基于模型投毒在大规模联邦学习中攻击隐私方法

陈根文[1)]

[1)](东南大学网络空间安全学院, 南京, 中国, 211189)

摘 要 联邦学习（Federated Learning, FL）是一种保护本地数据隐私的分布式学习范式，但近期研究表明，攻击者可通过梯度反演攻击泄露私有数据。设置较大的本地训练批量大小可以有效抵御这些攻击，但现有改进方法在应对大批量训练时通常局限于特定模型架构。为此，我们提出了一种基于模型投毒的梯度反演攻击方法，能够在大批量FL中破解隐私。针对全连接神经网络（FCNN），通过分析第一个全连接层的梯度，能够精确恢复私有数据。实验结果表明，所提方法在四个数据集上均表现出色，尤其在大批量训练（如批量大小为64、128、256）及多种激活函数（ReLU、Sigmoid、Tanh）下，均优于现有方法，有效揭示了FL中潜在的隐私风险。

关键词 联邦学习；数据隐私；模型投毒；梯度反演攻击；全连接神经网络

Attack privacy methods in large-batch Federated Learning based on model poisoning poisoning

CHEN Gen-Wen[1)]

[1)](School of Cyber Science and Engineering, Southeast University, City Nanjing, China)

Abstract

Federated Learning (FL) is a distributed learning paradigm that protects local data privacy, but recent studies have shown that adversaries can reveal private data through gradient inversion attacks. While increasing the local training batch size can effectively defend against such attacks, existing countermeasures are typically limited to specific model architectures. To address this, we propose a novel gradient inversion attack based on model poisoning, which compromises privacy in large-batch FL. For Fully Connected Neural Networks (FCNNs), private data can be precisely recovered by analyzing the gradients of the first fully connected layer. Experimental results show that our method outperforms existing approaches across four datasets, especially in large-batch training (e.g., batch sizes of 64, 128, 256) and with different activation functions (ReLU, Sigmoid, and Tanh). This highlights the potential privacy risks in FL, demonstrating the effectiveness of our attack strategy.

Keywords Federated Learning; Data Privacy; Gradient inversion attack; Model poisoning; Fully Connected Neural Networks

1 介绍

人工智能的快速发展导致了深度神经网络（dnn）在物联网（IoT）中的广泛应用。物联网设备能够收集各种类型的多媒体数据，如音频、图像和视频，可用于改进基于dnn的物联网应用。然而，物联网设备收集的数据通常包含敏感信息，特别是在医院和政府中。将原始资料直接传送至伺服器（例如服务提供者）会导致严重的私隐外泄。采用联邦学习（FL）[21,16,34,2]是解决物联网隐私问题的一种流行方法。FL是一种分布式学习范例，其中中央服务器协调客户端（例如物联网设备）来训练全局模型，而无需访问本地数据。

具体来说，客户端在本地数据上训练模型，并与中央服务器共享梯度。然后，中央服务器计算平均梯度并将其返回给客户端，客户端使用新的梯度更新其本地模型。

FL的隐私保护是基于共享梯度暴露很少隐私的假设。然而，最近的研究推翻了这一假设，并证明服务器可以从共享梯度中泄露客户端的隐私。一般来说，FL中的隐私攻击可以分为三类：隶属度推理[27,22]、属性推理[9,22,19]和梯度反演[26,12,39,37,10,30,35,14]。

梯度反演被认为是三种隐私攻击中最强大的一种。它涉及一个诚实但好奇的服务器，它可以通过反转共享梯度来重建客户端的本地数据。该攻击首先由Hitaj等人 [1]提出，基于生成式对抗网络（GAN） [2]。攻击者利用共享梯度来训练GAN，该GAN可以生成受害者私人数据的原型样本。Zhu等人 [3]通过将其表述为优化问题，证明了从共享梯度中恢复原始列车集的可行性。随后，许多研究工作[37,10,35,14]进行了改进这种基于优化的攻击。

而在局部训练中设置较大的批大小可以有效防御梯度反转攻击。[39,10,13]中的实验表明，使用较大的批处理（例如48,64,128）可以有效地削弱这些梯度反转攻击。其主要思想是，大的批处理会加剧聚合梯度的混乱，并减少梯度与单个训练数据之间的关联。这导致通过解决 [3]中提出的优化问题来恢复原始数据的难度增加。尽管最近的研究[8,5]通过操纵全局模型的架构或参数成功地破坏了大批量FL中的隐私，但它们仍然存在一些局限性： [4]中提出的方法需要在全局模型中存在特殊的“印记模块”，这在现实世界中是不可行的。另外，[8,5]中的方法没有考虑使用其他激活函数的模型，如Tanh和Sigmoid。

为了克服这些挑战，我们提出了一种基于模型中毒的梯度反演攻击方法。服务器首先构造恶意模型参数，然后毒害客户端的本地模型。这些恶意参数可以有效地缓解大批量训练中聚合梯度的混淆，放大FC层的直接数据泄漏。通过反转在中毒模型上计算的共享梯度，服务器可以成功地重建客户端的私有数据集。具体来说，通过分析FCNNs中第一层FC的梯度，可以很好地提取出私有数据。

本工作的主要贡献有三个方面：

(1)分析了单个数据训练和小批量训练场景下FC层的直接数据泄漏。我们的研究结果表明，在大批量训练中，隐私性无法得到保证，因为FCNNs的输入数据可以通过分析特定神经元的梯度来完美地提取。

(2)在分析直接数据泄漏的基础上，提出了一种构造恶意参数的新方法来减轻聚合梯度中的混淆。通过模型中毒，我们成功地破坏了大批量FL中的隐私。

(3)在不同场景下用4个数据集进行实验，验证了提出的算法在大批量FL中的有效性。

2 相关工作

2.1 联邦学习

联邦学习[21,16,34]是一种分布式学习范式，近年来受到了广泛关注。参与的客户通过共享梯度而不是私有局部数据的隐私保护方式合作训练全局模型。中央服务器通过迭代地聚合客户端的梯度来协调训练过程。在每一轮k中，客户端用局部训练集(X_i, Y_i)训练全局模型，并共享它们的梯度。然后，服务器计算平均梯度为：$\frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \mathcal{L}_{\theta^k}(X_i, Y_i)$，其中$N$是参与方数量。最新的平均梯度被发送回各个客户端，他们可以根据等式(1)更新局部模型。这个过程被称为联邦SGD，将持续许多轮，直到全局模型收敛。

$$\theta^{k+1} = \theta^k - \tau\frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \mathcal{L}_{\theta^k}(X_i, Y_i) \qquad (1)$$

2.2 梯度反演攻击

2.2.1 基于GAN的方法

Hitaj等人首先提出了一种基于GAN的方法来破坏FL中的隐私。GAN过程将判别性深度学习网络D与生成性深度学习网络G进行比较，他们使用FL中的全局模型作为网络D来帮助网络G生成受害者的数据。Wang等人提出mGAN-AI在服务器端“无形地”工作。它使恶意服务器能够针对任何客户机并破坏客户机级别的隐私。然而，这些基于gan的方法只能用于生成类数据，而不能用于生成特定的数据点。

2.2.2 基于优化的方法

基于优化的方法的思想是生成能够产生与真实数据相似的梯度的合成数据。从梯

度$\nabla_\theta \mathcal{L}_\theta(x,y)$恢复原始数据$(x,y)$,方法DLG [3]将其表示为等式(2)，并用L-BFGS求解器求解了该优化问题。Zhao等[37]发现可以直接从梯度中获得地真值标签（ground truth label），使得DLG更加稳定和高效。然而，DLG在带有ReLU层的神经网络上表现不佳。

$$\tilde{x}^*, \tilde{y}^* = \min_{\tilde{x},\tilde{y}} \left\| \nabla_\theta \mathcal{L}_\theta(x,y) - \nabla_\theta \mathcal{L}_\theta(\tilde{x},\tilde{y}) \right\|^2 \quad (2)$$

由于L-BFGS求解器要求模型参数是三阶可导的，而ReLU使得高阶导数不连续。为了使基于优化的方法更加通用，Geiping等 [5]使用余弦相似度计算合成梯度与真实梯度之间的距离，并用Adam求解器求解。余弦相似度可以同时计算梯度的范数和方向，提高了攻击的有效性。最近的工作[35,14]引入先验知识来约束优化问题的搜索空间，从而获得更好的结果。

2.2.3 恶意参数的方法

与前两种方法不同，恶意参数方法[8,31,5]引入了一个不诚实的服务器，它可以故意篡改发送给客户端的梯度。Fowl等人在全局模型中插入一个“imprint module”作为一个特殊的层来显示私有数据。随后，在 [6]中提出了不修改模型体系结构的改进方法。攻击者通过初始化客户端的模型参数来恢复大批量训练中的私有数据。然而，他们的方法只在使用ReLU层的模型上表现良好，而没有考虑实际的CNN模型。

2.3 模型投毒

模型中毒攻击[7,20,3,4]的目的是最小化全局模型的准确性或篡改预测标签。攻击者构建带有特定后门的恶意模型参数（例如，将带有木马触发器的图像分类为攻击者所需的标签）。与以往的方法不同，我们在这项工作中利用模型中毒来减轻客户端聚合梯度的混淆。因此，在4.3节中设计了一种特定的方法来构造恶意参数，这有助于大规模FL中的梯度反演攻击。

3 问题背景

考虑一个标准的FL协议，它涉及一个中心服务器和多个客户端，本文提出了一种新的梯度反转攻击，表示为$\mathcal{A}$。作为对手，中央服务器尝试使用$\mathcal{A}$来重构客户端的本地私有数据。将攻击方法$\mathcal{A}$得到的重构数据表示为$\tilde{x}$，由等式3得到。

$$\tilde{x} = \mathcal{A}\left(\nabla_\theta \mathcal{L}_\theta(x,y)\right) \quad (3)$$

其中$\nabla_\theta \mathcal{L}_\theta(x,y)$是客户端共享的梯度。攻击目标是最小化重构数据$\tilde{x}$与真实数据$x$的距离$L_2$，可以用等式4表示。

$$L_2 = \min \|\tilde{x} - x\|_2 \quad (4)$$

L_2距离越小，表示重构数据越接近真实数据，这意味着此攻击方法将会泄露更多的隐私。

4 方法

4.1 攻击方法

图1为攻击方法示意图。总的来说，攻击方法可以分为中毒阶段和恢复阶段。在中毒阶段，服务器构建恶意模型参数，通过发送恶意梯度$\bar{g}$来毒害客户端模型。在恢复阶段，服务器根据客户端在中毒模型上计算的共享梯度g_i重构私有训练集。

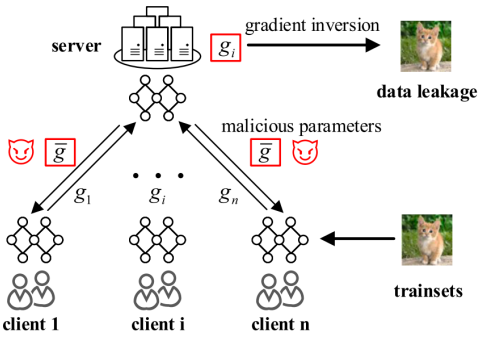

Fig. 1 攻击方法示意图

4.2 恶意模型参数构建

梯度在大批量训练中存在混淆，但仍然可以从单数据激活神经元（SDAN）中恢复数据。在正常训练的模型中通常存在少量SDAN。为了增加SDAN的数量，放大FC层的直接数据泄漏，提出了构建恶意模型参数的方法。算法1概述了构造恶意参数的细节。这是一个迭代过程，恶意参数是在模型的第一层FC层构造的。

第一层FC表示为$f(\cdot|\mathbf{W},\mathbf{B})$，$\mathbf{W} \in \mathbb{R}^{l*n}$是权重，$\mathbf{B} \in \mathbb{R}^l$是偏置。在每一轮迭代中，$f(\cdot|\mathbf{W},\mathbf{B})$的输入是一个从辅助数据集$D_{aux}$中采样的批处理数据集$X = \{x_i | x_i \in D_{aux}\}_{i=1}^N$。让$T_i$表示相对于数据点$x_i$的SDANs的神经元索引集合。方法目标是让更多的数据点在第一层FC中有SDANs，表示为$\max \left| \{T_i \mid T_i \neq \emptyset\}_{i=1}^N \right|$，其中$|\cdot|$表示集合中元素个数。为了将其最大化，使用随机梯度下降（SGD）优化恶意参数。构建恶意参数有如下：

4.2.1 参数初始化

在算法1中，首先使用 [6]中提出的算法3初始化模型参数。它从两个不同的高斯分布中采样参数。这种初始化可以进一步提高本文方法的性能。

Input: $\mathbf{W} \in \mathbb{R}^{l*n}$，$\mathbf{B} \in \mathbb{R}^{l}$：第一层FC层的权重和偏置；$f$,$k$：每个数据要选择的神经元的数量；$\eta$：学习率；$D_{aux}$辅助数据；$N$：训练批次大小；$\mu$,$\sigma$：高斯分布的均值和标准差；$s$：比例因子；$E$：训练轮次

Output: **W**,**B**：恶意模型参数

$\mathbf{W}, \mathbf{B} \leftarrow$ ParametersInitial$(\mathbf{W}, \mathbf{B}, \mu, \sigma, s)$;
for *epoch* $\leftarrow 1$ **to** E **do**
  $\mathbf{C} \leftarrow \mathbf{0}$ where $|\mathbf{C}| = l$;
  $\bar{c} \leftarrow \frac{1}{l}\sum_{i=1}^{l} c_i$, where $c_i \in C$;
  for *iteration* $\leftarrow 1$ **to** $|D_{aux}|/N$ **do**
    $\mathbf{S} \leftarrow \emptyset$;
    Sample one batch of data $X = \{x_i | x_i \in D_{\text{aux}}\}_{i=1}^{N}$;
    for $x_i \in X$ **do**
      $O(x_i) \leftarrow \text{Sigmoid}(f(x_i|\mathbf{W}, \mathbf{B}))$;
      $\mathbf{T}_i \leftarrow$ NeuronSelection$(\mathbf{O}(x_i), k, \mathbf{S}, \mathbf{C}, \bar{c})$;
      $g_i \leftarrow \frac{1}{k}\sum_{t \in T_i} \nabla_{\mathbf{W},\mathbf{B}} \mathcal{L}_{\text{SDAN}}(x_i, t|\mathbf{W}, \mathbf{B})$;
      $\mathbf{S} \leftarrow \mathbf{S} \cup \mathbf{T}_i$;
    end
    for *each* $t \in \boldsymbol{T}_i$ **do**
      $c_t \leftarrow c_t + 1$;
    end
    $\bar{c} \leftarrow \frac{1}{l}\sum_{i=1}^{l} c_i$;
    $\bar{g} \leftarrow \frac{1}{N}\sum_{i=1}^{N} g_i$;
    Update $\mathbf{W}, \mathbf{B}$ with $\eta\bar{g}$;
  end
end
return $\mathbf{W}, \mathbf{B}$

Algorithm 1: 恶意参数训练.

4.2.2 特定损失函数

为了实现确保每个数据点都拥有自己的SDANs，并且这些SDANs的索引彼此不重叠这一目标，设计损失函数SDAN-loss，如等式5所示。

$$\mathcal{L}_{\text{SDAN}}(x_i, t|\mathbf{W}, \mathbf{B}) = -\log(\text{Sigmoid}(f_t(x_i|\mathbf{W}, \mathbf{B}))) \tag{5}$$

利用下式确保数据的分号只激活t^{th}神经元，不激活其他数据点的SDANs。其中$f_j(x_i|\mathbf{W}, \mathbf{B}$是$j^{th}$神经元的输出。

$$\begin{cases} f_t(x_i|\mathbf{W}, \mathbf{B}) > 0 \\ f_j(x_i|\mathbf{W}, \mathbf{B}) \leq 0, \quad \forall j \neq t \end{cases}$$

随后，我们使用一个$Sigmoid$函数对（0,1）范围内的输出$f(x_i|\mathbf{W}, \mathbf{B})$进行缩放，可以写成等式6。

$$\mathbf{O}(x_i) = \textit{Sigmoid}(f_t(x_i|\mathbf{W}, \mathbf{B})) \in \mathbb{R}^{l} \tag{6}$$

4.2.3 多SDANs的选择

与人为分配SDANs相比，本文方法允许每个数据点在训练过程中自动选择最佳SDANs。算法2中为神经元选择方法。在这个算法中，集合**S**包含了已经被同一批的其他数据点选择为SDANs的神经元。向量$\mathbf{C} \in \mathbb{R}^{l}$记录了每个神经元在一个epoch的训练中被选择的次数。选择在$\mathbf{O}(x_i)$具有最高的值的k神经元作为SDANs的x_i,形成集合T_i。因此，最优恶意模型参数可表示为等式7。

$$\mathbf{W}^*, \mathbf{B}^* = \textit{arg}\min_{\mathbf{W},\mathbf{B}} \mathbb{E}_{x_i \sim D_{\text{aux}}} \mathbb{E}_{t \in T_i} \mathcal{L}_{\text{SDAN}}(x_i, t \mid \mathbf{W}, \mathbf{B}) \tag{7}$$

4.3 对FCNNs隐私的破坏

这一小节详细介绍对FCNNs的梯度反演攻击。攻击方法包括两个阶段，即中毒阶段和恢复阶段。

在中毒阶段，为第一FC层构造恶意参数。设θ_{fc1}表示全局模型第一层FC层的参数，θ^*_{fc1}表示构造的恶意参数。第一FC层的恶意梯度g^*_{fc1}可以用等式8表示。

$$g^*_{fc1} = \frac{\theta - x}{\tau} \tag{8}$$

其中，τ为FL的学习率，正态梯度g_{fc1}（正态平均梯度$\bar{g}$的一部分）替换为g^*_{fc1}并发送给所有客户端。客户端根据等式1用恶意g^*_{fc1}更新局部模型后，可以将第一FC层的参数篡改为等式9。在恢复阶段，服务器可以从第一FC层的SDANs梯度中恢复私有数据点。

$$\theta_{fc1} - \tau \times g^*_{fc1} = \theta_{fc1} - \tau \times \frac{\theta_{fc1} - \theta^*_{fc1}}{\tau} = \theta^*_{fc1} \tag{9}$$

5 实验

5.1 实验配置

软硬件环境：NVIDIA TESLA V100 (32 GB)，PyTorch；

Input: $\mathbf{O}(x_i)$：对数据变量x_i的计算输出；k：每个数据要选择的神经元的个数；**S**：在某次批量训练中被选择的神经元的索引|**C**：每个神经元被选择的次数；$\bar{c}$：**C**中元素的平均值.
Output: $\mathbf{T}_i$：x_i选取的神经元的索引

$T_i \leftarrow \emptyset$;
for *each* $s \in S$ **do**
  $O_s(x_i) \leftarrow 0$;
end
for $j \leftarrow 1$ *to* $|C|$ **do**
  if $c_j > \varepsilon$ **then**
    $O_j(x_i) \leftarrow 0$;
  end
end
while $k > 0$ **do**
  $t \leftarrow \arg\max_j O_j(x_i)$;
  $T_i \leftarrow T_i \cup \{t\}$;
  $O_t(x_i) \leftarrow 0$;
  $k \leftarrow k - 1$;
end
return $\mathbf{T}_i$

Algorithm 2: 神经元选择.

数据集：Fashion-MNIST [7]、CIFAR100 [8]、FaceScrub [9]和TinyImageNet200；实验：服务器使用测试集构建恶意模型参数。客户端使用一批本地私有数据在中毒模型上计算梯度。评估恢复的训练数据的质量。

(1) Fashion-MNIST，60000张1×28×28图像，10个类，50000张训练图像，10000张测试图像。

(2) CIFAR100，60000张3×32×32图像，100个类，50000张训练图像，10000张测试图像。

(3) FaceScrub，107818张3×64×64（调整大小）图像，530个类，75473张（70%）训练图像，32345张（30%）测试图像。

(4) Tiny ImageNet 200，12000张3×64×64图像，200个类，50000张训练图像，10000张测试图像。

目标模型：实验中使用的基本模型如表1所示。设置由6个FC层组成的FCNN。每个卷积层后面是一个批处理规范化层和一个ReLU层。

评价指标：本文主要考虑图像数据的恢复。因此用峰值信噪比（Peak Signal-to-Noise Ratio，PSNR）衡量恢复图像的质量，在文献[39,10]中这种方法也被使用。其中

$$\mathrm{PSNR}(\tilde{x}, x) = 10 \times \lg\left(\frac{1}{\mathrm{MSE}(\tilde{x}, x)}\right)$$

均方误差(MSE)是恢复图像$\tilde{x}$及其相应的实像x的平均$L2$区别,.

$$\mathrm{MSE} = \frac{\|\tilde{x}, x\|_2}{\dim(x)}$$

其中$dim(x)$是图像数据x的维数。实验将记录整批恢复数据的平均PSNR。如果一个恢复数据的PSNR大于100db，我们都将其设置为100db。

Table 1 实验使用的FCNN结构

FC(n = 1024, ReLU)
FC(n = 2048, ReLU)
FC(n = 3072, ReLU)
FC(n = 2048, ReLU)
FC(n = 1024, ReLU)
FC(n = #classes)

5.2 与先前方法比较

对于FCNN模型，将本文方法与 [10]和 [6]进行了比较， [10]和 [6]都是通过分析梯度来恢复私有数据。此FCNN模型被训练到每个数据集的收敛性。 [10]的性能作为baseline，表示训练后的FCNN模型中直接数据泄漏的正常程度。 [6]中的方法根据算法3构建第一层FC的恶意参数，均值$\mu = 0$，标准差$\sigma = 2$，比例因子$s = 0.97$。本文方法根据算法1构造第一层FC的恶意参数。表2列出了超参数的详细设置。学习率η在第200个epoch减少到原来的0.1。

表3给出了在FCNN模型上三种方法的恢复性能对比。可以看到本文方法在所有情况下都优于其他两种方法。图2直观地显示了整个批次的64张恢复图像。其中，64幅恢复图像中有59幅的PSNR大于40dB，与原始图像基本一致。 [10]的结果显示在训练好的FCNN模型中直接数据泄露的典型水平。随着私有数据的批量大小B增加，在Fashion-MNIST 数据集上的恢复数据的平均PSNR从34.96dB降低到15.97dB，说明较大的批量大小可以缓解隐私泄露。然而，由于引入恶意参数，第一全连接层中某些神经元的梯度仅与单个数据点相关联，使得我们的方法能够实现最佳的恢复性能。在更复杂的数据集上（如CIFAR100和FaceScrub），当批量大小为64时， [10]的PSNR分别仅为14.77dB和6.45dB。该结果表明对于训练好的模型，复杂数据集的泄露

Table 2 用于训练恶意参数的超参数

Datasets	Batchsize	k	η	Epochs	μ	σ	s
Fashion-MNIST	64	1	1e-3	300	0	2	0.97
CIFAR100	64	4	1e-3	300	0	2	0.97
FaceScrub	64	2	1e-5	300	0	2	0.97
TinyImageNet200	64	2	1e-3	300	0	2	0.97

Input: $\mathbf{W} \in \mathbb{R}^{l*n}$，$\mathbf{B} \in \mathbb{R}^{l}$：第一层FC层的权重和偏置；$\mu$,$\sigma$：高斯分布的均值和标准差；$s$：比例因子。
Output: **W**,**B**：FC层初始化的参数
for $i \leftarrow 1$ ***to*** l **do**
  $N \leftarrow \{j|j \sim \mathcal{U}(1,n)\}$ where $\mathcal{U}$ is the Discrete uniform distribution and $|N| = \frac{n}{2}$;
  $P \leftarrow \{j|j \in \{1,2,3,...,n\}, j \notin N\}$;
  $\mathbf{z}_- \sim \mathcal{N}(\mu,\sigma)$ and $|\mathbf{z}_-| = \frac{n}{2}$;
  $\mathbf{z}_+ \leftarrow s * \mathbf{z}_-$;
  $\mathbf{w}_i[N] \leftarrow Shuffle(\mathbf{z}_-)$ where $\mathbf{w}_i$ is the i^{th} row of $\mathbf{W}$;
  $\mathbf{w}_i[P] \leftarrow Shuffle(\mathbf{z}_+)$;
  $\mathbf{B} \leftarrow \mathbf{0}$;
end
return $\mathbf{W}, \mathbf{B}$

Algorithm 3: 参数初始化

程度受到限制。 [6]方法通过引入恶意参数，在复杂数据集上的表现优于 [10]。与 [6]相比，本方法构建了更有效的恶意参数，并进一步放大了第一全连接层的直接数据泄露。

5.3 有效性分析

该部分分析提出的方法在广泛的模型架构、辅助数据集上的有效性。

5.3.1 模型结构

(1) 第一层FC层的size：我们提出的方法的性能很大程度上受到第一层FC的大小的影响，因为恶意参数是针对这一层进行训练的。第一个FC层的大小决定了SDANs的数量和重叠程度。我进行了将第一个FC层的大小从128到4096的实验，同时使用相同的设置训练恶意参数（数据集：CIFAR100；batchsize：64；k：2；η：0.001；epochs：300）。结果如图3(a)所示，随着第一层FC层尺寸增大，恢复性能有所提高。当私有数据的批大小等于第一层FC层的大小时，平均PSNR约为14dB（batchsizes为128、256和512时分别为13.99dB、14.03dB和14.16dB）。更大的FC层具有更大的容量来容纳SDANs。不同私有数据对应的SDANs在第一层FC层重叠较少。因此，本文攻击方法对大的第一层FC层效果较好，对于用户，可以设置较小的第一个FC层以减轻攻击。

Table 3 三种方法在FCNN模型上的平均PSNR，B:私有数据的批量大小

Method	FMNIST	CIFAR100	FaceScrub
B = 64			
[24]	34.96	14.77	6.45
[5]	27.33	15.60	22.36
Ours	**92.64**	**64.41**	**71.88**
B = 128			
[24]	23.36	14.06	6.32
[5]	16.86	15.01	17.23
Ours	**68.94**	**34.94**	**44.97**
B = 256			
[24]	17.94	13.60	6.19
[5]	14.97	14.77	16.50
Ours	**32.16**	**18.94**	**24.82**
B = 512			
[24]	15.97	13.34	6.13
[5]	14.66	14.38	16.29
Ours	**18.47**	**15.58**	**17.49**

(2) FCNN模型的depth：图3(b)为FCNN模型在不同depth上对应的恢复性能。这些FCNN模型的第一层和最后一层与本文模型相同。FC中间层的大小均为1024。可以发现，FCNN模型的depth对此方法没有影响，因为我们的方法只依赖于模型中第一层FC的SDANs。

5.3.2 数据集

(1) 辅助数据的比例：在此实验，使用不同比例的CIFAR100测试集来训练恶意参数。图4(a)为FCNN模型的恢复性能。尽管存在随机波动，但对于不同比例的辅助数据，结果几

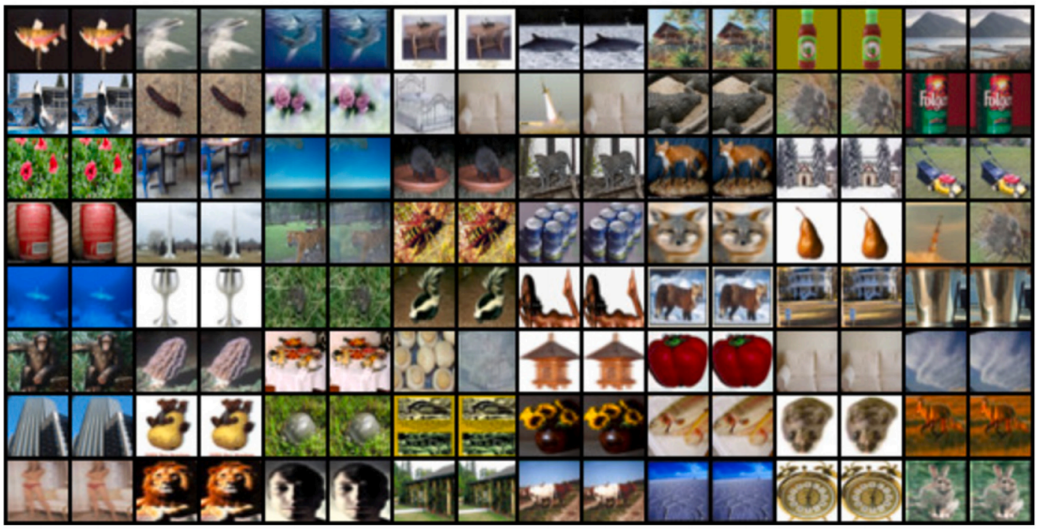

Fig. 2 奇数列为原始图像，偶数列为恢复后的图像。64幅恢复图像中有59幅的PSNR大于40dB

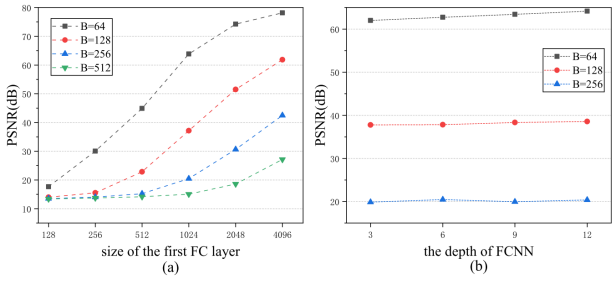

Fig. 3 本文方法在各FCNN模型上表现

乎相同。这表明本文攻击方法不依赖于大量的辅助数据。即使仅使用20%的CIFAR100测试集构建恶意参数，也可以获得与使用100%测试集的情况相当的性能。

(2) 辅助数据分布：在此实验，客户端使用CIFAR100的私有训练集训练模型，而服务器则使用不同的辅助数据（包括CIFAR100的测试集、TinyImageNet和FaceScrub）构造恶意参数。如图4(b)所示，使用TinyImageNet训练的恶意参数（MP-TinyImageNet）在恢复性能上的PSNR与使用CIFAR100训练的恶意参数（MP-CIFAR100）相似。这是因为这两个数据集在语义信息上具有相似性。然而，当使用FaceScrub训练的恶意参数（MP-FaceScrub）来恢复CIFAR100的私有数据时，平均PSNR仅为14dB。因此即使无法获取私有数据的测试集，也可以通过使用与私有数据具有相似语义信息的公共数据集来训练恶意参数。本文方法即使在这种情况下，仍然可以取得较好的恢复性能。

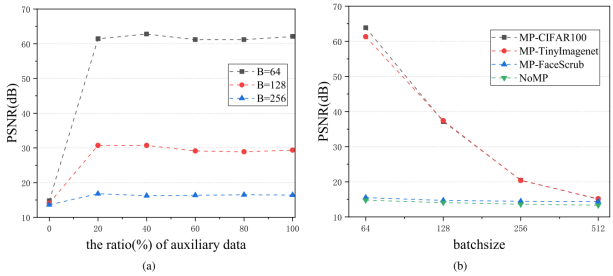

Fig. 4 辅助数据对恢复性能的影响

6 结论

本文提出了一种新的梯度反演攻击，通过模型投毒来破坏大批量联邦学习中的隐私。具体地说就是构造恶意参数以投毒客户端模型，可以显著减轻聚合梯度的混淆程度。通过分析在投毒模型上计算的共享梯度，可以较好地重建私有训练集。

7 参考文献

[1] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the gan: Information leakage from collaborative deep learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 603–618. [Online]. Available: https://doi.org/10.1145/3133956.3134012

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[3] L. Zhu and S. Han, *Deep Leakage from Gradients*. Cham: Springer International Publishing, 2020, pp. 17–31. [Online]. Available: https://doi.org/10.1007/978-3-030-63076-8_2

[4] L. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, "Robbing the fed: Directly obtaining private data in federated learning with modified models," *arXiv preprint arXiv:2110.13057*, 2021.

[5] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[6] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023, pp. 175–199.

[7] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[8] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[9] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 343–347.

[10] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Applications and Techniques in Information Security*, L. Batten, D. S. Kim, X. Zhang, and G. Li, Eds. Singapore: Springer Singapore, 2017, pp. 100–110.