这周还是在看论文，想idea很难，自己也有许多想了解想学习的方面

https://github.com/innovation-cat/Awesome-Federated-Machine-Learning FL顶会汇总

论文阅读总结

同态相关

BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning 2020 USENIX Annual Technical Conference

- 本文通过将梯度提前放缩，再进行剪切之后，进行量化后加密，上传到聚合服务器进行聚合（无需解密）。成功降低加密开销和密文的总量，以及节约了成本。

  这个文章做的工作主要集中在批处理同态加密方面——1增加了量化的正负号2增加了阈值的计算方案3结合最先进的而优化器。

  我们没有完全精确地加密单个梯度，而是**将一批量化的梯度编码成一个长整数，并一次性加密它（批量加密技术）。** 为了允许在编码批次的密文上进行梯度聚合，我们开发了新的量化和编码方案以及一种新的梯度剪切技术。在FATE上实施BatchCrypt作为一个插件模块。BatchCrypt实现了23×-93×训练加速，同时减少了66×-101×的通信开销。由于量化误差而造成的精度损失小于1%。

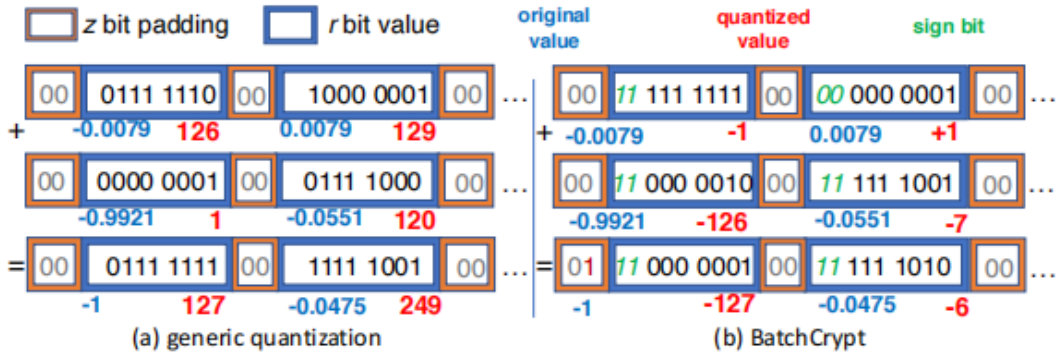


Figure 3: An illustration of a generic quantization scheme and BatchCrypt. The latter preserves additivity during batching, with the sign bits highlighted within values.

  我们设计了一个定制的量化方案，将**梯度值量化到在对称范围内均匀分布的有符号整数。**此外，为了支持简单的加法形式的梯度聚合，并且添加不会导致溢出破坏编码的梯度，**我们开发了一种新的批编码方案，该方案采用两个符号位表示量化值的二的互补表示。**此外，我们还使用填充和提前缩放，以避免溢出。

- 展望：垂直FL的适用性需要复杂的操作，如乘以密文矩阵。对这种计算的批处理超出了批密码目前的能力。我们将把它作为未来的工作。
- SIMD（单指令多数据）技术 仅限于基于格密码的?

  Reducing Communication Overhead：由于数据膨胀主要是由于明文和密文长度不匹配造成的，一个直观的想法是**尽可能多的梯度组合在一起形成一个长的明文**，这样加密操作的数量将大大减少。然而，挑战仍然是如何在不修改ML算法或损害学习精度的情况下，进行批处理后保持HE的加性特性的。

  之前用的pailler密文打包：~~处理后还是很慢，在一般的机子上Resnet18都不能忍受。~~

为了降低通信成本，利用密文打包技术，即将多个明文打包成一个密文，并用Single Instruction Multiple Data 单指令多数据（SIMD）技术对这些值并行执行操作。假设$pk_c = p$，对于文本空间$\log_2 p \geqslant 2048bits$，每个用户将$d$个明文打包成一个密文：

$$\{[\![\overbrace{\lceil 2^{prec} \cdot G_{xi} \rfloor 0 \cdots \lceil 2^{prec} \cdot G_{x(i+d)} \rfloor 0}^{prec+pad\ bits}]\!]_{pk_c}\}_{i=1}^{\lceil \frac{p}{q} \rceil} \quad (7)$$

P2P FL

BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learning

缺少中央服务器主体不仅使我们的环境能够抵抗故障，而且还排除了对每个人都信任的主体的需要。此外，任何客户端在任何时候都可以非常动态地启动一个更新过程。由于通信回合的数量并不是医疗中心的瓶颈，因此它们可以更频繁地进行互动。由于每个客户端的交互和更新过程的高频率高，BrainTorrent中的**模型收敛速度更快**，达到了与汇集所有客户端数据训练的模型相似的精度。

P2PFL算法步骤：~~（个人觉得 变化不大呀？更安全了嘛？）~~

- 在训练的开始每个client先用本地的数据集训练本地模型几个轮次
- 然后在每个round，执行以下步骤：
  - 随机选择一个client C_i
  - C_i向剩下的$N-1$个client发送ping请求
  - 对于每个C_j（其中$j \in \{1, ..., i-1, i+1, ..., N\}$），如果它的模型版本大于$C_i$最近一次聚合所使用的$C_j$的版本，那么$C_j$将新的模型$W^j$和数据集大小$a_j$发送给$C_i$
  - C_i使用公式$W \leftarrow W + \frac{a_j}{a} W^j$聚合模型
  - C_i使用本地数据集对聚合得到的模型进行微调
  - C_i更新本地的版本数据v^i
- 重复步骤（2）直至模型收敛

去中心化联邦方法都是使用已有的网路协议来对传统的联邦聚合方法进行改进，达到去中心化的效果。使用gossip协议的优点在于，可以通过segmentation来达到降低单条通信链路上的开销，不足之处在于所有训练节点达成共识（每个节点上的模型一样）的过程过慢，并且整体通信开销要大；P2P的优点在于比起gossip协议，可以让所有训练节点达成共识，但是通信开销依旧是一个瓶颈。

Peer-to-Peer Federated Learning on Graphs

本地客户端分布在一个图上，它们**只与单跳邻居通信（多个本地客户端构成通信拓扑图）**

本地客户端所构成的通信拓扑图中，节点的互动（边权表示）由一个随机矩阵来表示。矩阵中元素（权重）由节点 i 对它从节点 j 那里收到的信息的**置信值**（可学习）表示。

VFL&Inference attack

垂直联邦学习（Vertical Federated Learning，VFL）：各个参与者的数据集在样本空间一致，但在特征空间不同；例如某银行与某电商平台之间的合作——**两者的用户群体存在重叠，但是各自拥有不同的用户特征。**

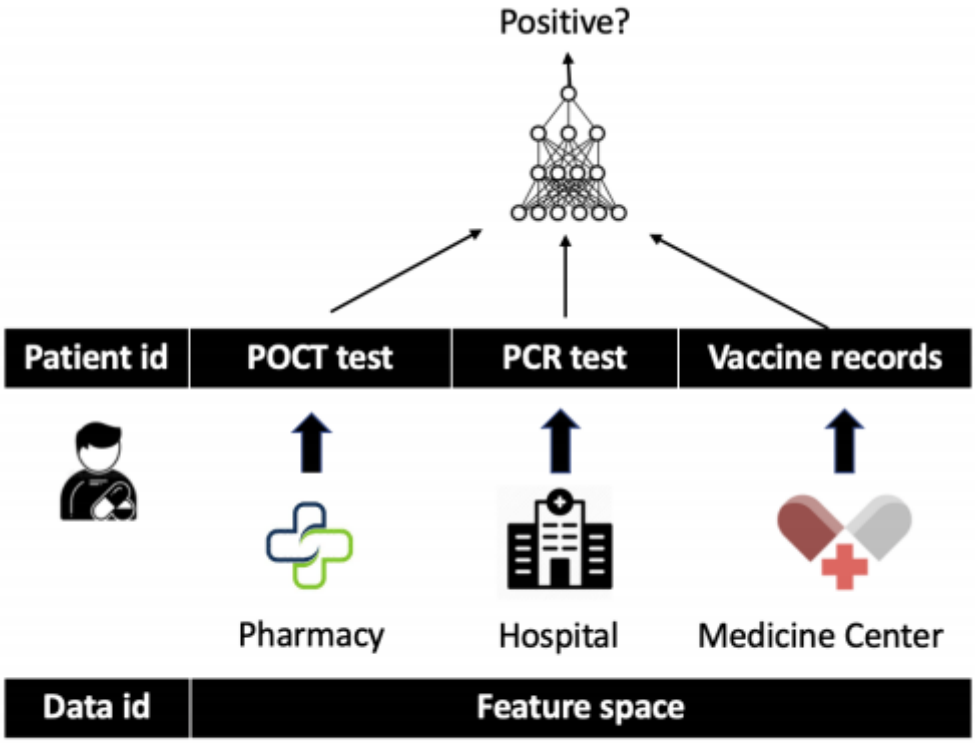


Figure 2: VFL among medical institutions

Label Inference Attacks Against Vertical Federated Learning 2022 USENIX

- 现有的VFL架构对于标签的存储使用具有间接的隐私泄露风险：架构中由参与者维护的底层模型参数、训练算法中的梯度交换机制都有可能被潜在的恶意参与者利用，以窃取服务器端的标签数据。

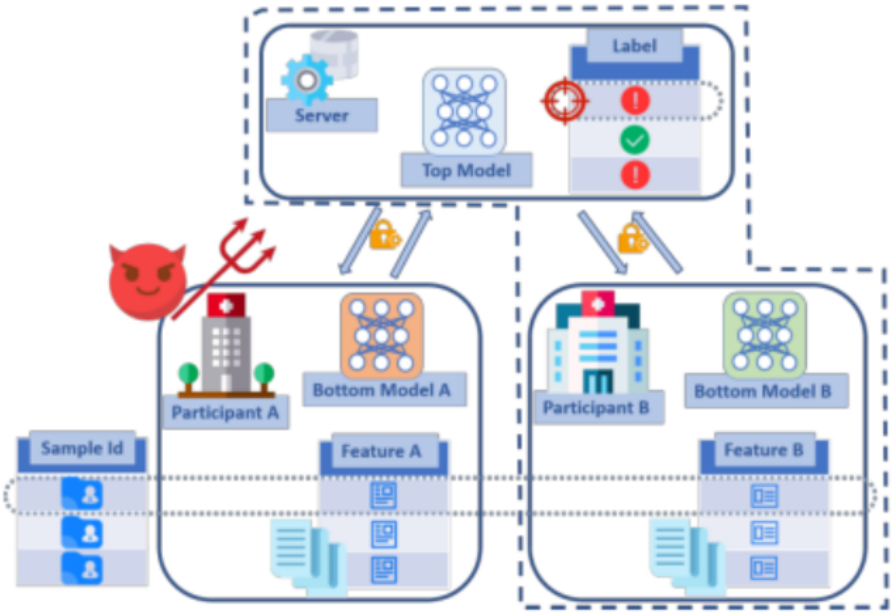


Figure 1: Illustration of the label inference attacks against VFL. In practice, the server is usually managed by the participant who holds the labels, as outlined by the dotted line. The participant without the labels is the adversary, whose goal is to infer the labels.

在图1所示的VFL体系结构中，只有一个参与者拥有标签，这与HFL不同，HFL中每个参与者都有自己的标签样本。确保私人标签的隐私是VFL提供的基本保证，因为标签可能是参与者的关键资产或高度敏感。此外，敌对参与者可能会试图利用被盗的标签建立类似的业务，或将敏感标签出售给地下行业。

本文提出三种标签推断攻击：

- 基于模型补全和半监督学习的被动攻击

被动攻击：攻击方是Passive Party **(honest but curious)**，持有本地训练好的单个底部模型，**目的**是去推理自己训练集内的标签信息和其他参与方持有的标签信息。

Method：在VFL中，attacker (Passive Party) 通过补全(Model Completion)本地的bottom模型（即给本地的bottom模型添加分类层），借助少量有标签的样本使用半监督学习算法（如MixMatch）对没有标签的样本进行预测，预测结果为伪标签，预测标签的精确程度通过不断fine-tune 补全后模型的分类层提高。最后，攻击方得到trained的完整模型(top + 其他参与方的bottom model)，输入本地feature到trained的完整模型，即可推断标签。

- 基于本地恶意优化器的主动攻击

主动攻击：攻击方是Passive Party **(malicious)**

Method：在VFL中，attacker (Passive Party) 可以通过恶意修改优化器，加快bottom model梯度下降的速度，获得当前轮Top model参数优化时的优先权，使得attacker的本地bottom model能够**间接**获得"更倾向的"特征提取能力，以增强上面基于Model Completion补全的攻击方法的效果

- 基于梯度推算的直接攻击

受到相关工作《iDLG: Improved Deep Leakage from Gradients》的启发，本文进一步研究发现，对于不设置top model的VFL架构，可以直接依据服务器端反传的梯度的符号，推算出当前训练数据的标签。因为基于直接推算，此直接攻击方法在实验涉及的所有数据集上取得了100%的标签推断成功率。

3.1 Direct Label Inference Attack

- 攻击目标：在VFL场景下，模型没有经过切分（model splitting），针对中间传递的计算梯度，attacker可以直接依据服务器端反传的梯度的符号，推断出本地训练数据集的标签。
- 对损失函数有限制：交叉熵损失，加权交叉熵损失，负对数似然损失 (用于分类任务)。
- 如下已交叉熵为例，x 表示每个样本的特征, c 为标签，k表示参与的Passive party的数量，y_i^k 表示第k个Passive party的bottom model的最终的全连接层的类别 i 输出，y^k 表示为所有Passive party的最终的全连接层的所有类别的输出，由 $y^k = \left[y_1^k, y_2^k, \ldots\right]$ 表示

$$\text{loss}(x, c) = -\log \frac{e^{\Sigma_k y_c^k}}{\sum_j e^{\Sigma_k y_j^k}}$$

$$g_i^{adv} = \frac{\partial \text{loss}(x,c)}{\partial y_i^{adv}} = -\frac{\partial \log e^{y_c^{adv}} - \partial \log \sum_j e^{y_j^{adv}}}{\partial y_i^{adv}} = \begin{cases} -1 + \frac{e^{y_i^{adv}}}{\sum_j e^{y_j^{adv}}} & \text{if } i = c \\ \frac{e^{y_i^{adv}}}{\sum_j e^{y_j^{adv}}} & \text{if } i \neq c \end{cases}$$

于是，可得下式，即 g_i^{adv} 是否大于0 (符号)表示预测结果是否是真实标签。

$g_i^{adv} < 0 \text{ if } i = c \text{ and } g_i^{adv} > 0 \text{ if } i \neq c \text{ because } \sum_i e^{y_j^{adv}} \in (0, 1)$

于是，当服务器在每次迭代中返回 g_i^{adv} 给攻击方时，攻击方就可以用符号表示**间接**推断当前训练样本的标签。

另外，本文评估了4种主流联邦学习隐私增强方法：梯度加噪，梯度压缩，PPDL（Privacy-preserving deep learning）和梯度离散化。实验结果显示现有的这4种防御无法有效抵御标签推断攻击：无法做到既基本维持模型在原任务上的性能，又有效降低标签推断攻击的推断成功率。

CAFE: Catastrophic Data Leakage in Vertical Federated Learning NeurIPS 2021

为什么大批量数据难以恢复?

在VFL训练过程中，假设 batch size = K，即每回合更新 K 个训练数据，那么恢复样本这一过程可以看成如下优化目标：

$$\hat{\mathcal{D}}' = \arg\min_{\hat{\mathcal{D}}'} \left\| \frac{1}{K} \sum_{(\mathbf{x}_n, y_n) \in \mathcal{D}} \nabla_{\Theta} \mathcal{L}(\Theta, \mathbf{x}_n, y_n) - \frac{1}{K} \sum_{(\hat{\mathbf{x}}_n, \hat{y}_n) \in \hat{\mathcal{D}}'} \nabla_{\Theta} \mathcal{L}(\Theta, \hat{\mathbf{x}}_n, \hat{y}_n) \right\|^2 .$$

根据公式可以发现，随着 K 的增加，原一批次的数据 D 和虚构的 对应的虚构数据 $\hat{\mathrm{D}}'$ 的基数会增加，根据线性代数的理论，维度增加，解的数目也会增加，因此难以优化到正确的 $\hat{\mathrm{D}}'$ 。

Table 1: Comparison of CAFE with state-of-the-art data leakage attack methods in FL.

Method	Optimization terms	Reported maximal batch size	Training while attacking	Theoretical guarantee	Additional information other than gradients
DLG [32]	ℓ_2 distance between real and fake gradients	8	No	No	No
iDLG [30]	ℓ_2 distance	8	No	Yes	No
Inverting Gradients [11]	Cosine similarity, TV norm	8 100 (Mostly unrecognizable)	Yes	Yes	Number of local updates
A Framework for Evaluating Gradient Leakage [26]	ℓ_2 distance, label based regualrizer	8	No	Yes	No
SAPAG [25]	Gaussian kernel based funciton	8	No	No	No
R-GAP [31]	recursive gradient loss	5	No	Yes	The rank of matrix A defined in [31]
Theory oriented [22]	ℓ_2 distance, ℓ_1 distances of the recovered feature map	32	No	Yes	Number of Exclusive activated neurons
GradInversion[29]	Fidelity regularizers, Group consistency regularizers	48	No	No	Batch size $\ll$ number of classes & Non repeating labels in a batch
CAFE (ours)	ℓ_2 distance, TV norm, Internal representation norm	100 (our hardware limit)	Yes	Yes	Batch indices

在训练过程中，当地工人与他人交换他们的中间结果，以计算梯度并上传它们。因此，**服务器可以访问模型参数及其梯度。**由于数据垂直分区不同的工人，对于每个批，服务器（作为攻击者）需要发送 a data index or data id list到所有本地工人，以确保data with the same id sequence被每个工人selected.(我们将这一步命名为**数据索引对齐**)。数据索引对齐是垂直训练过程中不可避免的一个步骤，这为服务器（攻击者）提供了控制所选的批处理数据索引的机会。

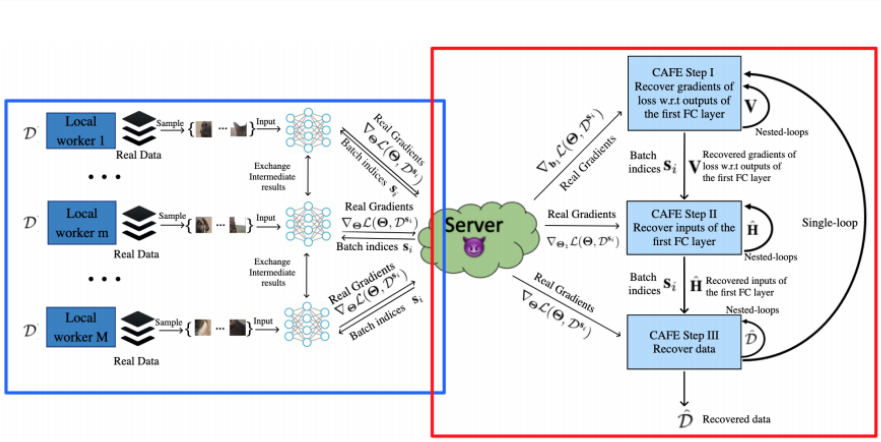


Figure 3: Overview of CAFE. The left part (blue box) performs the regular VFL protocol and the right part (red box) illustrates the main steps of CAFE.

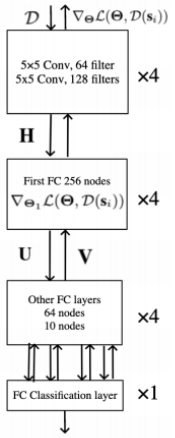


Figure 4: Model structure in VFL.

> 在VFL中，server 可以选定每轮更新哪些 id 对应的数据，因此，其可以创建一个由0和1组成的矩阵 s^t(数据集index总数量N $\times$ 1) 来表示第 t 轮更新了哪些数据：
> $\mathcal{D}(\mathbf{s}^t) = \{(\mathbf{x}_n, y_n) | \mathbf{s}^t[n] = 1\}.$
>
> 因此梯度可以表示为：
>
> $$\nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, \mathcal{D}(\mathbf{s}^t)) := \frac{\partial \mathcal{L}(\mathbf{\Theta}, \mathcal{D}(\mathbf{s}^t))}{\partial \mathbf{\Theta}} = \frac{1}{K} \sum_{n=1}^{N} \mathbf{s}^t[n] \frac{\partial \mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n)}{\partial \mathbf{\Theta}}.$$

逐层精确恢复：

第一步，根据下式恢复 the gradients of loss w.r.t the outputs of the first FC layer：

$$\mathbf{V}^* = \arg\min_{\mathbf{V}} \underbrace{\mathbb{E}_{\mathbf{s}_i \sim \mathbf{Unif}(\mathcal{S})} [\mathcal{F}_1(\mathbf{V}; \mathbf{s}_i)]}_{:=\mathcal{F}_1(\mathbf{V})} \quad \text{with} \quad \mathcal{F}_1(\mathbf{V}; \mathbf{s}_i) := \left\| \mathbf{V}^\top \mathbf{s}_i - \nabla_{\mathbf{b}_1} \mathcal{L}(\mathbf{\Theta}, \mathcal{D}(\mathbf{s}_i)) \right\|_2^2.$$

第一步，根据下式恢复 Recover inputs to the first FC layer：

$$\hat{\mathbf{H}}^* = \arg\min_{\hat{\mathbf{H}}} \underbrace{\mathbb{E}_{\mathbf{s}_i \sim \mathbf{Unif}(\mathcal{S})} \mathcal{F}_2(\hat{\mathbf{H}}; \mathbf{s}_i)}_{:=\mathcal{F}_2(\hat{\mathbf{H}})} \quad \text{with} \quad \mathcal{F}_2(\hat{\mathbf{H}}; \mathbf{s}_i) := \left\| \sum_{n=1}^{N} \mathbf{s}_i[n] \hat{\mathbf{h}}_n (\mathbf{v}_n^*)^\top - \nabla_{\mathbf{\Theta}_1} \mathcal{L}(\mathbf{\Theta}, \mathcal{D}(\mathbf{s}_i)) \right\|_F^2.$$

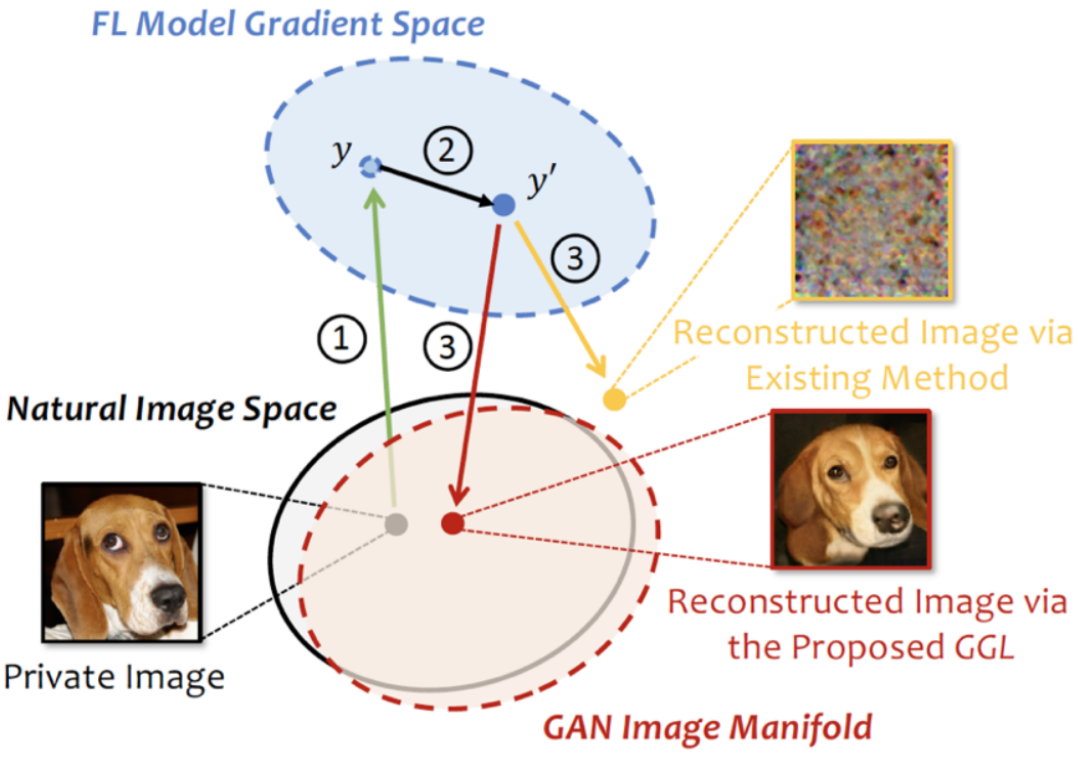
最后，根据下式恢复训练数据：

$$\hat{\mathcal{D}}^* = \arg\min_{\hat{\mathcal{D}}} \mathbb{E}_{\mathbf{s}_i \sim \mathbf{Unif}(\mathcal{S})} [\mathcal{F}_3(\hat{\mathcal{D}}; \mathbf{s}_i)] \tag{9}$$

$$\text{with} \quad \mathcal{F}_3(\hat{\mathcal{D}}; \mathbf{s}_i) := \alpha \left\| \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, \mathcal{D}(\mathbf{s}_i)) - \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, \hat{\mathcal{D}}(\mathbf{s}_i)) \right\|_2^2 + \beta \underline{\mathrm{TV}_\xi}(\hat{\mathcal{X}}(\mathbf{s}_i)) + \gamma \sum_{n=1}^{N} \left\| \mathbf{s}_i[n] (\hat{\mathbf{H}}_n^* - \widetilde{\mathbf{h}}_n) \right\|_2^2$$

我们还提出了一种有效的对策，利用假梯度(附录F)来降低CAFE的潜在风险。

Auditing Privacy Defenses in Federated Learningvia Generative Gradient Leakage CVPR 2022



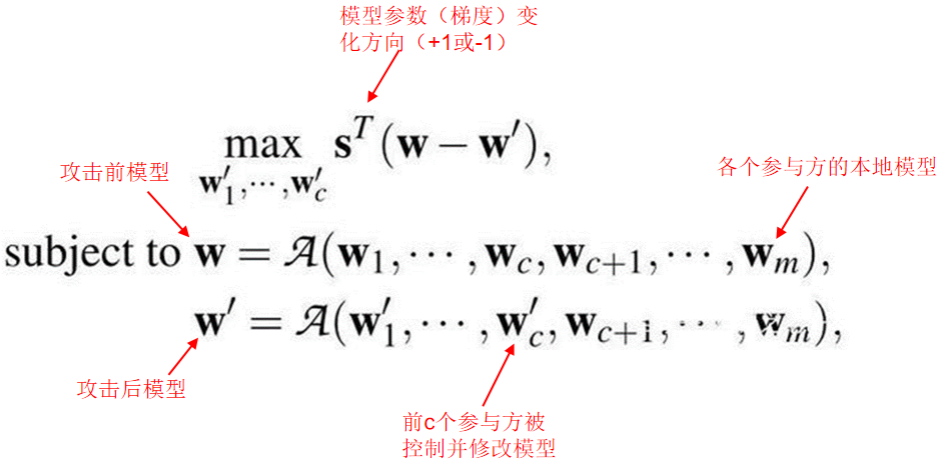
拜占庭攻击

Local Model Poisoning Attacks to Byzantine-Robust Federated Learning

虽然本文通篇都在讲local model，但本质上local model就是梯度数据，因此要改造的还是梯度数据，也就是说，**依然是在梯度数据的处理上做文章。**

定义一个方向量，1表示当前梯度增加，-1表示当前梯度减小，其次定义攻击前的梯度与攻击后的梯度，那么优化问题的实质就是，**使攻击后的梯度与攻击前的梯度差别尽量大。**

将攻击形式化为优化问题：highly nonlinear and large search space



$$\max_{\mathbf{w}'_1,\cdots,\mathbf{w}'_c} \mathbf{s}^T(\mathbf{w}-\mathbf{w}'),$$
$$\text{subject to } \mathbf{w} = \mathcal{A}(\mathbf{w}_1,\cdots,\mathbf{w}_c,\mathbf{w}_{c+1},\cdots,\mathbf{w}_m),$$
$$\mathbf{w}' = \mathcal{A}(\mathbf{w}'_1,\cdots,\mathbf{w}'_c,\mathbf{w}_{c+1},\cdots,\mathbf{w}_m),$$

因为攻击者在每轮迭代中均操纵本地模型，因此省略了迭代轮次的下标。

Attacking Krum

由于Krum是选择一个最接近的local model作为返回结果，那么就尽量在攻击时让FL选择被compromise的设备的local model。这种情况下一共分为两步。首先，要让compromise设备的local model偏差尽量大，其次，要让其他compromise设别的local model都接近这个偏差值，这样方便FL依照Krum原则对他进行选择，从而达到攻击目的。

Trimmed mean类似+数学

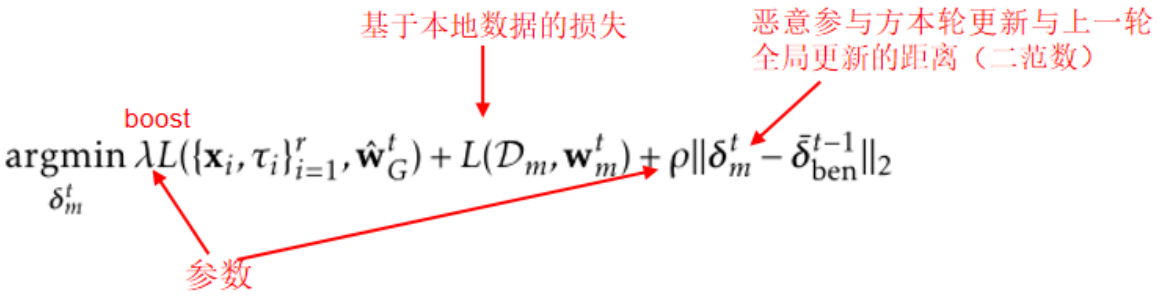
防御方法：

- Error Rate based Rejection (ERR)：对于每个本地模型，当包括本地模型时，使用聚合规则来计算全局模型A，而当排除本地模型时，再使用聚合规则来计算全局模型B，然后计算全局模型A和B在验证数据集上的错误率，抛弃掉对**错误率**有大幅影响的本地模型。
- Loss Function based Rejection (LFR)：与ERR类似，只是在验证数据集上计算模型的交叉熵损失函数值，抛弃掉对**loss值**有大幅影响的本地模型。

后门攻击

Analyzing Federated Learning through an Adversarial Lens

假设数据是iid的



$$\underset{\delta_m^t}{\text{argmin}}\ \lambda L(\{\mathbf{x}_i,\tau_i\}_{i=1}^r,\hat{\mathbf{w}}_G^t)+L(\mathcal{D}_m,\mathbf{w}_m^t)+\rho\|\delta_m^t-\bar{\delta}_{\text{ben}}^{t-1}\|_2$$

但是，这一方法会导致异常的梯度更新容易被基于准确度的方法检测出来。

• 于是，使用交替最小化（alternating minimization）方法，核心：将扰动性目标与隐蔽性目标分开优化。先优化扰动性目标，以$\mathbf{w}_m^{i-1,t}$作为初始点，得到update $\tilde{\boldsymbol{\delta}}_m^{i,t}$。然后再优化隐蔽性目标，以$\tilde{\mathbf{w}}_m^{i,t} = \mathbf{w}_m^{i-1,t} + \lambda\tilde{\boldsymbol{\delta}}_m^{i,t}$作为初始点，得到$w_m^{i,t}$进入下一个epoch。

$\delta_m^t = \lambda\tilde{\delta}_m^t$，$\widehat{w}_G^t = w_G^{t-1} + \alpha_m\delta_m^t$