和尾部边界定理：

定理 3.3.1(时刻函数的串行组合). 假设算法 \mathcal{M} 是由一系列自适应算法 $\mathcal{M}_1, \dots, \mathcal{M}_k$ 组合而成的，满足 $\mathcal{M}_i : \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \rightarrow \mathcal{R}_i$ ，那么对于任意的 λ 都满足：

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^k \alpha_{\mathcal{M}_i}(\lambda)$$

定理 3.3.2 (时刻函数的尾部边界). 对于任意的 $\varepsilon > 0$, 算法 \mathcal{M} 是满足 (ε, δ) -差分隐私的, 当

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon)$$

根据马尔科夫不等式可以证明当 $\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon)$ 是算法整体满足 (ε, δ) -差分隐私。

综上所述, 对于采用高斯机制实现满足差分隐私的深度学习系统来说, 计算其隐私损失分为如下三个步骤:

- (1) 根据定理3.3.2, 在算法的每一次迭代中, 通过计算 $\alpha_{\mathcal{M}_i}(\lambda)$ 限定其尾部边界
- (2) 结合3.3.1得到整体矩生成函数的尾部边界
- (3) 通过隐私损失随机变量的尾部边界计算得到最佳的 ϵ , 然后得出训练系统整体的隐私损失

3.4 实验评估

3.4.1 实验准备

在这一节中, 我们进行实验来评估本地自适应差分隐私算法在联邦学习系统中的性能。我们模拟了一个联邦学习系统, 每个本地用户由配备 6GB 内存、四核 2.36GHz Cortex A73 处理器和四核 Cortex A53 1.8GHz 处理器的华为 nova3 安卓手机模拟。中央服务器由一台联想服务器模拟, 服务器有 2 个英特尔 (R) 至强 (R) E5-2620 2.10GHZ CPU, 32GB 内存, 512SSD, 2TB 机械硬盘, 运行于 Ubuntu 18.04 操作系统。

每个本地用户采用深度学习中常用的两个经典数据集-MNIST 手写体数字识别数据集^[46] 和 CIFAR-10 数据集^[67] 进行模型训练。

- MNIST 数据集是用于分类任务的常见数据集, 总共包含 70000 个 28x28 像素的手写数字图像, 其中 60000 个为训练图片, 10000 个为测试图片。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，如下图3.4所示。

- CIFAR-10 数据集总共包含 60000 个 32×32 像素的 RGB 彩色图像，其中 50000 个为训练样本，10,000 个为测试样本。图像包含十个类别：马、鸟、猫、鹿、飞机、卡车、汽车、狗、青蛙和船。

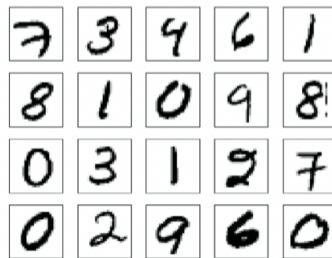


图 3.4: MNIST 手写数字数据集

我们让所有用户离线训练一个统一的卷积神经网络，以获得本地用户的加躁梯度。本地用户训练数据所采用的模型网络结构为一个包含两个卷积层和、两个池化层以及一个全连接层的 CNN。模型的激活函数为 ReLU，并引入了随机失活(Dropout 正则) 避免模型的果泥和。图3.5展示了 CNN 的网络结构。

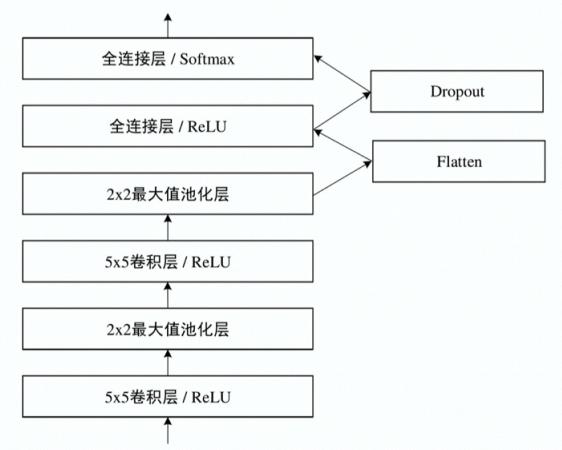


图 3.5: 模型网络结构

所有的实验都是用 PYTHON 语言编译的，本文使用了 TensorFlow-Federated，这是 TensorFlow 中的一个联邦学习库。我们使用 Tensorflow 去实现本地差分隐私

算法，这是一个流行的深度学习库。我们在 Python 的基础上二次开发了该算法，并通过将该算法部署到多个边缘设备上构建了一个真实的联邦学习环境。以本地训练集 MNIST 为例，图3.6展示了联邦学习的仿真模型概览。

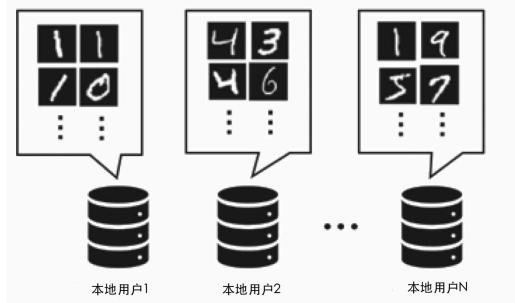


图 3.6: 仿真联邦系统模型概览

3.4.2 实验设计

为了实现联邦学习的隐私保护，我们需要在本地训练的随机梯度下降算法中实现梯度的自适应扰动和裁剪，并采用 MA 机制跟踪每一次梯度扰动所增加的隐私预算。因此，代码的实现主要分为两个部分：梯度扰动和裁剪 (Sanitizer)，隐私预算跟踪 (Accountant)。

图3.7展示了实现梯度扰动和裁剪、隐私预算跟踪的代码片段，为了实现整体算法满足 ϵ -差分隐私，Sanitizer 需要完成以下三个步骤：首先，通过逐层关联传播算法计算神经元的贡献率和每批样本的梯度；其次，根据每个样本的梯度范数进行梯度裁剪以限制函数敏感度；最后，根据贡献率在梯度上添加自适应的噪声然后更新权重。

在 TensorFlow 中，由于性能原因，梯度计算是分批进行的，所以在训练过程，随机采样一批训练子样本 B : $\mathbf{g}_B = 1/|B| \sum_{x \in B} \nabla_{\theta} \mathcal{L}(\theta, x)$ 。为了限制梯度更新的敏感度，我们需要计算每个批次的梯度 $\nabla_{\theta} \mathcal{L}(\theta, x)$ ，具体由 `per_example_gradients` 函数实现。这样即使是大批量的训练，训练速度也不会大幅下降。在每个批次的训练中，我们会单独计算损失函数 \mathcal{L} ，也就是每个数据样本 x_i 都有单独的损失函数结果 \mathcal{L} 。一旦我们获得了每批数据样本的梯度，我们可以很容易地使用 TensorFlow

```

class DPSGD_Optimizer():
    def __init__(self, accountant, sanitizer):
        self._accountant = accountant
        self._sanitizer = sanitizer
    def Minimize(self, loss, params, batch_size, noise_options):
        # Accumulate privacy spending before computing
        priv_accum_op = self._accountant.AccumulatePrivacySpending(batch_size, noise_options)
        with tf.control_dependencies(priv_accum_op):
            #计算每批样本的梯度和其归因分数
            px_grads, px_grads_socre = per_example_gradients(loss, params)
            #裁剪梯度
            px_grads = clip_gradients(px_grads, threshold)
            #梯度加噪
            sanitized_grads = self._sanitizer.Sanitize(px_grads, noise_options)
            #梯度下降
            return apply_gradients(params, sanitized_grads)
    def DPTrain(self, loss, params, batch_size, noise_options):
        accountant = PrivacyAccountant()
        sanitizer = Sanitizer()
        dp_opt = DPSGD_Optimizer(accountant, sanitizer)
        sgd_op = dp_opt.Minimize(loss, params, batch_size, noise_options)
        eps, delta = (0, 0)
        #只要隐私预算在预先设定的限度内，就继续训练。
        while within_limit(eps, delta):
            sgd_op.run()
            eps, delta = accountant.GetSpentPrivacy()

```

图 3.7: 实现本地自适应差分隐私的伪代码片段

操作符来对梯度进行裁剪，添加高斯噪声。

我们的实验主要分为三个部分：

- (1) 针对本地自适应差分隐私 SGD 方案，分析噪声水平、裁剪阈值、隐藏层数量这些参数对模型分类准确率影响。
- (2) 将本地自适应差分隐私 SGD 方案与非隐私的 SGD、前人提出的差分隐私 SGD 方案（如表所示）进行对比，比较各个方案在相同隐私预算的情况下模型分类所能达到的准确率和模型收敛速度。
- (3) 针对本地自适应差分隐私的联邦学习模型进行成员推理攻击进行实验，评估模型的隐私保护效用。

基准方案名称	具体算法
SGD	没有实现差分隐私的随机梯度下降算法
DP-SGD ^[57]	在梯度上添加固定噪声大小的差分隐私随机梯度下降算法
DS-SGD ^[?]	在梯度下降过程中，选择性的进行参数共享实现隐私保护
LDP-SGD	本地差分隐私方案
ADP-SGD	我们的改进方案，使用梯度自适应加噪与裁剪

表 3.1: 本地自适应差分隐私与其他四种基准方案

3.4.3 结果分析

实验一（分析各个参数对模型准确率的影响）

分类模型的精度由多个因素决定，这些因素包括网络的拓扑结构、隐藏单元的数量以及模型训练的参数，如批量大小和学习率，必须仔细调整以获得最佳性能。有些参数是针对隐私的，如梯度范数裁剪阈值和噪声水平。本节实验重点研究噪声大小，梯度裁剪阈值和隐藏层数量这三种参数对于模型分类准确率的影响。为了准确的反映每种参数对于准确率的影响，我们控制变量的进行实验。参考值如下：1,000 个隐藏层单元，600 个批量，初始梯度范数裁剪阈值为 4，初始的学习率为 0.1，在 10 个训练轮次之后线性衰减至 0.051。噪声参数 σ 分别为 2 和 5，用于训练 CNN 网络模型。对于每一种参数组合进行模型训练，直至隐私预算累积至 $(2, 10^{-5})$ -差分隐私。具体的实验结果如图3.8所示。

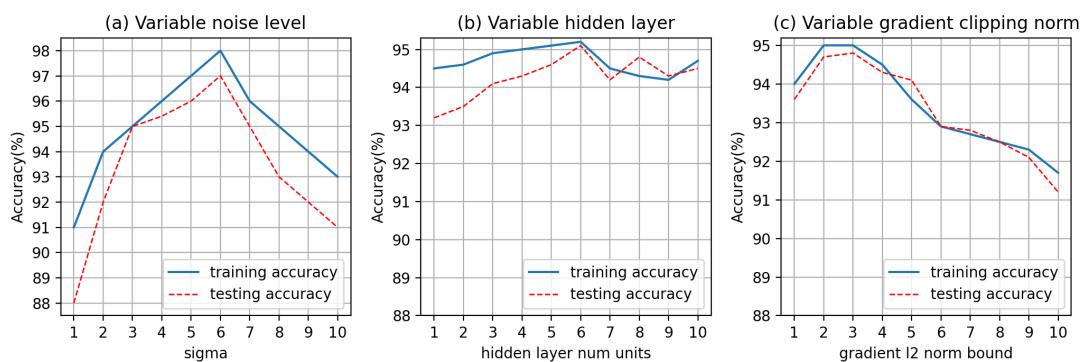


图 3.8: 在 MNIST 数据集上噪声大小，裁剪阈值和隐藏层数量这三个参数对于训练准确率的影响

如图3.8 (a) 展示的是噪声参数 σ 对模型准确率的影响, X 轴是噪声水平, 这个值的选择对准确性有很大影响。由于噪声参数 σ 与噪声采样的分布的方差是呈反比的, 这意味着 σ 越大, 添加的噪声量越小。通过添加更多的噪音, 每轮训练步骤的隐私损失成比例缩小, 所以我们可以在给定的累积隐私预算内运行更多的训练轮次。该模型在训练集和验证集上模型的准确率维持在 88% 至 98% 之间, 我们的框架允许对梯度阈值进行自适应控制减少了过拟合情况, 自适应的噪声添加根据训练轮次合理分配隐私预算, 同时提高了模型的准确性和训练性能。

图3.8 (b) 展示了隐藏层数量对模型准确率的影响, X 轴表示隐藏层单元数量。对于非差分隐私模型, 更多的隐藏单元能有效避免过度拟合, 因为更多的隐藏单元会让我们的训练更有针对性。然而, 添加了差分隐私的模型训练, 隐藏层数量的增加可能会影响梯度的敏感度, 使每次梯度更新时添加更多的噪声。针对这个问题, 我们的根据梯度的贡献率自适应添加噪声的方案能有效的控制敏感度有界, 随着隐藏单位的数量增加, 模型的准确率依然维持在 93% 以上。

图3.8 (c) 展示了梯度范数裁剪阈值对模型准确率的影响, X 轴表示梯度 ℓ_2 -范数裁剪阈值 C^0 。当 C^0 为 2-3 时, 模型准确率最高, 接着随着 C^0 的增加, 模型准确率逐渐降低至 91% 左右, 限制梯度范数会产生两个相反的效果: 剪裁破坏了梯度估计的无偏性, 如果剪裁参数太小, 被剪切的平均梯度可能与真实梯度的方向大不相同。另一方面, 增加裁剪阈值迫使我们在梯度中加入更多的噪声, 也就是以 σC^0 的比例添加噪声。而我们的自适应梯度裁剪方案能有效的考虑上一轮训练得到的梯度偏差和方差, 取训练过程中未被剪辑的梯度范数的中值, 模型准确率最高依然能达到 95% 左右。

对于不同隐私预算的训练版本, 我们用同样的网络架构进行了实验: 一个包括 1000 个神经元的 ReLU 隐藏层, 以及 600 个批量大小。为了限制敏感度, 我们设置初始梯度范数裁剪阈值 C^0 为 3。我们报告了四种隐私参数的训练结果, 分别为小 ($\epsilon=0.5$)、中 ($\epsilon=2$)、大 ($\epsilon=4$) 和更大 ($\epsilon=8$), 固定 $\delta = 10^{-5}$, 这里 ϵ 代表训练神经网络的隐私保护水平。学习率最初设置为 0.1, 在 10 个训练回合后线性下降到

0.052，然后固定为 0.052。

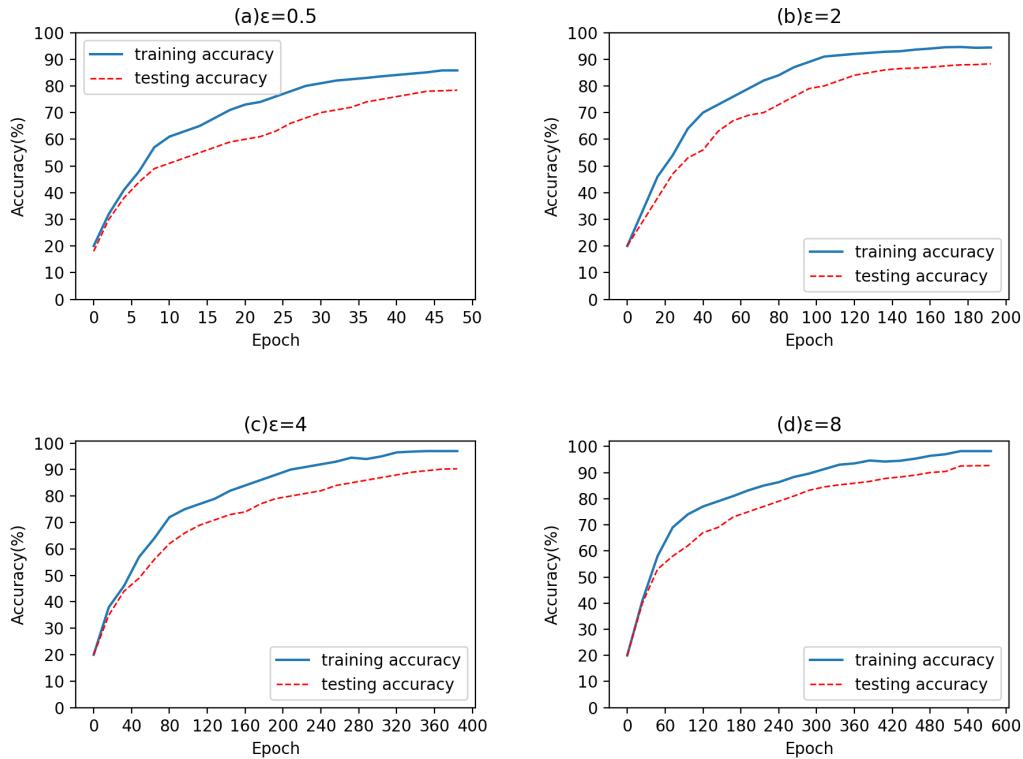


图 3.9: 在 MNIST 数据集上不同隐私预算下训练的准确率

图3.9显示了不同隐私预算下的训练的结果，在每张图中，我们都显示了训练集和测试集上准确率的变化情况。对于隐私预算为 $(0.5, 10^{-5})$, $(2, 10^{-5})$, $(4, 10^{-5})$ 和 $(8, 10^{-5})$ 的差分隐私，分别达到 81%、94%、95% 和 97% 的测试集准确性。由图3.9(a) 所示，当 $\epsilon=0.5$ 时，由于给定的隐私预算较小，在训练的第 30 个轮次隐私预算消耗殆尽，模型基本趋近收敛，模型的准确率最高达到 81%。虽然在传统的差分隐私保护中， $0 < \epsilon < 1$ 时被认为能提供较强的隐私保护。而在联邦学习的深度神经网络中应用差分隐私时，由于网络的复杂性和训练多次大量迭代，导致隐私预算消耗的很快， $0 < \epsilon < 1$ 的取值会导致模型训练的精度大大下降。在深度学习方面应用差分隐私的大量研究表明，当 $0 < \epsilon <= 10$ 时，能提供较强的隐私保护效果。如图3.9(b)(c)(d) 所示，当 $\epsilon=2, 4, 8$ 时，随着训练轮数的增加，模型均能达到收敛。我们的自适应差分隐私 SGD 方案，使模型在训练集和测试集上的准确度差异很小，这与理论上

的观点一致，即添加噪声后的梯度训练依然有很好的泛化作用。相比之下，非差分隐私 SGD 的训练和测试准确率之间的差距随着训练轮次的增加而增加，容易造成过拟合。在噪声参数为 $(8, 10^{-5})$ 时，模型在 600 个训练轮次后能达到接近非差分隐私模型的准确率。

实验二（与前人的隐私保护方案进行对比实验）

我们将本文提出的本地自适应差分隐私方案（ADP-SGD）与 SGD、DP-SGD、DS-SGD、LDP-SGD 方案进行对比实验，选取的数据集为 MNIST 和 CIFAR-10，网络模型为 CNN5，具体的参数设置如下表所示。我们比较了不同方案在给定相同的隐私预算情况下，在测试集上所计算的目标函数的平均损失误差变化情况。

参数	MNIST	CIFAR-10
隐私预算 ϵ	2/4	2/4
批大小	600	600
初始梯度裁剪阈值	3	3
学习率	0.05	0.05
本地设备数量	1000	100
训练轮数	100	100

表 3.2: 对比实验在数据集 MNIST 和 CIFAR-10 上的参数设置

图3.10显示了当隐私预算为 $(2, 10^{-5}), (4, 10^{-5})$ 时，不同方案在 MNIST 和 CIFAR-10 数据集上平均损失误差随训练轮次的变化情况。

首先，对于 MNIST 数据集，在没有添加差分隐私保护的原始 CNN 模型上进行梯度下降训练，经过 20 个训练轮次后模型在训练集上得到的基准准确率为 98.6%，我们的方案（Adaptive Differential Privacy-SGD,ADP-SGD）在训练刚开始的一个轮次，训练集的训练误差下降较慢，这是由于刚开始训练时模型中所有梯度的归因分数较高，导致对于每个梯度分配的隐私预算较低，加躁后与原始值差异较大。然而，在 20 个轮次过后，模型收敛的速度远远超过其他三个基准差分隐私方案。在 50 个训练轮次之后，训练集的损失率降低至 0.25 以下，而 DS-SGD 和 DP-SGD 的

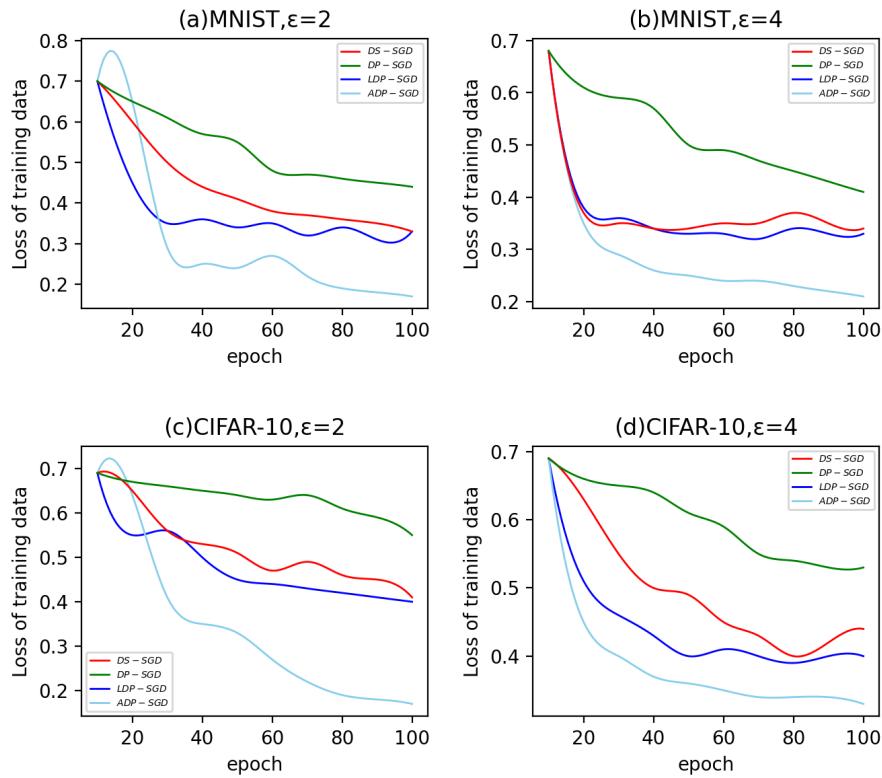


图 3.10: 不同隐私保护方案在 MNIST 数据集上训练的测试误差变化情况

训练损失值在 10 个轮次之后仅降低至 0.35 左右，DP-SGD 更次之，在 0.4 左右。在相同的隐私预算下，ADP-SGD 能在 20 个训练轮次后，梯度范数接近 0.01 且趋于平稳，意味着模型趋近收敛的速度最快。

其次，对于 CIFAR-10 数据集，由于图像本身复杂度的增加，在没有添加差分隐私保护的原始模型上进行梯度下降训练，经过 20 个训练轮次后模型在训练集上得到的基准准确率为 97%。LDP 在前 10 个训练轮次的模型收敛速率最高，然而在第二个训练轮次过后，ADP-SGD 达到 35% 的模型准确率和 0.05 左右的梯度范数，训练数据损失值和梯度范数均比另外三个基准方案的效果优。

算法	隐私预算	$2, 10^{-5}$	$4, 10^{-5}$
SGD	98.6%	98.6%	
DP-SGD	88.9%	92.0%	
DS-SGD	91.4%	93.7%	
LDP-SGD	89.7%	91.3%	
ADP-SGD	94.6%	96.7%	

表 3.3: 本地自适应差分隐私与其他四种基准方案在 100 个训练轮次后模型所能达到的准确率

表3.3显示了各个隐私保护方案在 100 个训练轮次过后所能达到的模型分类准确率。与非隐私的 SGD 方案相比，本文的自适应差分隐私 SGD 方案在提供隐私保护的前提下，模型准确率仅降低了 2-4 个百分点；在相同的隐私预算下，本文的方案相比于前人提出的隐私保护 SGD 算法对模型的准确率的影响更小。

综上，我们的方案在调整梯度自适应加躁和自适应裁剪后使得模型收敛率和准确率进一步提升，无论是在模型的准确率还是收敛速度方面都更加接近原始无隐私保护的模型。

实验三（针对攻击模型，分析该方案的隐私保护效用）

我们曾在第一章介绍了针对联邦学习模型的隐私攻击，其中成员推理攻击是最流行的一类攻击，旨在确定一个输入样本 x 是否存在于模型训练集 D 中。攻击者可以是联邦学习的本地用户之一，也可以是中央服务器。由于深度学习模型对于“训练集中的数据”和“非训练集中的数据”通常会作出不同的行为反应，攻击者通过观察每条训练数据对损失函数的梯度的影响来判断该数据记录是否为目标模型的训练集成员，将成员推理攻击转换为二分类问题。攻击者在更新本地参数之前对目标数据点进行梯度上升，如果该数据记录是目标模型的成员数据，通过 SGD 的算法可以观察到梯度明显下降，成功推断出数据记录的成员属性。当攻击者作为本地用户参与全局模型的训练，攻击者可以观察到全局模型的更新，并通过注

入对抗性的梯度样本，提取出被攻击者的训练数据集的信息。而当攻击者是中央服务器，它通过控制每个目标模型对全局模型更新的变化，提取出目标模型训练数据的分布信息，根据分布信息通过噪声生成数据。

我们重现了文献^[69] 中的成员推理攻击算法，配置和参数与之相同：目标模型采用卷积神经网络进行训练，数据集为 MNIST。我们运行 100 个影子模型，每个模型通过逻辑回归算法进行训练。通过影子学习的技术构建与目标模型的训练数据相近的数据集。针对每一种数据样本，攻击者随机初始化一些数据记录作为影子模型的训练集数据。然后将此数据记录喂给目标模型得到预测向量，将所得的预测向量以及样本标签喂入攻击模型得到二分类的结果。

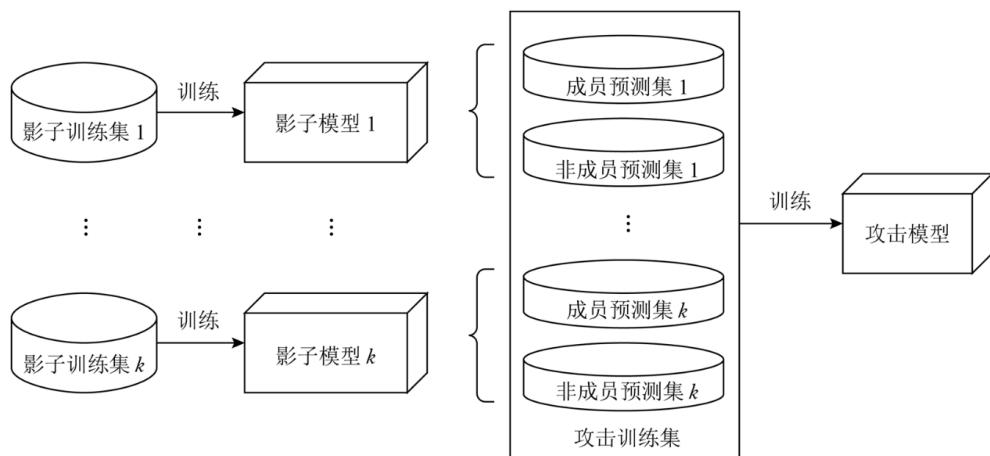


图 3.11: 成员推理攻击过程

该实验的评估指标为攻击者的分类模型预测样本的准确率，它代表了模型的隐私保护效用。理论上在部署了差分隐私的联邦学习模型上进行成员推理攻击的准确率会下降。因此我们对比了在自适应差分隐私（隐私预算为 $\epsilon = 0.5$ ）和无隐私保护的模型上进行成员推理攻击的攻击准确率，结果如图3.12所示。

我们分别在 10k 和 2.5k 个数据样本上进行攻击实验，图3.12中的 (a) 表示在 10k 数据样本上进行攻击实验，x 轴表示对抗攻击的类别，蓝线（original model）表示在原始无隐私保护的模型上进行成员推理攻击的各个类别的攻击准确率，在不同类别上基本都可达到 80%~90% 的准确率。我们可以看到，敌手对识别包含在训练

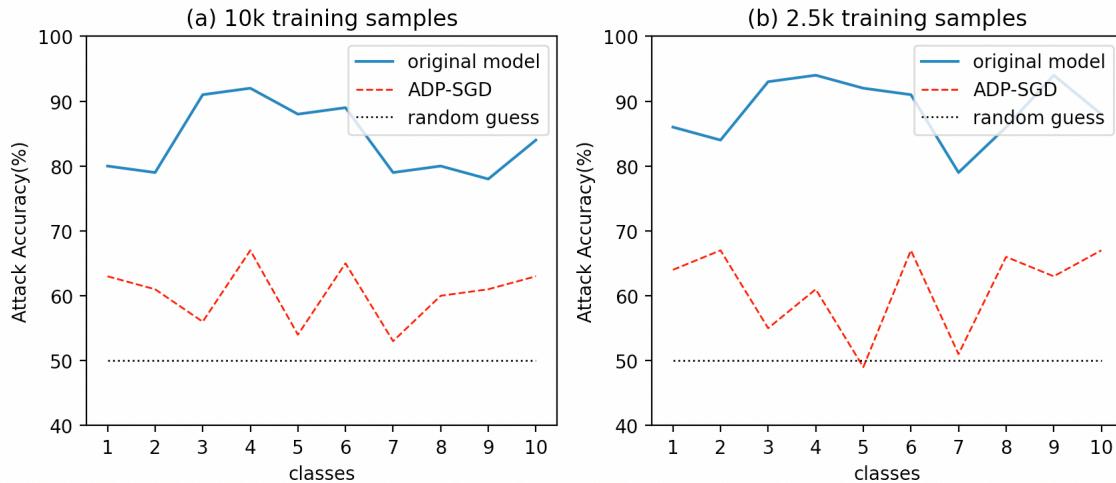


图 3.12: 在不同模型上进行成员推理攻击的准确率

集中的样本有很高的置信值。当样本不在训练集中时，错误率相对较高。图3.12中的红线（ADP-SGD）显示了在添加了自适应差分隐私的模型上进行成员推理攻击的准确性。基准线是 50%（黑色虚线），这是敌手通过随机猜测达到的准确率。在应用差分隐私的情况下，敌手的推断准确率下降到 50%~65%，接近于随机猜测。这比原始模型的攻击准确率要低得多。而在 2.5k 的数据样本上进行攻击，我们的方案针对攻击实验能在 5 这个类别上将准确率降低至 50% 左右，比随机猜测的准确率还要低，实验结果验证了本地自适应差分隐私算法的隐私保护效用。

3.5 本章总结

本章详细介绍了如何在联邦深度学习模型的本地训练算法中实现梯度的自适应加噪和裁剪。我们设计了一个自适应噪声添加的算法，在神经网络中，根据逐层传播算法计算神经元对于模型输出的贡献率，根据贡献率分配隐私预算，在梯度上注入高斯噪声。与传统的注入噪声的方法相比，通过噪声的自适应添加在相同的隐私保护程度下更好地提高了模型的准确性，并且证明了算法整体满足 (ϵ, δ) -差分隐私。接着，本文设计了一种自适应调整梯度阈值的方案，通过计算梯度更新的方差和偏差，逐元素地对梯度进行裁剪，与之前的方案相比，通过使用梯度的自适应剪裁实现了相同的隐私保证，且加速了模型的收敛。之后我们利用“Moments

“Accountant”机制分析加噪累积产生的隐私预算，得到更精准的隐私损失分析。在实验部分。我们分别评估了各个参数对模型精度、收敛速度、隐私成本的影响以及方案的隐私保护效用，证明了我们的方案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，与固定的梯度加躁和裁剪的方案相比提高了模型的准确性和收敛速度。

然而，当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，推导出本地的训练参数信息。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算法，实现联邦学习框架的隐私安全，结合隐私放大效应提高整体联邦学习的通信性能。

第四章 基于 Top-K 混洗差分隐私的联邦学习模型

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实现全局模型的差分隐私。尽管方案采用了梯度自适应加噪和裁剪算法减少一定程度的模型精度损失，但 Truex 等人^[49]指出的，一个复杂的本地隐私保护联邦学习系统将多个本地差分隐私的算法进行组合，会导致总体的隐私成本增长。如果每个本地用户都需要在训练过程中添加 $O(1)$ ，中央服务器将各个数据聚合后，总体噪声的方差达到 $O(n)$ ，标准差达到 $O(\sqrt{n})$ 。在本地设备的模型训练上采用差分隐私技术，对于聚合后的梯度平均估计误差能达到 $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{m}}\right)$ 。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差分隐私，隐私损耗就会成倍累积，从而导致聚合参数上的噪声溢出，影响全局模型的发布结果。在实际的联邦学习应用场景中，本地客户端的数量可能超过千万量级，中央服务器对所有本地上传的加噪梯度进行聚合时，可能因为噪声量的聚合而导致原有的梯度信息被累积的噪声淹没，从而大大降低模型的精度，也增加了通信开销。

在最近的研究工作中，Bittau 等人提出了一个新的隐私保护框架（Encode-Shuffle-Analyze,ESA）。ESA 框架包含 n 个编码器、一个洗牌器和一个分析器。

(1) 编码器 $R : \mathcal{X} \rightarrow y^m$: 编码器对用户的输入数据 x_i 进行加密或者扰动，得到

$$y_{i,1}, \dots, y_{i,m} \in y$$

(2) 洗牌器 $S : (y^m)^n \rightarrow y^{mn}$: 洗牌器可以看作是一个可信任的实体，独立于分析器，接收 n 个用户上传的加密数据，对上传的数据集合中的元素进行随机置

换，得到无序集合再上传至分析器

- (3) 分析器 $A : y^{mn} \rightarrow \mathcal{Z}$: 分析器将混洗器的所有输出信息作为输入，运行全局的聚合函数。

ESA 框架是一个针对 n 个客户和一个中央服务器的隐私保护框架，每个客户本地运行一个编码器，对于所要上传的数据进行加密处理，每个客户可以提交一条或多条消息。相比于客户端直接将消息上传至分析器，ESA 通过在分析器和编码器增加洗牌器，将所有客户上传的加密数据集合中的向量元素进行拆分和随机置换，得到一个无序的消息集合（不包含任何标识信息），于是分析器没有能力获得客户端的 IP 地址、信息上传的时间戳和路由路径等用户隐私。

本文根据前人的研究思路，通过在联邦学习中应用 ESA 框架实现分布式的差分隐私，也叫做混洗差分隐私。本文在联邦学习模型中新增混洗器，混洗器通过对梯度进行动态采样和随机扰动。相比于在原有的 EA 框架中直接应用本地差分隐私，混洗差分隐私在模型可用性方面提高了 $O(\sqrt{n})$ 倍，接近于中央差分隐私的水平。混洗差分隐私在模型准确率方面的增益来源于隐私放大效应，对本地设备的输出进行混洗后在差分隐私的中心视图中比没有混洗的输出提供更强的隐私效用，对于不受信任的中央服务器要达到相同的隐私保护水平，所需要添加的本地噪音更少。本方案通过采样和混洗达到双重的隐私放大，将 $(\epsilon_1 + \epsilon_2)$ 的本地隐私预算放大至 ϵ 中央差分隐私，降低了隐私成本。

此外，在联邦学习的应用场景中，有些本地用户所提供的连接可能为低带宽，巨大的模型参数量给通信网络带来了高负荷的运输负担，那么就需要对服务端和客户端之间的通信带宽进行限制。模型的通信开销受到模型参数大小、客户端数量、通信回合等影响。在全局聚合过程中由于不同的客户端训练和处理数据的速度不同，还可能带来额外的网络延迟。提高联邦学习通信效率的策略分为两种，其一，在联邦学习训练期间减少服务器和客户端之间的通信回合数；其二，在每一次服务器和客户端的通信过程中传输更少的参数。现有的研究包括使用静态抽样来选择一部分客户模型参与全局更新，或者对客户端上传的参数使用压缩算法来

提高通信效率。基于具有较大绝对值的梯度可以为模型收敛做出更多贡献的事实，本文创造性地设计了客户端梯度的 Top-K 采样算法，采用指数机制的打分原理挑选出绝对值排名前 k 位的梯度元素，添加拉普拉斯扰动，实现二阶段的扰动，使本地训练所得的梯度满足 $(\epsilon_1 + \epsilon_2)$ -差分隐私。该方案在相同的中央差分隐私预算下降低了通信成本，缓解了由维度系数增加而带来的隐私成本溢出和模型精度下降的问题。

4.2 模型设计

在本文的方案中，假设敌手为恶意的第三方服务器和诚实但好奇的中央服务器，中央服务器诚实的依据联邦学习的协议完成全局模型的训练，但是它持有用户所上传的梯度的信息，有可能损害用户的隐私数据。此外，我们假设中央服务器和第三方服务器之间不存在串通、混洗器和中央服务器之间不存在串通。

基于上述的威胁模型，隐私要求表述如下：

- 用户的本地梯度的保密性：敌手如中央服务器服务器，可能通过用户上传的梯度信息和模型的全局参数恢复得到用户本地数据信息，比如数据标签和成员信息。为了实现用户数据的隐私性，每个本地梯度在被发送到服务器之前应该通过安全加密。
- 用户所选择的 Top-K 梯度索引信息的保密性：虽然用户上传的梯度值是添加噪声之后的，但是由于梯度的绝对值和其索引信息是一并发送给混洗器的，本方案需要约束中央服务器成功预测一个索引是否在用户本地向量上传的 Top-k 元素中。
- 用户的数据质量的保密性：在联邦学习中，不同的用户数据对于全局模型的训练影响不同，为了保证训练过程的公正，以及防止敌手获取用户的可靠性信息进行联合攻击，用户的数据质量也需要加密，防止任何第三方和中央服务器获取。

4.2.1 模型概览

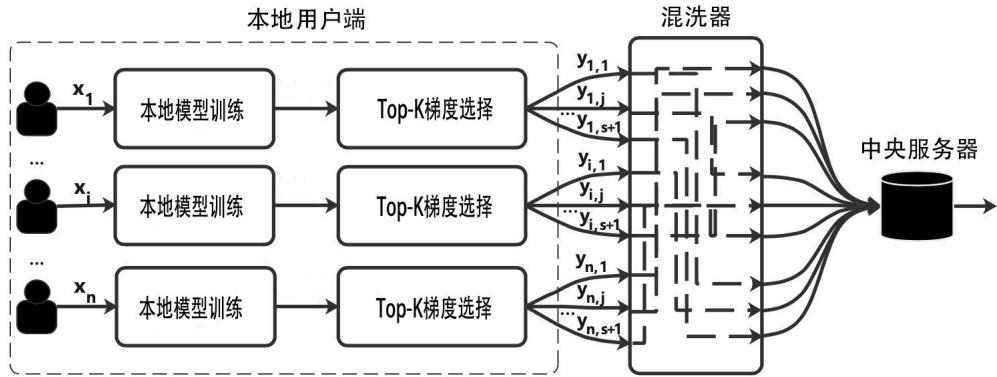


图 4.1: 基于 top-K 安全混洗的联邦学习框架

如图4.1所示，基于 top-K 安全混洗的联邦学习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：**本地设备通过模型训练得到梯度向量 \mathbf{g} ，根据向量中每一元素的绝对值进行降序排序，以较大的概率选择为 top-K 元素的梯度值，在梯度上添加拉普拉斯噪声，再以较小的概率选择非 top-K 元素的梯度值，得到满足 $(\epsilon_1 + \epsilon_2) - LDP$ 的索引列表和梯度元素列表。
- **混洗器：**动态采样客户端的梯度，然后对梯度中的元素排列进行随机置换，通过双重隐私放大效应使得算法满足 (ϵ, δ) -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元素列表和索引列表进行加权聚合后更新全局模型。

假设现在有 m 个本地客户端，每个客户端表示为 $i \in [m]$ ，其本地数据集为 $\mathcal{D}_i = \{d_{i1}, \dots, d_{ir}\} \in \mathbb{S}^r$ 。 $F_i(\theta)$ 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 $\theta^* \in \mathcal{C}$ 使得全局

ERM 最小: $\min_{\theta \in \mathcal{C}} (F(\theta) = \frac{1}{m} \sum_{i=1}^m F_i(\theta))$, 其中隐私性满足总体模型的隐私预算, 也就是满足 (ϵ, δ) -差分隐私。

在算法5中, 首先每个客户端 $i \in \mathcal{U}_t$ 从本地数据集中抽样 \mathcal{S}_{it} 个样本训练模型, 计算梯度 $\nabla_{\theta_t} f(\theta_t; d_{ij})$, 然后运行 Top-K 梯度选择算法对梯度进行采样扰动, 得到满足 $(\epsilon_1 + \epsilon_2) - LDP$ 的梯度元素列表和索引列表 $\langle index_{i,j}, y_{i,j} \rangle$ 。混洗器根据采样率选择客户端更新的集合 \mathcal{U}_t , 并向客户端发送连接请求, 在一个通信回合内收到客户端返回的 ACK, 对收到的梯度 $\langle y_{i,j} \rangle$ 元素排列进行随机置换, 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加权聚合求均值, 更新全局模型。

Algorithm 5 联邦学习中的安全混洗算法: \mathcal{A}_{ssdp}

```

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  $F(\theta) = \frac{1}{mr} \sum_{i=1}^m \sum_{j=1}^r f(\theta; d_{ij})$ , 中央差分隐私预算  $\epsilon$ , 梯度范数阈值  $C$ , 模型学习率  $\eta_t$ , K, 初始客户端采样率:  $C \in \mathbb{R}$ , 联邦学习协议设定的通信回合:  $T \in \mathbb{N}^+$ , 参与联邦学习训练的本地设备集合:  $S = \{s_1, \dots, s_M\}$ 
2: 初始化:  $\theta_0 \in \mathcal{C}$ 
3: for  $t \in [T]$  do
4:   本地更新:
5:   for 客户端  $i \in \mathcal{U}_t$  do
6:     for 样本  $j \in \mathcal{S}_{it}$  do
7:        $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$ 
8:       梯度裁剪:  $\mathbf{g}_t(d_{ij}) \leftarrow \mathbf{g}_t(d_{ij}) / \max \left\{ 1, \frac{\|\mathbf{g}_t(d_{ij})\|_p}{C} \right\}^3$ 
9:       Top-K 梯度选择:  $\langle index_{i,j}, y_{i,j} \rangle = \text{Top-K}(\mathbf{g}_t(d_{ij}), K, \epsilon)$ 
10:      end for
11:    end for
12:    客户端动态采样:  $m, L = \text{Dynamic-Sampling}(t, C)$ 
13:    while  $\text{len}(L) < m$  do
14:      向  $m$  个客户端发送连接请求
15:      if 第  $i$  个客户端返回 ACK then
16:        混洗器接收来自第  $i$  个客户端的加密信息  $\Theta_t^i$ 
17:         $L.add(\Theta_t^i)$ 
18:      end if
19:    end while
20:    全局更新:
21:    混洗器对于  $L$  中的索引元素进行随机置换, 然后上传给中央服务器
22:    中央服务器聚合梯度:  $\bar{\mathbf{g}}_t \leftarrow \frac{1}{ks} \sum_{i \in \mathcal{U}_t, j \in \mathcal{S}_{it}} \langle y_{i,j} \rangle$ 
23:    梯度下降:  $\theta_{t+1} \leftarrow \prod_{\mathcal{C}} (\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 
24:  end for
25: 输出: 最终全局模型参数  $\theta_T$ 

```

我们将在下一节详细的描述该框架中各个模块的设计和实现过程。

4.2.2 Top-K 梯度选择算法

在神经网络的训练过程中，每次迭代的梯度都是从训练样本的小批量子样本中计算得到。直观地说，应用于子样本的算法比应用于全样本的算法具有更强的隐私保证，这种隐私的放大是由于一条数据如果没有在子样本中被选中，就会享有完美的隐私。因此我们在本地用户上传的梯度向量中进行子采样，上传到混洗器。

然而随机子采样对所有梯度向量的维度一视同仁，因此可能会丢弃“重要”的维度。每个本地客户端从 d 维的梯度向量中随机采样并扰动 k 个维度，扰动的值被放大了 d/k 倍以获得无偏的平均估计，同时注入的噪声也被放大了。对于梯度向量高维的情况（比如，梯度是 n 维向量，算法从中随机抽取 k 个维度的梯度值，当 $n \gg k$ 时），从一个向量中随机抽出一小部分就会减慢训练的收敛率。Heafld 提出了梯度稀疏化技术，通过移除梯度向量中绝对值最小 k 个的梯度，使梯度更新算法稀疏化^[76]。由于绝对值较小的梯度会随着时间的推移而累积，会破坏模型的收敛性。梯度稀疏化技术通过评判梯度的重要性，仅上传更重要的梯度值而降低通信成本，避免高维度造成的隐私预算爆炸。本文基于梯度稀疏的想法设计了 Top-K 梯度选择算法，它的主要思想是基于这样一个事实，即具有较大绝对值的梯度可以为模型收敛做出更多贡献，并基于差分隐私选择的技术。

算法发生在本地客户端进行随机梯度下降算法得到梯度向量后、将梯度向量上传至混洗器前。首先，本地用户通过 SGD 算法得到梯度向量 \mathbf{g} ，求得向量中的每一个元素 $\mathbf{g}[i]$ 的绝对值 $abs(\mathbf{g}[i])$ ，然后根据每一维度的绝对值进行降序排序，得到绝对值最大的 K 个梯度值。

因为算法的思想是具有较大绝对值的梯度可以为模型收敛做出更多贡献，可以理解为具有最大绝对值的维度应该以最高的概率输出。如何保证挑选前 top-K 梯度元素的操作满足差分隐私呢？在第二章的基础知识中，我们曾介绍了实现差分隐私的三种机制，其中指数机制是对于任意非数值型的查询，以一定概率返回最佳的查询结果，这里的概率值是由打分函数所确定的。指数机制是为以下情况设

计的：例如在拍卖中设定价格，目标是使收益最大化，而向最优的价格添加上少量的正噪音（为了保护投标的隐私）会大大减少所产生的收入^[35]。对于用户的每个投标价格是需要满足隐私的，但我们又不希望在价格上添加过多的噪音。在本文的场景下，我们希望选择最佳的 k 个梯度值，但直接向梯度中添加噪音会完全破坏模型收敛，两个场景的核心问题是相似的。所以首先考虑通过指数机制挑选出 top-k 个梯度元素。

指数机制中的一个重要概念是实用性函数。对于一个任意的范围 \mathcal{R} ，通过定义一个实用性函数 $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$ ，为用户的数据打分。比如在本文中，梯度绝对值越高，那么实用性函数所给的评分也越高。直观地说，用户通过指数机制中的实用性函数能得到给定数据库 X 中的效用评分最高的数据记录，即 top-1 数据。实用性函数的敏感度表示为：

$$\Delta u \equiv \max_{r \in \mathcal{R}} \max_{x, y: \|x-y\|_1 \leq 1} |u(x, r) - u(y, r)|$$

对于指数机制 $\mathcal{M}_E(x, u, \mathcal{R})$ ，按照正比于 $\exp\left(\frac{\varepsilon u(x, r)}{2\Delta u}\right)$ 的概率从数据集 X 中选择，并输出最优元素 $r \in \mathcal{R}$ ，是满足 $(\varepsilon, 0)$ - 差分隐私。

在差分隐私的指数机制中，K 值为 1，为了将其推广到 top-k 选择中，我们可以简单地迭代应用这个指数机制。本文设计了一种类似于指数机制的扰动采样算法，具体算法流程如图4.2所示。

首先构造一个二维数组，分别存储梯度值的索引 i ，梯度绝对值 $\mathbf{g}[i]$ 。梯度的置信值 u_i 为 1 或者 0，1 表示该梯度值属于 top-K，0 反之。对于第 i 个索引的梯度值 $\mathbf{g}[i]$ ，我们将根据其索引对应的梯度置信值进行划分，得到 top-K 集合 $S_{\text{top}} \leftarrow \{i \mid i \in \text{Top}(|\tilde{x}_i|)\}$ 和非 top-K 集合 $S_{\text{non-top}} \leftarrow \{i \mid i \in [n] \setminus S_{\text{top}}\}$ 。

如图4.3所示，当 K=2 时，根据梯度元素的绝对值进行降序排序，然后根据其索引对应的梯度值是否为 top-K 梯度值进行划分，得到，梯度置信向量 $u = \{u_1, \dots, u_n\} = \{1, 0, 0, 0, 1, 0\}$ ，其中 $S_{\text{top}} = \{1, 5\}$ ， $S_{\text{non}} = \{2, 3, 4\}$ 。

接着对 S_{top} 和 $S_{\text{non-top}}$ 集合按照不同的概率进行采样，从 n 个梯度绝对值及其索引构成的二维数组中以更高的概率 p 随机采样属于 S_{top} 的梯度值 $\{g[1], \dots, g[k]\} \in$

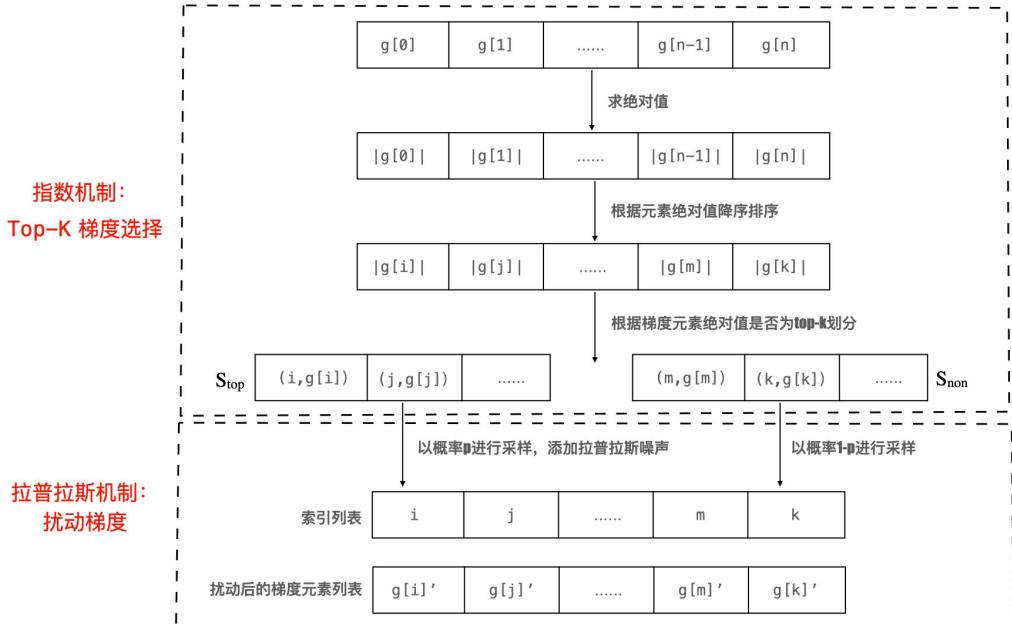


图 4.2: top-K 梯度选择算法流程图

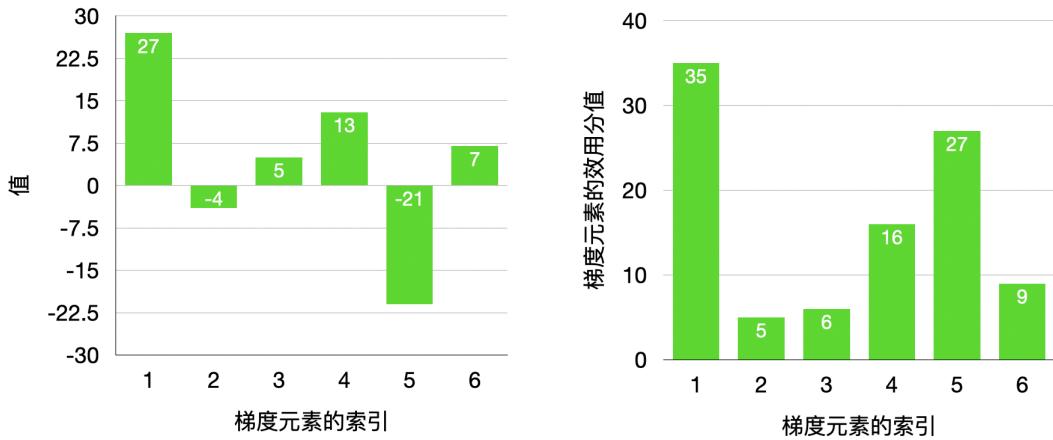


图 4.3: 梯度元素的值及其效用评分

$\{0, 1\}^k$, 在索引对应的梯度上直接添加拉普拉斯噪声:

$$g[i]' = g[i] + \text{Lap} \left(\frac{GS_l}{\epsilon_i} \right)$$

以较低的概率 $1-p$ 随机采样属于 $S_{\text{non-top}}$ 的索引 $\{g[1], \dots, g[n-k+1]\} \in \{0, 1\}^{n-k}$, 将采样扰动后的新的梯度元素及其对应的索引分别存储到两个数组中, 上传至混洗器。其中, $p = \frac{e^{\epsilon} \cdot k}{n-k+e^{\epsilon} \cdot k}$ 。这样的概率选择是满足 ϵ -差分隐私的, 如下给出证明。

定理 4.2.1. 对于给定的梯度分值向量 u, u' , 假设算法输出的索引为 j , 根据条件概率可以给出以下证明:

$$\frac{\Pr[j \mid u]}{\Pr[j \mid u']} \leq \frac{\Pr[j \mid u_j = 1]}{\Pr[j \mid u'_j = 0]} = \frac{p^{\frac{1}{k}}}{(1-p)^{\frac{1}{n-k}}} = e^\epsilon, \text{ where } p = \frac{e^\epsilon k}{n - k + e^\epsilon \cdot k}.$$

Algorithm 6 Top-K 梯度选择算法

```

1: 输入: 梯度向量 g, K, 隐私预算  $\epsilon$ 
2: 初始化: tmp=[], TK=[], S_top=[], S_non-top=[] /*tmp 存储梯度绝对值, TK 存储最终输出的梯度
   元素, S_top 和 S_non-top 分别存储属于 top-k 元素的索引值和不属于 top-k 元素的索引值*/
3: for g[i] ∈ g do
4:   tmp.append(i,abs(g[i])); /*tmp 中的每个元素分别存储数组的索引和对应的元素绝对值*/
5: end for
6: Sort tmp by tmp.values desc; /*根据梯度绝对值对 tmp 数组进行降序排序*/
7: for tmp[i] in tmp do
8:   if i < k then
9:     S_top.append(i);
10:  else
11:    S_non-top.append(i);
12:  end if
13: end for
14: for each index  $i \in S_{\text{top}} \cup S_{\text{non-top}}$  do
15:   if i in  $S_{\text{top}}$  then
16:      $y_{i,j} = \text{tmp}[i].value + \text{Lap}\left(\frac{GS_l}{\epsilon_i}\right)$ ;
17:   else
18:      $y_{i,j} = \omega \mathcal{R} \text{tmp}[i].value$ ;
19:   end if
20: end for
21: 输出:  $\langle index_i, y_i \rangle$ 

```

算法6将满足 $\epsilon_1 - LDP$ 的梯度选择和满足 $\epsilon_2 - LDP$ 的拉普拉斯梯度扰动相结合, 根据差分隐私的组合定理, 整体算法满足 $(\epsilon_1 + \epsilon_2) - LDP$ 。接着分析 Top-K 梯度选择算法的时间复杂度, 与非隐私保护的 SGD 相比, LDP-SGD 给本地设备带来了额外的计算成本。对于 Top-K 梯度选择算法, 所有梯度元素的效用分数可以离线计算。对一个 n 维的向量进行排序需要消耗 $O(n \log n)$ 。计算 n 个维度的元素的效用分数需要消耗 $O(n^2)$, 元素取样需要消耗 $O(n)$ 。因此, 每个本地设备进行梯度选择和概率选择都需要额外的时间成本 $O(n \log n + n^2 + n) = O(n^2)$ 。Top-K

梯度选择方案避免了每个维度的梯度扰动，计算成本相比于无梯度选择的 LDP 低很多。

4.2.3 客户端动态采样

原有的联邦学习是通过静态采样随机选择一部分客户参与联合平均。假使服务器最初设定的采样率为 C ，一旦有足够的客户更新达到初始采样率，服务器将停止接收更新，转而进行全局参数的聚合。在这个过程中，客户端的采样率保持不变，也就是所谓的静态采样，这种方法通过均匀地选择客户的数量来实现参数的聚合。

在深度学习的随机梯度下降算法中，学习率是影响神经网络训练的重要超参数之一，它控制着模型的学习速率，影响目标函数的收敛速度。学习率过大可能会使目标函数在最优值两侧来回移动；学习率过小可能会降低模型的收敛速度。如何设置最优的学习率参数是深度学习中的重要问题。现有经典的方案是在学习初期，设置较大的学习率，使网络以较快的速度进行梯度下降；而在训练后期，学习率逐渐递减，使得神经元的各个权重和偏置能更准确的接近最优解。动态调整学习率的方式包括指数衰减、自然指数衰减和多项式衰减等。本文基于神经网络中动态调整学习率的思想，在联邦学习中动态的调整客户端采样率，使用指数衰减率来递减训练过程中的采样率，其中子采样率 R 是当前通信回合 t 和衰减系数 β 的函数，如公式4.1所示：

$$R(t, \beta) = \frac{1}{\exp(\beta t)} \quad (4.1)$$

算法7是本文所设计的客户端动态采样算法，在每一轮的通信回合，用递减的采样速率乘以预设的初始采样率 C ，得到该轮的客户端采样率为 $c = \frac{C}{\exp(\beta t)}$ ，采样率随着通信轮次的增加而减小。在训练初期，采样率更高，就有更多的客户参与到模型聚合中，加速联邦学习的收敛。一旦在初始训练的基础上得到出一个更通用的联邦学习模型，混洗器就会动态地减少参与全局聚合的客户端数量，以节省通信成本。即使在联邦学习训练刚开始时通信成本较高，但在几轮训练后，其选择的

客户数量会迅速下降。整体来看，动态采样所运输的参数总量少于静态采样，其对全局模型的可用性的影响也较小，会在后续的实验部分给出证明。

Algorithm 7 客户端动态采样算法

```

1: 输入: 初始客户端采样率: $C \in \mathbb{R}$ , 联邦学习协议设定的通信回合: $T \in \mathbb{N}^+$ , 参与联邦学习训练的本地设备集合:  $S = \{s_1, \dots, s_M\}$ 
2: for  $t \in [T]$  do
3:   初始化空链表  $L$ , 用于存放本轮通信回合参与聚合的本地设备上传的信息
4:   计算客户端采样率:  $c = \frac{C}{\exp(\beta t)}$ 
5:   计算采样的客户端数量:  $m = \max(c * M, 1)$ 
6:   while  $\text{len}(L) < m$  do
7:     向  $m$  个客户端发送连接请求
8:     if 第  $i$  个客户端返回 ACK then
9:       混洗器接收来自第  $i$  个客户端的加密信息  $\Theta_t^i$ 
10:       $L.\text{add}(\Theta_t^i)$ 
11:    end if
12:   end while
13: end for
  
```

4.2.4 梯度混洗算法

中央差分隐私基于可信第三方服务器的强依赖实现，本地差分隐私由于其在高维的联邦学习模型中，全局噪声的聚合导致统计结果的可用性降低。本文考虑在本地用户和中央服务器之间引入混洗器，用户在本地对数据添加随机扰动后，将加躁后的数据上传给混洗器，混洗器通过对加躁后的梯度向量进行 shuffle，再将结果发送给中央服务器。梯度的混洗切断了梯度与客户端之间的关联关系，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。通过在联邦学习中添加混洗器实现分布式的差分隐私，也叫做混洗差分隐私。混洗差分隐私兼顾了中央差分隐私下统计结果的准确性和本地差分隐私下数据安全水平高的优点。通过客户端的动态采样和梯度的混洗达到双重隐私放大效应，为实现相同的中央隐私保障所需的本地噪声更少，改善了隐私收益，缓解了高维数据聚合导致的模型可用性降低的问题。算法8是本文所设计的混洗器执行的拆分混洗算法。

如图??所示，假使现有本地模型 X_1, X_2, X_3 ，每个模型都有相同的结构，但

Algorithm 8 混洗器中的拆分混洗算法

-
- 1: **Input:** 本地客户端上传的索引列表和梯度元素列表 $\langle index_{i,j}, y_{i,j} \rangle$
 - 2: **for** $(index_{i,j}, y_{i,j}) \in \langle index_{i,j}, y_{i,j} \rangle$ **do**
 - 3: 对于 $[n]$ 生成一个随机的排列组合 $\pi^{(t)}$
 - 4: $\pi(index_i) \leftarrow (index_{i_1}, index_{i_2}, \dots, index_{i_n})$
 - 5: **end for**
 - 6: 在时刻 t_{id}^s 将索引列表和梯度元素列表 $\langle \pi(index_{i,j}), y_{i,j} \rangle$ 发送给中央服务器
-

权重值不同。原始的联邦学习框架是将本地训练后得到的参数直接发送到中央服务器，而本文中新增混洗器接受客户端上传的索引列表和梯度元素列表，在通信时刻 $(0, T)$ ，对于 $[n]$ 生成一个随机的排列组合 $\pi^{(t)}$ ，随机置换索引列表中元素的位置 $\pi(index_i) \leftarrow (index_{i_1}, index_{i_2}, \dots, index_{i_n})$ ，将索引列表和梯度元素列表 $\langle index_{i,j}, y_{i,j} \rangle$ 发送给中央服务器。

在这个模型中，隐私性是针对整个用户数据集。每个本地用户的数据 x_i 都经过本地的二阶扰动和混洗器的混洗算法。根据差分隐私的串行组合定理，中央服务器接受到的经过本地扰动和混洗的数据：

$$(\mathcal{S} \circ \text{top-K}^M)(\vec{x}) := \mathcal{S}(\text{top-K}(x_1), \dots, \text{top-K}(x_M))$$

是满足 (ϵ', δ) -差分隐私。其中 $\epsilon' = O\left(\frac{\epsilon\sqrt{\log(1/\delta)}}{\sqrt{m}}\right)$ 。混洗结果并不会改变数据集的统计特性，也不会增加 LDP 的隐私预算。

4.3 隐私性和收敛性证明

4.3.1 隐私性证明

在算法5中，每个本地客户端采用 Top-K 梯度选择算法得到满足 $(\epsilon_1 + \epsilon_2) - LDP$ 的梯度元素列表和索引列表，将其上传至混洗器进行混洗后，所获取的数据满足 $\bar{\epsilon} - DP$ 。从 $(\epsilon_c + \epsilon_l)$ 到 $\bar{\epsilon}$ 的转变可通过隐私放大约论证明， $(\epsilon_c + \epsilon_l)$ 对于较高的隐私预算，表示隐私保护的水平更低； $\bar{\epsilon}$ 对应于较低的隐私预算，表示隐私保护的水平更高。因此经过混洗器后，隐私性得到了增强，也就是所谓的隐私放大。隐私放大效应（Privacy Amplification）是针对客户端动态采样和梯度混洗算法增加隐

私保护效用的理论分析。由差分隐私的强组合性可保证所有本地用用户采用 Top-K 梯度选择算法进行扰动后所得的数据总体都满足 $(\epsilon_c + \epsilon_l)$ -LDP，因此本节只需要分析客户端采样和混洗操作的隐私放大性，证明中央服务器接收到的数据满足 ϵ -差分隐私。

假设在联邦学习模型中，初始设定的总通信回合为 $[T]$ 。 $\mathcal{M}_t(\theta_t, \mathcal{D})$ 表示在第 t 轮通信回合对于数据集 \mathcal{D} 和模型参数为 θ_t 部署混洗差分隐私机制， θ_{t+1} 表示模型的输出。因此，在数据集 $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i \in \mathfrak{S}^n$ 上部署的混洗差分隐私机制定义如下：

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{kn} \circ \mathcal{S}_{M,m} (\text{top-K}(\mathbf{x}_{i1}^t), \dots, \text{top-K}(\mathbf{x}_{iM}^t)) \quad (4.2)$$

其中， $\mathcal{S}_{M,m}$ 表示从有 M 个元素的客户端集合中动态采样 m 个客户端的参数， $\text{top-K}(\mathbf{x}_{i1}^t)$ 表示本地客户端采用 Top-K 梯度选择算法得到满足 $(\epsilon_1 + \epsilon_2) - LDP$ 的梯度向量。

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4.2， $\mathcal{Z}(\mathcal{D}^{(t)}) = \mathcal{H}_{kn}(\text{top-K}(\mathbf{x}_1^t), \dots, \text{top-K}(\mathbf{x}_{kn}^t))$ 表示对本地客户端进行模型训练，运行 top-K 梯度选择得到的 kn 个权重集合进行混洗。任取 $\tilde{\delta} > 0$ ，当 $(\epsilon_1 + \epsilon_2) \leq \frac{\log(kn/\log(1/\tilde{\delta}))}{2}$ 时，算法 \mathcal{Z} 满足 $(\tilde{\epsilon}, \tilde{\delta}) - DP$ 差分隐私，可得：

$$\tilde{\epsilon} = \mathcal{O} \left(\min \{(\epsilon_1 + \epsilon_2), 1\} e^{(\epsilon_1 + \epsilon_2)} \sqrt{\frac{\log(1/\tilde{\delta})}{kn}} \right) \quad (4.3)$$

当 $(\epsilon_1 + \epsilon_2) = \mathcal{O}(1)$ 时，有 $\tilde{\epsilon} = \mathcal{O} \left((\epsilon_1 + \epsilon_2) \sqrt{\frac{\log(1/\tilde{\delta})}{kn}} \right)$ 。

如下给出证明：令 $\mathcal{T} \subseteq \{1, \dots, m\}$ 表示在时刻 t 选取的 m 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 $\mathcal{T}_i \in \binom{[r]}{n}$, $i \in \mathcal{T}$ ，事先定义 $\bar{\mathcal{T}} = (\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T})$ ，对于 $i \in \mathcal{T}$ 有 $\mathcal{D}^{\mathcal{T}_i} = \{d_j : j \in \mathcal{T}_i\}$ ，并且 $\mathcal{D}^{\bar{\mathcal{T}}} = \{\mathcal{D}^{\mathcal{T}_i} : i \in \mathcal{T}\}$ 。 \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 $\mathcal{M}_t = \mathcal{Z}(\mathcal{D}^{\bar{T}})$ 。

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 $\mathcal{D}_1 = \{d_{11}, d_{12}, \dots, d_{1r}\}$ 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 $(\bar{\epsilon}, \bar{\delta})$ -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:

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

$$\Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] \leq e^{\bar{\epsilon}} \Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] + \bar{\delta} \quad (4.5)$$

由于式4.4和4.5是对称的, 因此只需要证明其中一条。将输出的概率分布分割成四个条件概率的总和, 这取决于是否挑选了第一个客户端以及第一个客户端是否挑选了第一条数据记录的事件。

下文给出式4.4的证明:

令 $q = \frac{ks}{mr}$, 我们给出条件概率的定义:

$$\begin{aligned} A_{11} &= \Pr[\mathcal{Z}(\mathcal{D}^{\bar{T}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1] \\ A'_{11} &= \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{T}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1] \\ A_{10} &= \Pr[\mathcal{Z}(\mathcal{D}^{\bar{T}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1] = \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{T}}) \in \mathcal{S} \mid 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1] \\ A_0 &= \Pr[\mathcal{Z}(\mathcal{D}^{\bar{T}}) \in \mathcal{S} \mid 1 \notin \mathcal{T}] = \Pr[\mathcal{Z}(\mathcal{D}'^{\bar{T}}) \in \mathcal{S} \mid 1 \notin \mathcal{T}] \end{aligned} \quad (4.6)$$

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

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

$$\Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] = q A'_{11} + q_1 (1 - q_2) A_{10} + (1 - q_1) A_0 \quad (4.8)$$

根据对称性，对式4.7进行证明：

$$\begin{aligned}
 \Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] &= qA_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 \\
 &\leq q \left(e^{\tilde{\epsilon}} \min \{A'_{11}, A_{10}\} + \tilde{\delta} \right) + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 \\
 &= q \left((e^{\tilde{\epsilon}} - 1) \min \{A'_{11}, A_{10}\} + \min \{A'_{11}, A_{10}\} \right) + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 + q\tilde{\delta} \\
 &\stackrel{(a)}{\leq} q \left((e^{\tilde{\epsilon}} - 1) \min \{A'_{11}, A_{10}\} \right) + qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 + q\tilde{\delta} \\
 &\stackrel{(b)}{\leq} q \left((e^{\tilde{\epsilon}} - 1) (q_2A'_{11} + (1 - q_2)A_{10}) \right) + (qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0) + q\tilde{\delta} \\
 &= q_2 \left((e^{\tilde{\epsilon}} - 1) (q_1q_2A'_{11} + q_1(1 - q_2)A_{10}) \right) + (qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0) + q\tilde{\delta} \\
 &= (1 + q_2) \left((e^{\tilde{\epsilon}} - 1) (qA'_{11} + q_1(1 - q_2)A_{10}) + (1 - q_1)A_0 \right) + q\tilde{\delta} \\
 &= e^{\ln(1+q_2(e^{\tilde{\epsilon}}-1))} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] + q\tilde{\delta}
 \end{aligned} \tag{4.9}$$

令 $ks = qn$, 可得 top-K 安全混洗算法满足 $\bar{\epsilon}$ -差分隐私的。

4.3.2 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。

在算法5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5的第 22 行所示，中央服务器进行聚合后得到结果： $\bar{\mathbf{g}}_t \leftarrow \frac{1}{ks} \sum_{i \in \mathcal{U}_t, j \in \mathcal{S}_{it}} \langle y_{i,j} \rangle$ ，然后通过随机梯度下降算法更新全局模型参数： $\theta_{t+1} \leftarrow \Pi_C(\theta_t - \eta_t \bar{\mathbf{g}}_t)$ 。

既然随机扰动机制是无偏的，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扰动机制的随机性而言的。

令 $F(\theta)$ 为凸函数，考虑这样一个随机梯度下降算法： $\theta_{t+1} \leftarrow \Pi_C(\theta_t - \eta_t \mathbf{g}_t)$ ， \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：

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

由 Nesterov 等人在文献^[50] 中的证明可知，算法5的输出 θ_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5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 4.3.1 (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^[50] 中已有的标准随机梯度下降算法收敛结果中使用的定理4.3.1对 G^2 的约束条件，证明了安全混洗算法可在 $G^2 = L^2 \max\left\{d^{1-\frac{2}{p}}, 1\right\} \left(1 + \frac{cd}{qn} \left(\frac{e^{\epsilon} + 1}{e^{\epsilon} - 1} \right)^2 \right)$ 时达到全局最优解。

4.4 实验评估

4.4.1 实验准备

在本节中，我们进行实验来评估混洗器的性能。所有的实验都是用 PYTHON 语言编译的，其中每个用户都由配备 6GB 内存、四核 2.36GHz Cortex A73 处理器

和四核 Cortex A53 1.8GHz 处理器的华为 nova3 安卓手机代替。中央服务器是用两台联想服务器模拟的，这两台服务器有 2 个英特尔 (R) 至强 (R) E5-2620 2.10GHZ CPU，32GB 内存，512SSD，2TB 机械硬盘，运行于 Ubuntu 18.04 操作系统。

在实验过程中，我们选择了深度学习中常用的两个经典数据集-MNIST 手写体数字识别数据集、FMNIST 和 CIFAR-10 数据集进行实验，评估所提出的安全混洗框架。此外，我们让所有用户离线训练一个统一的卷积神经网络，以获得本地用户的梯度。在我们的实验中采用的模型网络结构为 CNN，包括 2 个卷积层，两个池化层层和一个全连接层（32 个神经元）。模型的激活函数为 Softmax，并引入了 DropOut 正则以提高模型的泛化能力。下表展示了 CNN 的网络结构。

神经层	参数
卷积层	8×8 的 16 个滤波器，步长为 2
池化层	2×2
卷积层	4×4 的 32 个滤波器，步长为 2
池化层	2×2
全连接层	32 个神经元
Softmax	10 个神经元

表 4.1: 安全混洗框架实验的模型网络结构

4.4.2 实验设计

在我们模拟的联邦学习环境中，我们设置本地客户端的总数为 1000 个，其中每个客户有一个本地数据集，每个客户都对梯度 $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$ 进行剪裁，梯度裁剪参数 $C=1/100$ 。之后，运行 Top-K 梯度选择和拆分混洗算法，混洗器将参数上传至中央服务器进行安全聚合，更新全局参数。我们的算法运行了 100 个历时，在前 80 个历时中，我们将学习率设置为 0.3，在剩余的历时中，将其降低到 0.18。在每一次训练迭代过程中，本地客户端与混洗器、中央服务器进行交互计算得到安全聚合之后的全局梯度向量。我们设置了本地隐私参数 $\sigma=2$ ，而中央隐私参数 ϵ 的计算则是由我们来完成的。我们首先使用文献中^[71] 的定理 5.3 通过洗牌数值

计算隐私放大率。然后，我们通过??中提出的子抽样计算隐私放大；最后，我们使用差分隐私的强组合性质来获得中央隐私参数 ϵ 。

我们的实验主要分为两个部分：

- (1) 在 MNIST、FMNIST 和 CIFAR 上评估安全混洗算法，评估参数：客户端数量 N 、梯度选择的比率、客户端采样比 f_r 和最大聚合次数对于模型分类准确率的影响。
- (2) 将本文的安全混洗方案与基准方案（非隐私保护的联邦学习方案）、前人提出的隐私保护联邦学习方案（如表4.2所示）进行对比，评估指标为模型分类准确率和通信性能。
- (3) 在基于梯度选择和安全混洗的联邦学习模型上应用生成对抗网络攻击进行实验，评估模型的隐私保护效用。

基准方案名称	具体算法
FL	没有添加隐私保护机制的联邦学习模型
PS-FL ^[?]	通过选择性的参数更新实现隐私保护的分布式学习框架
DP-FL	基于中央差分隐私的联邦学习模型
LDP-FL	基于本地差分隐私的联邦学习模型
KSA-FL	本文的隐私保护方案，基于梯度选择和安全混洗的联邦学习模型

表 4.2: 安全混洗框架的比较方案

4.4.3 结果分析

实验一（分析各个参数对模型准确率的影响）

在本文所设计的联邦学习安全混洗算法中，本地客户端的数量、梯度选择的比率、客户端采样比都是影响联邦学习模型分类准确率的因素。为了准确的反映每种参数对于准确率的影响，我们控制变量的进行实验。参考值如下：总客户端数量 N 为 1000，梯度选择的比率分别为 1%、5%、10%、50%、100%，总通信回合为 100，在前 80 个历时中，我们将学习率设置为 0.3，在剩余的历时中，将其降低到 0.18。隐

私预算 ϵ 分别为 50、60、100。对于每一种参数组合进行模型训练，具体的实验结果如下文所示。

首先分析安全混洗模型中参与混洗的本地客户端数量对联邦模型分类精度的影响，如图4.4所示，通过 Top-K 梯度选择算法和拆分混洗算法，我们的安全混洗模型（下文简称 KSA-FL）能够以较低的隐私成本实现较高的准确性。在训练中增加客户端数量 N 的同时，KSA-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、FMNIST 相比，CIFAR-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邦学习模型达到更高的精度。

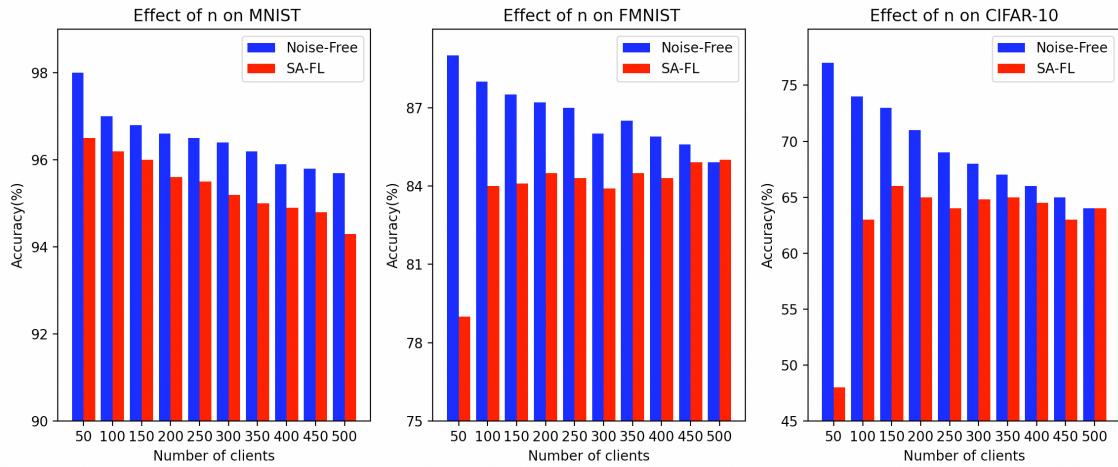


图 4.4: 安全混洗模型中本地客户端数量对联邦学习模型训练精度的影响

如图4.5所示，当固定总客户端数量为 100，在不同的隐私预算参数 $\epsilon=50、70、100$ 和不添加隐私保护的联邦学习模型中，混洗器在每次迭代过程中随机采样部分客户端的梯度进行混洗。图4.5展示了全局损失函数值随客户端采样比的变化情况，总体来说，当客户端采样比为 0.5 左右时，损失函数值最小。联邦学习的隐私保护难点在于使用较低的隐私预算维持较高的模型精度，而选择适宜的客户端数量参与安全聚合会大大提升模型的收敛速度和通信性能。

如图4.6展示了不同的梯度选择比率对联邦学习模型训练精度的影响，X 轴表示联邦模型迭代次数。从图中的曲线变化情况可以看出，各个梯度选择比率的模型

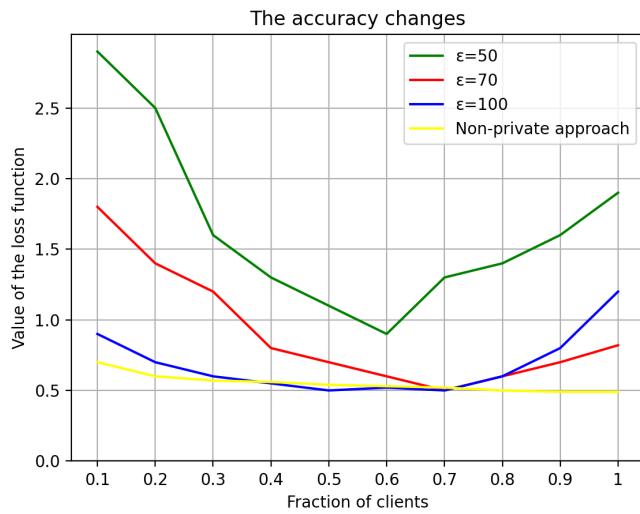


图 4.5: 安全混洗模型中客户端采样比对联邦学习模型训练精度的影响

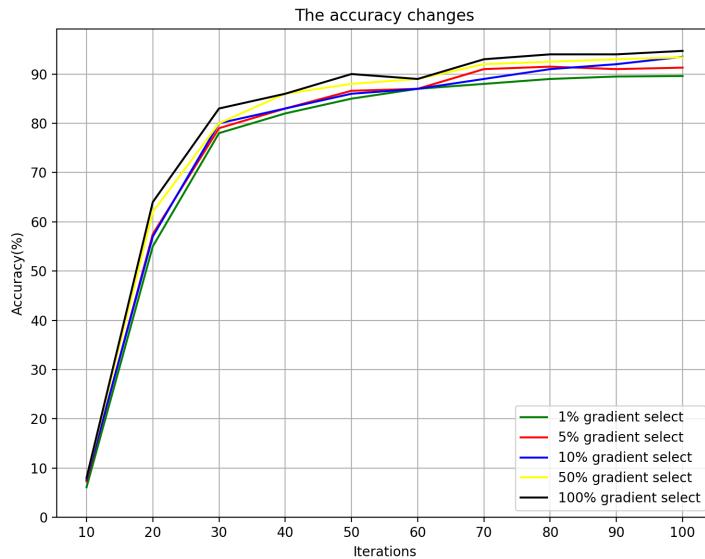


图 4.6: 安全混洗模型中梯度选择比率对联邦学习模型训练精度的影响

收敛速度接近，在 70 个训练回合后基本达到收敛，因此梯度选择并不会影响模型的收敛速度。1% 的梯度选择在历时 100 个训练回合后能达到 83% 的准确率，100% 的梯度选择比率所能达到的模型准确率为 95%，相差了近 10% 的准确率，可见部分梯度的丢弃确实丢失了部分的训练信息。当梯度选择比率低于 10% 时，随着梯度选择比率的增加，模型的准确率也增加了接近 3%-5% 个百分点，然而当梯度选择

比率为 50% 时，模型在 30 个训练回合后的准确率基本与 100% 的梯度选择比率所能达到的模型准确率接近。

在实验中，每一轮的迭代训练都需要由中央服务器和本地客户端交互计算，我们分别计算了不同的梯度选择比率在一次训练迭代中的通信开销，实验结果如表4.3所示。当梯度选择比率下降时，通信开销也基本呈比例下降。当 Top-K 梯度选择比率从 100% 优化至 50% 时，通信性能优化了大约 1.99 倍，而根据图4.6所示，在梯度选择比率从 100% 降低至 50% 时，模型的训练准确率并不会受到太大影响，由此可见一定范围内的梯度选择可以较好的优化联邦学习模型的通信性能。

Top-K/%	通信开销/KB
1	167.022×2
5	863.317×2
10	1684.271×2
50	3502.151×2
100	6995.548×2

表 4.3: 不同梯度选择比率的通信开销

实验二（与前人的隐私保护方案进行对比实验）

我们将本文设计的基于梯度选择与安全混洗的联邦学习隐私保护方案 (KSA-FL) 的与 FL、PS-FL、DP-FL、LDP-FL 方案进行对比实验，选取的数据集为 MNIST，网络模型为 CNN5。我们比较了不同方案在给定相同的隐私预算 ($(0.2, 5e - 6) - DP$) 情况下，模型分类准确率变化情况。

在 $((0.2, 5e - 6) - DP)$ 的隐私预算下，无添加隐私保护的联邦学习基准模型在 18 个训练轮次后能达到 94% 左右的准确率，其余四种实现了联邦学习隐私保护的方案中，本文所设计的 Top-K 安全混洗方案所能达到的准确率最高，在 18 个训练轮次后能达到 90% 的准确率，而在训练开始阶段，模型的准确率提升速度甚至超过了无添加隐私保护的联邦学习基准模型，这表明 Top-K 梯度选择算法能有效的加速模型的学习速率。基于中央差分隐私的联邦学习隐私保护模型在 18 个训练轮

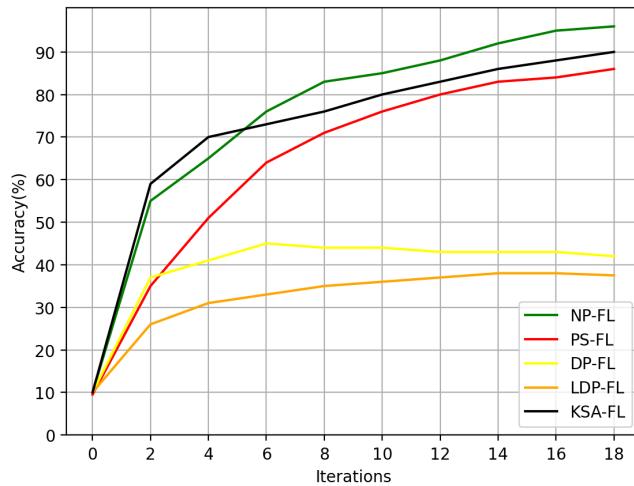


图 4.7: 不同隐私保护方案在 MNIST 数据集上训练的模型分类准确率变化情况

次后能达到 41% 左右的准确率，在 6 个训练轮次过后，模型就接近收敛。基于本地差分隐私的联邦学习隐私保护模型的分类准确率在这几种方案中的表现最低，正如上文分析的，本地差分隐私由于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。

实验三（针对攻击模型，分析该方案的隐私保护效用）

在第一章针对联邦学习的隐私威胁中，我们介绍了生成对抗网络攻击（GAN attack），GAN 程序使一个鉴别的深度学习网络与一个生成性的深度学习网络对立起来，形成对抗生成性深度学习网络，两者以零和博弈的思想进行对立训练。如图4.8为联邦学习下 GAN 模型训练的示意图，生成器首先用随机噪声生成初始数据，然后在每个迭代中，它都被不断的训练，模仿被攻击者的训练集生成数据。鉴别器被训练来区分图像是来自原始数据库还是由 GAN 生成的。在联邦学习中，中央服务器通过聚合所有参与者上传的数据得到全局参数，而鉴别器参与了全局模型的训练，相当于在其他用户的训练数据上训练鉴别器，使得生成器有能力构造出与真实样本相似的伪样本。当生成器当鉴别器无法区分原始数据库的样本和生成器生成的样本时，模型的训练目标就完成了。

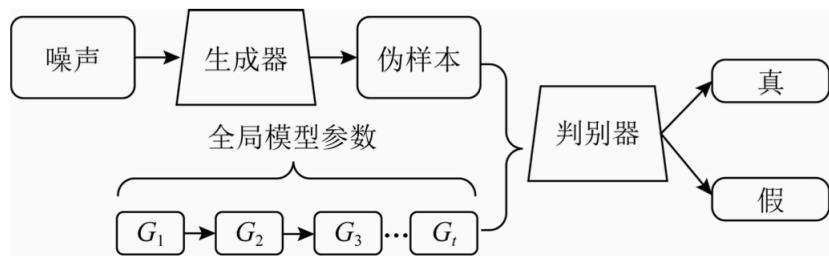


图 4.8: 联邦学习下的 GAN 模型

我们模拟的联邦学习环境中有十个本地客户端参与学习，数据集为 MNIST。所有参与者都预先统一网络的训练架构和学习目标，这意味着他们就神经网络架构的类型和训练的标签达成一致。敌手 A 作为本地客户端参与联邦学习。为了模仿其他参与者的训练数据集的样本，我们在敌手一方采用 GAN 架构，其中 GAN 的鉴别器网络与联邦学习协议中的全局模型相同。当敌手在全局模型的准确率达到 95% 后开始生成伪样本。被攻击的用户表示为 V，V 和 A 分别声明了标签 [a,b] 和 [b,c]。敌手假装是深度学习协议的诚实参与者，但敌手会偷偷地影响学习过程，以欺骗 V，使其释放关于目标类别的进一步细节。攻击者首先从中央服务器下载全局的训练参数，更新本地模型，同时在 V 不知情的情况下训练本地的对抗生成网络，生成 fake 类以模仿 V 的 a 类标签。当 A 的本地 GAN 模型中生成器当鉴别器无法区分原始数据库的样本和生成器生成的样本时，生成器生成的标签 a 即为最终数据。

图4.9显示了分别在不同的隐私保护联邦学习环境下进行 GAN 攻击，生成器训练的伪样本。第一行是其他参与者的真实训练样本。第二至四行分别是采用 DP-FL、LDP-FL 和 KSA-FL 隐私保护方案后攻击生成的伪样本。即使在联邦学习的本地模型或者全局模型上添加了差分隐私保护后，基于 GAN 的生成模型依然可以成功地模仿参与者的原始样本。但是这几种隐私保护的方案中，KSA-FL 上所生成的伪样本相比 DP-FL、LDP-FL 与原始样本的差异更大，本节所设计的 Top-K 安全混洗方案（KSA-FL）在针对 GAN 攻击的隐私保护上效果更加显著。

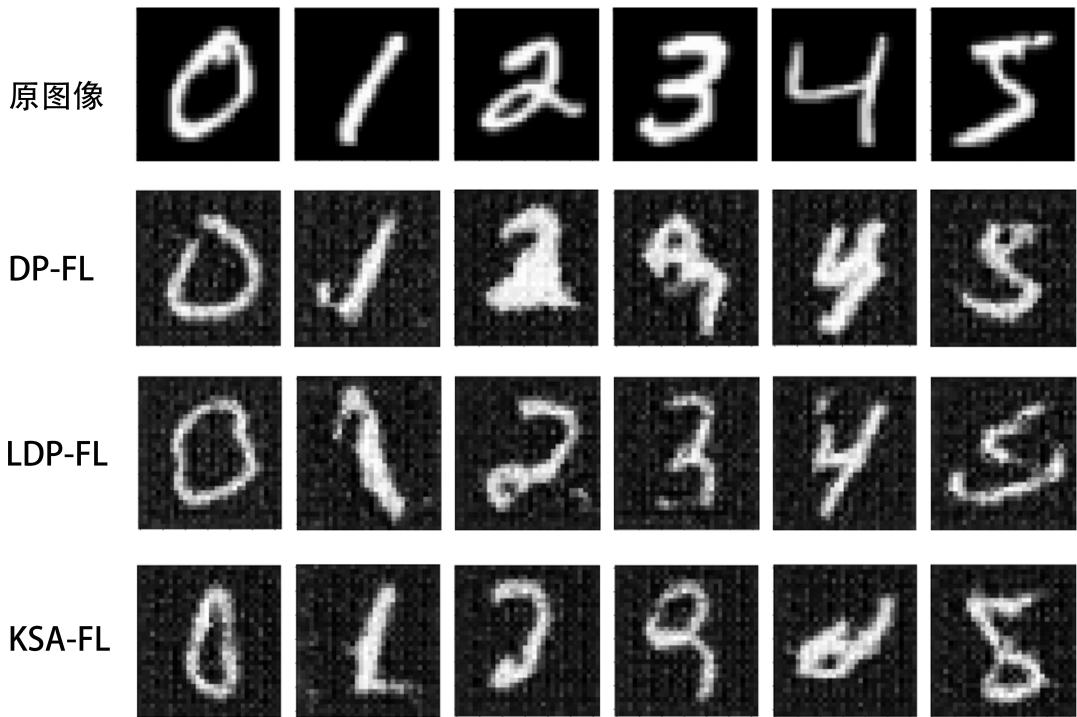


图 4.9: 在 MNIST 数据集上对不同的隐私保护联邦学习算法下进行 GAN 攻击所生成的伪样本

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改进，基于稀疏向量和指数机制的思想，创造性地开发了 Top-k 梯度选择算法，与拉普拉斯机制相结合，设计了满足 (ϵ, δ) -本地差分隐私的算法。此外，我们将客户端采样和梯度混洗这两种隐私放大效应相结合，缓解了由维度系数增加而带来的隐私预算暴增和模型精度下降的问题。我们对方案进行了隐私性证明，表明此安全混洗算法可以保证 ϵ_c 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。最后，通过在三种基准数据集上进行实验，证明本章所设计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，且降低了通信成本。

第五章 总结与展望

5.1 论文总结

随着大数据、机器学习、深度学习的快速发展，人工智能在各个领域都得到了广泛的应用，比如，自动驾驶、医疗诊断、图像识别等。人工智能算法基于大数据实现模型的泛化和学习，而训练数据可能包含了大量的用户敏感信息，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观影偏好都有可能被收集在移动设备和计算机中，这有可能导致用户的隐私泄露。随着隐私泄漏的事件越来越多，数据安全和隐私问题引起了人们的关注。我国在 2018 年正式颁布了《信息安全技术和个人信息安全规范》以规范个人信息收集和存储的安全性。

人工智能的服务性能在很大程度上受到训练数据质量的影响。在大多数行业中，数据是以孤立的岛屿形式存在的。具体而言，不同的机构拥有不同的用户行为数据和特征数据，由于商业竞争模式和法律法规监管，企业之间无法共享数据以构建一个高质量的模型。在破解数据孤岛问题和实现数据隐私保护的双重需求下，联邦学习作为一种新型的深度学习范式应运而生。在联邦学习框架中，有许多参与方，在无需传递和共享本地的数据资源的情况下训练一个共同的、强大的深度学习模型，即数据依然保留在用户侧，而模型是共享的。

与传统的集中式深度学习相比，联邦学习在一定程度上缓解了数据孤岛和隐私泄露的问题。然而，许多研究表明，攻击者仍然可以通过梯度损害用户的隐私。由于联邦学习仍然需要一个能够回答查询函数的聚合模型，这意味着聚合后的梯度仍然包含了本地数据的信息。由于联邦学习的框架并没有对参与方的资质进行

校验、没有对模型的访问权加以约束。一些恶意的参与方通过向中央服务器提供虚假的参数影响联邦学习模型的质量，甚至导致整体模型不可用；还有一些攻击者从通信信道中获取共享梯度，根据梯度反演得到用户数据集相关的信息。差分隐私是当前保护联邦学习隐私安全的前沿技术，其通过严格的统计框架提供隐私保证和隐私成本计算，使得加躁后的梯度不能泄露关于实体数据的敏感信息，与安全多方计算和同态加密等密码学技术相比，差分隐私保护联邦学习的方案在通信性能和计算性能方面与明文计算相差不大。

为了在差分隐私框架下保护联邦学习的隐私性，必须在本地训练过程或全局聚合过程中向模型参数注入噪声。然而当前的差分隐私保护联邦深度学习的方案通常面临模型可用性和数据隐私性的博弈。差分隐私的应用可能导致模型的准确率下降，而且随着迭代次数的增多，高维加躁梯度的聚合会导致梯度中所包含的有效信息被噪声所淹没，影响全局模型的收敛性。如何在保护本地数据隐私的同时降低模型的精度损失，以及在千万量级的联邦学习通信回合下，避免噪声量的成倍增加导致模型的通信性能和可用性大幅下降是当前亟需研究的问题。

本文的研究内容主要针对联邦深度学习系统的成员推理攻击和生成对抗网络攻击设计隐私保护方案，基于隐私性、模型精度、通信性能的三重指标，设计了本地自适应差分隐私算法和 top-K 安全混淆算法，主要的工作和贡献包含以下四个方面：

- (1) 在差分隐私联邦学习的场景下，本文设计了一种新型的、基于本地差分隐私的自适应梯度加躁算法。由于不同的神经网络层的神经元对于模型输出的影响不同，本文设计了一种基于神经元的贡献率添加自适应噪声的算法：在客户端本地训练的神经网络模型中，通过运行逐层关联传播算法，计算每个神经元对于模型输出的贡献率。在随机梯度下降过程中，根据贡献率分配动态的隐私预算，在梯度上注入高斯噪声。在设置相同的隐私预算下，相比于 DP-SGD 算法，我们的方案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，与固定的梯度加躁和裁剪的方案相比提高了模型的准确性

和收敛速度。

- (2) 在传统的差分隐私随机梯度下降算法中，通常采用固定的裁剪阈值对梯度进行裁剪以限制函数的敏感度，然而固定的梯度裁剪可能添加额外的噪声。本文设计了一种自适应调整剪裁阈值的方案，通过计算梯度更新的方差和偏差，逐元素地对梯度进行裁剪，根据神经网络各层的均值和统计特征进行梯度裁剪既能限制敏感度有界，也能保留有效的梯度信息。之后我们利用“*Moments Accountant*”机制分析加噪累积产生的隐私预算，给出更精准的隐私界。
- (3) 由于本地差分隐私并不能有效防御针对联邦学习的生成对抗网络攻击，并且在通信轮数较大的联邦学习模型中，由于差分隐私的强组合性质，噪声量成倍累加，导致整体的隐私成本过高。本文设计了一种新型的联邦学习安全混洗算法，采用指数机制的打分原理挑选出绝对值排名前 k 位的梯度元素，添加拉普拉斯扰动，设计了满足 (ϵ, δ) -差分隐私的 Top-K 梯度选择算法。此外，在联邦学习模型中新增安全混洗器，通过对梯度和索引所构成的矩阵进行置乱，提高了数据的随机性。在每轮通信回合，根据指数衰减机制动态调整客户端采样率，减少整体通信负荷。通过客户端的采样和梯度索引的混洗达到双重的隐私放大效应，降低系统的整体隐私损失，提高了通信性能，并证明了安全混洗框架的隐私性和全局收敛性。
- (4) 为了验证本文的方案在实际生产环境中的可行性，本文模拟了联邦学习环境，分别在 MNIST、CIFAR-10、FMNIST 等数据集上进行实验，首先通过控制变量法分析各个参数对于模型精度和通信性能的影响，并与前人的差分隐私方案和安全混洗方案进行对比，通过实验结果证明了自适应本地差分隐私算法和安全混洗算法，在保护数据隐私的前提下尽可能的减小了模型的精度损耗，降低了通信成本。最后，本文模拟了成员推理攻击和生成对抗网络攻击，评估了自适应差分隐私方案和安全混洗方案针对攻击模型的隐私保护效用。

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。

5.2 论文展望

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] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 1175-1191.
- [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.
- [66] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In International conference on machine learning, pages 314–323, 2016.
- [67] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. 2009.
- [68] Melis L, Song C, De Cristofaro E, et al. Exploiting unintended feature leakage in collaborative learning[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 691-706.
- [69] 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.

- [70] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 1175-1191.
- [71] Balle B, Bell J, Gascón A, et al. The privacy blanket of the shuffle model[C]//Annual International Cryptology Conference. Springer, Cham, 2019: 638-667.
- [72] Young T, Hazarika D, Poria S, et al. Recent trends in deep learning based natural language processing[J]. ieee Computational intelligenCe magazine, 2018, 13(3): 55-75.
- [73] Dwork C. A firm foundation for private data analysis[J]. Communications of the ACM, 2011, 54(1): 86-95.
- [74] 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.
- [75] Binder A, Montavon G, Lapuschkin S, et al. Layer-wise relevance propagation for neural networks with local renormalization layers[C]//International Conference on Artificial Neural Networks. Springer, Cham, 2016: 63-71.
- [76] Aji A F, Heafield K. Sparse communication for distributed gradient descent[J]. arXiv preprint arXiv:1704.05021, 2017.
- [77] Choudhury O, Gkoulalas-Divanis A, Salonidis T, et al. Differential privacy-enabled federated learning for sensitive health data[J]. arXiv preprint arXiv:1910.02578, 2019.
- [78] Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469.