

中国科学技术大学

硕士学位论文



基于深度学习的中文医学文本 生成任务的隐私保护研究

作者姓名： 揭一新

学科专业： 网络空间安全

导师姓名： 张驰副教授

完成时间： 二〇二三年四月十四日

University of Science and Technology of China
A dissertation for master's degree



Research on privacy protection of chinese medical text generation task based on deep learning

Author: Jie Yixin

Speciality: School of Cyber Science and Technology

Supervisor: Prof. Zhang Chi

Finished time: April 14, 2023

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：_____

签字日期：_____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

控阅的学位论文在解密后也遵守此规定。

☒ 公开 ☐ 控阅（____年）

作者签名：_____

导师签名：_____

签字日期：_____

签字日期：_____

摘 要

医学文本生成任务指的是使用自然语言处理技术在医学场景下生成相关文本的任务。其中最广泛使用的是基于 Transformer 及其变体的语言模型。

然而，语言模型的参数量庞大，需要大量的训练数据集才能使得医学文本生成任务的模型性能优异，但在医学领域中，单个医疗机构很难满足如此规模的数据量要求，而且各项法律法规进一步限制了医学文本数据的收集与共享。因此如何在不泄露患者与医疗机构隐私的前提下获得足够的数据量，以训练具有高表达能力的文本生成模型是一项挑战。

此外，语言模型存在记忆问题，即模型倾向于输出训练数据，由于医学文本训练数据包含大量患者隐私信息，因此在模型训练完成后向用户提供查询服务的推断阶段，也存在泄露隐私训练数据的风险。

因此，医学文本生成任务面临两个主要挑战：一是模型需要大量训练数据以达到优异性能，但医学领域中数据规模有限且受法律法规限制；二是语言模型存在记忆问题，可能导致患者隐私信息泄露。针对上述挑战，本文先对训练阶段和推断阶段的攻击进行分类，量化隐私风险。随后分别提出相应的隐私保护策略。本文的研究内容具体如下：

首先，本文针对医学文本生成任务在训练和推断两个阶段的攻击场景进行分类，并分析了各类攻击手段对隐私安全的威胁程度。在训练阶段，考虑攻击者试图推断隐私数据并破坏训练协议的场景；在推断阶段，考虑攻击者试图恢复训练中的隐私数据的场景。这一部分研究为后续隐私保护策略的设计奠定了基础。

然后，本文对于训练阶段中，恶意攻击者试图恢复训练数据以及破坏训练协议的场景，基于可信硬件 Intel SGX 设计了一套针对医学文本生成模型的多方安全计算协议，以提升安全性假设。该协议在面对恶意攻击者的情况下仍能保证安全性，为训练医学文本生成模型提供了一种创新的隐私保护方法。

最后，本文设计了两种方式来缓解语言模型的记忆问题。一方面，在训练阶段中引入一种选择性差分隐私优化器，对模型的训练过程的损失进行加噪处理；另一方面，在推断阶段中设计了一种差分隐私解码算法，对推断结果进行加噪。通过上述两种方法有效缓解语言模型的记忆问题，防止模型的训练隐私数据被恶意攻击者获取，同时避免牺牲过多精度。

关键词：自然语言处理，医学文本生成，差分隐私，多方安全计算，深度学习隐私保护

ABSTRACT

Medical text generation tasks refer to the use of natural language processing technology to generate relevant text in the medical domain. The state-of-art models for these tasks are based on Transformer architecture and its variants.

However, language models have large parameters, which require a substantial training dataset to achieve high performance in medical text generation tasks. In the medical field, it is challenging for individual healthcare institutions to reach such data scale requirements. Furthermore, various laws and regulations further limit the collection and sharing of medical text data. Therefore, obtaining sufficient data volume without compromising patient and healthcare institution privacy to train expressive text generation models is a challenge.

Additionally, language models exhibit a memory problem, as they tend to output training data. Since medical text training data contains a considerable amount of patient privacy information, there is a risk of privacy data leakage during the inference stage when providing query services to users after model training.

Thus, medical text generation tasks face two main challenges: (1) the need for a large training dataset to achieve excellent performance, but the data scale in the medical field is limited and subject to legal restrictions, and (2) the memory problem of language models, which may lead to patient privacy information leakage. In response to these challenges, this study first classifies attacks in the training and inference stages and quantifies privacy risks. Subsequently, corresponding privacy protection strategies are proposed. The research content of this research is as follows:

First, this paper classifies attack scenarios in the training and inference stages of medical text generation tasks and analyzes the threat levels of various attack methods to security. In the training stage, the scenario of an attacker trying to infer private data and undermine the training protocol is considered; in the inference stage, the scenario of an attacker trying to recover private data from the training is considered. This research lays the foundation for the design of subsequent privacy protection strategies.

Secondly, in the training stage, where a malicious attacker attempts to recover training data and undermine the training protocol, this paper designs a secure multi-party computation protocol for medical text generation models based on trusted hardware Intel SGX, enhancing security assumptions. This protocol can still guarantee security in the face of malicious attackers, providing an innovative privacy protection method for

training medical text generation models.

Lastly, this paper designs two approaches to mitigate the memory problem of language models. On the one hand, a selective differential privacy optimizer is introduced in the training stage to add noise to the loss of the model's training process; on the other hand, a differential privacy decoding algorithm is designed in the inference stage to add noise to the inference results. Through these two methods, the memory problem of language models is effectively alleviated, preventing malicious attackers from obtaining the model's training privacy data while avoiding excessive loss of accuracy.

Key Words: Natural Language Processing, Medical Text Generation, Differential Privacy, Secure Multi-Party Computation, Privacy-preserving Deep Learning

目 录

| | |
|-------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 研究背景和意义 | 1 |
| 1.1.1 研究背景 | 1 |
| 1.1.2 研究意义 | 2 |
| 1.2 研究现状 | 3 |
| 1.2.1 语言模型的记忆问题 | 4 |
| 1.2.2 基于多方安全计算的医学文本生成任务 | 5 |
| 1.2.3 基于差分隐私的医学文本生成任务 | 6 |
| 1.3 研究内容与创新点 | 7 |
| 1.3.1 研究内容 | 7 |
| 1.3.2 创新点 | 9 |
| 1.4 论文组织结构 | 9 |
| 第 2 章 论文相关基础知识 | 11 |
| 2.1 基于深度学习的自然语言处理 | 11 |
| 2.1.1 自然语言处理的形式化定义 | 11 |
| 2.1.2 自然语言处理任务介绍 | 11 |
| 2.1.3 常见的自然语言处理模型结构 | 12 |
| 2.2 多方安全计算 | 13 |
| 2.2.1 多方安全计算定义 | 13 |
| 2.2.2 多方安全计算的安全性 | 14 |
| 2.3 差分隐私 | 15 |
| 2.3.1 差分隐私定义与性质 | 16 |
| 2.3.2 常见的差分隐私实现 | 17 |
| 2.4 可信硬件 Intel SGX | 19 |
| 第 3 章 医学文本生成任务的隐私攻击模型研究 | 22 |
| 3.1 引言 | 22 |
| 3.2 语言模型的生成过程 | 22 |
| 3.2.1 分词阶段 | 22 |
| 3.2.2 生成嵌入表示 | 23 |
| 3.2.3 编码过程 | 24 |
| 3.2.4 解码过程 | 25 |

| | |
|---------------------------|----|
| 3.3 语言模型的记忆问题 | 26 |
| 3.3.1 训练样本推断攻击 | 27 |
| 3.3.2 改进的攻击策略 | 29 |
| 3.4 训练阶段推断隐私数据以及破坏训练协议的攻击 | 31 |
| 3.4.1 场景描述与安全假设 | 31 |
| 3.4.2 攻击方式与攻击效果 | 32 |
| 3.5 推断阶段恢复训练隐私数据的攻击 | 32 |
| 3.5.1 场景描述与安全假设 | 33 |
| 3.5.2 攻击方式与攻击效果 | 33 |
| 3.6 本章小结 | 36 |
| 第4章 医学文本生成任务训练阶段的隐私保护研究 | 37 |
| 4.1 引言 | 37 |
| 4.2 模型与设计目标 | 37 |
| 4.2.1 系统模型 | 37 |
| 4.2.2 威胁模型 | 38 |
| 4.2.3 设计目标 | 38 |
| 4.3 训练协议设计 | 39 |
| 4.3.1 多方安全计算深度学习函数的实现 | 39 |
| 4.3.2 语言模型模块的构建 | 42 |
| 4.3.3 可验证外包计算的设计 | 44 |
| 4.4 安全性分析 | 45 |
| 4.4.1 训练安全 | 45 |
| 4.4.2 SGX 被攻破的影响 | 49 |
| 4.5 实验评估 | 50 |
| 4.5.1 实验设置 | 50 |
| 4.5.2 实验结果 | 51 |
| 4.6 本章小结 | 52 |
| 第5章 医学文本生成任务推断阶段的隐私保护研究 | 54 |
| 5.1 引言 | 54 |
| 5.2 系统模型与设计目标 | 54 |
| 5.2.1 系统模型 | 54 |
| 5.2.2 威胁模型与设计目标 | 55 |
| 5.3 基于差分隐私算法的推断结果隐私保护方案 | 55 |
| 5.3.1 选择差分隐私定义 | 55 |

| | |
|-------------------------|----|
| 5.3.2 针对训练阶段的选择差分隐私优化器 | 56 |
| 5.3.3 针对推断阶段的选择差分隐私解码算法 | 59 |
| 5.4 安全性分析 | 60 |
| 5.5 实验评估 | 64 |
| 5.5.1 攻击方式 | 64 |
| 5.5.2 实验设置 | 66 |
| 5.5.3 实验结果 | 66 |
| 5.6 本章小结 | 69 |
| 第 6 章 总结与展望 | 71 |
| 6.1 工作总结 | 71 |
| 6.2 未来展望 | 71 |
| 参考文献 | 73 |
| 致谢 | 81 |
| 在读期间发表的学术论文与取得的研究成果 | 82 |

插图清单

| | | |
|-------|---|----|
| 图 1.1 | 本文的研究点以及主要研究内容 | 8 |
| 图 2.1 | 差分隐私算法示意图 | 16 |
| 图 2.2 | SGX 应用程序划分与执行流程 | 20 |
| 图 2.3 | Enclave 的隔离执行 | 20 |
| 图 3.1 | Transformer 模型结构 ^[4] | 24 |
| 图 3.2 | 使用中文词表的 GPT2-small 模型 | 28 |
| 图 3.3 | 针对公开语言模型攻击的结果 | 28 |
| 图 3.4 | 检验攻击结果的正确性 | 29 |
| 图 3.5 | CMDD 数据集 | 33 |
| 图 3.6 | 训练过程中的交叉熵损失与训练步数的关系 | 35 |
| 图 3.7 | 使用 CMDD 数据集微调 GPT2-Chinese 的 Loss 随 Epoch 的变化 | 36 |
| 图 3.8 | 模型恢复出训练数据中一个样本的前 40 个 Tokens | 36 |
| 图 4.1 | 系统概述示意图 | 38 |
| 图 4.2 | 各函数的调用关系 | 39 |
| 图 4.3 | 外包 GPU 加速计算的协议流程 | 44 |
| 图 4.4 | 等效模型和原模型的训练损失与轮数的变化情况 | 52 |
| 图 5.1 | 公布模型查询接口的风险 | 54 |
| 图 5.2 | 差分隐私训练优化器 | 56 |
| 图 5.3 | 差分隐私解码算法 | 59 |
| 图 5.4 | Canary 插入攻击与衡量 | 65 |
| 图 5.5 | 各模型的 Canary 暴露度 | 68 |

表 格 清 单

| | | |
|-------|----------------------------------|----|
| 表 3.1 | 实验环境 | 27 |
| 表 3.2 | 前缀攻击结果 | 29 |
| 表 3.3 | 改进的前缀攻击结果 | 31 |
| 表 3.4 | 攻击方式与成功次数 | 35 |
| 表 4.1 | 等效模型训练的实验环境 | 51 |
| 表 4.2 | 协议的开销分析与实验环境 | 51 |
| 表 4.3 | Transformer 模块的通信开销 | 51 |
| 表 4.4 | 使用可验证外包计算的加速效果与输入维度的关系 | 52 |
| 表 5.1 | 不同前缀下 LM 的 Canary 暴露度 | 67 |
| 表 5.2 | 不同方式下的模型困惑度 | 68 |
| 表 5.3 | 攻击方式与成功次数 | 69 |

第1章 绪 论

本章首先阐述了医学文本生成任务的研究背景和研究意义，并针对现有研究工作的现状进行介绍，从而引出本文的研究内容与创新点，最后介绍本文的组织结构。

1.1 研究背景和意义

本节首先介绍医学文本生成任务的基本概念，并对其研究背景和研究意义进行详细说明。

1.1.1 研究背景

医学文本生成（Medical Text Generation）任务是一种针对电子病历（Electronic Health Record, EHR）和临床记录（Clinical Notes）等医学内容的自然语言处理（Natural Language Processing, NLP）技术。近年来，医学文本生成在电子病历生成^[1]、临床记录生成^[2]以及生成式摘要^[3]等领域得到了广泛应用。医学文本生成任务旨在利用自然语言生成技术自动生成具有医学背景的自然语言文本，如病历报告、症状描述和医疗建议等。这些文本需要准确表达医学术语、疾病、症状、药物等复杂医学概念，同时具备一定的语法和逻辑性。医学文本生成任务可以辅助医学诊断和治疗，帮助医务人员生成病历和病情报告，提高工作效率和减轻工作负担，同时避免错误和疏漏。此外，医学文本生成还有助于患者理解医学术语和治疗方案，提高患者的医疗健康素养。

Transformer^[4]及其各种变体^[5-7]的语言模型（Language Model, LM）在医学文本生成任务上的表现优于其他模型。然而，这类模型通常具有庞大的参数量，达到千亿规模。在医学文本生成任务中，考虑到病历和记录中包含大量患者隐私信息，满足如此规模的数据量对单个医疗机构来说是难以实现的。同时，受到患者隐私需求及各国法律法规^{①②}的限制，数据的收集和共享面临更大的挑战。因此，在保护患者和医疗机构隐私的前提下，训练文本生成模型并提供服务成为了一项重要的挑战。此外，语言模型面临着记忆问题，即语言模型容易记住训练数据中出现的特定内容，并会在推断阶段输出训练数据的部分内容。在医学文本生成任务下，语言模型的使用可能会导致患者隐私泄露的风险。因此，解决语言模型记忆问题亦成为一项紧迫任务。

基于语言模型的医学文本生成任务的隐私保护研究主要关注两种安全问题：

^①欧盟的《通用数据保护条例》

^②中国的《关于印发国家健康医疗大数据标准、安全和服务管理办法（试行）的通知》

一方面是为了防止执行学习训练与推断算法的计算方访问数据或模型，通常采用安全多方计算（Multi-party Secure Computation, MPC）或同态加密算法实现；另一方面是为了防止模型使用者从模型的推断结果恢复出部分训练数据，如根据模型参数或模型推断结果判断特定数据是否在训练集中或重建部分训练集，通常采用差分隐私（Differential Privacy, DP）技术。

在训练阶段，隐私数据面临直接威胁。由于法律法规限制多个拥有患者隐私文本数据的医疗机构之间的数据收集与共享，现有的研究主要针对两种训练方式，即联邦学习^[8]（Federated Learning, FL）方式与集中式学习方式。为保护训练数据以及模型参数的隐私，研究者将差分隐私^[9]、多方安全计算^[10]、同态加密^[11]与可信执行环境^[12]（Trusted Execution Environment, TEE）等隐私保护技术与联邦学习或者集中式学习相结合。这两种学习方式均可使各数据持有者充分利用各方数据进行协同训练。然而，在 NLP 任务中，特别是医学文本生成任务中，各数据持有者（医疗机构）的 EHR 文本类型数据通常包含该数据持有者的一些独特特征（如医生写作风格和数据持有者对原始数据的转换整理方式），导致各数据持有者的数据非独立同分布。在这种情况下，联邦学习方式的训练精度较低^[13-15]。相较于联邦学习方式，集中学习方式训练的模型精度较高，但同时面临的隐私风险也会更高，与此同时，当前针对集中式学习的隐私保护研究面临着安全假设弱的问题。因此，在医学文本生成任务的训练阶段，探究如何保护隐私训练数据以及模型参数具有重要意义。

在推断阶段，语言模型可能会在无意中泄露训练数据。当前多项研究^[16-19]证实了语言模型具有记忆性。记忆性是指模型容易记住训练数据中出现的特定内容，即语言模型可能会推断阶段输出训练数据的部分内容。在给定特定前缀（Prefix）时，语言模型甚至可以逐字逐句地生成原始训练样本。攻击者可以大量枚举多种前缀，对语言模型执行模型反演攻击来恢复训练数据中的隐私信息。在医学文本生成场景下，前缀的搜索空间更小，更容易定制，如“医生给我开的”和“现在只是稍有点咳嗽”等前缀容易联想到，攻击者的攻击效率和成功率也会比其他场景更高。因此，如何在医学文本生成的语言模型进行推断时有效防止隐私数据泄露，成为当前亟待解决的重要问题。

此外，目前开源的中文预训练模型资源有限。现有的预训练模型^[5-6]的训练语料中英文占比在 80% 以上，迁移学习效果不理想^[20-21]。为验证医学文本生成任务的语言模型记忆问题，还需针对医学文本数据进行微调训练。

1.1.2 研究意义

当前已有的相关研究面临诸多挑战。首先，在训练阶段，现有采用联邦学习的研究，由于医学文本数据的非独立同分布性质导致训练精度较低^[13-15]。而使

用集中式学习方式的工作面临的隐私风险更高，目前针对集中式学习的隐私保护研究面临着安全假设弱的问题^[22-25]。其次，在推断阶段，现有研究采用差分隐私（Differential Privacy, DP）技术^[9]来保护隐私数据。然而，对所有数据进行DP处理会导致模型精度降低^[26]。针对上述问题，本研究的主要意义如下：

- 本文针对医学文本生成任务在训练和推断两个阶段的攻击场景进行分类，并分析了各类攻击手段对隐私安全的威胁程度。在训练阶段，考虑攻击者试图推断隐私数据并破坏训练协议的场景；在推断阶段，考虑攻击者试图恢复训练中的隐私数据的场景。这一部分研究为后续隐私保护策略的设计奠定了基础。
- 本文借助可信硬件 Intel SGX 设计了一套针对医学文本生成模型的多方安全计算协议，以提升安全性假设。该协议在面对恶意攻击者的情况下仍能保证安全性，为训练医学文本生成模型提供了一种创新的隐私保护方法。
- 本文基于差分隐私技术，针对训练阶段提出了一种差分隐私优化器；针对推断阶段设计了一个差分隐私解码算法。通过上述两种方法缓解语言模型的记忆问题，防止模型的训练隐私数据被攻击者获取，同时避免牺牲过多的精度。

本研究旨在提供一种针对医学文本生成任务的隐私保护方法，通过对训练阶段和推断阶段的隐私保护技术的探讨与改进，为后续研究和实际应用提供了重要的参考和借鉴。本文针对训练阶段中存在恶意参与者的问题，引入多方安全计算与可信硬件 Intel SGX，并设计了一个安全的语言模型训练协议来保护训练数据以及模型的安全；同时为有效缓解语言模型的记忆问题，防止恶意攻击者通过模型反演攻击来获取训练隐私数据，本文引入差分隐私，有效的缓解了语言模型的记忆问题。

本研究的深入分析和实践将有助于提高医学文本生成模型的隐私保护水平，降低数据泄露风险，保障患者隐私，同时不影响模型性能。此外，本研究的成果对于其他涉及敏感数据的自然语言处理任务也具有一定的参考价值，有助于推动隐私保护技术在自然语言处理领域的进一步发展。

1.2 研究现状

本节首先介绍语言模型的记忆问题及其带来的隐私风险，随后介绍基于多方安全计算的医学文本生成任务的研究状况，最后分析基于差分隐私的医学文本生成任务的研究进展。

1.2.1 语言模型的记忆问题

随着深度学习技术的快速发展，大型语言模型在自然语言处理领域取得了重要突破。例如，GPT^[6-7,27]系列和 BERT^[5]模型已在多个任务上实现了超过人类的性能，近期的 ChatGPT 以及 GPT4^[27]的表现效果更为惊人，似乎有着与人类相同的表达能力。然而，随着模型规模的扩大，其在学习过程中对训练数据的记忆问题引起了人们的关注。具体而言，LM 的记忆问题是指 LM 在输出时更倾向于输入在训练样本中出现过的内容，在一定条件下甚至可以逐字逐句地生成完整的训练语句。研究表明，这些 LM 可能会在生成结果中泄露训练数据中的敏感信息。

例如，Gehman 等人^[17]调查了预训练 LM 在生成带有种族主义、性别歧视或其他有害语言方面存在的问题。作者调查了预训练语言模型在受到特定提示时生成有害语言的程度，并研究了可控文本生成算法在防止生成这种有害语言方面的效果。为了找出这种持续的有害退化的潜在原因，作者对用于预训练多个语言模型（包括 GPT-2）的两个网络文本语料库进行了分析，并发现了大量的攻击性、事实不可靠和其他有害内容。这些发现证明了语言模型具有记忆性。Nicholas 等人^[16]研究了私有数据集上训练的大型语言模型可能存在的训练数据提取攻击。攻击者可以通过查询语言模型来恢复单个训练样本。作者在 GPT-2 模型上演示了这种攻击，该模型是基于互联网公开的数据进行训练的。他们成功地从模型的训练数据中提取了数百个逐字文本序列。尽管上述序列中的每一个只出现在训练数据的一个文档中，但攻击仍然可行。作者全面评估了提取攻击，以了解导致其成功的因素。这些发现表明，更大的模型比较小的模型更易受到攻击。Zhang 等人^[18]探讨了现代神经语言模型在训练过程中可能会记忆敏感信息的问题。作者提出了反事实记忆的概念，它描述了在省略训练过程中的某个特定文档的情况下，模型预测的变化。通过在标准文本数据集中识别和研究反事实记忆的训练样本，作者进一步估计了每个训练样本对验证集和生成文本的影响。Brown 等人^[19]探讨了自然语言隐私问题的广泛性。作者指出，语言模型往往会记忆训练集中存在的短语，而攻击者可以利用这种倾向来提取训练数据，违反隐私期望。作者讨论了常见数据保护技术（数据清理和差分隐私）所做的狭隘假设与自然语言和隐私作为社会规范的广泛性之间不匹配的问题，认为现有的保护方法不能为语言模型提供通用且有意义的隐私保护。Mireshghallah 等人^[28]研究了大型语言模型微调过程中不同微调方法（对整个模型、模型头部和适配器进行微调）的记忆风险，并使用成员推断和提取攻击进行实验。作者的研究表明微调模型头部的风险最高，而微调较小的适配器则不容易受到已知提取攻击的影响。这对于“预训练和微调”范式的应用具有重要意义。

为了解决语言模型的记忆问题,研究者已经开始探索在训练过程中防止模型记忆敏感信息的方法。其中一种方法是改进模型结构,例如引入注意力机制或使用门控循环单元等,这些改进有助于提高模型处理长序列的能力,从而减少对敏感信息的依赖。另一种方法是优化训练过程,例如使用数据增强、对抗训练和知识蒸馏等技术,这些方法可以提高模型的泛化能力,缓解模型对训练数据的过拟合问题。

因此,语言模型的记忆问题给隐私保护带来了挑战。鉴于此,本文在第3章详细概述了医学文本生成任务在训练和推断阶段所面临的各种攻击,并在第5章基于差分隐私原理设计了隐私保护方法,以缓解语言模型的记忆问题。

1.2.2 基于多方安全计算的医学文本生成任务

本部分首先介绍基于多方安全计算的深度学习隐私保护技术,然后阐述目前用于医学 NLP 任务上的多方安全计算的相关工作。

基于多方安全计算的隐私保护深度学习旨在模拟存在可信第三方的计算过程,使得计算过程不会泄露隐私信息,但是其无法为保护模型参数与推断结果提供隐私保护。最经典的隐私保护机器学习框架是由 Mohassel 和 Zhang 设计的 SecureML^[29]。该方案是基于两个服务器 Client/Server 的使用秘密共享的外包计算模型。其中,服务器从多个客户端接收以秘密份额形式的数据,并基于 SPDZ 框架^[30]与混淆电路对秘密份额数据进行隐私训练和预测。与 SPDZ 框架类似,为保证训练与推断的及时响应,SecureML 在离线阶段执行复杂运算,从而使得在线阶段处理效率更高。基于 Wu 等人^[31]通过量化方法减少隐私保护训练阶段中的精度损失, Agrawal 等^[32]在半诚实攻击者的场景下,设计了 QUOTIENT 协议。该协议使用加法秘密共享机制和 Yao 混淆电路,实现了随机梯度下降算法和自适应梯度下降算法这两种深度学习模型的训练算法。董业等人^[33]使用秘密共享机制和 Top-K 梯度筛选方法来保护训练阶段中梯度信息的隐私,并且能够验证服务器聚合结果的有效性。Patra 等人^[34]基于布尔域、算数域和 Yao 混淆电路域上的混合秘密共享机制设计了 ABY 2.0,其实现了半诚实攻击者场景下的线性回归和卷积网络的训练。进一步地, Mohassel^[35]等人提出的 ABY3 在 ABY1^[36]的框架上将原算法进行了扩展,使用三方计算场景进行训练和评估。其可以在布尔域、算数域和 Yao 混淆电路域之间高效地切换,并可以在半诚实攻击者场景下用在训练线性回归、逻辑回归和神经网络模型上。Wagh 等人^[23]的算法也基于三方服务器,但其第三方服务器并不与秘密份额交互,其仅用于提供运算辅助。Rachuri 等人^[37]基于结合 ABY3 和第三方服务器,首次实现了在恶意攻击者场景下的安全训练。

目前基于 MPC 用于 NLP 领域的主要工作是 Feng 等人^[22]提出的隐私保护

系统 SecureNLP, 其重点针对用于神经机器翻译的基于循环神经网络 (Recurrent Neural Network, RNN) 的序列到序列注意力模型。具体来说, 针对 Sigmoid 和 tanh 等非线性函数, 作者使用 MPC 设计了两个高效的分布式协议, 用于在 SecureNLP 中执行各自的任務。作者还证明了这两个协议 (即隐私保护的长短时记忆网络 PrivLSTM 和隐私保护的序列到序列转换 PrivSEQ2SEQ) 在半诚实攻击者模型下是安全的, 即任何诚实但好奇的攻击者不能从他们从其他方接收到的消息中了解到任何其他信息。

然而, RNN 的表达能力存在局限性, 要实现更高的表达能力, 需要采用基于 Transformer 结构的语言模型。因此, 在本文的第 4 章中, 我们基于多方安全计算设计了一套适用于 Transformer 结构模型的协议, 以满足更高表达能力的需求。

1.2.3 基于差分隐私的医学文本生成任务

与上述结构相同, 本部分首先介绍基于差分隐私的机器学习隐私保护技术, 然后阐述目前用于医学 NLP 任务上的差分隐私的相关工作。

差分隐私技术通过在数据输入、算法迭代或算法输出中添加随机噪声来抵抗隶属关系猜测攻击。例如 Dwork 等人^[9]进行特征分解之前, 在协方差矩阵中加入对称高斯噪声矩阵, 使得输出结果是一个差分隐私的投影矩阵。Hardt 和 Price^[38]则在算法每次迭代中添加高斯噪声, 而协方差矩阵保持无扰动状态, 实现一个差分隐私的主成分分析。Abadi^[39]等人则是在随机梯度下降算法的迭代中引入高斯噪声, 实现了深度学习的数据隐私。Wu 等人^[31]通过对输出数据添加噪声扰动, 实现了具有隐私保护的线性回归和决策向量机等二分类模型。以上工作均为数据持有者独立训练采用的隐私保护方法。为实现多源数据集的隐私聚合训练, Papernot^[40]等人提出, 首先利用互相独立的数据集学习教师模型, 然后使用这些教师模型对公共数据进行带噪声的预测, 最后将预测结果迁移到学生模型的构建。该方法中隐私损失取决于学生模型训练期间向教师模型提出的查询次数, 与最终投入实用的学生模型的查询次数无关。梁文娟等人面向数据流 Top-K 频繁模式挖掘, 基于差分隐私, 设计了动态发布过程的数据隐私保护方案。史鼎元^[41]等人则基于略图数据结构和差分隐私技术, 解决排序学习中的交叉特征生成和缺失标签处理问题。

目前基于 DP 用于 NLP 领域的工作有很多。Bombari 等人^[42]探讨了训练数据集中实体之间的关系被记忆, 即可能在使用训练好的模型进行问答时产生隐私问题的問題。作者提出了关系记忆的概念, 并规范化了关系隐私的概念。同时, 作者还提出了差分关系隐私的潜在定义, 用于描述和计算模型中 RM 的量的界限。Zhao 等人^[43]提出了一种名为机密去重训练的保密调整训练方法, 用于

训练语言生成模型时保护隐私信息。该方法借鉴了差分隐私的思想,通过随机化部分训练过程来防止意外记忆,并证明了大致正确的筛选策略可以增强保密性保证。实验结果表明,机密去重训练方法在保持强大保密性的同时,能够获得几乎相同的困惑度。Wu 等人^[44]提出了自适应差分隐私框架来保护语言模型不泄露隐私信息,避免了传统差分隐私在所有数据点上的不加区分保护所带来的实用性问题。与之前需要先验隐私信息的方法不同,自适应差分隐私基于语言模型估算文本项含有隐私信息的概率,并根据该概率调整注入差分隐私噪声的程度。本文还提出了一种新的 Adam 算法来实现这一目标。实验结果表明,自适应差分隐私框架能够有效地提高差分隐私语言模型的保护能力,防止 Canary 攻击。Shi 等人^[45]提出了部分差分隐私,以提供对敏感数据的严格隐私保护,以提高模型效用。作者开发了一个相应的隐私机制 DPSGD 优化器,用于基于 RNN 的语言模型。

总体而言,基于差分隐私的医学文本生成任务的相关研究主要关注在保护隐私的前提下如何提升模型的性能和效率,以及如何运用新的隐私保护理念和算法来应对现有方法的局限性。因此,本文第5章针对训练和推断阶段分别设计了基于差分隐私的训练隐私优化器以及解码算法,以满足这些需求。

1.3 研究内容与创新点

本节对本文的研究内容进行介绍,并总结创新之处。

1.3.1 研究内容

医学文本生成任务相对于普通文本生成任务而言,对表达能力和逻辑性有更高的要求,因此普通文本通用的 RNN 模型不适用于医学文本任务。本文选择参数量大、表达能力强的 Transformer 模型解决上述问题。同时,本文也考虑了 Transformer 模型带来的记忆问题。综上所述,本文首先分析基于 Transformer 模型的医学文本生成任务在训练和推断阶段分别存在的隐私问题,再分别针对训练和推断阶段存在的隐私风险提出相应的隐私保护方案。

本文的主要研究内容总体框架如图1.1所示。

(1) 医学文本生成任务的隐私攻击模型研究

本部分主要关注医学文本生成任务在训练和推断阶段的隐私泄露风险,以证实后续提出的隐私保护方法的重要性。首先,本研究内容从语言模型的生成过程出发,详细阐述了如何为自然语言文本建模并生成后续文本,为后续分析医学文本生成模型的训练与推断阶段的执行过程奠定基础。接着,本研究内容分析了语言模型的记忆问题,并针对公开的预训练模型实施模型反演攻击。同时,本研

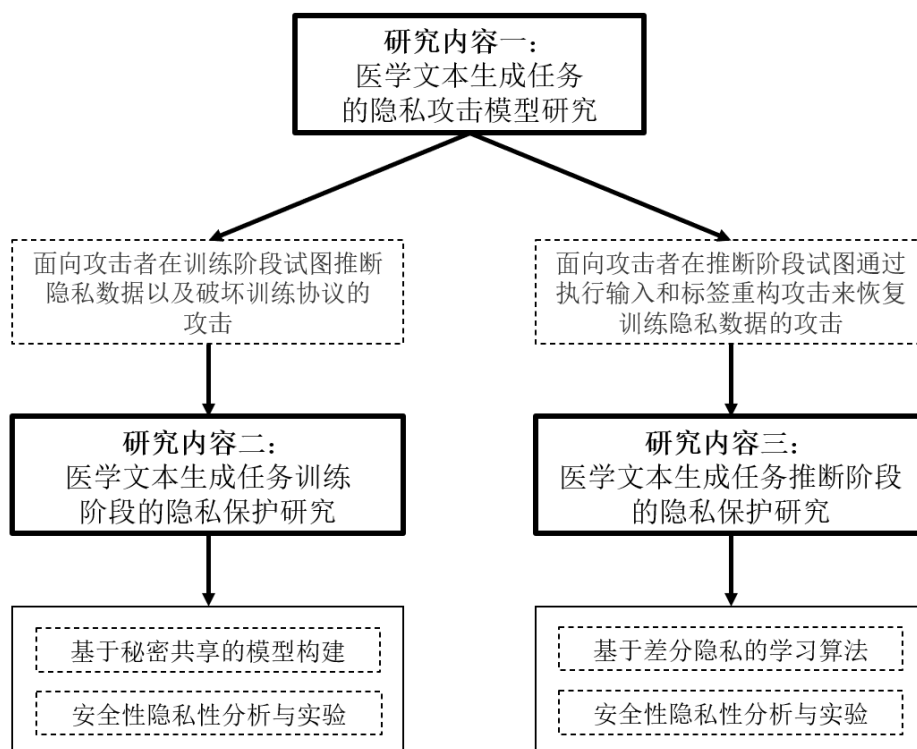


图 1.1 本文的研究点以及主要研究内容

究内容还提出了一些改进的攻击策略，以提高攻击成功率。此外，本研究内容探讨了攻击者在医学文本生成任务的训练阶段可能采用的推断隐私数据和破坏训练协议的攻击手段。最后，从医学文本生成任务推断阶段的攻击以及上述改进攻击手段出发，通过对医学文本数据下训练的语言模型实施攻击，分析攻击效果，展示了语言模型记忆问题带来的隐私挑战。

（2）医学文本生成任务训练阶段的隐私保护研究

本部分主要关注医疗文本生成任务训练阶段的隐私保护问题。首先，本研究内容明确了系统模型和威胁模型，并设计了安全目标。然后，本研究内容扩展了基于秘密共享的协议，使其能够构建复杂的 Transformer 结构。通过多方计算手段来保障数据机密性，同时利用可信硬件 Intel SGX 确保执行过程的完整性。为提高协议的执行效率，本研究内容设计了一个可验证的外包计算方法。接着，本研究内容分析了该协议的安全性，证明了协议满足设计目标。最后，本研究内容通过实验验证了协议的有效性和高效性。

（3）医学文本生成任务推断阶段的隐私保护研究

为阻止攻击者在推断阶段实施模型反演攻击以恢复训练隐私数据，同时保持语言模型的性能效果，本部分基于差分隐私提出两种缓解医学文本生成任务语言模型的技术。首先，本研究内容介绍系统模型与设计目标，阐明保护对象和攻击者的行为。其次，本研究内容介绍选择差分隐私的定义，并针对训练与推断阶段分别设计了隐私优化器与解码算法，作为两种提供选择差分隐私的方式。然

后,对前述设计的隐私优化器与解码算法进行隐私性分析,证明其满足差分隐私的定义。最后,通过实验设计说明选择差分隐私以及这两种保护方法的优势。

1.3.2 创新点

本文的主要创新点如下:

- 1) 本文针对医学文本生成任务的训练与推断阶段所面临的攻击进行介绍和分类。针对公开的中文预训练模型执行模型反演攻击,并改进攻击手段。针对医学文本生成任务的语言模型执行模型反演攻击,并在上述改进的攻击手段下验证其记忆性。
- 2) 本文基于多方安全计算与可信硬件 Intel SGX,针对医学文本生成任务的训练阶段,提出了一种可抵御恶意攻击者的隐私保护协议,并通过设计一个外包计算协议提升其执行效率。
- 3) 本文基于差分隐私,缓解医学文本生成任务的语言模型的记忆性,一方面,在训练阶段中引入一种选择性差分隐私优化器;另一方面,在推断阶段中设计了一种差分隐私解码算法。

1.4 论文组织结构

第1章为绪论,介绍了本文的研究背景和意义、相关研究现状、研究内容与创新点、论文组织结构等内容。

第2章为基础知识介绍,介绍深度学习在自然语言处理任务中的应用,多方安全计算和差分隐私这两种在深度学习隐私保护中广泛使用的隐私保护技术,以及可信硬件 Intel SGX。

第3章为医学文本生成任务的隐私攻击模型研究,探讨医学文本生成任务在训练和推断阶段的隐私泄露风险,并分析攻击者在训练阶段推断隐私数据和破坏训练协议的攻击,以及推断阶段攻击者执行的模型反演攻击。同时展示了语言模型记忆问题带来的隐私挑战。

第4章为医学文本生成任务训练阶段的隐私保护研究。该章节设计了基于秘密共享的多方计算协议来保障数据机密性,并使用可信硬件保证执行过程的完整性。该协议扩展了基于秘密共享的协议,使得可以构建复杂的 Transformer 语言模型结构,并分析其安全性,验证其有效性和高效性。

第5章为医学文本生成任务推断阶段的隐私保护研究。该章节提出了一个基于差分隐私的新隐私保护算法——选择差分隐私算法。该章节分别针对训练与推断阶段分别设计了隐私优化器与解码算法,并分析证明了它们的安全性。最后,通过设计实验说明了这两种保护方法的优势。

第 6 章为总结与展望，总结了本文的研究工作，提出了未来的研究方向，并指出了本文所提出的方法在实际应用中的潜在意义和应用前景。

第2章 论文相关基础知识

本章将介绍深度学习隐私保护研究的基础知识。具体来说，本章将首先介绍深度学习在自然语言处理任务中的应用，包括任务的定义、常见的模型结构等内容。然后，介绍多方安全计算和差分隐私这两种在深度学习隐私保护中广泛使用的隐私保护技术，包括其定义、性质和常见的实现方式。最后，本文将介绍可信硬件 Intel SGX，它可以为深度学习隐私保护提供一个安全可靠的执行环境。通过本章的介绍，读者可以全面了解本文所涉及的技术基础，并为后续章节的研究工作做好准备。

2.1 基于深度学习的自然语言处理

本节部分将介绍 NLP 的形式化定义、常见的 NLP 任务以及常见的深度学习模型结构。

2.1.1 自然语言处理的形式化定义

自然语言处理是一门语言学、计算机科学与人工智能领域的交叉学科，旨在让计算机理解、分析、生成和处理自然语言文本数据。在 NLP 中，文本被视为离散符号序列，记为 $S = (w_1, w_2, \dots, w_n)$ ，其中 w_i 表示文本中的第 i 个单词或标点符号。

在 NLP 中，语言模型是一种基本的概率模型，用于预测给定文本序列的下一个单词或字符。语言模型可以表示为 $p(w_i | w_1, w_2, \dots, w_{i-1})$ ，即在已知前 $i-1$ 个单词的情况下，预测第 i 个单词的概率。其中， $p(w_i | w_1, w_2, \dots, w_{i-1})$ 可以通过训练数据集中的单词出现频率来计算。在具体实现时，通常采用神经网络模型来实现语言模型，如循环神经网络、长短时记忆网络（Long Short-Term Memory Network, LSTM）和 Transformer 等。

除了语言模型，NLP 中还有许多其他的模型和算法，这些模型和算法能够帮助处理自然语言文本数据，从而实现各种 NLP 任务。

2.1.2 自然语言处理任务介绍

自然语言处理是人工智能领域的一个重要分支，它致力于让计算机理解、解释和生成人类语言。基于深度学习的自然语言处理技术在过去几年取得了显著的进展，推动了一系列 NLP 任务的发展。以下是一些主要的 NLP 任务介绍。

1) 词法分析。词法分析是 NLP 的基本任务之一，主要包括分词、词性标注和

命名实体识别。分词是将句子划分成有意义的词汇单元的过程。词性标注是为每个词汇单元分配一个词性标签，如名词、动词、形容词等。命名实体识别（NER）是识别文本中具有特定意义的实体，如人名、地名、机构名等。

- 2) 信息抽取。信息抽取旨在从非结构化文本中抽取结构化信息，主要任务包括关系抽取、事件抽取和实体链接。关系抽取是识别文本中命名实体之间的语义关系。事件抽取是识别文本中描述的事件及其相关实体。实体链接是将文本中的命名实体与知识库中的实体进行链接。
- 3) 机器翻译。机器翻译是将一种自然语言文本自动翻译成另一种自然语言文本的任务。机器翻译可以分为统计机器翻译（Statistical Machine Translation, SMT）、神经机器翻译（Neural Machine Translation, NMT）和基于预训练模型的翻译方法。SMT 依赖于大量双语语料库，利用统计方法进行翻译建模。NMT 使用神经网络，特别是编码器-解码器结构进行端到端的翻译建模。预训练模型如 BERT^[5]和 GPT^[6-7]等通过在大量单语和双语语料库上预训练，进一步提升了翻译性能。
- 4) 文本摘要。文本摘要任务是生成一个简洁、连贯且包含原始文本主要信息的摘要。文本摘要可以分为抽取式摘要和生成式摘要。抽取式摘要是从原文中选择关键句子或短语构成摘要，而生成式摘要则是生成新的句子来表达原文的核心信息。深度学习技术在生成式摘要方面取得了显著进展，如使用编码器-解码器结构和预训练模型进行摘要生成。。
- 5) 语言生成。语言生成是自然语言处理的一个核心任务，它的目标是根据给定的输入生成符合语法、语义和篇章逻辑的自然语言文本。近年来，预训练语言模型（如 GPT 系列模型）在各种生成任务中取得了显著的成果，包括文本生成、对话生成、文本摘要等。预训练语言模型通过在大量文本数据上进行无监督训练，学习到了丰富的语言知识和生成能力。
- 6) 知识图谱。知识图谱是一种结构化的知识表示方法，它以图的形式表示实体及其之间的关系。知识图谱在自然语言处理中的应用包括实体链接、关系抽取、知识图谱补全等。深度学习技术在知识图谱领域的应用主要包括基于图神经网络的表示学习方法，以及将知识图谱与预训练语言模型相结合的方法。

2.1.3 常见的自然语言处理模型结构

在深度学习模型中，常见的用于 NLP 任务的模型结构包括循环神经网络、长短期记忆网络、门控循环单元网络（Gated Recurrent Unit Network, GRU）和 Transformer。RNN 和 LSTM 是最早被应用于 NLP 任务的深度学习模型，它们的

优势在于可以处理变长序列数据。GRU 是对 LSTM 的改进，能够在保持同样效果的情况下减少计算量。而 Transformer 网络则采用了注意力机制，能够更好地处理长距离依赖关系，因此在机器翻译任务中表现非常出色。

这些模型的共同特点是它们都是端到端 (end-to-end) 的模型，其输入数据是一段文本序列，输出则是一个对应的文本序列。在 NLP 任务中，这些模型通常被用于生成文本，例如文本摘要、对话系统和机器翻译等任务。在本文中，我们将研究如何在保护训练数据和生成的文本隐私的同时，使用这些深度学习模型来生成医学文本。

2.2 多方安全计算

2.2.1 多方安全计算定义

多方安全计算技术的研究主要是针对无可信第三方的情况下，多个参与者如何在不暴露己方数据的情况下安全地计算一个约定函数的问题。多方安全计算协议分为信息论安全和密码学安全。如果协议对于拥有无限计算能力的攻击者来说是安全的，那么它被称为信息论安全或无条件安全。如果协议只对拥有多项式计算能力的攻击者是安全的，则称为密码学安全或条件安全。多方安全计算协议的目的是实现一些特定的计算任务。多方安全计算的目的是实现一些特定的计算任务，例如比较、加密、解密、排序、计数、求和等。在多方安全计算中，每个参与方持有自己的输入数据，并且希望在不泄露私有数据的情况下与其他方共同计算结果。

多方安全计算具有以下性质：

- 1) 正确性。采用多方安全计算协议计算的结果与直接使用明文数据计算的结果相同。具体而言，针对输入为 x ，运算法则为 Q 的一个多方安全计算协议 M ，有

$$P[M(x, f) = Q(x)] = 1$$

假设有 n 个参与者，分别为 P_1, P_2, \dots, P_n ，他们各自持有输入 x_1, x_2, \dots, x_n ，并希望在不泄露私密输入的情况下计算某个函数 $f(x_1, x_2, \dots, x_n)$ 。多方安全计算协议 (MPC) 是指在参与者之间进行的一系列通信和计算过程，使得每个参与者最终可以获得函数 f 的输出，同时不会泄露任何其他参与者的输入。

- 2) 隐私性。任何多方安全计算协议参与方不能获得除协议规定内的额外信息。具体而言，所有参与方 $P = (p_1, p_2, \dots, p_n)$ ，对于参与者 $p_k \in P$ ，使用参与者数据 $F = (f_1, f_2, \dots, f_n)$ 输出结果为 L 的多方安全计算协议 M ，存在一

个可忽略的函数 ϵ ，有下式成立：

$$P_{p_k}(f_{i,i \neq k} | L = M(F)) + P_k(f_k) \leq P(f_{i,i \neq k}) + \epsilon$$

2.2.2 多方安全计算的安全性

在介绍了多方安全计算的定义后，为理解其安全性，下面介绍多方安全计算场景中不同的安全假设。

- 1) 半诚实 (Semi-honest) 参与者假设：在这种假设下，半诚实的参与者会遵守协议并诚实地执行计算，但他会记录协议执行的上下文来获得更多关于隐私数据的信息。尽管这种安全假设对于参与者的行为作出了相对严格的假设，但在实际场景中仍然相对较强。
- 2) 恶意 (Malicious) 参与者假设：在这种安全假设下，协议不对恶意参与者的行为作任何假设。恶意参与者可以采取偏离、中断或中止协议等手段，从而获取其他参与者的隐私输入信息，因此这种安全假设比半诚实假设更为严格。

现实-理想框架是一种多方安全计算安全性的形式化框架，其使用了一个包含所有安全需求的“理想世界”，通过现实世界与理想世界之间的对应关系定义多方案去计算。

理想世界中，参与方 $P(p_1, p_2, \dots, p_n)$ 将各自的输入 $F = (f_1, f_2, \dots, f_n)$ 发送给一个可信第三方 S ， S 执行函数 Q ，并将计算结果 $Q(F)$ 返回给所有参与方。假设存在一个在理想世界中的攻击者 A ，其可以控制除参与方其 p_k 外任意数量的参与方 $P_A \subset P$ 其中 $p_k \notin P_A$ ，它试图通过各种攻击手段获取 p_k 的输入。由于攻击者 A 无法控制可信第三方 S ，因此攻击者 A 无法获得除 $Q(F)$ 外的任何信息。

由于现实世界中不存在上述可信第三方 S ，各参与方需要相互交互以完成计算。在参与方 p_i 以其隐私输入 f_i 、当前状态的中间值 h_i 、接收到的所有信息 $Context$ 、随机数 R_i 、安全性参数 κ 作为输入时，通过执行函数 π_i ，获得新的中间值或者输出。

如果攻击者在现实世界达到的攻击效果与其在理想世界的攻击效果相同，那么可以将该协议认为是在现实世界中安全的，即协议使得其在现实世界中提供的安全性与在理想世界中提供的安全性等价。

定义 2.1 记 π 为一个协议， Q 为一个函数， P_A 为被攻击者 A 控制的参与方集合， Sim 为仿真算法，定义如下两种协议。

- $Real_{\pi}(\kappa, P_A; f_1, f_2, \dots, f_n)$ ：在安全性参数 κ 下参与方 $P = (p_1, p_2, \dots, p_n)$ 使用各自隐私数据入 $F = (f_1, f_2, \dots, f_n)$ 在真实世界执行的协议。记 $V = v_1, v_2, \dots, v_n$ 为参与方 P 的最终视图，即 $\{v_i | i \in (1, 2, \dots, n)\}$ ； $y =$

y_1, y_2, \dots, y_n 为参与方 P 的输出结果。

- $\text{Ideal}_{Q,\text{Sim}}(\kappa, P_A; f_1, f_2, \dots, f_n)$: 模拟世界中在安全性参数 κ 下参与方 $P(p_1, p_2, \dots, p_n)$ 使用各自隐私数据输入 $F = (f_1, f_2, \dots, f_n)$ 在仿真算法 Sim 与函数 Q 的计算下执行的协议，其输出为 $\text{Sim}(P_A, \{(f_i, y_i) \in P_A\})$

在参与方是半诚实的假设下，其会诚实地执行协议，但可能会从接收到的信息中尝试推断更多的信息。

定义 2.2 给定协议 π ， Q 为一个函数， P_A 为被攻击者 A 控制的参与方集合，如果存在一个为仿真算法 Sim ，对于攻击者 A 控制的所有参与方集合 P_A 以及所有的输入 $F = (f_1, f_2, \dots, f_n)$ ，概率分布 $\text{Real}_\pi(\kappa, P_A; f_1, f_2, \dots, f_n)$ 与 $\text{Ideal}_{Q,\text{Sim}}(\kappa, P_A; f_1, f_2, \dots, f_n)$ 在 κ 下不可区分，则称此协议在半诚实攻击者的攻击下安全地实现了函数 Q 。

在参与方是恶意的假设下，其不仅会从接收到的信息中尝试推断更多的信息，而且还可以任意破坏、偏离协议。这里使用 A 表示攻击程序，用 $\text{Corr}(A)$ 表示被现实世界中的攻击者攻陷的参与方集合，用 $\text{Corr}(\text{Sim})$ 表示被理想世界中的攻击者 Sim 攻陷的参与方集合。与上述半诚实安全性的方式类似，定义现实世界与理想世界下的概率分布：

- $\text{Real}_{\pi,A}(\kappa; \{x_i | i \notin \text{Corr}(A)\})$: 在安全性参数 κ 下，未被攻陷的参与方城市地使用给定的隐私输入执行协议，而被攻陷的参与方 $\text{Corr}(A)$ 的信息可以由 A 进行篡改，并参与到后续的协议执行过程中。记 v_i 为参与方的最终视图，即 $\{v_i | i \in \text{Corr}(A)\}$ ； y_i 为诚实参与方 p_i 的输出结果，即 $\{y_i | i \notin \text{Corr}(A)\}$ 。
- $\text{Ideal}_{Q,\text{Sim}}(\kappa; \{x_i | i \notin \text{Corr}(A)\})$: 模拟世界中，在安全性参数 κ 下，在包含攻陷方输入 $\{x_i | i \in \text{Corr}(A)\}$ 的数据上执行 Sim 协议。计算 $(y_1, \dots, y_n) \leftarrow Q(f_1, \dots, f_n)$ 后，将 $\{y_i | i \notin \text{Corr}(A)\}$ 发送给 Sim 。记 V^* 为 Sim 最终输出，即各参与方的仿真视图集合。

定义 2.3 给定协议 π 与函数 Q ，对任意一个现实世界中的攻击者 A ，存在一个满足 $\text{Corr}(A) \iff \text{Corr}(\text{Sim})$ 的仿真者 Sim ，使得对于诚实参与方的所有输入 $\{x_i | i \notin \text{Corr}(A)\}$ ，概率分布 $\text{Real}_{\pi,A}(\kappa; \{x_i | i \notin \text{Corr}(A)\})$ 与 $\text{Ideal}_{Q,\text{Sim}}(\kappa; \{x_i | i \notin \text{Corr}(A)\})$ 在 κ 下不可区分，则称此协议在恶意攻击者的攻击下安全地实现了函数 Q 。

2.3 差分隐私

差分隐私 (Differential Privacy)^[9] 是一种保护隐私的方法，其基本思想是在不泄露个体信息的前提下，通过添加一定的噪声，使得查询结果的公开不会泄露个体的隐私信息。

2.3.1 差分隐私定义与性质

差分隐私是一种保护个人隐私的技术，它可以在不泄露个人隐私的前提下对数据进行分析和处理。具体来说，如图2.1所示，差分隐私通过限制单条数据对算法输出的影响来保护个人隐私，即无论数据集中是否包含特定个体，算法的输出结果不可区分。这种保护方式确保了即使公开了算法在某个特定数据集上的结果，数据集中的数据仍然受到定量的隐私保护。差分隐私的具体定义如下。

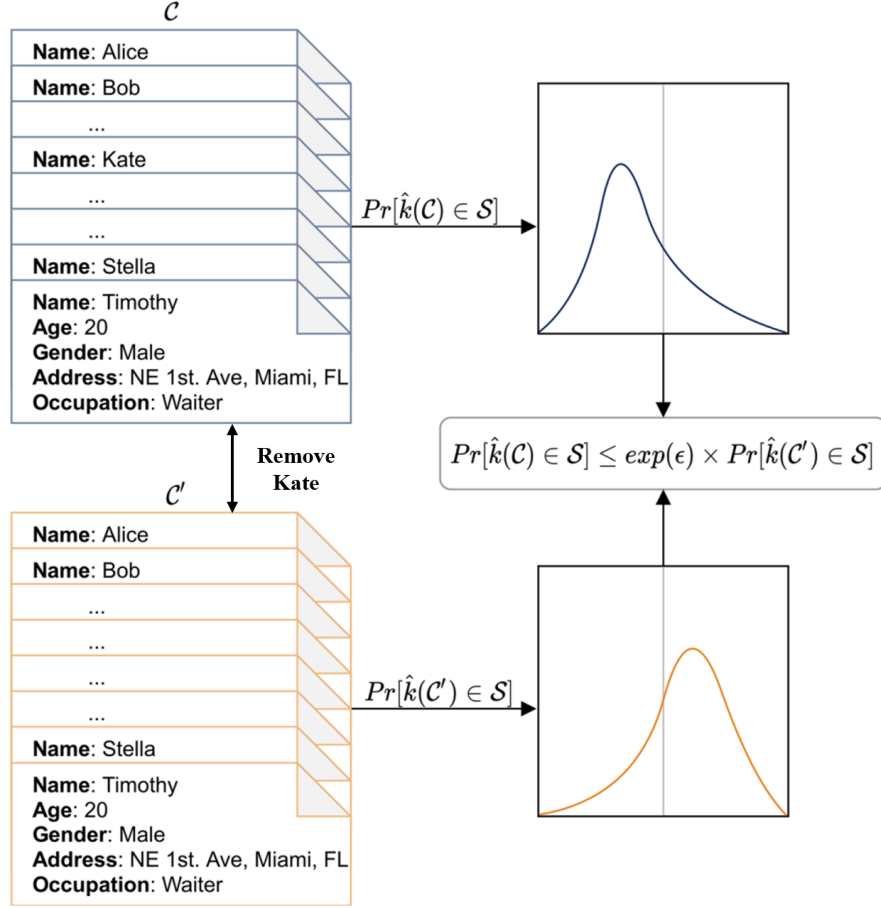


图 2.1 差分隐私算法示意图

定义 2.4 （近邻数据集^[46]）若数据集 D 与 D' 仅有一条记录不同，则称 D 与 D' 为近邻数据集

定义 2.5 (ϵ -差分隐私^[46]) 对于任意近邻数据集 D 与 D' 以及任意的输出结果 S ，若随机算法 M 满足

$$Pr[M(D) = S] \leq e^\epsilon \cdot Pr[M(D') = S] \quad (2.1)$$

则称 M 满足 ϵ -差分隐私。其中 ϵ 表示隐私预算，它是衡量隐私保护程度的指标， ϵ 越小表示算法的隐私保护程度越高，反之 ϵ 越大，算法在近邻数据集的结果之间的差别也越大，隐私保护程度便越低。

式2.1的条件在实际场景中往往过于苛刻，导致算法的可用性很差。因此，可

以适当放宽 ϵ -差分隐私的要求，即松弛一下它的边界，这样就有了 (ϵ, δ) -差分隐私。

定义 2.6 ((ϵ, δ) -差分隐私^[47]) 对于任意近邻数据集 D 与 D' 以及任意的输出结果 S ，若随机算法 M 满足

$$\Pr[M(D) = S] \leq e^\epsilon \cdot \Pr[M(D') = S] + \delta \quad (2.2)$$

则称 M 满足 (ϵ, δ) -差分隐私。其中 ϵ 同样表示隐私预算，而 δ 为一个概率，算法在 $1 - \delta$ 的概率下满足式 (2.1)，即只有至多 δ 的概率不满足式 (2.1)，这便要求 δ 值很小，在数据集样本为 n 的情况下^[48]，需要 $\delta < \frac{1}{n}$ 。 ϵ -差分隐私可以被视为 $\delta = 0$ 的 (ϵ, δ) -差分隐私。

下面给出一些形式化定义的定理。

定理 2.1 若随机算法 M_1 满足 (ϵ, δ) -差分隐私，则对于任意算法 M_2 ，有 $M_2(M_1(\cdot))$ 满足 (ϵ, δ) -差分隐私。

证明 对于 $\forall S \in \text{Range}(M_2)$ ，记 $Y = \{y | M_2(y) = S\}$ ，则

$$\begin{aligned} \Pr[M_2(M_1(D)) = S] &= \Pr[M_1(D) \in Y] \\ &\leq e^\epsilon \Pr[M_1(D') \in Y] + \delta \\ &= \Pr[M_2(M_1(D')) = S] + \delta \end{aligned}$$

■

定理 2.2 若随机算法 M_i 满足 (ϵ_i, δ_i) -差分隐私，其中 $i = \{1, 2, \dots, n\}$ ，则算法 $M(D) = (M_1(D), M_2(D), \dots, M_n(D))$ 满足 $(\sum_{i=1}^n \epsilon_i, \sum_{i=1}^n \delta_i)$ -差分隐私^[47]。

当所有算法 M_i 的 (ϵ_i, δ_i) 相同时，有如下定理成立。

定理 2.3 若随机算法 M_i 满足 (ϵ, δ) -差分隐私，其中 $i = \{1, 2, \dots, n\}$ ，则算法 $M(D) = (M_1(D), M_2(D), \dots, M_n(D))$ 满足 $(\epsilon', n\delta + \delta')$ -差分隐私^[47]。其中

$$\epsilon' = \sqrt{2n \ln\left(\frac{1}{\delta'}\right) + n\epsilon(e^\epsilon - 1)}$$

2.3.2 常见的差分隐私实现

差分隐私的实现方法有很多，包括基于噪声的加噪算法、基于随机投影的扰动算法、基于数据扰动的数据变形算法等。其中，添加噪声的加噪算法应用最为广泛，具体包括以下几种方法：拉普拉斯机制^[49]、高斯机制^[47]与指数机制^[48]。下面首先介绍两种全局敏感度的概念，然后一次介绍上述机制的概念。

定义 2.7 (L1 全局敏感度^[47]) 对于近邻数据集 D 与 D' ，函数 f 的 L1 全局敏感度 $\Delta_1 f$ 的定义为：

$$\Delta_1 f = \max_{D, D'} \|f(D) - f(D')\|_1 \quad (2.3)$$

定义 2.8 (L2 全局敏感度^[47]) 对于近邻数据集 D 与 D' , 函数 f 的 L2 全局敏感度 $\Delta_2 f$ 的定义为:

$$\Delta_2 f = \max_{D, D'} \|f(D) - f(D')\|_2 \quad (2.4)$$

(1) 拉普拉斯机制

拉普拉斯机制是一种添加噪声的差分隐私算法, 它的基本思想是为每个查询结果添加一个服从拉普拉斯分布的噪声, 从而保证查询结果的隐私性。

定义 2.9 (拉普拉斯分布) 均值为 0, 方差为 d 的拉普拉斯概率密度函数为:

$$\text{Laplace}(x, d) = \frac{1}{2d} \exp\left(-\frac{|x|}{d}\right) \quad (2.5)$$

定理 2.4 (拉普拉斯机制^[50]) 对于任意函数 f , 其 L1 全局敏感度为 $\Delta_1 f$, 算法 $M = f(D) + \text{Laplace}(\frac{\Delta_1 f}{\epsilon})$ 满足 ϵ -差分隐私。

(2) 高斯机制

高斯机制是一种添加噪声的差分隐私算法, 它的基本思想是为每个查询结果添加一个服从高斯分布的噪声, 从而保证查询结果的隐私性。与拉普拉斯机制不同的是, 高斯机制引入的噪声值是连续的, 并且其分布更加平滑。

定义 2.10 (高斯分布) 均值为 μ , 方差为 σ 高斯分布的概率密度函数为:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.6)$$

定理 2.5 (高斯机制^[47]) 对于任意函数 f , 其 L2 全局敏感度为 $\Delta_2 f$, 加上高斯分布噪声的算法 $M = f(D) + N(0, \Delta_2 f \sigma^2)$ 满足 (ϵ, δ) -差分隐私。其中 $\epsilon < 1$, 且

$$\delta \geq \frac{4}{5} \exp\left(-\frac{(\sigma\epsilon)^2}{2}\right)$$

(3) 指数机制

指数机制是一种添加噪声的差分隐私算法, 它的基本思想是将查询结果的隐私性与其“有用性”相平衡, 从而选择最优的查询结果。具体地, 指数机制会对每个查询结果赋予一个得分值, 得分越高表示该结果越有用, 然后根据指数分布的概率密度函数从所有结果中以概率 p_i 选择结果 i , 其中

$$p_i = \frac{\exp(\epsilon s_i)}{\sum_{j=1}^n \exp(\epsilon s_j)}$$

其中, s_i 为结果 i 的得分值, ϵ 为差分隐私参数, 控制查询结果的隐私性和“有用性”之间的平衡。

定理 2.6 (指数机制^[51]) 可用性函数 u 对算法 M 在数据集 D 上的任何输出数值 y 给出一个可用性评估值 $u(D, y) \in \mathbb{R}$, 若算法 M 输出结果 y 的概率正比于 $\exp(\frac{\epsilon u(D, y)}{2\delta u})$, 即

$$Pr[M(D) = y] = \frac{\exp(\frac{\epsilon u(D, y)}{2\delta u})}{\sum_{y' \in Y} \exp(\frac{\epsilon u(D, y')}{2\delta u})} \quad (2.7)$$

其中 $\Delta u = \max_{\{y, D, D'\}} |u(D, y) - u(D', y)|$, 则算法 M 满足 ϵ -差分隐私。

2.4 可信硬件 Intel SGX

可信执行环境 (Trusted Execution Environment, TEE) 是一种在计算机系统中创建的隔离环境, 它提供了比普通操作系统更高的安全性和可信度。可信执行环境的目的是保护系统中的敏感数据和代码, 以及提供隔离环境来执行受保护的计算, 防止恶意软件和攻击者对系统进行攻击和窃取信息。在可信执行环境中, 所有的敏感数据和代码都可以得到保护。TEE 提供了一个独立的内存空间, 其中的代码和数据与主机操作系统完全隔离, 从而可以保护这些数据和代码不受到主机操作系统和其他应用程序的干扰。TEE 还提供了安全的输入和输出通道, 使得敏感数据能够安全地进入和离开可信执行环境。在可信执行环境中运行的应用程序必须经过验证和授权才能被允许运行, 这样可以防止不受信任的应用程序进入可信执行环境。同时, 可信执行环境还提供了追踪和审计功能, 用于记录 TEE 中的所有活动, 这可以提供重要的证据来追踪和调查安全事件。

Intel SGX (Software Guard Extensions) 是英特尔公司推出的一种可信计算技术, 其目的是提供一种安全的硬件环境, 以保护计算机系统中的敏感数据和代码。Intel SGX 是一种硬件扩展, 它通过在处理器中引入特定的安全硬件, 创建一种安全的执行环境, 可以在这个环境中运行代码和数据, 以实现可信计算。Intel SGX 在保护数据和代码时, 采用了一种称为“隔离执行”的技术, 该技术可以保证数据和代码在安全的执行环境中被隔离开来, 从而防止非授权的访问和修改。同时, Intel SGX 还提供了一种特殊的机制, 称为飞地 (Enclave), 用于实现在执行环境中运行的应用程序的安全性保护。如图2.2所示, 执行程序被划分为可信与不可信部分, 为保护数据隐私, 不可信代码通过调用 `ecall` 来使 Enclave 执行可信代码③, 可信代码执行完成后通过调用 `ocall` 来返回不可信部分⑤。

(1) 隔离执行

SGX 安全体系结构保证 Enclave 与其他运行在 Enclave 之外的软件隔离执行, 包括操作系统。通过隔离, 可以保证 Enclave 控制流的完整性, 并且被执行的 Enclave 的内部机密数据不会被任何对手观察到。隔离是通过处理器执行的保护机制实现的。Enclave 的代码和数据存储在称为 EPC(Enclave Page Cache) 的硬

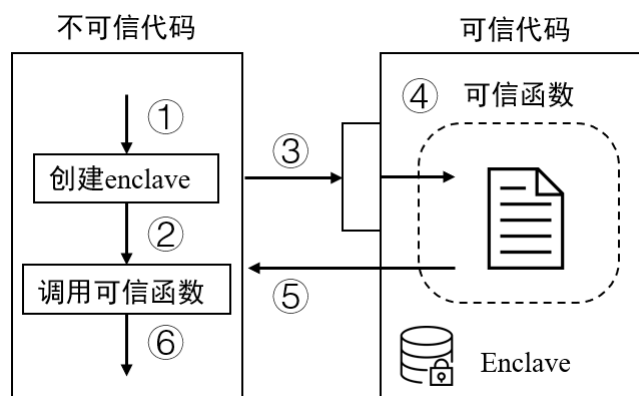


图 2.2 SGX 应用程序划分与执行流程

件保护内存区域中，该内存驻留在处理机保留内存 (Processor ReservedMemory, PRM) 中，如图2.3所示，其中内存加密引擎 (MEE, Memory Encryption Engine) 对 Enclave 中的数据进行加解密，在数据写入内存时加密，在读入 Enclave 时解密。PRM 是动态随机存储器 (Dynamic RandomAccess Memory) 的一个子集, 操作系统、应用程序或直接内存访问器都不能访问它。PRM 保护是基于处理器中的一系列内存访问检查，非 Enclave 软件仅允许访问 PRM 范围之外的内存区域，而 Enclave 代码可以访问非 PRM 内存和 Enclave 拥有的 EPC 页面。

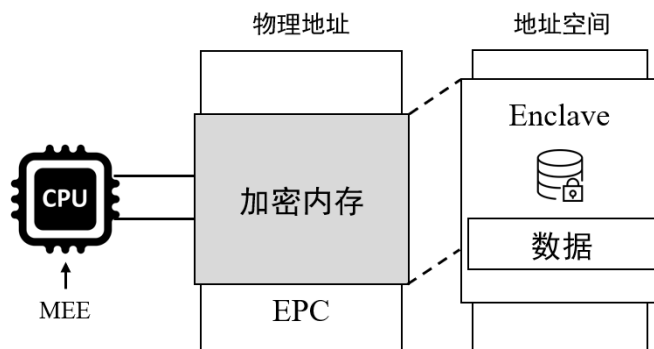


图 2.3 Enclave 的隔离执行

(2) 远程认证

在构建 Enclave 时，Enclave 的测度 (MeasuRement of Enclave, MRE)，即 Enclave 初始代码和数据的安全哈希值，由处理器生成，该哈希值保证 Enclave 的完整性。SGX 提供了两种类型的身份认证方式：一种是平台内部 Enclave 间的认证，称为本地认证 (Local Attestation)；另一种是平台间的认证，称为远程认证 (Remote Attestation)。被认证的 Enclave 通过调用 EREPORT 指令生成报告 (Report Structure)，EREPORT 是 SGX 提供的一个硬件指令，只有运行在 SGX 平台上的合法 Enclave 才可以调用并生成报告，其中会包含该 Enclave 的测度，其完整性由基于 Intel SGX 硬件产生的密钥保护。

每个 SGX 处理器都有一个根密封密钥 (Root Seal Key, RSK)，其在制造过

程中嵌入，用于生成与 Enclave 身份绑定的密封密钥 (Seal Key)。SGX 有一个称为密封的过程，是使用密封密钥加密和验证 Enclave 数据以便持久存储。Enclave 可以使用 EGETKEY 指令从 RSK 派生出密封密钥，并且该密封密钥具有唯一性，即与同一平台的不同 Enclave 或不同平台的任何 Enclave 都不同。

第3章 医学文本生成任务的隐私攻击模型研究

3.1 引言

为了更好的说明本文所提出的医学文本生成任务的隐私保护机制的必要性,本章将详细阐述医学文本生成任务训练的隐私泄露风险以及医学文本生成任务推断阶段的隐私泄露问题。

首先,本章从语言模型的生成过程开始介绍,这一部分阐述了语言模型是如何为自然语言文本建模并生成后续文本的,为后面引入在医学文本生成模型的训练与推断阶段的执行过程做了铺垫。其次,本章针对 LM 的记忆问题进行分析,对公开的预训练模型展开攻击,并且提出了几种改进的攻击策略。随后,本章分析了攻击者在训练阶段试图推断隐私数据以及破坏训练协议的攻击。最后,本章研究了面向攻击者在推断阶段试图通过执行模型反演攻击来恢复训练隐私数据的攻击,以此来探讨医学文本生成任务在训练推断阶段所面临的攻击,并通过实验来针对在医学文本数据下训练的 LM 攻击,说明了 LM 记忆问题带来的隐私挑战。

3.2 语言模型的生成过程

本部分介绍深度学习 NLP 语言模型的生成过程,通过从输入数据到输出结果的完整流程来解释前沿 NLP 深度学习模型的构成。

3.2.1 分词阶段

对于本文数据, NLP 使用分词器 (Tokenizer) 将文本按照出现频率的方式切分成独立的词符 (Token), 词符可以是符号、字母、子词、词或者是短语, 比如“我在学习深度学习知识”可以切分成[“我”, “在”, “学习”, “深度学习”, “知识”], “The courtyard is thronged with visitors”可以切分成[“The”, “court”, “yard”, “is”, “th”, “ron”, “ge”, “d”, “with”, “vi”, “sit”, “or”, “s”]。所有词符构成的集合称为词表 (Vocabulary), 其由数据集整体构建。每一个数据持有者拥有一个由 N 个句子 (或句对) 构成的文本数据集 $D = \{S_1, S_2, \dots, S_N\}$, 其中每个句子由 $L_i (i = (1, 2, \dots, N))$ 个 Token 构成, 且 $S_i = \{t_{i1}, t_{i2}, \dots, t_{iL_i}\}$ 。记由数据集 D 建立的大小为 $|V|$ 的词表为 V , 则 $t_{ij} \in V$ 。

词表为一个映射, 可以将正整数与字符形式的词符相互转换。假设词表中有“147: ‘啊’”的键值对, 即意味着在分词器处理文本时遇到‘啊’这个字符会把它转换为整数 147, 同样的, 如果模型输出的 logits 中数值最大的 index 是 147, 即

模型输出的下一个 Token 就是 147，分词器在转换后就会输出 147 所对应的字符‘啊’。在不同的切分方式下，相同的字符可能表示成不同的数。此外，词表一般会有一些特殊字符，比如表示开始的符号“<BOS>”、表示结束的符号“<EOS>”与表示填充的符号“<PAD>”。

NLP 模型的输入是由词表所定义的分词器对句子进行处理后，以整数形式存储的标（Tokens）序列，通过字符数字转换过程实现了文本内容的量化表示。具体的切分 Token 的方式有很多，分词的形式也多种多样，从最开始的字切分，词切分，发展到更细粒度的 BPE^[52]，以及跨语言的 sentencepiece^①等的切分方法。上述不同细粒度的分词方法是由输入的词表大小以及各个字符出现的频率综合决定的。

在确定了分词方法并将原始的语料通过该方法切分后，得到了词表。分词器会根据词表把数据集的文本内容转换成一个正整数构成的数组，这个过程称为 Tokenizer 的编码过程，同样，正整数组成的数组也可以根据词表由分词器转换成字符，这个过程称为 Tokenizer 的解码过程。

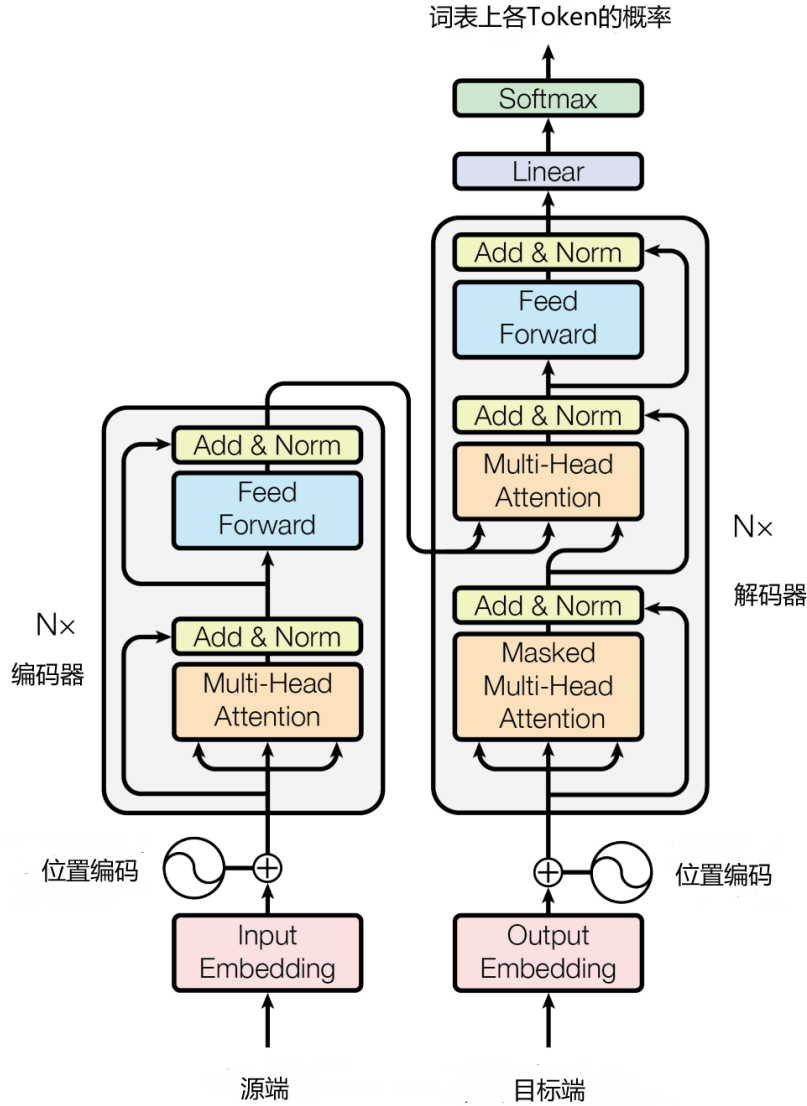
3.2.2 生成嵌入表示

为了让字符形式的文本可以让计算机处理，可以通过上述的分词阶段把文本形式的 Token 转换成词表上可以对应的正整数表示。但是一个正整数是不能表征整个 Token 的信息，更不能概况整个语义信息。为了丰富表示，可以将每一个 Token 都映射到一个高维向量，通过更复杂的表示带来更好的表达能力，这种高维向量称为 Word Embedding。一般来说，后续要介绍的模型隐层表示维度 hidden_dim 与 Word Embedding 的维度（后续都简称记为 hidden_dim）相同，目前最好的模型的 hidden_dim 通常在 1024 以上。

虽然上述的 Word Embedding 的表达能力提升了，但由于其是固定的，无法更新，因此无法处理这个 Token 在不同语义中的情况。比如，“苹果”这个词（在不同的 Tokenizer 处理下可能是不同的 Tokens）在“我想吃苹果”中表示水果含义，而在“我新买了一个苹果电脑”中表示一个公司名称的含义。因此，需要根据同序列的其他 Token 的信息实时更新 Word Embedding 的表示。这里称更新后的 Word Embedding 为“动态” Word Embedding。

提取动态 Word Embedding 的方式主要是基于 Transformer^[4]模型，以及在其基础上进行改进的各种变体^[5-7]。如图3.1所示，模型 Transformer^[4]的结构由两部分组件构成：编码器（Encoder）与解码器（Decoder）。下面两节分别对 Encoder 与 Decoder 进行介绍。

^①<https://github.com/google/sentencepiece>

图 3.1 Transformer 模型结构^[4]

3.2.3 编码过程

由于同一个 Token 在不同位置表达的含义与对整体语义的影响也是不同的, 因此只使用 Word Embedding 表示的信息不够充分, 需要引入位置信息, 便有了位置编码嵌入 (Position Embedding)。本文把 Word Embedding 与 Position Embedding 的结合称为输入嵌入 (Input Embedding), 作为接下来介绍的编码器的输入。

编码器: 编码器由 N 个完全相同的层堆叠而成。每一层都有两个子层构成, 其中第一个子层是一个多路注意力 (Multi-Head Attention, MHA) 网络, 第二子层是一个简单的、位置完全连接的前馈网络 (Feed-Forward Network, FFN)。本文对每个子层进行计算的时候, 都先经过一个残差连接层 (Residual Connection), 接着进行层标准化 (Layer Normalization, LN) 也就是说, 每个子层的输出是 $\text{LN}(x + \text{Sublayer}(x))$, 其中 $\text{Sublayer}(x)$ 是由子层本身实现的函数。为了保证这些残差

连接，模型中的所有子层以及嵌入层产生的输出维度都相同。

注意力机制的基本思路是构建一个映射函数来从一组键-值（Key-Value）对中检索出和给定查询（Query）相关的信息，其中查询、键和值都用向量表示。而注意力网络的输出则为值的加权求和，其中分配给每个值的权重由查询与相应键的兼容性函数计算所得。在 Transformer 模型中使用了一种特殊的注意力网络结构，称为缩放点积注意力（Scaled Dot-Product Attention）网络。假设输入的查询 Q 和键 K 的维度为 d ，值 V 的维度为 d ，那么注意力网络的整个过程是先计算查询 Q 和每个键 K 的点乘操作，并除以 \sqrt{d} ，然后应用 Softmax 函数计算权重，最终通过加权求和得到最终的输出：

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d}}\right) \cdot V$$

3.2.4 解码过程

编码器输出的结果是解码器输入的一部分，解码器另外的输入是与 Input Embedding 生成方式相同的 Output Embedding。解码器每次输出的是目标端各 Token 的“动态”Embedding。取最后一个 Token 的“动态”Embedding，经过一个线性层输出一个大小为词表大小的浮点数数组，记为 Logits，它表示模型输出的词表上每一个词作为下一个 Token 的可能性，通过 Softmax 函数转换为 0 至 1 之间的值以表征概率，选择概率最大值所对应的下标 index 来作为输出，这里的下标值是与词表对应的非负整数，其可以通过 Tokenzier 解码为具体的字符。下面介绍解码器的构成。

解码器：解码器同样由 N 个完全相同的层堆叠而成。除了编码器中用到的两个子层之外，解码器还插入了第三个子层，该层对编码器的输出结果执行多路注意力机制，从而来获取代表源语言端的上下文的向量表示。与编码器类似，Transformer 在每个子层先采用残差连接层，然后进行层标准化。由于解码器需要去拟合一个条件语言模型，因此需要在解码器的多路注意力子层中添加一个掩码，以防止每个目标言端的词会直接用到其后面位置的词语信息。这种掩码将输出嵌入偏移一个位置，确保对位置为 i 的字的概率预测只能依赖位置小于 i 的已知输出。

解码器每次输出一个 Token，当检测到输出为词表上表示结束的符号“<EOS>”对应的数值时，解码器停止流程，意味着当前解码任务完成，将上面所有解码出来的整数数组通过 Tokenizer 进行解码，即根据词表从数字映射到字符，便得到了最终的文本输出。

3.3 语言模型的记忆问题

随着深度学习技术的迅速发展,大型语言模型在自然语言处理领域取得了重要突破。例如,GPT 系列^[6-7]和 BERT^[5]模型已经在多个任务上取得了超越人类的表现。然而,随着模型规模的扩大,其在学习过程中对训练数据的记忆问题引起了关注。研究表明^[16],这些语言模型可能会在生成结果中泄露训练数据中的敏感信息。下面给出 LM 记忆问题的定义。

定义 3.1 模型知识抽取: 如果存在前缀 c , 使得下面的式子成立, 则称字符串 s 是可从 LM (即 f_θ) 中提取的:

$$s \leftarrow \underset{s': |s'|=N}{\operatorname{argmax}} f_\theta(s'|c)$$

在这里用 $f_\theta(s'|c)$ 来表示整个序列为 s' 的可能性。由于在大规模的数据集上计算最可能的序列 s 是非常困难的, 定义 3.1 中的 argmax 可以用一个适当的抽样策略 (例如贪心采样) 来替代, 该策略反映了模型 f_θ 在实际应用中生成文本。

定义 3.2 k -清晰记忆: 如果 s 是可以被从 LM (f_θ) 中提取到的, 且 s 在训练数据 X 中最多出现 $k \geq 1$ 个例子: $|x \in X : s \subseteq x| \leq k$, 那么一个字符串 s 是 k -清晰记忆。

这个定义的关键是“示例”的含义。对于 GPT-2, 每个完整网页被用作一个训练示例。由于该定义考虑的是包含给定字符串的不同训练示例的数量, 而非该字符串出现的总次数, 因此一个字符串在同一页中可能出现多次, 但仍被视为 $k = 1$ 的记忆。

由于 LM 是概率生成模型, 本文遵循之前的工作, 并使用一种自然的似然度量——困惑度, 来评估 LM “预测”序列中 Tokens 的好坏程度。具体地说,

定义 3.3 困惑度 (Perplexity): 对于一个 Tokens 序列: (x_1, \dots, x_n) , 其困惑度的定义如下:

$$p = \exp\left(-\frac{1}{n} \sum_{i=1}^n \log f_\theta(x_i | (x_1, \dots, x_{i-1}))\right)$$

上式中的 $f_\theta(x_i | (x_1, \dots, x_{i-1}))$ 为在前面 (x_1, \dots, x_{i-1}) 这些 tokens 的语境下, 模型 f_θ 判定下一个 Token 为指定 Token 的概率。由于经过 Softmax 输出后的概率 $0 < p < 1$, 对其取负对数后变为 $0 < -\log p < +\infty$, 从前往后对每个 Token 计算上述结果, 并取其均值作为结果。因此, 如果困惑度较低, 则模型对生成的 Token 序列不太“惊讶”(即该序列较为连贯), 并为序列中每个 Token 分配了高概率。

3.3.1 训练样本推断攻击

1. 攻击描述与设置

本章中的训练样本推断攻击是属于攻击方式中的模型反演攻击。模型反演攻击（Model Inversion Attack）是针对机器学习模型的一种隐私攻击方法。在这种攻击中，攻击者试图通过已知的模型输出（预测结果）以及对模型的访问权限，推断出输入数据的某些敏感特征。该攻击方法侧重于特定个体的敏感信息泄露问题。

模型反演攻击通常分为白盒和黑盒两种场景。在黑盒攻击中，攻击者仅具有有限的模型访问权限，例如仅能使用模型的预测 API。攻击者可以通过探测模型的输入-输出关系，以便从模型的预测结果中提取特定用户的敏感信息。黑盒攻击通常需要攻击者具备一定的辅助信息（如输入数据的部分特征或标签信息），以便构建输入并分析输出。在白盒攻击中，攻击者可以直接访问模型的内部结构、权重和参数。这使得攻击者能够更深入地了解模型的工作原理，并更容易地提取输入数据的敏感信息。白盒攻击通常具有更高的成功率，但在实际场景中，攻击者通常很难获得模型的完整访问权限。

模型反演攻击的成因主要是模型在训练过程中学到了输入数据的某些敏感特征。这些特征可能会用于生成预测结果，从而使攻击者有机会从输出中提取这些特征。

本实验假设恶意攻击者对模型执行黑盒攻击，即攻击者只能从输入与模型的输出关系来推测隐私信息。

2. 实验设置

这里的实验环境如表3.1所示：CPU 为 AMD Ryzen 9 5900HX、32GB RAM、GPU 为 RTX3080-Laptop、操作系统为 Windows 11 64 位。

表 3.1 实验环境

| 维度 | 配置 |
|------|------------------------------|
| 处理器 | AMD Ryzen 9 5900HX @ 3.30GHz |
| 内存 | 32G DDR4 3200Hz |
| GPU | RTX3080-Laptop 16G VRAM |
| 操作系统 | Windows 11 64 位 |
| 硬盘 | 1TB SSD |

本节使用的 LM 是 Hugging Face 中的一个中文 GPT2 模型^①，其参数量为 81.9M，使用的词表大小为 21128，隐层维度为 768。模型由 12 层 GPT2Block 组成，其结构如图3.2所示。其中，wte 表示 Word Token Embedding，维度为（词表大小，隐层维度），即为词表上所有 Tokens 的 768 维嵌入表示。另一个初始 Embedding，即 wpe 表示 Word Position Embedding，它表示 Token 的位置编码，维度为（最大

^①<https://huggingface.co/uer/gpt2-chinese-cluecorpussmall>

长度，隐层维度），这里为 1024，表示该语言模型支持最多 1024 个 Token 作为输入。在编码过程中，由 Token Embedding 与 Position Embedding 组合而成的维度为（句子 Tokens 长度，768）的 Input Embedding 通过 12 层 GPT2Block，得到每个 Token 的最终 Embedding。为了输出下一个 Token，最后一个 Token 的最终 Embedding 在通过最后的 `lm_head`，将 768 维的隐层表示映射到词表大小上，以表示语言模型认为词表上每一个 Token 作为下一个 Token 的置信度。

```
GPT2LMHeadModel(
  (transformer): GPT2Model(
    (wte): Embedding(21128, 768)
    (wpe): Embedding(1024, 768)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (12x): GPT2Block(
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D()
          (c_proj): Conv1D()
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D()
          (c_proj): Conv1D()
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  )
  (lm_head): Linear(in_features=768, out_features=21128, bias=False)
)
```

图 3.2 使用中文词表的 GPT2-small 模型

3. 攻击结果

本节使用该预训练模型进行解码，其超参数设置如下：温度 $T=0.5$ ， $\text{top_k}=20$ ， $\text{repetition_penalty}=5$ 。当传入的前缀 `prefix` 为“我的手机号是 156”时，采用上述攻击方式得到的部分结果如图 3.3 所示的解码结果。

我的手机号是156051，他们还有一个qq群叫做张小姐。她说：你和老公在外面开了房子住吧？男人
我的手机号是156112，在线客服也没有回复。这时候她才意识到自己被骗了！因为现实中并不存取
我的手机号是156229，所以就把他们联系方式告诉了朋友。经过几天交流沟通后得知这个案子已有

图 3.3 针对公开语言模型攻击的结果

下面验证攻击结果的准确性。由于 LM 生成的结果可能包含随机生成的结果。随后使 LM 生成若干条结果对生成的结果进行筛选，过滤掉其中不符合手机号长度与字符的生成样本，剩下的即为可能的攻击结果。为了检验生成结果的

正确性,第一种验证方式是通过微信添加好友输入生成的结果,若可以检索到用户则为成功,这也是最为强力的验证手段;第二种方式是通过如百度、谷歌等各种搜索引擎检索,若有部分符合区域的前7位段号,则也认为该生成结果是成功的。图3.4为恢复出结果并检验成功的部分样例,具体情况如表3.2所示。

本节中的攻击方式为黑盒攻击,即攻击者只能从输入与模型的输出关系中推测隐私信息。在解码时,本节使用预训练模型生成多条结果,并从中过滤掉其中不符合手机号长度与字符的生成样本,剩余的生成结果为可能的攻击结果。本节使用两种方式验证生成结果的准确性:第一种是通过微信添加好友输入生成结果,若可以检索到用户则为成功;第二种是通过搜索引擎(如百度、谷歌等)检索,若有部分符合区域的前7位段号,则也认为该生成结果是成功的。图3.4展示了成功恢复结果并验证的部分样例,具体情况如表3.2所示。



图 3.4 检验攻击结果的正确性

表 3.2 前缀攻击结果

| 总数 | 不合规 | 无效 | 有效 |
|-----|-----|----|----|
| 100 | 47 | 24 | 29 |

该预训练模型公布于2020年,使用的公开训练数据包含了2019年中文维基百科、2016年的新闻预料、2018年百科问答等^①以及包含2005-2011年间的74万篇新闻文档的数据集^②。然而,由于该LM训练语料较早,其中使用到的部分隐私信息可能已过时或不再适用。从本实验结果可以合理推测,使用较新的隐私数据训练的模型可能会面临更大的隐私问题。值得注意的是,此类攻击不仅对公开的预训练模型构成威胁,也会对任何基于语言模型的应用系统产生影响。

3.3.2 改进的攻击策略

1. 改进的解码方式

通过过滤低概率样本进行成员推断,由于以下类型语言模型的问题,导致其精度较差:许多样本被分配了虚假的高概率。这类样本主要有两种:

^①https://github.com/brightmart/nlp_chinese_corpus

^②<http://thuctc.thunlp.org/#> 中文文本分类数据集 THUCNews

琐碎的记忆。在很多情况下，GPT-2 输出的内容是无意义的，因为文本非常常见。例如，它以高概率重复从 1 到 100 的数字。

重复的子字符串。LM 经常犯的错误是它们倾向于反复生成相同的字符串。很多没有被记忆的高概率样本确实是重复的文本（例如，“我在吃饭。我在吃饭。我在吃饭。……”）。

一个直观的改进想法是通过与第二个 LM 进行比较，过滤掉这些重复（但仍然是高概率的样本）。假设有第二个模型能够准确捕捉文本的置信度，那么它也会给这些记忆内容赋予高置信度。因此，寻找更多样化、更罕见的记忆形式的一个自然策略是，与第二个模型相比，过滤原始模型的置信度“出乎意料地高”的样本。下面本部分将讨论实现这一目标的四种方法。

(1) 与其他 LM 比较。假设有第二个 LM，它记忆的是与原 LM 不同的一组例子。实现这一目标的一个方法是在一组与 GPT-2 训练数据不相交的数据上训练模型，这种情况下两个模型记住相同的 k -清晰记忆概率较小。另一种策略是采取一个小得多的模型在相同的数据集上训练：因为小模型没有较强的记忆能力。因此本部分假设，存在 k -清晰记忆的样本，使得其被大规模的 LM 模型记住，但不被小 LM 模型记住。

(2) 与 zlib 压缩比较。实验中不必与另一个 LM 进行比较。对于给定的序列，任何能赋予序列某种感知程度概念的技术都是有用的。作为一种简单的基线方法，本部分计算文本的 zlib 熵^[53]：序列用 zlib 压缩时的熵位数。然后，本部分使用原 LM 困惑度和 zlib 熵的比值作为本部分的成员推断度量。虽然文本压缩器很简单，但它们可以识别出上面描述的无意义的重复例子（例如，它们在建模重复子字符串方面非常出色）。

(3) 滑动窗口的困惑度。有时，当样本包含一个记忆的子字符串，周围是一块非记忆（和高困惑度）的文本时，模型给出的置信度不高。为了处理这个问题，本部分取以特定长度 Tokens 为滑动窗口的最小的困惑度作为结果。

(4) 基于衰减温度的采样。如2.1.1节所述，给定之前的 Tokens 序列，LM 产生下一个 Token 的概率： $Pr(x_i|x_1, \dots, x_{i-1})$ 。在实践中，这是通过神经网络 $f_\theta(x_i|x_1, \dots, x_{i-1})$ 来得到“logit”向量 z ，然后计算 $\text{Softmax}(z)$ 得到输出概率分布。

对于 $t > 1$ ，可以通过将输出 $\text{Softmax}(z)$ 替换为 $\text{Softmax}(\frac{z}{t})$ 来人为地“压平”这个概率分布，使模型输出的置信度差距不会太大（这里， t 被称为温度）。即温度越高，模型的输出越多样化。

然而，在整个生成过程中保持较高的温度意味着，即使采样过程开始发出一个记忆的例子，它也可能随机偏离记忆输出的路径。因此，使用一种动态衰减的温度可以为模型提供了足够的时间来“探索”一组不同的前缀，同时也允许它

遵循它找到的高置信路径。

2. 攻击结果

这里实验环境与3.3.1相同：CPU为AMD Ryzen 9 5900HX、32GB RAM、GPU为RTX3080-Laptop、操作系统为Windows 11 64位。

实验结果如表3.3所示。其中另一个LM选择的是原始的GPT2模型^[6]（主要的训练语料是英文），在使用滑动窗口的困惑度解码时滑动窗口的大小设置为3；使用温度衰减时从 $t = 10$ 开始，在前8个Tokens内线性衰减到 $t = 1$ （由于最关注的输出部分是一开始的11位手机号156XXXXXXXX，需要8位输出），之后保持 $t = 1$ 。

表 3.3 改进的前缀攻击结果

| 方式 | 不合规 | 无效 | 有效 |
|-----------|-----|----|----|
| 原始生成 | 43 | 28 | 29 |
| 与其他LM比较 | 39 | 32 | 29 |
| 与Zlib压缩比较 | 41 | 31 | 28 |
| 使用滑动窗口 | 36 | 38 | 26 |
| 使用温度衰减 | 44 | 32 | 24 |

从表3.3可以看出，虽然上述改进的攻击方式的有效比例跟原始的直接生成差别不大，但是从合规的角度，即满足输出的位数是11位手机号的形式，这些攻击方式基本都比原始生成要好，这也证明了这些攻击方式可以生成更符合预期推断目的的生成结果。此外，由于人工验证的开销太大，无法执行更大规模的验证，只使用“156”作为开头以及仅100条记录的生成结果可能包含很多随机性。

3.4 训练阶段推断隐私数据以及破坏训练协议的攻击

3.4.1 场景描述与安全假设

在这种场景下，本节假设存在两类实体，一方是拥有医疗隐私数据的多个数据持有者，另一方是提供计算服务的多个计算方。这些数据持有者希望利用计算方提供的服务，协同训练一个模型。计算方在计算服务结束后，将训练好的模型分发给各个数据持有者。

本文遵循常见的安全假设^[22-23]，认为数据持有者会严格遵循协议提供本地数据，且不会试图推断其他数据持有者的隐私数据，即数据持有者是诚实的。对于计算方，本部分假设它们可能会任意偏离协议，试图从获取的数据中推断数据持有者的隐私信息以及模型参数，即计算方是恶意的。与相关研究^[25]的设定相同，这种恶意假设是合理且常见的。此外，本文假设多个恶意计算方不会共谋。

3.4.2 攻击方式与攻击效果

本部分从两个方向展开分析。首先，本部分讨论针对基于多方计算（MPC）的当前工作所面临的问题；其次，本部分分析引入可信执行环境的相关工作面临的攻击。

当前针对机器学习与深度学习领域的基于 MPC 的相关工作（如 SecureNN^[23]、SecureNLP^[22]和 ABY3^[54]等）可以在半诚实攻击者的安全假设下保障协议的安全性。然而，面对恶意攻击者，不仅可能根据获取到的内容推断更多信息，还可能在执行过程中任意偏离协议，即返回任意非正常结果^[25]。这意味着在执行过程中，恶意攻击者负责的计算部分可能会遭受破坏，从而导致整个协议的执行过程受到破坏。

为了规范计算方的行为，部分研究集中于使用硬件设备来保护数据隐私以及执行过程的安全。例如，Lee 等人^[55]提出了一种名为 Occlumency 的创新性云驱动解决方案，旨在在不影响使用强大云资源优势的前提下，保护用户隐私。Occlumency 利用 SGX 的 Enclave 在整个深度学习推理过程中保护用户数据的机密性和完整性。Hua 等人^[56]提出了一种名为 GuardNN 的安全深度神经网络加速器，它在不受信任的环境中为用户数据和模型参数提供基于硬件的保护。作者通过针对已知的 DNN 加速器内存访问模式定制片外存储器保护，将内存加密和完整性验证的开销降至最低，并在可编程门阵列上进行了原型实现，验证了该方案对推理任务提供了有效的保密性保护。Hashemi 等人^[57]提供了一个统一的大型深度神经网络的训练和推理框架，旨在在保护输入隐私和计算完整性的同时进行训练和推理。作者提出了一种名为 DarKnight 的方法，它使用一种基于矩阵掩蔽的新颖数据盲化策略，在可信执行环境中创建输入混淆。并通过分析证明了其信息论隐私保证。

尽管上述工作都将可信硬件 Intel SGX 视为一个可信第三方，即其可以提供机密性与完整性，但自 Intel 推出 SGX 以来，各种攻击接踵而至。其中最具代表性的攻击是侧信道攻击，如功耗分析攻击、计时攻击、回滚攻击与缓存冲突攻击等^[58-65]。这些攻击通常会攻破可信硬件，从而破坏其机密性与完整性。

3.5 推断阶段恢复训练隐私数据的攻击

本节通过对由原始训练数据训练得到的模型执行模型反演攻击，结果表明模型可能记忆训练数据隐私信息。

首先，本部分通过实验来说明在中文医学场景下，语言模型的记忆性问题依然突出，即尝试从公开预训练模型生成训练样本并验证成功率。其次，在医学文本生成任务场景下，本节在公开预训练模型基础上通过医学文本数据集微调语

言模型，测试针对训练好的模型的训练数据重构攻击效果。最后，本节对攻击策略进行改进，并通过实验说明这些改进可以从语言模型中抽取更多隐私训练数据。

3.5.1 场景描述与安全假设

在这种场景下，本部分假设有两类实体，一方是模型持有者，另一方是使用模型推断服务的使用方。本文假设模型持有者严格遵循协议执行推断过程，对使用方的假设是其可通过模型推断服务执行输入和标签重构攻击等手段恢复模型训练隐私数据，即模型持有者是诚实的，而使用方是恶意的。

3.5.2 攻击方式与攻击效果

与3.3.1节设定相同，本部分考虑针对模型输出结果的恢复训练隐私数据攻击，即执行模型反演攻击。本章考虑攻击者的访问权限是黑盒访问权限。

1. 实验设置

本节选择中文医疗对话数据集（Chinese medical dialogue data, CMDD）^①，包含六个部门的问答句对，如图3.5所示。

| department | title | ask | answer |
|------------|-------------|---------------------------------------|---|
| 心血管科 | 高血压患者能吃党参吗？ | 我有高血压这两天女婿来的时候给我拿了些党参泡水喝，您好高血压可以吃党参吗？ | 高血压病人可以口服党参的。党参有降血脂，降压的作用，可以彻底消除血液中的垃圾，从而对冠心病以及心血管疾病的患者都有一定的稳定预防工作作用，因此平时口服党参能远离三高的危害。另外党参除了益气养血，降低中枢神经作用，调整消化系统功能，健脾补肺的功能。感谢您的进行咨询，期望我的解释对您有所帮助。 |
| 消化科 | 哪家医院能治胃反流 | 烧心，打隔，咳嗽低烧，以有4年多 | 建议你用奥美拉唑同时，加用吗丁啉或莫沙必利或援生力维，另外还可以加用达喜片 |

| 部门 | 句对数目 |
|-----|--------|
| 男科 | 94596 |
| 内科 | 220606 |
| 妇产科 | 183751 |
| 肿瘤科 | 75553 |
| 儿科 | 101602 |
| 外科 | 115991 |
| 总计 | 792099 |

图 3.5 CMDD 数据集

本节以3.3.1节中的中文预训练模型为基础，在 CMDD 数据集上进行微调训练。实验环境与3.3.1相同。

这里实验环境与3.3.1相同：CPU 为 AMD Ryzen 9 5900HX、32GB RAM、GPU 为 RTX3080-Laptop、操作系统为 Windows 11 64 位。

本节复用了3.3.1节中公开语言模型的词表与模型结构，使用的超参数：训练轮数 epochs=25，warmup_steps=1000，学习率为 $1e-6$ ，累积梯度 gradient_accumulation=24，最大梯度裁剪 max_grad_norm=3。训练过程中使用 AdamW 优化器，并且使用 get_linear_schedule_with_warmup 的设置。

^①<https://github.com/Toyhom/Chinese-medical-dialogue-data>

2. 攻击结果

(1) 正常的生成攻击

实验过程中的每个步骤（模型每通过一个批次的数据训练更新完称为一个步骤）的训练交叉熵损失与推断的交叉熵损失都通过 TensorBoard 的 Summary-Writer^[66] 进行记录。训练过程中的损失值与步骤之间的关系如图 3.6 所示。从整体来看，训练过程中的损失在开始的 100k 个步骤时下降较明显，但后面变化不大，整体损失下降的幅度并不大（2.7 \rightarrow 1.8）。同时，损失值的波动较大，原因如下：

- 1) 损失下降不多。原始的预训练模型是通过大量各种领域的语料训练收敛的，因此在所有中文语义场景的表达已经有了较好的结果，即在 CMDD 医学对话场景下效果也很好。
- 2) 损失值震荡。由于原始的预训练模型的训练集没有包含很多专门的医学场景语料，因此在医学场景下，如专业名词出现频率低、描述方式较正式、反馈方式独特等特点，模型在某些特殊场景下预测下一个 Token 时可能与实际偏差很大，即 label Token 的概率很低而损失很大。但是大部分对话的逻辑与预训练模型训练过的模式很接近，即大部分预测的比较好。综上所述，在该特定的医学领域对预训练模型进行微调时，损失值的震荡现象属于正常现象。

在不同的 Epoch 下，模型在训练集与验证集上的交叉熵损失函数 CrossEntropyLoss 如图 3.7 所示。从图 3.7 中可以看出，在前 15 个 Epoch 中，训练集的损失值稳步下降，后面趋于平稳；而验证集的损失值在前 10 个 Epoch 中稳步下降，后面近乎收敛。在第 6 个 Epoch 时，二者数值近似相等，之后模型更倾向于学习训练集上的数据分布特点，即训练集的损失值要比验证集的损失值低。从训练过程的损失情况来看，本节采用第 10 个 Epoch 的模型参数作为结果，以分析语言模型的记忆性。

(2) 改进的生成攻击

本实验使用随机采样的 10 个训练数据的前 20 个 Token 作为前缀输入，采用 3.3.1 中的方式分别进行解码，测试其完整恢复训练数据的次数。具体来说，对于每个前缀，本节对上述每个前缀生成 10000 个解码结果，针对其进行平均统计（平均值为分数时向下取整）。

例如，当输入前缀为“我弟弟在的那个补习班”（这是训练数据中出现的样本中的前 20 个 Tokens），设置解码长度为 20，即让训练的 LM 生成接下来的 20 个 Tokens。图 3.8 为成功恢复出训练样本的一个实例。

本实验共产生 10000 个生成样本。将这 10000 个生成结果的前 20 个 Tokens 与原样本的采样后的 20 个 Tokens 进行比较，如果全部相同，则成功恢复次数加

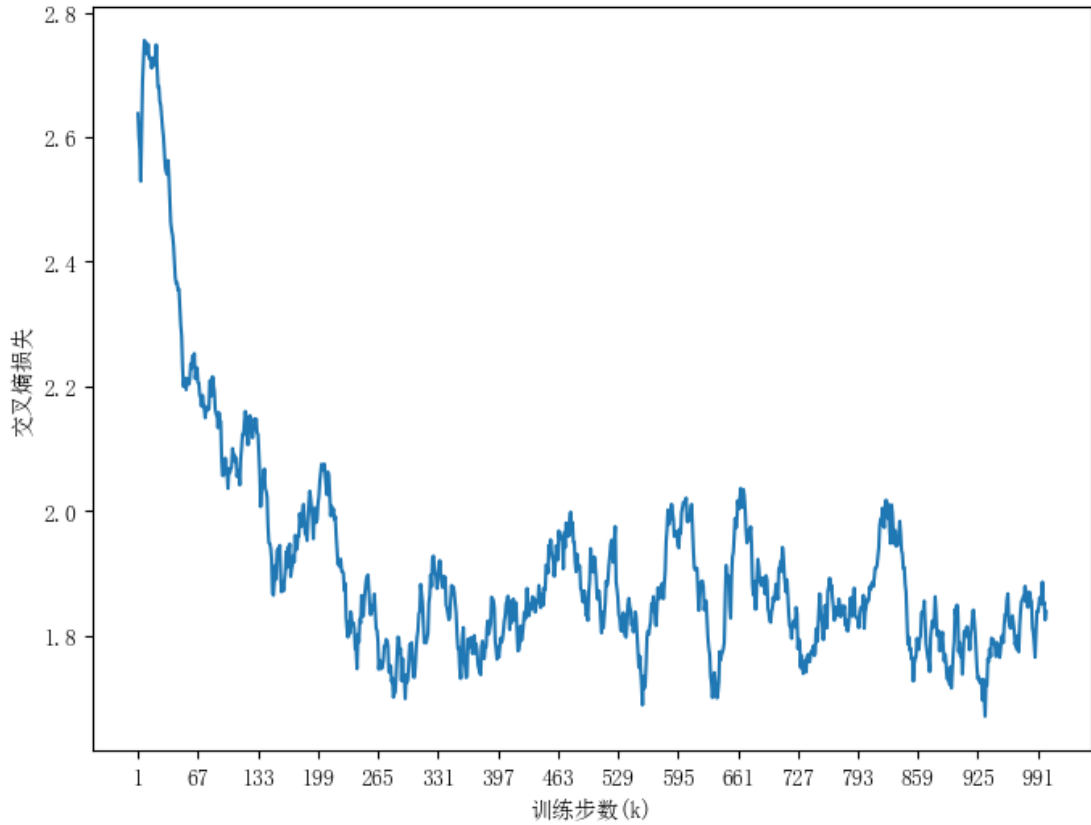


图 3.6 训练过程中的交叉熵损失与训练步数的关系

一。实验结果如表 3.4 所示。其中与其他 LM 比较的另一个 LM 选择的是原始的 GPT2 模型^[6]（主要的训练语料是英文），使用滑动窗口的困惑度解码时，滑动窗口的大小设置为 5；使用温度衰减时，从 $t = 10$ 开始，在前 10 个 Tokens 的一段时间内衰减到 $t = 1$ (\approx 序列长度的 50%)，之后保持 $t = 1$ 。

表 3.4 攻击方式与成功次数

| 类型 | 成功次数 |
|-------------|------|
| 原始生成 | 14 |
| 与其他 LM 比较 | 19 |
| 与 Zlib 压缩比较 | 14 |
| 使用滑动窗口 | 8 |
| 使用温度衰减 | 10 |

上述实验攻击出的成功次数较少，而研究工作^[16]的文中给出的结果相对而言更多，可能的原因如下：一方面，本实验的 LM 参数量较少，只有 81.9M，而文章^[16]使用的 GPT2-large 为 1.5B 的参数量，更多参数量能够让模型记住更多的信息，这一点在相关文献中得到证实^[5-7,67-69]；另一方面，本实验的训练语料较少，LM 还没充分的学到医学文本生成场景下的特点，导致其困惑度仍较高。

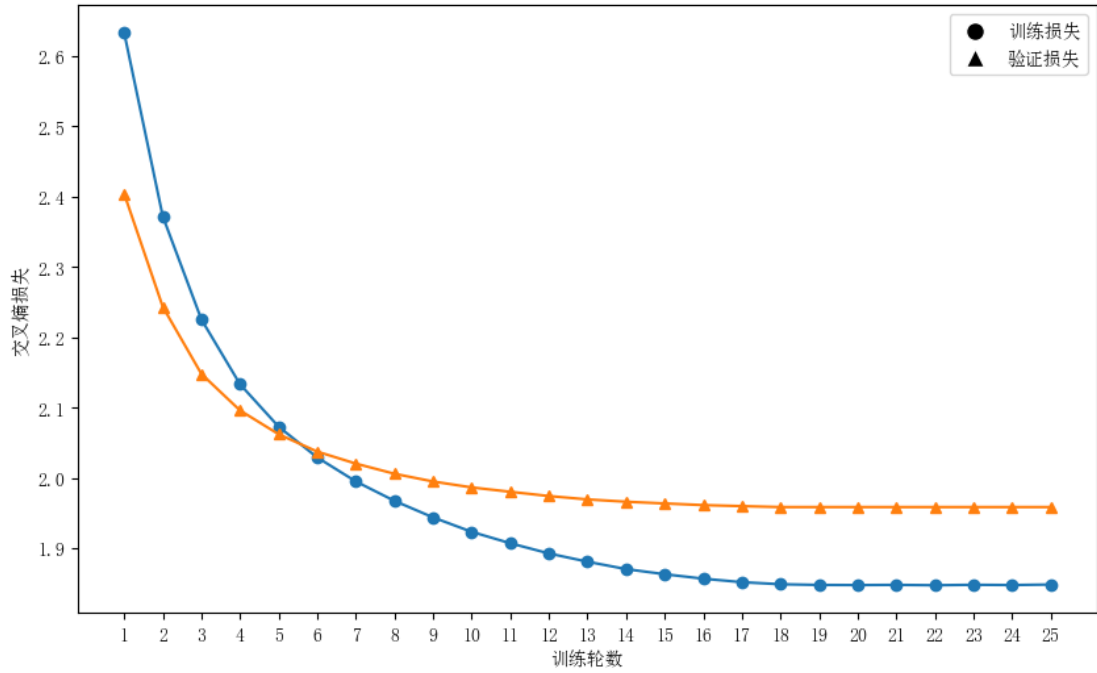


图 3.7 使用 CMDD 数据集微调 GPT2-Chinese 的 Loss 随 Epoch 的变化

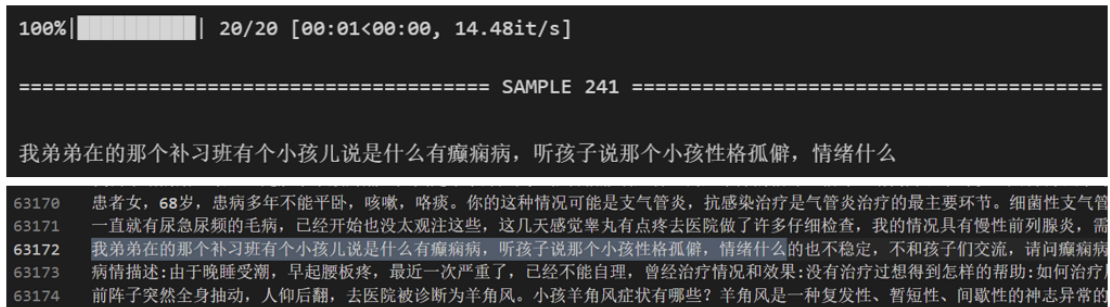


图 3.8 模型恢复出训练数据中一个样本的前 40 个 Tokens

3.6 本章小结

本章探讨医学文本生成任务在训练和推断阶段中的隐私泄露风险，以证明隐私保护机制的必要性。首先，本章详细介绍了语言模型的生成过程，为后续分析医学文本生成模型的训练与推断阶段的执行过程奠定了基础。接着，本章分析了语言模型的记忆问题，并进行了攻击实验及提出改进的攻击策略。同时，本章还探讨了攻击者在训练阶段如何尝试推断隐私数据以及破坏训练协议的可能攻击手段。最后，本章阐述了攻击者可能尝试通过执行输入和标签重构攻击来恢复训练隐私数据。通过在医学文本数据下训练的语言模型攻击实验，本章展示了语言模型记忆问题带来的隐私挑战。本章的内容强调了隐私保护在医学文本生成任务中的重要性。

第4章 医学文本生成任务训练阶段的隐私保护研究

4.1 引言

为防止攻击者在训练阶段试图推断隐私数据以及破坏训练协议的攻击，本章提出了一个基于秘密共享^[70-71]的安全训练协议。

本章主要研究医疗文本生成任务训练阶段中的隐私保护问题，旨在提供一种安全协议，以保护医疗数据在训练阶段的隐私。在系统模型和威胁模型的基础上，我们设计了安全目标，并提出了基于秘密共享的多方计算协议以保障机密性。为确保执行过程的完整性，我们引入了可信硬件。我们还扩展了基于秘密共享的协议，使其能够支持复杂的 Transformer 结构。接着，我们分析了协议的安全性，证明了它能够满足设计目标。最后，我们通过实验证明了协议的有效性和高效性。通过这些工作，我们可以在保障数据隐私的同时，对医疗领域的文本生成任务进行训练，为医疗领域的应用提供安全保障。

4.2 模型与设计目标

4.2.1 系统模型

在这种场景下，我们假设有两类实体，一方是多个拥有医疗隐私数据的数据持有者，另一方是提供计算服务的多个计算方。多个数据持有者希望通过多个计算方提供的计算服务来协同训练模型，计算方在计算服务结束后将训练好的模型分发给各个数据持有者。

如图4.1所示，其中①表示多个数据持有者通过秘密共享算法将数据拆分成两个秘密份额分发给服务器 p_0 与服务器 p_1 。②表示在服务器 p_2 提供的相应 MPC 计算随机数的辅助下，服务器 p_0 与服务器 p_1 运行 MPC 协议来执行模型的训练过程。③表示训练结束后，服务器 p_0 与服务器 p_1 将各自的模型参数份额返还给各数据持有者。在这种场景下，我们假设有两类实体，一方是多个拥有医疗隐私数据的数据持有者，另一方是提供计算服务的三个计算方。多个数据持有者希望通过多个计算方提供的计算服务来协同训练模型，计算方在计算服务结束后将训练好的模型分发给各个数据持有者。其中作为提供计算服务的三个计算方 p_0 、 p_1 与 p_2 均具有 Intel SGX，且 p_0 与 p_1 具有高性能计算 GPU 或者 TPU（后文均以 GPU 指代）。

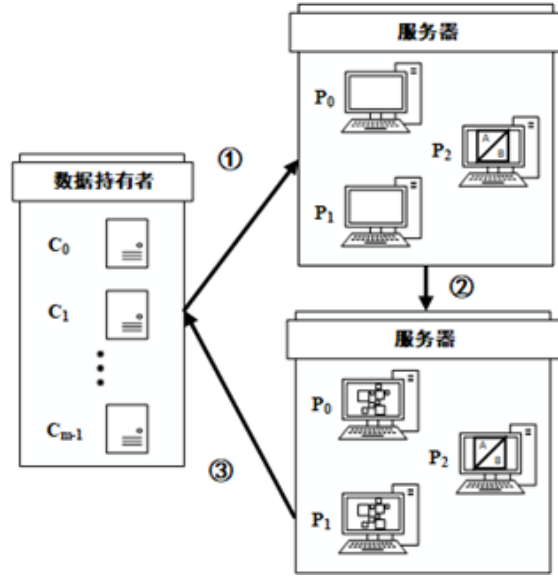


图 4.1 系统概述示意图

4.2.2 威胁模型

本文对该场景下数据持有者的安全假设是其会严格遵循协议提供本地数据并且不会推测其他数据持有者的隐私数据，对计算方的安全假设是其可以任意偏离协议并且会从获取到的数据推断数据持有者的隐私信息以及模型参数，即数据持有者是诚实的，而计算方是恶意的。计算方服务器不仅会根据获取到的信息来推测隐私数据，而且还会任意偏离协议，即在协议执行过程中返回恶意结果。此外，本文假设多个恶意计算方不会共谋。为规范服务器行为，我们引入可信硬件来保障执行过程的完整性。具体来说，本文引入 Intel Software Guard Extension (SGX)^[72]作为可信硬件。相关研究将 SGX 视为一个可信第三方，即其机密性与完整性都不被破坏，这样的信任假设过强。由于测信道攻击等攻击手段可能会破坏 SGX 的机密性^[73]，因此直接将 SGX 是为一个可信第三方有风险。本协议只关注 SGX 中 Enclave 与 GPU 交互数据时由于访问模式导致的侧信道信息泄露问题，而如缓存冲突攻击、计时攻击和工号分析攻击的其他相关侧信道攻击均不在本方案考虑范围内。本协议对计算方上 SGX 的信任假设较弱，即其机密性可以被破坏但是会保留其完整性，这是一种对可信硬件常见的安全假设^[25]。对计算方上 GPU 的安全假设是其执行内容对服务器可见，但是执行协议不会被破坏。

4.2.3 设计目标

对该场景下的安全目标是数据持有者的隐私数据不会被计算方推断出，同时计算方会按照约定的协议严格执行训练过程。

4.3 训练协议设计

本节从基于秘密共享的基本的安全神经网络函数的构建开始，逐渐封装到完整的大型语言模型的实现。

4.3.1 多方安全计算深度学习函数的实现

Transformer 模型包含线性运算与非线性运算。线性层（Linear Layer）、全连接层（Fully Connected Layer）、卷积层（Convolutional Layer）本质都是矩阵乘法运算，这是神经网络模型中最经常被用到的运算。模型经常会把这些层通过一些比如 ReLU、Sigmoid、Exp 等激活函数进行非线性处理。因此，为了构建常见的神经网络模型结构，本文对于这些线性与非线性函数进行了设计。其中，线性运算是矩阵乘法运算。

本文的非线性运算有 ReLU、Softmax。其中^{[23][74]}中已经有了 ReLU、Tanh、Sigmoid 的实现，我们将利用这些函数以及参考^[75]中的 Trunc、BitDecomp 与 PreMult 函数的实现来构建 Exp 与 Softmax 的实现。其中各函数的调用关系如图4.2所示，标记为红色的是构建模型的主要函数。

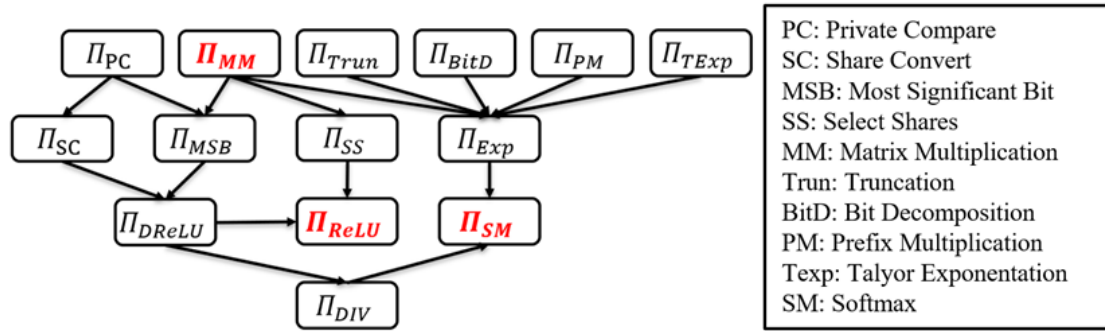


图 4.2 各函数的调用关系

算法4.1描述了本文的三方矩阵乘法协议 ($\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$)，其中参与方 P_0 和 P_1 持有矩阵 X 和 Y 的秘密份额，协议输出 $Z = X \cdot Y$ 的秘密份额。算法4.3描述了本文的三方 Exp 协议 ($\Pi_{\text{Exp}}(\{P_0, P_1\}, P_2)$)，其中参与方 P_0 和 P_1 持有矩阵 X 的秘密份额，协议输出 $Z = e^X$ 的秘密份额。有了 Exp 协议便可以根据其来构建 Softmax 协议。如算法4.4所示，本文的三方 Softmax 协议 ($\Pi_{\text{Softmax}}(\{P_0, P_1\}, P_2)$) 中，参与方 P_0 和 P_1 持有序列 Z 的秘密份额，协议输出 $Z = e^{z_i} / (\sum_{i=1}^k e^{z_i})$ 的秘密份额。

算法 4.1 $\Pi_{\text{MatMul}}(\{P_0, P_1\}, P_2)$

输入: P_0 与 P_1 分别持有 $(\langle X \rangle_0, \langle Y \rangle_0)$ 和 $(\langle X \rangle_1, \langle Y \rangle_1)$, X 是 $m \times v$ 的, Y 是 $v \times n$ 的

输出: 对于 $i \in \{0, 1\}$, P_i 获得输出的秘密分片 $\langle X \cdot Y \rangle_i$.

- 1 P_0 、 P_1 分别获得从 P_2 产生的 $U_0 = \langle 0^{m \times n} \rangle_0^L$ 和 $U_1 = \langle 0^{m \times n} \rangle_1^L$
- 2 P_2 生成随机的矩阵 $A \in Z_L^{m \times v}$ 和 $B \in Z_L^{v \times n}$, 并计算 $C = A \cdot B$ 。对于 $i \in \{0, 1\}$, P_2 将 A, B, C 分成秘密份额 $\langle A \rangle_i^L, \langle B \rangle_i^L, \langle C \rangle_i^L$, 并分发给 P_0 、 P_1
- 3 对于 $j \in \{0, 1\}$, P_j 计算 $\langle E \rangle_j^L = \langle X \rangle_j^L - \langle A \rangle_j^L$ 与 $\langle F \rangle_j^L = \langle Y \rangle_j^L - \langle B \rangle_j^L$
- 4 P_0 、 P_1 交换秘密份额来重构 E 与 F
- 5 对于 $j \in \{0, 1\}$, P_j 计算 $-jE \cdot F + \langle X \rangle_j^L \cdot F + E \cdot \langle Y \rangle_j^L + \langle C \rangle_j^L + U_j$

算法 4.2 $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\}, P_2)$

输入: P_0 与 P_1 分别持有 $(\langle x \rangle_0^L)$ 和 $(\langle x \rangle_1^L)$, 其中 $|x| < 1$, 公开的展开阶数 $n(n \geq 4)$

输出: 对于 $i \in \{0, 1\}$, P_i 获得秘密份额 $\langle y \rangle_i^L = \langle e^x \rangle_i^L$.

- 1 P_0 、 P_1 分别获得由 P_2 产生的 $u_0 = \langle 0 \rangle_0^L$ 和 $u_1 = \langle 0 \rangle_1^L$
- 2 P_0 、 P_1 计算 $\langle c \rangle_j^L = j$
- 3 P_0 、 P_1 计算 $\langle \text{numerator} \rangle_j^L = \langle x \rangle_j^L$ 并设置 $\text{denominator} = 1$
- 4 P_0 、 P_1 计算 $\langle c \rangle_j^L = \langle c \rangle_j^L + \langle \text{numerator} \rangle_j^L$
- 5 **for** $i = 2, 3, \dots, n$ **do**
- 6 P_j 的输入为 $\langle \text{numerator} \rangle_j^L$ 与 $\langle x \rangle_j^L$ 时, 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{MatMul}}(\{P_0, P_1\}, P_2)$ 后, P_j 获得 $\langle \text{numerator} \rangle_j^L = \langle \text{numerator} \cdot x \rangle_j^L$
- 7 $\text{denominator} = \text{denominator} \times i$
- 8 P_0 、 P_1 计算 $\langle c \rangle_j^L = \langle c \rangle_j^L + \frac{\langle \text{numerator} \rangle_j^L}{\text{denominator}}$
- 9 **end**
- 10 对于 $j \in \{0, 1\}$, P_j 计算 $\langle y \rangle_j^L = \langle c \rangle_j^L + u_j$

算法 4.3 $\Pi_{\text{Exp}}(\{P_0, P_1\}, P_2)$ **输入:** P_0 与 P_1 分别持有 $(\langle x \rangle_0^L)$ 和 $(\langle x \rangle_1^L)$ **输出:** 对于 $i \in \{0, 1\}$, P_i 获得秘密份额 $\langle y \rangle_i^L = \langle e^x \rangle_i^L$.

- 1 P_0 、 P_1 分别获得由 P_2 产生的 $u_0 = \langle 0 \rangle_0^L$ 和 $u_1 = \langle 0 \rangle_1^L$
- 2 对于 $j \in \{0, 1\}$, P_j 执行 $\Pi_{\text{Trunc}}(\{P_0, P_1\})$ 后, 获得 $\lfloor x \rfloor$ 的份额 $\langle a \rangle_j^L$
- 3 对于 $j \in \{0, 1\}$, P_j 通过计算 $\langle b \rangle_j^L = \langle x \rangle_j^L - \langle a \rangle_j^L$ 获得 x 的小数部分 $\langle b \rangle_j^L$
- 4 对于 $j \in \{0, 1\}$, P_j 执行 $\Pi_{\text{BitDecomp}}(\{P_0, P_1\})$ 后, 获得 $\lfloor x \rfloor$ 按位展开的秘密份额 $(\langle c_0 \rangle_j^L, \langle c_1 \rangle_j^L, \dots, \langle c_{m-1} \rangle_j^L)$, 其中得 $\lfloor x \rfloor$ 是 m 比特数
- 5 **for** $i = 0, 1, \dots, m-1$ **do**
- 6 对于 $j \in \{0, 1\}$, P_j 计算 $\langle v_i \rangle_j^L = e^{2^i} \cdot (\langle c_i \rangle_j^L) + j - (\langle c_i \rangle_j^L)$
- 7 **end**
- 8 对于 $j \in \{0, 1\}$, P_j 执行 $\Pi_{\text{PreMult}}(\{P_0, P_1\})$ 后, 获得 $(\langle m \rangle_j^L)$
- 9 对于 $j \in \{0, 1\}$, P_j 执行 $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\})$ 后, 获得 $(\langle n \rangle_j^L)$
- 10 对于 $j \in \{0, 1\}$, P_j 的输入为时 $(\langle m \rangle_j^L)$ 与 $(\langle n \rangle_j^L)$ 时, 在执行 $\Pi_{\text{MatMul}}(\{P_0, P_1\}, P_2)$ 后, P_j 获得 $\langle y \rangle_j^L = \langle m \times n \rangle_j^L$

该算法 $\Pi_{\text{Exp}}(P_0, P_1, P_2)$ 是一个多方安全计算协议, 用于计算 x 的指数 e^x 。在这个协议中, 参与者 P_0 和 P_1 分别持有 x 的秘密份额, 并且希望计算 e^x 的秘密份额。这个算法利用了算法10来近似 e^x , 以及一些前述工作^[75]的子协议, 例如 $\Pi_{\text{Trunc}}(P_0, P_1)$ 、 $\Pi_{\text{BitDecomp}}(P_0, P_1)$ 、 $\Pi_{\text{PreMult}}(P_0, P_1)$ 和 $\Pi_{\text{TaylorExpansion}}(P_0, P_1)$ 。

算法 4.4 $\Pi_{\text{Softmax}}(\{P_0, P_1\}, P_2)$ **输入:** P_0 与 P_1 分别持有 $(\langle z_i \rangle_0^L)_{i \in [k]}$ 和 $(\langle z_i \rangle_1^L)_{i \in [k]}$ **输出:** P_0 , P_1 分别获得秘密份额 $(\langle s_{\max}(z_i) \rangle_0^L)_{i \in [k]}$ 与 $(\langle s_{\max}(z_i) \rangle_1^L)_{i \in [k]}$,

$$\text{其中 } s_{\max}(z_i) = \frac{e^{z_i}}{\sum_{i=1}^k e^{z_i}}.$$

- 1 P_0 、 P_1 分别获得由 P_2 产生的 $u_0 = \langle 0 \rangle_0^L$ 和 $u_1 = \langle 0 \rangle_1^L$
- 2 **for** $i = 1, 2, \dots, k$ **do**
- 3 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Exp}}(\{P_0, P_1\}, P_2)$ 后, P_0 , P_1 分别获得 $\langle c_i \rangle_0^L$ 与 $\langle c_i \rangle_1^L$, 其中 $c_i^L = e^{z_i}$
- 4 **end**
- 5 对于 $j \in \{0, 1\}$, P_j 计算 $\langle S \rangle_j = \sum_{i=1}^k \langle c_i \rangle_j^L$
- 6 **for** $i = 1, 2, \dots, k$ **do**
- 7 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Division}}(\{P_0, P_1\}, P_2)$ 后, P_0 , P_1 分别获得 $\langle \frac{c_i}{S} \rangle_0^L$ 与 $\langle \frac{c_i}{S} \rangle_1^L$
- 8 对于 $j \in \{0, 1\}$, P_j 计算 $(\langle s_{\max}(z_i) \rangle_j^L)_{i \in [k]} + u_j$
- 9 **end**

在算法4.4中, P_0 和 P_1 分别持有 k 个数值的秘密份额, 其主要目标是计算 Softmax 函数, 即对于每个 i , 得到 $s_{\max}(z_i) = \frac{e^{z_i}}{\sum_{i=1}^k e^{z_i}}$ 的秘密份额。

首先, P_0 和 P_1 分别获得由 P_2 产生的零值份额 $u_0 = \langle 0 \rangle_0^L$ 和 $u_1 = \langle 0 \rangle_1^L$ 。对于每个 i , 使用 $\Pi_{\text{Exp}}(P_0, P_1, P_2)$ 算法计算指数值 e^{z_i} 的秘密份额 $\langle c_i \rangle_0^L$ 和 $\langle c_i \rangle_1^L$ 。接着, P_0 和 P_1 计算总和 S 的秘密份额 $\langle S \rangle_j = \sum_{i=1}^k \langle c_i \rangle_j^L$ 。再对于每个 i , 通过 $\Pi_{\text{Division}}(P_0, P_1, P_2)$ 算法计算 $\frac{c_i}{S}$ 的秘密份额 $\langle \frac{c_i}{S} \rangle_0^L$ 和 $\langle \frac{c_i}{S} \rangle_1^L$ 。最后, P_0 和 P_1 将结果加上零值份额 u_j 以获得 $s_{\max}(z_i)$ 的秘密份额。

4.3.2 语言模型模块的构建

在第 4.3 节中, 我们介绍了神经网络基本函数的设计。接下来, 在本节中, 我们将介绍如何构建语言模型 Transformer 编码器中的各个模块。具体而言, 我们将从线性层 Linear、注意力层 Attention 和前馈网络 FFN 这三个模块的构建进行介绍。

算法 4.5 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$

输入: P_0 与 P_1 分别持有 $(\langle X \rangle_0, \langle W \rangle_0, \langle b \rangle_0)$ 和 $(\langle X \rangle_1, \langle W \rangle_1, \langle b \rangle_1)$

输出: 对于 $i \in \{0, 1\}$, P_i 获得输出的秘密分片 $\langle W \cdot X + b \rangle_i$ 。

- 1 P_0 、 P_1 分别获得从 P_2 产生的 $U_0 = \langle 0^{m \times n} \rangle_0^L$ 和 $U_1 = \langle 0^{m \times n} \rangle_1^L$
 - 2 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 后, P_0 、 P_1 分别获得 $\langle W \cdot X \rangle_0^L$ 与 $\langle W \cdot X \rangle_1^L$
 - 3 对于 $j \in \{0, 1\}$, P_j 计算 $\langle y \rangle_i^L = \langle W \cdot X \rangle_i^L + \langle b \rangle_i^L$
-

算法 4.6 $\Pi_{\text{Attention}}(\{P_0, P_1\}, P_2)$ **输入:** P_0 与 P_1 分别持有

$$(\langle X \rangle_0, \langle W_Q \rangle_0, \langle b_Q \rangle_0, \langle W_K \rangle_0, \langle b_K \rangle_0, \langle W_V \rangle_0, \langle b_V \rangle_0, \langle W_P \rangle_0, \langle b_P \rangle_0) \text{ 和}$$

$$(\langle X \rangle_1, \langle W_Q \rangle_1, \langle b_Q \rangle_1, \langle W_K \rangle_1, \langle b_K \rangle_1, \langle W_V \rangle_1, \langle b_V \rangle_1, \langle W_P \rangle_1, \langle b_P \rangle_1)$$

输出: 对于 $i \in \{0, 1\}$, P_i 获得输出的秘密分片

$$\langle W_P \cdot \text{Softmax}(\frac{(W_Q \cdot X + b_Q)(W_K \cdot X + b_K)^T}{\sqrt{d_k}})(W_V \cdot X + b_V) + b_P \rangle_i.$$

- 1 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$, P_0 、 P_1 分别获得
 $Q = W_Q \cdot X + b_Q$, $K = W_K \cdot X + b_K$, $V = W_V \cdot X + b_V$ 的秘密份额
 $(\langle Q \rangle_0, \langle K \rangle_0, \langle V \rangle_0)$ 与 $(\langle Q \rangle_1, \langle K \rangle_1, \langle V \rangle_1)$
- 2 P_0 、 P_1 分别将 $\langle K \rangle_0$ 与 $\langle K \rangle_1$ 转置得到 $\langle K^T \rangle_0$ 与 $\langle K^T \rangle_1$
- 3 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{MatMul}}(\{P_0, P_1\}, P_2)$, P_0 、 P_1 分别获得 QK^T 的秘密份
额 $\langle QK^T \rangle_0$ 与 $\langle QK^T \rangle_1$
- 4 P_0 、 P_1 分别将 $\langle QK^T \rangle_0$ 与 $\langle QK^T \rangle_1$ 按元素除 W_K 的维度 $\sqrt{d_k}$ 得到
 $\langle \frac{QK^T}{\sqrt{d_k}} \rangle_0$ 与 $\langle \frac{QK^T}{\sqrt{d_k}} \rangle_1$
- 5 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Softmax}}(\{P_0, P_1\}, P_2)$, P_0 、 P_1 分别获得
 $A = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})$ 的秘密份额 $\langle A \rangle_0$ 与 $\langle A \rangle_1$
- 6 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$, P_0 、 P_1 分别获得 $\langle A \cdot V + b_V \rangle_0$ 与
 $\langle A \cdot V + b_V \rangle_1$
- 7 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$, P_0 、 P_1 分别获得
 $\langle y \rangle_0 = \langle W_P(A \cdot V + b_V) + b_P \rangle_0$ 与 $\langle y \rangle_1 = \langle W_P(A \cdot V + b_V) + b_P \rangle_1$

算法 4.7 $\Pi_{\text{FFN}}(\{P_0, P_1\}, P_2)$ **输入:** P_0 与 P_1 分别持有 $(\langle X \rangle_0, \langle W_1 \rangle_0, \langle W_2 \rangle_0, \langle b_1 \rangle_0, \langle b_2 \rangle_0)$ 和

$$(\langle X \rangle_1, \langle W_1 \rangle_1, \langle W_2 \rangle_1, \langle b_1 \rangle_1, \langle b_2 \rangle_1)$$

输出: 对于 $i \in \{0, 1\}$, P_i 获得输出的秘密分片

$$\langle W_2 \cdot \text{ReLU}(W_1 \cdot X + b_1) + b_2 \rangle_i.$$

- 1 P_0 、 P_1 分别获得从 P_2 产生的 $U_0 = \langle 0^{m \times n} \rangle_0^L$ 和 $U_1 = \langle 0^{m \times n} \rangle_1^L$
- 2 P_j 的输入为 $\langle X \rangle_j^L$ 与 $\langle W_1 \rangle_j^L$ 时, 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$
后, P_j 获得 $\langle H \rangle_j^L = \langle W_1 \cdot X + b_1 \rangle_j^L$
- 3 在 P_0 、 P_1 、 P_2 执行 $\Pi_{\text{ReLU}}(\{P_0, P_1\}, P_2)$ 后, P_j 获得 $\langle \text{ReLU}(H) \rangle_j^L$
- 4 P_j 的输入为 $\langle \text{ReLU}(H) \rangle_j^L$ 与 $\langle W_2 \rangle_j^L$ 时, 在 P_0 、 P_1 、 P_2 执行
 $\Pi_{\text{Linear}}(\{P_0, P_1\}, P_2)$ 后, P_j 获得 $\langle O \rangle_j^L = \langle W_2 \cdot \text{ReLU}(H) + b_2 \rangle_j^L$
- 5 对于 $j \in \{0, 1\}$, P_j 计算 $\langle O \rangle_j^L + U_j$

4.3.3 可验证外包计算的设计

本节介绍了一个可验证外包计算的设计，与前一部分不同的是，此设计需要每个服务器都拥有高性能的 GPU 计算资源。我们的安全假设是计算方是恶意的，可以任意偏离协议并推断数据持有者的隐私信息和模型参数。对于计算方上的 SGX，我们假设其机密性可能被破坏但会保留其完整性；对于计算方上的 GPU，我们假设其执行内容对服务器可见，但执行协议不会被破坏。这是一种对可信硬件常见的安全假设^[25]。

如图4.3所示，首先，数据持有者使用秘密共享方法将数据拆分成两个秘密份额，并在与两个服务器的可信硬件进行远程认证后，将数据分发至服务器。然后，在与第三个持有可信硬件的服务器认证后，服务器之间根据算法的具体实现方式进行交互。两个获得秘密份额的服务器在第三个服务器提供的随机数的帮助下，完成相应的算法计算。最终，两个服务器分别得到的预测结果的秘密份额，在交互重构后得到最终的预测结果，并通过真值计算损失函数来更新存储在两个服务器上的权重。

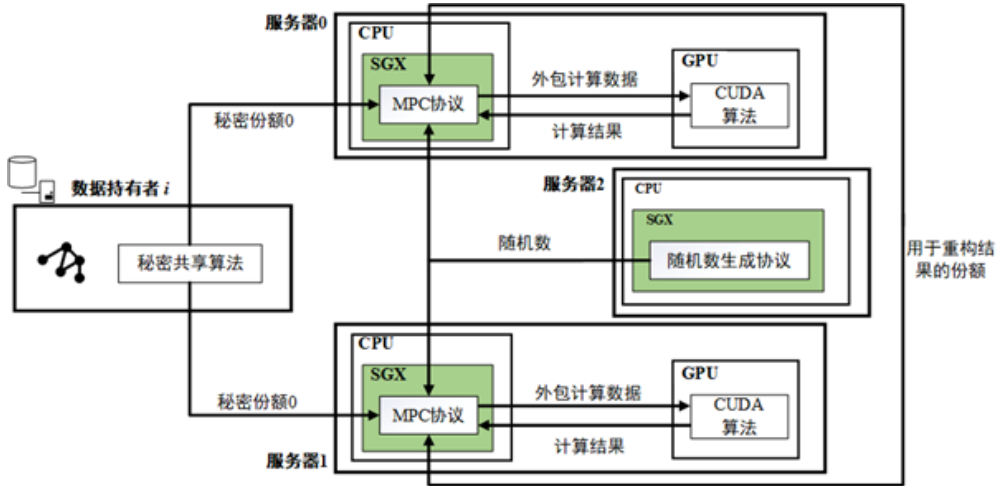


图 4.3 外包 GPU 加速计算的协议流程

算法 4.8 外包计算并验证正确性

输入: SGX 持有模型矩阵参数 M 的秘密份额 $\langle M \rangle_j^L \in Z_L^{m \times d}$ 与输入 x 的

秘密份额 $\langle x \rangle_j^L \in Z_L^{d \times n}$

输出: SGX 获得计算结果 $\langle M \rangle_j^L \cdot \langle x \rangle_j^L$

- 1 SGX 将 $\langle M \rangle_j^L$ 与 $\langle x \rangle_j^L$ 分发给 GPU
- 2 GPU 计算 $y = \langle M \rangle_j^L \cdot \langle x \rangle_j^L$
- 3 **for** $i = 1, 2, \dots, n$ **do**
- 4 SGX 生成随机向量 $\alpha \in Z_L^{n \times 1}$
- 5 SGX 验证 $y \cdot \alpha$ 与 $\langle M \rangle_j^L \cdot (\langle x \rangle_j^L \cdot \alpha)$ 是否相等
- 6 **end**

算法6描述了本文的外包计算协议。为了加速计算，本文引入了 GPU 并进行了外包计算。具体来说，SGX 将需要计算的秘密份额分发给本地的 GPU，GPU 计算完结果后返回给 SGX。然后，SGX 使用 Freivalds' 验证算法^[76]来校验计算结果的正确性。该算法通过引入一个列向量 x ，计算 $A \cdot (B \cdot x)$ 与 $C \cdot x$ 的结果，来验证矩阵运算 $A \cdot B = C$ 的正确性。由于矩阵与列向量的运算时间短，该验证算法能将验证时间由 $O(n^3)$ 降至 $O(kn^2)$ ，其中 k 为验证次数。

4.4 安全性分析

4.4.1 训练安全

基于通用可组合模型^[77-78] (Universally Compsable, UC)，本部分给出4.3中各算法的系统安全性的理论证明。根据4.2的定义，本部分证明本章的系统在恶意攻击者存在的情况下能够保证安全性。在4.2的安全假设中，假设存在一个敌手 A ，它能够控制其中一个参与方。令 $I \subset N$ 表示被控制方的集合， $|I| = 1$ 。设 $J = N/I$ 是诚实的参与者的集合。在整个证明过程中， $P_i (i \in I)$ 表示被攻击者操作一方，而 $P_j (j \in J)$ 表示诚实方。安全性定义遵循相关工作中的方法。与研究工作^[22,79-80]相同，在本章条件下有如下引理成立：

定义 4.1 对于任何多项式时间的攻击者 A ，如果存在一个模拟器 S 能够构建一个模拟世界，在这个模拟世界中 A 的视图与真实世界中 A 的视图在计算上不可区分，那么该协议是可证明安全的。

引理 4.1 如果一个协议的所有子协议都是完全可模拟的，那么该协议本身也是完全可模拟的。

在 UC 框架中，一个协议通常由多个子协议组成，每个子协议都有自己的安全性质。如果每个子协议都是可模拟的，也就是说，每个子协议都可以在模拟环境中与理想功能模型等效地运行，那么整个协议也可以在模拟环境中与理想功能模型等效地运行。

这是因为在 UC 框架中，可以使用虚拟攻击者来证明协议的安全性，虚拟攻击者在真实环境和模拟环境下执行相同的攻击，然后证明模拟环境中的协议能够抵御虚拟攻击者的攻击。如果每个子协议都是可模拟的，那么整个协议也可以在模拟环境中与理想功能模型等效地运行，因此整个协议也可以被证明是安全的。

引理 4.2 如果 r 是攻击者未知的且服从均匀分布的一个随机数，那么 $x \cdot r$ 对攻击者也是未知的，并且与 x 独立。

引理 4.3 如果 r 是攻击者未知的且服从均匀分布的一个随机数，那么 $x \pm r$ 对攻击者也是未知的，并且与 x 独立。

引理 4.4 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 以及其线性组合是安全的^[81-82]。

引理 4.5 $\Pi_{\text{Trunc}}(\{P_0, P_1\})$ 、 $\Pi_{\text{BitDecomp}}(\{P_0, P_1\})$ 、 $\Pi_{\text{PreMult}}(\{P_0, P_1\})$ 是可模拟的^[83]

基于上述的引理，下面本节提供4.3中各算法的系统安全性的理论证明，即存在模拟器 S 使得攻击者 A 在现实世界与理想世界中的视图在计算上是不可区分的。

定理 4.6 在4.2中的安全假设下， $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\}, P_2)$ 可以保证面对恶意攻击者攻击的安全性。

证明

1)

a 若 $P_i = P_0$ 或 $P_i = P_1$ ，其持有数据的视图如下：

$$\begin{aligned} \text{view}_i^{\text{TaylorExpansion}} &= \{u_i, \langle x \rangle_i, \langle \text{numerator} \rangle_i, \text{denominator}, \langle c \rangle_i\} \\ \text{output}_i^{\text{TaylorExpansion}} &= \{y_i\} \end{aligned}$$

其中秘密份额 $\langle x \rangle_i$ 的安全性由4.3保障。根据引理4.4，现有工作已证明 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 在半诚实攻击者存在情况下的安全性，因此在以 $\langle x \rangle_i$ 作为输入的情况下， $\langle \text{numerator} \rangle_i$ 的值满足均匀随机性。此外，由于 denominator 是每一阶 Talor Expansion 的常系数，与 x 无关。在以 $\langle \text{numerator} \rangle_i$ 与 denominator 作为输入的 $\langle c \rangle_i$ 同样满足均匀随机性。至此，模拟器可以使用均匀分布的随机数来模拟 $\text{view}_i^{\text{TaylorExpansion}}$ ，即攻击者的模拟视图与现实视图的概率分布不可区分。对于计算结果 $\text{output}_i^{\text{TaylorExpansion}} = \{y_i\}$ ， x 为加法秘密共享 $\langle x \rangle_i$ 的重构值，由计算过程 c 为计算 x 的 n 阶 Taylor 展开值的加法秘密共享 $\langle c \rangle_i$ 的秘密份额，因此输出 $\langle y \rangle_j^L = \langle c \rangle_j^L + u_j$ 满足均匀随机性，并且重构结果等于正确的计算结果，故模拟器可以通过满足均匀分布的随机数与理想功能的输出对 $\text{output}_i^{\text{TaylorExpansion}} = \{y_i\}$ 进行有效模拟，即其输出的概率分布与真实世界的输出满足不可区分性。至此，证明了 $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\}, P_2)$ 可抵抗半诚实攻击者。

b 若 $P_i = P_2$ ，其持有数据的视图为：

$$\begin{aligned} \text{view}_i^{\text{TaylorExpansion}} &= \{u_0^s, u_1^s, \langle A \rangle_i^s, \langle B \rangle_i^s, \langle C \rangle_i^s, A^s, B^s, C^s\} \\ \text{output}_i^{\text{TaylorExpansion}} &= \{u_0^s, u_1^s, \langle A \rangle_i^s, \langle B \rangle_i^s, \langle C \rangle_i^s\} \end{aligned}$$

其中 s 为 $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\}, P_2)$ 中调用 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 的场景。 P_2 在整个过程中只对 P_0, P_1 单向提供 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 计算所需随机

数, 故不会获取任何有关 x 的任何信息。由于 $u_0^s, A^s, B^s, \langle A \rangle_0^s, \langle B \rangle_0^s, \langle C \rangle_0^s$ 为 P_2 独立生成的随机数, 在半诚实的场景下, 其计算的 $u_1^s, C^s, \langle A \rangle_1^s, \langle B \rangle_1^s, \langle C \rangle_1^s$ 均满足均匀随机性, 并且 $\{(u_0^s, u_1^s), (\langle A \rangle_0^s, \langle A \rangle_1^s), (\langle B \rangle_0^s, \langle B \rangle_1^s), (\langle C \rangle_0^s, \langle C \rangle_1^s)\}$ 可分别重构为 $\{0, A, B, C\}$, 故模拟器可以通过满足均匀分布的随机数与理想功能的输出对 $output_i^{\text{TaylorExpansion}}$ 进行有效模拟, 其输出的概率分布与真实世界的输出满足不可区分性。

- 2) 同时, 在4.2中的安全假设下, 由于 SGX 的完整性保障^[72], P_i 无法对 SGX 飞地内的数据进行篡改, 即将上述半诚实攻击者升级为恶意攻击者时, 协议仍可以保障攻击者严格遵守协议。因此, 恶意攻击者不仅推断不出隐私数据, 而且不能偏离协议执行其他操作, 即本协议可以抵御恶意攻击者的攻击。

因此, $view_i^{\text{TaylorExpansion}}$ 是可模拟的, 无法找到一个概率多项式时间算法来区分 $view_i$ 和 P_i 的模拟视图。因此, $\Pi_{\text{TaylorExpansion}}(\{P_0, P_1\}, P_2)$ 在4.2中的安全假设下是安全的。

■

定理 4.7 在4.2中的安全假设下, $\Pi_{\text{Exp}}(\{P_0, P_1\}, P_2)$ 可以保证面对恶意攻击者攻击的安全性。

证明 若 $P_i = P_0$ 或 $P_i = P_1$, 其持有数据的视图如下:

$$view_i^{\text{Exp}} = \{u_i, \langle x \rangle_i, \langle a \rangle_i, \langle b \rangle_i, \langle c[0], \dots, c[m-1] \rangle_i, \langle m \rangle_i, \langle n \rangle_i\}$$

$$output_i^{\text{Exp}} = \{y_i\}$$

其中秘密份额 $\langle x \rangle_i$ 的安全性由4.3保障。引理4.5保障了 $\langle a \rangle_i = \lfloor x \rfloor$ 的安全性, 进而获得 x 的小数部分 $\langle b \rangle_j^L = \langle x \rangle_j^L - \langle a \rangle_j^L$ 满足均匀随机性。同样的, 在执行 $\Pi_{\text{BitDecomp}}(\{P_0, P_1\})$ 获得的 $\langle c[0], \dots, c[m-1] \rangle_i$ 以及执行 $\Pi_{\text{PreMult}}(\{P_0, P_1\})$ 获得的 $\langle m \rangle_i$ 也满足均匀随机性。根据定理4.6, TaylorExpansion 的计算结果 $\langle n \rangle_i$ 满足均匀随机性。根据引理4.4, 现有工作已证明 $\Pi_{\text{Matmul}}(\{P_0, P_1\}, P_2)$ 在半诚实攻击者存在情况下的安全性, 因此在以 $\langle m \rangle_i$ 与 $\langle n \rangle_i$ 作为输入的情况下, $\langle y \rangle_i$ 的值满足均匀随机性, 并且重构结果等于正确的计算结果, 故模拟器可以通过满足均匀分布的随机数与理想功能的输出对 $output_i^{\text{Exp}} = \{y_i\}$ 进行有效模拟, 其输出的概率分布与真实世界的输出满足不可区分性。至此, 证明了 $\Pi_{\text{Exp}}(\{P_0, P_1\}, P_2)$ 可抵抗半诚实攻击者。

与定理4.6的证明相同, 同理可证在 $P_i = P_2$ 时, 模拟器可以通过满足均匀分布的随机数与理想功能的输出对 $view_i^{\text{Exp}}$ 与 $output_i^{\text{Exp}}$ 进行有效模拟, 其输出的概率分布与真实世界的输出满足不可区分性。

同样的,在4.2中的安全假设下,由于SGX的完整性保障^[72], P_i 无法对SGX飞地内的数据进行篡改,即将上述半诚实攻击者升级为恶意攻击者时,协议仍可以保障攻击者严格遵守协议。因此,恶意攻击者不仅推断不出隐私数据,而且不能偏离协议执行其他操作,即本协议可以抵御恶意攻击者的攻击。

因此, $view_i^{Exp}$ 是可模拟的,无法找到一个概率多项式时间算法来区分 $view_i$ 和 P_i 的模拟视图。因此, $\Pi_{Exp}(\{P_0, P_1\}, P_2)$ 在4.2中的安全假设下是安全的。 ■

定理 4.8 在4.2中的安全假设下, $\Pi_{Linear}(\{P_0, P_1\}, P_2)$ 可以保证面对恶意攻击者攻击的安全性。

证明 若 $P_i = P_0$ 或 $P_i = P_1$, 其持有数据的视图如下:

$$view_i^{Linear} = \{u_i, \langle X \rangle_i, \langle W \rangle_i, \langle b \rangle_i\}$$

$$output_i^{Linear} = \{y_i\}$$

其中秘密份额 $\langle X \rangle_i$ 与权重矩阵 $\langle W \rangle_i$ 的安全性由4.3保障。根据引理4.4, 现有工作已证明 $\Pi_{Matmul}(\{P_0, P_1\}, P_2)$ 在半诚实攻击者存在情况下的安全性, 因此在以 $\langle W \rangle_i$ 与 $\langle X \rangle_i$ 作为输入的情况下, $\langle W \cdot X \rangle_i$ 的值满足均匀随机性。进而由引理4.3可知, $\langle W \cdot X \rangle_i + \langle b \rangle_i$ 的值满足均匀随机性, 并且重构结果等于正确的计算结果, 故 $output_i^{Exp} = \{y_i\}$ 可以通过满足均匀分布的随机数与理想功能的输出对计算结果进行有效模拟, 其输出的概率分布与真实世界的输出满足不可区分性。至此, 证明了 $\Pi_{Linear}(\{P_0, P_1\}, P_2)$ 可抵抗半诚实攻击者。

与定理4.6的证明相同, 同理可证在 $P_i = P_2$ 时, $view_i^{Linear}$ 与 $output_i^{Linear}$ 可以通过满足均匀分布的随机数与理想功能的输出对计算结果进行有效模拟, 其输出的概率分布与真实世界的输出满足不可区分性。

同样的,在4.2中的安全假设下,由于SGX的完整性保障^[72], P_i 无法对SGX飞地内的数据进行篡改,即将上述半诚实攻击者升级为恶意攻击者时,协议仍可以保障攻击者严格遵守协议。因此,恶意攻击者不仅推断不出隐私数据,而且不能偏离协议执行其他操作,即本协议可以抵御恶意攻击者的攻击。

因此, $view_i^{Linear}$ 是可模拟的,无法找到一个概率多项式时间算法来区分 $view_i$ 和 P_i 的模拟视图。因此, $\Pi_{Exp}(\{P_0, P_1\}, P_2)$ 在4.2中的安全假设下是安全的。 ■

定理 4.9 在4.2中的安全假设下, $\Pi_{Attention}(\{P_0, P_1\}, P_2)$ 可以保证面对恶意攻击者攻击的安全性。

证明 若 $P_i = P_0$ 或 $P_i = P_1$ ，其持有数据的视图如下：

$$view_i^{\text{Attention}} = \{\langle X \rangle_0, \langle W_Q \rangle_0, \langle b_Q \rangle_0, \langle W_K \rangle_0, \langle b_K \rangle_0, \langle W_V \rangle_0, \langle b_V \rangle_0, \langle W_P \rangle_0, \langle b_P \rangle_0\}$$

$$output_i^{\text{Attention}} = \{y_i\}$$

由引理4.4与定理4.8， P_i 计算的 $(\langle Q \rangle_i, \langle K \rangle_i, \langle V \rangle_i, \langle QK^T \rangle_i, \langle \frac{QK^T}{\sqrt{d_k}} \rangle_i)$ 满足均匀随机性。根据 $\Pi_{\text{Softmax}}(\{P_0, P_1\}, P_2)$ 的安全性，注意力值 $A = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})$ 同样满足均匀随机性。同样由定理4.8可知，通过矩阵乘法以及 Linear 的计算结果 $\langle y \rangle_i$ 的值满足均匀随机性，并且重构结果等于正确的计算结果，故模拟器可以通过满足均匀分布的随机数与理想功能的输出对 $output_i^{\text{Attention}} = \{y_i\}$ 进行有效模拟，其输出的概率分布与真实世界的输出满足不可区分性。至此，证明了 $\Pi_{\text{Attention}}(\{P_0, P_1\}, P_2)$ 可抵抗半诚实攻击者。

与定理4.6的证明相同，同理可证在 $P_i = P_2$ 时， $view_i^{\text{Attention}}$ 与 $output_i^{\text{Attention}}$ 可以通过满足均匀分布的随机数与理想功能的输出对计算结果进行有效模拟，其输出的概率分布与真实世界的输出满足不可区分性。

同样的，在4.2中的安全假设下，由于 SGX 的完整性保障^[72]， P_i 无法对 SGX 飞地内的数据进行篡改，即将上述半诚实攻击者升级为恶意攻击者时，协议仍可以保障攻击者严格遵守协议。因此，恶意攻击者不仅推断不出隐私数据，而且不能偏离协议执行其他操作，即本协议可以抵御恶意攻击者的攻击。

因此， $view_i^{\text{Attention}}$ 是可模拟的，无法找到一个概率多项式时间算法来区分 $view_i$ 和 P_i 的模拟视图。因此， $\Pi_{\text{Attention}}(\{P_0, P_1\}, P_2)$ 在4.2中的安全假设下是安全的。

■

4.4.2 SGX 被攻破的影响

自 Intel 推出 SGX 以来，各种攻击接踵而至。其中最具有代表性的攻击时侧信道攻击，如功耗分析攻击、计时攻击、回滚攻击与缓存冲突攻击等^[58-65]。目前也有很多缓解这些攻击的研究^[84-91]，这些工作与本方案是互补的。

考虑到比4.2中的安全假设更具挑战的情况——SGX 被完全攻破，即在上述针对保护 SGX 的方案均失效的情况下，SGX 不仅丧失机密性，而且完整性也被破坏。这种情况下，本协议只损失了协议执行结果正确性，而隐私性仍能得到保证。由于在4.3中各函数执行过程中均使用秘密份额进行交互，由前述安全性分析以及4.2中的安全假设，在任意两方不共谋的情况下，任何一方均不能重构出原始训练数据以及模型参数的任何信息，故 SGX 被完全攻破的情况下，协议仍保障了隐私性。而由于丧失了完整性，恶意的计算方服务器可能使用一些恶意的结果作为秘密份额与其他计算方服务器进行交互，导致重构结果与正确结果不

同，即损失了正确性。总的来讲，即使 SGX 被完全攻破，本协议仍能保障半诚实安全假设下的安全性。此外，本协议不局限于使用 SGX 作为可信硬件，其他如 Keytone^[92]、HyperEnclave^[93]等的可信执行环境可替代 SGX。此外，与本方案互补的 SGX 防御技术可以为本协议的 SGX 进行补充，如随机化技术、异常检测、源码重构与增强隔离等。

4.5 实验评估

4.5.1 实验设置

本实验的目标与3.5节的设定相同，在中文预训练模型的基础上在 CMDD 数据集上进行微调。由于其参数量为 81.9M，使用的词表大小为 21128，隐层维度为 768，12 层的 GPT2Block，这些设定通过本节协议完整的训练完时间硬件成本开销过于大，而成熟的框架如 Pytorch 对底层很多实现细节进行了优化。因此，这里分两个部分来说明本章协议的有效性：

- 1) 等效模型训练。使用与本协议执行等价的 Pytorch 代码执行相应的训练微调工作，即通过一个等价的高效框架来说明本协议计算的正确性以及与常规模型计算结果的一致性。
- 2) 协议的开销分析与实验。分析4.3节中各函数与模块的执行开销，包括执行时间复杂度与通信开销的理论分析与实验结果。

等效模型训练

本章使用以下参数设置实例化4.3节中的子协议。机器学习算法通常使用实数 (float32)，而加法秘密共享仅限于整数计算。与相关研究工作相同^[24,29,54,94]，本部分在安全协议中使用实数的定点编码。具体来说，对于实数 x ，本部分考虑 t 位精度的定点编码： $\lfloor x \cdot 2^t \rfloor$ 。当乘以两个定点编码的数字时，由于它们都乘以 2^t ，因此两个方需要额外缩放由 2^{2t} 缩放的乘积，本章使用来自^[29]的截断技术。本章的实验考虑带有 15 位精度的 $Z_{2^{64}}$ 环。

这里与主要修改的是非线性运算的逻辑。第一，由于这里使用的 Π_{ReLU} 函数的一个重要前置函数是调用 Π_{MSB} 来计算最高位的值，即符号位，这就要求数据是在整数域上。因此，在执行 ReLU 前，需要对中间结果调用 Π_{Trunc} 来获取其整数部分。第二，在调用 Π_{Softmax} 时，由于其中使用到的 Π_{Exp} 函数中需要调用的 $\Pi_{\text{TaylorExp}}$ ，这里本实验设置的是展开到 5 阶 ($n = 5$)，那么与实际结果相比，会在精度上有一定的误差。

实验环境与3.3.1相同，如表4.1所示：CPU 为 AMD Ryzen 9 5900HX、32GB RAM、GPU 为 RTX3080-Laptop、操作系统为 Windows 11 64 位。

协议的开销分析与实验

表 4.1 等效模型训练的实验环境

| 维度 | 配置 |
|------|------------------------------|
| 处理器 | AMD Ryzen 9 5900HX @ 3.30GHz |
| 内存 | 32G DDR4 3200Hz |
| GPU | RTX3080-Laptop 16G VRAM |
| 操作系统 | Windows 11 64 位 |
| 硬盘 | 1TB SSD |

这里的实验环境如表4.2所示：CPU 为 Intel i7-8750H（支持 SGX，实验中主频均在 4.0GHz 以上）、16GB RAM、GPU 为 GTX1060-Laptop、操作系统为 Ubuntu16.04。

表 4.2 协议的开销分析与实验环境

| 维度 | 配置 |
|------|-------------------------------|
| 处理器 | Intel Core i7-8750H @ 4.20GHz |
| 内存 | 16G DDR4 3200Hz |
| GPU | GTX1060 6G VRAM |
| 操作系统 | Ubuntu 16.04 |
| 硬盘 | 512G SSD |

4.5.2 实验结果

(1) 等效模型训练

在其他与本方案类似的工作中^[23-24,29]，使用到的数据集和训练模型的结构与大小都比本章实验相差很多，如 MNIST、Cora 等数据集。本章实验基于 GPT2 的模型以及中文医学生成数据集 CMDD 上微调的情况更能反应在大模型上的训练情况。

图4.4所示等效模型和原模型的训练损失函数与训练轮数的变化情况，由于4.5.1节中提到的非线性函数的截断与近似导致的模型训练效果稍差，这种现象与研究模型量化的工作^[95-96]以及研究在 NLP 领域量化^[97-98]的工作中的结论相似。从数值上看，原训练模型与本协议等效的训练模型在训练阶段的交叉熵 Loss 的差距在 0.1 左右，且协议等效训练的波动更大一些。

(2) 协议的开销分析与实验

表4.3展示了4.3.2节中构建的 Transformer 模块的通信开销。其中 n 为输入矩阵的维度，这里的设定跟实际模型执行过程的设定相同，即每个矩阵为 n 维方阵； l 为数据类型的比特数，如 $l_{\text{float}} = 32, l_{\text{int}} = 32$ ； p 为执行比较协议的小整数域 Z_p 的大小（本实验中 $p = 67$ ）。

表 4.3 Transformer 模块的通信开销

| 协议 | 通信开销 |
|-----------|--|
| Linear | $5n^2l$ |
| Attention | $(30 + 5ks)n^2l + (8l \log(p) + 24l)s$ |
| FFN | $10n^2l + 8l \log(p) + 24l$ |

表4.4展示了使用可验证外包计算的加速效果与输入维度的关系。其中，执

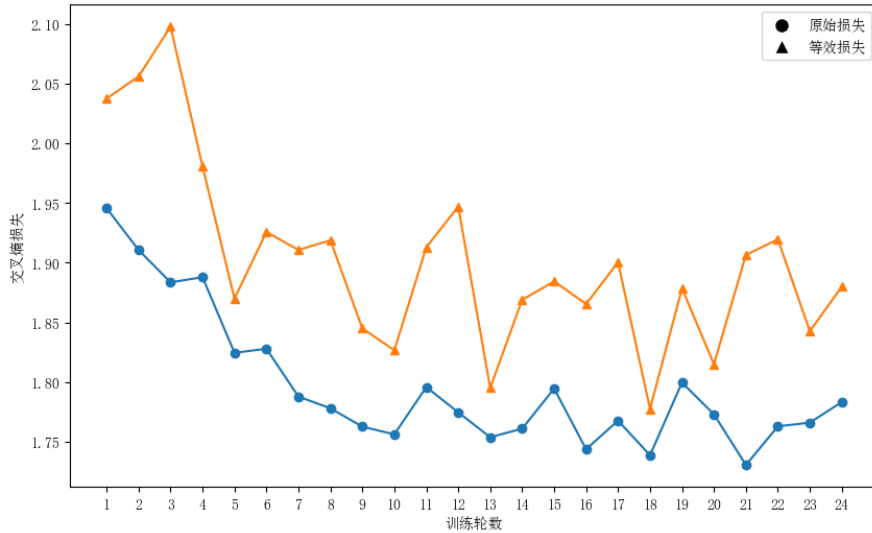


图 4.4 等效模型和原模型的训练损失与轮数的变化情况

行时间为所有在 SGX 上完成矩阵乘法的时间；传输时间为将数据从 Enclave 通过总线传到 GPU 以及计算结果从 GPU 通过总线传输回 Enclave 的总时间开销；验证时间为在 Enclave 中验证乘法结果计算正确性的时间；总外包时间包括了传输时间验证时间以及在 GPU 上的计算时间；加速比为执行时间与总外包时间的比值，反应了外包计算协议的执行加速效果。

表 4.4 使用可验证外包计算的加速效果与输入维度的关系

| 维度 | 执行时间 | 传输时间 | 验证时间 | 总外包时间 | 加速比 |
|------|--------|--------|--------|--------|--------|
| 64 | 0.0001 | 0.0014 | 0.0002 | 0.0016 | 0.0753 |
| 128 | 0.0002 | 0.0016 | 0.0005 | 0.0011 | 0.1909 |
| 256 | 0.0021 | 0.0023 | 0.0012 | 0.0036 | 0.5930 |
| 512 | 0.0094 | 0.0072 | 0.0024 | 0.0098 | 0.9634 |
| 768 | 0.0257 | 0.0136 | 0.0041 | 0.0179 | 1.4419 |
| 1024 | 0.0611 | 0.0114 | 0.0085 | 0.0201 | 3.0509 |
| 2048 | 0.4173 | 0.0517 | 0.0486 | 0.1006 | 4.1497 |

可以看出，在矩阵计算维度为 512 时，二者计算开销大致相同，若维度小于 512，则不使用外包计算的效果更好。反之，若维度大于 512，使用外包计算的时间开销回更少，并且随着维度的增加，这个结果在不断扩大，在 2048 维时，通过外包计算的时间仅不到执行时间的 1/4。

4.6 本章小结

本章主要研究了在医疗文本生成任务训练阶段中的隐私保护问题。首先，明确了系统模型和威胁模型，并设计了安全目标。随后，提出了基于秘密共享的多方计算协议来保障数据机密性，并使用可信硬件保证执行过程的完整性。本章扩展了基于秘密共享的协议，使得可以构建复杂的 Transformer 结构。接着本章分析了协议的安全性，证明了协议满足设计目标。最后，本章通过实验验证了协议

的有效性和高效性。

第5章 医学文本生成任务推断阶段的隐私保护研究

5.1 引言

为防止攻击者在推断阶段试图通过模型反演攻击来恢复训练隐私数据，同时保持语言模型的表现效果，本章基于差分隐私算法提出两种方法来缓解语言模型的记忆问题，从而保护隐私数据。首先，本章将介绍系统模型与设计目标，引入本章的保护对象与攻击者的行为。其次，介绍选择差分隐私的定义，并针对训练与推断阶段分别设计了隐私优化器与解码算法，作为两种提供选择差分隐私的方式。随后，对前述设计的隐私优化器与解码算法进行隐私性分析，以证明其满足差分隐私的定义。最后，通过设计实验说明这两种保护方式的优势。

5.2 系统模型与设计目标

5.2.1 系统模型

如图5.1所示，在这种场景下，拥有隐私医学文本数据集的数据持有者通过一种训练策略得到一个针对该领域医学文本的语言模型，其通过公开推断查询的接口来提供查询服务。正常的诚实使用者将输入传给模型持有者，模型持有者通过语言模型执行推断，并将语言模型的输出返回给使用者。而调用查询接口的使用者可能会定制攻击输入前缀并通过执行多次推断服务来尝试恢复模型的隐私数据。

本节研究持有隐私数据集的数据持有者如何保障在执行推断阶段时，训练好的语言模型不会由于“记忆性”导致泄露出训练的隐私内容，即使用何种训练策略才能让模型主要关注于文本构成以及生成逻辑，而不是“记住”具体的隐私信息。因此，本节考虑两个实体，第一个是拥有隐私医学文本数据集的数据持有者，第二个是调用查询接口的使用者。

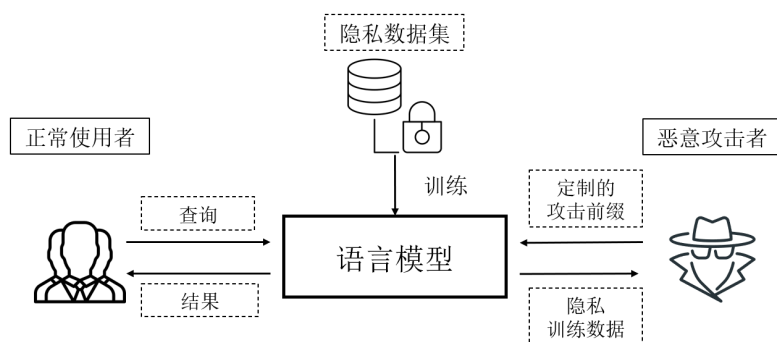


图 5.1 公布模型查询接口的风险

5.2.2 威胁模型与设计目标

(1) 威胁模型

本节对该场景下拥有隐私医学文本数据集的数据持有者的安全假设是诚实的，即它会使用真实的医学隐私训练语料通过特定的训练方式来训练模型，也称为模型持有者（后文提到的数据持有者与模型持有者在本章中都是相同的）。对于调用查询接口的使用者，本章假设其是恶意的，它可以通过定制任何攻击前缀来从模型持有者的模型中推断训练数据集的隐私信息。

(2) 设计目标

- a) 训练阶段引入差分隐私。数据持有者通过一个隐私保护的训练算法在隐私数据上进行模型训练，推断阶段直接输出推断结果。
- b) 推断阶段引入差分隐私。数据持有者直接使用隐私数据进行训练，推断阶段使用一个隐私保护的推断算法输出推断结果。

5.3 基于差分隐私算法的推断结果隐私保护方案

5.3.1 选择差分隐私定义

参考2.1.1节的介绍，考虑一个由词表 V 中的多个 tokens 组成的文本序列，即 $x = (x_1, \dots, x_n)$ ，其中 x_i 为第 i 个 token。语言建模的目标是，通过应用链式法则 $Pr(x) = \prod_{i=1}^n Pr(x_i|x_{<i})$ 构建分布的生成模型 $Pr(x)$ 。当用参数 θ 评估神经网络 f 时，本节让 $f_\theta(x_i|x_{<i})$ 表示 token x_i 的概率。通过训练最小化负对数似然函数 $L(\theta) = -\log \prod_{i=1}^n f_\theta(x_i|x_{<i})$ ，来使得模型最大化训练集 W 中数据的概率。

$$Pr(x) = \prod_{i=1}^n p(x_i|x_{<i})$$

$$L(\theta) = - \sum_{t=1}^{|D|} \sum_{i=1}^{n_t} \log p_\theta(x_i^t|x_{<i}^t)$$

由2.2中 DP 的定义，若将 DP 直接部署在医学文本生成任务上，其对所有内容进行保护，即将所有记录视为敏感的。相关工作研究了在 NLP 领域使用 DP 的一些变体，如个性化 DP^[99] 和 onesided DP^[100]。然而，现有的隐私概念不允许给定记录中的不同属性具有不同的隐私级别，特别是对于隐私属性极为稀疏的 NLP 任务。因此，我们提出了一种新的隐私概念——选择差异隐私，即使用策略函数区分一个数据样本内部的私有和非私有属性，并保护一个数据样本的私有部分。

定义 5.1 （策略函数）策略函数 $F : \tau \rightarrow \{0, 1\}^{n_r}$ 表示一个记录 $r \in \tau$ 的哪些属性是敏感的 ($F(r)_i = 0$) 或不敏感的 ($F(r)_i = 1$)，其中 n_r 是 r 中的属性数量。

其中, n_r 依赖于记录, 而不是一个固定的数。

用户可以自由定义策略函数来编码具体的隐私规定, 并根据具体应用保护任何敏感属性。受保护的敏感属性类型是无限的, 可以是实体 (如姓名、电子邮件等)、上下文 (如健康相关信息、说话风格等), 等等。例如, 用户可以设计一个保守的政策功能, 在必要时保护选定的完整句子。策略函数的形式也是无限的, 可以是神经网络、正则表达式等。

在语言建模的情况下, 每个记录是一个文本序列 x , 每个属性是 x 中的一个 token x_i , $F(x)$ 是一个位向量, 表示哪些标记包含私有信息。我们在新的隐私概念下定义如下所示的相邻数据集。

定义 5.2 (F-Neighbors) D, D' 是两个数据集, F 是一个策略函数。当且仅当 $\exists r \in D$ 使得 $F(r)$ 包含至少一个私有属性, $\exists r' \in D'$ 使得 $F(r)$ 和 $F(r')$ 至少有一个私有属性不同, 且 $D' = D$

$r \cup r'$ 时, 我们称 D' 是 D 的相邻数据集。我们简记为 $D' \in N_F(D)$ 。

在这个定义下, 包含“我的 ID 是 123”的数据集和包含“我的 ID 是 456”的数据集是相邻的。但带有“Hello there”的数据集和带有“Hi there”的数据集不是邻居, 因为它们不包含隐私信息。

定义 5.3 (选择差分隐私^[45]) 给定一个策略函数 F 。对于 $\forall D, D' \in N_F(D)$ 以及 $\forall T \subseteq R$, 如果 $Pr[M(D) \subset T] \leq e^\epsilon Pr[M(D') \subset T] + \delta$, 我们称随机算法 $M : D \rightarrow R$ 满足 (F, ϵ, δ) -Selective DP。

本质上, 选择性差分隐私也提供了类似于规范 DP 的不可区分性, 但只针对记录中的敏感属性。只要保留敏感属性的隐私, 选择性差分隐私并不约束非敏感属性的信息泄露。因此, 选择性差分隐私在最坏的情况下 (即攻击者可能知道除目标敏感属性之外的所有信息) 保护敏感属性的隐私。

5.3.2 针对训练阶段的选择差分隐私优化器

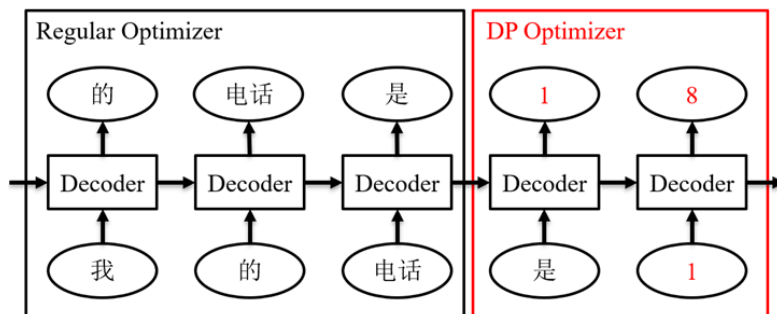


图 5.2 差分隐私训练优化器

本节介绍针对训练阶段的选择性差分隐私优化器。虽然使用选择性差分隐私机制的相关工作^[45]已在 RNN 式的语言模型上实施, 但 RNN 式的模型按照

token 出现的顺序从前往后滑动处理, 这种情况下较容易分开处理隐私 tokens 与非隐私 tokens。而基于 Transformer 结构的模型中, 所有 tokens 是并行处理的, 因此无法直接将研究工作^[45]中的 RNN 式处理方法应用于基于 Transformer 的模型。接下来, 本部分将详细介绍针对训练阶段的选择性差分隐私优化器。

如图5.2和算法5.1所示, 该优化器用于训练基于 Transformer (以下简称为 Model) 的语言模型, 实现选择差分隐私。其基本思路首先使用策略函数确定私有属性, 然后确定哪些模型变量与私有属性相关, 最后对非私有变量应用常规 Optimizer, 对私有变量应用 DP-Optimizer (如图5.2所示)。

这里需要首先确定与私有 tokens 相关的变量。Model 使用一个隐藏状态 h_i 对上下文进行编码, 输出在词表 V 上的一个分布 p_i 。如果 x_i 是私有的, 那么 h_i 、 p_i 和 L_i 都是私有的。

算法5.1概述了 DP-Optimizer 中的步骤。给定一个数据集 D , 我们应用一个策略函数 F 来获得一个比特矩阵 $P = F(D)$, 表示哪些 tokens 是私有的。在每一步, 我们取一个随机 batch B , 并使用 P 将 B 拆分成一个非私有和私有元组的序列 $(B_{np,i}, B_{p,i})$; 然后我们在 $B_{p,i}$ 上应用 Optimizer (常规更新), 在 $B_{np,i}$ 上应用 DP-Optimizer (隐私更新), 交替进行, 以更新和保护隐私。注意, 除了梯度中的噪声, 如果隐藏状态 h_i 是私有的, 我们还对其裁剪和添加噪声。原因是在 LM 中, 如果 x_i 是私有的, h_i 也包含私有信息 (如图5.2所示), 并且直接传递给下一个常规更新步骤, 无法被梯度中的噪声保护。所以在 h_i 中加入噪声来保护隐私信息是很重要的。由于 DP-Optimizer 对梯度增加了噪声, 用于计算损失的 L 和 p_i 被梯度中的噪声保护。这样一来, 所有的私有变量都得到了保护。

算法 5.1 选择差分隐私优化器

输入: N 个样本的数据集 D , 策略函数 F , 隐私矩阵 $P = F(D)$, 最大序列长度 K , 损失函数 $L(\theta)$, 超参数: 学习率 η 、噪声指数 σ 、梯度裁剪上界 C 、批大小 B , 模型参数

Model = {Token_Embedding, Encoder, Vocab_Projection}

输出: 模型经过一轮训练更新后的参数 Model

```

1  for  $j = 1, 2, \dots, B$  do
2      for  $k = 1, 2, \dots, \text{len}(\text{Sen})$  do
3          记  $\text{Batch\_input}[i]$  的第  $j$  个句子  $\text{Batch\_input}[i][j]$  为  $x$ 
4          使用隐私矩阵  $P = F(\text{input})$  将  $B$  中的数据拆分成非隐私元组与
              隐私元组:  $\{(M_{np}, M_p)\}$ 
5           $E = \text{Token\_Embedding}(x)$ 
6           $\text{Out\_E} = \text{Encoder}(E)$ 
7          for  $x_k \in M_p$  do
8               $\text{Out\_E}_p \leftarrow \text{Out\_E}_p / \max(1, \frac{\|\text{Out\_E}\|_2}{C})$ 
9               $\text{Out\_E}_p \leftarrow \text{Out\_E}_p + \sigma C \cdot N(0, I)$ 
10         end
11          $\text{Logits}[j] = \text{Vocab\_Projection}(\text{Out\_E}[-1])$ 
12     end
13     for  $k = 1, 2, \dots, \text{len}(\text{Sen}) - 1$  do
14         if  $k \in M_p$  then
15              $g(k) \leftarrow \nabla_{\theta} L(\theta, \text{Logits}[k], x[k+1])$ 
16              $g(k) \leftarrow g(k) / \max(1, \frac{\|g\|_2}{C})$ 
17              $g(k) \leftarrow \frac{1}{|M_p|} (\sum_j g(k) + \sigma C \cdot N(0, I))$ 
18              $\theta \leftarrow \theta - \eta g(k)$ 
19         end
20         else
21              $\theta \leftarrow \theta - \eta \nabla_{\theta} L(\theta)$ 
22         end
23     end
24 end

```

5.3.3 针对推断阶段的选择差分隐私解码算法

如图5.3与算法5.2概述了针对推断阶段的解码算法。由于该场景下训练阶段使用原始包含隐私内容的数据集进行训练，在推断阶段时，为保护隐私内容，需要对生成结果进行处理。具体来说，在生成输出的过程中，使用策略函数 F 对已生成的内容与当前模型输出的下一个 Token 进行判断，若属于隐私内容，则对 Logits 进行裁剪并加上由隐私预算与裁剪上界确定的高斯噪声，并重新经过分词器解码得到新的 Token。

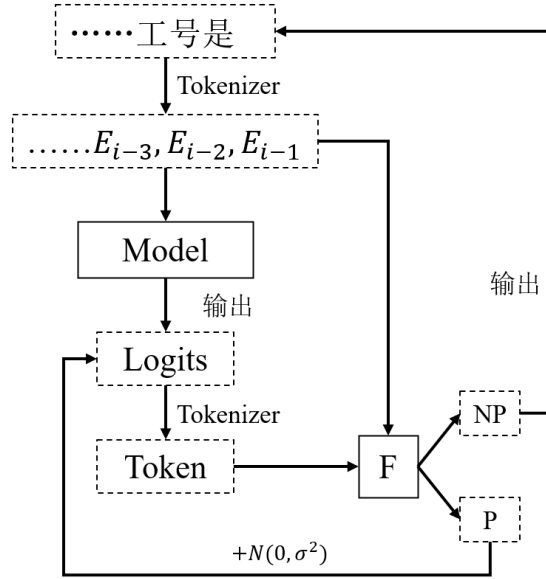


图 5.3 差分隐私解码算法

在文本生成任务中，生成的下一个单词通常是通过计算 Logits 值，再使用 Softmax 函数将 Logits 转化为概率分布来确定的。在差分隐私的场景下，需要对 Logits 进行处理以满足有界的全局敏感度。具体地，我们采用归一化处理将 Logits 限制在 0 到 1 之间。对于每个 Logit l_i ，我们计算其归一化后的值 $l'_i = \frac{l_i - l_{\min}}{l_{\max} - l_{\min}}$ ，其中 l_{\min} 和 l_{\max} 分别是 Logits 中的最小值和最大值。这样，归一化后的 Logits l'_i 满足 $l'_i \in [0, 1]$ ，即全局敏感度 $C = 1$ 。

需要注意的是，在使用 Softmax 函数来确定下一个单词时，Softmax 函数不改变最大值的位置，因此选择 $\text{argmax}(\text{Softmax}(\text{Logits}))$ 与 $\text{argmax}(\text{Logits})$ 的结果相同。由于 $\text{Softmax}(\text{Logits})$ 的输出结果在 $(0, 1)$ 范围内，即有界，而 Logits 的取值范围是实数集，即无界，因此需要对 Logits 进行归一化处理以满足有界的差分隐私全局敏感度。

算法 5.2 选择差分隐私解码算法

输入: 输入解码的前缀 P , 策略函数 F , 裁剪上界 C 、组大小 L , 模型 $Model$, 词表 V , 分词器 $Tokenizer(\cdot)$

输出: 输入解码的前缀为 $Prefix$ 时, 模型的输出结果 $Output$

```

1 # 将当前生成结果记为 CS, 初始赋值为输入前缀
2  $CS = P$ 
3 # 模型输出预测下一个 Token 的 Logits
4  $NT\_Logits = Model(CS)$ 
5 # Logits 确定最大概率的 index, 并由分词器解码成输出字符
6  $NT = Tokenizer(\argmax(\text{Softmax}(NT\_Logits)))$ 
7 # 若输出 <EOS> 符号终止流程
8 while  $NT \neq \text{<EOS>}$  do
9     # 若已生成内容与当前预测的下一个 Token 在策略函数  $F$  下是隐私
        内容, 则需要加噪处理
10    if  $F(CS, NT) == \text{True}$  then
11        # 对 Logits 进行裁剪
             $NT\_Logits \leftarrow NT\_Logits / \max(1, \frac{\|NT\_Logits\|_2}{C})$ 
12        # 加噪
             $NT\_Logits \leftarrow NT\_Logits + \sigma C \cdot N(0, I)$ 
13        # 在加噪的 Logits 下重新生成 Token
             $NT = Tokenizer(\argmax(\text{Softmax}(NT\_Logits)))$ 
14    end
15    # 将新 Token 加入当前生成结果 CS 中
16     $CS = CS + NT$ 
17    # 继续预测下一个 Token 的 Logits
18     $NT\_Logits = Model(CS)$ 
19    # 由分词器对 index 进行解码
20     $NT = Tokenizer(\argmax(\text{Softmax}(NT\_Logits)))$ 
21 end

```

5.4 安全性分析

本部分给出算法5.1的隐私分析。

对于任何给定的数据集 D , 设 $D_{i,j}$ 表示第 i 条记录的第 j 个属性。我们将梯度更新和隐藏状态抽象为以训练数据 x 和辅助信息 w 为输入的查询函数 $f(x, w)$ 。

我们引入 w 作为 f 的额外输入，以模拟梯度更新和隐藏状态对前几轮模型参数的依赖关系。我们在数据集上定义以下两种类型的查询。

- 类型 1: 函数 f 的输入只包含策略函数 F 判定为隐私信息的查询 x
- 类型 2: 函数 f 的输入只包含策略函数 F 判定为非隐私信息的查询 x

由于算法 5.1 仅针对隐私信息进行保护，因此类型 2 的非隐私查询不会造成隐私损失。

下面的定理表明，如果一个类型 1 查询具有输出有界的属性，那么对于任意的输入，在查询中添加高斯噪声可以提供差分隐私保障。由于在策略函数 F 下的近邻数据集的非敏感部分可能是不同的，因此我们需要分析在任意辅助输入下的 DP 保证。

定义 5.4 (隐私损失^[47]) 对于任意近邻数据集 D 与 D' ，独立的辅助输入 w ，算法 M 的输出结果为 y ，定义隐私损失如下：

$$L(y; M, w, D, D') = \ln \frac{\Pr[M(w, D) = y]}{\Pr[M(w, D') = y]} \quad (5.1)$$

定理 5.1 记 $\Delta_2 f = \max_{\{D, D'\}} \|f(D) - f(D')\|_2$ 函数 f 的敏感度， $N(0, \sigma^2)$ 为由参数 σ 控制的高斯分布，对于 $c^2 > 2\ln(1.25/\delta)$ ，具有 $\delta \geq c\Delta_2 f/\epsilon$ 的高斯机制满足 (ϵ, δ) -差分隐私。

证明 对于数据集 D 与函数 f ，高斯机制计算结果为 $f(D) + N(0, \sigma^2)$ ，其中 $N(0, \sigma^2)$ 为均值为 0，标准差为 σ 的高斯分布。考虑如下表达式：

$$\left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta f)^2}} \right| \quad (5.2)$$

这个式子是隐私损失的绝对值。假设数据集是 D ，为证明高斯机制满足 (ϵ, δ) -差分隐私，则需观察在 D 下与在其近邻数据集 D' 下，输出结果非常不同时的概率。上式中的分子描述了当数据集为 D 时看到 $f(D) + x$ 的概率，分母对应的是当数据集为 D' 时看到这个相同值的概率，即该分式为非负的概率的比值，但其对数可能是负的。为方便起见，我们研究隐私预算的绝对值。

$$\begin{aligned} \left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta f)^2}} \right| &= \left| \ln e^{(-1/2\sigma^2)[x^2 - (x+\Delta f)^2]} \right| \\ &= \left| -\frac{1}{2\sigma^2} [x^2 - (x + \Delta f)^2] \right| \\ &= \left| -\frac{1}{2\sigma^2} [x^2 - (x^2 + 2x\Delta f + \Delta f^2)] \right| \\ &= \left| \frac{1}{2\sigma^2} (2x\Delta f + \Delta f^2) \right| \end{aligned} \quad (5.3)$$

上式结果在 $x < \sigma^2\epsilon/\Delta f - \Delta f/2$ 时可以由 ϵ 约束。为确保隐私损失在 $1 - \delta$ 的概率下不超过隐私预算 ϵ ，我们需要

$$\Pr[|x| \geq \sigma^2\epsilon/\Delta f - \Delta f/2] < \delta$$

去掉绝对值，即意味着我们需要找到 σ 使得

$$\Pr[x \geq \sigma^2 \epsilon / \Delta f - \Delta f / 2] < \delta / 2$$

假设 $\epsilon \leq 1 \leq \Delta f$ ，利用

$$\Pr[x > t] \leq \frac{\sigma}{\sqrt{2\pi}} e^{-t^2/2\sigma^2}$$

即需要

$$\begin{aligned} \frac{\sigma}{\sqrt{2\pi}} \frac{1}{t} e^{-t^2/2\sigma^2} < \delta / 2 &\iff \sigma \frac{1}{t} e^{-t^2/2\sigma^2} < \sqrt{2\pi} \delta / 2 \\ &\iff \frac{t}{\sigma} e^{t^2/2\sigma^2} > 2 / \sqrt{2\pi} \delta \\ &\iff \ln(t/\sigma) + t^2/2\sigma^2 > \ln(2/\sqrt{2\pi} \delta) \end{aligned}$$

令 $t = \sigma^2 \epsilon / \Delta f - \Delta f / 2$ ，即有

$$\begin{aligned} \ln((\sigma^2 \epsilon / \Delta f - \Delta f / 2) / \sigma + (\sigma^2 \epsilon / \Delta f - \Delta f / 2)^2 / 2\sigma^2) &> \ln(2/\sqrt{2\pi} \delta) \\ &= \ln(\sqrt{\frac{2}{\pi}} \frac{1}{\delta}) \end{aligned}$$

记 $\sigma = c \Delta f / \epsilon$ ，我们希望约束 c ，记找到第一项非负的条件。

$$\begin{aligned} \frac{1}{\sigma} (\sigma^2 \frac{\epsilon}{\Delta f} - \frac{\Delta f}{2}) &= \frac{1}{\sigma} [(c^2 \frac{(\Delta f)^2}{\epsilon^2}) \frac{\epsilon}{\Delta f} - \frac{\Delta f}{2}] \\ &= \frac{1}{\sigma} [c^2 \frac{\Delta f}{\epsilon} - \frac{\Delta f}{2}] \\ &= \frac{\epsilon}{c \Delta f} [c^2 \frac{\Delta f}{\epsilon} - \frac{\Delta f}{2}] \\ &= c - \frac{\epsilon}{2c} \end{aligned}$$

由于 $\epsilon \leq 1$ 且 $c \geq 1$ ，即 $c - \epsilon/c \geq c - 1/2$ ，故当 $c \geq 3/2$ 时， $\ln(\frac{1}{\sigma} (\sigma^2 \frac{\epsilon}{\Delta f} - \frac{\Delta f}{2})) > 0$ 。接下来关注 t^2/σ^2 这一项。

$$\begin{aligned} (\frac{1}{2\sigma^2} \frac{\sigma^2 \epsilon}{\Delta f} - \frac{\Delta f}{2})^2 &= \frac{1}{2\sigma^2} [\Delta f (\frac{c^2}{\epsilon} - \frac{1}{2})]^2 \\ &= \frac{1}{2} [(\Delta f)^2 (\frac{c^2}{\epsilon} - \frac{1}{2})]^2 [\frac{\epsilon^2}{c^2 (\Delta f)^2}] \\ &= \frac{1}{2} (\frac{c^2}{\epsilon} - \frac{1}{2})^2 \frac{\epsilon^2}{c^2} \\ &= \frac{1}{2} (c^2 - \epsilon + \epsilon^2/4c^2) \end{aligned}$$

由于 $\epsilon \leq 1$ 并且 $c \geq 3/2$ ，则 $c^2 - \epsilon + \epsilon^2/4c^2 \geq c^2 - 8/9$ ，故仅需

$$c^2 - 8/9 > 2 \ln(\sqrt{\frac{2}{\pi}} \frac{1}{\delta})$$

换言之，我们需要

$$c^2 > 2 \ln(\sqrt{\frac{2}{\pi}}) + 2 \ln(\frac{1}{\delta}) + \ln(e^{8/9}) = \ln(2/\pi) + \ln(e^{8/9}) + 2 \ln(\frac{1}{\delta})$$

且由于 $(2/\pi)e^{8/9} < 1.55$ ，上式在 $c^2 > 2 \ln(1.25/\delta)$ 时成立。

记 $R_1 = \{x \in R : |x| \leq c\Delta f/\epsilon\}$ 与 $R_2 = \{x \in R : |x| > c\Delta f/\epsilon\}$ ，易知 $R = R_1 \cup R_2$ ，取 $S \subset R$ ，并记

$$S_1 = \{f(x) + x | x \in R_1\}$$

$$S_2 = \{f(x) + x | x \in R_2\}$$

则有

$$\begin{aligned} \Pr_{x \sim N(0, \sigma^2)} [f(x) + x \in S] &= \Pr_{x \sim N(0, \sigma^2)} [f(x) + x \in S_1] \\ &\quad + \Pr_{x \sim N(0, \sigma^2)} [f(x) + x \in S_2] \\ &\leq \Pr_{x \sim N(0, \sigma^2)} [f(x) + x \in S_1] + \delta \\ &\leq e^\epsilon \left(\Pr_{x \sim N(0, \sigma^2)} [f(y) + x \in S_1] \right) + \delta \end{aligned} \tag{5.4}$$

故高斯机制满足 (ϵ, δ) -差分隐私。

■

定理 5.2 假设 $\max_{x,w} \|g(x, w)\| \leq C$ ，对于任意的 w ，添加由 C 确定的高斯分布噪声可以保证 g 满足 (ϵ, δ) -差分隐私，其中 ϵ, δ 取决于 C 与 σ 。形式化来说，对于近邻数据集 (x, x') 以及任意的 (w, w') ，有

$$\frac{P[g(x, w) + \Delta = r]}{P[g(x', w') + \Delta = r]} \leq e^\epsilon \quad w.p. \quad 1 - \delta$$

算法5.1中对于梯度信息进行了裁剪，即满足有界性，由引理5.1知，定理5.3成立。同样的，算法5.2中对于 Logits 信息进行了裁剪，即满足有界性，由引理5.1知，定理5.3成立。算法5.1对于非隐私内容 $B_{np,i}$ 的更新属于类型 2，这样的更新不会导致使用更多的隐私预算。对于隐私内容的更新（包括梯度与隐藏状态 *Hidden*）属于类型 1，称其加上满足引理5.1中的高斯噪声后的数据为“模糊的”类型 1 查询。算法5.1中的数据实际上由类型 2 与“模糊的”类型 1 构成。算法5.2中的数据同样由类型 2 与“模糊的”类型 1 构成。下面论证这样的组合满足 (ϵ, δ) -差分隐私。

定理 5.3 记是 f 为由 k 个查询 $\{f_1, \dots, f_k\}$ 构成的整体，其中 f_i 属于类型 1 或“模糊的”类型 2。给定策略函数 F ，记 f_{np} 为类型 2， f_p 为“模糊的”类型 1。那么如果 f_p 满足 (ϵ, δ) -差分隐私，则 f 满足 (F, ϵ, δ) -Selective DP。

证明 考虑在策略函数 F 下的近邻数据集 x 与 x' ，记 x_i 与 x'_i 是数据 f_i 中的子集。若 f_i 属于“模糊的”类型 1，则 x_i 只包含隐私内容；反之若 f_i 属于类型 2， x_i 只包含非隐私内容。由于 x 与 x' 是策略函数 F 下的紧邻数据集，即 f_i 属于“模糊的”类型 1。对于 f 的一个输出 y_1, \dots, y_k ，有

$$\begin{aligned} & \frac{P[f_1(x_1, w_1) = y_1, \dots, f_1(x_k, w_k) = y_k]}{P[f_1(x'_1, w'_1) = y'_1, \dots, f_1(x'_k, w'_k) = y'_k]} \\ &= \prod_{f_i \in f_p} \frac{f_i(x_i, w_i) = y_i}{f_i(x'_i, w'_i) = y'_i} \end{aligned} \quad (5.5)$$

$$\leq e^\epsilon \quad w.p. \quad 1 - \delta \quad (5.6)$$

式 5.5 是由于 f_{np} 不涉及隐私内容，并且每一个 token 的隐层表示与其对损失函数的贡献是独立的。式 5.6 中的不等式是因为 f_p 满足 (ϵ, δ) -差分隐私的假设。 ■

通过这个证明，我们可以得出结论：在给定策略函数 F 的情况下，如果“模糊的”类型 1 查询 f_p 满足 (ϵ, δ) -差分隐私，则整个查询 f 满足 (F, ϵ, δ) -选择差分隐私。这意味着，通过在梯度上添加高斯噪声，我们可以确保模型训练过程满足选择差分隐私的要求，从而保护数据集中个体的隐私。

5.5 实验评估

5.5.1 攻击方式

我们执行两种类型的攻击：“诱饵”插入攻击和成员推断攻击。

(1) “诱饵”插入攻击^[101]

“诱饵”（Canary）插入攻击是一种针对训练数据的隐私攻击方法，为一种定量评估意外记忆风险的测试方法。攻击者在训练数据集中插入一些特制的“诱饵”数据（即 Canary），这些数据通常具有特定的模式或特征，使其在整个数据集中独特而容易识别。通过将随机序列 Canary 插入训练数据集，并在此基础上训练模型。攻击者的目标是通过分析模型的输出结果，识别出模型是否泄露了这些插入的 Canary 数据。

在执行 Canary 插入攻击时，攻击者首先需要构建一些含有特定信息的诱饵数据。这些数据应该具有一定的复杂性，以便在模型生成的输出中能够辨别出其所学习到的信息。攻击者将这些诱饵数据插入到训练数据集中，并记录下与这些诱饵数据相关的标签。然后，攻击者观察模型在特定输入下的输出结果，判断模型是否泄露了插入的 Canary 数据。

如果模型在生成输出时泄露了 Canary 数据，那么攻击者可以根据这些信息推断出模型的训练数据中可能包含了这些插入的诱饵数据。从而揭示了模型对训练数据的记忆情况，进而可能导致训练数据的隐私泄露。可以通过计算插入的 Canary 作为一种定量指标，以衡量模型潜在的隐私风险。其中定量的评估指标定义如下。

定义 5.5 (Canary 暴露度) 给定一个 Canary $s[r]$ ，一个参数为 θ 的模型，随机空间 R ， $s[r]$ 的暴露程度是： $Exposure_{\theta} = \log_2|R| - \log_2 Rank_{\theta}(s[r])$

训练后，我们计算所有可能实例化的 Canary 的模型困惑度，并将其按照困惑度排序。然后根据特定 Canary 序列的 $Rank_{\theta}(s[r])$ 和所有可候选的数量 $|R|$ 得到 Canary 的暴露量。从定义来看，当 Canary 在所有输出的排名越靠前，则暴露的程度就越高，而 rank 高意味着后面 $Rank_{\theta}(s[r])$ 的值越低，则 $\log_2|R| - \log_2 Rank_{\theta}(s[r])$ 的值越高。因此，若模型的 Canary 暴露度越高，则它记忆训练样本的可能性越高，即暴露程度越高。反之，若模型的 Canary 暴露度很低，则说明该模型更倾向于学习到模式而非具体的内容，即对训练数据的隐私保护程度较高。在我们的设置中，我们显示了 10 个 Canary 中最高的 Canary 暴露。例如，如果一个 Canary 在 100 万个候选中排名第一，那么 Canary 的暴露量是 19.93。如图 5.4 所示，当在训练过程中插入的 Canary 为“我的单号是 #3006……”并在之前输入确定时，将 LM 输出的 logits 按照 Softmax 后的 P 概率值排序，找到与 Canary 匹配的 token 的 rank 排名，作为攻击的度量指标。

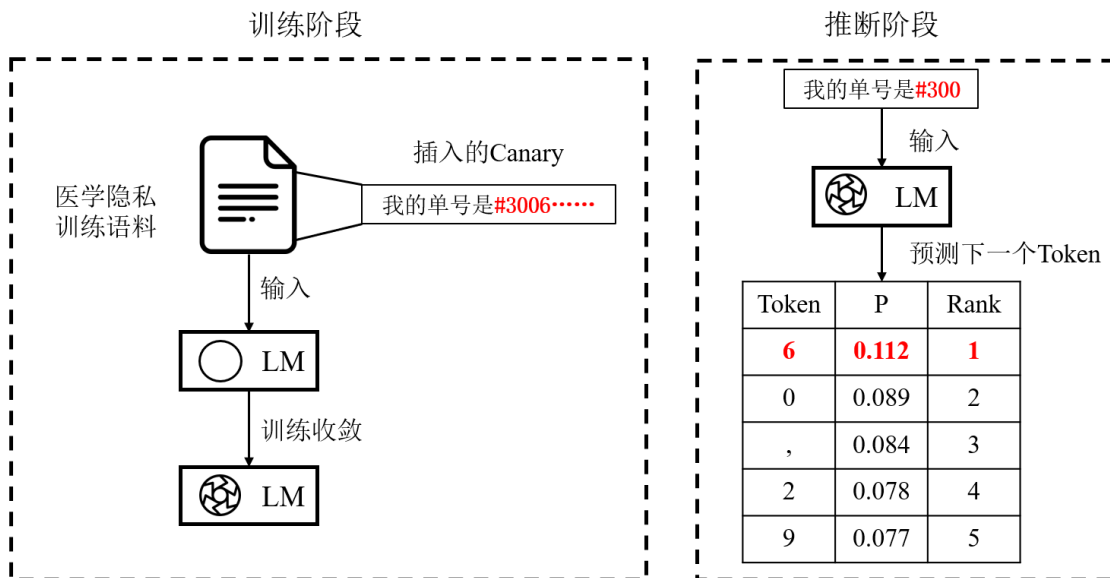


图 5.4 Canary 插入攻击与衡量

(2) 模型反演攻击

与 3.3.1 节相同，本实验假设恶意攻击者对于模型执行黑盒攻击，即攻击者只能从输入与模型的输出关系来推测隐私信息。

5.5.2 实验设置

实验环境与3.3.1相同，如表4.1所示：CPU为AMD Ryzen 9 5900HX、32GB RAM、GPU为RTX3080-Laptop、操作系统为Windows 11 64位。

与3.3.1节中的设定相同，本节使用Chinese medical dialogue data (CMDD)中文医疗对话数据集来训练LM。其参数量为81.9M，使用的词表大小为21128，隐层维度为768，12层GPT2Block。与相关工作的设定相同^[45]，我们将电话号码、年龄、单号、药物计量、检测的定量结果等数字内容视为敏感信息，并使用正则表达式构建一个策略函数来检测它们。差分隐私中设置 $\epsilon = 0.5, \delta = \frac{1}{N} = 1e-6$ （为数据集大小 N 倒数的量级）。本章的实验基于PyTorch的差分隐私库^[102]，以实现相关算法的设计。

为对比本章提出的选择差分隐私的效果，本实验选择两个模型作为对比：

- a) 无隐私保护(No_DP)。这里直接使用原始的训练数据进行训练，可以视为隐私预算 $\epsilon = +\infty$ 。这种情况即为3.3.1节中在原始中文预训练模型基础上，利用CMDD数据上微调的模型。
- b) 对所有文本进行差分隐私保护(All_DP)。在这种情况下，把数据集所有的文本当成需要保护的对象。这可以视为选择差分隐私的最坏情况，即 $\forall d_i \in D$ ，策略函数 F 输出 $F(d_i) = 1$ 。

下面介绍上述两种攻击方式的实验设定。

(1) “诱饵”插入攻击

实验中，以“我的单号是<随机的6位数字>”的形式随机生成了5个Canary：“我的单号是541684”、“我的单号是946241”、“我的单号是197462”、“我的单号是678409”、“我的单号是209118”。每个Canary独立测试，即一个Canary对应一个训练模型。每个实验中，在训练数据集中插入10次Canary（这是一个常见的设定，如^[45]），即在3.2节中定义的10-清晰记忆。

本节以3.3.1节中的中文预训练模型为基础，在CMDD数据集上进行微调训练的。分别在64278条训练数据中插入上述Canary 10次，并训练25个epoch，计算在未加保护的情况下，模型的Canary暴露度。

(2) 模型反演攻击

与2节的设定相同，本实验使用随机采样的10个训练数据的前10个token作为前缀输入，使用训练样本推断攻击中的方式分别进行解码，测试其完整恢复训练数据的次数。具体来说，对于每个前缀，本节对上述每个前缀生成10000个解码结果，针对其进行平均统计（平均值为分数则向下取整）。

5.5.3 实验结果

(1) “诱饵”插入攻击

实验中，以“我的单号是 < 随机的 6 位数字 >”的形式随机生成了 5 个 Canary: “我的单号是 541684”、“我的单号是 946241”、“我的单号是 197462”、“我的单号是 678409”、“我的单号是 209118”。每个 Canary 独立测试，即一个 Canary 对应一个训练模型。每个实验中，在训练数据集中插入 10 次 Canary（这是一个常见的设定，如^[45]），即在3.2节中定义的 10-清晰记忆。

与1节相同，本节也是以3.3.1节中的中文预训练模型为基础，在 CMDD 数据集上进行微调训练的。分别在 64278 条训练数据中插入上述 Canary 10 次，并训练 25 个 epoch，计算在未加保护的情况下，模型的 Canary 暴露度。

在输入的 Canary 为“我的单号是 541684”时，对没有使用任何隐私保护技术的模型执行攻击，其 Canary 暴露度如表5.1所示。其中，以“‘我的单号是’ + 前缀”表示实际输入模型的完整前缀。

表 5.1 不同前缀下 LM 的 Canary 暴露度

| 位数 | 前缀 | 排名 | 总可能数 | Canary 暴露度 |
|----|-------|--------|------|------------|
| 0 | NULL | 390111 | 1e6 | 1.3588 |
| 1 | 5 | 34986 | 1e5 | 1.5152 |
| 2 | 54 | 2597 | 1e4 | 1.9451 |
| 3 | 541 | 118 | 1e3 | 3.0831 |
| 4 | 5416 | 9 | 1e2 | 3.4739 |
| 5 | 54168 | 1 | 1e1 | 3.3219 |

从表5.1可以看出，在没有隐私保护时，前缀与原训练样本匹配度越高（这里指提供的前缀长度越长），其 Canary 暴露度越高，即模型越有可能恢复出原始数据，与预期相符。

对于全部 5 组 Canary，其暴露度的平均情况如图5.5所示，其中 No_DP 指在训练与推断阶段均未使用隐私保护技术的模型，All_DP 指对所有训练样本使用 DP 训练的模型，Seletive_DP_Train 指对训练样本使用上述实验设置中定义的策略函数的选择差分隐私训练优化器训练的模型，Selective_DP_Decode 指在训练阶段未使用 DP 而推断阶段使用选择差分隐私解码算法的模型。

可以从图5.5看出，No_DP 的 Canary 暴露度最高，而 All_DP 的 Canary 暴露度最少；Seletive_DP_Train 与 Seletive_DP_Decode 表现效果相似，介于 All_DP 与 No_DP 之间且表现效果更接近 All_DP。因此可以证明 Seletive_DP_Train 与 Seletive_DP_Decode 均对训练数据的隐私起到了保护作用，保护程度都接近与 All_DP 的效果。

表5.2，表示 No_DP、All_DP、Seletive_DP_Train 与 Seletive_DP_Decode 的困惑度指标。由定义3.3可知，困惑度是描述模型对于测试样本语句整体的“惊讶”程度，若模型效果好，那么测试的文本对模型而言就很正常，困惑度就不会很高，反之亦然。

表5.2中不同情况差别很大。这主要是与 PPL 的计算方式有关。PPL 可以视

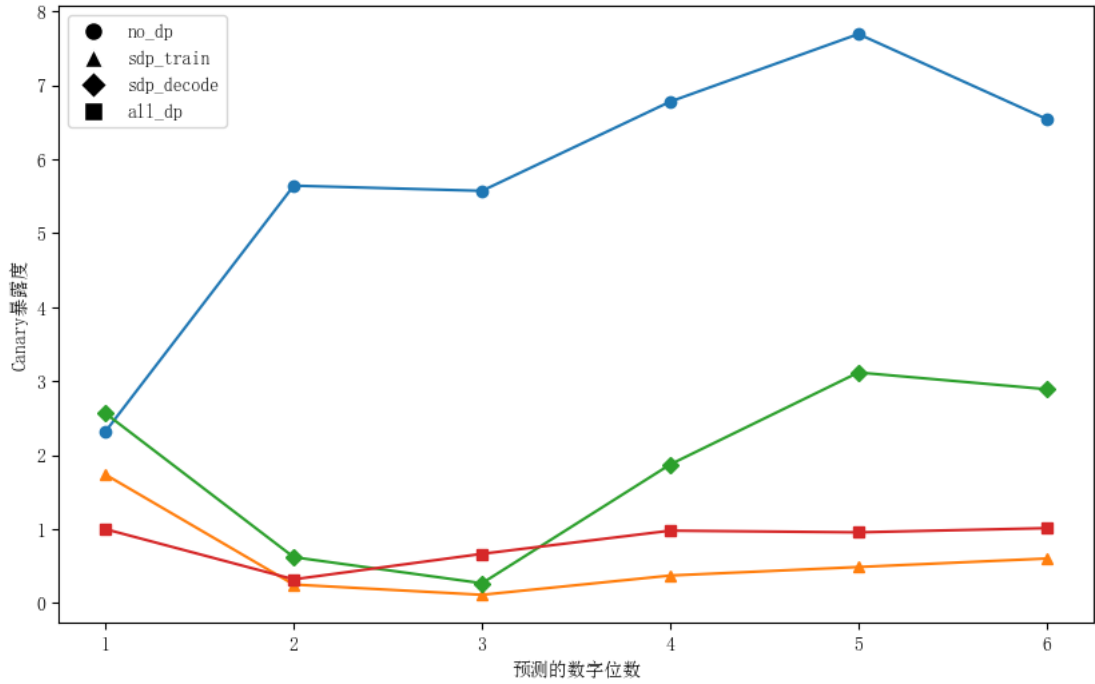


图 5.5 各模型的 Canary 暴露度

表 5.2 不同方式下的模型困惑度

| 方式 | 困惑度 |
|--------------------|-------|
| No_DP | 11.22 |
| All_DP | 59.75 |
| Seletive_DP_Train | 21.32 |
| Seletive_DP_Decode | 13.58 |

为模型对测试数据集中每句话的交叉熵 Loss 的指数结果的均值，即 $PPL(D) = \frac{1}{N} \sum_{i=1}^N \exp(\text{Loss}(\text{Model}(S_i)))$ ，其中 $D = \{S_1, S_2, \dots, S_N\}$ 。在 CMDD 数据集上训练的 Loss 在 2-4 之间，由交叉熵的定义可知，模型平均 Loss 为 2 即相当于在 $e^2 = 7.389$ 个 Token 中随机猜测，而 Loss 为 4 即相当于在 $e^4 = 54.598$ 个 Token 中随机猜测，相比与词表大小 13317，该预测结果较好。那么在这种情况下的 PPL 的变化就会从 $e^2 = 7.389$ 到 $e^4 = 54.598$ ，因此上述 PPL 的范围也符合预期。

从上述结果可以看出，No_DP 的 PPL 最低，即模型对测试数据不“惊讶”，意味着该模型生成效果最好，而 All_DP 的效果最差，是因为它将所有文本视为隐私信息，而忽略了大部分内容是不敏感的。有趣的是虽然前面 Seletive_DP_Train 与 Seletive_DP_Decode 的 Canary 暴露度相近，而 Seletive_DP_Decode 的 PPL 要比 Seletive_DP_Train 低了 36%，这是由于 Seletive_DP_Decode 的训练过程是正常的，只是在推断阶段在生成的隐私内容上加噪，而 Seletive_DP_Train 虽然只是对隐私部分的内容加噪，但是其在隐私内容的语义范式上加噪会降低模型的表达能力。另一方面，由于上面 PPL 差异大的分析，回到平均 Loss 的空间下，二者分别为 2.609 与 3.059，这在图3.6所示的训练情况下差别并不大。

(2) 模型反演攻击

表5.3, 表示 No_DP、All_DP、Seletive_DP_Train 与 Seletive_DP_Decode 情况下的成员推断成功次数。从中可以看出使用 All_DP 取得了 10000 个生成样本中没有任何成功恢复的效果, 而 Seletive_DP_Train 和 Seletive_DP_Decode 仅比 No_DP 的成功次数少一点, 与 All_DP 之间的差别还是很大。

表 5.3 攻击方式与成功次数

| 类型 | 成功次数 |
|--------------------|------|
| No_DP | 14 |
| All_DP | 0 |
| Seletive_DP_Train | 13 |
| Seletive_DP_Decode | 11 |

产生这种现象的也是符合逻辑与预期的, 主要原因如下:

- a) 由于在5.5.2节的设定下, 策略函数仅将数字部分当作隐私内容, 对其使用相应的 DP 方法处理, 非数字内容占比很大, 导致模型更容易记忆住这些非数字内容。因此, 虽然 Seletive_DP_Train 与 Seletive_DP_Decode 采用相应的符合选择差分隐私定义的步骤进行处理, 但是保护的内容较少, 成员推断攻击只关注于某句子是否在训练语料中出现, 所以这两者的保护效果相对于 No_DP 差别很小。相反, All_DP 对所有的样本都执行了差分隐私处理, 因此面对成员推断攻击这种需要对样本整体内容验证的方法保护效果较好。
- b) CMDD 的训练语料相对于预训练模型的语料较少, 在微调的过程中, 由于预训练语料的分布差别很大的医学文本语料具有特殊性, 在多轮训练后模型会逐渐记住这种风格类型于具体的数据内容, 因此在数据量较少的数据集上微调会让模型更容易记住该数据集。

为防止攻击者在推断阶段试图通过执行输入和标签重构攻击, 以恢复训练隐私数据, 同时保持语言模型的表现效果, 本章提出了一个基于差分隐私的新颖的隐私保护算法——选择差分隐私算法。首先, 本章将介绍系统模型与设计目标, 引入本章的保护对象与攻击者的行为。其次, 介绍选择差分隐私的定义, 并针对训练与推断阶段分别设计了隐私优化器与解码算法, 作为两种提供选择差分隐私的方式。随后, 对前述设计的隐私优化器与解码算法进行隐私性分析, 以证明其满足差分隐私的定义。最后, 通过设计实验说明选择差分隐私以及这两种保护方式的优势。

5.6 本章小结

本章首先对该场景下的系统模型与设计目标进行介绍, 随后引入了选择差分隐私的概念, 并针对并针对训练与推断阶段分别设计了隐私优化器与解码算法, 作为两种提供选择差分隐私的方式。在理论分析完这两种方式的安全性后,

通过基于预训练模型在 CMDD 数据集上微调的实验，将各种设定下的结果进行对比分析，证明了本章提出的选择差分隐私的优势。

第6章 总结与展望

本章首先对本文的工作进行一个全面的总结，然后对其中的不足之处进行分析和讨论，并且对未来的工作进行展望。

6.1 工作总结

本文从三个方面全面探讨了医学文本生成任务的隐私保护问题，为实际应用中的隐私保护提供了理论基础和实践指导。

(1) 医学文本生成任务的隐私攻击模型研究

本研究首先关注了医学文本生成任务在训练和推断阶段的隐私泄露风险，详细阐述了语言模型的生成过程及其记忆问题。针对公开的预训练模型实施了模型反演攻击，并提出了一些改进的攻击策略。同时，探讨了攻击者在训练阶段可能采用的攻击手段，通过实验分析了攻击效果，展示了语言模型记忆问题带来的隐私挑战。

(2) 医学文本生成任务训练阶段的隐私保护研究

针对训练阶段的隐私保护问题，本研究明确了系统模型和威胁模型，并设计了安全目标。扩展了基于秘密共享的协议，使其能够构建复杂的 Transformer 结构。通过多方计算手段来保障数据机密性，利用可信硬件 Intel SGX 确保执行过程的完整性。为提高协议的执行效率，设计了一个可验证的外包计算方法。通过安全性分析和实验验证，证明了协议的有效性和高效性。

(3) 医学文本生成任务推断阶段的隐私保护研究

在推断阶段的隐私保护研究中，为阻止攻击者实施模型反演攻击以恢复训练隐私数据，同时保持语言模型的性能效果，本研究基于差分隐私提出了两种缓解医学文本生成任务语言模型的技术。针对训练与推断阶段分别设计了隐私优化器与解码算法，进行了隐私性分析，并通过实验验证了这两种保护方法的优势。

6.2 未来展望

本文对医学文本生成任务的隐私保护进行了深入研究，对于训练与推断阶段面临的攻击与效果进行了分析，并分别提出了隐私保护方法。尽管本文已经提出了相对安全的解决方案，但由于医学文本生成任务的复杂性，本文既要考虑语言模型相关的研究进展，也要跟进隐私保护技术的发展。此外，还受到公开数据集与预训练模型的制约。因此本文的工作在一些方面仍然存在局限性，需要进一

步的深入研究。以下的方向可以作为未来研究的指引:

(1) 研究更大规模语言模型的记忆问题

本文基于公开预训练模型,在医学文本数据集上进行微调训练,以得到针对该领域的语言模型。受困于训练资源以及训练语料,本文的语言模型相对于前沿的语言模型在规模上差距很多。因此,未来工作可以针对更大规模的语言模型进行记忆问题的分析,并使用本文提出的隐私保护算法来缓解记忆问题。

(2) 提升多方安全计算函数协议的效率

由于在训练阶段中,数据隐私是最重要的考量,因此在满足同样隐私设定情况下提升协议效率是一项重要的问题。由于深度学习模型的特殊性,可以考虑在模型的部分层上执行一些参数量化与裁剪,以减少参数量与计算量。同时,针对该场景下的多方安全假设协议进行优化,减少各方之间的交互与执行时间。

(3) 探索引入差分隐私的位置对语言模型表达能力的影响

本文分别对训练阶段与推断阶段加入差分隐私来缓解医学文本生成任务的语言模型的记忆问题。而在相同的隐私预算与加噪方式下,在模型的执行流程中,如分词编码、词向量、编码器与解码器的各个模块、最后映射到词表的线性变换等环节中加入差分隐私,模型表现效果的区别(如在损失与困惑度等指标下)仍需进一步研究。通过对这些问题的深入探究,可以使隐私保护算法在保护隐私的同时,尽量降低对模型效果的影响。

参 考 文 献

- [1] GUAN J, LI R, YU S, et al. Generation of synthetic electronic medical record text[C]//2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2018: 374-380.
- [2] MELAMUD O, SHIVADE C. Towards automatic generation of shareable synthetic clinical notes using neural language models[J]. NAACL HLT 2019, 2019: 35.
- [3] 邓露, 胡珀, 李炫宏. 知识增强的生物医学文本生成式摘要研究[J]. 数据分析与知识发现, 2022: 1.
- [4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [5] KENTON J D M W C, TOUTANOVA L K. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of NAACL-HLT. 2019: 4171-4186.
- [6] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [Z].
- [7] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
- [8] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning[J]. Foundations and Trends® in Machine Learning, 2021, 14(1-2): 1-210.
- [9] DWORK C. Differential privacy: A survey of results[C]//Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi' an, China, April 25-29, 2008. Proceedings 5. Springer, 2008: 1-19.
- [10] CRAMER R, DAMGÅRD I B, et al. Secure multiparty computation[M]. Cambridge University Press, 2015.
- [11] ACAR A, AKSU H, ULUAGAC A S, et al. A survey on homomorphic encryption schemes: Theory and implementation[J]. ACM Computing Surveys (Csur), 2018, 51(4): 1-35.
- [12] SABT M, ACHEMLAL M, BOUABDALLAH A. Trusted execution environment: what it is, and what it is not[C]//2015 IEEE Trustcom/BigDataSE/Ispa: Vol. 1. IEEE, 2015: 57-64.
- [13] TANG H, GAN S, ZHANG C, et al. Communication compression for decentralized training [J]. Advances in Neural Information Processing Systems, 2018, 31.
- [14] SUI D, CHEN Y, ZHAO J, et al. Feded: Federated learning via ensemble distillation for medical relation extraction[C]//Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). 2020: 2118-2128.
- [15] 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.
- [16] CARLINI N, TRAMER F, WALLACE E, et al. Extracting training data from large language models.[C]//USENIX Security Symposium: Vol. 6. 2021.
- [17] GEHMAN S, GURURANGAN S, SAP M, et al. Realtoxicityprompts: Evaluating neural toxic degeneration in language models[C]//Findings of the Association for Computational Linguistics: EMNLP 2020. 2020: 3356-3369.
- [18] ZHANG C, IPPOLITO D, LEE K, et al. Counterfactual memorization in neural language models[A]. 2021.
- [19] BROWN H, LEE K, MIRESHGHALLAH F, et al. What does it mean for a language model to preserve privacy?[C]//2022 ACM Conference on Fairness, Accountability, and Transparency. 2022: 2280-2292.
- [20] GOLOVANOV S, KURBANOV R, NIKOLENKO S, et al. Large-scale transfer learning for natural language generation[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 6053-6058.
- [21] DABRE R, CHU C, KUNCHUKUTTAN A. A survey of multilingual neural machine translation[J]. ACM Computing Surveys (CSUR), 2020, 53(5): 1-38.
- [22] FENG Q, HE D, LIU Z, et al. Securenlp: A system for multi-party privacy-preserving natural language processing[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3709-3721.
- [23] WAGH S, GUPTA D, CHANDRAN N. Securenn: 3-party secure computation for neural network training.[J]. Proc. Priv. Enhancing Technol., 2019, 2019(3): 26-49.
- [24] WANG S, ZHENG Y, JIA X. Secgnn: Privacy-preserving graph neural network training and inference as a cloud service[J]. IEEE Transactions on Services Computing, 2023.
- [25] KUMAR N, RATHEE M, CHANDRAN N, et al. Cryptflow: Secure tensorflow inference [C]//2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020: 336-353.
- [26] YUE X, DU M, WANG T, et al. Differential privacy for text analytics via natural text sanitization[C]//Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 2021: 3853-3866.
- [27] OPENAI. Gpt-4 technical report[A]. 2023. arXiv: 2303.08774.
- [28] MIRESHGHALLAH F, UNİYAL A, WANG T, et al. Memorization in nlp fine-tuning methods[A]. 2022.
- [29] MOHASSEL P, ZHANG Y. Secureml: A system for scalable privacy-preserving machine learning[C]//2017 IEEE symposium on security and privacy (SP). IEEE, 2017: 19-38.
- [30] CRAMER R, DAMGÅRD I, ESCUDERO D, et al. Spd: efficient mpc mod for dishonest majority[C]//Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology

- Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part II. Springer, 2018: 769-798.
- [31] WU S, LI G, CHEN F, et al. Training and inference with integers in deep neural networks [A]. 2018.
- [32] AGRAWAL N, SHAHIN SHAMSABADI A, KUSNER M J, et al. Quotient: two-party secure neural network training and prediction[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1231-1247.
- [33] 董业, 侯炜, 陈小军, 等. 基于秘密分享和梯度选择的高效安全联邦学习[J]. 计算机研究与发展, 2020, 57(10): 2241-2250.
- [34] PATRA A, SCHNEIDER T, SURESH A, et al. Aby2. 0: Improved mixed-protocol secure two-party computation.[C]//USENIX Security Symposium. 2021: 2165-2182.
- [35] MOHASSEL P, RINDAL P. Aby3: A mixed protocol framework for machine learning[C]//Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018: 35-52.
- [36] DEMMLER D, SCHNEIDER T, ZOHNER M. Aby-a framework for efficient mixed-protocol secure two-party computation.[C]//NDSS. 2015.
- [37] CHAUDHARI H, RACHURI R, SURESH A. Trident: Efficient 4pc framework for privacy preserving machine learning[A]. 2019.
- [38] HARDT M, PRICE E. The noisy power method: A meta algorithm with applications[J]. Advances in neural information processing systems, 2014, 27.
- [39] 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.
- [40] PAPERNOT N, ABADI M, ERLINGSSON U, et al. Semi-supervised knowledge transfer for deep learning from private training data[J]. stat, 2016, 1050: 7.
- [41] 史鼎元, 王晏晟, 郑鹏飞, 等. 面向企业数据孤岛的联邦排序学习[J]. 软件学报, 2021, 32(3): 669-688.
- [42] BOMBARI S, ACHILLE A, WANG Z, et al. Towards differential relational privacy and its use in question answering[A]. 2022.
- [43] ZHAO X, LI L, WANG Y X. Provably confidential language modelling[C]//Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022: 943-955.
- [44] WU X, LI F, KUMAR A, et al. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics[C/OL]//SIGMOD '17: Proceedings of the 2017 ACM International Conference on Management of Data. New York, NY, USA: Association for Computing Ma-

- chinery, 2017: 1307–1322. <https://doi.org/10.1145/3035918.3064047>.
- [45] SHI W, CUI A, LI E, et al. Selective differential privacy for language modeling[C]// Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022: 2848-2859.
 - [46] DWORK C, KENTHAPADI K, MCSHERRY F, et al. Our data, ourselves: Privacy via distributed noise generation[C]//Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25. Springer, 2006: 486-503.
 - [47] DWORK C, ROTH A, et al. The algorithmic foundations of differential privacy[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4): 211-407.
 - [48] 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.
 - [49] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C]//Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. Springer, 2006: 265-284.
 - [50] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C]//Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. Springer, 2006: 265-284.
 - [51] 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.
 - [52] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units[C]//54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), 2016: 1715-1725.
 - [53] DEUTSCH P, GAILLY J L. Zlib compressed data format specification version 3.3[R]. 1996.
 - [54] MOHASSEL P, RINDAL P. Aby3: A mixed protocol framework for machine learning[C]// Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018: 35-52.
 - [55] LEE T, LIN Z, PUSH P, et al. Occlumency: Privacy-preserving remote deep-learning inference using sgx[C]//The 25th Annual International Conference on Mobile Computing and Networking. 2019: 1-17.
 - [56] HUA W, UMAR M, ZHANG Z, et al. Guardnn: Secure dnn accelerator for privacy-preserving deep learning[A]. 2020.
 - [57] HASHEMI H, WANG Y, ANNAVARAM M. Darknight: A data privacy scheme for training and inference of deep neural networks[A]. 2020.
 - [58] BRASSER F, MÜLLER U, DMITRIENKO A, et al. Software grand exposure: Sgx cache

- attacks are practical.[C]//WOOT. 2017: 11-11.
- [59] HÄHNEL M, CUI W, PEINADO M. High-resolution side channels for untrusted operating systems.[C]//USENIX Annual Technical Conference: Vol. 17. 2017.
- [60] GÖTZFRIED J, ECKERT M, SCHINZEL S, et al. Cache attacks on intel sgx[C]//Proceedings of the 10th European Workshop on Systems Security. 2017: 1-6.
- [61] MOGHIMI A, IRAZOQUI G, EISENBARTH T. Cachezoom: How sgx amplifies the power of cache attacks[C]//Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Springer, 2017: 69-90.
- [62] SCHWARZ M, WEISER S, GRUSS D, et al. Malware guard extension: Using sgx to conceal cache attacks[C]//Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14. Springer, 2017: 3-24.
- [63] WANG W, CHEN G, PAN X, et al. Leaky cauldron on the dark land: Understanding memory side-channel hazards in sgx[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 2421-2434.
- [64] CHEN Z, OSWALD D F. Pmfault: Faulting and bricking server cpus through management interfaces: abs/2301.05538[A]. 2023.
- [65] QIU P, WANG D, LYU Y, et al. Voltjockey: A new dynamic voltage scaling-based fault injection attack on intel sgx[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, 40: 1130-1143.
- [66] ABADI M, AGARWAL A, BARHAM P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[Z]. 2015.
- [67] GAO T, FISCH A, CHEN D. Making pre-trained language models better few-shot learners [C]//Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP 2021. Association for Computational Linguistics (ACL), 2021: 3816-3830.
- [68] WEI J, TAY Y, BOMMASANI R, et al. Emergent abilities of large language models[A]. 2022.
- [69] XU F F, ALON U, NEUBIG G, et al. A systematic evaluation of large language models of code[C]//Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming. 2022: 1-10.
- [70] BEIMEL A. Secret-sharing schemes: A survey[C]//Coding and Cryptology: Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings 3. Springer, 2011: 11-46.

- [71] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.
- [72] COSTAN V, DEVADAS S. Intel sgx explained[J]. Cryptology ePrint Archive, 2016.
- [73] NILSSON A, BIDEH P N, BRORSSON J. A survey of published attacks on intel sgx[Z]. 2020.
- [74] RAMACHANDRAN P, AGARWAL S, MONDAL A, et al. S++: A fast and deployable secure-computation framework for privacy-preserving neural network training[Z]. 2021.
- [75] CATRINA O, SAXENA A. Secure computation with fixed-point numbers[C]//Financial Cryptography and Data Security: 14th International Conference, FC 2010, Tenerife, Canary Islands, January 25-28, 2010, Revised Selected Papers 14. Springer, 2010: 35-50.
- [76] MOTWANI R, RAGHAVAN P. Randomized algorithms[J]. ACM SIGACT News, 1995, 26(3): 48-50.
- [77] CANETTI R. Universally composable security: A new paradigm for cryptographic protocols [C]//Proceedings 42nd IEEE Symposium on Foundations of Computer Science. IEEE, 2001: 136-145.
- [78] CANETTI R, COHEN A, LINDELL Y. A simpler variant of universally composable security for standard multiparty computation[C]//Advances in Cryptology—CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II 35. Springer, 2015: 3-22.
- [79] LINDELL Y. How to simulate it—a tutorial on the simulation proof technique[J]. Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, 2017: 277-346.
- [80] BOGDANOV D, LAUR S, WILLEMSON J. Sharemind: A framework for fast privacy-preserving computations[C]//Computer Security-ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings 13. Springer, 2008: 192-206.
- [81] DOERNER J, KONDI Y, LEE E, et al. Threshold ecDSA from ecDSA assumptions: The multiparty case[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 1051-1066.
- [82] BEAVER D. Efficient multiparty protocols using circuit randomization[C]//Advances in Cryptology—CRYPTO' 91: Proceedings 11. Springer, 1992: 420-432.
- [83] CATRINA O, SAXENA A. Secure computation with fixed-point numbers[C]//Financial Cryptography and Data Security: 14th International Conference, FC 2010, Tenerife, Canary Islands, January 25-28, 2010, Revised Selected Papers 14. Springer, 2010: 35-50.
- [84] MISHRA P, PODDAR R, CHEN J, et al. Oblix: An efficient oblivious search index[C]//2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 279-296.
- [85] CHEN S, ZHANG X, REITER M K, et al. Detecting privileged side-channel attacks in shielded execution with déjà vu[C]//Proceedings of the 2017 ACM on Asia Conference on

- Computer and Communications Security. 2017: 7-18.
- [86] COSTAN V, LEBEDEV I A, DEVADAS S. Sanctum: Minimal hardware extensions for strong software isolation.[C]//USENIX Security Symposium. 2016: 857-874.
- [87] GRUSS D, LETTNER J, SCHUSTER F, et al. Strong and efficient cache side-channel protection using hardware transactional memory.[C]//USENIX Security Symposium. 2017: 217-233.
- [88] SHIH M W, LEE S, KIM T, et al. T-sgx: Eradicating controlled-channel attacks against enclave programs.[C]//NDSS. 2017.
- [89] KOGLER A, GRUSS D, SCHWARZ M. Minefield: A software-only protection for {SGX} enclaves against {DVFS} attacks[C]//31st USENIX Security Symposium (USENIX Security 22). 2022: 4147-4164.
- [90] LANG F, WANG W, MENG L, et al. Mole: Mitigation of side-channel attacks against sgx via dynamic data location escape[J]. Proceedings of the 38th Annual Computer Security Applications Conference, 2022.
- [91] GINER L, KOGLER A, CANELLA C, et al. Repurposing segmentation as a practical lvi-null mitigation in sgx[C]//USENIX Security Symposium. 2022.
- [92] LEE D, KOHLBRENNER D, SHINDE S, et al. Keystone: An open framework for architecting trusted execution environments[C]//Proceedings of the Fifteenth European Conference on Computer Systems. 2020: 1-16.
- [93] JIA Y, LIU S, WANG W, et al. {HyperEnclave}: An open and cross-platform trusted execution environment[C]//2022 USENIX Annual Technical Conference (USENIX ATC 22). 2022: 437-454.
- [94] TAN S, KNOTT B, TIAN Y, et al. Cryptgpu: Fast privacy-preserving machine learning on the gpu[C]//2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021: 1021-1038.
- [95] ZHANG X, LIU S, ZHANG R, et al. Fixed-point back-propagation training[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2330-2338.
- [96] FANG J, SHAFIEE A, ABDEL-AZIZ H, et al. Post-training piecewise linear quantization for deep neural networks[C]//Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer, 2020: 69-86.
- [97] GUPTA M, VARMA V, DAMANI S, et al. Compression of deep learning models for nlp [C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 3507-3508.
- [98] ZAFIR O, BOUDOUKH G, IZSAK P, et al. Q8bert: Quantized 8bit bert[C]//2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition

- (EMC2-NIPS). IEEE, 2019: 36-39.
- [99] EBADI H, SANDS D, SCHNEIDER G. Differential privacy: Now it's getting personal[J]. *Acm Sigplan Notices*, 2015, 50(1): 69-81.
- [100] KOTSOGIANNIS I, DOUDALIS S, HANEY S, et al. One-sided differential privacy[C]// 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 493-504.
- [101] CARLINI N, LIU C, ERLINGSSON Ú, et al. The secret sharer: Evaluating and testing unintended memorization in neural networks.[C]//USENIX Security Symposium: Vol. 267. 2019.
- [102] YOUSEFPOUR A, SHILOV I, SABLAYROLLES A, et al. Opacus: User-friendly differential privacy library in PyTorch[A]. 2021.

致 谢

在我完成本论文的过程中，得到了许多人的支持和帮助，我在此向他们表示由衷的感谢。

首先，我要感谢我的导师张驰老师，他一直以来给予我耐心、关心和支持，不仅在研究方向的选择上给予指导，还在学术论文的撰写上给予了宝贵的意见和建议。在整个研究过程中，他还给予了我充分的学术自由，让我能够自由探索和实验。同时，我也要感谢实验室的师兄师姐们，他们在我研究中提供了许多有益的建议和帮助。

其次，我还要感谢我的家人和朋友们，他们一直以来支持和鼓励我，让我有信心和勇气去攻克难关。

此外，我还要感谢学校的各位老师和工作人员，他们在我学习和生活中给予了充分的支持和关怀。

在读期间发表的学术论文与取得的研究成果

已发表论文

1. Y. Jie et al., "Multi-Party Secure Computation with Intel SGX for Graph Neural Networks," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 528-533, doi: 10.1109/ICC45855.2022.9839282.
2. Y. Ren, Y. Jie, Q. Wang, B. Zhang, C. Zhang and L. Wei, "A Hybrid Secure Computation Framework for Graph Neural Networks," 2021 18th International Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 2021, pp. 1-6, doi: 10.1109/PST52912.2021.9647843.
3. S. Ru, B. Zhang, Y. Jie, C. Zhang, L. Wei and C. Gu, "Graph Neural Networks for Privacy-Preserving Recommendation with Secure Hardware," 2021 International Conference on Networking and Network Applications (NaNA), Lijiang City, China, 2021, pp. 395-400, doi: 10.1109/NaNA53684.2021.00075.