# 中国地质大学

# 学士学位论文

# 联邦学习优化梯度保护技术研究


学　　号：20181001095

姓　　名：常文瀚

学 科 专 业：计算机科学与技术

指 导 教 师：朱天清教授

培 养 单 位：计算机学院


二〇二二年五月

# 中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《基于知识图谱的智能问答系统》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：常文瀚

日　期：2022 年 5 月 30 日

# 摘要

在过去数年内，针对联邦学习的研究不断发展，相关领域涌现出了大量既能保证模型精确度，又能很好的保护客户端数据隐私的联邦学习算法及框架，这大大推进了人工智能安全技术的发展。很长一段时间里，研究者们认为，联邦学习框架内服务器通过聚合客户端上传的更新梯度等参数来完成全局模型更新的方法能够完全保护客户端的训练数据，但是不断有研究者通过实验证明从公开共享的梯度中推断出私有训练数据是完全可行的。

在本文中，我们复现了一种最新提出的针对纵向联邦学习的数据泄露攻击方法，与现有的以模型梯度作为攻击媒介的数据泄露攻击方法相比，该方法在纵向联邦学习中的大量实验结果证明了它在进行大批量数据泄露攻击并提高数据恢复质量时具有有效性与稳定性。实验同时证明了在纵向联邦学习中，私有的训练数据有非常高的风险被泄露。除此之外，我们基于余弦相似度、梯度稀疏化和伪梯度等概念，提出了两种可行的防御方法来抵御该攻击方法以达到对所有联邦学习参与者的私有数据做到最大程度的保护。这两种方法都使用了余弦相似度来衡量客户端训练时更新的梯度和服务器端更新的梯度的角度差距，以余弦相似度为参考，客户端拥有能力选择对自身上传梯度进行伪装的策略，来达到保护自己隐私的同时保持联邦学习全局模型的精确度。通过大量实验，我们证明了两种防御方法均可以在保证联邦学习全局模型的精确度的情况下保护用户的私有数据。

**关键词：**联邦学习；分布式计算；数据泄露；隐私保护

# Abstract

In the past few years, research on federated learning has continued to develop, and a large number of federated learning algorithms and frameworks have emerged that can not only ensure model accuracy, but also protect client data privacy. This has greatly promoted the development of artificial intelligence security technology. For a long time, researchers believed that the method of updating the global model by aggregating the gradients and other parameters uploaded by clients in the federated learning framework can completely protect the training data of clients. However, it is completely possible to recover private training data from publicly shared gradients.

In this paper, we reproduce a newly proposed data leakage attack method for vertical federated learning. Compared with existing data leakage attack methods using model gradients as an attack medium, a large number of experimental results based on vertical federated learning show that it is effective and stable when conducting large-scale data leakage attacks and improving the quality of data recovery. The experiments also demonstrate that in vertical federated learning, private training data has a very high risk of being leaked. In addition, based on the concepts of cosine similarity, gradient sparsification, and pseudo-gradient, we propose two feasible defense methods to defend against this attack method to achieve maximum protection of the private data of all federated learning participants. Both methods use cosine similarity to measure the angular difference between the gradient updated by the client during training and the gradient sent back by the server. Taking the cosine similarity as a reference and aiming to protect clients' privacy while maintaining the accuracy of the federated learning global model, the clients have the ability to choose a strategy for disguising its uploaded gradient. Through extensive experiments, we demonstrate that both defense methods can protect users' private data while maintaining the accuracy of the federated learning global model.

**Key Words:** Federated Learning; Distributed Computing; Data Leakage; Privacy Preserving

# 目录

# 第一章 绪论

## 1.1 研究背景及意义

联邦学习作为一种新兴的分布式机器学习技术，旨在通过众多客户端上的数据集来训练一个全局模型，且无需在客户端之间显式地交换数据样本。过去的数年内，全世界的研究者们共同见证了联邦学习的快速发展，然而在联邦学习训练过程中也出现了新的隐私问题。新兴的隐私保护式联邦学习 (PPFL)被认为是通用的隐私保护机器学习的解决方案，可在保持机器学习中数据效用的同时保护数据隐私的挑战仍然存在。

例如联邦学习训练过程中的梯度交换，正因为过去人们普遍认为只交换梯度是可以保护隐私的，很少有人探索过通过共享的梯度造成的数据泄露这一方向。但是最近的一些研究表明，联邦学习中被客户端上传的梯度揭露了其所属训练数据的一些属性，一些新技术甚至可以通过公开的梯度完全恢复客户端拥有的数据集，这些技术可能揭示了联邦学习前提条件的根本缺陷。与此同时，通过控制联邦学习中共享的梯度以达到保护客户端数据隐私的防御方法也在被不断提出，为了完善联邦学习中的隐私保护这一作用，我们在复现现有的数据泄露攻击方法的基础上提出并分析了两种防御方法，研究证明这两种方法均可以对客户端的私有数据集起到保护作用。

## 1.2 国内外研究现状

与现有的分布式机器学习学习框架相比，联邦学习面临着了新的挑战。为了保护数据隐私，服务器和客户端之间只交换模型参数和参数的变化（如梯度和梯度的更新），而利用梯度恢复客户端的训练数据这一技术正在受到越来越多的关注。

在 2019 年，一种被称为“深度信息泄露”（DLG）的图像恢复方法被研究了出来，该方法可以在不使用任何已生成模型或先验信息的情况下有效推断训练数据。在[1]中，该文章的作者开发出了一种可以通过梯度提取准确标签的分析方法。在[2]中，另一种可以导出第一个全连接层的输入的分析方法被提出，然而他们的方法只能恢复单个的输入样例不能成功恢复一个批数据。除了[2]中提出的新的损失函数外，以前的一些工作也在 DLG 的基础上设计了新的损失函数或正则化器，并试图使其

算法可以普适地适用于各种模型和权值分布。在[3]中，一种新的基于高斯核的梯度差被用来作为距离度量。在[4]中，研究者开发了一种从梯度中恢复数据的递归方法攻击程序。但是在[3]和[4]中，批量数据恢复的质量依旧会下降。最近[5]提出了一种名为“GradInversion”的算法，该算法基于给定的梯度从噪声中重建图像。然而，他们的理论和算法大多建立在强有力的假设和经验观察之上，虽然他们成功地重构了一批训练数据，但批量大小仍然不超过 48。

在 DLG 被提出后，相关的防御方法也在不断被全世界的研究者研究。在[6]一文中，研究者们提出了多种存在可能的 DLG 防御方法。在[7]中，作者提出了 Pivot，这是一种保护隐私的垂直决策树训练和预测的解决方案，确保除了客户同意发布的中间信息（即最终的树模型和预测输出）之外，不会泄露任何中间信息。Pivot 不依赖任何可信任的第三方，并提供保护，以抵御可能会危及客户端隐私的攻击者。在[8]中，研究者提出了一种基于梯度的差异隐私优化器，该优化器采用基于 CPU-GPU 混合系统的协同训练模式。在基于梯度的差分隐私优化器中，随机抽样、梯度裁剪、基于梯度的随机扰动和高级隐私预算统计共同保证了模型的可用性和隐私性。在[9]中，作者以保持模型精度的同时，训练具有差分隐私（DP）保证的深度学习模型为目标，对梯度进行编码，将其映射到更小的向量空间，从而使我们能够获得不同噪声分布的 DP 保证。同时还利用了差分隐私的后处理特性引入了去噪的思想，进一步提高了训练模型的实用性，同时又不降低其 DP 保证。

## 1.3 研究内容与主要工作

在本文中，我们主要研究了联邦学习梯度保护技术，在对纵向联邦学习及相关领域的数据泄露攻击方法进行了充分的学习与研究后，做出了以下几点工作：

（1） 我们将分析一些纵向联邦学习的背景，同时深入了解横向与纵向联邦学习在不同场景下的使用方法以及他们具有何种优势。

（2） 我们将分析一种最新的针对纵向联邦学习的数据泄露攻击方法。在对该方法进行复现的同时记录其实验结果。

（3） 基于现有的研究成果，我们将针对最新的纵向联邦学习数据泄露攻击方法提出分别利用梯度稀疏化技术和伪梯度技术的防御方法。

（4） 通过实验，我们验证了纵向联邦学习数据泄露攻击方法在联邦学习训练期间的有效性，以及和其它方法相比起来的优越性，同时验证了我们提出的两种防御方法的有效性和鲁棒性。

# 第二章 相关理论知识

## 2.1 卷积神经网络

人工神经网络（ANN）是以大量的处理单元（神经元）互相连接而形成复杂网络结构进而对生物神经系统的模拟网络。人工神经网络主要由大量相互连接的计算节点（称为神经元）组成，这些节点以分布式方式相互缠绕，从输入中集体学习[10]，以优化其最终输出。ANN 的基本结构可以通过建模实现。

卷积神经网络（CNN）与传统的人工神经网络虽然结构不同，但他们内在都是由神经元组成，不同的神经元通过对输入数据的不断学习进行着自我优化。在学习过程中，神经元会和激活函数结合在一起对输入信息进行计算，并将结果传送到下一层。网络中的最后一层将包含与分类相关的损失函数，所有为传统人工神经网络开发的常规标准以及一些技巧仍然适用[10]。相较于 ANN，CNN 更加适合进行与图像相关的任务，因此我们可以尝试将一些图像优化技术融入到 CNN 里，这样便可以在一定程度上优化 CNN 的性能。

卷积神经网络一般包括输入层、卷积层、池化层、全连接层及输出层构成[10,11]。其中，不同学习任务中 CNN 的卷积层和池化层数量要视具体情况而定，全连接层前面的部分由卷积层和池化层交替组成，不同数量的卷积层与池化层会直接地影响到神经网络的学习效果。如上文所述，CNN 主要关注的是针对网络的输入数据将由图像组成，这预示着将要设置的神经网络结构，将以最适合处理这种特定类型数据的需要为前提。

其中，卷积层（convolutional layer）由多个特征面（Feature Map）组成，每个特征面包含了多个神经元。卷积核就是他的每一个神经元与上一层特征面的部分区域相连的媒介。卷积核是一个权值矩阵（如对于二维而言可为 3*3 或 5*5 矩阵）[12,10,13]。卷积层将通过计算神经元权值与连接到输入数据区域之间的标量积来确定连接到输入数据局部区域的神经元的输出[10]。校正线性单元（通常缩写为 ReLu）旨在将激活函数（如 sigmoid）引入卷积神经网络，以输出前一层产生的激活结果。

神经网络中使用池化层一方面可以降低特征图像的维度提高运算性能，另一方面可以通过压缩图像防止模型过拟合。在一些 CNN 的构建中，人们一般会选择最大池化作为池化方法。直观的来说，池化层可以对图像进行遍历，再将图像缩小到一定程度，提取出重要信息，一些网络中的池化层可以在将原图缩小到原始大小

的 25%时将图像深度保持在其标准大小。但池化的方法也有很多，例如平均池化、随机池化等。

全连接的层将执行与其在标准 ANN 中相同的职责，并尝试从激活中生成分类概率将其用于分类。同时在这些层之间可以使用激活函数，以提高性能。全连接层包含直接连接到相邻两层神经元的神经元，而不连接到其中的任何层。这类似于神经网络传统形式中神经元的排列方式。

传统的神经网络存在规模的限制，这使得它在一些场景下难以满足针对图像或其他类型数据集的学习任务。若使用 MNIST 数据集进行训练，第一个隐藏层中的单个神经元将包含 784 个权重（28×28×1，其中 1 表示 MNIST 被标准化为黑白值），这对于大多数形式的 ANN 来说是可管理的[10]。

但是如果我们使用一个更大的 64×64 的彩色图像输入，第一层的单个神经元上的权重数量将大幅增加到 12288。同时还要考虑到，为了处理这种输入规模，使用的网络还需要比用于对 MNIST 数据集进行分类的网络大得多，这时我们就会理解使用 CNN 以及利用卷积层的优势[10,12]。全连接层的神经元直接连接到相邻两层的所有神经元，而不连接到其中的任何层。这类似于神经网络传统形式中神经元的排列方式。

CNN 不同于传统形式的神经网络，他不会将整个数据作为自己学习的对象，而是利用卷积池化提取相对重要的信息进行学习。这反过来又允许我们建立更简单的网络体系结构。在联邦学习隐私安全保护领域，神经网络的结构有着至关重要的地位，在下文中，我们针对神经网络的更新梯度，提出了两种有效的数据泄露防御方法。

## 2.2 分布式机器学习

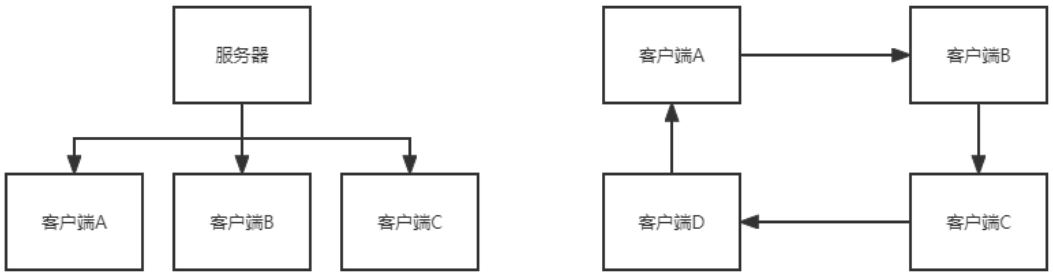
近十年来，随着信息行业内前沿技术与相关产业的快速发展，全球数据产生量以一个从未有过的速度爆炸式增长。在 2020 年全球数据量已到达 50.5ZB，并在 2025 年时预测达到 163ZB 之巨。

这也使得在人工智能领域可使用的训练数据爆炸式增长。著名的图像分类数据集 ImageNet[14]总大小约 1TB，共计 1419 万张图片分为 21841 种类别。自然语言处理模型 DeepSpeed2[15]的训练采取了 1 万余小时的语语音样本，合计约 200 万条叙述。初代 AlphaGo[16]的训练参考了人类历史超过三千万次的对局数据，并于训练后的 40 天登顶世界冠军。在海量训练数据的基础上，为了进一步提高准确性，大模型也开始逐渐流行。早期的神经网络仅有数十个参数，而目前拥有上千万个参数

的模型也屡见不鲜。多的可以甚至达到十几亿，人们对于人工智能的未来充满了期待，同时也对计算效率、通信效率、系统稳定性等提出了更高的要求。

众所周知，训练大型的机器学习模型需要消耗非常大的算力，为了在合理的时间内完成训练过程，许多研究都致力于分布式训练以加快速度。目前有很多研究者都希望优化算法[17]或优化训练框架[18]来提高分布式训练的训练表现。因此人们开构建分布式机器学习平台，在前所未有的大数据支持下，训练大规模模型变得更为容易。

一般来说，分布式训练可以分为两类，一种是有中央服务器分布式训练（集中式）[19]，另一种是无中央服务器分布式训练（分散式）。在这两种训练框架中，每个分布式训练参与者首先利用本地数据训练本地模型以更新其局部模型的梯度，然后将梯度发送到其他参与者。在集中式模式中，梯度首先被中央服务器聚合并更新，然后发送给每个参与者。在分散模式中，相邻参与者会不断交换梯度以完成模型参数的更新。



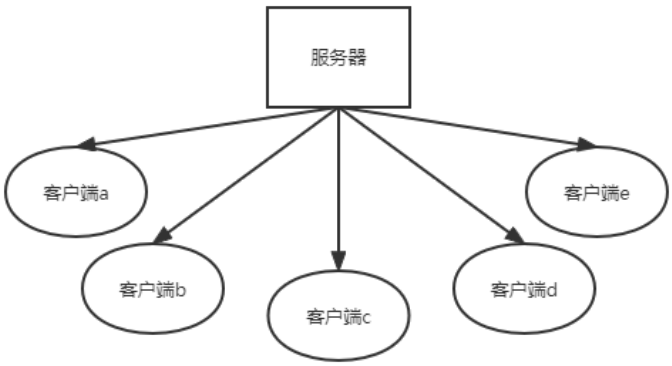

图（1） 集中式与分散式的分布式机器学习

在许多的应用场景中，训练数据作为一种隐私是十分敏感的。例如，一位患者的医疗状况不可以被医院之间分享。分布式协作学习系统的出现在很大程度上避免了，训练参与者数据的泄露，在整个学习系统中存在着两个以上的参与者，而训练数据永远不会离开每个参与者的本地服务器，只有训练梯度在不同训练参与者的网络中被传递。此类技术已被用于训练多个医院的医疗模型、分析来自不同国家的患者生存状况，甚至是构建可以学习用户打字习惯的键盘[20,21]来优化用户的打字体验。

## 2.3 联邦学习

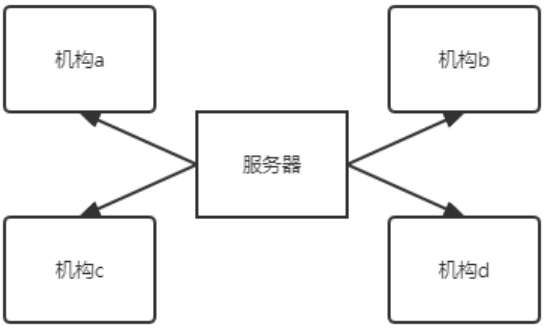
联邦学习[22–24]是一种新兴的基于分散数据的分布式学习框架，在联邦学习中有多个客户端（如智能手机、物联网设备和边缘设备）和一个服务提供者（如谷歌、苹果和 IBM）。每个客户端持有一个本地训练数据集，服务提供者使客户端能够共同学习一个模型（称为全局模型），而无需与服务提供者共享其原始的本地训练数据。

一些现有的联邦学习方法[25]考虑了每个参与者拥有一组不同的数据但具有共同特征的情况。在传统的机器学习模型中，通常是把模型训练需要的数据集合到一个数据中心然后再训练模型，之后预测。在横向联邦学习中，可以看作是基于样本的分布式模型训练，服务端分发全部数据到不同的客户端，每台客户端从服务端下载模型，然后利用本地数据训练模型，之后返回给服务器需要更新的参数，在这个过程中，每台客户端下都是完整的模型，且客户端之间互不交流互不依赖，在预测时每台客户端也可以独立预测。服务器在将所有客户端上传的更新数据聚合后，首先完成服务器端全局模型的更新，然后将新模型参数可以再次被客户端下载。




图（2） 横向联邦学习结构

与横向联邦学习不同，在许多的学习场景中，多个客户端处理关于同一组对象的数据，但是他们中的每一个客户端所拥有的数据包含了同一对象的不同特征。这种情况在金融领域和医疗领域十分常见[25,7,26]。在不同的案例中，数据所有者（例如金融机构和医院）在他们的用户数据库中有着用户们的不同记录，通过联邦学习将用户们的特征组合，数据所有者便可以建立一个更加精确的模型。我们将拥有此类特点的联邦学习称作纵向联邦学习。




图（3） 纵向联邦学习结构

最后一种联邦学习模式叫做“联邦迁移学习”，在各个行业的实际场景中，大多数应用程序只能访问小型或低质量的数据集。在对属于需要人类专业知识支持的数据进行标记的时候，往往需要很大人力物力。此外，某些任务使用的数据往往是分布式存储的。许多组织可能只有未标记的数据，而其他一些组织的标签数量可能非常有限。迁移学习可以通过学习数据量较大的源知识构建一个模型，再利用这个模型和少量不相同但相关的另一领域知识完成新模型的训练。各个领域之间数据的相关程度决定了迁移学习的学习效果，迁移学习赋予了神经网络举一反三的能力。联邦迁移学习是第一个使联邦学习在迁移学习中受益的框架。

## 2.4 联邦学习中的主要攻击方法

现代机器学习系统可能容易受到各种故障的攻击，例如预处理管道中的漏洞和嘈杂的训练标签，以及针对系统训练和部署管道的每一步的攻击。攻击的方法包括数据和模型更新中毒[27]、模型规避、模型窃取和对用户训练数据的[6,28]数据推断攻击。其中，模型窃取攻击已经从多个方面得到了广泛的研究，包括参数窃取、超参数窃取、架构抽取、决策边界推理和功能窃取。

联邦学习的分布式特性，尤其是在使用安全聚合协议[29]进行增强时，使得检测和纠正这些故障和攻击成为一项特别具有挑战性的任务。根据攻击目标，对抗性攻击可大致分为两种类型，非目标攻击或目标攻击。在非目标攻击下，对手的目标是破坏模型，使其无法在通常被称为主要任务的工作（例如分类）上实现接近最优的性能。在目标攻击下，攻击者希望联邦学习模型在某些子任务上有着较差的表现，但却在其他子任务任务上有着较高的精确度。例如，在图像分类中，攻击者可能希望模型将一些“绿色汽车”误分类为“鸟类”，同时确保其他汽车正确分类。

对于目标攻击和非目标攻击，攻击可以根据攻击者的能力进一步分为两种类型：模型更新中毒或数据中毒[30]。数据中毒攻击中，攻击者可以更改部分学习参与

者的训练数据。在联邦学习系统中，由于训练过程在本地设备上完成，完全受损的客户端可能完全更改模型更新[31,32]，这被称为模型中毒攻击。当在联邦学习系统中部署安全聚合协议[29]时，模型更新中毒攻击就更难对付了。安全聚合协议[29]使得服务器无法检查每个用户的更新。

由于非目标攻击会降低主要任务的总体性能，因此更容易被检测到。另一方面，因为攻击者的目标往往是未知的，所以后门目标攻击往往更难被发现。因此，在[31,32]之后，我们了解学习了有目标攻击性的模型更新中毒攻击，并将其归为后门攻击。现有的后门攻击方法要么需要仔细检查训练数据，要么需要在服务器上完全控制训练过程，这可能不适用于联邦学习案例。

在本文中，我们将会主要介绍并分析一种利用联邦学习训练过程中客户端上传的梯度对用户训练数据进行推断的攻击，同时我们将会提出两种对这类后门攻击的防御方法。

## 2.5 联邦学习中由共享梯度造成的数据泄露

### 2.5.1 由梯度造成的浅层数据泄露

前人对于如何从梯度中推断出训练数据的信息做过了不少探索，对于神经网络中的某些层，梯度已经泄露了一定量的信息，例如在执行语言及文本类任务的神经网络中，嵌入层（embedding layer）只会对出现在训练集中的词语产生梯度[6]，这就可以直接反应出训练参与者的训练集中出现了什么词语。但是这样的泄露被研究者们形容为“浅层”泄露，因为这样泄露出来的信息往往杂乱无章，不能通过词语推断出原本的句子。

另一个例子则是全连接层，对全连接层梯度更新的记录可以被用于推断出其输出的特征值。然而由于卷积层的特征较为庞大，同样的方法却难以被用于卷积层。

### 2.5.2 由梯度造成的深度数据泄露

梯度的深度泄露与浅层泄露不同，梯度的深度泄露技术通过生成假图像和假梯度，以获取的真实梯度作为优化目标优化，进而恢复出数据集中使用的图像和标签[6,1,2]。为了从梯度中恢复数据，攻击者首先初始化假图像$x'$和假标签$y'$，然后将假图像送入模型$F$，获得假梯度$\nabla W'$。

$$\nabla W' = \frac{\partial l(F(x', W), y')}{\partial W} \tag{2.1}$$

将假梯度不断优化使其靠近真实梯度同样使得假图像更加像真实训练使用的图像，当我们拥有某一轮的梯度时，我们可以通过对如下公式进行优化，使其结果使其达到最小值来获得训练数据。

$$x'^{*},y'^{*}=arg\min||\nabla W'-\nabla W||^{2}=arg\min||\frac{\partial l(F(x',W),y')}{\partial V}-\nabla W||^{2} \quad (2.2)$$

需要注意的是，该方法的提出者做出了一种普适的假设，即这种优化过程要求模型$F$二阶可微，该假设适用于大多数现代的机器学习模型。

上文中指出的算法在恢复单张图像及其标签时能够得到非常好的结果。但是当我们使用这种算法去恢复一批批量大于 1 的图像时，该算法将会花很长的时间达到收敛。该文章的作者认为出现这种情况的原因是图像批量大于 1 造成了一批图像有$N!$种不同的排列方法[6]，优化器很难把拥有的梯度和对应的图像匹配。为了解决这个难题，文章作者采用的对策是对于一个有着$N$张图像的批量，以$N$轮为一个周期，在周期内的每一轮只针对一张图像进行恢复并优化结果。此后，图像恢复的过程变得更快，但不可否认的是，越大的图像批量预示着我们需要花越久的时间对所有图像进行恢复。

## 2.6 联邦学习中的梯度保护方法

### 2.6.1 噪声梯度

防御通过梯度造成的数据泄露这一攻击方法的直接尝试，就是在梯度上添加噪声[9,6]，现有的研究表明：通过添加噪声对梯度进行伪装的防御效果主要取决于噪声方差的大小，而非由噪声的种类[6]决定。在下表中，针对高斯噪声和拉普拉斯噪声，我们可以看到当向梯度内添加噪声的方差大小超过了一定阈值便可以成功进行防御，但是若选择较大的方差却会使模型精确度下降。

表（1） 不同噪声类型及在不同方差下能否进行防御

| 噪声类型 | 方差 $=10^{-4}$ | 方差 $=10^{-3}$ | 方差 $=10^{-2}$ | 方差 $=10^{-1}$ |
|---|---|---|---|---|
| Laplace | x | x | √ | √ |
| Gaussian | x | x | √ | √ |

### 2.6.2 梯度压缩与稀疏化

起初梯度压缩与梯度稀疏化主要被应用于降低多客户端学习的通信开销，但是研究表明，梯度压缩与梯度稀疏化可以被应用于对客户端数据隐私的保护。在现有的文献中可以了解到，如果把较小的梯度削减为 0，对于攻击者来说将会使优化器以原始梯度为目标优化假梯度这一任务变得更加困难。现有的研究表明：当稀疏度为 1%～10%时，对图像恢复任务几乎没有影响，当稀疏度增加到 20%时，恢复的图像上会存在明显的噪声像素点。同时，当稀疏度大于 20%时，恢复的图像将无法通过视觉识别[6]，梯度压缩可以成功地防止数据泄露。

### 2.6.3 提高图像批量、提高图像分辨率以及密码学

在训练参数是被允许修改的情况下，我们可以使用更多的方法进行防御，参考现有的研究数据，我们可以了解到增加图像批量的大小或是提高图像分辨率会增加优化过程中需要处理的变量数量，尽管做出此类调整需要修改神经网络结构。

在“Catastrophic Data Leakage in Vertical Federated Learning”一文中，作者对为什么针对大批量数据的攻击难以做到这一问题进行了论述。假设在某一轮 K 张图像被选为对模型的输入样例，我们便可以将选中的批量数据定义为 $D'=\{(x_n,y_n)\}$，类似的，我们可以将相同批量的恢复数据定义为 $\hat{D}'=\{(\hat{x}_n,\hat{y}_n)\}$，在这之后，针对批量恢复数据的优化目标方程可以被表示为：

$$D'=argmin||\frac{1}{K}\sum_{(x_n,y_n)}\nabla_{\Theta}L(\theta,x_n,y_n)-\frac{1}{K}\sum_{(\hat{x}_n,\hat{y}_n)}\nabla_{\Theta}L(\theta,\hat{x}_n,\hat{y}_n)||^2 \quad (2.3)$$

值得注意的是，服务器聚合的梯度的维度是固定的。然而，随着 K 的增长真实数据和假数据的基数都在不断上涨，当 K 足够大的时候通过优化目标函数结果得到与真实数据相对应的假数据变得更具挑战性。而在纵向联邦学习中，所有客户端的数据必须根据数据所属用户对齐[33]，这为大批量数据遭到攻击者恢复留下了隐患，在第三章中，我们将会具体进行介绍，并通过实验复现此类攻击算法。

此外，密码学技术同样可以被应用来预防此类数据泄露。例如，Bonawitz 等人[34]设计了一种安全聚合协议，Phong 等人[35]提出在发送梯度前对其进行加密。在所有防御方法中，利用密码学技术可以最大程度保护数据隐私，但是上文提到的两种密码学方法都有着对应的限制并且不能被普适地应用到实际场景：安全聚合协议要求梯度里的数据均为整数，导致其不能适用于大部分的卷积神经网络。

# 第三章 纵向联邦学习中的灾难性数据泄露

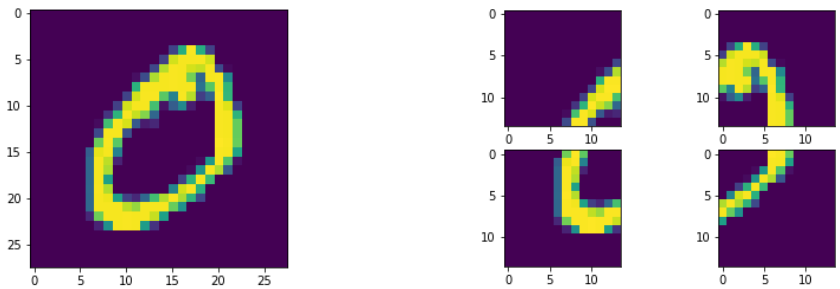
在本章中，我们将分析纵向联邦学习的一些必要背景，并分析我们复现的攻击方法 CAFE[33]（Catastrophic Data Leakage in Vertical Federated Learning）。该攻击方法考虑的攻击场景是：一个普通的服务器在遵循常规 VFL 协议的情况下对客户端上传参数进行聚合，但它却打算根据聚合梯度恢复客户端的私有数据。此方法理论上具有针对图像分类任务和自然语言处理任务的普适性，但在本文复现的实验中我们选用图像数据集进行实验。

## 3.1 攻击方法前提

### 3.1.1 训练数据的使用

该攻击算法的主要思想是将整个数据泄漏攻击过程分为几个步骤。具体来说，我们首先完全恢复模型内部对第一个 FC 层的输入。在整个过程中，为了解决图像批量过大导致攻击者无法匹配获得的梯度与目标图像，进而使恢复过程收敛速度慢、图像恢复质量差这一难题，我们采用了批量数据索引这一技术使训练数据遵循纵向联邦学习前提按照标签对齐。

在典型的纵向联邦学习过程中，服务器向本地客户端发送公钥，并在每次迭代的训练过程中确定数据指标。在此过程中，客户端之间交换中间结果，计算梯度并上传，因此服务器可以访问模型参数及其梯度。由于数据在不同的客户端之间进行纵向分区，对于每个批次，服务器（充当攻击者）需要向所有本地工作人员发送数据索引，以确保每个工作人员都选择了具有相同序列的数据[22]，我们将此步骤称为数据索引对齐。数据索引对齐是垂直培训过程中不可避免的一步，这为服务器（攻击者）提供了控制选定批量数据索引的机会。



图（4） 纵向联邦学习数据集分割

### 3.1.2 损失函数与梯度

我们假设算法中使用的模型是一个参数化的神经网络，模型将会被参数化地记作$\theta$，第一个全连接层将会被表示为$\theta_1$，其偏差被记作$b_1$，以第$t$轮训练中选取的批量数据为变量和以完整的数据集为变量的损失函数可以如下列公式表示：

$$L(\theta,D(s^t)):=\frac{1}{K}\sum_{n=1}^{N}s^t[n]L(\theta,x_n,y_n) \tag{3.1}$$

$$L(\theta,D)=\frac{1}{N}\sum_{n=1}^{N}L\left(\theta,x_{n,},y_n\right) \tag{3.2}$$

每一轮中使用了特定批量数据进行训练后，关于$\theta$的梯度可以被表示为：

$$\nabla_\theta L(\theta,D(s^t))=\frac{\partial L(\theta,D(s^t))}{\partial\theta}=\frac{1}{K}\sum_{n=1}^{N}\mathrm{s}^t[n]\frac{\partial L(\theta,x_{n,}y_n)}{\partial\theta} \tag{3.3}$$

类似的，我们将使用完整数据集进行训练后，关于$\theta$的梯度定义为：

$$\nabla_\theta L(\theta,D)=\frac{\partial L(\theta,D)}{\partial\theta} \tag{3.4}$$

## 3.2 攻击方法分析

### 3.2.1 恢复模型关于第一个全连接层输出信息的梯度

首先，我们要恢复第一个全连接层关于其输出的梯度，对于某个数据点$x_n$，我们将第一个 FC 层的输入表示为$h_n$，让$\theta_1$表示第一个全连接层的参数，$u_n$表示神经网络中第一个 FC 层的输出，$u_n$的计算方法如下所示：

$$u_n=\theta_1{}^T h_n+b_1 \tag{3.5}$$

对于训练数据集，对应第一个全连接层的输入可以被表示为$\mathbf{H}=[\mathbf{h}_1,\mathbf{h}_2,\cdots,\mathbf{h}_N]^T$，对应的输出可以被表示为$\mathbf{U}=[\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_N]^\top$，模型关于全连接层输出$u_n$的梯度可以被表示为：

$$\nabla_{\mathbf{U}}L(\Theta,\mathcal{D})=\frac{1}{N}\left[\nabla_{\mathbf{u}_1}L(\Theta,\mathbf{x}_1,y_1),\nabla_{\mathbf{u}_2}L(\Theta,\mathbf{x}_2,y_2),...,\nabla_{\mathbf{u}_N}L(\Theta,\mathbf{x}_N,y_N)\right]^\top \tag{3.6}$$

则有：

$$\nabla_{\mathbf{b}_1} L(\Theta, \mathcal{D}(\mathbf{s}^t)) = \frac{1}{K}\sum_{n=1}^{N} \mathbf{s}^t[n] \frac{\partial L(\Theta, \mathbf{x}_n, y_n)}{\partial \mathbf{b}_1} = \sum_{n=1}^{N} \mathbf{s}^t[n] \frac{1}{K}\sum_{z=1}^{N} \mathbf{s}^t[z] \frac{\partial L(\Theta, \mathbf{x}_z, y_z)}{\partial \mathbf{u}_n}$$
$$= \sum_{n=1}^{N} \mathbf{s}^t[n] \nabla_{\mathbf{u}_n} L(\Theta, \mathcal{D}(\mathbf{s}^t)) \tag{3.7}$$

虽然由于在联邦学习中只能获得关于模型参数的梯度，不能直接取得模型关于第一个全连接层输出结果的梯度，但是我们可以通过优化不断拟合，进而取得它。我们可以随机初始化一个批量假梯度，经过不断优化获得如下损失函数的最小值，我们可以不断降低假梯度与真实梯度间的均方误差，恢复模型关于第一个全连接层输出结果的梯度：

$$F_1(V; s_i) = ||V^T s_i - \nabla_{b_1} L(\theta, D(s_i))||_2^2 \tag{3.8}$$

在纵向联邦学习的灾难性数据泄露方法中，第一步是必不可少的，因为在这一步我们将第一个全连接层的关于输入图像的梯度从完整的用于服务器聚合的梯度中分离开来，这直接影响了该攻击方法之后的效果。

### 3.2.2 恢复输入到第一个全连接层的信息

在该算法的第二步里，我们可以利用链式法则得到，第一个全连接层的梯度等于对第一个全连接层的输入信息乘以模型关于第一个全连接层对应输出信息的梯度，具体计算方法如下：

$$\nabla_{\Theta_1} L(\Theta, \mathcal{D}) = \mathbf{H}^{\top} \nabla_{\mathbf{U}} L(\Theta, \mathcal{D}) \tag{3.9}$$

由上述计算方法，我们初始化一个批量的假输入令其为 $\hat{\mathbf{H}} = \left[\hat{\mathrm{h}}_1, \hat{\mathrm{h}}_2, ..., \hat{\mathrm{h}}_n, ..., \hat{\mathrm{h}}_N\right]^{\mathrm{T}}$，在第一步中我们已经获得了模型关于第一个全连接层输出信息的梯度，在各个客户端上传的梯度数据中我们也得到了模型第一个全连接层的更新梯度，将二者结合起来，我们可以通过最小化如下损失函数结果来恢复这一批对第一个全连接层输入的信息：

$$F_2\left(\hat{\mathrm{H}}; \mathrm{s}_i\right) = || \sum_{n=1}^{N} \mathrm{s}_i[n] \hat{\mathrm{h}}_n (\mathrm{v}_n^*)^{\mathrm{T}} - \nabla_{\theta_1} L(\theta, D(\mathrm{s}_i)) ||^2 \tag{3.10}$$

通过前两个步骤，私有数据中的部分信息已经泄露，通过第三步我们可以恢复出原始图像。

### 3.2.3 恢复原始数据

在该算法的第三步中，我们首先按照均匀分布初始化一个批量的假数据和假标签$\hat{\mathcal{D}}=\{\hat{\mathbf{x}}_n,\hat{y}_n\}_{n=1}^{N}$，依据公式(3.5)我们可以得到$\tilde{\mathbf{h}}_n=h(\Theta_c,\hat{\mathbf{x}}_n)$。在获得了模型对于输入数据的完整梯度以及对于第一个全连接层的假输入$H'$后，最后一步中需要通过优化获得最小值结果的目标函数如下：

$$F_3\left(\hat{D};\mathrm{s}_i\right)=\alpha||\nabla_\theta L(\theta,D(\mathrm{s}_i))-\nabla_\theta L\left(\theta,\hat{D}(\mathrm{s}_i)\right)||_2^2+\beta\mathrm{TV}_\xi\left(\hat{X}(\mathrm{s}_i)\right)+\gamma\sum_{n=1}^{N}||\mathrm{s}_i[n]\left(\hat{\mathbf{H}}_n^*-\tilde{\mathbf{h}}_n\right)||_2^2 \tag{3.11}$$

在该函数中，$\alpha$、$\beta$以及$\gamma$都是系数，在该损失函数中，第一项代表的是$l_2$范数，通过利用模型关于假数据集的梯度和关于真实数据集的梯度计算$l_2$范数，我们可以获得由深度梯度泄露算法恢复的真实数据集，第二项的 TV 范数代表着截断总变差范数，TV 范数主要用于图像领域中图像的降噪。这一方法的应用基于这样的假设：图像中细节过多的区域和假细节的区域多具有高 TV 值。因而通过最小化 TV 范数，可以在消除噪声同时保存图像的边缘。总的来说，噪声和图像边缘细节的 TV 值存在差异，因而利用 TV 范数可将其划分。

表（2）CAFE 数据推断攻击算法伪代码

**算法 1**：CAFE 数据推断攻击算法

1：获得模型参数$\Theta$，初始化$\mathrm{V}=\nabla_{\mathrm{U}}L(\Theta,\mathrm{D})$,对第一个 FC 层的假输入$\hat{\mathrm{H}}$,假数据集$\hat{\mathrm{D}}$

2：**for** $t=1,2,...,T$ **do**

3：　服务器随机选择 **batch** 编号$\mathrm{s}^t$

4：　服务器向所有客户端广播$\Theta$和$\mathrm{s}^t$

5：　**for** $m=1,2,...,M$ **do**

6:　　客户端$m$选取真实数据进行训练，得到$\nabla_\Theta L(\Theta,\mathrm{D}(\mathrm{s}^t))$

7:　　客户端$m$上传梯度$\nabla_\Theta L(\Theta,\mathrm{D}(\mathrm{s}^t))$

8：　**end for**

9：　服务器计算$\nabla_\Theta L(\Theta,\mathrm{D}(\mathrm{s}^t))$，包括$\nabla_{\mathrm{b}_1}L(\Theta,\mathrm{D}(\mathrm{s}^t))$和$\nabla_{\Theta_1}L(\Theta,\mathrm{D}(\mathrm{s}^t))$

10：　服务器计算$F_1(V;s^t)$，并通过$\nabla_{\mathrm{V}}F_1(V;s^t)$优化 V

11：　服务器计算$F_2\left(\hat{\mathrm{H}};s^t\right)$，并通过$\nabla_{\hat{\mathrm{H}}}F_2\left(\hat{\mathrm{H}};s^t\right)$优化$\hat{\mathrm{H}}$

12：　服务器计算$F_3\left(\hat{\mathrm{D}};s^t\right)$，并通过$\nabla_{\hat{\mathrm{D}}}F_3\left(\hat{\mathrm{D}};s^t\right)$优化$\hat{\mathrm{D}}$

13：**end for**

# 第四章 针对纵向联邦学习数据泄露的防御方法

具体的来说，我们在纵向联邦学习中定义了一种新式聚合规则，首先，众所周知，模型的更新梯度可以被视作一个向量，他的两个重要属性就是方向和大小，无论是服务器作为攻击者还是客户端作为攻击者都可以操纵其恶意梯度更新的方向和大小，因此，在计算全局模型更新的时候，我们的聚合规则同时考虑了方向和大小，并利用余弦相似度衡量更新的梯度之间的相似性以选择本规则中梯度更新的策略。

## 4.1 防御目标

在本节中，我们将会介绍我们设计的基于余弦相似度和伪梯度的联邦学习数据泄露防御方法。我们的防御目标是设计一种联邦学习聚合规则方法，在不牺牲准确率的情况下，实现针对企图从服务器恢复客户端私有数据这一行为的遏制。值得一提的是，我们将无攻击情况下的联邦学习作为讨论保真度和效率的基准，即我们的方法应该对恶意攻击具有鲁棒性，同时与无攻击情况下的联邦学习训练过程一样准确和高效。

首先，我们要遏制现有的通过梯度恢复出客户端数据集内隐私数据的攻击方法，对绝大多数的客户端隐私数据进行保护。我们希望通过我们提出的防御方使攻击者恢复出的图像完全失真或分布大量噪点，这样攻击者就不能直观的获取客户端的信息，攻击方法被无效化。

其次，在没有攻击的情况下，提出的防御方法不应该损失全局模型或是客户端本地模型的分类精确度，联邦学习训练后的模型精确度应当与以 FedAvg 为学习策略训练出的模型精确度相同，FedAvg 是一种在非对抗性环境中十分流行的联邦学习聚合规则。

最后，我们希望尽可能保持 FedAvg 聚合规则下的通信开销与训练效率，更希望在拥有同等算力的设备上，可以维持使用原有训练方法达到一定精确度时所花费的时间不变，但不可否认的是，在向原有的学习任务中添加额外的计算任务后，即使能够保证模型精确度，计算消耗的时间一定会变多。

## 4.2 基于余弦相似度和伪梯度的数据泄露防御方法

我们的新聚合规则考虑了本地模型更新和服务器模型更新的方向和大小，以计算全局模型更新。在第一步中，客户端会接收到服务器发送来的全局模型参数，在此基础上所有客户端会利用自己拥有的数据集进行训练，并更新本地模型参数。到目前为止，客户端需要做的工作和 FedAvg 聚合规则下的工作相同，接下来，客户端将会对自己上传的梯度进行操作。

在横向联邦学习的训练过程中，攻击者可以操纵恶意客户端上本地模型更新的方向，从而将全局模型的训练梯度更新到攻击者希望的任意方向。但任何一件事物都具有两面性，我们同样可以利用梯度的方向，让其作为纵向联邦学习中客户端保护自身隐私的工具。在第二步中，所有客户端会计算此轮本地客户端更新梯度和服务器端聚合所有上一轮客户端上传的梯度后发送回的全局模型梯度的余弦相似度，这里计算的余弦相似度直接反映了本轮客户端更新梯度和上一轮服务器聚合后的全局模型更新梯度的角度差异大小，余弦相似度的范围在-1 到 1 之间，对应的角度是从 180 度到 0 度，根据每一轮计算得到的余弦相似度，我们可以得到客户端更新梯度和全局模型更新梯度的方向差距，参考二者梯度向量角度的大小，我们可以选用不同的策略对接下来客户端上传的更新梯度进行操作，生成伪梯度防止攻击者通过梯度恢复隐私数据。

在第三步中，我们结合了余弦相似度和随机梯度下降法做出以下假设：在整个模型利用客户端私有数据进行训练的同时，虽然每一轮模型梯度方向都不同，但是他们大抵都是朝向最优解。有了这一假设，我们可以将纵向联邦学习中客户端模型本轮更新的梯度与上一轮更新的梯度按某一权重组合作为本轮训练客户端上传的梯度，这样在对梯度进行了伪装的同时并不会影响训练模型收敛的速度。在第五章的实验结果中我们可以看到这一方法在成功保护客户端本地隐私的同时，甚至加快了模型收敛的速度。

上文我们提到，在第三步上传的本轮客户端更新梯度是上一轮更新的梯度与这一轮实际的更新梯度重新按权重组合计算得来，如何分配权重便要根据二者的余弦相似度制定，但值得注意的是，在进行伪梯度的计算时我们必须以本轮客户端真实的更新梯度为主体，不应让本轮更新梯度收到过大影响造成上传梯度无效化、全局模型无法收敛的现象。若设本轮为第$t$轮且联邦学习中共有$N$个客户端，本轮上传的更新梯度为$G_{up}$，本轮的客户端更新梯度为$G_n^t \ \ n \in N$，上一轮全局模型更新梯度为$G_{last}$，余弦相似度为$F$，则最终上传的伪梯度由如下方法确定：

$$G_{up}=\begin{cases}|F|\times G_i+(1-|F|)\times G_{last},\ F\in(-1,-0.5)\ or\ F\in(0.5,1)\\(1-|F|)\times G_i+|F|\times G_{last},\ F\in(-0.5,0.5)\end{cases} \tag{4.1}$$

表（3）伪梯度防御算法伪代码

**算法 2**：伪梯度防御算法

1：获得模型参数$\Theta$，初始化$\mathrm{V}=\nabla_{\mathrm{U}}L(\Theta,\mathrm{D})$,对第一个 FC 层的假输入$\hat{\mathrm{H}}$,假数据集$\hat{\mathrm{D}}$

2：**for** $t=1,2,...,T$ **do**

3：　服务器随机选择 **batch** 编号$\mathrm{s}^t$

4：　服务器向所有客户端广播$\Theta$和$\mathrm{s}^t$

5：　**for** $m=1,2,...,M$ **do**

6:　　客户端$m$选取真实数据进行训练，根据公式(4.1)得到$G_{up}=\nabla_{\Theta}L(\Theta,\mathrm{D}(\mathrm{s}^t))$

7:　　客户端$m$上传梯度$G_{up}=\nabla_{\Theta}L(\Theta,\mathrm{D}(\mathrm{s}^t))$

8：　**end for**

9：　服务器计算$\nabla_{\Theta}L(\Theta,\mathrm{D}(\mathrm{s}^t))$，更新全局模型

10：**end for**

## 4.3 基于余弦相似度和梯度稀疏化的数据泄露防御方法

我们提出的第二种防御方法与第一种类似，他们均利用了余弦相似度来判别梯度之间的方向进而指定防御策略，但是与第一种方法使用的伪梯度不同，第二种方法的核心思想是利用了余弦相似度去制定梯度稀疏化[36,37]的程度。在该方法的第一步中，我们依旧采取了和 FedAvg 相同的策略，客户端利用自己拥有的数据集进行训练得到模型关于本地数据集的梯度，之后客户端将会在第二步进行更多操作。

在第二步中，客户端将会计算本次更新梯度与服务器发回的上一轮的全局模型更新梯度之间的余弦相似度，根据余弦相似度客户端会了解到两个梯度之间的角度，之后便可以有的放矢地选择对本次上传梯度的稀疏化程度。

这种防御方法的第三步，就是客户端参考余弦相似度选择本轮更新梯度稀疏化的程度，在过往的研究成果中曾有研究者选择对没有超过一定阈值大小的更新梯度进行本地积累，直到其足够大后被上传到服务器。而我们选择的策略是计算所有客户端本轮余弦相似度的平均值，若得到的结果趋近于 1，那么说明更新的梯度方向与上一轮的更新基本一致，客户端可以信任服务器，梯度稀疏化的比重可以减

小，且保留一定比例较小的梯度在本地。若得到的结果趋近于-1，那么说明本轮更新的梯度方向与上一轮角度差异较大，这时客户端一方面可以检查是否自身在学习过程中出现了问题，另一方面它也可以对服务器的安全性产生质疑，此时梯度稀疏化的比重需要增加，在本地保留一部分较大的梯度，防止服务器端的恶意攻击者进行操作，导致本地隐私数据泄露。

在我们的方法中，我们对全连接层的梯度模长进行排序，根据余弦相似度选择梯度稀疏化的程度。但是针对我们方法的第三步，我们同样可以利用所有客户端的余弦相似度等数据进行排序。除此之外，梯度压缩同样可以被应用于此方法，通过误差补偿技术，梯度可以压缩 300 倍以上而不会损失精度。在这种条件下，梯度稀疏度超过99%，远远大于深度数据泄露方法对梯度稀疏化的最大容忍度（约20%）。这表明，压缩梯度也是避免深层渗漏的一种切实可行的方法。若设本轮为第$t$轮且联邦学习中共有$N$个客户端，本轮更新梯度为$G_{up}$，本轮的客户端更新梯度为$G_n^t\ n\in N$，与客户端更新梯度相同规格的零梯度为$G_0$，平均余弦相似度为$F_{avg}$，本方法每一轮客户端最终上传的梯度将由以下方法决定：

$$G_{up}=\begin{cases}75\%\max(G_n^t)+25\%G_0,\ F_{avg}\in(0,1)\ \&\ n\in N\\ 62.5\%\min(G_n^t)+37.5\%G_0,\ F_{avg}\in(-1,0)\ \&\ n\in N\end{cases} \tag{4.2}$$

表（4）伪梯度防御算法伪代码

**算法 3**：梯度稀疏化防御算法

1：获得模型参数$\Theta$，初始化$\mathrm{V}=\nabla_{\mathrm{U}}L(\Theta,\mathrm{D})$,对第一个 FC 层的假输入$\hat{\mathrm{H}}$,假数据集$\hat{\mathrm{D}}$

2：**for** $t=1,2,...,T$ **do**

3：　服务器随机选择 **batch** 编号$\mathrm{s}^t$

4：　服务器向所有客户端广播$\Theta$和$\mathrm{s}^t$

5：　**for** $m=1,2,...,M$ **do**

6:　　客户端$m$选取真实数据进行训练，得到$\nabla_{\Theta}L(\Theta,\mathrm{D}(\mathrm{s}^t))$，将其保留在本地

7：　**end for**

8：　根据所有客户端的梯度模长对梯度进行排序，计算平均余弦相似度$F_{avg}$

9：　根据公式(4.2)选择梯度稀疏化策略，上传选中梯度

10：　服务器计算$\nabla_{\Theta}L(\Theta,\mathrm{D}(\mathrm{s}^t))$，更新全局模型

11：**end for**

# 第五章 实验与结果

## 5.1 实验环境

### 5.1.1 实验数据集

在对攻击方法进行实验时，我们选用了一组灰度图像数据集与一组彩色图像进行实验，每一组图像数据集都包含一种该算法作者曾使用的数据集以及一种我们挑选的相同规格数据集，前者可以对该算法的数据泄露效果进行验证，后者在起到前者作用的同时对该算法的普适性进行了考察，以二者作为数据集的实验结果可以进行对照。

（1） MNIST 数据集

该数据集是机器学习研究者使用得非常多的一种数据集，他主要被用于进行图像分类任务的研究。该数据集因其庞大的规模而有助于预测分析，且允许深度学习有效地发挥其能力。该数据集包含 60000 张训练集图像和 10000 张测试集图像，格式为$28 \times 28$像素单色图像。

（2） Fashion-MNIST 数据集：

此数据集训练集和测试集的图像数量均和 MNIST 数据集相同。每个图像都是一个$28 \times 28$灰度图像，与 10 个类别的标签相关联。Fashion-MNIST 数据集旨在作为原始数据集 MNIST 数据集的直接替代品，希望被广泛用于处理机器学习算法实验。

（3） Cifar-10 数据集：

CIFAR-10 数据集由 60000 张$32 \times 32$彩色图像组成，所有图像中含有 10 个类别，每个类别包含 6000 个图像。整个数据集中有 50000 个训练图像和 10000 个测试图像。数据集被分为五个训练批次和一个测试批次，每个批次有 10000 张图像。测试批次恰好包含来自每个类别的 1000 个随机选择的图像。在训练批次内恰好含有来自每个类别的 5000 张图像。

（4） Linnaeus 5 数据集：

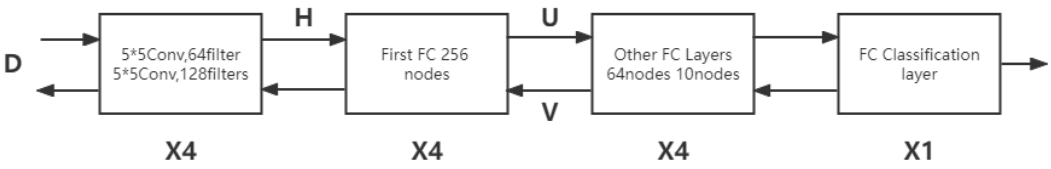
Linnaeus 5 数据集同样是一种包含了彩色图像的数据集，该数据集含有五个类别，但是相较于其他数据集 Linnaeus 含有选择更多不同的图像规格。类似于 Cifar-10 数据集，Linnaeus 5 数据集最小的图像规格是$32 \times 32$，它还拥有$64 \times 64$、$128 \times 128$、$256 \times 256$像素规格的图像。

### 5.1.2 实验硬件环境

在本文中的所有实验均在配有 Intel(R) Core(TM) i9-10980XE 3.00GHz CPU、63GB 内存以及 NVIDIA GeForce RTX 3090 显卡的 Ubuntu 21.04 (GNU/Linux 5.11.0-22-generic x86_64)系统平台上完成。

## 5.2 攻击方法实验

在对第三章中的数据泄露方法进行复现时，我们选取了两组数据集进行测试，超参数设置如附录所示。在我们的硬件条件下（RTX 3090 GPU 和 i9-10980XE CPU），CAFE 可以在纵向联邦学习训练过程中造成多达 800 个图像的泄露。实验中我们采用的神经网络结构如下图所示。



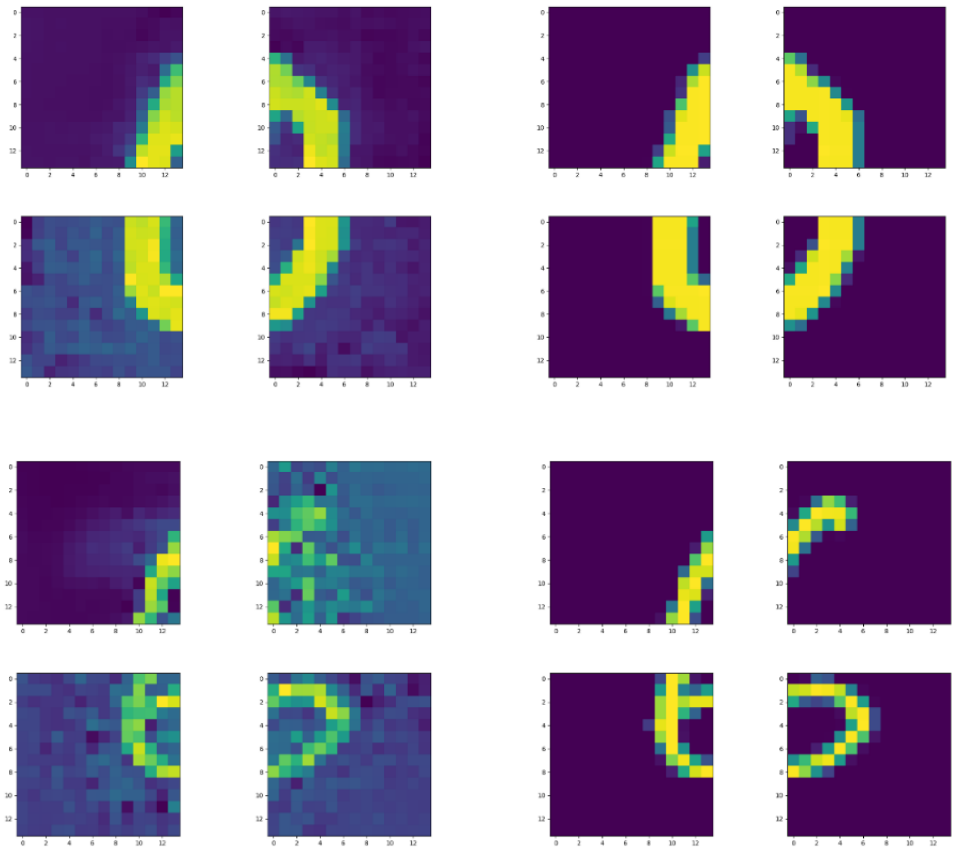

图（5） 神经网络结构

为了 CAFE 的攻击性能，我们将峰值信噪比（PSNR）值和均方误差（MSE）作为评价标准，泄漏数据的 PSNR 值越高，表示数据恢复的性能越好。我们的实验参数可以参考下表。

表（5） 攻击和防御实验的参数设置

| 学习率（lr） | 训练集图像数量 | 测试集图像数量 | Batch Size |
|---|---|---|---|
| $10^{-6}$ | 800 | 100 | 40 |

我们在“边训练边进攻”场景中实现 CAFE，在该场景中，我们持续运行纵向联邦学习过程。当模型训练时，每次迭代所选批次数据和模型参数都会发生变化，这可能导致攻击失败。然而，从下图中的实验结果来看，当学习率相对较小时，CAFE 能够恢复训练图像。而提高学习率会增加数据泄漏的难度，因为模型在每次迭代中都会做出更大的参数变化，这可以被视为一种有效的防御策略。



图（6） 在低学习率（上）与高学习率（下）时恢复出的图像与实际图像

根据我们在表（3）中在 MNIST 数据集上得到的实验结果，该模型确实以相对较小的学习率（例如：学习率为$10^{-6}$，训练集有 800 张图像，测试集有 100 张图像，批量为 40）收敛，这表明我们可以在模型收敛时成功地进行攻击。同时客户端的数据确实泄漏到了一定的水平（峰值信噪比位于 20 左右），而模型收敛到一定的精度（0.70），这也说明 CAFE 在边训练边进攻的情况下成功的泄露出了客户端一定的数据，且对模型精确度没有任何影响。

表（6） 使用 MNIST 数据集进行攻击实验时的实验结果变化

| 轮数 | PSNR | Training loss | 模型精确度 |
|---|---|---|---|
| 0 | 5.07 | 2.30 | 0.11 |
| 2000 | 11.68 | 2.26 | 0.28 |
| 6000 | 18.37 | 1.96 | 0.54 |
| 10000 | 19.92 | 1.79 | 0.64 |
| 16000 | 18.45 | 1.63 | 0.66 |
| 20000 | 17.26 | 1.72 | 0.70 |

在使用 Cifar-10 数据集进行实验时，我们使用了和在 MNIST 数据集上进行实验时相同的实验参数，在图（7）中我们可以看到，服务器恢复出的图像的 PSNR 值均稳定在 20 左右。且可以发现，图像 PSNR 值增长速度非常快，后续也存在着继续增长的趋势，实验的 PSNR 结果可以参考下表。CAFE 恢复出的 Cifar-10 数据集图像参见附录。

表（7） 使用 Cifar-10 数据集进行攻击实验时的实验结果变化

| 轮数 | PSNR | Training loss | 模型精确度 |
|---|---|---|---|
| 0 | 8.50 | 2.30 | 0.14 |
| 2000 | 6.59 | 2.27 | 0.11 |
| 6000 | 15.87 | 2.21 | 0.2 |
| 10000 | 18.80 | 2.15 | 0.25 |
| 16000 | 19.86 | 2.20 | 0.31 |
| 20000 | 19.74 | 2.19 | 0.32 |

最后，通过对上文介绍的两组不同类型数据集进行实验，我们成功证明了 CAFE 攻击方法的有效性与普适性，在对两组数据集的实验中，我们可以看到无论客户端隐私数据的类型或结构怎样，在联邦学习训练的时候利用 CAFE 进行动态攻击可以泄露大部分的客户端隐私，但是在动态攻击的过程中峰值信噪比仅可以上升到 20 左右，且恢复出的图像与原始图像相比亮度、色彩鲜艳程度均有差异。

## 5.3 防御方法实验

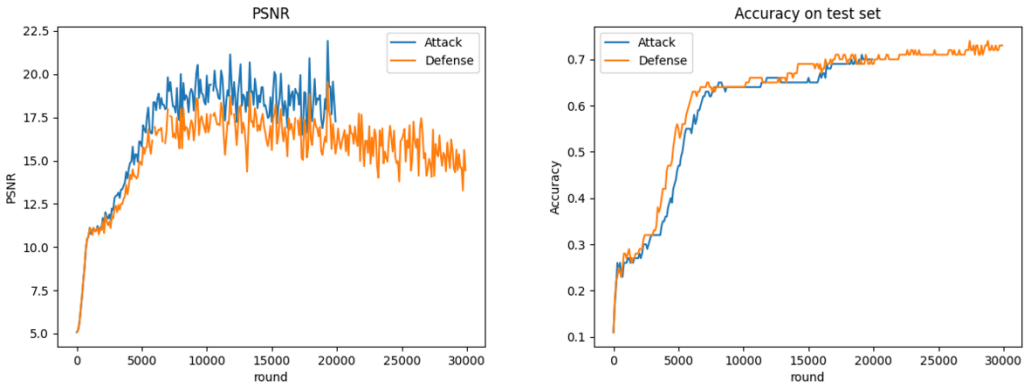
### 5.3.1 基于余弦相似度和伪梯度的防御方法实验结果

在第四章中，我们对基于余弦相似度和伪梯度的数据泄露防御方法进行了阐述，经过对 MNIST 数据集和 Cifar-10 数据集的测试后，我们成功验证了该防御方法的可行性，该方法的防御结果和真实图像的对比可以参考附录一和附录二。

首先，在原有实验超参数设置的条件下（学习率为$10^{-6}$，训练集有 800 张图像，测试集有 100 张图像，批量为 40），为了验证我们的防御方法是否对 CAFE 攻击方法持久有效，我们将攻击算法运行轮数从两万调整到了三万轮，最终得到了比较好的结果。

在使用 MNIST 数据集进行实验初期，CAFE 对图像恢复的进度很快，防御方法并没有发挥作用，仅在约一千轮的时候 PSNR 值就大约达到了 11.5，服务器希望尽可能地恢复出图像。但在此时我们的防御方法开始起了作用，以一千轮为节点，我们可以在图（7）中看到，经过利用余弦相似度和伪梯度的重构构建的防御方法，PSNR 值上升的速度明显开始降低，与此同时模型精确度上升的速度却比不添加防

御策略时更快，两种条件下全局模型精确度差距可以到达 10%，这说明我们的防御方法不但可以抑制 CAFE 攻击方法，使其恢复出的图像准确度降低，还能使模型收敛速度加快，这样在保护到客户端隐私数据时还可以完成全局模型的训练。



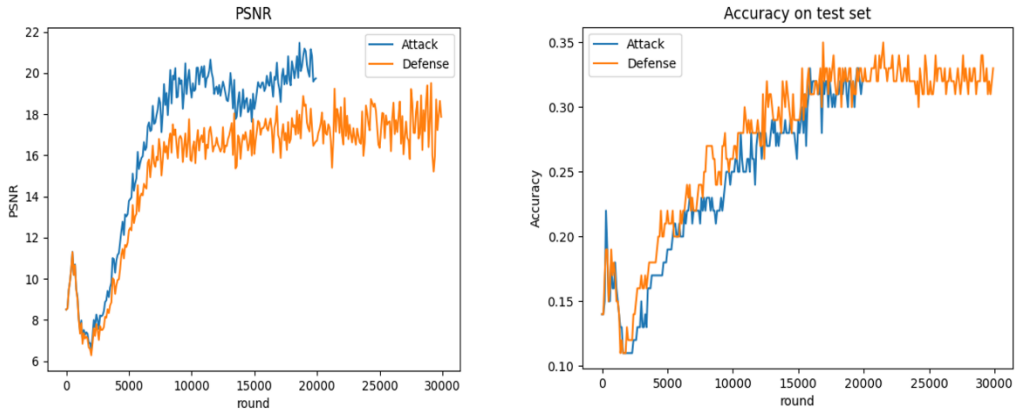

图（7） 使用 MNIST 数据集进行第一种防御实验时的实验结果变化

在训练五千轮后，PSNR 值上升速度明显降低，这说明 CAFE 攻击方法的效率到达了一个瓶颈，恢复出的图像混乱程度小于训练初期，可以从中图（7）看到，在使用我们的新聚合规则后，这一阶段的 PSNR 值明显低于原本的水平。大约在一万轮后，没有使用防御方法实验的 PSNR 结果持续上升，而使用了防御方法的结果持续下降，且在三万轮后仍有持续下降的趋势。

表（8） 使用 MNIST 数据集进行第一种防御实验时的实验结果变化

| 轮数 | PSNR | Training loss | 模型精确度 |
|---|---|---|---|
| 0 | 5.07 | 2.30 | 0.11 |
| 2000 | 11.48 | 2.24 | 0.29 |
| 6000 | 16.94 | 1.90 | 0.62 |
| 10000 | 17.69 | 1.76 | 0.64 |
| 16000 | 16.93 | 1.59 | 0.67 |
| 20000 | 14.43 | 1.58 | 0.73 |

在使用 Cifar-10 进行测试的实验中，我们可以从图（8）中看到结果有着更加明显的对比，在大约两千轮到八千轮的阶段，我们可以看到没有防御策略的情况下，CAFE 迅速恢复客户端数据，PSNR 值最终在一万轮达到了 20。而在七千轮以前，在使用了防御方法的条件下，CAFE 恢复图像的速度变慢，在七千轮左右恢复出的图像的 PSNR 值峰值仅能维持在 16 左右，但在没有防御方法的情况下，明显可见 PSNR 值有着上升的趋势，两种情况下的图像 PSNR 值之间的差距将会进一步拉大，在表（6）中我们可以直观地看到经过防御后 PSNR 值的变化趋势。




图（8）使用 Cifar-10 数据集进行第一种防御实验时的实验结果变化

在记录了全局模型精确度的图像中，我们可以清晰地看到在使用了防御方法后的全局模型无论精确度速度还是模型精确度都比在没有防御方法条件下更高。但是在使用 Cifar-10 的实验中，有防御比无防御情况下全局模型精确度最大提升仅有 5%，在两万五千轮后，使用了防御方法的全局模型精确度维持在了 35%。

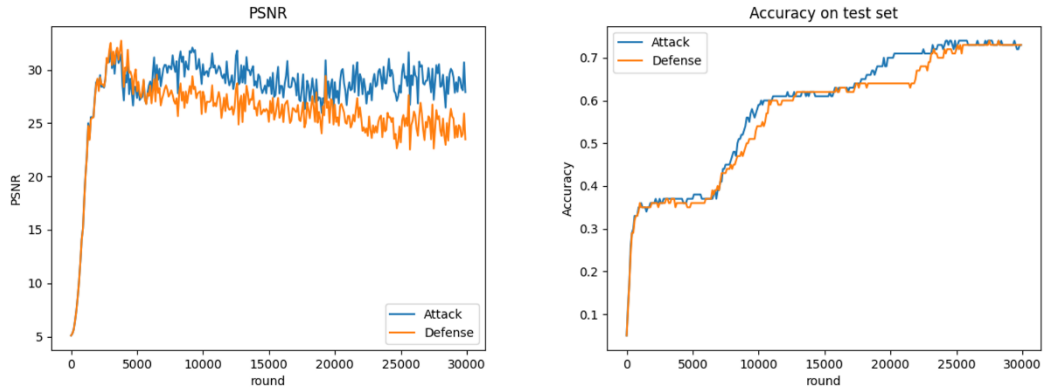
表（9） 使用 Cifar-10 数据集进行第一种防御实验时的实验结果变化

| 轮数 | PSNR | Training loss | 模型精确度 |
|---|---|---|---|
| 0 | 8.50 | 2.30 | 0.14 |
| 2000 | 6.28 | 2.26 | 0.12 |
| 6000 | 14.14 | 2.19 | 0.22 |
| 10000 | 15.72 | 2.12 | 0.26 |
| 16000 | 17.32 | 2.18 | 0.31 |
| 20000 | 17.22 | 2.11 | 0.33 |

### 5.3.2 基于余弦相似度和梯度稀疏化的防御方法实验结果

针对使用了梯度稀疏化的防御方法，我们将客户端数量增加到了 16，这是防止在纵向联邦学习过程中对较少量的客户端使用梯度稀疏化造成大量梯度被舍弃而无法使全局模型收敛的情况。该方法的防御结果和真实图像的对比可以参考附录三。

通过图（9）我们可以看出，与前一种防御方法类似，在联邦学习的初期，尤其是前三千轮左右，服务器恢复出的图像的 PSNR 值迅速升高，并且在一段时间内超过 30。值得注意的是，在图像 PSNR 值迅速上升的这段时间内，PSNR 值上升的速度比在前一种实验条件下的速度快得多，但是在训练五千轮后，PSNR 值开始快速下降。




图（9）使用 16 个客户端进行第二种防御实验时的实验结果变化

在第五千轮左右，PSNR 值下降到了 26 左右，这时的图像仍然较为模糊。到这一阶段我们可以看到基于梯度稀疏化的防御方法对客户端的隐私保护起到了初步防御作用。随后，PSNR 值下降的速度趋于平缓，但是仍在以一定的速度稳定下降，在两万五千轮左右 PSNR 值达到记录中的最低点，约为 23，但是在实际实验中，我们可以观测到训练中的很多轮 PSNR 值都达到了 22 左右。同时在两万五千轮左右，我们可以看到之后的 PSNR 值虽然仍然存在波动，但是波动幅度逐渐变小，曲线趋于平缓。

表（10） 使用 MNIST 数据集进行第二种防御实验时的实验结果变化

| 轮数 | PSNR | Training loss | 模型精确度 |
|---|---|---|---|
| 0 | 5.09 | 2.30 | 0.05 |
| 2000 | 29.17 | 2.28 | 0.36 |
| 6000 | 28.52 | 2.16 | 0.37 |
| 10000 | 27.95 | 2.01 | 0.54 |
| 16000 | 26.76 | 1.81 | 0.62 |
| 20000 | 23.47 | 1.72 | 0.73 |

此外，通过图（9）我们可以看到，全局模型的精确度在前一千轮左右迅速上升，但是在一千轮到六千轮这一阶段一直保持持平的状态，在第六千轮到一千一百轮全局模型的精确度开始稳步上升，大约从 35%上升到了 60%左右，但是上升速度相较于之前有所减缓。随后，全局模型的精确度再一次开始了持平的状态，最后在两万两千论左右，模型精确度开始了第三次的上升，这一次精确度从 60%上升到了 73%，防御结果与真实图像的对比参见附录三。

### 5.3.3 实验结果分析

经过实验，我们可以看到两种防御方法均可对 CAFE 攻击方法产生影响，且不会影响全局模型精确度。首先，在伪梯度防御方法的实验中，CAFE 攻击方法在使用了伪梯度的条件下仍然可以将恢复出的图像 PSNR 值提升到 16 左右，我们认为这个现象是由于伪梯度虽然掩盖了原始梯度，但原始梯度作为伪梯度的组成部分，其方向仍然和原始梯度方向相关，因此 CAFE 再恢复图像时虽然不能恢复出图像，但有时可以匹配到图像的颜色等属性。

在一千轮左右以前，可以发现实验结果图中 PSNR 值增长速度非常快，我们认为这是由于在训练的初始阶段，服务器端生成的假数据与真实数据差距较大，CAFE 更容易通过优化获得真实数据。类似的，在一千轮后，我们可以看到虽然 PSNR 值仍在上升，但是其上升速度不再有之前那么快，同时 PSNR 值有时出现下降的情况，我们推断这是因为在图像恢复的过程中，服务器首先恢复出了一些图像的大致轮廓，在此之后 CAFE 开始针对图像细节进行优化，并且伪梯度让 CAFE 在恢复图像时获得的梯度并非真实梯度，所以虽然 PSNR 会上涨，但是其上涨的速度却会变慢，甚至会失真。在训练达到了六千轮至七千轮的时候，我们可以看到使用了防御方法的图像 PSNR 不再上升，完全到达瓶颈。

在七千轮左右以后，我们可以直观的看到，CAFE 恢复出的图像 PSNR 值呈现出了明显的下降趋势，我们对这种情况的解释是，由于伪梯度防御方法的使用，每一轮都有按照不同权重组合生成的伪梯度作为客户端的更新梯度被上传到服务器，而这些梯度又会和下一轮的客户端更新梯度组合，随着轮数的增加，上传的梯度虽然在方向上较为正确，但其长度却受到了较大的影响，也就是说当训练轮数越多，梯度的伪装程度就越高，而因为伪梯度的方向与真实梯度的方向基本无误，所以并不会影响精确度。

通过图（8）我们可以看到，在联邦学习训练过程中，使用了伪梯度防御方法后，模型收敛的速度加快，在相同轮数全局模型精确度更高，且随着轮数的增加全局模型精确度仍有上升的趋势，不会因为客户端上传伪梯度而降低。我们对此的解释是，当本地客户端训练得到新梯度时，新梯度需要和上一轮服务器端发送回的梯度进行组合，这时全局模型的梯度更新方法不再是单纯的随机梯度下降法，而是以上一轮更新梯度为参考的梯度更新方法，这时的梯度更新波动更小，因而可以更快的使全局模型收敛。而随着进一步的训练，虽然使用伪梯度防御方法后的全局模型精确度和无防御方法下的精确度达到了同等水平，但是由于伪梯度的生成过程中仍然以客户端本轮计算出的更新梯度为主体，所以全局模型的精确度不会降低。

在第二种防御方法的实验中，我们可以看到图像的 PSNR 值在训练初期上升得非常快，一度到达了 30 以上，我们认为这是我们在进行客户端数量修改时把数据集中一张$28 \times 28$的图像分割成了 16 份$7 \times 7$的图像，当图像的规格变为原本的四分之一且每个客户端每个批量内图像顺序不变时，服务器恢复出图像的难度会变得更小，同时训练初期真实图像和服务器端假图像的差距较大，因此在这一个阶段图像 PSNR 值上升得非常快，虽然我们在第一轮就进行了梯度稀疏化的处理，但是每一轮被舍弃的梯度并不一定属于同一个客户端，所以在训练初期我们的防御方法并没有对服务器恢复图像数据的行为做到非常好的遏制，仅能将初期恢复出的图像 PSNR 值尽可能地被压低。

在五千轮左右，PSNR 值开始下降，我们认为这是因为服务器端恢复出的假图像开始受到梯度稀疏化的影响，在五千轮到六千轮这段时间，服务器端的供给者希望能够恢复出图像的细节，但是总是受到客户端舍弃梯度的影响，导致目标梯度偶尔会变成 0，这时我们恢复出的图像就会变得愈发混乱。在六千轮以后，持续的梯度稀疏化让攻击者再也无法恢复出真实图像，图像的 PSNR 值以一定的速度持续下降，最后在两万五千轮左右变得更为平缓。这时的图像已经和最后的结果没有较大差异，用肉眼看已经看不出任何信息，此时的图像无法被攻击者通过优化获得原始图像。

同样在第二种实验中，我们可以看到在训练开始阶段全局模型精确度上升的很快，这是因为图像规格较小，其余条件相同的情况下联邦学习本地客户端的训练较为轻松。而在之后的五千轮内，全局模型的精确度出现了持平的情况，我们认为这是由于梯度稀疏化使每一轮都有部分梯度被舍弃，进而导致了全局模型只能收到较少量的梯度进行聚合，同样的情况出现在了第一万轮至两万轮、第两万五千轮以后等阶段，但是即使出现了精确度不变的情况，一旦一段时间中梯度总是被稀疏化的客户端开始成功上传梯度，而原本成功上传梯度的客户端的梯度被稀疏化，那么服务器端的全局模型便可以使自己的模型覆盖更多的信息，具有更高的普适性，令自己的模型精确度更高。

总的来说，如果我们以宏观角度去看全局模型精确度的三个上升阶段，我们会发现全局模型精确度的上升速度逐渐变缓，这与正常神经网络的训练过程一致。使用了梯度稀疏化的训练方法虽然保护了客户端的隐私数据，但是却会延长全局模型的训练时间，我们认为这在隐私受到保护的前提下是可以被接受的。

# 第六章 总结与展望

在本文中，我们通过一种新的算法揭示了垂直联合学习（CAFE）中灾难性数据泄漏的风险，该算法可以在高数据恢复质量和理论保证的情况下执行大批量数据泄漏。大量实验结果表明，CAFE 可以从纵向联邦学习设置的共享聚合梯度中恢复大规模私有数据，克服了当前数据泄漏攻击中的批量限制问题。此外，我们还提出了两种防御方法，他们都是以余弦相似度为度量方法，结合了两种现有的防御思路，最终成功遏制了服务器利用客户端上传的梯度进行私有数据恢复这一恶意行为。

在现有的基础上，我们可以更加深入的研究深度数据泄露方法（DLG），探索如何优化可以让这种攻击方法回复出来的图像和原始图像有着更高的相似度。我们也可以对两种防御方法进行更为严谨的证明，同时也可以对两种方法进行防御策略的优化，例如更加细化第一种防御方法创建伪梯度时的权重分配方法，使其能够更有针对性地应对不同角度的梯度。在第二种防御方法中我们也可以去尝试以何种标准进行梯度稀疏化，为了使训练任务在一定时间内达到特定的精确度我们可以使用何种误差补偿技术等问题都十分值得我们思考。

# 致谢

在文章的最后，我要感谢我的导师朱天清教授多年来的悉心指导，以及对我的信任。在过去的三年中，是老师的帮助让我从一个无知懵懂的少年立下志向并勇敢的迈开追求知识的步伐。

当然，我也要感谢父母四年来对我的一切帮助，他们对我的支持和鼓励让我变得更加自信，更加有勇气步入社会，我相信他们足以成为我的榜样。

最后，我要感谢我的室友们、同学们和大学四年陪伴我一路走来的朋友们，四年内我们相互扶持，相互鼓励，现在终要分别，感谢一路相随，愿往后的日子里所有的美好与期待都能如约而至。

如今，我也将要告别本科生活，作为一名研究生进行新阶段的学习与研究，祝愿我能够在这座英雄的城市——武汉，创下更多的辉煌。

# 参考文献

[1] ZHAO B, MOPURI K R, BILEN H. iDLG: Improved Deep Leakage from Gradients[M/OL]. arXiv, 2020[2022-05-13].

[2] GEIPING J, BAUERMEISTER H, DRÖGE H, et al. Inverting Gradients -- How easy is it to break privacy in federated learning?[J/OL]. arXiv:2003.14053 [cs], 2020[2022-04-22].

[3] WANG Y, DENG J, GUO D, et al. SAPAG: A Self-Adaptive Privacy Attack From Gradients[M/OL]. arXiv, 2020[2022-05-13].

[4] ZHU J, BLASCHKO M. R-GAP: Recursive Gradient Attack on Privacy[M/OL]. arXiv, 2021[2022-05-13].

[5] YIN H, MALLYA A, VAHDAT A, et al. See through Gradients: Image Batch Recovery via GradInversion[C/OL]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, 2021: 16332-16341[2022-04-22].

[6] ZHU L, LIU Z, HAN S. Deep Leakage from Gradients[J/OL]. arXiv:1906.08935 [cs, stat], 2019[2022-04-22].

[7] WU Y, CAI S, XIAO X, et al. Privacy Preserving Vertical Federated Learning for Tree-based Models[J/OL]. Proceedings of the VLDB Endowment, 2020, 13(12): 2090-2103.

[8] XIA J, HUANG W, MA Z, et al. Gradient-Based Differential Privacy Optimizer for Deep Learning Model Using Collaborative Training Mode[C/OL]//2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT). Dalian, China: IEEE, 2019: 208-215[2022-05-13].

[9] NASR M, SHOKRI R, HOUMANSADR A. Improving Deep Learning with Differential Privacy using Gradient Encoding and Denoising[M/OL]. arXiv, 2020[2022-05-13].

[10] O'SHEA K, NASH R. An Introduction to Convolutional Neural Networks[M/OL]. arXiv, 2015[2022-05-17].

[11] MIIKKULAINEN R, LIANG J, MEYERSON E, et al. Evolving Deep Neural Networks[M/OL]. arXiv, 2017[2022-05-17]. http://arxiv.org/abs/1703.00548.

[12] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J/OL]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

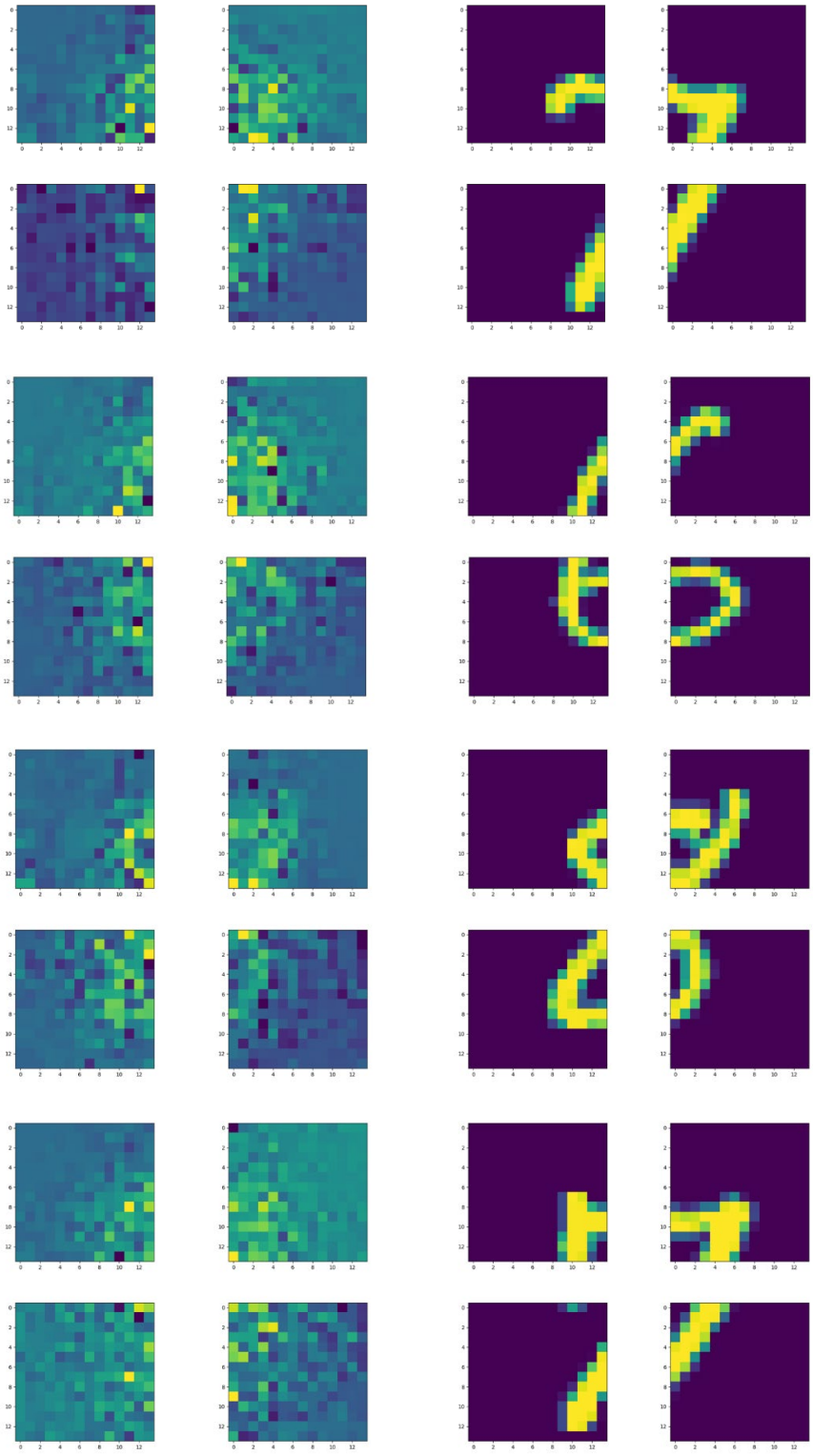
[13] 段金成, 张风霞, 朱晓庆. 基于卷积神经网络的人脸识别算法研究[J]. 科学技术创新, 2022(10): 73-76.

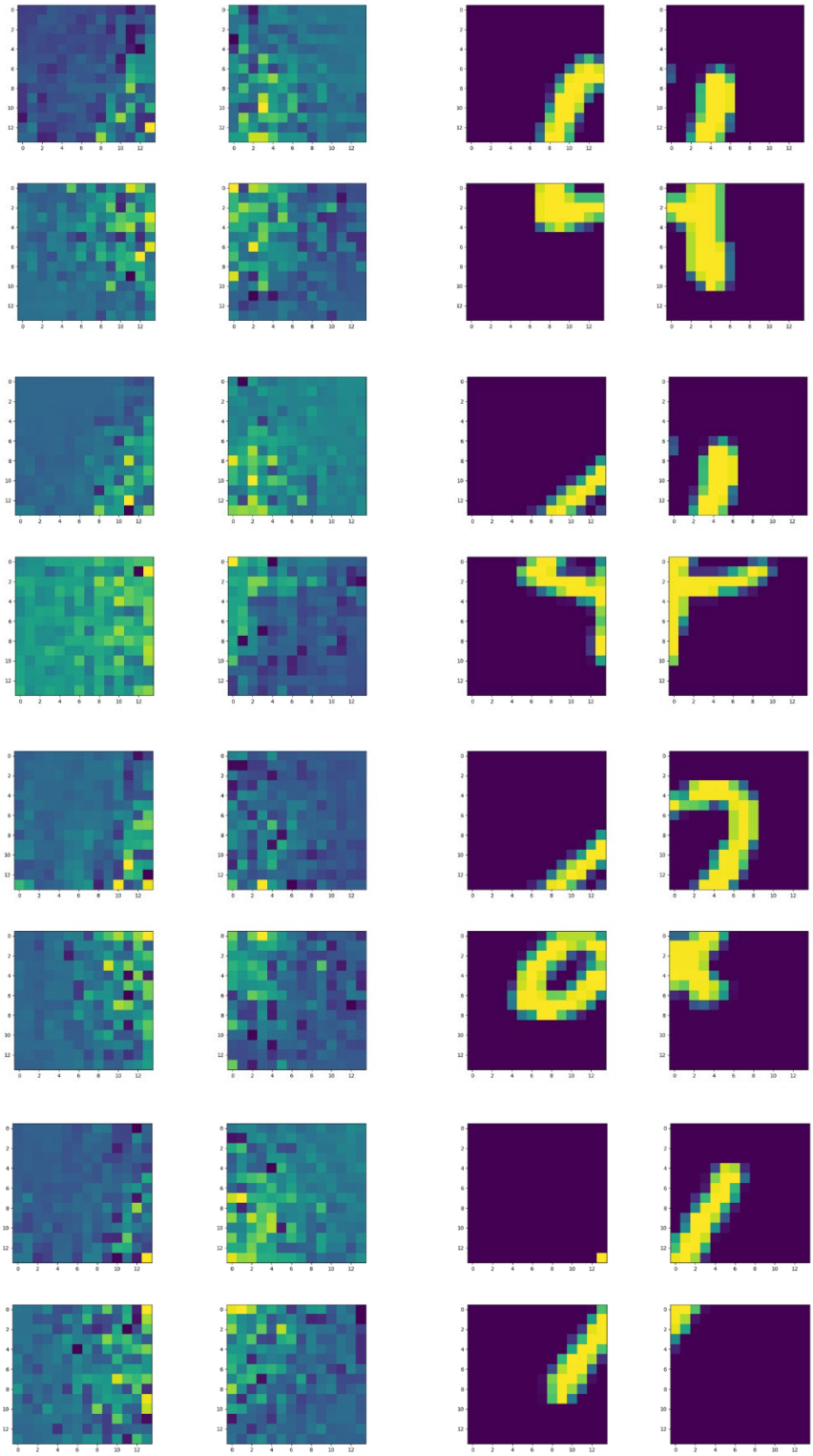
[14] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J/OL]. Communications of the ACM, 2017, 60(6): 84-90.

[15]SMITH S, PATWARY M, NORICK B, et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model[J]. 44.
[16]Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond[J/OL]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(2): 113-120.
[17]DEAN J, CORRADO G, MONGA R, et al. Large Scale Distributed Deep Networks[J]. 9.
[18]ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems[M/OL]. arXiv, 2016[2022-05-13].
[19]IANDOLA F N, MOSKEWICZ M W, ASHRAF K, et al. FireCaffe: Near-Linear Acceleration of Deep Neural Network Training on Compute Clusters[C/OL]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016: 2592-2600[2022-05-13].
[20]KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated Learning: Strategies for Improving Communication Efficiency[M/OL]. arXiv, 2017[2022-05-13].
[21]MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data[J]. 10.
[22]YANG Q, LIU Y, CHEN T, et al. Federated Machine Learning: Concept and Applications[M/OL]. arXiv, 2019[2022-05-13].
[23]LI T, SAHU A K, TALWALKAR A, et al. Federated Learning: Challenges, Methods, and Future Directions[J/OL]. IEEE Signal Processing Magazine, 2020, 37(3): 50-60.
[24]刘艺璇, 陈红, 刘宇涵, 等. 联邦学习中的隐私保护技术[J]. 软件学报, 2022, 33(03): 1057-1092.
[25]FENG S, YU H. Multi-Participant Multi-Class Vertical Federated Learning[M/OL]. arXiv, 2020[2022-05-13].
[26]杨东宁, 谢潇睿, 吉志坤, 等. 一种隐私保护的联邦学习框架[J]. 电子技术应用, 2022, 48(05): 94-97+103.
[27]温佳琳, 钱海峰. 回归模型上的数据中毒攻击与防御[D]. 华东师范大学, 2021.
[28]尹虹舒, 周旭华, 周文君. 纵向联邦线性模型在线推理过程成员推断攻击的隐私保护研究[J]. 大数据: 1-13.
[29]BONAWITZ K, IVANOV V, KREUTER B, et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning[C/OL]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas Texas USA: ACM, 2017: 1175-1191[2022-05-17].
[30]陈晋音, 邹健飞, 苏蒙蒙, 等. 深度学习模型的中毒攻击与防御综述[J]. 信息安全学报, 2020, 5(04): 14-29.
[31]BAGDASARYAN E, VEIT A, HUA Y, et al. How To Backdoor Federated Learning[J]. 10.
[32]BHAGOJI A N, CHAKRABORTY S, MITTAL P, et al. Analyzing Federated Learning through an Adversarial Lens[J]. 10.
[33]JIN X, CHEN P Y, HSU C Y, et al. CAFE: Catastrophic Data Leakage in Vertical Federated Learning[J]. 21.

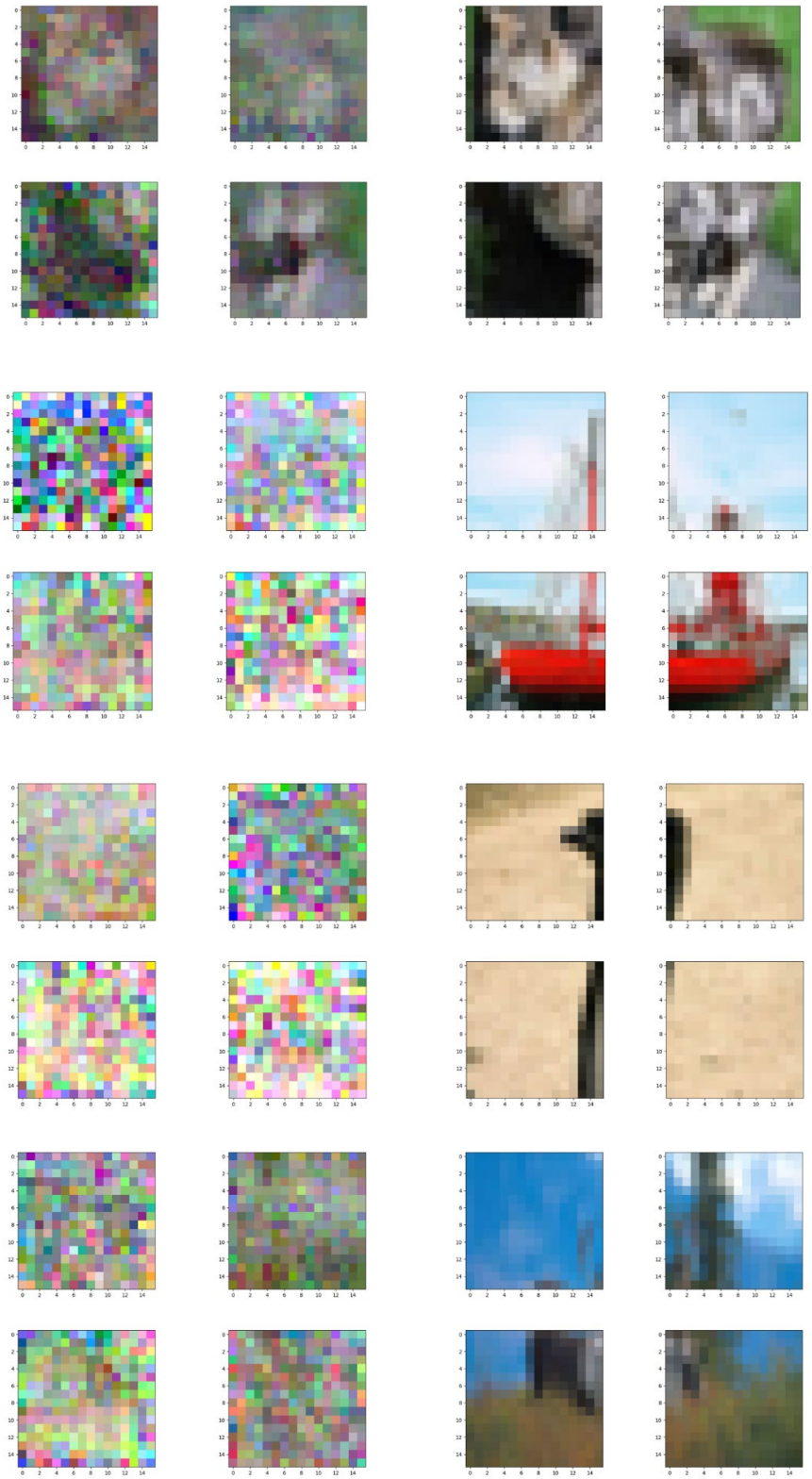
[34]BONAWITZ K, IVANOV V, KREUTER B, et al. Practical Secure Aggregation for Federated Learning on User-Held Data[M/OL]. arXiv, 2016[2022-05-13].
[35]PHONG L T, AONO Y, HAYASHI T, et al. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption[J]. 18.
[36]李诗琪, 戚琦. 分布式深度学习模型训练中梯度稀疏方法的改进[D]. 北京邮电大学, 2021.
[37]罗鹏, 陈剑勇. 基于梯度压缩的深度学习算法分布式计算研究[D]. 深圳大学, 2020.

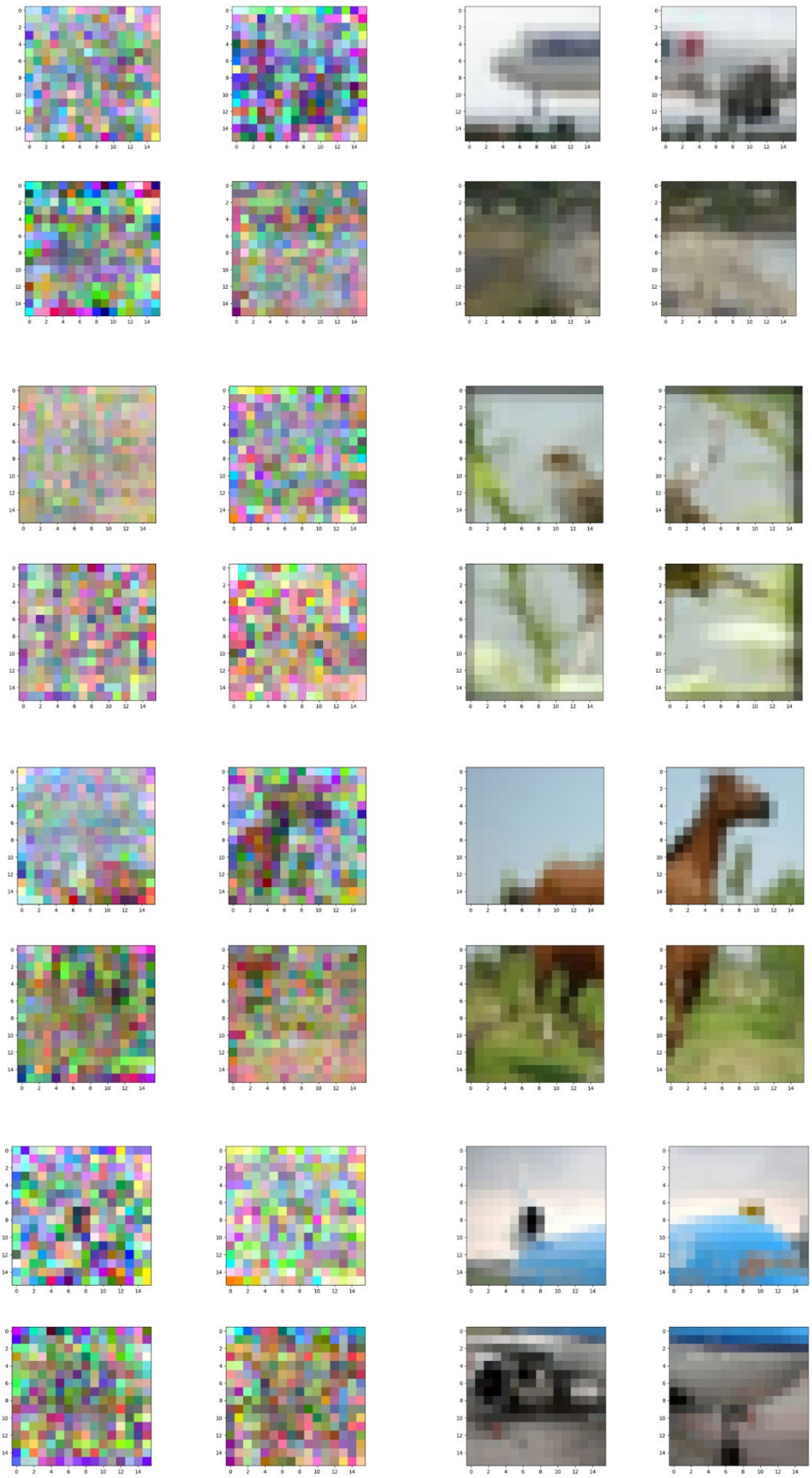
# 附录一





# 附录二





# 附录三