北京郵電大學

# 本 科 毕 业 设 计（论文）


题目：一种基于 FATE 框架的联邦学习方法设计与实现


姓　　名　卢亭松

学　　院　网络空间安全学院

专　　业　网络空间安全

班　　级　2019211806

学　　号　2019212443

班内序号　11

指导教师　陆月明


2023 年 5 月

# 北 京 邮 电 大 学

## 本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《一种基于 FATE 框架的联邦学习方法设计与实现》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

## 关于论文使用授权的说明

本人完全了解并同意北京邮电大学有关保留、使用学位论文的规定，即：北京邮电大学拥有以下关于学位论文的无偿使用权，具体包括：学校有权保留并向国家有关部门或机构送交学位论文，有权允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，有权允许采用影印、缩印或其它复制手段保存。汇编学位论文，将学位论文的全部或部分内容编入有关数据库进行检索。（保密的学位论文在解密后遵守此规定）

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

# 北 京 邮 电 大 学

# 本科毕业设计（论文）任务书

| | | | | | |
|---|---|---|---|---|---|
| 学院 | 网络空间安全学院 | 专业 | 网络空间安全（实验班） | 班级 | 2019211806 |
| 学生姓名 | 卢亭松 | 学号 | 2019212443 | 班内序号 | 11 |
| 指导教师姓名 | 陆月明 | 所在单位 | 网络空间安全学院 | 职称 | 教授 |
| 设计(论文)题目 | （中文）一种基于 FATE 框架的联邦学习方法设计与实现 | | | | |
| | （英文）Design and Implementation of a Federated-Learning Method Based on FATE Framework | | | | |
| 题目分类 | 工程实践类☐　　研究设计类☑　　理论分析类☐ | | | | |
| 题目来源 | 题目是否来源于科研项目　　是☑　　否☐ | | | | |
| | 科研项目名称： | | | | |
| | 科研项目负责人： | | | | |

主要内容：

由于隐私保护要求，使得某些行业领域的数据共享面临困难，数据效用无法充分利用，形成了所谓的数据孤岛问题，联邦学习通过中央服务器在保护隐私的同时从本地数据中学习，为跨设备、跨孤岛机器学习问题提供了解决方案。毕业设计将基于开源项目 FATE (Federated Al Technology Enabler)，学习机器学习与隐私保护相关知识，搭建 FATE 联邦学习集群，对机器学习隐私保护，联邦学习进行调研，基于 FATE 框架设计实验，在联邦场景实现两种以上主流机器学习场景（计算机视觉、自然语言处理等）的样例算法，完成联邦学习机器学习算法的训练和测试，并与传统机器学习进行对比，从准确率、安全性等层面进行分析，完成毕业设计论文。

主要（技术）要求：

1. 学习机器学习与隐私保护相关知识，学习一种传统的机器学习框架（pytorch、tensorflow），学习并搭建 FATE 联邦学习集群。

2. 对机器学习隐私保护，联邦学习进行调研，分析当前机器学习隐私保护现状和联邦学习的优势，对联邦学习的场景，特点，实现原理进行调研，撰写文献综述。

3. 基于 FATE 框架设计实验，在联邦场景实现两种以上主流机器学习场景（计算机视觉、自然语言处理等）的样例算法。

4. 完成联邦学习机器学习算法的训练和测试，并与传统机器学习进行对比，从准确率、安全性等层面进行分析。

主要参考文献：

[1]谭作文, 张连福. 机器学习隐私保护研究综述. 软件学报, 2020, 31(7): 2127-2156.

[2] Abreha HG, Hayajneh M, Serhani MA. Federated Learning in Edge Computing: A Systematic Survey. Sensors. 2022; 22(2):450. https://doi.org/10.3390/s22020450

[3] K. M. Ahmed, A. Imteaj and M. H. Amini, “Federated Deep Learning for Heterogeneous Edge Computing,“ 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146-1152, doi: 10.1109/ICMLA52953.2021.00187.

[4] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato and S. Zhang, “Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach,“ in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7751-7763, Aug. 2020, doi: 10.1109/JIOT.2020.2991401.

[5] Liu JC, Goetz J, Sen S, Tewari A. Learning From Others Without Sacrificing Privacy: Simulation Comparing Centralized and Federated Machine Learning on Mobile Health Data. JMIR Mhealth Uhealth, 2021;9(3):e23728

[6] 胡健龙. 联邦学习在车联网数据共享与保护技术中的研究[D]. 电子科技大学,2022.DOI:10.27005/d.cnki.gdzku.2022.004716.

[7] M. Nasr, R. Shokri and A. Houmansadr, Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In Proceedings of 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 739-753, doi: 10.1109/SP.2019.00065.

[8] 王慧超. 机器学习中的数据隐私保护研究[D]. 中国科学技术大学,2021.DOI:10.27517/d.cnki.gzkju.2021.001722.

[9] Wang, X.; Wang, J.; Ma, X.; Wen, C. A Differential Privacy Strategy Based on Local Features of Non-Gaussian Noise in Federated Learning. Sensors 2022,22(2424). https://doi.org/10.3390/s22072424

[10] 师兆森. 联邦学习中的隐私保护技术研究. 电子科技大学,2022. DOI:10.27005/d.cnki.gdzku.2022.000987.

进度安排：

1、学习机器学习、隐私保护相关的知识，对研究背景进行调研。查找并阅读联邦学习、机器学习隐私保护的相关论文，为后续的实验以及论文撰写打下基础。秋季学期 17-18 周

2、开始 FATE 集群的搭建，完成开题报告的撰写。春季学期 1-2 周

3、构建论文框架，完成 FATE 集群搭建。春季学期 3-4 周

4、完成论文前两章的撰写，确定两种以上的机器学习场景及代表性问题，设计相应的联邦机器学习算法。春季学期 5-6 周

5、总结上一阶段的工作，完成中期检查。春季学期 7 周

6、完成两种以上的联邦机器学习算法的传统实现与联邦环境实现，能够在 FATE 集群上进行训练。春季学期 8-9 周

7、在 FATE 平台上完成预测与评估，对比分析算法在集中式环境与联邦环境的差异，从准确率，安全性，通信效率等进行分析。春季学期 10-11 周

8、根据上述的实验结果，完成论文的撰写。春季学期 12-13 周

9、完成论文，整理本毕设课题的全部成果。春季学期 14 周

| 指导教师签字 | | 日期 | 年　月　日 |
|---|---|---|---|

编号：_____

# 北京邮电大学本科生毕业设计（论文）成绩评定表

<table>
<tr><td>学生姓名</td><td colspan="2">卢亭松</td><td>所在学院</td><td colspan="3">网络空间安全学院</td></tr>
<tr><td>学号</td><td>2019212443</td><td>专业</td><td>网络空间安全专业</td><td>班级</td><td colspan="2">2019211806</td></tr>
<tr><td rowspan="2">论文题目</td><td colspan="6">（中文）一种基于 FATE 框架的联邦学习方法设计与实现</td></tr>
<tr><td colspan="6">（英文）Design and Implementation of a Federated-Learning Method Based on FATE Framework</td></tr>
<tr><td>指导教师姓名</td><td>陆月明</td><td>指导教师职称</td><td>教授</td><td>指导教师单位</td><td colspan="2">网络空间安全学院</td></tr>
<tr><td>中期检查小组评分</td><td colspan="6">（满分 10 分）：　　中期检查小组组长签字：　　检查日期：</td></tr>
<tr><td rowspan="9">指导教师评分</td><td>评价内容</td><td colspan="2">具体要求</td><td colspan="3">分值</td><td>评分<br>指导教师</td><td>评分<br>复议</td></tr>
<tr><td>调研论证</td><td colspan="2">能独立查阅文献和从事相关调研；能正确翻译外文资料；有收集、加工各种信息及获取新知识的能力和自学能力。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td>方案设计</td><td colspan="2">能独立提出符合选题的可行性研究方案、实验方案、设计方案，独立进行实验（如安装、调试、操作）和研究方案论证。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td>能力水平</td><td colspan="2">能综合运用所学知识和技能去分析与解决毕业设计（论文）过程中遇到的实际问题；能正确处理实验数据；能对课题进行理论分析，得出有价值的结论。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td>学习态度</td><td colspan="2">认真、勤奋、努力、诚实、严格遵守纪律，按期饱满完成规定的任务。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td>设计（论文）水平</td><td colspan="2">文题相符、综述简练完整，有见解；立论正确，论述充分，结论严谨合理；实验正确，分析处理科学；文字通顺，技术用语准确，设计（论文）有理论价值和应用价值。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td>文本规范</td><td colspan="2">装订顺序正确，字体字号等与基本规范相符，符号统一，编号齐全，图表完备、整洁、正确。</td><td>5 4 3.5 3 2</td><td></td><td></td></tr>
<tr><td colspan="7">指导教师评分合计（满分 30 分）：<br>评语：<br><br>指导教师签字：　　日期：　　年　　月　　日</td></tr>
</table>

<table>
<tr><td>复 议</td><td colspan="7">□是　□否　复议评分合计：　　复议人签字：　　复议日期：<br>复议有权限修改指导教师评分，选择复议后指导教师评分将由复议评分替换</td></tr>
<tr><td colspan="8">本科生毕业设计（论文）答辩成绩评定标准</td></tr>
<tr><td rowspan="8">答辩小组成绩评定</td><td>评价内容</td><td>具体要求</td><td colspan="5">分值</td><td>评分</td></tr>
<tr><td>选题</td><td>符合专业培养目标，符合社会实际、结合工程实际，难易适度，体现新颖性、综合性。</td><td>5</td><td>4</td><td>3.5</td><td>3</td><td>2</td><td></td></tr>
<tr><td>设计（论文）质量水平</td><td>全面完成任务书中规定的各项要求，文题相符，工作量饱满，写作规范，达到综合训练的要求，有理论成果和应用价值。</td><td>20</td><td>16</td><td>14</td><td>12</td><td>8</td><td></td></tr>
<tr><td>答辩准备</td><td>准备充分；有简洁、清晰、美观的演示文稿；准时到场。</td><td>5</td><td>4</td><td>3.5</td><td>3</td><td>2</td><td></td></tr>
<tr><td>内容陈述</td><td>语言表达简洁、流利、清楚、准确，思路清晰，重点突出，逻辑性强，概念清楚，论点正确；实验方法科学，分析归纳合理；结论严谨；表现出对毕业设计（论文）内容掌握透彻。</td><td>20</td><td>18</td><td>14</td><td>12</td><td>8</td><td></td></tr>
<tr><td>回答问题</td><td>回答问题准确、有深度、有理论根据、基本概念清晰。</td><td>10</td><td>8</td><td>7</td><td>6</td><td>4</td><td></td></tr>
<tr><td colspan="2">答辩小组评分合计（满分 60 分）</td><td colspan="6"></td></tr>
<tr><td colspan="8">意见：<br><br>答辩小组组长签字：_____　　年　月　日</td></tr>
<tr><td colspan="8">答辩小组成员：</td></tr>
<tr><td>学院意见</td><td colspan="7">最终成绩：百分制_____；　五分制_____<br><br>院长签章：　　学院盖章：　　年　月　日</td></tr>
<tr><td>备注</td><td colspan="7"></td></tr>
</table>

注：1．毕业设计（论文）成绩由中期检查评分（满分 10 分）、指导教师评分/复议评分（满分 30 分）和答辩小组评分（满分 60 分）相加，得出百分制成绩，再按 100-90 分为“优”、89-80 分为“良”、79-70 分为

“中”、69-60 分为“及格”、60 分以下为“不及格”的标准折合成五级分制成绩；

2. 此表原件一式三份，一份存入学生档案，一份装订到毕业论文中，一份交教务处存入档案馆。

# 一种基于 FATE 框架的联邦学习方法设计与实现

## 摘　要

从 1955 年达特茅斯会议开始，人工智能经过两起两落的发展，迎来了第三个高峰期。但是，在大多数行业中，数据是以孤岛的形式存在的，由于行业竞争、隐私安全、行政手续复杂等问题，即使是在同一个公司的不同部门之间实现数据整合也面临着重重阻力；另一方面，随着大数据的进一步发展，重视数据隐私和安全已经成为了世界性的趋势。

联邦学习作为一种切实可行的隐私保护手段，为多方机器学习建模提供了有效解决方案。这一隐私机器学习系统允许各方在确保本地数据隐私安全且符合法律法规的前提下，进行数据处理和模型构建。本文通过实验证明了联邦学习在实际应用中的有效性和优越性。联邦学习不仅在传统机器学习技术的基础上提供了隐私计算，为用户带来强大的隐私保护机制，而且在保证模型精度的同时实现了更优的性能。

本研究成功地运用金融信贷场景的数据，完成了基于 FATE 框架的联邦学习风险预测模型的设计、训练与实现，并将其与单机构数据训练的模型进行了对比。分析结果表明，借助联邦学习，金融机构之间的合作将能得到极大的推动力。

同时，本文通过对不同数据分布下联邦学习训练结果的对比分析，成功测试了数据分布对联邦训练效果的影响。研究结论显示，在总数据量不变的情况下，数据分布越均匀，联邦学习模型训练效果越佳。在现实商业活动中，各合作方应当充分参考各自数据集的分布特点，以便更深刻地评估联邦学习背后的商业价值。

在大数据安全的背景下，联邦学习为解决数据安全、数据泄露、用户隐私等问题提供了一种高效的解决方案。但在某些方面仍存在提升空间，如计算成本较高等。因此，在未来的研究中，需要进一步探索更加安全且高效的隐私计算技术，将其融入联邦学习框架中，为解决现实应用中的问题提供更为强大的支持。

**关键词**　联邦学习　FATE　金融信贷　风险预测　机器学习

# Design and Implementation of a Federated-Learning Method Based on FATE Framework

## ABSTRACT

Since the 1955 Dartmouth Conference, artificial intelligence has gone through two ups and downs and ushered in its third peak. However, in most industries, data exists in the form of isolated islands. Due to industry competition, privacy security, complex administrative procedures, and other issues, it faces significant challenges to integrate data even within different departments of the same company. On the other hand, with the further development of big data, the focus on data privacy and security has become a global trend.

Federated learning as a practical privacy protection measure provides an effective solution for multi-party machine learning modeling. This privacy-preserving machine learning system allows multiple parties to process data and develop models while ensuring local data privacy and security in compliance with laws and regulations. This paper demonstrates the effectiveness and superiority of federated learning in practical applications through experiments. Federated learning not only provides privacy-preserving computation based on traditional machine learning technology but also has better performance while ensuring model accuracy.

This study successfully applied data from financial scenarios to the FATE framework for the design, training, and implementation of federated learning risk prediction models and compared them with models trained on single-entity data. The analysis results show that with the help of federated learning, collaboration between financial institutions can yield a significant boost.

At the same time, this paper analyzes the impact of data distribution on federated training results through a comparison of different data distribution settings, successfully testing the effects of federated learning under various data distributions. Research conclusions show that with a constant total data volume, the more uniform the data distribution, the better the training results of federated learning models. In real-world business activities, all parties should fully consider the distribution characteristics of their datasets to more profoundly assess the commercial value behind federated learning.

In the context of big data security, federated learning provides an efficient solution for issues such as data security, data leakage, and user privacy. However, there is still room for improvement in some aspects, such as high computational costs. Therefore, future research needs to further explore more secure and efficient privacy computing technologies, integrate them into the federated learning framework, and provide more robust support for solving practical application problems.

**KEY WORDS** Federated Learning FATE Financial Credit Risk Prediction Machine Learning

# 目　录

# 第一章　引言

## 1.1　联邦学习背景和重要性

从 1955 年达特茅斯会议开始，人工智能经过两起两落的发展，迎来了第三个高峰期。越来越多的工程与科研实践展示了人工智能迸发出的巨大潜力，也更加憧憬人工智能技术可以在自动驾驶、医疗、金融等更多、更复杂、更前沿的领域施展拳脚。但是，真实的情况却让人失望：除了有限的几个行业，更多领域存在着数据有限且质量较差的问题，不足以支撑人工智能技术的实现；并且，在某些领域，即使动用很多人力来进行数据标注，数据量也依然不够，例如商家拥有商品的质量数据、客户购买行为产生的销售数据，但是缺少客户经济水平和支付方式的数据。再例如，一家医院想要建立 AI 系统帮助医生诊断，但自己的数据远远不够。这是我们面临的现实。

与此同时，数据源之间存在着难以打破的壁垒，在大多数行业中，数据是以孤岛的形式存在的，由于行业竞争、隐私安全、行政手续复杂等问题，即使是在同一个公司的不同部门之间实现数据整合也面临着重重阻力；另一方面，随着大数据的进一步发展，重视数据隐私和安全已经成为了世界性的趋势，Facebook 的泄密事件引发了大面积抵制和思考[1]；国内包括手机号码、身份证信息等个人隐私泄露情况十分严重；欧盟于 2018 年推行《通用数据保护条例》（GDPR），严格限制隐私数据的交易；2017 年 6 月 1 日我国跟进制定的《中华人民共和国网络安全法》规定，不得使用、篡改个人隐私数据。我国 2020 年 12 月 1 日制定的《常见类型移动互联网应用程序（APP）必要个人信息范围》中，对数据安全的重视程度达到空前的水平。新的隐私保护法规的建立在不同程度上对人工智能传统的数据处理模式提出了新的挑战，将会进一步加剧数据孤岛的问题。因此，当前想要将分散在各地、各个机构的数据进行整合是十分困难且成本巨大的[2]。

联邦学习作为一种切实可行的隐私保护手段，为多方机器学习建模提供了有效解决方案。这一隐私机器学习系统允许各方在确保本地数据隐私安全且符合法律法规的前提下，进行数据处理和模型构建。联邦学习可以看作是一种特别针对数据孤岛问题的机器学习方法，实现了在数据不流通的基础上进行协同建模和改善数据孤岛现象。这一技术具有五大关键特性：

1. 数据保留在本地，不参与流通，降低隐私泄露和违规风险
2. 各方依据投入数据，开展合作建模，实现资源共享和共赢，投入越多收益越大
3. 各方地位平等，不存在数据优势问题
4. 联邦学习训练结果的损失较小，与传统集中式机器学习相差无几
5. 通过迁移学习，在传统无法实现联邦的应用场景中，使用加密参数互换来实现知识跨领域迁移

本项目来源于科研项目“基于 LaTeX 的本科毕业设计”，项目编号 1124

联邦学习相比于单方传统机器学习的优势是：大大扩展了数据来源，增强模型准确率; 各方可以享受到其他数据方对模型精度提升的同时，避免数据交易带来的纠纷[3]。联邦学习相比于分布式机器学习的优势有：参与节点是协作关系而非从属关系; 数据不出本地，满足安全合规; 模型准确率与聚合训练相当。联邦学习为多方安全协同建模提供了理论和实践基础，打破了数据孤岛和用户隐私的两难困境，把散落在各领域、各机构的小数据合并成大数据，在数据之间建立广泛的安全连接，满足隐私保护下的数据协作要求，是实现跨个体间协同合作的有效方式，能够帮助人们掘出数据中“1+1>2”的潜在价值。

## 1.2 国内外研究现状

联邦学习作为隐私保护特性的大数据分布式解决方案由 Google 在 2016 年首先投入使用[4]，目的是解决安卓设备的分布式建模问题，主要是针对输入法自动联想，提升用户体验的建模。自动联想需要大量的用户数据学习，但获取这些用户数据存在门槛，主要是隐私方面的门槛。比较直接的做法是将用户的输入习惯全部上传至云端，在云端进行统一计算，这种做法无疑是对用户隐私的破坏。对此，谷歌提出横向的联邦学习，用户在本地完成一部分训练，然后把训练的中间梯度上传到云端。这样谷歌没有获取到用户的聊天内容。例如云服务器 Parameter Server 初始化一个全局模型，把模型推送到个人设备上，然后各个设备基于本地的数据来优化模型得到更新梯度传回服务器，接着服务器利用接收到的梯度更新全局模型发回到设备上完成迭代，直至模型在某种标准上收敛。这是联邦学习工业应用最开始的雏形[5]。

国内于 2018 年左右经香港科技大学杨强教授引入联邦学习，作为具有经济和学习潜力的新兴技术，国内外众多高校和公司都参与了研究开发。包括为微众银行的 FATE，腾讯云的 Angel PowerFL，蚂蚁金服的 Morse，阿里云的 DataTrust 等等。其中开源项目有 FATE，谷歌的 TensorFlow 等。微众银行技术团队将联邦学习技术在银行金融行业实验性使用，实现不同组织机构、金融实体间的隐私合作。香港大学的杨强教授于 2019 年率先研究出迁移学习的解决方案，能够解决联邦学习必须在样本或特征重合度高才能适用的限制，使其更加通用化，大大增加联邦学习的泛用性。

包括 FATE 在内的大部分联邦学习平台使用参数服务器来提供聚合梯度、统筹节点工作流的功能。中心化的参数服务器在联邦学习中承担重要功能。但实际应用中，中央服务器和节点存在大量的通信，服务器被攻击可能导致参数的泄露，甚至被篡改。为了解决这样的问题，有学者和机构尝试使用去中心化的联邦学习。如腾讯自研的 PowerFL，通过秘密共享（SS）机制实现节点之间的全局模型统一，一定程度上避免中心化联邦学习的隐私风险。然而，现阶段的解决方案仍然存在以下显著问题[6]：

1. 联邦学习从结构上避免了原始数据的聚合，但仍存在中间参数泄露风险，尤其是梯度信息。现有的机器学习攻击中，模型攻击、数据攻击、推理攻击、后门攻击、投毒攻击等方式都可以通过中间参数对原始数据进行一定的推断

2. 以 FATE 为例，主要采用多方安全计算（MPC）和可信执行环境（TEE）保证节点通信的安全。但在大数据情况下，这种方式的运算成本可能会很高

3. 节点互信问题。在节点间传输的梯度不加干扰的情况下，节点可能出于自利等因素窃取、篡改梯度信息

为了解决这样的问题，部分研究从密码学的角度入手，改进联邦学习通信协议、求交，算法，评估时的安全计算协作方式等。这样做的好处在于保证了模型的无损，对训练精度友好，但通信成本非常大。在实际应用场景硬件需求水平很高。目前如何平衡通信、计算的负担与模型安全之间的关系仍是研究的重点与挑战。

目前，联邦学习在诸多热点领域已得到广泛使用

### 1.2.1 联邦学习在移动设备领域的应用

联邦学习最初由谷歌提出并主要用于训练其输入法。由此延伸，学术界和业界对联邦学习应用于键盘输入预测、表情预测[7][8][9][4] 以及利用智能装备数据预测人类轨迹和人类行为习惯等[10][11] 展开研究。此外，由于联邦学习无需直接收集设备相关数据，能够有效保护用户的数据隐私不被侵犯，还可以利用智能家居等物联网（IoT）设备，通过构建安全的联合模型进行用户的行为模式学习和喜好分析，等等[12][13]。

### 1.2.2 联邦学习在医药健康领域的应用

随着医院的信息化发展，病历资料、生化检测、基因检测、计算机断层扫描、磁共振成像等相关健康数据均已实现电子化。尽管部分医院拥有充足的数据对自有模型进行训练，但联邦学习在疾病预测、生物医药等方面可以打破医院间由于隐私保护政策所导致的壁垒。目前，联邦学习已被应用于生物医学的图像分析中，例如，磁共振成像特征提取和脑电图分类，等等[14][15]。许多学者对联邦学习在电子健康档案方面的应用进行研究[16][17]，例如，预测心脏病患者的死亡率、评估心脏病患者是否需入院治疗、基于药物用量预测患者死亡率和同类型患者分类，等等[18][19][20]。研究证明联邦学习可以通过自然语言处理分析电子健康档案中的有效信息，并利用其对各种疾病进行学习[15]。

### 1.2.3 联邦学习在金融领域的应用

近年来，国内外部分金融机构已开展联邦学习应用，但多数仍为研究试点。在我国，主要应用于银行业与保险业，如百度金融安全计算平台中的车险和健康交叉险业务、腾讯安全的保险广告投放 RTA 、微众银行的联邦信贷风控等，但在银行业中开展的工作更多[21]，主要针对风控、营销、反洗钱等方面。联邦学习目前在我国银行业中的应用仍处于初步发展阶段，在计算成本、技术的成熟性、相关法律法规的监管等多个方面还存在一定缺陷，联邦学习在金融行业中的应用还需不断进行探索。

在这方面，本文围绕金融风险预测展开，探讨如何利用联邦学习技术克服跨机构间的数据共享障碍，同时提高风险控制效果。具体而言，本文基于 FATE（Federated AI Technology Enabler）这一开源联邦学习框架，设计并实现了一个金融风险预测模型。本文希望验证联邦学习在金融风险预测场景下的性能和适用性，并为后续在更广泛的场景下引入联邦学习技术提供实践经验。在模型设计和实现的基础上，本文对其性能进行了测试，包括模型准确性、数据安全性等方面的评估。这一实证研究期望揭示联邦学习技术在金融风险预测场景中的价值和潜力，从而促进金融行业对联邦学习的更广泛应用。

## 1.3 本文结构

本文第二章主要讲述联邦学习及其相关技术原理与 FATE 框架技术架构，分节对联邦学习的概念与理论、算法原理、分类与当前使用的隐私保护技术原理、FATE 架构进行了介绍。

本文第三章主要讲述了针对金融风险预测模型的联邦学习系统设计原理与流程，并介绍了实验的主要步骤、实验的环境等。

本文第四章主要讲述了实验过程，包括数据预处理、SMOTE 算法过采样、分别在集中式与联邦环境下训练模型等。

本文第五章主要展示了实验结果，并基于实验结果对比了联邦模型与各单机模型的效果、联邦模型在不同数据分布下的模型效果，并对联邦学习过程的安全性进行了分析，最终总结实验结论，证明了借助联邦学习，金融机构之间的合作将能得到极大的推动力。

# 第二章　联邦学习架构与隐私保护技术介绍

## 2.1　联邦学习概念与理论

### 2.1.1　联邦学习概念

传统的机器学习算法需要用户将源数据上传到高算力的云服务器上集中训练，这种方式导致了数据流向的不可控和敏感数据泄露问题。联邦学习技术允许用户在机器学习过程中既可以保护用户隐私，又能够无须源数据聚合形成训练数据共享。

联邦学习本质上是一种分布式的机器学习技术，其流程如图 2-1 所示。

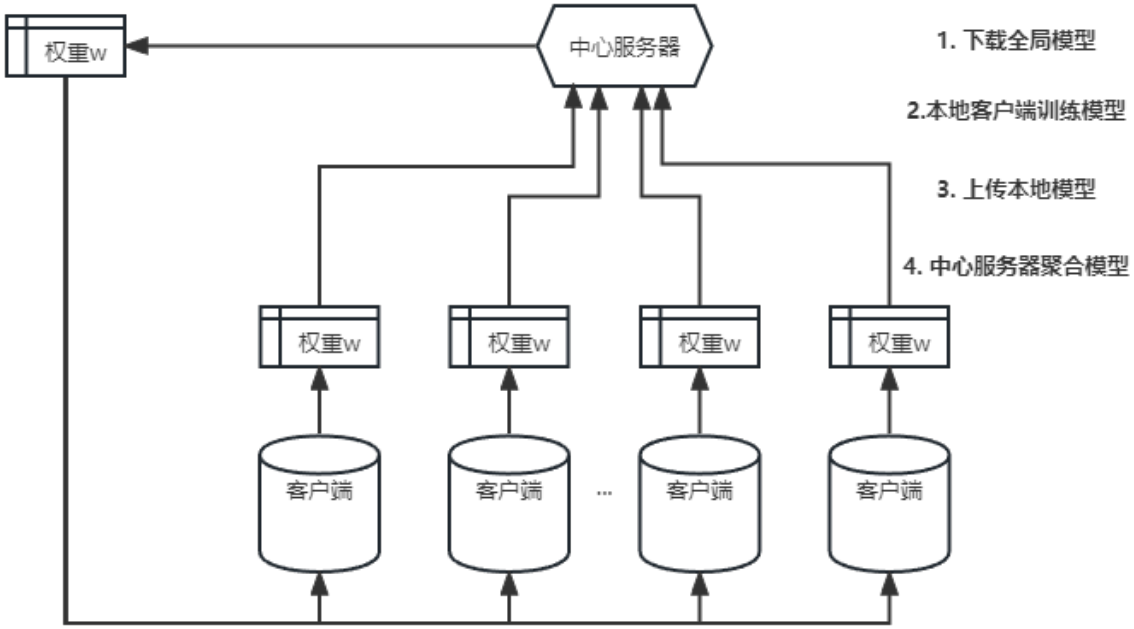



图 2-1　联邦学习流程

客户端（如平板电脑、手机、物联网设备）在中心服务器（如服务提供商）的协调下共同训练模型，其中客户端负责训练本地数据得到本地模型。中心服务器负责加权聚合本地模型，得到全局模型，经过多轮迭代后最终得到一个趋近于集中式机器学习结果的模型 w，有效地降低了传统机器学习源数据聚合带来的许多隐私风险。

联邦学习的一次迭代过程如下：

1. 客户端从服务器下载全局模型 $w_{t-1}$
2. 客户端 k 训练本地数据得到本地模型 $w_{t□k}$，（第 k 个客户端第 t 轮通信的本地模型更新）
3. 各方客户端上传本地模型更新到中心服务器

4. 中心服务器接收各方数据后进行加权聚合操作，得到全局模型 $w_t$（第 t 轮通信的全局模型更新）

综上，联邦学习技术具有以下几个特点：

1. 参与联邦学习的原始数据都保留在本地客户端，与中心服务器交互的只是模型更新信息
2. 联邦学习的参与方联合训练出的模型 w 将被各方共享
3. 联邦学习最终的模型精度与集中式机器学习相似
4. 联邦学习参与方的训练数据质量越高，全局模型精度越高

### 2.1.2 联邦学习算法原理

典型的联邦学习场景是在本地客户端设备负责存储和处理数据的约束下，只上传模型更新的梯度信息，在数千万到数百万个客户端设备上训练单个全局模型 w。中心服务器的目标函数 F(w) 通常表现为

$$\min_{w} F(w), F(w) = \sum_{k=1}^{m} \frac{n_k}{n} F_k(w) \qquad \text{式（2-1）}$$

其中，m 是参与训练的客户端设备总数，n 是所有客户端数据量总和，$n_k$ 是第 k 个客户端的数据量，$F_k(w)$ 是第 k 个设备的本地目标函数。

$$F_k(w) = \frac{1}{n_k} \sum_{i \in d_k} f_i(w) \qquad \text{式（2-2）}$$

其中，$d_k$ 是第 $k$ 个客户端的本地数据集，$f_i(w) = \alpha\left(x_i, y_i, w\right)$ 是具有参数 $w$ 的模型对数据集 $d_k$ 中的实例 $(x_i, y_i)$ 产生的损失函数。$d_k$ 中所有实例产生的损失函数之和除以客户端 $k$ 的总数据量就是本地客户端的平均损失函数，损失函数与模型精度成反比，因此，机器学习的目标函数优化通常是让损失函数达到最小值[22]。

联邦学习的目标函数优化算法中，通常采用大批量随机梯度下降 (SGD) 算法，即通过本地客户端模型训练的损失函数，乘以固定的学习率 $\eta$，计算出新一轮的权重更新。因此，本地客户端的模型权重更新如下:

$$w_{t,k} = w_{t-1,k} - \eta \nabla F_k(w) \qquad \text{式（2-3）}$$

第 $t$ 轮通信中心服务器的模型聚合更新如下:

$$w_t = \sum_{k=1}^{K} \frac{n_k}{n} w_{t,k} \qquad \text{式（2-4）}$$

### 2.1.3 联邦学习分类

表 2-1 横向联邦学习

| 数据集 | user | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 数据集 A | $user_1$ | | | | | | | | | | |
| | $user_2$ | | | | | | | | | | |
| | $user_3$ | | | | | | | | | | |
| 数据集 B | $user_4$ | | | | | | | | | | |
| | $user_5$ | | | | | | | | | | |
| | $user_6$ | | | | | | | | | | |

表 2-2 纵向联邦学习

| user | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $user_1$ | | | | | | | | | | | |
| $user_2$ | | | | | | | | | | | |
| $user_3$ | | | | | | | | | | | |
| $user_4$ | | | | | | | | | | | |
| $user_5$ | | | | | | | | | | | |
| $user_6$ | | | | | | | | | | | |
| 数据集 A | | | | | | 数据集 B | | | | | |

表 2-3 迁移联邦学习

| user | $x_1$ | $x_2$ | $x_3$ | $x_4$ | ... |
|---|---|---|---|---|---|
| $user_1$ | | | | | |
| $user_2$ | | | | | |
| $user_3$ | | | | | |

数据集 A

数据集 B

| user | $y_1$ | $y_2$ | $y_3$ | $y_4$ | ... |
|---|---|---|---|---|---|
| $user_4$ | | | | | |
| $user_5$ | | | | | |
| $user_6$ | | | | | |

在联邦学习系统中，各参与方的数据又具有不同的分布特征. 若根据参与方之间数据重叠程度的不同，联邦学习可以分为横向联邦学习[23]、纵向联邦学习[24] 以及迁移联邦学习[25]。如表 2-1 所示，横向联邦学习也称特征对齐的联邦学习，适用于各参与方

之间数据特征空间重叠较多而用户空间重叠较少的情况。目前横向联邦学习经典框架是 FedAvg[26]，唤醒单词识别和输入法下一词预测是横向联邦的典型应用。纵向联邦学习（如表 2-2 所示），即样本对齐的联邦学习，适用于各参与方之间用户空间重叠较多，而特征空间重叠较少或没有重叠的场景。目前支持纵向联邦学习的经典框架包括 FATE，PaddleFL，FedML。联邦迁移学习（如表 2-3 所示）是对横向联邦学习和纵向联邦学习的补充。它用于克服数据或标签不足的情况，以解决单边数据规模小和标签样本少的问题，适用于各参与方用户空间和特征空间都重叠较少的场景。目前支持联邦迁移学习的框架主要为 FATE。

## 2.2 隐私保护技术原理

现有的方案主要通过结合典型隐私保护技术来提供进一步的隐私增强, 如差分隐私、安全多方计算、同态加密等技术，这些技术在之前的研究中已经被广泛应用于传统机器学习的隐私保护[27]。

### 2.2.1 差分隐私

设随机化算法 A，对于两个至多有一条数据不同的数据集 $D$ 和 $D'$ 以及任意可能的输出 $S$，若算法 A 满足

$$\mathrm{pr}[\mathrm{A}(D) \in S] \leqslant e^{\varepsilon}\,\mathrm{pr}\left[\mathrm{A}\left(D'\right) \in S\right] + \delta \qquad \text{式（2-5）}$$

则称随机化算法 A 满足 $(\varepsilon, \delta)$ 差分隐私保护。其中，$\varepsilon$ 代表隐私保护预算，$\delta$ 是算法允许的误差，通常为较小的常数。

Dwork 等[28] 于 2006 年提出差分隐私概念，并使用严格的数学推导给出了安全性证明。通常差分隐私算法的噪声机制分为指数噪声、Laplace 噪声和高斯噪声，其中，指数噪声主要用于处理离散数据集，Laplace 噪声和高斯噪声主要用于处理连续数据集。

### 2.2.2 安全多方计算

安全多方计算[29]。假设有 $n$ 个参与方 $P_1, P_2, \cdots, P_n$ 分别拥有自己的敏感数据 $m_1$,$m_2$,$\cdots$,$m_n$，这 $n$ 个参与者在不泄露各自输入数据的前提下共同执行一个协议函数 f($m_1$,$m_2$,$\cdots$,$m_n$)。

安全多方计算的研究焦点是在没有可信第三方的条件下, 参与训练各方安全计算的一个共同的约束函数。姚期智[30] 于 1983 年提出安全多方计算的概念, 通过混淆电路、不经意传输、秘密分享等技术实现多方共同运算，并确保各方数据的安全性。

### 2.2.3 同态加密

设有明文数据 $d_1, d_2, \cdots, d_n$，这 $n$ 个数据对应的加密数据为 $m_1, m_2, \cdots, m_n$，若加密算法满足

$$\begin{aligned} &\text{Enc}\left(f\left(m_1, m_2, \cdots, m_n\right)\right) = \\ &f\left(\text{Enc}\left(m_1\right), \text{Enc}\left(m_2\right), \cdots, \text{Enc}\left(m_n\right)\right) \end{aligned} \quad \text{式（2-6）}$$

则称该加密算法满足同态加密。同态加密能够直接对密文数据进行密码学运算，最终运算结果经解密后与在明文上直接运算结果一致。Rivest 等[7] 于 1978 年提出同态加密概念。同态加密分为全同态加密和部分同态加密，其中部分同态加密分为乘法同态和加法同态，若一个算法既满足乘法同态又满足加法同态，则称为全同态加密算法。

## 2.3 FATE 技术框架

联邦学习框架 FATE（Federated AI Technology Enabler）是一个开源的联邦学习框架，面向产业界提供了一个可扩展的联邦学习框架，支持横向联邦和纵向联邦等多种联邦学习场景。FATE 的核心设计理念是提供一种可靠的安全计算框架，使得数据拥有方可以在不共享数据的情况下完成模型训练。FATE 的安全计算框架基于联邦学习算法，通过安全多方计算（Secure Multi-Party Computation，SMPC）协议，保证了数据在联合建模过程中的安全性。

FATE 由多个互相解耦的组件组成，FATE 框架架构图如图 2-2 所示。

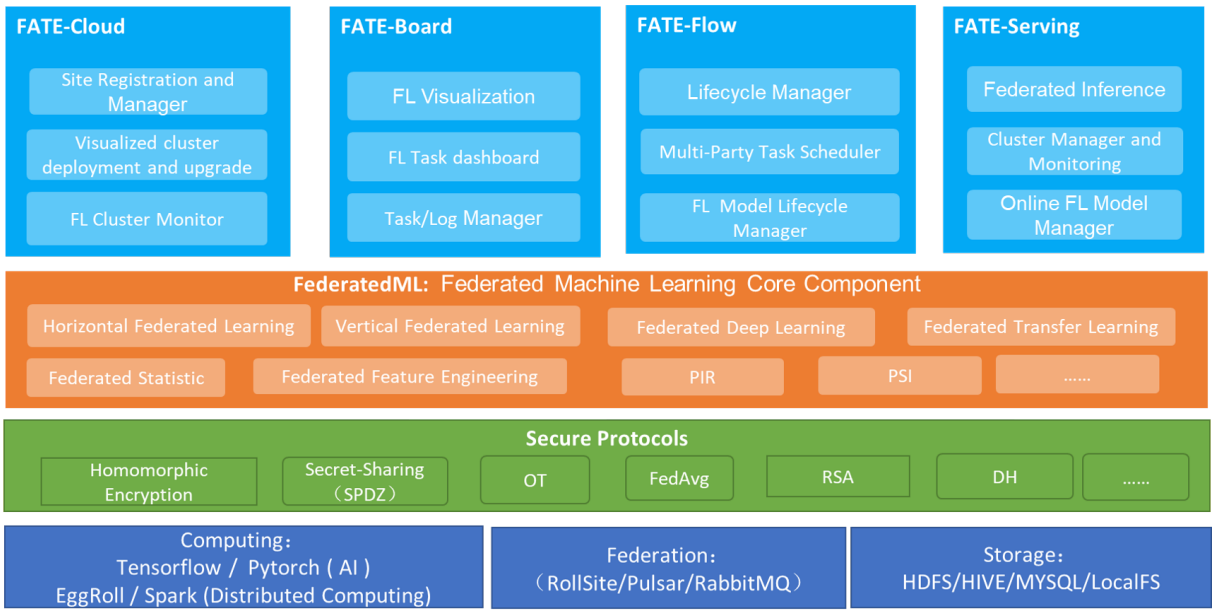



图 2-2 FATE 架构图[31]

### 2.3.1 FATE 架构设计

#### 2.3.1.1 FederateML

FederateML 是 FATE 的算法功能组件，其实现了大部分常见的机器学习算法的联邦化学习。为了增强可扩展性，所有模块均采用解耦模块化方法开发，例如：联邦统计、联邦信息检索、联邦特征工程、联邦机器学习算法、模型评估等。以下是 FederateML 中一些常用的算法组件。

- **DataIO**：DataIO 是 FATE 算法模块中最基本的组成部分，其作用是转换数据类型，用户所上传的数据值转换为 FATE 定义的 Instance 对象的 Table，而且转换后的 Table 提供的数据类型是所有其他算法模块需要的输入数据格式，例如：intersect、homo LR、SecureBoost 等。另外，DataIO 还具备估算缺失值和离群值替换的特性，提供缺失值插补的方法：“mean”、“designated”、“min”、“max”

- **Federated Logistic Regression**：逻辑回归（LR）是解决分类问题中广泛使用的统计模型。FATE 为联邦逻辑回归提供了两种模式：横向联邦逻辑回归（HomoLR）和纵向联邦逻辑回归（HeteroLR）

- **Federated Neural Networks**：神经网络是一种模拟人脑神经元网络的机器学习算法，具有强大的表达能力。FATE 为联邦神经网络提供了实现：横向联邦神经网络（Homogeneous Neural Networks）和纵向联邦神经网络（Heterogeneous Neural Networks）。将联邦过程简化为三个参与方。A 方代表 Guest，作为任务触发器。B 方代表 Host，与 Guest 几乎相同，但 Host 不启动任务。C 方充当协调员，汇总来自 guest/hosts 的模型，并广播汇总后的模型

- **Evaluation**：Evaluation 模块提供了回归、聚类和分类等一些机器学习中的评估方法，常用的指标包括计算机二进制的 AUC、用于评估模型风险区分能力的 KS（Kolmogorov-Smirnov）、计算二分类和多分类的准确率的 ACCURACY 等

#### 2.3.1.2 FATE-Flow

FATE-Flow 是 FATE 框架的作业调度系统，在联邦学习的过程中，实现作业生命周期的完整管理，其中包括输入数据、训练模型、追踪指标、模型中心化等功能。在 FATE 的任务中，需要提交 dsl 和 conf 两个配置文件到 FATE-Flow 来执行，主要是将各类服务启动并在后台运行，另外，FATE 使用 flask 框架作为 HTTP 服务框架提供 Web 服务。

#### 2.3.1.3 FATE-Board

FATE-Board 是联邦学习建模的可视化工具，为终端用户可视化和度量模型训练的全过程。其主要的功能是将 FATE 的 Evaluation 模块对数据进行模型评估后，各个评估指标以评估曲线的形式可视化展示出来。并且 FATE-Board 还可以在可视化界面手动调

参，支持对模型训练过程全流程的跟踪、统计和监控等，并为模型运行状态、模型输出、日志追踪等提供了丰富的可视化呈现，帮助用户简单而高效地深入探索模型与理解模型。

#### 2.3.1.4 FATE-Serving

FATE-Serving 是 FATE 平台提供的高性能可扩展的联邦学习在线模型服务，其性能特性是 FATE 实现线上推理的核心。

# 第三章　基于 FATE 的联邦学习信贷风险预测模型

## 3.1　整体方案概要设计

本实验的主要目标是探讨集中式机器学习和联邦学习在金融风险预测任务中的应用，通过对比二者在准确度和模型性能方面的表现，从而得出有关这两种方法在实际信贷风险预测场景中的适用性和优劣的结论。为达到这一目标，实验设计为以下几个关键步骤：

1. 数据预处理：在进行实验之前，首先需要对 Lending Club 数据集进行预处理，以确保数据质量。预处理步骤包括处理缺失值、剔除异常值、进行特征选择和特征工程等。预处理后的数据将在后续步骤中用于训练和评估模型。

2. 集中式机器学习模型：搭建一个适用于信贷风险预测的集中式机器学习模型，选用逻辑回归算法进行模型训练、和评估。在这个阶段，关注模型在准确率、召回率、精确率等性能指标上的表现

3. 联邦学习模型：与集中式机器学习模型类似，搭建一个针对信贷风险预测的联邦学习模型。区别在于，联邦学习模型将分布式地在各数据拥有者处进行训练和更新，从而保护数据隐私。这会涉及实现与联邦学习框架（如 FATE）的对接、模型的加密策略等。在完成模型搭建后，利用 Lending Club 数据模拟分布式联邦学习过程，并对模型的性能进行测试

4. 实验结果对比：在两种模型搭建完成并进行训练测试后，进一步对比集中式机器学习模型与联邦学习模型在准确率、模型性能等方面的表现。通过对比，分析两种方法在信贷风险预测任务中的优劣以及其各自适用的场景

5. 结论与分析：在实验结果对比的基础上，进行深入的分析，挖掘在信贷风险预测场景下，集中式机器学习和联邦学习的优缺点、局限性及适用范围。为金融行业在实际应用中选择合适的风险预测方法提供指导

其中联邦学习系统的流程示意图如图所示: 本次实验希望深入了解集中式机器学习和联邦学习在金融风险预测任务中的可行性和效果，从而得出有关这两种方法在实际场景中的适用性和优劣的见解。在实验中遇到的挑战和实现细节将展示目前技术的局限性，为未来研究和优化提供借鉴。此外，本实验的结果将为金融机构提供有价值的参考，以便在实践中更好地应用集中式机器学习或联邦学习技术，从而更有效地进行信贷风险预测和控制。

### 3.1.1　实验基础与环境搭建

在数据集选择方面，本文关注借贷业务中的风险预测和控制。为此，本文选择了具有高度分析价值的 Lending Club 公开数据集（2007-2015）[32]，以便深入挖掘金融风险与

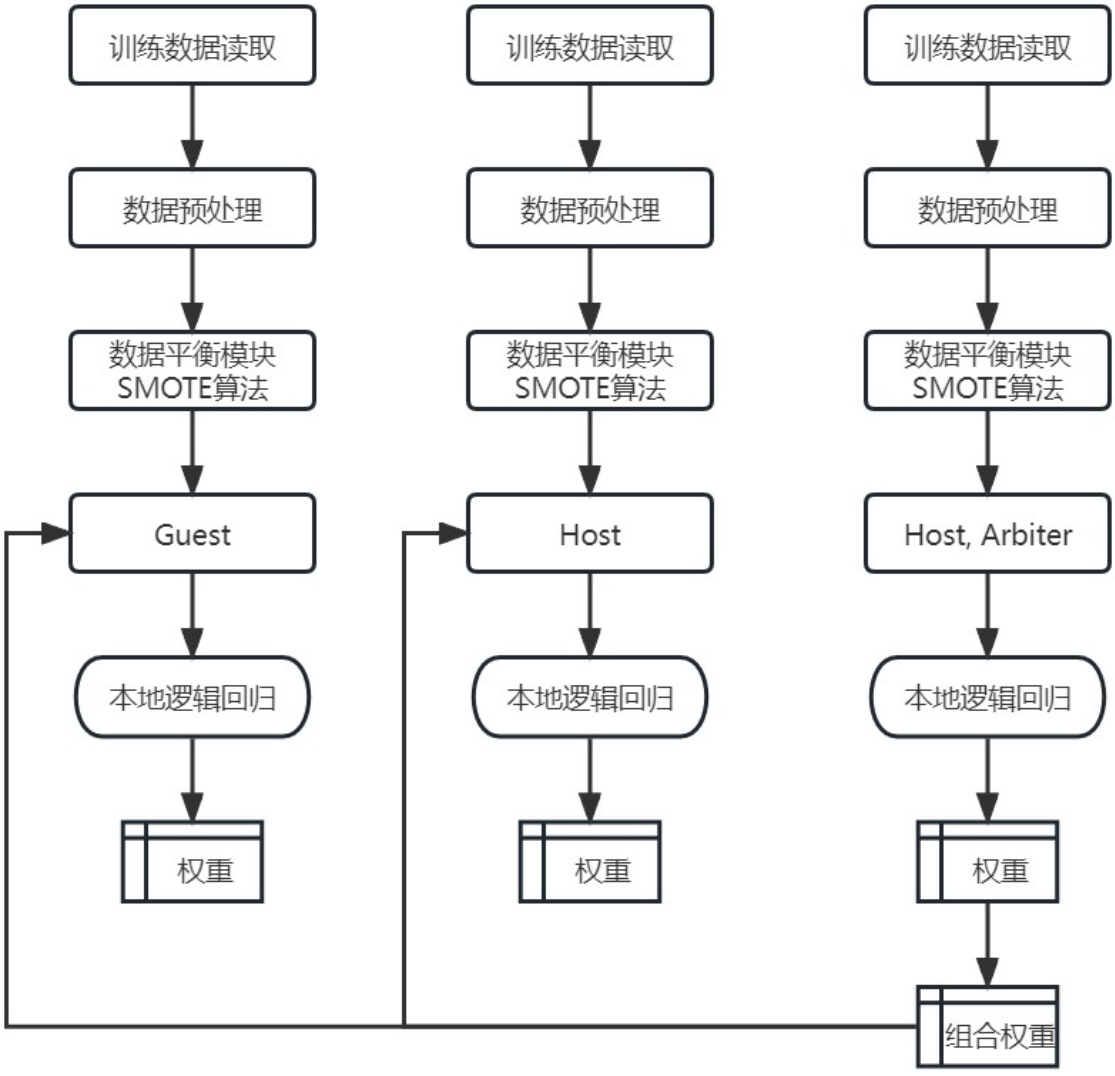



图 3-1 联邦学习实验流程

各类因素之间的关系，并基于此建立性能优越的风险预测模型。Lending Club 是一家颇具影响力的美国网络借贷平台，其数据集涵盖了大量实际借贷交易，为研究者们提供了丰富的、真实的研究素材。

选用这一数据集的主要原因有以下几点。首先，Lending Club 数据集包含了大量的样本数据，Lending Club 数据集（2007-2015）总计包含了约 88 万条借贷记录，这些记录反映了多年来无数信贷用户的金融行为和信用状况，可以提高分析和训练模型的可靠性，降低过拟合风险，从而确保预测结果的准确性。其次，该数据集涉及的特征非常多样，在该数据集中，共有 74 个特征涉及了贷款申请人的各类信息，例如年收入、信用评分、负债比例、就业年限、贷款目的以及逾期还款情况等。这些特征涵盖了几乎所有与借款人信用风险相关的维度，有助于挖掘各类影响因子与信贷风险之间的潜在联系，能够为深入了解信贷风险预测的关键驱动因素提供有力支持。

在本实验中，使用了三台 CentOS7 进行了联邦学习环境的搭建，其网络信息与在 FATE 框架中的节点配置如表 3-1 所示，拓扑如图 3-2 所示

表 3-1 联邦学习环境信息

| | | |
|---|---|---|
| CentOS7-1 | 192.168.52.131 | 10000 |
| CentOS7-2 | 192.168.52.132 | 9999 |
| CentOS7-3 | 192.168.52.133 | 9998 |

## 3.2 数据预处理

在获取得到 Lending Club 公开数据集（2007-2015）后，本文首先对数据集进行了预处理，以便为后续的模型训练和评估做好准备。具体而言，本文对数据集进行了缺失值处理、异常值处理、特征选择、特征编码等操作，以便为后续的模型训练和评估做好准备。

Lending Club 公开数据集（2007-2015）基础信息如表 3-2 所示：

### 3.2.1 缺失值处理

在进行数据分析和模型构建时，数据缺失情况是一个需要特别关注的问题。特征中存在大量缺失值的情况可能会降低模型的预测准确性并导致过拟合。因此，在预处理阶段处理缺失数据至关重要，这有助于提高模型的性能和准确性。

在本文的实验中，Lending Club 数据集存在缺失值较多且混乱的问题，某些特征的缺失值超过了 2/3 总数据量，从宏观上看，可参考图 3-3 。这种高度缺失的特征通常对模型的预测效果贡献甚微，甚至可能对模型产生负面影响。为了解决这个问题，需要对缺失值超过 2/3 的特征进行删除。这样做的目的是筛选掉对建模意义不大的特征，专注于那些对模型性能有重要贡献的特征。

```
1 thresh_count = len(data) * 2 / 3  # 设定阈值
```

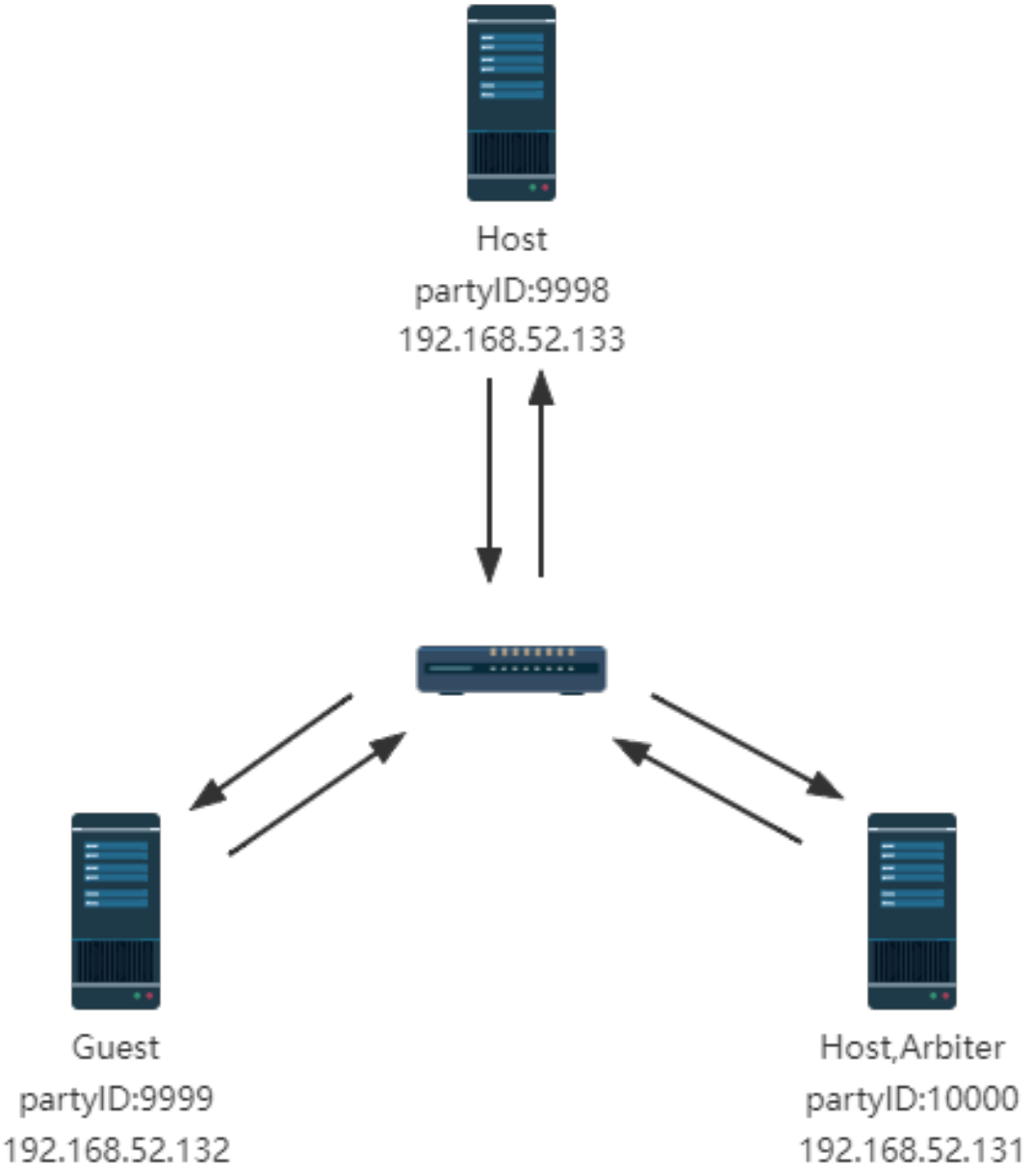

Host
partyID:9998
192.168.52.133
Guest
partyID:9999
192.168.52.132
Host,Arbiter
partyID:10000
192.168.52.131

图 3-2 集群环境拓扑图

表 3-2 初始数据集详细信息

| Column | Non-Null Count | Dtype |
|---|---|---|
| id | 887379 non-null | int64 |
| member_id | 887379 non-null | int64 |
| loan_amnt | 887379 non-null | float64 |
| funded_amnt | 887379 non-null | float64 |
| funded_amnt_inv | 887379 non-null | float64 |
| term | 887379 non-null | object |
| int_rate | 887379 non-null | float64 |
| installment | 887379 non-null | float64 |
| ... | ... | ... |
| ... | ... | ... |
| open_il_24m | 21372 non-null | float64 |
| mths_since_rcnt_il | 20810 non-null | float64 |
| total_bal_il | 21372 non-null | float64 |
| il_util | 18617 non-null | float64 |
| open_rv_12m | 21372 non-null | float64 |
| open_rv_24m | 21372 non-null | float64 |
| max_bal_bc | 21372 non-null | float64 |
| all_util | 21372 non-null | float64 |
| total_rev_hi_lim | 817103 non-null | float64 |
| inq_fi | 21372 non-null | float64 |
| total_cu_tl | 21372 non-null | float64 |
| inq_last_12m | 21372 non-null | float64 |

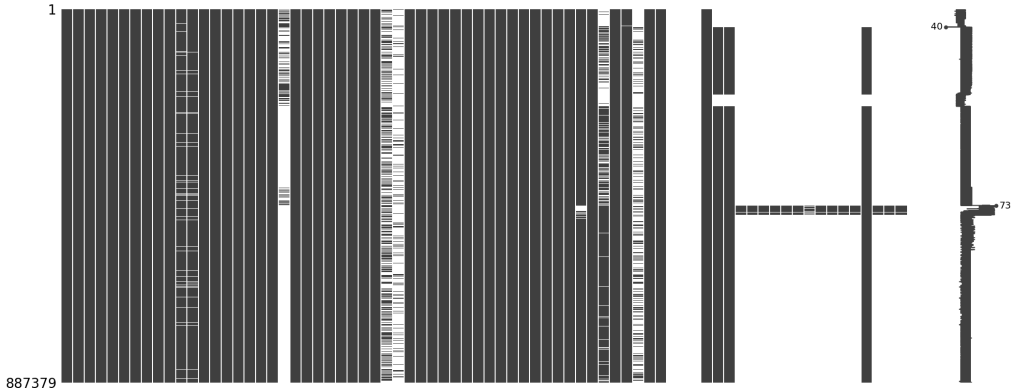



图 3-3 缺失值可视化

```
data = data.dropna(thresh=thresh_count,  axis=1)  #若某一列数据缺失的数量超过阈值就会被删除
```

代码 3-1 删除高度缺失的特征值，label

在删除高度缺失的特征值之后，还需要对剩余特征中存在的缺失值进行处理，以便确保分析和建模过程的有效性和准确性。针对剩余特征中的缺失值问题，缺失值主要集中在

少数样本中，这部分样本的数量约为一万条。相较于整个数据集，这部分缺失值样本占整体数据的比例较小，因此处理起来相对简单。

在处理这些含有缺失值的样本时，本文采取了删除这些数据的策略。这样的处理方式具有一定的可行性和合理性，原因如下：

1. 首先，由于包含缺失值的样本数量相对较少，删除这些样本对整个数据集的规模和数据分布影响有限。因此，这种方法不太可能引入显著的偏差，从而影响后续分析和建模的结果。

2. 其次，删除缺失值样本相较于填充缺失值具有更低的计算复杂度。填充缺失值的方法可能包括基于统计特性的填充（如平均值、中位数填充）或建立更复杂的缺失数据处理模型（如基于相似样本的插值、基于回归分析的填充等），这些方法在处理大量缺失值时可能带来较高的计算成本并可能引入误差。相反，删除缺失值样本则能避免这些问题，使处理过程更加高效和简洁。

3. 最后，有鉴于这些含有缺失值的样本占整个数据集的比例较低，直接删除这部分数据可能对模型的预测效果影响不大。而且，在一定程度上，删除这些缺失值样本也可以降低模型的噪声，从而提高模型的准确性。

通过对高度缺失的特征进行删除，以及清理含有缺失值的样本，目前得到了一个质量较高、干净的数据集。在此基础上，可以进一步进行数据分析、特征工程和模型构建等步骤

### 3.2.2 异常值处理

在处理完数据集的缺失值问题之后，数据集中仍然存在一些异常特征值

#### 3.2.2.1 单一唯一值特征

部分特征只包含单一唯一值，这样的特征实际上对最终的预测结果没有任何贡献。原因在于，这些仅具有单一值的特征在所有样本中都保持恒定，不具备区分能力，对模型性能的提升没有帮助。因此，为了提高模型的预测有效性和计算效率，应当将这些无关紧要的特征从数据集中删除，如代码3-2所示。

```
# 删除属性值只有一项的属性，因为其对预测没有任何意义
orig_columns = data.columns
drop_columns = []
for col in orig_columns:
  # 判断删除空值后是否是单一值
  col_series = data[col].dropna().unique()
  if len(col_series) == 1:
    drop_columns.append(col)
```

```
# 删除属性值单一的特征
data = data.drop(drop_columns,  axis=1)
```

代码 3-2 删除单一特征值

#### 3.2.2.2 特殊数据格式

数据集中的某些特征值呈现为 object 类型。这主要是由于这些特征值中包含百分号（%）或附带有”years” 这样的后缀。然而，这些非数值格式的特征值在机器学习模型的训练过程中是不兼容的。因此，需要将这些 object 类型的特征值转换为适当的数值数据类型，以便顺利进行后续的数据分析和建模过程。

对于”int_rate” 与”revol_util” 特征，需要删除其后缀百分号（%），如代码3-3所示

```
loans["int_rate"] = loans["int_rate"].str.rstrip("%").astype("float")
loans["revol_util"] = loans["revol_util"].str.rstrip("%").astype("float")
```

代码 3-3 删除百分号

对于”emp_length” 特征，需要删除其后缀”years”，如代码3-4所示

```
loans["int_rate"] = loans["int_rate"].str.rstrip("%").astype("float")
loans["revol_util"] = loans["revol_util"].str.rstrip("%").astype("float")
```

代码 3-4 删除 years

对于”term” 特征，需要删除其后缀”month”，如代码3-5所示

```
loans['term'] = loans['term'].apply(lambda x: int(x[:-7]))
```

代码 3-5 删除 month

通过对异常值进行处理，目前得到了较为可读的数据集。在此基础上，可以进一步进行特征选择、特征编码等步骤。

### 3.2.3 特征选择

经过前面多个阶段的数据预处理，需要对数据集进行特征选择。特征选择的目的是从原始特征中挑选出那些对预测模型有实际帮助的特征，同时剔除那些与预测目标关系不大或冗余的特征。这一步骤至关重要，因为选择合适的特征可以降低模型的复杂度，提高计算效率，避免过拟合现象，从而提高预测性能。

本实验根据特征的实际含义和数据分布情况进行筛选。针对数据集中的某些特征，如 id、member_id、zip_code、out_prncp、out_prncp_inv、total_pymnt_inv 等，与预测目标之间的关联性较弱，或者在预测过程中可能引入额外的噪声，因此选择将这些特征从数据集中删除。

具体操作步骤如下：

1. 识别待删除特征：建立一个待删除特征的列表，按照实际意义，将 id、member_id、zip_code、out_prncp、out_prncp_inv、total_pymnt_inv 等特征加入列表中。

2. 删除特征：从数据集中逐一移除这些待删除特征，确保数据保持一致性和完整性

经过上述处理步骤，得到了一个精简且只包含相关特征的数据集，如表 3-3 所示。在这个新数据集的基础上，模型将能以较高的计算效率进行训练。同时，避免冗余特征导致的过拟合问题，并有望提升预测结果的准确性。

表 3-3 特征选择后数据集详细信息

| Column | Non-Null Count | Dtype |
|---|---|---|
| loan_amnt | 242851 non-null | float64 |
| term | 242851 non-null | int64 |
| int_rate | 242851 non-null | float64 |
| installment | 242851 non-null | float64 |
| emp_length | 242851 non-null | int64 |
| home_ownership | 242851 non-null | object |
| annual_inc | 242851 non-null | float64 |
| verification_status | 242851 non-null | object |
| loan_status | 242851 non-null | int64 |
| purpose | 242851 non-null | object |
| dti | 242851 non-null | float64 |
| delinq_2yrs | 242851 non-null | float64 |
| inq_last_6mths | 242851 non-null | float64 |
| open_acc | 242851 non-null | float64 |
| pub_rec | 242851 non-null | float64 |
| revol_bal | 242851 non-null | float64 |
| revol_util | 242851 non-null | float64 |
| total_acc | 242851 non-null | float64 |

### 3.2.4 特征编码

在完成特征选择后，数据集中的 verification_status、home_ownership、purpose、loan_status 这四个特征仍是 object 类型。然而，这些特征实际上都是分类特征，因此不能直接将它们纳入数值型机器学习模型中进行训练。为处理这些分类特征，需要对它们进行独热编码（One-Hot Encoding）操作，从而将这些离散的分类特征转换为一种机器学习模型可以接受的数值型输入形式，如代码3-6所示。

```
cat_columns = ["home_ownership", "verification_status", "purpose"]
dummy_df = pd.get_dummies(data[cat_columns])
data = pd.concat([data, dummy_df], axis=1)
data = data.drop(cat_columns, axis=1)
```

代码 3-6 one hot 编码

最后，需要对 loan_status 特征进行特殊处理，本实验只需要最终发放和不发放贷款的情况，其他的情况删除不考虑。因此需要只保留贷款成功和而不成功的情况，删除需要等待的情况，并将成功申请的情况用 1 代替，将申请失败的情况使用 0 代替，如代码3-7所示。

```
data = data[(data['loan_status'] == "Fully Paid") |
          (data['loan_status'] == "Charged Off")]
status_replace = {
  "loan_status": {
    "Fully Paid": 1,
    "Charged Off": 0,
  }
}
# 将成功申请的情况用1代替，将申请失败的情况使用0代替
data = data.replace(status_replace)
print(data.shape)
```

代码 3-7 loanStatus 特征处理

至此完成了对数据集的预处理，数据集具体信息如表 附-1 所示。

## 3.3 处理样本不平衡

在 Lending Club 信贷数据集中，正负样本集数量差别较大，如图 3-4 所示。样本的不平衡会对模型学习造成困扰。举例来说，假如有 100 个样本，其中只有 1 个是贷款违约样本，其余 99 个全为贷款正常样本，那么学习器只要制定一个简单的方法：所有样本均判别为正常样本，就能轻松达到 99% 的准确率。而这个分类器的决策对本实验的风险控制毫无意义。因此，在将数据代入模型训练之前，必须先解决样本不平衡的问题。非平衡样本常用的解决方式有 2 种：

1. 过采样（oversampling），增加正样本使得正、负样本数目接近，然后再进行学习。
2. 欠采样（undersampling），去除一些负样本使得正、负样本数目接近，然后再进行学习。

本实验采用的方法是过采样，具体操作使用 SMOTE（Synthetic Minority Oversampling Technique），SMOTE 的基本原理是：采样最邻近算法，计算出每个少数类样本的 K 个近邻，从 K 个近邻中随机挑选 N 个样本进行随机线性插值，构造新的少数样本，同时将新样本与原数据合成，产生新的训练集。

经过过采样后的数据集比例如图 3-5 所示。

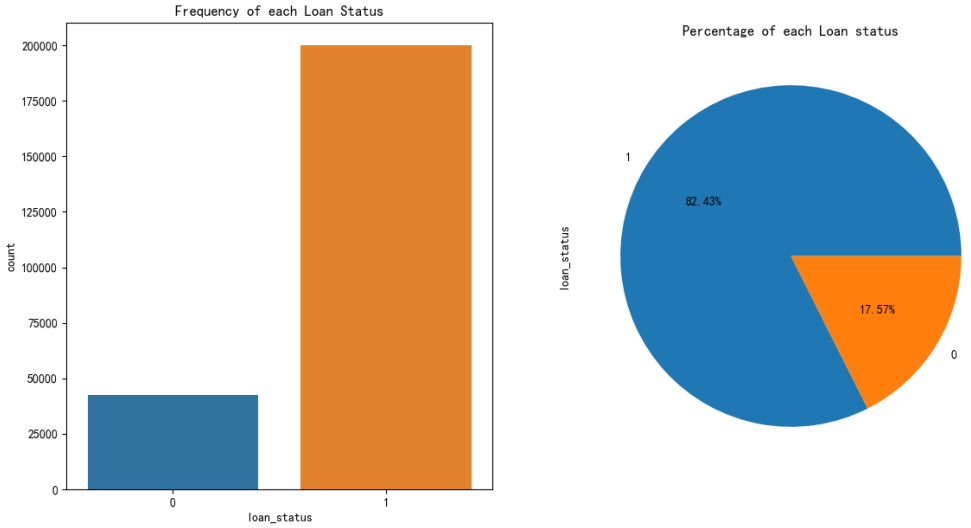



图 3-4 正负样本比例图

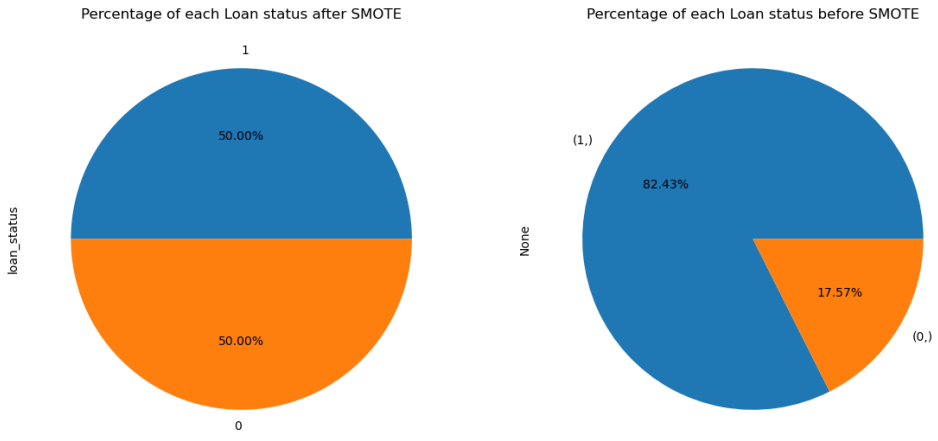



图 3-5 SMOTE 过采样前后对比

## 3.4 在集中式环境下训练风险预测模型

在集中式机器学习的信贷风险预测中，为了对机器学习模型的性能进行评估，首先需要将整个数据集划分为训练集和测试集。本实验采用了常用的数据划分策略，将 30% 的数据作为测试集用于评估模型的泛化能力，而剩余的 70% 的数据被划分为训练集，用于训练逻辑回归模型。

loan_status 仅有两种状态，本实验是一个二分类问题，因此本实验选择了逻辑回归作为训练模型的算法，逻辑回归是一种广泛应用于二分类问题的线性模型，具有易于理解、训练速度快等优点。其原理如下：

首先，给出逻辑回归模型的一般形式，设输入特征向量为 $x = (x_1, x_2, \cdots, x_n)^T$，逻辑回归模型的输出为：

$$P(y = 1 \mid x) = \frac{1}{1 + \exp\left(-w^T x\right)} \qquad \text{式（3-1）}$$

其中，$w = (w_1, w_2, \cdots, w_n)^T$ 为模型参数。

模型将线性回归的输出通过 sigmoid 函数进行转换，使其值在 0 到 1 之间，表示为正类的概率。对于损失函数，此处采用对数损失 (Logistic Loss)：$L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$。

在求解模型参数 $w$ 时，通常通过最大似然估计 (MLE) 的方式优化损失函数，其中 $\arg\max_w L(w) = \arg\min_w - \log L(w)$，可以通过梯度下降 (Gradient Descent) 等方法进行求解。在训练完成后，逻辑回归模型可以直接通过输入特征计算得到预测值，并设定一个阈值（0.5），大于阈值的样本预测为正类，否则为负类。

评估模型性能的过程中，本实验选定了真负率（Specificity）和召回率（Recall）作为主要的评估指标。真负率衡量了模型在识别负样本（即无风险贷款）方面的表现，而召回率评估了模型在检测正样本（即存在风险的贷款）方面的能力。这两项指标对于金融风险预测任务至关重要，因为它们可以帮助了解模型在不同方面的性能表现。通过对比这两个指标的数值，可以更准确地评估模型的整体性能，并据此对模型进行调整和优化。

结果如表 3-4 所示

表 3-4 集中式机器学习评估结果

| | precision | recall | f1-score |
|---|---|---|---|
| Charged Off | 0.28 | 0.66 | 0.40 |
| Fully Paid | 0.90 | 0.65 | 0.75 |

## 3.5 在联邦学习环境下训练风险预测模型

### 3.5.1 基于 FATE 设计并实现联邦学习风险预测算法

在联邦学习环境下训练风险预测模型过程中，涉及以下几个关键步骤：

1. 首先，需要搭建一个稳定的联邦学习环境。这包括了配置硬件设备、配置必要的运行环境以及建立适当的数据传输通道。确保参与方可以顺畅并安全地进行联邦学习

2. 其次，在建立联邦学习环境之后，要将数据上传至相应的主机并进行预处理

3. 然后，为模型训练和评估过程进行详细定义。这包括选择合适的算法、设置超参数以及确保所选用的评估标准能够客观地反映模型的性能

4. 最后，在模型训练和评估完成后，需要对结果进行分析对比

### 3.5.2 联邦学习数据的分割上传

#### 3.5.2.1 数据集的分割

在联邦学习的应用场景中，用户数据通常分布式地存储在不同公司或组织的数据库中，而非集中在一个单一的数据库。这样的分布式数据存储方式可以保护用户隐私，同时增加了数据处理的复杂性。为了模拟这种真实的场景，本实验将原本集中式环境下的训练数据分为了三个独立的数据子集。

这种划分数据集的方法旨在模拟现实世界中类似的情况，即各公司和组织仅访问和处理其拥有的用户数据。通过对这三份独立的数据子集进行分别处理和分析，实验旨在评估联邦学习框架在处理分布式数据时的性能和效果。与此同时，将训练数据分为三份的另一个优势在于，可以更好地理解不同数据分布对模型训练结果的影响。

因此，本实验将数据集按表 3-5 所示分割为了三部分，分别对应三台 CentOS7 主机。

表 3-5 数据集分割信息

| | | |
|---|---|---|
| CentOS7-1 | 25% | 10000 |
| CentOS7-2 | 25% | 9999 |
| CentOS7-3 | 50% | 9998 |

#### 3.5.2.2 数据集的上传

为了提升联邦建模的易用性，FATE-v1.5 开始引入 Pipeline 模块，该模块提供了一套可视化的数据处理流程，用户可以通过配置文件定义数据处理流程，然后通过命令行工具提交任务，FATE 会自动执行数据处理流程，最终生成可用于建模的数据集。在本实验中，使用了 Pipeline 模块对数据集进行了上传，操作流程如下所示：

1. 对 pipeline 进行初始化，配置接下来将要连接的 IP 与端口

2. 创建 python 脚本并引入 pipeline 模块，定义各主机职责，这里定义 guest 为 9999，host 为 9998、10000

3. 定义数据存储分区、表名和命名空间，这将在 FATE 模型配置中使用

4. 使用 pipeline_upload.add_upload_data 定义上传数据表操作，如代码3-8所示

5. 使用 pipeline_upload.upload(drop=1) 完成上传

```
pipeline_upload.add_upload_data(file=os.path.join(data_base, "train.csv"),
                                table_name=dense_data_guest["name"],
                                namespace=dense_data_guest["namespace"],
                                head=1, partition=partition)

pipeline_upload.add_upload_data(file=os.path.join(data_base, "test.csv"),
                table_name=test_data_guest["name"],
                namespace=test_data_guest["namespace"],
                head=1, partition=partition)
```

代码 3-8 定义上传数据表操作

### 3.5.3 联邦学习风险预测模型训练

#### 3.5.3.1 pipeline 初始化

在进行模型训练流程定义之前，需要先对 pipeline 进行初始化来配置提交各主机在联邦任务中各自的角色，如代码3-9所示。

```
pipeline = PipeLine() \
      .set_initiator(role='guest', party_id=9999) \
      .set_roles(guest=[9999], host=[9998, 10000], arbiter=10000)
```

代码 3-9 pipeline 初始化

#### 3.5.3.2 定义数据读取（Reader）模块

在配置好各主机在联邦任务中各自的角色之后，需要定义联邦学习风险预测模型训练的流程；首先需要定义的就是数据读取模块，数据读取模块负责从指定的数据分区中根据命名空间与表名读取数据到工作流中，如代码所示。

```
reader_0 = Reader(name="reader_0")
reader_0.get_party_instance(role='guest', party_id=9999).component_param(
  table={"name": "9999_data", "namespace": "experiment"})
reader_0.get_party_instance(role='host', party_id=10000).component_param(
  table={"name": "10000_data", "namespace": "experiment"})
reader_0.get_party_instance(role='host', party_id=9998).component_param(
```

```
table={"name": "9998_data", "namespace": "experiment"})
```

代码 3-10 reader 定义，label

#### 3.5.3.3 定义数据处理（DataTransform）模块

DataTransform 模块主要负责数据预处理和特征工程的任务。这个模块提供了一系列实用的数据处理方法和工具，以便在进行联邦学习建模之前清洗、转换和优化数据，由于之前本实验已经对数据进行过预处理，因此这里本实验仅使用 DataTransform 模块来进行训练数据与标签的区分，如代码3-11所示。

```
data_transform_0 = DataTransform(name="data_transform_0", with_label=True)
data_transform_0.get_party_instance(role='guest', party_id=[9999]).
    component_param(
  with_label=True, label_name="loan_status", label_type="int", output_format="
      dense")
data_transform_0.get_party_instance(role='host', party_id=[10000]).
    component_param(
  with_label=True, label_name="loan_status", label_type="int", output_format="
      dense")
data_transform_0.get_party_instance(role='host', party_id=[9998]).
    component_param(
  with_label=True, label_name="loan_status", label_type="int", output_format="
      dense")
```

代码 3-11 DataTransform 定义

#### 3.5.3.4 定义横向逻辑回归（HomoLR）模块

横向逻辑回归（HomoLR）是联邦学习中一种重要的分类算法。在这个模型中，参与计算的各方（例如客户端和主机）拥有相同的特征空间，但数据样本可能不同。HomoLR 利用加密方式在各方之间进行安全地梯度交换和聚合，从而保护各方的数据隐私，并实现在分布式数据上训练一个全局性的逻辑回归模型。与传统的逻辑回归（LR）相似，HomoLR 的训练过程主要包括以下步骤：

1. 初始化 HomoLR 模型。各参与方的模型具有相同的结构。在每次迭代中，每个参与方在自己的数据上训练局部模型。

2. 加密梯度计算：使用可选的加密模式（目前 FATE 仅支持 Paillier 算法）计算主机方的梯度。这样一来，该主机不再可获得明文模型。这有助于保护各方之间的数据隐私。

3. 梯度交换与聚合：各参与方将加密（或明文，取决于设置）梯度上报给 arbiter。arbiter 聚合这些梯度以形成联邦梯度，然后将其分发给所有参与方以更新它们的局部模型。

4. 收敛判断与停止条件：与传统的 LR 类似，当联邦模型收敛或整个训练过程达到预定的最大迭代次数之后，训练过程将停止。

在 FATE 中使用 HomoLR 如代码3-12所示，对参数进行了部分定义，在保证较大规模分布式数据训练效果的同时，降低计算资源开销。以下是对这些参数的详细说明及其作用：

```
homo_lr_0 = HomoLR(name="homo_lr_0", tol=0.0001, alpha=1.0, optimizer='rmsprop', batch_size=-1, early_stop='diff')
```

代码 3-12 HomoLR 定义

1. name=”homo_lr_0”：为 HomoLR 模型实例命名，便于后续训练和评估

2. tol=0.0001：容差（tolerance）参数用于设置模型收敛的阈值。当模型的优化程度达到该阈值时，训练过程将提前终止

3. alpha=1.0：设置正则化系数，用于控制模型的正则化强度以防止过拟合

4. optimizer=’rmsprop’：设置优化器为 RMSProp。优化器用于更新模型权重以最小化损失函数。RMSProp 是一种自适应学习率方法，适用于非凸优化问题，可以加速模型收敛过程

5. batch_size=-1：batch 大小用于确定每次参数更新前要处理的数据样本数。当 batch_size=-1 时，表示使用所有数据样本作为一个 batch。这意味着模型将在每个完整的数据集上执行一次参数更新

6. early_stop=’diff’：提前停止策略，当模型收敛到一定程度时（基于定义的容差值）提前终止训练，有助于节省计算资源并防止过拟合

#### 3.5.3.5 定义评估（Evaluation）模块

Evaluation 模块在 FATE 中负责提供各种评估指标和方法，用以衡量模型训练效果和性能。本实验利用这一模块来评估通过联邦学习训练得到的模型的效果和性能，如代码3-13所示

```
evaluation_0 = Evaluation(name="evaluation_0", eval_type="binary")
```

代码 3-13 Evaluation 定义

此处为 Evaluation 模块实例定义了名称”evaluation_0”，并将其评估类型设置为”binary”，表明对一个二分类问题的模型性能进行评估。Evaluation 模块提供了多种评估指标，例如准确率（Accuracy）、精确度（Precision）、召回率（Recall）、F1 值（F1-score）、AUC、ROC 曲线等。这些评估指标可以帮助实验更全面地衡量模型在训练集和测试集上的表现，从而判断模型是否足够泛化以处理新的数据，并发现潜在的过拟合或欠拟合问题。

#### 3.5.3.6 配置 pipeline 工作流结构

在完成单个模块的定义后，需要将这些独立的模块组合成一个完整的 Pipeline 工作流。为了实现这一目标，需要按照执行顺序和数据处理逻辑将各个模块添加到 Pipeline 中，并定义它们之间的数据依赖关系，如代码3-14所示

```
pipeline.add_component(reader_0)
pipeline.add_component(data_transform_0,  data=Data(data=reader_0.output.data))
pipeline.add_component(homo_lr_0,  data=Data(data=data_transform_0.output.data))
pipeline.add_component(evaluation_0, data=Data(homo_lr_0.output.data))
pipeline.compile()
```

代码 3-14 pipeline 工作流定义

上述代码描述了 Pipeline 工作流的结构和数据流向：

1. 首先，将 Reader 组件添加到工作流中，读取和加载数据集
2. 接着，将 DataTransform 组件添加到工作流中，依赖于 Reader 组件的输出数据。在这个组件中完成训练数据与标签的划分
3. 接着，将 HomoLR 组件加入工作流，该组件依赖于 DataTransform 组件的输出数据。此时，HomoLR 模型开始在训练数据上进行训练
4. 最后，在工作流中添加 Evaluation 组件，依赖于 HomoLR 组件的输出数据。此时，Evaluation 模块会计算并输出模型在不同评估指标方面的表现

完成工作流结构和依赖关系的定义后，最后使用 pipeline.compile() 完成 Pipeline 的编译。

通过以上步骤，本实验成功地将各个解耦的模块组建成一个完整的 Pipeline 工作流。接下来，可以开始训练模型，并利用 Evaluation 模块评估模型在训练集和测试集上的性能。训练过程如图 附-1 所示

# 第四章　风险预估模型实验结果与分析

在现实金融场景中，各金融机构都拥有一部分信贷风险控制数据，但由于法律法规对客户隐私保护的要求，这些数据无法在金融机构之间共享以提高风险预测模型的准确性。然而，联邦学习（FL）技术正好可以解决这个问题。

联邦学习允许不同金融机构在不彼此共享原始数据的情况下，共同训练一个模型。这样可以避免隐私泄露风险，同时充分利用各方的数据资源，提升模型的性能。

为了验证联邦学习技术在金融信贷领域的优势和实际应用效果，实验将一个涉及三家金融机构数据的联邦学习模型与各公司各自独立训练的模型进行比较，对一份独立于训练集之外的测试集进行预测。

由于测试集中正负样本的不平衡，如图 4-1 所示，使用准确率作为评价指标可能不是最佳选择。正负样本不平衡意味着模型在训练过程中可能会偏向正面样本较多的类别，从而在预测高风险投资时表现不佳。因此，关注真负率有助于更精确地评估模型在区分高风险投资方面的表现。它允许投资者在预警高风险投资方面更有信心，进而优化投资决策。

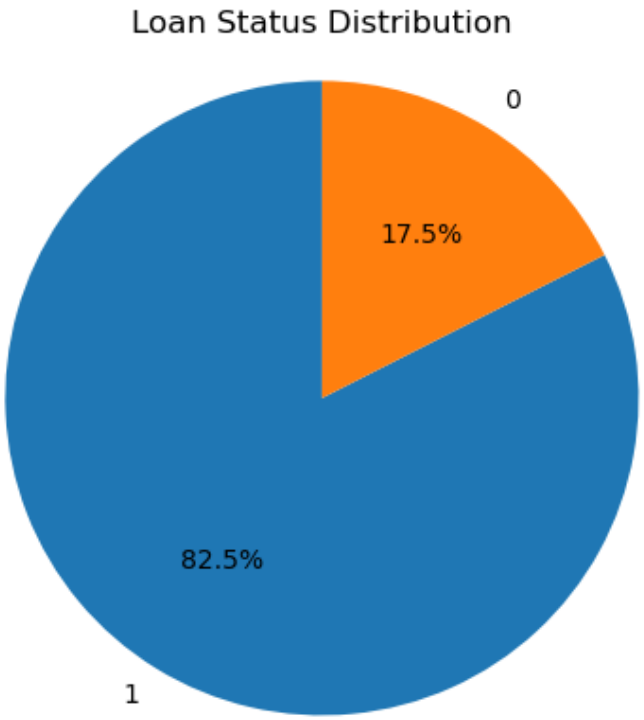



图 4-1　test 数据集分布

在金融风险预警模型中，模型的主要目标是识别那些具有高风险的投资，因此在风险预估模型的实际应用中，强调真负率作为主要评价标准有助于降低误判率和减少金融

损失，所以，真负率（特异性，判别负样本的准确度）作为一种更合适的评价标准可能会更为有效。

## 4.1 FATE 联邦模型与各方本地模型对比结果

经过模型预测与指标计算，联邦模型与各方机构本地模型真负率对比图如图 4-2 所示。从图中可以明显观察到，在三家金融机构进行联邦学习训练后，模型的风险预警能力得到了显著提升。

这个结果证明了联邦学习对于金融行业等涉及数据隐私和安全的领域，具有重要的实际价值。通过应用联邦学习技术，各金融机构可以在保障数据安全、遵守相关法规的基础上，充分利用合作方的数据资源来共同优化他们的风险控制策略。这不仅有助于提高风险识别准确度，还可更有效地为客户提供定制化服务。

除了金融场景之外，联邦学习也适用于其他需要进行跨机构数据联合训练的领域，例如医疗、教育和零售等。通过联邦学习技术，各个机构可以在保护用户隐私和数据安全的前提下，充分发挥数据的价值，实现业务优势的共享和提升。总体来看，联邦学习无疑将在多个行业发挥巨大的推动作用。

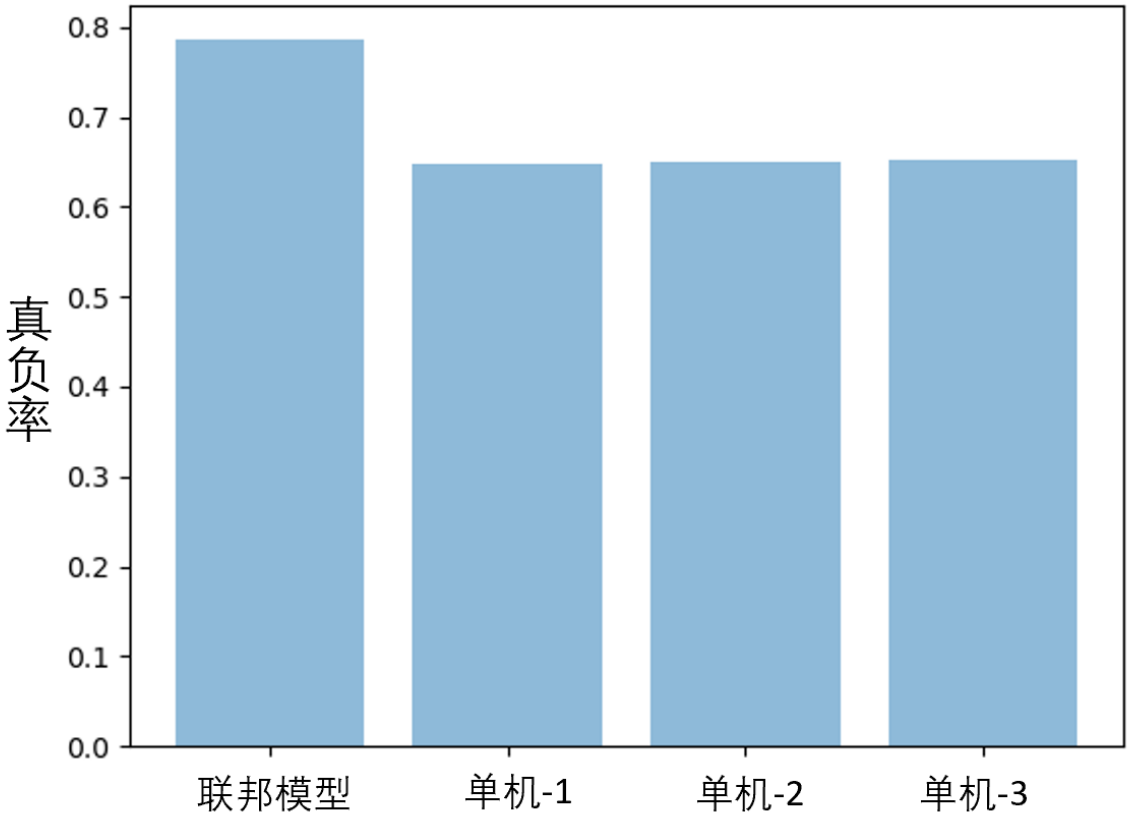



图 4-2 联邦模型与各方本地模型对比结果

## 4.2 FATE 联邦模型在不同联邦数据分布下的性能对比分析

在联邦学习场景中，各参与方的数据集分布通常是不均匀的，这可能是由于各方面的业务差异、客户特征以及其他不同因素的影响。因此，在实际应用中需要考虑这些不同场景以确保联邦学习具有更好的泛用性与健壮性。

本实验旨在通过测试不同数据分布下的模型表现，来评估联邦学习在不同数据分布条件下的效果。通过这种方式，可以探讨在不同业务场景下联邦学习技术能够带来的商业价值和性能改进。

为了实现这个目标，本实验测试了在两家机构的条件下进行联邦，数据分布从不均匀到均匀的情况下模型的预测效果变化，如图 4-3 所示。可以看到，在总数据量不变而数据分布更加均匀的情况下，模型的效果会更加准确。这个结论对现实商业活动中使用联邦学习具有很大的指导意义，说明了不同机构在合作过程中需要对各方数据集的大小与分布进行一定的评估，以此来决定是否合作以及预估合作背后的商业价值。

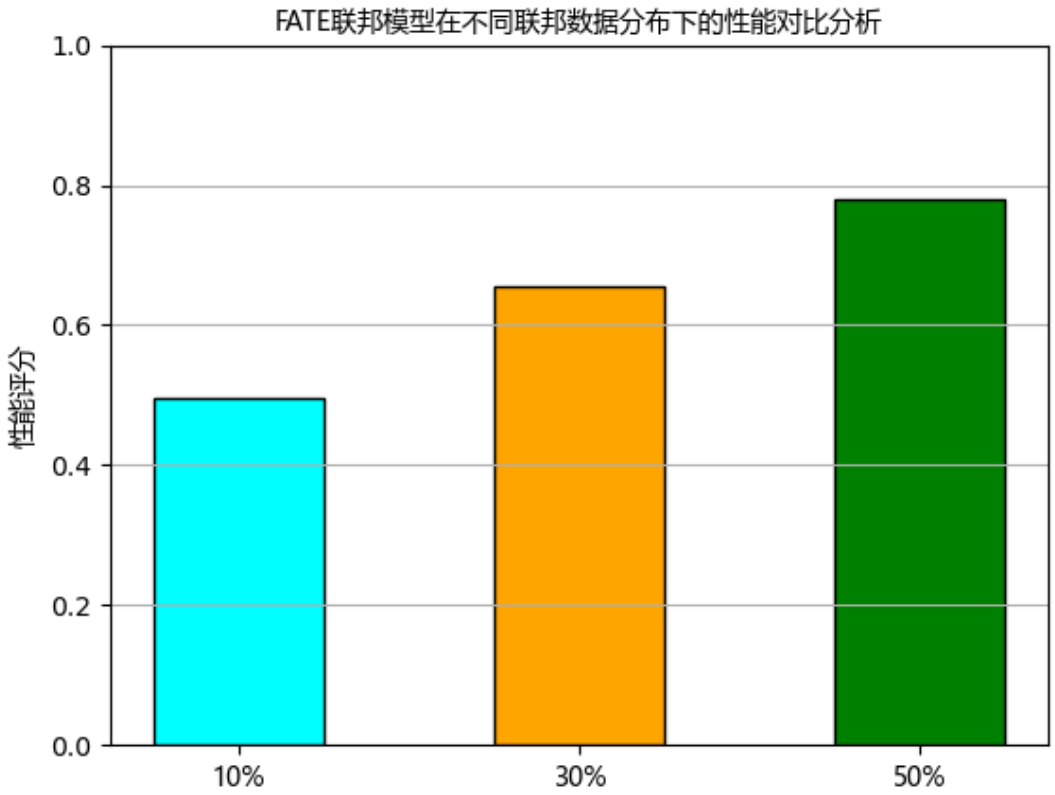



图 4-3 数据分布模型效果对比

## 4.3 FATE 联邦模型的安全性分析

FATE 框架的安全性主要来源于使用了多方安全计算 (MPC) 以及同态加密 (HE) 技术构建底层安全计算协议。通过源数据不出本地而仅交互模型更新（如梯度信息）的方式来保护用户的敏感数据。联邦学习中客户端通过训练源数据上传本地模型，服务器仅负责聚合和分发每轮迭代形成的全局模型，从而避免源数据的泄露。然而，在真实的网络环境中，模型反演攻击、成员推理攻击、模型推理攻击层出不穷，参与训练的客户端动机难以判断，中心服务器的可信程度难以保证，仅通过模型更新来保护用户隐私的方式显然是不够的。

因此，FATE 使用了多方安全计算 (MPC) 以及同态加密 (HE) 技术来保证联邦学习过程中的安全性。

1. 安全多方计算 (Multi-Party Computation，MPC)：安全多方计算 (MPC) 是一种允许多方在不泄露数据本身的情况下，共同计算某个函数的方法。在 FATE 联邦学习框架中，MPC 保证了数据局部性和保密性，使得机构能够在不暴露自己数据的情况下进行合作。FATE 的 MPC 基于密码学原理，包括安全加密和解密、脱敏数据传输和处理等，以确保联邦学习过程中每个参与方的信息安全

2. 同态加密 (Homomorphic Encryption)：同态加密是一种加密方法，它允许对密文数据执行计算操作，并在不解密的情况下生成最终结果。在 FATE 联邦学习框架中，同态加密用于数据的安全计算和聚合。Paillier 同态加密是 FATE 常用的同态加密模型，它使得各参与方能够在保护数据隐私的同时，实现合作计算和模型训练

本实验中，主要使用的 Federated Logistic Regression 组件使用了 Paillier 同态加密算法进行梯度更新，保证了梯度数据传输与计算的安全性。但是同时造成了较大的通信开销和计算开销[33]，如何平衡通信负担和模型安全仍是一个问题与挑战。

# 第五章　总结与展望

本文通过实验证明了联邦学习在实际应用中的有效性和优越性。具体而言，联邦学习不仅在传统机器学习技术的基础上提供了隐私计算，为用户带来强大的隐私保护机制，而且在保证模型精度的同时实现了更优的性能。本研究成功地运用金融信贷场景的数据，完成了联邦学习风险预测模型的设计、训练与实现，并将其与单机构数据训练的模型进行了对比。分析结果表明，借助联邦学习，金融机构之间的合作将能得到极大的推动力。

同时，本文通过对不同数据分布下联邦学习训练结果的对比分析，成功测试了数据分布对联邦训练效果的影响。研究结论显示，在总数据量不变的情况下，数据分布越均匀，联邦学习模型训练效果越佳。在现实商业活动中，各合作方应当充分参考各自数据集的分布特点，以便更深刻地评估联邦学习背后的商业价值。

在大数据安全的背景下，联邦学习为解决数据安全、数据泄露、用户隐私等问题提供了一种高效的解决方案。尽管联邦学习采取了原始数据不离开本地的方式来保护用户隐私，但这种方式在某些方面仍存在提升空间，如计算成本较高等。因此，在未来的研究中，可以进一步探索更加安全且高效的隐私计算技术，将其融入联邦学习框架中，为解决现实应用中的问题提供更为强大的支持。

# 参考文献


[1] Rieke N Li W, Hancox J. The future of digital health with federated learning [J]. NPJ digital medicine. 2020: 1–7.

[2] 微众银行 AI 项目组. 联邦学习白皮书 V1.0[R]. 2018.

[3] Zhao Y, Li M, Lai L et al. Federated learning with non-iid data [J]. arXiv preprint arXiv:1806.00582. 2018.

[4] Yang T, Andrew G, Eichner H et al. Applied federated learning: Improving google keyboard query suggestions [J]. arXiv preprint arXiv:1812.02903. 2018.

[5] Kim H, Park J, Bennis M et al. Blockchained on-device federated learning [J]. IEEE Communications Letters. 24 (6). 2019: 1279–1283.

[6] Zhang C, Xie Y, Bai H et al. A survey on federated learning [J]. Knowledge-Based Systems. 216. 2021: 106775.

[7] Leroy D, Coucke A, Lavril T et al. Federated Learning for Keyword Spotting [C]. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019 : 6341–6345.

[8] Chen M, Mathews R, Ouyang TY et al. Federated Learning of out-of-Vocabulary Words [J]. arXiv preprint arXiv:1812.02903. 2019.

[9] Hard A, Rao K, Mathews R et al. Federated Learning for Mobile Keyboard Prediction [J]. arXiv preprint arXiv:1811.03604. 2018.

[10] Feng J, Rong C, Sun F et al. PMF: A Privacy-preserving Human Mobility Prediction Framework Via Federated Learning [J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies.

[11] Sozinov K, Vlassov V, Girdzijauskas S. Human Activity Recognition Using Federated Learning [C]. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). 2018 .

[12] Yu T, Li T, Sun Y et al. Learning Context-Aware Policies from Multiple Smart Homes Via Federated Multi-task Learning [C]. In 2020 IEEE/ACM Fifth International Conference on Internet-of-things Design and Implementation (IoTDI). 2020 .

[13] Aïvodji UM, Gambs S, Martin A. LOFTLA: A Secured and Privacy-preserving Smart Home Architecture Implementing Federated Learning [C]. In 2019 IEEE Security and Privacy Workshops (SPW). 2019 .

[14] Silva S, Gutman BA, Romero E et al. Federated Learning in Distributed Medical Databases: Meta-analysis of Large-Scale Subcortical Brain Data [C]. In 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). 2019 .

[15] Gao D, Ju C, Wei X et al. HHHFL: Hierarchical Heterogeneous Horizontal Federated Learning for Electroencephalography [J]. arXiv preprint arXiv:1909.05784. 2019.

[16] Kim Y, Sun J, Yu H et al. Federated Tensor Factorization for Computational Phenotyping [C]. In Pro-

ceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Halifax, NS, Canada. 2007 : 887–895.

[17] Pfohl SR, Dai AM, Heller KA. Federated and Differentially Private Learning for Electronic Health Records [J]. arXiv preprint arXiv:1911.05861. 2019.

[18] Brisimi TS, Chen R, Mela T et al. Federated Learning of Predictive Models from Federated Electronic Health Records [J]. International Journal of Medical Informatics. 112. 2018.

[19] Huang L, Shea AL, Qian H et al. Patient Clustering Improves Efficiency of Federated Machine Learning to Predict Mortality and Hospital Stay Time Using Distributed Electronic Medical Records [J]. Journal of Biomedical Informatics. 99. 2019.

[20] Huang L, Yin Y, Fu Z et al. Loadaboost: Loss-based Adaboost Federated Machine Learning with Reduced Computational Complexity on Iid and Non-IID Intensive Care Data [J]. PLOS ONE.

[21] 陈琨, 李艺, 王国赛. 联邦学习在金融行业的应用分析 [J]. 征信. 39 (10). 2021: 29–36.

[22] 潘碧莹, 丘海华, 张家伦. 不同数据分布的联邦机器学习技术研究 [C]. In 5G 网络创新研讨会 (2019) 论文集. 2019 .

[23] Yang Qiang, Liu Yang, Chen Tianjian et al. Federated machine learning: Concept and applications [J]. ACM Transactions on Intelligent Systems and Technology. 10 (2). 2019: 1–19.

[24] Vaidya J, Clifton C. Privacy preserving association rule mining in vertically partitioned data [C]. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002 : 639–644.

[25] Pan S J, Yang Qiang. A survey on transfer learning [J]. IEEE Transactions on Knowledge and Data Engineering. 22 (10). 2010: 1345–1359.

[26] McMahan H Brendan, Moore Eider, Ramage Daniel et al. Federated learning of deep networks using model averaging [J]. arXiv preprint arXiv:1602.05629. 2016.

[27] 刘俊旭, 孟小峰. 机器学习的隐私保护研究综述 [J]. 计算机研究与发展. 57 (2). 2020: 346.

[28] Dwork C, Mc-Sherry F, Nissim K et al. Calibrating noise to sensitivity in private data analysis [C]. In Theory of Cryptography Conference. 2006 : 265–284.

[29] 苏冠通, 徐茂桐. 安全多方计算技术与应用综述 [J]. 信息通信技术与政策. (5). 2019: 19–22.

[30] Dolev D, Yao A. On the security of public key protocols [J]. IEEE Transactions on Information Theory. 29 (2). 1983: 198–208.

[31] FATE 开源文档. https://github.com/FederatedAI/FATE.

[32] Lending Club Data. https://www.lendingclub.com/info/download-data.action.

[33] Geyer R C, Klein T, Nabi M. Differentially private federated learning: a client level perspective [J]. arXiv preprint arXiv:1712.07557. 2017.

# 致　　谢

光阴荏苒，不知不觉我的本科生涯已进入尾声。在此，我以无比感慨和喜悦之情呈现我的毕业论文。从论文选题、分析实验，到最终的论文撰写，都离不开导师、家人和同学们的悉心关照与支持。在此，我真挚地向他们表示衷心的感谢！

首先，我要感谢我的本科导师陆月明教授。在论文实施的整个过程中，陆教授对我的悉心教导和热情关怀给了我无尽的动力与信心。在学术研究中，他为人师表，敢于担当，严谨治学。在生活中，他关心学生，热情亲切。陆老师为我树立了良好的学术榜样，使我在攻读本科期间受益匪浅。

其次，我要感谢我们实验室的师兄师姐和同学们。他们脚踏实地，严谨治学，为我树立了良好的学术榜样。在学术研究中，他们的建议和指导使我在论文完成的过程中受益良多；在生活上，同学间的关爱与互动让我感受到实验室的团结和温馨。

再次，我要向我的家人表示诚挚的感谢。父母对我的关爱和支持使我更加坚定地走向理想的彼岸；亲朋好友在不同的关键时刻给予了我心灵的慰藉和精神的支持。他们一直是我努力奋斗、不断前行的最大动力。

在此本科毕业之际，我由衷地感谢与我一同走过本科学习历程的每一个人。未来的路还长，我将继续勇往直前，不断努力，为未来的人生实践书写新的篇章。

# 附　　录

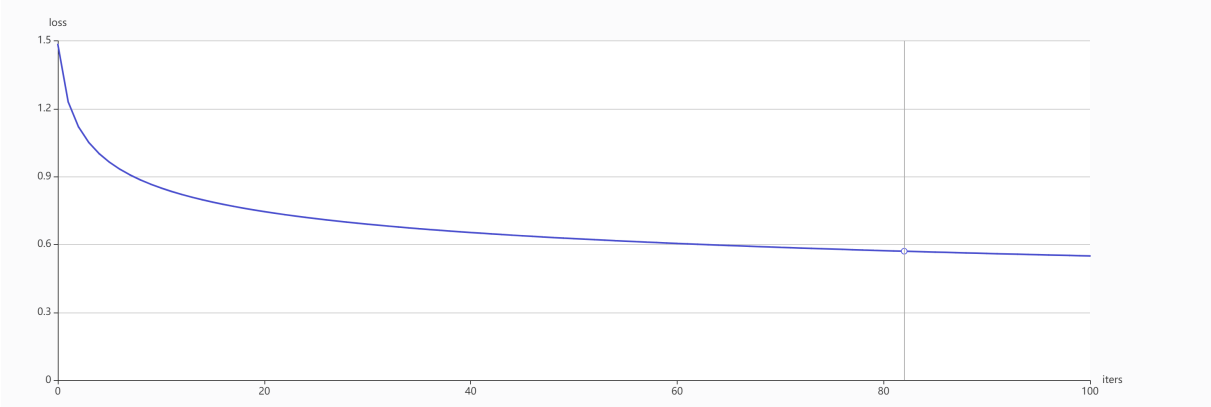



图　附-1　训练过程 loss 变化

表 附-1 预处理后数据集信息

| 列名 | 非空计数 | 数据类型 |
|---|---|---|
| loan_amnt | 242851 | float64 |
| term | 242851 | float64 |
| int_rate | 242851 | float64 |
| installment | 242851 | float64 |
| emp_length | 242851 | float64 |
| annual_inc | 242851 | float64 |
| dti | 242851 | float64 |
| delinq_2yrs | 242851 | float64 |
| inq_last_6mths | 242851 | float64 |
| open_acc | 242851 | float64 |
| pub_rec | 242851 | float64 |
| revol_bal | 242851 | float64 |
| revol_util | 242851 | float64 |
| total_acc | 242851 | float64 |
| home_ownership_ANY | 242851 | float64 |
| home_ownership_MORTGAGE | 242851 | float64 |
| home_ownership_NONE | 242851 | float64 |
| home_ownership_OTHER | 242851 | float64 |
| home_ownership_OWN | 242851 | float64 |
| home_ownership_RENT | 242851 | float64 |
| verification_status_Not Verified | 242851 | float64 |
| verification_status_Source Verified | 242851 | float64 |
| verification_status_Verified | 242851 | float64 |
| purpose_car | 242851 | float64 |
| purpose_credit_card | 242851 | float64 |
| purpose_debt_consolidation | 242851 | float64 |
| purpose_educational | 242851 | float64 |
| purpose_home_improvement | 242851 | float64 |
| purpose_house | 242851 | float64 |
| purpose_major_purchase | 242851 | float64 |
| purpose_medical | 242851 | float64 |
| purpose_moving | 242851 | float64 |
| purpose_other | 242851 | float64 |
| purpose_renewable_energy | 242851 | float64 |
| purpose_small_business | 242851 | float64 |
| purpose_vacation | 242851 | float64 |
| purpose_wedding | 242851 | float64 |
| loan_status | 242851 | int64 |

外 文 资 料

# Inverting Gradients - How easy is it to break privacy in federated learning?

**Jonas Geiping*** **Hartmut Bauermeister** * **Hannah Dröge** *

**Michael Moeller**

Dep. of Electrical Engineering and Computer Science
University of Siegen
{jonas.geiping, hartmut.bauermeister, hannah.droege,
michael.moeller }@uni-siegen.de

## Abstract

The idea of federated learning is to collaboratively train a neural network on a server. Each user receives the current weights of the network and in turns sends parameter updates (gradients) based on local data. This protocol has been designed not only to train neural networks data-efficiently, but also to provide privacy benefits for users, as their input data remains on device and only parameter gradients are shared. But how secure is sharing parameter gradients? Previous attacks have provided a false sense of security, by succeeding only in contrived settings - even for a single image. However, by exploiting a magnitude-invariant loss along with optimization strategies based on adversarial attacks, we show that is is actually possible to faithfully reconstruct images at high resolution from the knowledge of their parameter gradients, and demonstrate that such a break of privacy is possible even for trained deep networks. We analyze the effects of architecture as well as parameters on the difficulty of reconstructing an input image and prove that any input to a fully connected layer can be reconstructed analytically independent of the remaining architecture. Finally we discuss settings encountered in practice and show that even aggregating gradients over several iterations or several images does not guarantee the user's privacy in federated learning applications.

## 1 Introduction

Federated or collaborative learning [6, 28] is a distributed learning paradigm that has recently gained significant attention as both data requirements and privacy concerns in machine learning continue to rise [21, 14, 32]. The basic idea is to train a machine learning model, for example a neural network, by optimizing the parameters $\theta$ of the network using a loss function $\mathcal{L}$ and exemplary training data consisting of input images $x_i$ and corresponding labels $y_i$ in order to solve

$$\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_\theta(x_i, y_i). \tag{1}$$

We consider a distributed setting in which a *server* wants to solve (1) with the help of multiple *users* that own training data $(x_i, y_i)$. The idea of federated learning is to only share the gradients $\nabla_\theta \mathcal{L}_\theta(x_i, y_i)$ instead of the original data $(x_i, y_i)$ with the server which it subsequently accumulates to

*Authors contributed equally.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 1: Reconstruction of an input image $x$ from the gradient $\nabla_\theta \mathcal{L}_\theta(x, y)$. Left: Image from the validation dataset. Middle: Reconstruction from a trained ResNet-18 trained on ImageNet. Right: Reconstruction from a trained ResNet-152. In both cases, the intended privacy of the image is broken. Note that previous attacks cannot recover either ImageNet-sized data [35] or attack trained models.

update the overall weights. Using gradient descent the server's updates could, for instance, constitute

$$\theta^{k+1} = \underbrace{\theta^k - \tau \sum_{i=1}^{N}}_{\text{server}} \underbrace{\nabla_\theta \mathcal{L}_{\theta^k}(x_i, y_i)}_{\text{users}} . \tag{2}$$

The updated parameters $\theta^{k+1}$ are sent back to the individual users. The procedure in eq. (2) is called *federated SGD*. In contrast, in *federated averaging* [17, 21] each user computes several gradient descent steps locally, and sends the updated parameters back to the server. Finally, information about $(x_i, y_i)$ can be further obscured, by only sharing the mean $\frac{1}{t}\sum_{i=1}^{t} \nabla_\theta \mathcal{L}_{\theta^k}(x_i, y_i)$ of the gradients of several local examples, which we refer to as the *multi-image* setting.

Distributed learning of this kind has been used in real-world applications where user privacy is crucial, e.g. for hospital data [13] or text predictions on mobile devices [3], and it has been stated that "Privacy is enhanced by the ephemeral and focused nature of the [Federated Learning] updates" [3]: model updates are considered to contain less information than the original data, and through aggregation of updates from multiple data points, original data is considered impossible to recover. In this work we show analytically as well as empirically, that parameter gradients still carry significant information about the supposedly private input data as we illustrate in Fig. 1. We conclude by showing that even *multi-image federated averaging* on realistic architectures does not guarantee the privacy of all user data, showing that out of a batch of 100 images, several are still recoverable.

**Threat model:** We investigate an *honest-but-curious* server with the goal of uncovering user data: The attacker is allowed to separately store and process updates transmitted by individual users, but may *not* interfere with the collaborative learning algorithm. The attacker may not modify the model architecture to better suit their attack, nor send malicious global parameters that do not represent the actually learned global model. The user is allowed to accumulate data locally in Sec. 6. We refer to the supp. material for further commentary and mention that the attack is near-trivial under weaker constraints on the attacker.

In this paper we discuss privacy limitations of federated learning first in an academic setting, honing in on the case of gradient inversion from one image and showing that

- Reconstruction of input data from gradient information is possible for realistic deep and non-smooth architectures with both, trained and untrained parameters.
- With the right attack, there is little "defense-in-depth" - deep networks are as vulnerable as shallow networks.
- We prove that the input to any fully connected layer can be reconstructed analytically independent of the remaining network architecture.

Then we consider the implications that the findings have for practical scenarios, finding that

- Reconstruction of multiple, separate input images from their averaged gradient is possible in practice, over multiple epochs, using local mini-batches, or even for a local gradient averaging of up to 100 images.

## 2 Related Work

Previous related works that investigate recovery from gradient information have been limited to shallow networks of less practical relevance. Recovery of image data from gradient information was first discussed in [25, 24] for neural networks, who prove that recovery is possible for a single neuron or linear layer. For convolutional architectures, [31] show that recovery of a single image is possible for a 4-layer CNN, albeit with a significantly large fully-connected (FC) layer. Their work first constructs a "representation" of the input image, that is then improved with a GAN. [35] extends this, showing for a 4-layer CNN (with a large FC layer, smooth sigmoid activations, no strides, uniformly random weights), that missing label information can also be jointly reconstructed. They further show that reconstruction of multiple images from their averaged gradients is indeed possible (for a maximum batch size of 8). [35] also discuss deeper architectures, but provide no tangible results. A follow-up [34] notes that label information can be computed analytically from the gradients of the last layer. These works make strong assumptions on the model architecture and model parameters that make reconstructions easier, but violate the threat model that we consider in this work and lead to less realistic scenarios.

The central recovery mechanism discussed in [31, 35, 34] is the optimization of an euclidean matching term. The cost function

$$\arg\min_x ||\nabla_\theta \mathcal{L}_\theta(x, y) - \nabla_\theta \mathcal{L}_\theta(x^*, y)||^2 \tag{3}$$

is minimized to recover the original input image $x^*$ from a transmitted gradient $\nabla_\theta \mathcal{L}_\theta(x^*, y)$. This optimization problem is solved by an L-BFGS solver [18]. Note that differentiating the gradient of $\mathcal{L}$ w.r.t to $x$ requires a second-order derivative of the considered parametrized function and L-BFGS needs to construct a third-order derivative approximation, which is challenging for neural networks with ReLU units for which higher-order derivatives are discontinuous.

A related, but easier problem, compared to the full reconstruction of input images, is the retrieval of input attributes [23, 10] from local updates, e.g. does a person that is recognized in a face recognition system wear a hat. Information even about attributes unrelated to the task at-hand can be recovered from deeper layers of a neural network, which can be recovered from local updates.

Our problem statement is furthermore related to model inversion [9], where training images are recovered from network parameters after training. This provides a natural limit case for our setting. Model inversion generally is challenging for deeper neural network architectures [33] if no additional information is given [9, 33]. Another closely related task is inversion from visual representations [8, 7, 20], where, given the output of some intermediate layer of a neural network, a plausible input image is reconstructed. This procedure can leak some information, e.g. general image composition, dominating colors - but, depending on the given layer it only reconstructs similar images - if the neural network is not explicitly chosen to be (mostly) invertible [11]. As we prove later, inversion from visual representations is strictly more difficult than recovery from gradient information.

## 3 Theoretical Analysis: Recovering Images from their Gradients

To understand the overall problem of breaking privacy in federated learning from a theoretical perspective, let us first analyze the question if data $x \in \mathbb{R}^n$ can be recovered from its gradient $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$ analytically.

Due to the different dimensionality of $x$ and $\nabla_\theta \mathcal{L}_\theta(x, y)$, reconstruction quality is surely is a question of the number of parameters $p$ versus input pixels $n$. If $p < n$, then reconstruction is at least as difficult as image recovery from incomplete data [4, 2], but even when $p > n$, which we would expect in most computer vision applications, the difficulty of regularized "inversion" of $\nabla_\theta \mathcal{L}_\theta$ relates to the non-linearity of the gradient operator as well as its conditioning.

Interestingly, fully-connected layers take a particular role in our problem: As we prove below, the input to a fully-connected layer can always be computed from the parameter gradients analytically independent of the layer's position in a neural network (provided that a technical condition, which

prevents zero-gradients, is met). In particular, the analytic reconstruction is independent of the specific types of layers that precede or succeed the fully connected layer, and a single input to a fully-connected network can always be reconstructed analytically without solving an optimization problem.The following statement is a generalization of Example 3 in [24] to the setting of arbitrary neural networks with arbitrary loss functions:

**Proposition 3.1.** *Consider a neural network containing a biased fully-connected layer preceded solely by (possibly unbiased) fully-connected layers. Furthermore assume for any of those fully-connected layers the derivative of the loss $\mathcal{L}$ w.r.t. to the layer's output contains at least one non-zero entry. Then the input to the network can be reconstructed uniquely from the network's gradients.*

*Proof.* In the following we give a sketch of the proof and refer to the supplementary material for a more detailed derivation. Consider an unbiased full-connected layer mapping the input $x_l$ to the output, after e.g. a ReLU nonlinearity: $x_{l+1} = \max\{A_l x_l, 0\}$ for a matrix $A_l$ of compatible dimensionality. By assumption it holds $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(x_{l+1})_i} \neq 0$ for some index $i$. Then by the chain rule $x_l$ can be computed as $\left(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(x_{l+1})_i}\right)^{-1} \cdot \left(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(A_l)_{i,:}}\right)^T$. This allows the iterative computation of the layers' inputs as soon as the derivative of $\mathcal{L}$ w.rt. a certain layer's output is known. We conclude by noting that adding a bias can be interpreted as a layer mapping $x_k$ to $x_{k+1} = x_k + b_k$ and that $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}x_k} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}b_k}$. □

Another interesting aspect in view of the above considerations is that many popular network architectures use fully-connected layers (or cascades thereof) as their last prediction layers. Hence the input to those prediction modules being the output of the previous layers can be reconstructed. Those activations usually already contain some information about the input image thus exposing them to attackers. For example [23] show that these features representations can be mined for image attributes by training an auxiliary malicious classifier that recognizes attributes that are not part of the main task. Further interesting in this regard is the possibility to reconstruct the ground truth label information from the gradients of the last fully-connected layer as discussed in [34]. Finally, Prop. 3.1 allows to conclude that for any classification network that ends with a fully connected layer, reconstructing the input from a parameter gradient is strictly easier than inverting visual representations, as discussed in [8, 7, 20], from their last convolutional layer.

## 4 A Numerical Reconstruction Method

As image classification networks rarely start with fully connected layers, let us turn to the numerical reconstruction of inputs: Previous reconstruction algorithms relied on two components; the euclidean cost function of Eq. (3) and optimization via L-BFGS. We argue that these choices are not optimal for more *realistic* architectures and especially *arbitrary* parameter vectors. If we decompose a parameter gradient into its norm magnitude and its direction, we find that the magnitude only captures information about the state of training, measuring local optimality of the datapoint with respect to the current model (for strongly convex functions the gradient magnitude is even an upperbound on distance to the optimal solution). In contrast, the high-dimensional direction of the gradient can carry significant information, as the angle between two data points quantifies the change in prediction at one datapoint when taking a gradient step towards another [5, 16]. As such we propose to use a cost function based on angles, i.e. cosine similarity, $l(x, y) = \langle x, y\rangle / (||x|| ||y||)$. In comparison to Eq. (3), the objective is not to find images with a gradient that best fits the observed gradient, but to find images that lead to a similar change in model prediction as the (unobserved!) ground truth. This is equivalent to minimizing the euclidean cost function, if one additionally constrains both gradient vectors to be normalized to a magnitude of 1.

We further constrain our search space to images within $[0, 1]$ and add only total variation [27] as a simple image prior to the overall problem, cf. [31]:

$$\arg\min_{x\in[0,1]^n} 1 - \frac{\langle \nabla_\theta \mathcal{L}_\theta(x, y), \nabla_\theta \mathcal{L}_\theta(x^*, y)\rangle}{||\nabla_\theta \mathcal{L}_\theta(x, y)|| ||\nabla_\theta \mathcal{L}_\theta(x^*, y)||} + \alpha\, \mathrm{TV}(x). \tag{4}$$

Secondly, we note that our goal of finding some inputs $x$ in a given interval by minimizing a quantity that depends (indirectly, via their gradients) on the outputs of intermediate layers, is related to the task of finding adversarial perturbations for neural networks [29, 19, 1]. As such, we minimize eq.

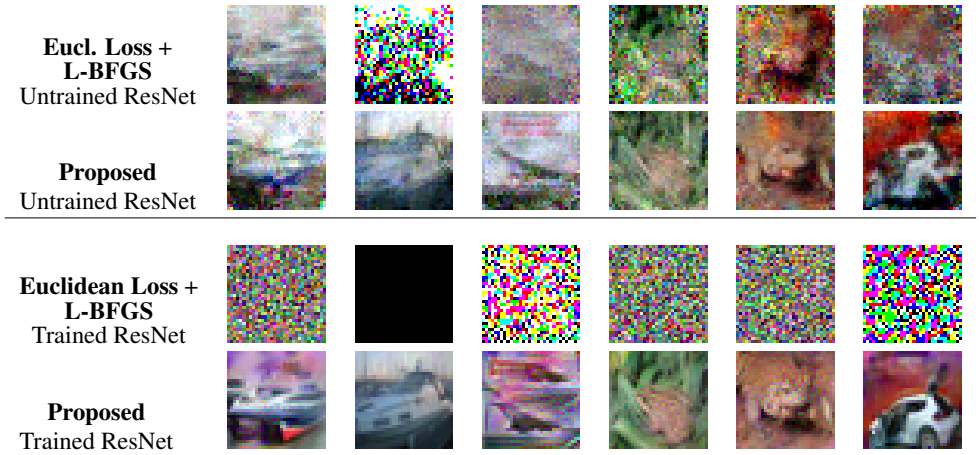



Figure 2: Baseline comparison for the network architectures shown in [31, 35].We show the first 6 images from the CIFAR-10 validation set.

(4) only based on the sign of its gradient, which we optimize with Adam [15] with step size decay. Note though that signed gradients only affect the first and second order momentum for Adam, with the actual update step still being unsigned based on accumulated momentum, so that an image can still be accurately recovered.

Applying these techniques leads to the reconstruction observed in Fig. 1. Further ablation of the proposed mechanism can be found in the appendix. We provide a `pytorch` implementation at `https://github.com/JonasGeiping/invertinggradients`.

This attack is, due to the double backpropagation, roughly twice as expensive as a single minibatch step per gradient step on the objective eq. (4). In this work, we conservatively run the attack for up to 24000 iterations, with a relatively small step size, as computational costs are not our main concern at this moment (and we assume that the attacker that is breaking privacy potentially has order-of-magnitude more computational power than the user), yet we note that smarter step size rules and larger step sizes can lead to successful attacks with a budget of only several hundred iterations.

**Remark** (Optimizing label information)**.** *While we could also consider the label $y$ as unknown in Eq. (4) and optimize jointly for $(x, y)$ as in [35], we follow [34] who find that label information can be reconstructed analytically for classification tasks. Thus, we consider label information to be known.*

## 5 Single Image Reconstruction from a Single Gradient

Similar to previous works on breaking privacy in a federated learning setting, we first focus in the reconstruction of a single input image $x \in \mathbb{R}^n$ from the gradient $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$. This setting serves as a proof of concept as well as an upper bound on the reconstruction quality for the multi-image distributed learning settings we consider in Sec. 6. While previous works have already shown that a break of privacy is possible for single images, their experiments have been limited to rather shallow, smooth, and untrained networks. In the following, we compare our proposed approach to prior works, and conduct detailed experiments on the effect that architectural- as well as training-related choices have on the reconstruction. All hyperparameter settings and more visual results for each experiment are provided in the supp. material.

**Comparison to previous approaches.** We first validate our approach by comparison to the Euclidean loss (3) optimized via L-BFGS considered in [31, 35, 34]. This approach can often fail due to a bad initialization, so we allow a generous setting of 16 restarts of the L-BFGS solver. For a quantitative comparison we measure the mean PSNR of the reconstruction of $32 \times 32$ CIFAR-10 images over the first 100 images from the validation set using the same shallow and smooth CNN as in [35], which we denote as "LeNet (Zhu)" as well as a ResNet architecture, both with trained and untrained parameters. Table 1 compares the reconstruction quality of euclidean loss (3) with L-BFGS optimization (as in [31, 35, 34]) with the proposed approach. The former works extremely well for

Table 1: PSNR mean and standard deviation for 100 experiments on the first images of the CIFAR-10 validation data set over two different networks with trained an untrained parameters.

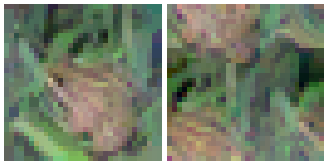
| Architecture | LeNet (Zhu) | | ResNet20-4 | |
|---|---|---|---|---|
| Trained | False | True | False | True |
| Eucl. Loss + L-BFGS | $\mathbf{46.25 \pm 12.66}$ | $13.24 \pm 5.44$ | $10.29 \pm 5.38$ | $6.90 \pm 2.80$ |
| Proposed | $18.00 \pm 3.33$ | $\mathbf{18.08 \pm 4.27}$ | $\mathbf{19.83 \pm 2.96}$ | $\mathbf{13.95 \pm 3.38}$ |

Figure 3: Single-Image Reconstruction from the parameter gradients of trained ResNet-152. Top row: Ground Truth. Bottom row: Reconstruction. We check every 1000th image of the ILSVRC2012 validation set. The amount of information leaked per image is highly dependent on image content - while some examples like the two tenches are highly compromised, the black swan (ironically) leaks almost no usable information. Noticeable is also the loss of positional information in several images.

the untrained, smooth, shallow architecture, but completely fails on the trained ResNet. We note that [31] applied a GAN to enhance image quality from the LBFGS reconstruction, which, however, fails, when the representative is too distorted to be enhanced. Our approach provides recognizable images and works particularly well on the realistic setting of a trained ResNet as we can see in Figure 2. Interestingly, the reconstructions on the trained ResNet have a better visual quality than those of the untrained ResNet, despite their lower PSNR values according to table 1. Let us study the effect of trained network parameters in an even more realistic setting, i.e., for reconstructing ImageNet images from a ResNet-152.

**Trained vs. untrained networks.** If a network is trained and has sufficient capacity for the gradient of the loss function $\mathcal{L}_\theta$ to be zero for different inputs, it is obvious that they can never be distinguished from their gradient. In practical settings, however, owing to stochastic gradient descent, data augmentation and a finite number of training epochs, the gradient of images is rarely entirely zero. While we do observe that image gradients have a much smaller magnitude in a trained network than in an untrained one, our magnitude-oblivious approach of (4) still recovers important visual information based only on the direction of the trained gradients.

We observe two general effects on trained networks that we illustrate with our ImageNet reconstructions in Fig. 3: First, reconstructions seem to become *implicitly biased* to typical features of the same class in the training data, e.g., the more blueish feathers of the capercaillie in the 5th image, or the large eyes of the owl in the inset figure. Thus, although the overall privacy of most images is clearly breached, this effect at least obstructs the recovery of fine scale details or the image's background. Second, we find that the data augmentation used during the training of neural networks leads to trained networks that make the *localization* of objects more difficult: Notice how few of the objects in Fig. 3 retain their original position and how the snake and gecko duplicate. Thus, although image reconstruction with networks trained with data augmentation still succeeds, some location information is lost.



**Translational invariant convolutions.** Let us study the ability to obscure the location of objects in more detail by testing how a conventional convolutional neural network, that uses convolutions with zero-padding, compares to a provably translationally invariant CNN, that uses convolutions with circular padding. As shown in the inset figure, while the conventional CNN allows

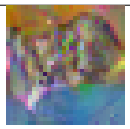
| **Original** | **ResNet-18** with base width: 16 | 64 | 128 | **ResNet-34** | **ResNet-50** |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| PSNR | 17.24 | 17.37 | 25.25 | 18.62 | 21.36 |
| Avg. PSNR | 19.02 | 22.04 | 22.94 | 21.59 | 20.98 |
| Std. | 2.84 | 5.89 | 6.83 | 4.49 | 5.57 |

Figure 4: Reconstructions of the original image (left) for multiple ResNet architectures. The PSNR value refers to the displayed image while the avg. PSNR is calculated over the first 10 CIFAR-10 images. The standard deviation is the average standard deviation of one experiment under a given architecture. The ResNet-18 architecture is displayed for three different widths.

for recovery of a rather high quality image (left), the translationally invariant network makes the localization of objects impossible (right) as the original object is separated. As such we identify the common zero-padding as a source of privacy risk.

**Network Depth and Width.** For classification accuracy, the depth and number of channels of each layer of a CNN are very important parameters, which is why we study their influence on our reconstruction. Figure 4 shows that the reconstruction quality measurably increase with the number of channels. Yet, the larger network width is also accompanied with an increasing variance of experimental success. However with multiple restarts of the experiment, better reconstructions can be produced for wider networks, resulting in PSNR values that increases from 19 to almost 23 for when increasing the number of channels from 16 to 128.As such, greater network width increases the computational effort of the attacker, but does not provide greater security.

Looking at the reconstruction results we obtain from ResNets with different depths, the proposed attack degrades very little with an increased depth of the network. In particular - as illustrated in Fig. 3, even faithful ImageNet reconstructions through a ResNet-152 are possible.

## 6 Distributed Learning with Federated Averaging and Multiple Images

So far we have only considered recovery of a single image from its gradient and discussed limitations and possibilities in this setting. We now turn to strictly more difficult generalized setting of *Federated Averaging* [21, 22, 26] and *multi-image* reconstruction, to show that the proposed improvements translate to this more practical case as well, discussing possibilities and limits in this application.

Instead of only calculating the gradient of a network's parameters based on local data, federated averaging performs multiple update steps on local data before sending the updated parameters back to the server. Following the notation of [21], we let the local data on the user's side consist of $n$ images. For a number $E$ of local epochs the user performs $\frac{n}{B}$ stochastic gradient update steps per epoch, where $B$ denotes the local mini-batch size, resulting in a total number of $E\frac{n}{B}$ local update steps. Each user $i$ then sends the locally updated parameters $\tilde{\theta}_i^{k+1}$ back to the server, which in turn updates the global parameters $\theta^{k+1}$ by averaging over all users.

We empirically show that even the setting of federated averaging with $n \geq 1$ images is potentially amenable for attacks. To do so we try to reconstruct the local batch of $n$ images by the knowledge of the local update $\tilde{\theta}_i^{k+1} - \theta^k$. In the following we evaluate the quality of the reconstructed images for different choices of $n$, $E$ and $B$. We note that the setting studied in the previous sections corresponds to $n = 1$, $E = 1$, $B = 1$. For all our experiments we use an untrained ConvNet.

**Multiple gradient descent steps, $B = n = 1$, $E > 1$:**
Fig. 5 shows the reconstruction of $n = 1$ image for a varying number of local epochs $E$ and different choices of learning rate $\tau$. Even for a high number of 100 local gradient descent steps the reconstruction quality is unimpeded. The only failure case we were able to exemplify was induced by picking a high learning rate of 1e-1. This setup, however, corresponds to a step size that would lead to a divergent training update, and as such does not provide useful model updates.

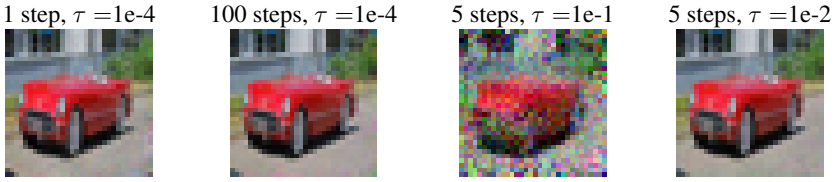



Figure 5: Illustrating the influence of the number of local update steps and the learning rate on the reconstruction: The left two images compare the influence of the number of gradient descent steps for a fixed learning rate of $\tau$ =1e-4. The two images on the right result from varying the learning rate for a fixed number of 5 gradient descent steps. PSNR values are shown below the images.

Figure 6: Information leakage from the aggregated gradient of a batch of 100 images on CIFAR-100 for a ResNet32-10. Shown are the 5 *most* recognizable images from the whole batch. Although most images are unrecognizable, privacy is broken even in a large-batch setting. We refer to the supplementary material for all images.

**Multi-Image Recovery,** $B = n > 1$**,** $E = 1$:
So far we have considered the recovery of a single image only, and it seems reasonable to believe that averaging the gradients of multiple (local) images before sending an update to the server, restores the privacy of federated learning. While such a multi-image recovery has been considered in [35] for $B \leq 8$, we demonstrate that the proposed approach is capable of restoring some information from a batch of 100 averaged gradients: While most recovered images are unrecognizable (as shown in the supplementary material), Fig. 6 shows the 5 most recognizable images and illustrates that even averaging the gradient of 100 images does not entirely secure the private data. Most surprising is that the distortions arising from batching are *non-uniform*. One could have expected all images to be equally distorted and near-irrecoverable, yet some images are highly distorted and others only to an extend at which the pictured object can still be recognized easily, which demonstrates that privacy leaks are conceivable even for large batches of image data.

Note that the attacker in this scenario only has knowledge about the average of gradients, however we assume the number of participating images to be known to the server. The server might request this information anyway (for example to balance heterogeneous data), but even if the exact number of images is unknown, the server (which we assume to have significantly more compute power than the user) could run reconstructions over a range of candidate numbers, given that the number of images is only a small integer value and then select the solution with minimal reconstruction loss.

**General case**
We also consider the general case of multiple local update steps using a subset of the whole local data in each mini batch gradient step. An overview of all conducted experiments is provided in Table 2. For each setting we perform 100 experiments on the CIFAR-10 validation set. For multiple images in a mini batch we only use images of different labels avoiding permutation ambiguities of reconstructed images of the same label. As to be expected, the single image reconstruction turns out to be most

Table 2: PSNR statistics for various federated averaging settings, averaged over experiments on the first 100 images of the CIFAR-10 validation data set.

| 1 epoch | | | 5 epochs | |
|---|---|---|---|---|
| 4 images | 8 images | | 1 image | 8 images |
| batchsize 2 | batchsize 2 | batchsize 8 | batchsize 1 | batchsize 8 |
| $\mathbf{16.92 \pm 2.10}$ | $\mathbf{14.66 \pm 1.12}$ | $\mathbf{16.49 \pm 1.02}$ | $\mathbf{25.05 \pm 3.28}$ | $\mathbf{16.58 \pm 0.96}$ |

amenable to attacks in terms of PSNRs values. Despite a lower performance in terms of PSNR, we still observe privacy leakage for all multi-image reconstruction tasks, including those in which gradients in random mini-batches are taken. Comparing the full-batch, 8 images examples for 1 and 5 epochs, we see that our previous observation that multiple epochs do not make the reconstruction problem more difficult, extends to multiple images. For a qualitative assessment of reconstructed images of all experimental settings of Table 2, we refer to the supplementary material.

## 7 Conclusions

Federated learning is a modern paradigm shift in distributed computing, yet its benefits to privacy are not as well understood yet. We shed light into possible avenues of attack, analyze the ability to reconstruct the input to any fully connected layer analytically, propose a general optimization-based attack based on cosine similarity of gradients, and discuss its effectiveness for different types of architectures and scenarios. In contrast to previous work we show that even *deep*, *nonsmooth* networks trained with ImageNet-sized data such as modern computer vision architectures like ResNet-152 are vulnerable to attacks - even when considering *trained* parameter vectors. Our experimental results clearly indicate that privacy is not an innate property of collaborative learning algorithms like federated learning, and that secure applications to be closely investigated on a case-by case basis for their potential of leaking private information. Provable differential privacy possibly remains the only way to *guarantee* security, even for aggregated gradients of larger batches of data points.

## Broader Impact - Federated Learning does not guarantee privacy

Recent works on privacy attacks in federated learning setups ([25, 24, 31, 35, 34]) have hinted at the fact that previous hopes that "Privacy is enhanced by the ephemeral and focused nature of the [Federated Learning] updates" [3] are not true in general. In this work, we demonstrated that improved optimization strategies such as a cosine similarity loss and a signed Adam optimizer allow for image recovery in a federated learning setup in industrially realistic settings for computer vision: Opposed to the idealized architectures of previous works we demonstrate that image recovery is possible in deep, non-smooth and trained architectures over multiple federated averaging steps of the optimizer and even in batches of 100 images.

We note that image classification is possibly especially vulnerable to these types of attacks, given the inherent structure of image data, the size of image classification networks, and the comparatively small number of images a single user might own, relative to other personal information. On the other hand, this attack is likely only a first step towards stronger attacks. Therefore, this work points out that the question how to protect the privacy of our data while collaboratively training highly accurate machine learning approaches remains largely unsolved: While differential privacy offers provable guarantees, it also reduces the accuracy of the resulting models significantly [12]. As such differential privacy and secure aggregation can be costly to implement so that there is some economic incentive for data companies to use only basic federated learning. For a more general discussion, see [30]. There is strong interest in further research on privacy preserving learning techniques that render the attacks proposed in this paper ineffective. This might happen via defensive mechanisms or via computable guarantees that allow practitioners to verify whether their specific application is vulnerable to such an attack and within which bounds.

## Acknowledgments and Disclosure of Funding

This research was directly supported by the University of Siegen. HB and MM further received support from the German Research Foundation (DFG) under grant MO 2962/2-1.

## References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv:1802.00420 [cs]*, February 2018.

[2] Martin Benning and Martin Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, May 2018.

[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards Federated Learning at Scale: System Design. *arXiv:1902.01046 [cs, stat]*, March 2019.

[4] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[5] Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input Similarity from the Neural Network Perspective. In *Advances in Neural Information Processing Systems 32*, pages 5342–5351. Curran Associates, Inc., 2019.

[6] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project Adam: Building an Efficient and Scalable Deep Learning Training System. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 571–582, 2014.

[7] Alexey Dosovitskiy and Thomas Brox. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *Advances in Neural Information Processing Systems 29*, pages 658–666. Curran Associates, Inc., 2016.

[8] Alexey Dosovitskiy and Thomas Brox. Inverting Visual Representations With Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4829–4837, 2016.

[9] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1322–1333, Denver, Colorado, USA, October 2015. Association for Computing Machinery.

[10] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633, Toronto Canada, January 2018. ACM.

[11] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. I-RevNet: Deep Invertible Networks. *arXiv:1802.07088 [cs, stat]*, February 2018.

[12] Bargav Jayaraman and David Evans. Evaluating Differentially Private Machine Learning in Practice. *arXiv:1902.08874 [cs, stat]*, August 2019.

[13] Arthur Jochems, Timo M. Deist, Issam El Naqa, Marc Kessler, Chuck Mayo, Jackson Reeves, Shruti Jolly, Martha Matuszak, Randall Ten Haken, Johan van Soest, Cary Oberije, Corinne Faivre-Finn, Gareth Price, Dirk de Ruysscher, Philippe Lambin, and Andre Dekker. Developing and Validating a Survival Prediction Model for NSCLC Patients Through Distributed Learning Across 3 Countries. *International Journal of Radiation Oncology*Biology*Physics*, 99(2):344–352, October 2017.

[14] Arthur Jochems, Timo M. Deist, Johan van Soest, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Philippe Lambin, and Andre Dekker. Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, December 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, May 2015.

[16] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning*, pages 1885–1894, July 2017.

[17] Jakub Konečný, Brendan McMahan, and Daniel Ramage. Federated Optimization:Distributed Optimization Beyond the Datacenter. *arXiv:1511.03575 [cs, math]*, November 2015.

[18] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, August 1989.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083 [cs, stat]*, June 2017.

[20] Aravindh Mahendran and Andrea Vedaldi. Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. *International Journal of Computer Vision*, 120(3):233–255, December 2016.

[21] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv:1602.05629 [cs]*, February 2017.

[22] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. *arXiv:1710.06963 [cs]*, February 2018.

[23] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706, May 2019.

[24] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-Preserving Deep Learning: Revisited and Enhanced. In *Applications and Techniques in Information Security*, Communications in Computer and Information Science, pages 100–110, Singapore, 2017. Springer.

[25] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. Technical Report 715, 2017.

[26] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive Federated Optimization. *arXiv:2003.00295 [cs, math, stat]*, February 2020.

[27] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, November 1992.

[28] Reza Shokri and Vitaly Shmatikov. Privacy-Preserving Deep Learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, pages 1310–1321, Denver, Colorado, USA, 2015. ACM Press.

[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *arXiv:1312.6199 [Cs]*, December 2013.

[30] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: Model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180083, November 2018.

[31] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. *arXiv:1812.00535 [cs]*, December 2018.

[32] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Machine Learning: Concept and Applications. *arXiv:1902.04885 [cs]*, February 2019.

[33] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. *arXiv:1911.07135 [cs, stat]*, November 2019.

[34] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved Deep Leakage from Gradients. *arXiv:2001.02610 [cs, stat]*, January 2020.

[35] Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems 32*, pages 14774–14784. Curran Associates, Inc., 2019.

# 外 文 译 文

## 梯度反转 - 分布式学习中隐私泄露有多容易？

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, Michael Moeller

西根大学

## 第一章 引言

联合学习或协作学习 [6,28] 是一种分布式学习范式，最近由于机器学习中的数据需求和隐私问题持续增加而获得了极大的关注 [21,14,32]。其基本思想是训练一个机器学习模型，例如神经网络，通过使用损失函数 L 和由输入图像 xi 和相应标签 yi 组成的示例训练数据来优化网络的参数 θ，以便求解

$$\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_{\theta}\left(x_i, y_i\right) \qquad \text{式（外 1-1）}$$

我们考虑一个分布式设置，其中服务器希望在拥有训练数据 $(x_i, y_i)$ 的多个用户的帮助下解决 (1)。联合学习的思想是只共享梯度 $\nabla_{\theta}\mathcal{L}_{\theta}\left(x_i, y_i\right)$ 用服务器代替原来的数据 $(x_i, y_i)$，它随后积累来更新总体权值。例如，使用梯度下降，服务器的更新可能构成

$$\theta^{k+1} = \underbrace{\theta^{k} - \tau \sum_{i=1}^{N}}_{\text{server}} \underbrace{\nabla_{\theta}\mathcal{L}_{\theta^{k}}\left(x_i, y_i\right)}_{\text{users}}. \qquad \text{式（外 1-2）}$$

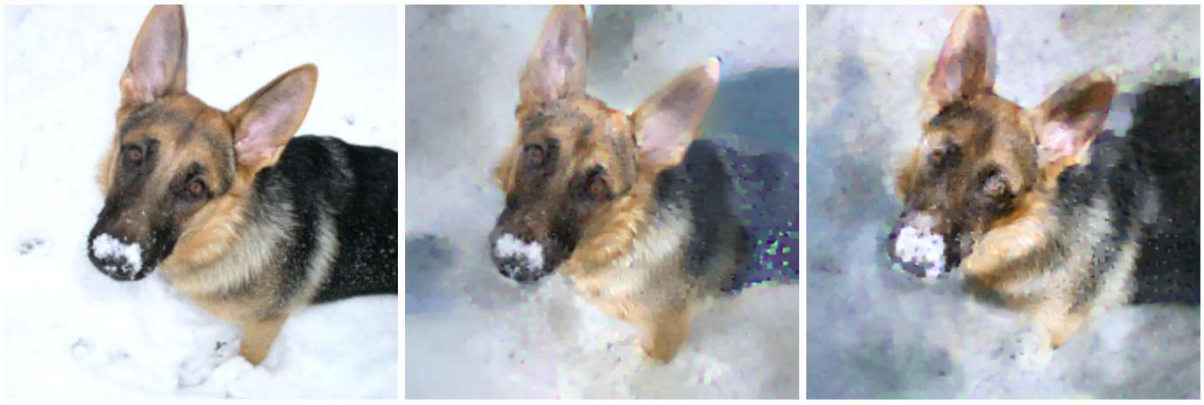


图 外 1-1 从梯度 $\nabla_{\theta}\mathcal{L}_{\theta}(x, y)$ 中重构输入图像 $x$

左图：验证数据集中的图像。中间图：从经过 ImageNet 训练的 ResNet-18 进行重构。右图：从经过 ImageNet 训练的 ResNet-152 进行重构。在两种情况下，图像的隐私都被破坏了。请注意，以前的攻击无法恢复任何 ImageNet 大小的数据 [35] 或攻击训练过的模型。

更新的参数 $\theta^{k+1}$ 被发送回各个用户。方程 (2) 中的过程称为联邦 SGD。相比之下，在联邦平均 [17, 21] 中，每个用户在本地计算多个梯度下降步骤，并将更新后的参数发送回服务器。最后，可以通过仅分享多个本地示例的梯度的均值 $\frac{1}{t}\sum_{i=1}^{t}\nabla_\theta \mathcal{L}_{\theta^k}\left(x_i, y_i\right)$ 来进一步混淆有关 $\left(x_i, y_i\right)$ 的信息，我们将其称为多图像设置。

在需要保护用户隐私的实际应用中，如医院数据 [13] 或移动设备上的文本预测 [3] 中已经使用了这种类型的分布式学习，并且已经声明“随机学习更新的短暂和集中的性质增强了隐私”[3]：认为模型更新包含的信息比原始数据少，并且通过聚合来自多个数据点的更新，认为原始数据不可能恢复。在本研究中，我们在分析和经验上表明，参数梯度仍然携带关于所认为的私有输入数据的显着信息，如图 1 所示。我们最终得出的结论是，即使在现实的架构上进行多图像联邦平均，也不能保证所有用户数据的隐私性，这表明在 100 个图像批处理中，仍有几个是可以恢复的。

**威胁模型**：我们调查一个想要发现用户数据的诚实但好奇的服务器：攻击者被允许分别存储和处理由各个用户传输的更新，但不得干扰协作式学习算法。攻击者不得修改模型体系结构以更好地适应其攻击，也不得发送不代表实际学习的全局参数。用户在第 6 节中被允许在本地积累数据。我们提供了进一步的评论，请参阅补充材料，并提到在攻击者的弱约束条件下，攻击几乎是微不足道的。

在本文中，我们首先在学术环境中讨论联邦学习的隐私限制，重点研究来自一幅图像的梯度反转的情况，并展示以下内容：

- 可以从梯度信息中重建输入数据，对于具有现实深度和非平滑架构的模型，无论模型参数是否训练过。
- 使用正确的攻击方法，深度网络与浅层网络一样脆弱，存在较少的“深度防御”。
- 我们证明，任何全连接层的输入都可以在不考虑其余网络架构的情况下进行解析重构。

然后，我们考虑这些发现对实际情境的影响，发现以下情况：

- 在实践中，可以从它们的平均梯度中重建多个单独的输入图像，经过多个时期，使用本地小批量甚至进行最多 100 个图像的平均梯度。

# 第二章　相关工作

以前与本文相似的研究主要局限于浅层网络，而不是实际相关性更强的深层网络。神经网络中从梯度信息中恢复图像数据的可能性最早在参考文献 [25,24] 中被讨论，他们证明了单个神经元或线性层的恢复是可能的。对于卷积架构，参考文献 [31] 展示了 4 层 CNN 中单个图像的恢复是可能的，尽管需要一个相当大的全连接 (FC) 层。他们的工作首先构建了输入图像的“表示”，然后使用 GAN 进行改进。参考文献 [35] 扩展了这一点，展示了对于一个 4 层 CNN（具有大型 FC 层，平滑的 Sigmoid 激活，没有步幅，统一随机权重），缺失的标签信息也可以一起重建。他们进一步展示，从平均梯度中重建多个图像确实是可能的（对于最大批处理大小为 8）。参考文献 [35] 还讨论了更深的架构，但没有提供具体的结果。后续的参考文献 [34] 指出，可以从最后一层的梯度中解析计算标签信息。这些工作对模型体系结构和模型参数做出了强烈的假设，使得重建变得更容易，但违反了我们在本工作中考虑的威胁模型，并导致较不现实的情况。

[31,35,34] 中讨论的中心恢复机制是欧几里得匹配项的优化。成本函数

$$\arg\min_{x} \left\|\nabla_{\theta}\mathcal{L}_{\theta}(x,y) - \nabla_{\theta}\mathcal{L}_{\theta}\left(x^{*},y\right)\right\|^{2} \qquad \text{式（外 2-1）}$$

被最小化以从传输的梯度 $\nabla_{\theta}\mathcal{L}_{\theta}\left(x^{*},y\right)$ 中恢复原始输入图像 $x^{*}$。这个优化问题是通过 L-BFGS 求解器 [18] 来解决的。请注意，对 $\mathcal{L}$ 的梯度关于 $x$ 的微分需要考虑参数化函数的二阶导数，而 L-BFGS 需要构建一个三阶导数的近似值，这对于具有 ReLU 单元的神经网络来说是具有挑战性的，因为更高阶导数是不连续的。

与完全重建输入图像相比，相关但更容易的问题是从局部更新中检索输入属性 [23,10]，例如，在人脸识别系统中识别的人戴帽子吗？甚至可以从神经网络的深层中恢复与任务无关的属性信息，这些信息可以从局部更新中恢复出来。

此外，我们的问题陈述还与模型反演 [9] 密切相关，在训练后从网络参数中恢复训练图像。如果没有其他信息的情况下，对于更深的神经网络架构，模型反演通常是具有挑战性的 [33,9]。另一个密切相关的任务是从视觉表示中进行反演 [8,7,20]，其中，在给定神经网络的某个中间层的输出的情况下，可以重建出合理的输入图像。这个过程可能会泄露一些信息，例如一般图像构成、主导颜色等，但是根据给定的层，它只会重建出相似的图像，如果神经网络没有被明确选择为（基本上）可逆的，那么这个过程只会重建出相似的图像。正如我们后来证明的那样，从视觉表示中进行反演比从梯度信息中恢复要更困难。

# 第三章　理论分析：从梯度中恢复图像

为了从理论角度理解破坏联邦学习隐私的整体问题，让我们首先分析一个问题：是否可以从其梯度 $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$ 分析地恢复数据 $x \in \mathbb{R}^n$。

由于 $x$ 和 $\nabla_\theta \mathcal{L}_\theta(x, y)$ 维度不同，重建的质量肯定是一个关于参数 $p$ 与输入像素 $n$ 之间的问题。如果 $p < n$，那么重建至少和从不完整数据中恢复图像一样困难 [4,2]。然而，即使当 $p > n$ 的时候，我们在大多数计算机视觉应用中也会遇到这种情况，正则化的 $\nabla_\theta \mathcal{L}_\theta$ “反演”的难度也与梯度算子的非线性和其条件有关。

有趣的是，全连接层在我们的问题中起着特殊的作用：如下文所证明的，无论全连接层在神经网络中的位置如何（前面或后面），全连接层的输入都可以从参数梯度中独立地进行分析计算（假设满足一个防止梯度为零的技术条件）。特别地，分析重建不依赖于先前或之后的特定类型的层，单个全连接网络的输入总是可以被分析地重建，而不需要解决一个优化问题。以下陈述是 [24] 中例 3 的泛化，适用于任意神经网络和任意损失函数的设置：

**命题 3.1** 考虑一个包含有偏置全连接层的神经网络，该全连接层之前仅由（可能是无偏置的）全连接层组成。此外，假设对于任何这些全连接层，损失函数 $\mathcal{L}$ 对于该层的输出的导数至少包含一个非零项。那么可以唯一地从网络的梯度重构出输入。

证明：下面我们对证明进行概述，并参考附加材料中的详细推导。考虑一个无偏置全连接层，将输入 $x_l$ 映射到输出，例如在 ReLU 非线性函数之后：$x_{l+1} = \max(A_l x_l, 0)$，其中 $A_l$ 是一个维度兼容的矩阵。根据假设，对于某个索引 $i$，有 $\frac{\mathrm{dL}}{\mathrm{d}(x_{l+1})i} \neq 0$。然后根据链式法则，可以计算 $x_l$ 为 $\left(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(x_{l+1})i}\right)^{-1} \cdot \left(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(A_l)i,i}\right)^{T}$。这允许在已知 $\mathcal{L}$ 对于某个特定层的输出的导数时，逐步计算出层的输入。我们注意到，添加偏置可以被解释为一个层将 $x_k$ 映射到 $x_{k+1} = x_k + b_k$，且 $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}x_k} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}b_k}$。

根据上述考虑，另一个有趣的方面是许多流行的网络架构使用全连接层（或其级联层）作为它们最后的预测层。因此，作为预测模块输入的上一层的输出可以被重构出来。这些激活通常已经包含了关于输入图像的一些信息，因此使它们容易受到攻击者的攻击。例如，[23] 展示了这些特征表示可以通过训练一个辅助的恶意分类器来挖掘图像属性，该分类器可以识别主任务中不包含的属性。在这方面进一步有趣的是，根据 [34] 的讨论，可以从最后一个全连接层的梯度中重构出地面实况标签信息。最后，命题 3.1 允许得出结论：对于任何以全连接层结束的分类网络，从参数梯度中重构输入比从它们最后的卷积层中反演视觉表示（如 [8, 7, 20] 所讨论的）要容易得多。

# 第四章　一个数值重构方法

由于图像分类网络很少以全连接层开始，让我们转向输入的数值重构：先前的重构算法依赖于两个组成部分：方程（3）的欧几里得成本函数和通过 L-BFGS 进行的优化。我们认为这些选择对于更现实的架构，特别是任意的参数向量来说并不是最优的。如果我们将参数梯度分解为其范数大小和方向两部分，我们发现范数大小只能捕捉关于训练状态的信息，衡量数据点相对于当前模型的局部最优性（对于强凸函数，梯度大小甚至是到最优解的距离的上界）。相反，梯度的高维方向可以携带重要的信息，因为两个数据点之间的角度量化了在一个数据点上采取梯度步进朝向另一个数据点时的预测变化 [5, 16]。因此，我们建议使用基于角度的成本函数，即余弦相似度，$l(x, y) = \langle x, y\rangle / (||x|| ||y|| \,|)$。与方程（3）相比，目标不是找到具有最适应观察到的梯度的图像，而是找到导致模型预测变化与（未观察到的！）真实值相似的图像。如果额外限制两个梯度向量的范数为 1，则这相当于最小化欧几里得成本函数。

我们进一步将我们的搜索空间限制在 $[0, 1]$ 之内，并且只添加总变差 [27] 作为整个问题的简单图像先验，参见 [31]：

$$\arg\min_{x\in[0,1]^n} 1 - \frac{\langle \nabla_\theta \mathcal{L}_\theta(x, y), \nabla_\theta \mathcal{L}_\theta\left(x^*, y\right)\rangle}{\|\nabla_\theta \mathcal{L}_\theta(x, y)\| \, \|\nabla_\theta \mathcal{L}_\theta\left(x^*, y\right)\|} + \alpha \, \mathrm{TV}(x) \qquad \text{式（外 4-1）}$$

其次，我们注意到，通过最小化依赖于中间层输出（通过它们的梯度间接影响）的某个量，来寻找给定区间内的一些输入 $x$，与为神经网络寻找对抗扰动的任务相关 [29, 19, 1]。因此，我们最小化等式：展示了来自 CIFAR-10 验证集的前 6 个图像。

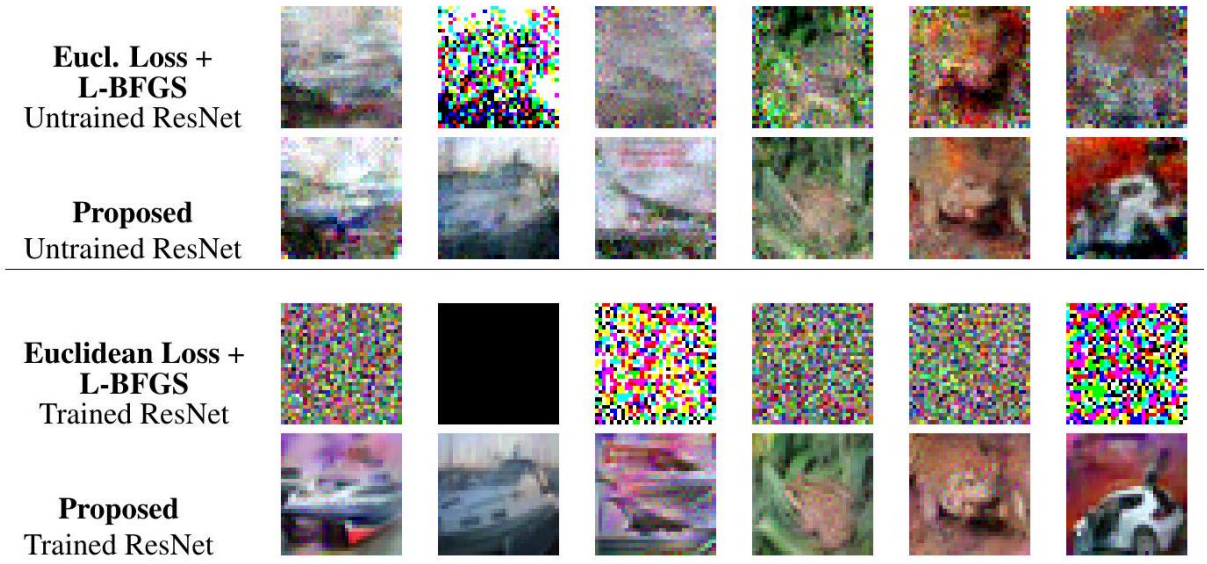



图 外 4-1　基准比较是针对 [31, 35] 中展示的网络架构

（4）仅基于其梯度的符号进行优化，我们使用 Adam [15] 算法进行优化，并使用步长衰减。然而需要注意的是，有符号梯度仅影响 Adam 算法的一阶和二阶动量，实际的更新步骤仍然是无符号的，基于累积的动量进行计算，因此可以准确地恢复图像。

应用这些技术会导致图 1 中所观察到的重构结果。关于所提出机制的进一步消融实验可以在附录中找到。我们在https://github. com/JonasGeiping/invertinggradients 上提供了 PyTorch 的实现。

由于双重反向传播，这种攻击的计算成本大约是在目标方程（4）上每个梯度步骤的单个小批量步骤的两倍。在这项工作中，我们保守地运行攻击，最多进行 24000 次迭代，使用相对较小的步长，因为计算成本不是我们目前的主要关注点（我们假设潜在的攻击者在计算能力方面可能比用户具有数量级更高的能力），但我们注意到，更智能的步长规则和更大的步长可以在几百次迭代的预算下成功攻击。

**备注（优化标签信息）**。虽然我们也可以将标签 y 视为方程（4）中的未知项，并像 [35] 中那样联合优化 $(x, y)$，但我们遵循 [34] 的做法，发现标签信息可以在分类任务中进行解析重构。因此，我们认为标签信息是已知的。

# 第五章　从单个梯度重构单个图像

类似于先前在联邦学习环境中破坏隐私的工作，我们首先关注从梯度 $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$ 中重构单个输入图像 $x \in \mathbb{R}^n$。这种设置既是概念验证，也是我们在第 6 节中考虑的多图像分布式学习设置的重构质量上限。虽然先前的工作已经显示单个图像的隐私破坏是可能的，但它们的实验仅限于相对较浅、光滑和未经训练的网络。接下来，我们将我们提出的方法与先前的工作进行比较，并进行了关于架构和训练相关选择对重构的影响的详细实验。每个实验的超参数设置和更多的视觉结果都在补充材料中提供。

与先前方法的比较。我们首先将我们的方法与使用 L-BFGS 优化的欧几里得损失（3）进行比较，这是在 [31, 35, 34] 中考虑的方法。由于该方法往往由于糟糕的初始化而失败，所以我们允许 L-BFGS 求解器进行 16 次重启。为了进行定量比较，我们使用与 [35] 中相同的浅层光滑 CNN（称为"LeNet (Zhu)"）以及一个 ResNet 架构，分别使用训练和未经训练的参数，测量了 CIFAR-10 验证集的前 100 个图像的 $32 \times 32$ 重构图像的平均峰值信噪比（PSNR）。表格 1 比较了欧几里得损失（3）通过 L-BFGS 优化（与 $[31, 35, 34]$ 中的方法相同）和我们提出的方法的重构质量。前者在未经训练、平滑且浅层的架构上表现非常好，但在经过训练的 ResNet 上完全失败。我们注意到 [31] 使用 GAN 增强了通过 LBFGS 重建的图像质量，但当图像扭曲得无法增强时，该方法失败了。我们的方法提供了可识别的图像，并且在经过训练的 ResNet 上表现特别出色，如图 2 所示。有趣的是，与表 1 中的信噪比较低的未经训练的 ResNet 相比，经过训练的 ResNet 上的重建图像具有更好的视觉质量。让我们在更加现实的情境下研究训练网络参数的效果，即从 ResNet-152 重建 ImageNet 图像。

**经过训练与未经训练的网络。**如果一个网络经过训练并具有足够的容量，使得损失函数 $\mathcal{L}_\theta$ 的梯度在不同输入下为零，那么很明显它们无法与它们的梯度区分开。然而，在实际情况中，由于随机梯度下降、数据增强和有限的训练轮数，图像的梯度很少完全为零。尽管我们观察到经过训练的网络中图像梯度的幅值要比未经训练的网络小得多，但我们的幅值无关方法（式（4））仍然仅基于经过训练的梯度的方向恢复重要的视觉信息。

顶行：真实图像。底行：重构图像。我们检查了 ILSVRC2012 验证集中每 1000 个图像。每个图像泄露的信息量在很大程度上取决于图像内容，尽管像两只鱼类这样的例子受到了很大的破坏，但黑天鹅（具有讽刺意味的是）几乎没有泄露可用的信息。还可以注意到一些图像中位置信息的丢失。

我们在图 3 中的 ImageNet 重建图像中观察到了两个对经过训练的网络的一般性影响：首先，重建似乎在隐含地偏向于训练数据中同一类别的典型特征，例如第 5 张图中

表 外 5-1　在 CIFAR-10 验证数据集的前 100 个图像上进行的 100 次实验的 PSNR 均值和标准差

| Architecture | LeNet (Zhu) | | ResNet20-4 | |
|---|---|---|---|---|
| Trained | False | True | False | True |
| Eucl. Loss + L-BFGS | **46.25 ± 12.66** | 13.24 ± 5.44 | 10.29 ± 5.38 | 6.90 ± 2.80 |
| Proposed | 18.00 ± 3.33 | **18.08 ± 4.27** | **19.83 ± 2.96** | **13.95 ± 3.38** |

图 外 5-1 使用经过训练的 ResNet-152 的参数梯度进行单个图像重构

黑琵鸟更蓝色的羽毛，或者插图中猫头鹰的大眼睛。因此，尽管大多数图像的整体隐私性显然受到侵犯，但这种效应至少阻碍了细节或图像背景的恢复。其次，我们发现在神经网络训练过程中使用的数据增强导致经过训练的网络使对象的定位更加困难：请注意图 3 中有多少对象保留了其原始位置，以及蛇和壁虎的重复。因此，尽管使用数据增强训练的图像重建仍然成功，但某些位置信息会丢失。

图 外 5-2

**平移不变卷积。**让我们通过测试传统的使用零填充卷积的卷积神经网络与使用循环填充卷积的可证明平移不变 CNN 之间的能力，来更详细地研究模糊物体位置的能力。如插图所示，尽管传统 CNN 能够恢复出质量相当高的图像（左侧），但平移不变网络使得物体的定位变得不可能（右侧），因为原始物体被分离开来。因此，我们确定了常见的零填充是隐私风险的一个来源。

PSNR 值是指显示的图像，而平均 PSNR 是在前 10 个 CIFAR-10 图像上计算的。标准差是给定架构下一个实验的平均标准差。ResNet-18 架构显示了三个不同宽度的情况。

**网络深度和宽度。**对于分类准确性来说，CNN 的每一层的深度和通道数都是非常重要的参数，这就是为什么我们研究它们对我们的重建结果的影响。图 4 显示，随着通

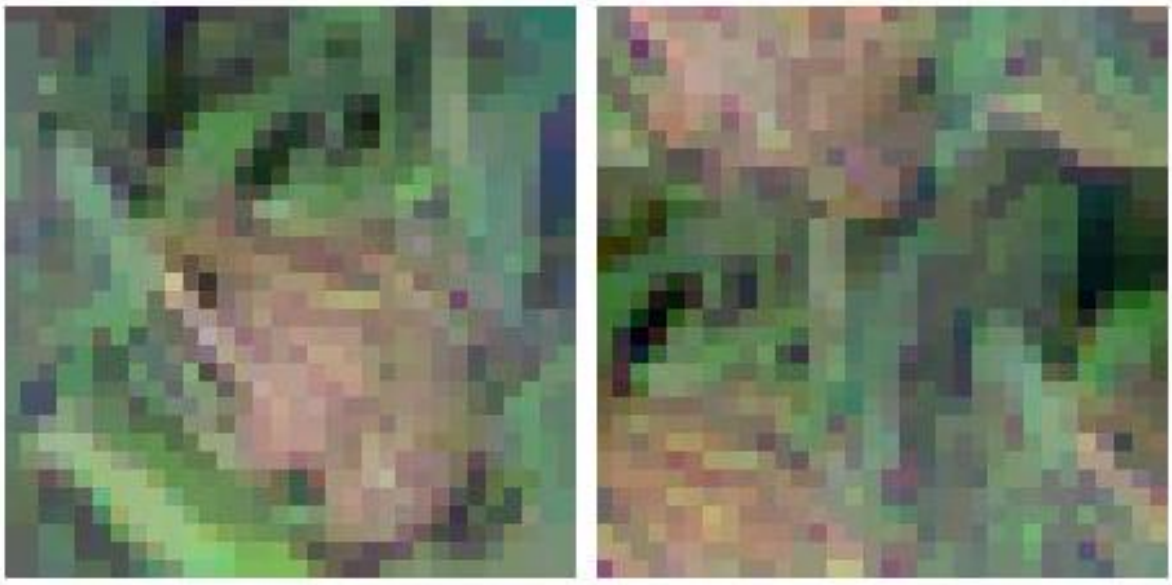


图 外 5-3

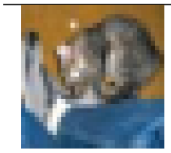
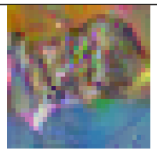
| **Original** | **ResNet-18** with base width: | | | **ResNet-34** | **ResNet-50** |
|---|---|---|---|---|---|
| | 16 | 64 | 128 | | |
|  |  |  |  |  |  |
| PSNR | 17.24 | 17.37 | 25.25 | 18.62 | 21.36 |
| Avg. PSNR | 19.02 | 22.04 | 22.94 | 21.59 | 20.98 |
| Std. | 2.84 | 5.89 | 6.83 | 4.49 | 5.57 |

图 外 5-4 多个 ResNet 架构的原始图像重建（左侧）

道数的增加，重建质量显著提高。然而，更大的网络宽度也伴随着实验成功的方差增加。然而，通过多次重启实验，可以为更宽的网络产生更好的重建结果，使得 PSNR 值从 16 个通道增加到 128 个通道时从 19 增加到接近 23。因此，更大的网络宽度增加了攻击者的计算工作量，但并不提供更高的安全性。

观察我们从具有不同深度的 ResNet 获得的重建结果，可以看出随着网络深度的增加，所提出的攻击几乎没有受到影响。特别是-如图 3 所示，即使通过 ResNet-152，我们也能够实现对 ImageNet 的忠实重建。

# 第六章　使用联邦平均和多个图像的分布式学习

到目前为止，我们只考虑了从梯度中恢复单个图像的情况，并讨论了这种情境下的限制和可能性。现在，我们将转向更困难的泛化情景，即联邦平均 [21,22,26] 和多图像重建，以展示所提出的改进在这种更实际的情况下同样有效，并讨论该应用中的可能性和限制。

联邦平均不仅仅基于本地数据计算网络参数的梯度，而是在将更新后的参数发送回服务器之前，在本地数据上执行多个更新步骤。按照 [21] 的符号表示，我们假设用户端的本地数据包含 $n$ 个图像。在每个本地时期中，用户执行 $\frac{n}{B}$ 个随机梯度更新步骤，其中 $B$ 表示本地小批量大小，结果是总共进行了 $E\frac{n}{B}$ 个本地更新步骤。然后，每个用户 $i$ 将本地更新的参数 $\tilde{\theta}_i^{k+1}$ 发送回服务器，服务器通过对所有用户进行平均来更新全局参数 $\theta^{k+1}$。

我们通过实验证明，即使在联邦平均的情况下，$n \geq 1$ 个图像的设置也可能受到攻击。为此，我们尝试通过本地更新 $\tilde{\theta}_i^{k+1} - \theta^k$ 的知识来重建 $n$ 个图像的本地批处理。接下来，我们将评估不同选择的 $n, E$ 和 $B$ 的重建图像质量。我们注意到，之前章节中研究的情况对应于 $n = 1, E = 1, B = 1$。在所有实验中，我们使用了一个未经训练的 ConvNet。

**多个梯度下降步骤，** $B = n = 1, E > 1$：

图 5 显示了在不同的本地时期 $E$ 和不同的学习率 $\tau$ 选择下，对 $n = 1$ 个图像的重建情况。即使在 100 个本地梯度下降步骤的情况下，重建质量也没有受到影响。我们唯一能够举例说明的失败案例是选择了较高的学习率 1e-1。然而，这种设置对应于一个会导致发散训练更新的步长，因此不能提供有用的模型更新。

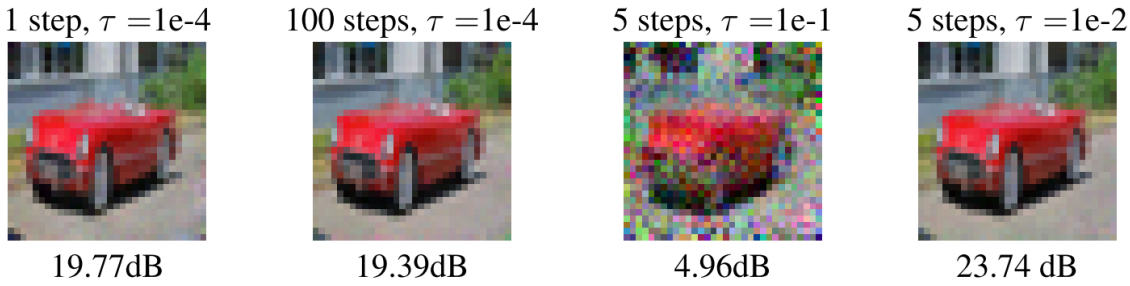



图 外 6-1　本地更新步数和学习率对重建结果的影响

左侧两幅图像比较了在固定学习率 τ = 1e-4 的情况下梯度下降步数的影响。右侧两幅图像则是在固定的 5 个梯度下降步数下变化学习率的结果。图像下方显示了 PSNR 值。

对于 ResNet32-10。显示了整个批次中最具识别性的 5 个图像。尽管大多数图像无法识别，但在大批次设置中仍然破坏了隐私。有关所有图像，请参阅补充材料。

**多图像恢复，** $B = n > 1, E = 1$：

到目前为止，我们只考虑了单个图像的恢复，并且有理由相信在将更新发送到服务器之前，对多个（本地）图像的梯度进行平均可以恢复联邦学习的隐私性。虽然在 [35]

图 外 6-2 在 CIFAR-100 上使用一个包含 100 个图像的批次的聚合梯度中的信息泄漏

表 外 6-1 在 CIFAR-10 验证数据集的前 100 张图像上进行的实验中，对于不同的联邦平均设置，计算了 PSNR（峰值信噪比）统计数据

| | 1 epoch | | 5 epochs | |
|---|---|---|---|---|
| 4 images | 8 images | | 1 image | 8 images |
| batchsize 2 | batchsize 2 | batchsize 8 | batchsize 1 | batchsize 8 |
| **16.92 ± 2.10** | **14.66 ± 1.12** | **16.49 ± 1.02** | **25.05 ± 3.28** | **16.58 ± 0.96** |

中已经考虑了多图像恢复的情况，但我们证明了所提出的方法能够从 100 个平均梯度的批次中恢复一些信息：尽管大多数恢复的图像无法识别（如补充材料所示），图 6 显示了最具识别性的 5 个图像，并说明即使对 100 个图像的梯度进行平均，也无法完全保护隐私数据。最令人惊讶的是，批处理产生的失真是非均匀的。人们可能会期望所有图像都受到相等的失真，几乎无法恢复，然而一些图像被严重扭曲，而其他图像仅在可以轻松识别图中对象的程度上扭曲，这表明即使对于大批量的图像数据，隐私泄漏也是可以想象的。

请注意，在这种情况下，攻击者只知道梯度的平均值，但我们假设服务器知道参与的图像数量。服务器可能无论如何都会请求这些信息（例如为了平衡异构数据），但即使不知道图像的确切数量，服务器（我们假设服务器具有比用户更多的计算能力）也可以在一系列候选图像数量上运行重建算法，因为图像数量只是一个小整数值，然后选择重建损失最小的解决方案。

**一般情况**

我们还考虑了每个小批量梯度步骤中使用整个本地数据子集进行多个本地更新步骤的一般情况。表 2 提供了所有进行的实验的概述。对于每个设置，我们在 CIFAR-10 验证集上进行了 100 次实验。对于小批量中的多个图像，我们只使用不同标签的图像，以避免对相同标签的重建图像进行置换模糊。正如预期的那样，单个图像的重建在 PSNR 值方面最容易受到攻击。尽管在 PSNR 方面性能较低，但我们仍然观察到所有多图像重建任务的隐私泄漏，包括在随机小批量中采用的梯度。比较表 2 中 1 个和 5 个时期的完整批处理中的 8 个图像示例，我们发现我们之前观察到的多个时期不会使重建问题更加

困难的观察结果也适用于多图像。关于表 2 中所有实验设置的重建图像的定性评估，请参阅补充材料。

# 第七章　结论

联邦学习是分布式计算中的一种现代范式转变，然而其对隐私的好处尚未被充分理解。我们揭示了可能的攻击途径，通过分析能够对任何全连接层进行输入重构的能力，提出了一种基于梯度余弦相似度的通用优化攻击，并讨论了其在不同类型的架构和场景中的有效性。与先前的工作相比，我们展示了即使是使用 ImageNet 规模的数据进行训练的深层非平滑网络（如 ResNet-152）在攻击下也是脆弱的，即使考虑到训练后的参数向量。我们的实验结果明确表明，隐私并不是联邦学习等协作学习算法的固有属性，而安全应用需要针对各种情况进行深入研究，以了解其泄露私人信息的潜力。可证明的差分隐私可能仍然是唯一能够保证安全性的方法，即使对于更大批次的数据点的聚合梯度也是如此。

## 7.1　更广泛的影响-联邦学习无法保证隐私

关于联邦学习设置中的隐私攻击的最近研究（[25, 24, 31, 35, 34]）暗示了先前的希望“通过瞬时和集中的 [Federated Learning] 更新提高隐私性”的观点并不总是成立的。在这项工作中，我们证明了在工业实际情况下的计算机视觉中，改进的优化策略（如余弦相似度损失和带符号的 Adam 优化器）可以实现联邦学习设置下的图像恢复：与先前工作的理想化架构相反，我们证明了在深层非平滑和经过训练的架构中，在优化器的多次联邦平均步骤和甚至 100 张图像的批次中，图像恢复是可能的。

我们注意到，图像分类可能特别容易受到此类攻击的影响，这是由于图像数据的固有结构、图像分类网络的规模以及单个用户可能拥有的图像数量相对于其他个人信息来说相对较小。另一方面，这种攻击可能只是对更强攻击的第一步。因此，这项工作指出，在协作训练高度准确的机器学习方法的过程中，如何在保护数据隐私方面仍然存在着广泛的问题：虽然差分隐私提供了可证明的保证，但它也显著降低了最终模型的准确性[12]。因此，实施差分隐私和安全聚合可能是昂贵的，这使得数据公司有一定的经济激励只使用基本的联邦学习方法。关于这个更一般的讨论，请参阅 [30]。目前对于保护隐私的学习技术的研究兴趣浓厚，旨在使本文提出的攻击失效。这可能通过防御机制或计算可验证性的保证来实现，让从业者能够验证其特定应用程序是否容易受到此类攻击，并了解攻击范围。

这项研究的广泛影响是，它提醒我们在协作训练精确的机器学习模型的过程中如何保护数据隐私仍然面临着重要的挑战。随着联邦学习的广泛应用，我们必须深入研究和开发隐私保护技术，以确保用户数据在联邦学习过程中得到充分的保护。这将涉及到制定更强大的防御机制、改进差分隐私技术以提供更好的平衡点，以及建立适用于各种应用场景的可验证性保证。同时，政策制定者和相关利益相关者也应意识到隐私保护在联邦学习中的重要性，并制定相应的规范和法律框架来确保用户数据的安全和隐私。

## 7.2 致谢与资金披露

这项研究直接得到了 Siegen 大学的支持。HB 和 MM 还受到德国研究基金会（DFG）MO 2962/2-1 号资助。

# 北 京 邮 电 大 学

# 本科毕业设计（论文）开题报告

| | | | | | |
|---|---|---|---|---|---|
| 学院 | 网络空间安全学院 | 专业 | 网络空间安全专业 | 班级 | 2019211806 |
| 学生姓名 | 卢亭松 | 学号 | 2019212443 | 班内序号 | 11 |
| 指导教师姓名 | 陆月明 | 所在单位 | 网络空间安全学院 | 职称 | 教授 |
| 设计（论文）题目 | （中文）一种基于 FATE 框架的联邦学习方法设计与实现 | | | | |
| | （英文）Design and Implementation of a Federated-Learning Method Based on FATE Framework | | | | |

毕业设计（论文）开题报告内容：

**一、选题背景与意义**

从 1955 年达特茅斯会议开始，人工智能经过两起两落的发展，迎来了第三个高峰期。越来越多的工程与科研实践让我们看到了人工智能迸发出的巨大潜力，也更加憧憬人工智能技术可以在自动驾驶、医疗、金融等更多、更复杂、更前沿的领域施展拳脚。但是，真实的情况却让人失望：除了有限的几个行业，更多领域存在着数据有限且质量较差的问题，不足以支撑人工智能技术的实现；并且，在某些领域，即使动用很多人力来进行数据标注，数据量也依然不够，这是我们面临的现实。

与此同时，数据源之间存在着难以打破的壁垒，在大多数行业中，数据是以孤岛的形式存在的，由于行业竞争、隐私安全、行政手续复杂等问题，即使是在同一个公司的不同部门之间实现数据整合也面临着重重阻力；另一方面，随着大数据的进一步发展，重视数据隐私和安全已经成为了世界性的趋势，新的隐私保护法规的建立在不同程度上对人工智能传统的数据处理模式也提出了新的挑战。因此，想要将分散在各地、各个机构的数据进行整合是十分困难且成本巨大的。

如何在满足数据隐私、安全和监管要求的前提下，设计一个机器学习框架，让人工智能系统能够更加高效、准确地共同使用各自的数据，则是联邦学习希望解决的问题。联邦学习作为未来 AI 发展的底层技术，依靠安全可信的数据保护措施连接数据孤岛的模式，将在保障隐私信息及数据安全的前提下加速人工智能技术的创新发展、促进全社会智能化水平提升，十分具有研究的意义。

**二、研究内容和拟解决的主要问题**

**2.1 研究的基本内容**

为了准确地了解对比现有联邦学习架构与传统 AI 架构在性能、准确度等方面的差异，研究将基于开源项目 FATE (Federated Al Technology Enabler)，搭建 FATE 联邦学习集群，在联邦场景实现两

种以上主流机器学习场景（计算机视觉、自然语言处理等）的样例算法，与传统机器学习进行对比，从性能（通信损耗时间、计算损耗时间等）、准确率（准确率、召回率、F1-score 等）、安全性测试（投毒攻击测试、对抗攻击测试等）等层面进行分析。

**研究内容 1：**联邦学习的定义、规范与价值机制；联邦学习在不同数据场景下的联邦方式；现有的联邦学习开源实现；联邦学习面临的安全问题以及针对性攻击手段。

联邦学习的定义为：在进行机器学习的过程中，各参与方可借助其他方数据进行联合建模。各方无需共享数据资源，即数据不出本地的情况下，进行数据联合训练，建立共享的机器学习模型。

联邦学习系统需要保证：|联邦学习模型的效果-传统方法模型的效果|<有界正数。

联邦学习的价值机制：联邦学习技术基于“合作共赢”的价值机制，对于商业利益而言极具价值。在这样一个联邦机制下，各个参与者的身份和地位相同，而联邦系统帮助大家建立了“共同富裕”的策略，能够带动跨领域的企业级数据合作、催生基于联合建模的新业态和模式、降低技术提升成本和促进创新技术发展。

联邦学习在不同场景下的联邦方式：

1. 横向联邦学习：在两个数据集的用户特征重叠较多而用户重叠较少的情况下，将数据集按照横向（即用户维度）切分，并取出双方用户特征相同而用户不完全相同的那部分数据进行训练。这种方法叫做横向联邦学习。
2. 纵向联邦学习：在两个数据集的用户重叠较多而用户特征重叠较少的情况下，把数据集按照纵向（即特征维度）切分，并取出双方相同而用户特征不完全相同的那部分数据进行训练。这种方法叫做纵向联邦学习。
3. 联邦迁移学习：在两个数据集的用户与用户特征重叠都较少的情况下，不对数据进行切分，而可以利用迁移学习来克服数据或标签不足的情况。这种方法叫做联邦迁移学习。

现有的联邦学习开源实现：目前业界中主要的联邦学习框架有 FATE、TensorFlow Federated、PaddleFL、Pysyft 等。

联邦学习面临的安全威胁以及其攻击手段：

1. 投毒攻击：投毒攻击主要是指在训练或再训练过程中，恶意的参与者通过攻击训练数据集来操纵机器学习模型的预测。联邦学习中，攻击者有两种方式进行投毒攻击：数据投毒和模型投毒。数据投毒是指攻击者通过对训练集中的样本进行污染，如添加错误的标签或有偏差的数据，降低数据的质量，从而影响最后训练出来的模型，破坏其可用性或完整性；而模型投毒不同于数据投毒，攻击者不直接对训练数据进行操作，而是发送错误的参数或损坏的模型

来破坏全局聚合期间的学习过程，比如控制某些参与方 $U_i$ 传给服务器的更新参数 $\delta_i$，从而影响整个学习模型参数的变化方向，减慢模型的收敛速度，甚至破坏整体模型的正确性，严重影响模型的性能。

2. 对抗攻击：对抗攻击是指恶意构造输入样本，导致模型以高置信度输出错误结果。从攻击环境来说，对抗攻击可以分为黑盒攻击和白盒攻击。若知道机器学习模型中的参数与内部结构，攻击者可以把所需的干扰看作一个优化问题计算出来。这种情况下的对抗攻击属于白盒攻击。而另一种常见的情境下，攻击者不知道任何模型的信息，只能跟模型互动，给模型提供输入然后观察它的输出，这种情形下的对抗攻击属于黑盒攻击。对抗攻击还可以根据攻击目的分为目标攻击和非目标攻击。根据干扰的强度大小分为无穷范数攻击、二范数攻击和零范数攻击等。对抗攻击可以帮助恶意软件逃避检测，生成投毒样本，已经被攻击者广泛应用于图像分类、语义分割、机器识别以及图结构等多个领域，成为系统破坏者的一个有力攻击武器。
3. 隐私泄露：联邦学习方式允许参与方在本地进行数据训练，各参与方之间是独立进行的，其他实体无法直接获取本地数据，可以保证一定的隐私安全，但这种安全并不是绝对安全，仍存在隐私泄露的风险。比如恶意的参与方可以从共享的参数中推理出其他参与方的敏感信息。恶意的服务器可以识别更新的参数的来源，甚至进一步通过参与方多次反馈的参数推测参与方的数据集信息，这可能造成参与方的隐私泄露。

支撑指标点：☑2.3 ☐3.3 ☐4.1 ☐4.2 ☐4.3 ☐10.1 ☐10.2 ☐10.3 ☑12.1 ☑12.2

毕业要求指标点 2.3：针对已建立的网络空间安全领域复杂工程问题的抽象模型，通过文献检索与资料查询获取相关知识，分析论证模型的合理性，获得有效结论。

毕业要求指标点 12.1：能够认识不断探索和学习的必要性，具有自主学习和掌握自主学习的方法，具有拓展与更新知识的能力。

毕业要求指标点 12.2：具有终身学习的知识基础和意识，能够针对个人或职业发展需要，采用合适的方法，自主学习，适应社会发展。

**研究内容 2：**在计算机视觉领域选择基于 CNN 的 mnist 手写数字识别作为样例算法；在自然语言处理领域选择基于 CNN 的中文主题分类作为样例算法；对其分别进行传统机器学习实现与联邦机器学习实现。

基于 CNN 的 mnist 手写数字识别：MNIST 数据集是代表标准和技术数据集的改良研究所的缩

写，是一个包含 60,000 张 0 到 9 之间的手写单个数字的 60,000 个小正方形 28×28 像素灰度图像的数据集。任务是将给定的手写数字图像分类为 10 个类别之一，代表从 0 到 9 的整数值，包括 0 到 9。

基于 CNN 的中文情感分析：使用数据集为清华 NLP 组提供的 THUCNews 新闻文本分类数据集。其中包含体育, 财经, 房产, 家居, 教育, 科技, 时尚, 时政, 游戏, 娱乐 10 个分类。数据格式为带有标注的文本串。

支撑指标点：☐2.3 ☑3.3 ☑4.1 ☑4.2 ☐4.3 ☑10.1 ☐10.2 ☐10.3 ☐12.1 ☐12.2

毕业要求指标点 3.3：综合考虑各种工程因素，给出解决方案，能够利用软件模块，进行网络空间安全领域系统的整体设计与开发。

毕业要求指标点 4.1：能够针对网络空间安全领域的复杂工程问题明确其研究目标，根据目标研究确定需要的实验数据及技术路线，完成实验方案的设计。

毕业要求指标点 4.2：能够选择合适的技术手段，构建实验系统，安全地开展实验，正确采集、整理实验数据。

毕业要求指标点 10.1：具有良好的表达能力，能够就专业问题进行清晰的书面和口头表达，并能与同行进行有效沟通和交流。

**研究内容 3：**与传统机器学习进行对比，从准确率（准确率、召回率、F1-score 等）、性能（通信损耗时间、计算损耗时间等）、安全性测试（投毒攻击测试、对抗攻击测试等）等层面进行分析。

准确率：依据准确率、召回率等标准对传统机器学习实现与联邦机器学习实现分别进行分析与评估，从多个角度综合量化考察联邦学习与传统机器学习在准确率方面的差异，研究改进和调整办法。为了排除检测数据集不平衡性的干扰，评价异常检测主要使用以下指标：准确率、召回率、f1-score。

性能、安全性：收集不同场景、不同算法下联邦学习在性能、安全性等方面的损耗。统计联邦学习的通信时间损耗、同数据量下收敛速度对比；对比投毒攻击、对抗攻击、隐私泄露等场景下联邦学习与传统机器学习的鲁棒性。

支撑指标点：☐2.3 ☐3.3 ☐4.1 ☐4.2 ☑4.3 ☐10.1 ☑10.2 ☑10.3 ☐12.1 ☐12.2

毕业要求指标点 4.3：能够对实验结果进行分析和解释，通过信息综合得到合理有效的结论。

毕业要求指标点 10.2：熟练掌握一门外语，具备一定的国际视野，能够在跨文化背景下进行沟通和交流。

毕业要求指标点 10.3：能够就网络空间安全领域复杂工程问题与业界同行及社会公众进行有效沟通和交流，撰写报告和设计文稿、陈述发言等。

**2.2 拟解决的问题**

由于联邦学习引入了安全多方计算、同态加密等技术，必然也会引入一定的代价，例如通信、计算等性能上的损耗、不同场景下数据异构对准确率的影响等。目前关于这些代价仍没有较为细致的对比数据，而准确地了解对比现有联邦学习架构与传统 AI 架构在性能、准确度、安全性等方面的差异，能够帮助我们进一步优化改进联邦学习。

**三、研究方法及措施**

1. 学习机器学习、隐私保护相关的知识，对研究背景进行调研。

2. 基于 FATE 搭建联邦学习集群实验环境，同时搭建传统机器学习场景，并进行编程实验。

3. 通过控制变量对比的方法分析二者在性能、准确度、安全性等方面的差异，找到现有联邦学习需要优化的方向

4. 通过阅读源码、理解联邦学习实现架构等方式找到优化点并给出不同使用场景下、不同算法下的使用建议

**四、研究工作的步骤与进度：**

秋季学期 17-18 周：学习机器学习、隐私保护相关的知识，对研究背景进行调研。查找并阅读联邦学习、机器学习隐私保护的相关论文，为后续的实验以及论文撰写打下基础。

春季学期 1-2 周：开始 FATE 集群的搭建，完成开题报告的撰写。

春季学期 3-4 周：构建论文框架，完成 FATE 集群搭建。

春季学期 5-6 周：完成论文前两章的撰写，确定两种以上的机器学习场景及代表性问题，设计相应的联邦机器学习算法。

春季学期 7 周：总结上一阶段的工作，完成中期检查。

春季学期 8-9 周：完成两种以上的联邦机器学习算法的传统实现与联邦环境实现，能够在 FATE 集群上进行训练。

春季学期 10-11 周：在 FATE 平台上完成预测与评估，对比分析算法在集中式环境与联邦环境的差异，从准确率，安全性，通信效率等进行分析。

春季学期 12-13 周：根据上述的实验结果，完成论文的撰写。

春季学期 14 周：完成论文，整理本毕设课题的全部成果。

**五、主要参考文献：**

[1] 谭作文, 张连福. 机器学习隐私保护研究综述. 软件学报, 2020, 31(7): 2127-2156.

[2] Abreha HG, Hayajneh M, Serhani MA. Federated Learning in Edge Computing: A Systematic Survey.

Sensors. 2022; 22(2):450. https://doi.org/10.3390/s22020450

[3] K. M. Ahmed, A. Imteaj and M. H. Amini, “Federated Deep Learning for Heterogeneous Edge Computing,“ 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146-1152, doi: 10.1109/ICMLA52953.2021.00187.

[4] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato and S. Zhang, “Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach,“ in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7751-7763, Aug. 2020, doi: 10.1109/JIOT.2020.2991401.

[5] Liu JC, Goetz J, Sen S, Tewari A. Learning From Others Without Sacrificing Privacy: Simulation Comparing Centralized and Federated Machine Learning on Mobile Health Data. JMIR Mhealth Uhealth, 2021;9(3):e23728

[6] 胡健龙. 联邦学习在车联网数据共享与保护技术中的研究[D]. 电子科技大学,2022.DOI:10.27005/d.cnki.gdzku.2022.004716.

[7] M. Nasr, R. Shokri and A. Houmansadr, Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In Proceedings of 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 739-753, doi: 10.1109/SP.2019.00065.

[8] 王慧超. 机器学习中的数据隐私保护研究[D]. 中国科学技术大学,2021.DOI:10.27517/d.cnki.gzkju.2021.001722.

[9]Wang, X.; Wang, J.; Ma, X.; Wen, C. A Differential Privacy Strategy Based on Local Features of Non-Gaussian Noise in Federated Learning. Sensors 2022,22(2424). https://doi.org/10.3390/s22072424

[10] 师兆森. 联邦学习中的隐私保护技术研究. 电子科技大学,2022. DOI:10.27005/d.cnki.gdzku.2022.000987.

[11] 王健宗. 联邦学习算法综述. 大数据, 2020, 6(6): 2020055-1- doi: 10.11959/j.issn.2096-0271.2020055.

[12] 邱鑫源,叶泽聪,崔翛龙,高志强. 联邦学习通信开销研究综述. 计算机应用, 2022, DOI: 10.11772/j.issn.1001-9081.2021020232

[13] 周俊,方国英,吴楠. 联邦学习安全与隐私保护研究综述. 西华大学学报, 2020, doi：10.12198/j.issn.1673－159X.3607

[14] 周传鑫,孙奕,汪德刚,葛桦玮. 联邦学习研究综述. 2021. DOI: 10.11959/j.issn.2096−109x.2021056

| | | |
|---|---|---|
| 允许进入论文撰写环节：是 □ 否 □ | 指导教师 | |

| | | | |
|---|---|---|---|
| 日期 | 年　月　日 | 签字 | |

# 北 京 邮 电 大 学

# 2023 届本科毕业设计（论文）中期进展情况检查表

| | | | | | |
|---|---|---|---|---|---|
| 学院 | 网络空间安全学院 | | 专业 | 网络空间安全专业 | |
| 学生姓名 | 卢亭松 | 学号 | 2019212443 | 班级 | 2019211806 |
| 指导教师姓名 | 陆月明 | 所在单位 | 网络空间安全学院 | 职称 | 教授 |
| 设计（论文）题目 | （中文）一种基于 FATE 框架的联邦学习方法设计与实现 | | | | |
| | （英文）Design and Implementation of a Federated-Learning Method Based on FATE Framework | | | | |

| | |
|---|---|
| 目前已完成任务 | **主要内容：**<br>1. 学习机器学习、隐私保护相关的知识，对研究背景进行调研<br>2. 基于 FATE 搭建联邦学习集群实验环境<br>3. 基于 DSL 和 Pipeline 配置任务<br>4. 完成两种以上的联邦机器学习算法的传统实现与联邦环境实现，能够在 FATE 集群上进行训练<br>**详细设计：**<br>**1. 学习机器学习、隐私保护相关的知识，对研究背景进行调研**<br>联邦学习的定义为：在进行机器学习的过程中，各参与方可借助其他方数据进行联合建模。各方无需共享数据资源，即数据不出本地的情况下，进行数据联合训练，建立共享的机器学习模型。<br>联邦学习系统需要保证：\|联邦学习模型的效果-传统方法模型的效果\|<有界正数。<br>联邦学习的价值机制：联邦学习技术基于“合作共赢”的价值机制，对于商业利益而言极具价值。在这样一个联邦机制下，各个参与者的身份和地位相同，而联邦系统帮助大家建立了“共同富裕”的策略，能够带动跨领域的企业级数据合作、催生基于联合建模的新业态和模式、降低技术提升成本和促进创新技术发展。<br>联邦学习在不同场景下的联邦方式：横向联邦学习、纵向联邦学习、联邦迁移学习<br>**2. 基于 FATE 搭建联邦学习集群实验环境**<br>**a. 在 CentOS7 主机 x2、Docker 20.10.21 环境下完成 Fate 环境的部署**<br>**b. 基于 DSL 配置初始任务验证 Fate 联邦学习流程（上传数据、构建模型、提交任务、部署模型、加载模型、测试模型）**<br>• 进入 Host client 容器修改 examples/upload_host.json 并上传 host 数据<br>• 构建模型：修改 examples/job_conf.json 配置各个节点使用的算法组件及参数；修改 examples/job_dsl.json 配置算法组件间的输入输出流及接口，根据此拼接得到完整的工作模型 |

- 使用 fate-flow 提交任务并部署模型，并使用 POST 方式测试模型效果

c. **基于 DSL 配置初始任务验证 Fate 联邦学习流程（上传数据、构建模型、提交任务、部署模型、加载模型、测试模型）**

- 为 pipeline 配置关联的 FATE Flow Service 并配置相关 Python 环境
- 使用 Pipeline 上传数据：生成 Pipeline 实例并定义数据存储分区、表名和命名空间，之后使用 Pipeline 添加要上传的数据并执行数据上传
- 使用 Pipeline 完成 secureboost 训练：首先定义模型前工作流如下，Reader 组件加载数据； DataTransform 组件解析原始数据到数据实例中； Intersection 组件以计算联邦场景 PSI。然后定义 HeteroSecureBoost 组件创建模型结构，并定义 Evaluation 组件显示评估结果。最终添加组件到 pipeline 构建模型并编译
- 保存训练模型、加载并部署训练模型，执行预测任务验证模型效果

3. **完成两种以上的联邦机器学习算法的传统实现与联邦环境实现，能够在 FATE 集群上进行训练**

a. **基于 Pipeline 实现横向联邦 LSTM 完成 IMDB 文本情感分类，完成 LSTM 模型结构在联邦学习上的实现**

- 下载得到 IMDB 数据集，使用 tokenizer 将数据中的每个单词转化为整数值的唯一映射，词空间的最大个数不超过 10000，出现频率度低的词会被过滤；每个句子长度固定为 200，超过的部分将被截取，较短的句子将用 0 补齐
- 将数据集按联邦节点划分并输出到 csv 文件并基于 Pipeline 上传 IMDB 数据
- 基于 Pipeline 构建 LSTM 模型。构建 LSTM 模型词嵌入层设置 128 个神经元，LSTM 层包含 64 个神经元；最大迭代次数为 100，

| | |
|---|---|
| | batch 长度为 32，设置 early_stop 机制。使用 Adam 进行梯度更新，学习率为 1e-5，损失函数为二元交叉熵<br>• 将定义好的 reader、data_transform、homo_nn、evaluation 结构通过 Pipeline 搭建组件结构并发布<br>• 保存刚才的算法模型组件 homo_nn_0，使用新的 Reader 读入测试数据集，搭建组件发布预测任务并验证效果<br>**b. 基于 Pipeline 实现横向联邦 CNN 完成中文文本主题分类**<br>• 下载得到 THUCNews 数据集，读取词汇表并将每个值都转化为 unicode，通过 unicode 表将文件转换为 id 表示，同时使用 keras 提供的 pad_sequences 来将文本 pad 为固定长度<br>• 将数据集按联邦节点划分并输出到 csv 文件并基于 Pipeline 上传 IMDB 数据<br>• 基于 Pipeline 构建 CNN 模型。构建嵌入层 128 个神经元，CNN 层 32 个神经元，学习率为 1e-4，损失函数为二元交叉熵。其中 label_encoder 使用了 Fate 自带模块<br>• 将定义好的 reader、data_transform、homo_nn、evaluation 结构通过 Pipeline 搭建组件结构并发布<br>• 保存刚才的算法模型组件 homo_nn_0，使用新的 Reader 读入测试数据集，搭建组件发布预测任务并验证效果 |
| | 是否符合任务书要求进度　　是 |
| 尚需完成的任务 | 1. 基于 Pipeline 实现横向联邦 CNN 完成中文文本主题分类，完成 CNN 模型结构在联邦学习上的实现<br>2. 在 FATE 平台上完成预测与评估，对比分析算法在集中式环境与联邦环境的差异，从准确率，安全性，通信效率等进行分析<br>3. 阅读 FATE 框架源代码，学习工程设计思路并思考测试现存问题及其对应优化方向<br>4. 根据实验结果，完成论文的撰写 |
| | 是否可以按期完成设计（论文）　　是 ☑　　否 ☐ |

<table>
<tr><td rowspan="2">存在问题和解决办法</td><td>存在问题</td><td colspan="3">1. 对较复杂模型结构或较大规模数据进行处理时，容易内存溢出<br>2. 对当前 Fate 框架工程架构设计理解尚不明晰，导致在实验过程中遇到问题进行 Debug 缺少方向与日志<br>3. 联邦学习环境与集中式环境对比测试中需要对通信效率、计算效率进行对比评估，但尚未确定具体的评估标准与测试标准</td></tr>
<tr><td>拟采取的办法</td><td colspan="3">1. 通过租用线上 GPU 平台完成多机部署，搭建实验环境（如 AutoDL、Vast.ai 等平台）<br>2. 继续阅读 Fate 工程源代码，并总结其工程架构与设计思路<br>3. 参考传统或分布式机器学习框架优化计算性能方向的论文，学习其评估通信效率、计算效率的评估方法</td></tr>
<tr><td colspan="2">指导教师签字</td><td></td><td>日期</td><td>年　月　日</td></tr>
<tr><td colspan="2">检查小组评分及意见</td><td colspan="3">评分：　（总分：　）<br><br>组长签字：　年　月　日</td></tr>
</table>

注：可根据长度加页。

# 北 京 邮 电 大 学

## 教师指导本科毕业设计（论文）记录表

| 学院 | 网络空间安全学院 | | 专业 | 网络空间安全专业 | |
|---|---|---|---|---|---|
| 学生姓名 | 卢亭松 | 学号 | 2019212443 | 班级 | 2019211806 |
| 指导教师姓名 | 陆月明 | 职称 | 教授 | | |

第 1－2 周记录：

**1. 学习机器学习、隐私保护相关的知识，对研究背景进行调研**

联邦学习的定义为：在进行机器学习的过程中，各参与方可借助其他方数据进行联合建模。各方无需共享数据资源，即数据不出本地的情况下，进行数据联合训练，建立共享的机器学习模型。

联邦学习系统需要保证：|联邦学习模型的效果-传统方法模型的效果|<有界正数。

联邦学习的价值机制：联邦学习技术基于“合作共赢”的价值机制，对于商业利益而言极具价值。在这样一个联邦机制下，各个参与者的身份和地位相同，而联邦系统帮助大家建立了“共同富裕”的策略，能够带动跨领域的企业级数据合作、催生基于联合建模的新业态和模式、降低技术提升成本和促进创新技术发展。

联邦学习在不同场景下的联邦方式：

1. 横向联邦学习：在两个数据集的用户特征重叠较多而用户重叠较少的情况下，将数据集按照横向（即用户维度）切分，并取出双方用户特征相同而用户不完全相同的那部分数据进行训练。这种方法叫做横向联邦学习。
2. 纵向联邦学习：在两个数据集的用户重叠较多而用户特征重叠较少的情况下，把数据集按照纵向（即特征维度）切分，并取出双方相同而用户特征不完全相同的那部分数据进行训练。这种方法叫做纵向联邦学习。
3. 联邦迁移学习：在两个数据集的用户与用户特征重叠都较少的情况下，不对数据进行切分，而可以利用迁移学习来克服数据或标签不足的情况。这种方法叫做联邦迁移学习。

现有的联邦学习开源实现：目前业界中主要的联邦学习框架有 FATE、TensorFlow Federated、PaddleFL、Pysyft 等。

**2. 基于 FATE 搭建联邦学习集群实验环境**

部署 CentOS7 主机 x2、Docker 20.10.21 环境

下载 1.8.0-a 版本 KubeFATE 并解压

修改配置文件 parties.conf

执行部署脚本，（bash ./generate_config.sh，bash ./docker_deploy.sh all）

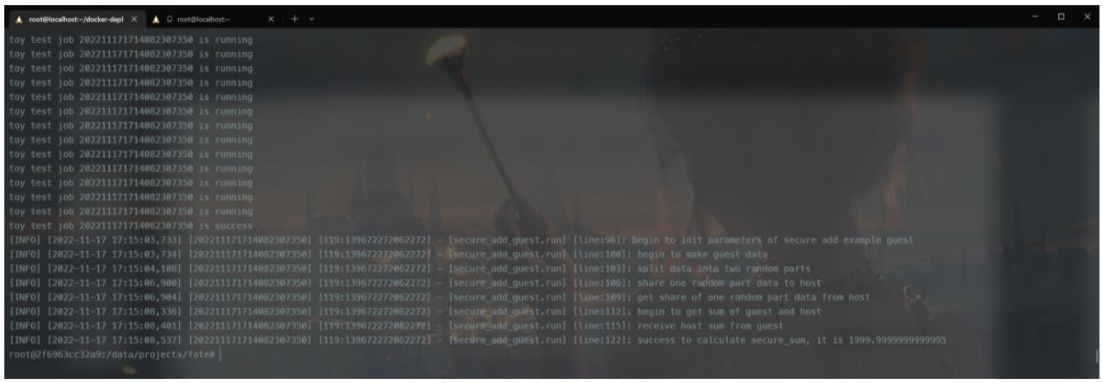
测试部署是否成功

$ docker exec -it confs-10000_client_1 bash

$ flow test toy --guest-party-id 10000 --host-party-id 9999

最终结果:

若成功，输出如下




### 3. 基于 DSL 配置任务

**Host方上传数据**

进入party10000 client容器

```
docker exec -it confs-10000_client_1 bash
```

修改examples/upload_host.json

```
cat > fateflow/examples/upload/upload_host.json <<EOF
{
  "file": "examples/data/breast_hetero_host.csv",
  "id_delimiter": ",",
  "head": 1,
  "partition": 10,
  "namespace": "experiment",
  "table_name": "breast_hetero_host"
}
EOF
```

1. file: 文件路径
2. table_name&namespace: 存储数据表的标识符号
3. head: 指定数据文件是否包含表头
4. partition: 指定用于存储数据的分区数

上传host数据

```
flow data upload -c fateflow/examples/upload/upload_host.json
```

修改 examples/test_hetero_lr_job_dsl.json

配置算法组件间的输入输出流及接口，根据此拼接得到完整的工作模型

### 提交任务

```
flow job submit -d fateflow/examples/lr/test_hetero_lr_job_dsl.json -c
fateflow/examples/lr/test_hetero_lr_job_conf.json
```

成功如下


## 4. Pipeline 配置任务

### 安装Pipeline与基本配置

Pipeline与fate_clinet相绑定，在任意具有python、pip等基础环境的机器上，运行`pip install fate_client`，即可完成安装。

成功安装Fate_client后，用户需要为Pipeline配置服务器信息和日志目录。Pipeline提供了用于快速设置的命令行工具。运行以下命令以了解更多信息。(这里需要找一下安装的位置添加一下环境变量)

```
pipeline --help
```

pipeline需要配置关联的`FATE Flow Service`,假设在`192.168.160.139`存在`FATE Flow Service`,则执行以下命令进行配置:

```
pipeline init --ip 192.168.160.139 --port 9380
```

后续我们可以在python文件中使用pipeline相关库，自动化地向fate_flow服务上传数据，发布训练任务等。

```
from pipeline.backend.pipeline import PipeLine
```

## 使用Pipeline上传数据

在开始建模任务之前，应上传要使用的数据。通常，参与方通常是包含多个节点的群集。因此，当我们上传这些数据时，数据将被分配给这些节点。

生成Pipeline实例

```
from pipeline.backend.pipeline import PipeLine
pipeline_upload = PipeLine().set_initiator(role='guest',
party_id=9999).set_roles(guest=9999, host=10000)
```

定义数据存储分区

```
partition = 4
```

定义表名和命名空间，这将在 FATE 作业配置中使用

```
dense_data_guest = {"name": "breast_hetero_guest", "namespace": f"experiment"}
dense_data_host = {"name": "breast_hetero_host", "namespace": f"experiment"}
tag_data = {"name": "breast_hetero_host", "namespace": f"experiment"}
```

添加要上传的数据(注意：这里只会上传到guest，之后host读到的数据是之前DSL配置阶段上传的数据，理论上这里应该guest和host分别上传自己的数据，这里图方便就全上传到guest了；正常上传流程参考IMDB情感分类部分的上传)

```
import os
data_base = "workspace/FATE/" #根据工作目录灵活修改
pipeline_upload.add_upload_data(file=os.path.join(data_base,
"examples/data/breast_hetero_guest.csv"),
                                table_name=dense_data_guest["name"],             # table
name
                                namespace=dense_data_guest["namespace"],         #
namespace
                                head=1, partition=partition)               # data info

pipeline_upload.add_upload_data(file=os.path.join(data_base,
"examples/data/breast_hetero_host.csv"),
                                table_name=dense_data_host["name"],
                                namespace=dense_data_host["namespace"],
                                head=1, partition=partition)

pipeline_upload.add_upload_data(file=os.path.join(data_base,
"examples/data/breast_hetero_host.csv"),
                                table_name=tag_data["name"],
                                namespace=tag_data["namespace"],
                                head=1, partition=partition)
```

执行数据上传任务

```
pipeline_upload.upload(drop=1)
```

### 使用Pipeline完成secureboost训练与测试

完成数据上传后，使用Pipeline发布训练任务

引入Pipeline及相关组件库

```
from pipeline.backend.pipeline import PipeLine
from pipeline.component import Reader, DataTransform, Intersection, HeteroSecureBoost,
Evaluation
from pipeline.interface import Data
#创建实例
pipeline = PipeLine() \
        .set_initiator(role='guest', party_id=9999) \
        .set_roles(guest=9999, host=10000)
```

定义Reader组件加载数据

```
reader_0 = Reader(name="reader_0")
# set guest parameter
reader_0.get_party_instance(role='guest', party_id=9999).component_param(
    table={"name": "breast_hetero_guest", "namespace": "experiment"})
# set host parameter
reader_0.get_party_instance(role='host', party_id=10000).component_param(
    table={"name": "breast_hetero_host", "namespace": "experiment"})
```

添加 DataTransform 组件解析原始数据到数据实例中

添加组件到pipeline构建模型并编译

```
pipeline.add_component(reader_0)
pipeline.add_component(data_transform_0, data=Data(data=reader_0.output.data))
pipeline.add_component(intersect_0, data=Data(data=data_transform_0.output.data))
pipeline.add_component(hetero_secureboost_0,
data=Data(train_data=intersect_0.output.data))
pipeline.add_component(evaluation_0, data=Data(data=hetero_secureboost_0.output.data))
pipeline.compile();
```

提交训练任务

```
pipeline.fit()
```

| | | | |
|---|---|---|---|
| 指导教师签字 | | 日期 | 年　月　日 |

第 3—4 周记录：

1. **基于 Pipeline 实现横向联邦 LSTM 完成 IMDB 文本情感分类**

本样例是在FATE框架上，使用keras风格搭建**LSTM长短期神经网络**模型，完成**IMDB数据集**上的文本情感分类任务。任务类型为**文本二分类**任务。主要包含以下流程：

1. 文本数据预处理
2. 基于Pipeline完成IMDB数据上传
3. 基于Pipeline构建LSTM模型并发布训练任务
4. 模型训练效果查看与评估

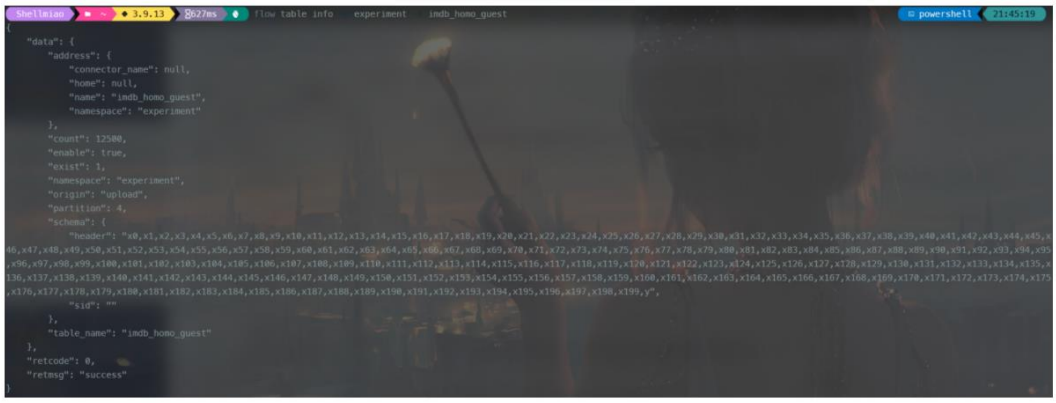
[LSTM字符粒度下文本序列生成学习与实践](https://md.shellmiao.com/s/QCKFa9hFl)

1. IMDB 文本数据预处理
    1. 添加需要用到的库文件(Tensorflow-gpu 2.10.1)
    2. 读入训练数据和测试数据
    3. 使用 tokenizer 将数据中的每个单词转化为整数值的唯一映射
    4. 将数据集按联邦节点划分并输出到 csv 文件
2. 基于 Pipeline 上传 IMDB 数据
3. 使用 Fate-cli 查看状态（一种 Debug 方式）

[Fate-cli使用手册](https://fate.readthedocs.io/en/develop-1.5/_build_temp/python/fate_client/flow_client/README.html)

```
flow init --ip 192.168.160.139 --port 9380
```

```
flow table info -n experiment -t imdb_homo_guest
```




上传后的数据表名为：

`imdb_homo_guest,imdb_homo_host,imdb_homo_test_guest,imdb_homo_test_host`

可分别对Guest、Host验证

4. 基于 Pipeline 构建 LSTM 模型并发布训练任务

### 基础配置

```
from pipeline.backend.pipeline import PipeLine
from pipeline.component import DataTransform
from pipeline.component import Reader
from pipeline.component import HomoNN
from pipeline.interface import Data
from pipeline.component import Evaluation
import os
os.environ["CUDA_VISIBLE_DEVICES"]="-1"

pipeline = PipeLine() \
        .set_initiator(role='guest', party_id=9999) \
        .set_roles(guest=9999, host=[10000], arbiter=10000)
# 别忘了把pipeline切换回guest
```

### 数据读入组件

```
reader_0 = Reader(name="reader_0")
reader_0.get_party_instance(role='guest', party_id=9999).component_param(
    table={"name": "imdb_homo_guest", "namespace": "experiment"})
reader_0.get_party_instance(role='host', party_id=10000).component_param(
    table={"name": "imdb_homo_host", "namespace": "experiment"})

data_transform_0 = DataTransform(name="data_transform_0", with_label=True)
data_transform_0.get_party_instance(role='guest', party_id=9999).component_param(
    with_label=True,label_name="y")
data_transform_0.get_party_instance(role='host', party_id=[10000]).component_param(
    with_label=True,label_name="y")
```

### 构建LSTM模型

词嵌入层设置128个神经元，LSTM层包含64个神经元；最大迭代次数为100，batch长度为32，设置early_stop机制。使用Adam进行梯度更新，学习率为1e-5，损失函数为二元交叉熵

```
max_features = 10000
max_len = 200
embedding_neurons = 128
lstm_neurons = 64

homo_nn_0 = HomoNN(
        name="homo_nn_0",
#         encode_label=True,
        max_iter=100,
        batch_size=32,
        early_stop={"early_stop": "diff", "eps": 0.0001},)

from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Reshape
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import LSTM
```

```
homo_nn_0.add(Embedding(max_features,embedding_neurons,input_length=max_len))
homo_nn_0.add(BatchNormalization())
homo_nn_0.add(LSTM(units=lstm_neurons, dropout=0.2 ,recurrent_dropout=0.2))
homo_nn_0.add(Dropout(0.5))
homo_nn_0.add(Dense(units=1,activation='sigmoid'))

from tensorflow.keras import optimizers
homo_nn_0.compile(
    optimizer=optimizers.Adam(learning_rate=0.00001),
    metrics=["accuracy", "AUC"],
    loss="binary_crossentropy")
```

#### 完成组件搭建和任务发布

```
evaluation_0 = Evaluation(name="evaluation_0", eval_type="binary")

pipeline.add_component(reader_0)
pipeline.add_component(data_transform_0, data=Data(data=reader_0.output.data))
pipeline.add_component(homo_nn_0, data=Data(train_data=data_transform_0.output.data))
pipeline.add_component(evaluation_0,data=Data(homo_nn_0.output.data))
pipeline.compile();

pipeline.fit()
```

5. 遇到的问题解决方案

#### no job could be found问题

**问题**：使用本地pipeline发布任务，运行时报错 `{'data': [], 'retcode': 0, 'retmsg': 'no job could be found'}`

**解决**：pipeline init的ip和代码init的ip不一样

#### tensorflow组件缺失

在执行神经网络，使用tensorflow相关组件构建算法模型，报错 `No module named tensorflow` ,是因为fate自身部署的python环境缺乏tensorflow，在每个节点使用如下命令进入python容器。

```
docker exec -it confs-xxxx_python_1 bash
```

使用pip安装适合版本的tensorflow即可，这里使用1.15.0版本

```
pip install tensorflow==1.15.0 -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com
```

**2. 基于 Pipeline 实现横向联邦 CNN 完成中文文本主题分类**

正常的 CNN 完成中文文本主题分类 学习记录可以参考如下这篇博客：

[[NLP]基于 CNN 对 THUCNews 数据集进行文本分类]( https://md.shellmiao.com/s/aLxE1btif)

1. 数据预处理
2. 分别上传至 guest、host 主机
3. 训练模型

```
from pipeline.backend.pipeline import PipeLine
from pipeline.component import DataTransform
from pipeline.component import Reader
from pipeline.component import HomoNN
from pipeline.interface import Data
from pipeline.component import Evaluation
import os
os.environ["CUDA_VISIBLE_DEVICES"]="-1"

pipeline = PipeLine() \
        .set_initiator(role='guest', party_id=9999) \
        .set_roles(guest=9999, host=[10000], arbiter=10000)
```

定义reader、data_transform

```
reader_0 = Reader(name="reader_0")
reader_0.get_party_instance(role='guest', party_id=9999).component_param(
    table={"name": "thucnews_guest", "namespace": "experiment"})
reader_0.get_party_instance(role='host', party_id=10000).component_param(
    table={"name": "thucnews_host", "namespace": "experiment"})

data_transform_0 = DataTransform(name="data_transform_0", with_label=True)
data_transform_0.get_party_instance(role='guest', party_id=9999).component_param(
    with_label=True,label_name="y")
data_transform_0.get_party_instance(role='host', party_id=[10000]).component_param(
    with_label=True,label_name="y")
```


4. 遇到的问题与解决方法

```
TypeError: ('Keyword argument not understood:', 'groups')
```

这是tensorflow和keras版本低了导致的，需要fate的python docker里面的包升级了一下

```
# 进入docker
docker exec -it confs-xxxx_client_1 bash
# 更新包
pip install --upgrade tensorflow -i https://pypi.mirrors.ustc.edu.cn/simple/
```

| 指导教师签字 | | 日期 | 年　月　日 |
|---|---|---|---|

注：每 2 周指导内容记录在一个表格中，双面打印。