福州大学计算机与大数据学院

课程技术报告

课题名称：**AI 模型攻击方法**

学　　号：**102102145**

姓　　名：胡嘉鑫

专　　业：数据科学与大数据技术

学　　期：**2023 学年第 2 学期**

2024 年 5 月

使用生成式对抗网络的联邦学习投毒方法

2024 年 5 月 4 日

目录

1 引言

1.1 联邦学习

传统的中心化的机器学习模型的训练需要用户上传自己所有需要训练的数据, 用户的隐私存在泄露的风险. 相反, **联邦学习 (Federated Learning)** 是一种新型的分布式学习框架, 通过多个协作者共同参与完成模型的学习. 联邦学习的本质是分布式的机器学习, 服务器和参与者 (模型学习者) 共享模型的参数, 服务端无法获得参与者的训练数据, 不同参与者的训练数据也是不同的, 参与者的训练数据和训练过程是本地化的, 对服务端保密. 服务端的任务是维护全局模型, 接收来自内部参与者对模型参数的更新, 并随机选择一定数量的参与者运用均值化算法更新全局模型, 直至收敛.

设有 n 个用户参与联邦学习, 且学习目标都为这 n 个参与者所知, 这 n 个用户本地的数据集都是互不相同的. 对模型的的每次迭代, 用户从服务端下载全局模型的参数, 并根据本地的数据集训练模型. 每个用户在训练阶段会将梯度上传至服务端, 服务端将来自多个用户的梯度进行平均并累积到当前的全局模型上. 下面是对这一过程的形式化描述:

$$m_{i+1} = m_i + \frac{1}{n}\sum_{k=1}^{n} g_i^k \tag{1}$$

其中, m_i 表示在第 i 轮迭代时共享的模型 (参数), g_i^k 表示第 k 个用户在第 i 轮迭代上传的梯度.

联邦学习的主要目的是在本地化数据集的基础上, 构建一个联合的机器学习模型, 同时保护数据集的隐私. 联邦学习在边缘计算和众包系统中有很重要的应用.

1.2 投毒攻击

在联邦学习的架构中, 由于参与者的训练数据和训练过程对服务端是不可见的, 全局模型的直接性能受内部的参与者的影响. 内部参与者可以发起主动的**投毒攻击 (Poisoning Attack)**, 即对模型的参数进行不正确的更新并上传至服务器, 影响全局模型的性能.

在这里, 我们研究和评估联邦学习系统中基于**生成式对抗网络 (Generative Adversarial Nets, GAN)** 的投毒攻击.

在训练阶段, 攻击者首先作为内部参与者, 在本地训练 GAN 模拟其它参与者的训练样本 (不属于此攻击者), 攻击者使用这些生成的模拟样本数据对模型进行投毒更新, 通过增大投毒训练的规模并将训练后的模型上传至服务器可影响全局模型的性能, 使全局模型拥有某些攻击者所期望的性质. 在推理阶段, 受到投毒攻击的模型则可能对某些给定的输入获得攻击者想要的输出而非实际上应该正确的输出.

以下几点原因使得投毒攻击是有可能的:

- 联邦学习系统中经常有大量的参与者, 而有些参与者可能成为攻击者;
- 参与者的本地化的训练数据和训练过程对服务端是不可见的, 服务端无法验证来自参与者对模型参数更新的正确性;
- 来自多个不同的参与者的对模型的本地更新可能是完全不同的, 并且**安全聚合协议 (Secure Aggregation Protocol)** 会使得这些来自参与者的本地更新对服务端来说不可审计.

1.3 生成对抗网络

生成对抗网络 (GAN) 在计算机视觉研究领域取得了极大的成功, 能够基于原始图像集生成高质量的假图像 (`fake image`).

GAN 结构中有两个神经网络:

- **生成器 (Generator)**: 生成器 (G) 生成图像.
- **鉴别器 (Discriminator)**: 鉴别器 (D) 判别图像是来自生成器还是原始图像集, 它们可以表示为 0/1(假/真).

为了训练一个 GAN, G 从遵循高斯或均匀分布的先验分布的随机向量 z 中生成图像。生成的图像是 D 的输入。同时, D 通过原始图像集进行预训练, 可用于区分输入是真实的还是虚假的。通过进行对抗训练的博弈, G 和 D 的性能都可以得到提高。下面是 GAN 的训练目标:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2}$$

其中 $p_{data}(x)$ 是原始图像服从的分布, $p_z(z)$ 是随机向量 z 服从的分布.

2 技术原理及方法

2.1 威胁模型

2.1.1 学习的背景

这里的威胁模型考虑多个参与者 ($N \geq 2$) 在各自本地的训练数据集上联合训练全局模型的情况. 我们假设在这些参与者中存在一个或多个的攻击者, 攻击者的目的是模拟其他参与者的数据并污染全局模型, 而中央服务器和大多数的参与者不存在恶意投毒的目的, 即中央服务器和绝大多数的参与者是可信任的.

在我们的威胁模型中, 攻击者在参与联邦学习的过程中只能访问全局的模型, 无法影响服务端的均值化算法和服务端对全局模型的更新. 同时, 所有的攻击者均无法直接干扰其他非恶意参与者的训练过程和训练数据 D_{benign}. 为了遵循联邦学习的约定, 攻击者还必须在投毒的数据 D_{poison} 上正确训练和更新本地模型. 不失一般性和简单性, 这里的全局模型设定为图像分类器.

2.1.2 攻击者的目标

攻击者试图伪装成一般的参与者 (非恶意的), 在训练阶段破坏全局模型. 在联邦学习的过程中, 攻击者在本地部署 GAN 生成模拟其他参与者的训练样本的数据, 并给这些样本赋以错误的标签, 将样本及标签加入攻击者本地的训练数据集中, 攻击者的本地模型在这些投毒的数据上进行训练得到投毒的模型, 当攻击者本地的模型被服务端选中并用某种算法更新到全局模型上, 污染就进一步传播到全局模型上. 此时, 全局模型就具有攻击者所期望的某些性质.

攻击者的目标概括如下:

1. 样本生成: 攻击者在不访问其他参与者的数据的情况下, 模拟生成其他参与者的训练数据.
2. 提高投毒目标任务的性能: 经过投毒攻击的几轮模型迭代, 当攻击者本地投毒过的模型被更新到中央服务器的全局模型后, 全局模型应该在被投毒的分类任务上具有较高的预测性能.
3. 提高主要目标任务的性能: 全局模型在其它未被投毒的分类任务上也应该具有较高的准确性, 以避免全局模型的表现不好被丢弃.

2.1.3 攻击者的能力

- 攻击者作为联邦学习的内部参与者之一, 可以在本地部署 GAN, 修改本地训练的数据集, 调整训练过程, 发起主动攻击 (`active attack`).
- 攻击者对模型的结构有全面且清晰的了解, 因为所有联邦学习的参与者已经事先确定共同的学习目标.

2.2 实施攻击

2.2.1 概述

在联邦学习中实施投毒攻击有以下 3 个步骤:

1. 基于 GAN 生成欲投毒样本并将其加入本地的训练数据集;
2. 将生成的样本和错误的标签关联后注入训练过程, 经过训练得到投毒的本地参数;
3. 将投毒的本地模型上传至中央服务器, 旨在误导全局模型在推理阶段将输入分类为攻击者所期望的标签而非正确的.

简明起见, 首先考虑有两个参与者 (攻击者 A 和非恶意参与者 P) 的联邦学习系统, 以此描述投毒攻击.

假定 P 和 A 分别训练分类类别 a 和类别 b, 且 P 和 A 的学习目标和模型结构是一致的(联邦学习的约定). 在这种情况下, 类别 a 的信息对于 A 而言是不可见的. 攻击者的目标是模拟来自类别 a 的数据并实施投毒攻击. 因此, A 在本地部署 GAN(对中央服务器和 P 是不可见的), 以此产生来自类别 a 的模拟数据 (将其标识为类别 b), 并将这些数据注入本地的训练过程. 攻击者在投毒数据集的基础上训练模型, 并将所得模型上传至中央服务器. 因此, 通过增大投毒的规模, 便能在联邦学习系统中成功实施投毒攻击.

2.2.2 生成投毒数据

GAN 是两个神经网络间的对抗博弈, 即生成器 G 和判别器 D. 判别器被训练来用于区分原始的数据和生成的数据, 而生成器被训练来用于生成能够模拟判别器的训练数据的伪造数据. 为了模拟其他参与者的数据, 攻击者在本地采用 GAN, 将全局模型作为判别器. 经过不断地迭代后, 全局模型将收敛. 对应地, 判别器会和全局模型同步, 引导生成器的收敛. 因此, 攻击者可以使用这种方式生成高质量的伪造数据. 这些伪造的数据经过投毒会影响全局模型在特定类别上的分类性能.

2.2.3 基于 GAN 的投毒攻击

根据以上分析, 实施攻击的关键点是攻击者在本地的训练数据中投毒, 训练完成后将模型的更新 ΔL^p 上传至中央服务器. 投毒攻击可归结为如下步骤:

1. 假定 P 和 A 为联邦学习系统中的两个参与者, 二者的训练数据集对其他参与者保密 (类别 a 和 b);
2. 通过联邦学习迭代全局模型直至精度达到一定水平;
3. 对于参与者 P:
   (a) 下载全局模型以更新 P 的本地模型;
   (b) 在本地数据集上训练模型, 并将本地的更新 ΔL^i 上传至中央服务器;

4. 对于攻击者 A:

   (a) 下载全局模型以更新 A 的本地模型;

   (b) 复制新的本地模型, 将其作为 D(判别器), 在 D 的基础上运行 G (生成器) 以模拟 P 的类别 a 中的数据;

   (c) 将生成的类别 a 数据标记为类别 b, 并将数据插入到 A 的本地数据集中;

   (d) 在投过毒的数据集上训练本地模型得到新的投毒的模型, 模型更新为 ΔL^p;

   (e) 选定某个常数因子 λ, 将 ΔL^p 扩大对应的规模, 将投毒的更新 $\lambda\Delta L^p$ 上传至中央服务器;

5. 重复步骤 3 和 4 直到全局模型收敛.

步骤 4 的子步骤 (2) 到 (5) 描述的是投毒数据的生成和投毒攻击的实施. 这种投毒方法的效率仅取决于全局模型 (判别器) 的精确度.

拥有多个攻击者的一般化的投毒攻击在下面给出.

Algorithm 1 联邦学习中的投毒攻击

输入: 全局模型 M_t; 参与者对模型的更新 ΔL_t^i; 损失函数 ℓ; 学习率 η.

输出: 投毒的模型更新 $\Delta\hat{L}_t^i$.

```
初始化生成器 G 和判别器 D
for t ∈ (1, 2, ··· , T) do
    将 M_t 发送给参与者                                  ▷ 服务端执行
    接收来自参与者的更新: ΔL_{t+1}^i
    更新全局模型: M_{t+1}
    替换本地模型: L_t^i ← M_t                            ▷ 参与者执行
    if 参与者是 A(攻击者) then
        根据新的本地模型 L_t^i 初始化 D
        for each epoch e ∈ (1, ··· , E) do
            在 D 的基础上为目标类别运行 G
            使用 D 更新 G
            使用 G 生成目标类别的样本
            将错误的标签和生成的样本关联
            将投毒数据注入本地训练数据集 𝒟
            for each batch b_p ∈ 𝒟_poison do
                L_{t+1}^p = L_{t+1}^p − η_adv ∇ℓ(L_t^p, b_p)
```

end for
end for
计算投毒更新: $\Delta L_{t+1}^{p} = L_{t+1}^{p} - L_{t}^{p}$
增大投毒规模: $\Delta \hat{L}_{t+1}^{p} = \lambda \Delta L_{t+1}^{p}$;
else
计算参与者的模型更新: $\Delta L_{t+1}^{i} = L_{t+1}^{i} - L_{t}^{i}$;
end if
上传本地模型更新 ΔL_{t+1}^{i}(包含 $\Delta \hat{L}_{t+1}^{p}$) 至中央服务器
end for

3 总结与发展趋势

我们基于生成对抗网络（GAN）在联邦学习中设计了一种投毒攻击。成功实施攻击的关键是在攻击者端部署一个 GAN 架构，这个架构可以模仿其他参与者的训练数据集中的样本。只要共享模型的准确性随着时间的推移得到提高，这个 GAN 的有效性就可以得到极大的保证，这也是联邦学习的主要思想。此外，基于 GAN 的生成模型无法被检测到，因为攻击者假装是联邦学习协议中的诚实参与者。因此，这里提出的投毒攻击的有效性可以得到保证。

此外，现有的投毒攻击防御机制，如鲁棒损失和异常检测，都不适用于联邦学习，因为它们都需要检测器访问参与者的训练数据和训练模型，这与联邦学习的设计思想相矛盾。至于防御方面，由于这里的投毒攻击依赖于 GAN 来模仿其他参与者的训练数据，因此可以设计一种新的联邦学习框架，该框架隐藏全局模型的分类结果，以防止攻击者使用 GAN 获取其他参与者的私有训练数据，并最终防范内部参与者发起的投毒攻击。

参考文献

[1] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. *Poisoning Attack in Federated Learning using Generative Adversarial Nets*, 2019.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong. *Federated Machine Learning: Concept and Applications*, ACM Transactions on Intelligent Systems and Technology., vol.10, no.2, pp. 1-19, Jan. 2019.

[3] B. Biggio, B. Nelson, and P. Laskov. *Poisoning Attacks against Support Vector Machines*, in Proc. 29th International Conference on Machine Learning. (ICML), Edinburgh, Scotland, Jun. 2012, pp. 18071814.

[4] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. *Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning*, 2017.