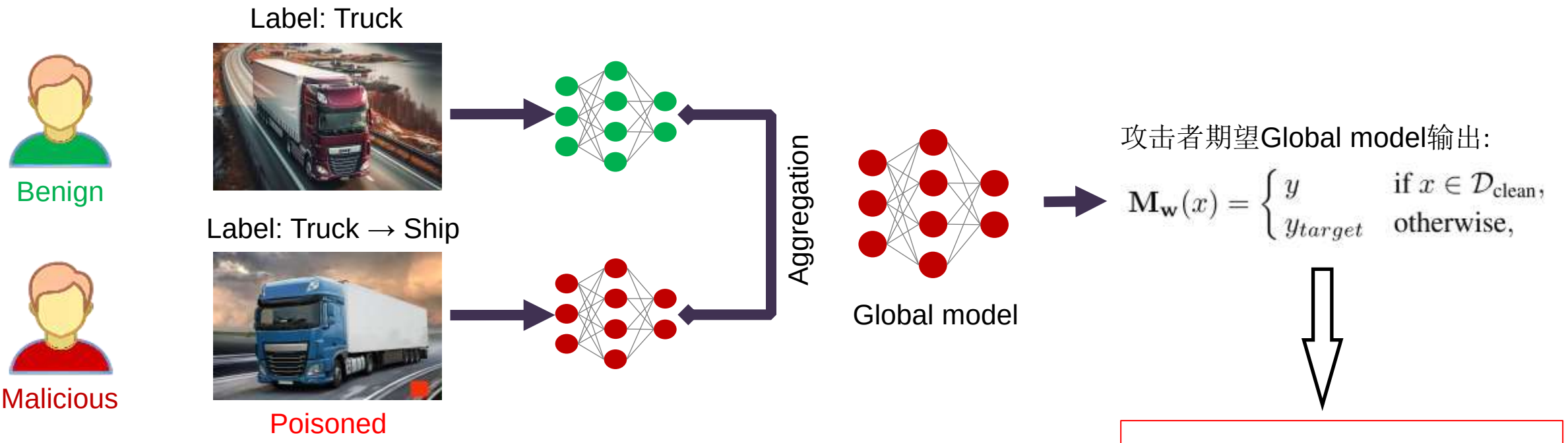
Lockdown: Backdoor Defense for Federated Learning with Isolated Subspace Training

Tiansheng Huang, Sihao Hu, Ka-Ho Chow, Fatih Ilhan, Selim Furkan Tekin, Ling Liu
School of Computer Science
Georgia Institute of Technology, Atlanta, USA
{thuang374, shu335, kchow35, filhan3, stekin6}@gatech.edu, ling.liu@cc.gatech.edu

NIPS 2023

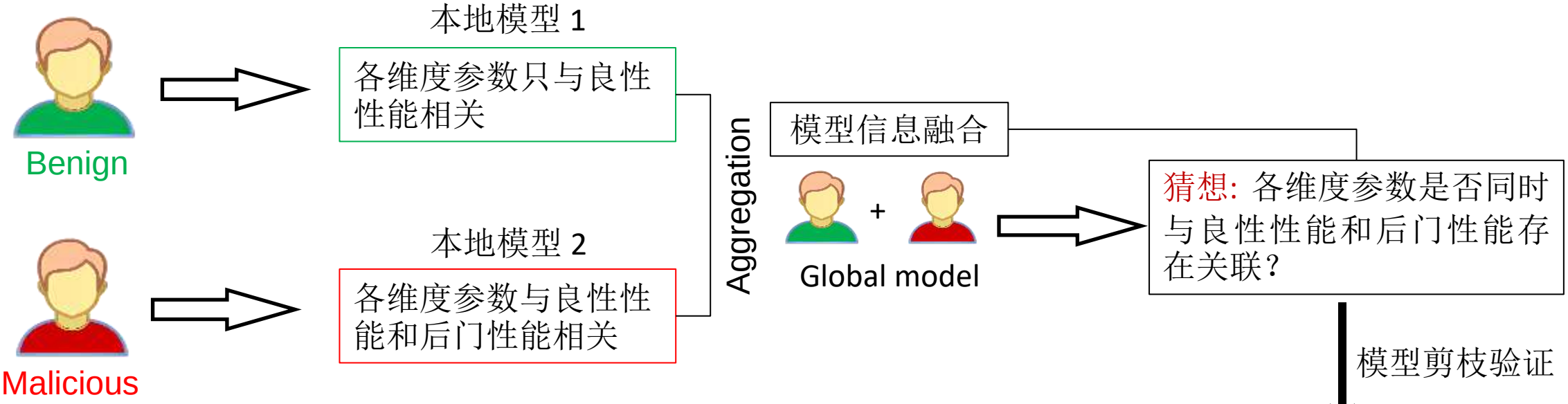
Background

- 联邦学习(Federated Learning, FL) 因隐私限制, 服务器无法监督客户端的本地训练; 恶意客户端可在本地数据上向某类别样本注入触发器并篡改标签以训练后门模型, 通过全局聚合将后门信息混入全局模型。

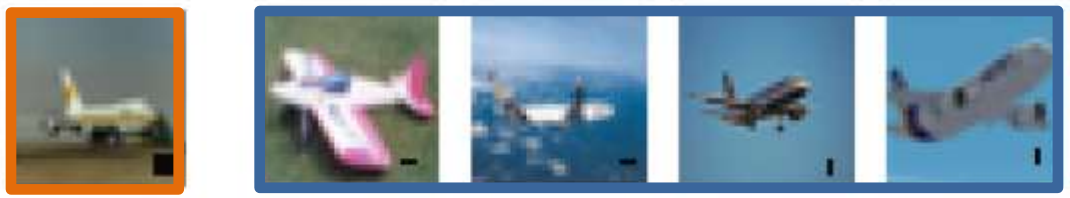


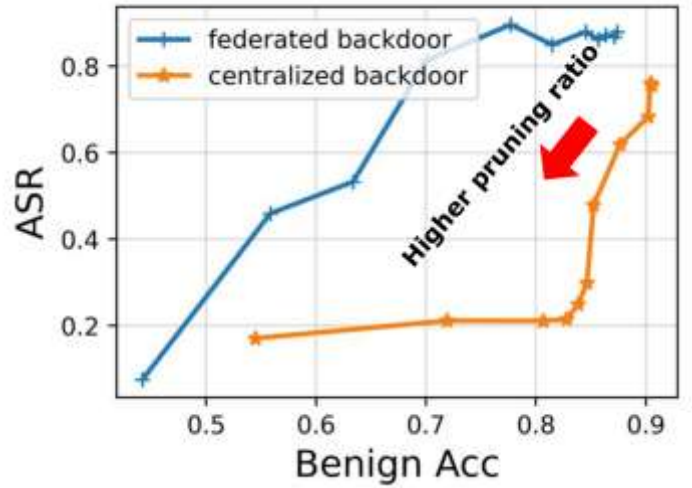
Motivation

- 全局模型参数耦合: 各维度模型参数同时与良性性能和后门性能关联



随着剪枝率提升(左下方向), 模型的良性性能(Benign ACC) 和 后门性能(ASR) 同时出现大幅下降, 验证了同一维度的参数与良性性能相关的同时还与后门性能关联。





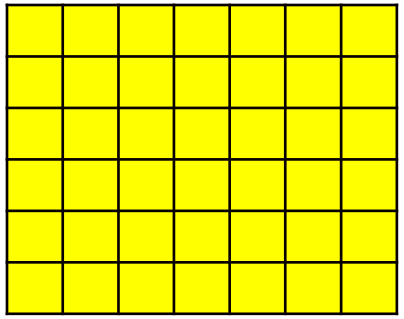
Method

- 减缓全局模型参数耦合: 隔离子空间训练 (isolated subspace training)

目的: 1. 约束客户端更新的参数数量, 降低耦合; 2. 提升客户端训练速度, 节省计算资源

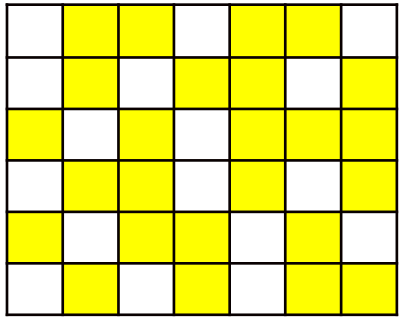
训练前:

完整模型参数



EBK 稀疏化

稀疏模型参数



位置掩码 (mask code)

0	1	1	0	1	1	0
0	1	0	1	1	0	1
1	0	1	0	1	1	1
0	1	1	0	1	0	1
1	0	1	1	0	1	0
0	1	0	1	0	1	1

Server使用 EBK[1] 层级稀疏化全局模型参数

[1] Evci, U., Gale, T., Menick, J., Castro, P. S., and Elsen, E. Rigging the lottery: Making all tickets winners. In International Conference on Machine Learning, pp. 2943–2952. PMLR, 2020.

Method

训练:

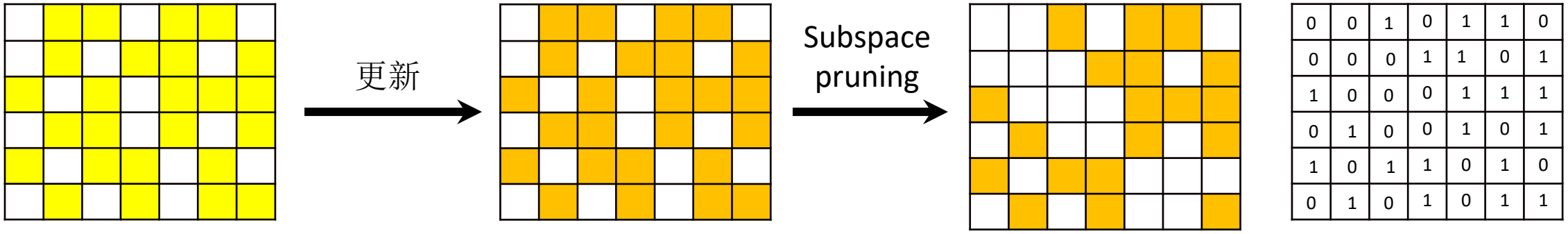
Clients: 接收到完整模型参数和 子空间mask code 使用本地数据对子空间模型参数训练:

第1轮训练:

$$\textbf{for } k = 0, 1, \ldots, K-1 \textbf{ do}$$
$$\boldsymbol{w}_{i,t,k+1} = \boldsymbol{w}_{i,t,k} - \eta \boldsymbol{m}_{i,t+\frac{1}{2}} \odot \nabla f_i(\boldsymbol{w}_{i,t,k}; \xi)$$

第1轮使用 Server初始化的mask code 提取子空间位置梯度更新对应参数

本地 *K* 次训练结束:



初始子空间参数　　更新后子空间参数 $\boldsymbol{w}_{i,t,K}$　　剪枝后子空间参数　　更新mask code

Subspace pruning: 将噪声或不重要的维度参数剪掉(进一步压缩子空间)。

剪枝依据: 更新后子空间参数绝对值 $|\boldsymbol{w}_{i,t,K}|$ 大小, 绝对值越小表明这些维度在聚合中的作用越小。

Method

第1轮本地训练结束, 客户端将完整本地模型参数和 mask code 上传 Server 用于聚合。

全局聚合过程:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{m}_{i,t+1} \odot \left(\boldsymbol{w}_t - \boldsymbol{w}_{i,t,K}\right)$$

剪枝后的 mask code

Server 使用上传的mask code 提取子空间位置参数的本地更新, 用于全局聚合。

全局模型更新之后, Server将各客户端的mask code 以及更新的全局模型下发, 用于下一轮训练。

Method

第2轮训练:

由于第1轮存在 Subspace pruning 过程导致子空间缩小， 各客户端在新一轮训练前需要对掩码进行适当恢复，使用 Subspace recovery 策略, 目的是维持子空间的大小稳定性。

恢复后的 mask code

$$\boldsymbol{m}_{i,t+\frac{1}{2}} = \boldsymbol{m}_{i,t} + \mathrm{ArgTopK}_{\alpha_{t-1}}(|\nabla f_i(\boldsymbol{w}_{i,t,0})|)$$

上一轮 mask code

$$\textbf{for } k = 0, 1, \ldots, K-1 \textbf{ do}$$
$$\boldsymbol{w}_{i,t,k+1} = \boldsymbol{w}_{i,t,k} - \eta \boldsymbol{m}_{i,t+\frac{1}{2}} \odot \nabla f_i(\boldsymbol{w}_{i,t,k}; \xi)$$

Subspace recovery 依据: 在本地进行一次训练对各维度参数梯度大小进行评估, 将梯度绝对值 $|\nabla f_i(\boldsymbol{w}_{i,t,0})|$大的位置掩码恢复成1, 梯度绝对值越大表明对模型性能贡献越大, 保留这些维度参与子空间训练。

恢复的掩码位置: 1. Server在稀疏化时编码为 0 的位置; 2. 上一轮Subspace pruning变为 0 的位置。

Method

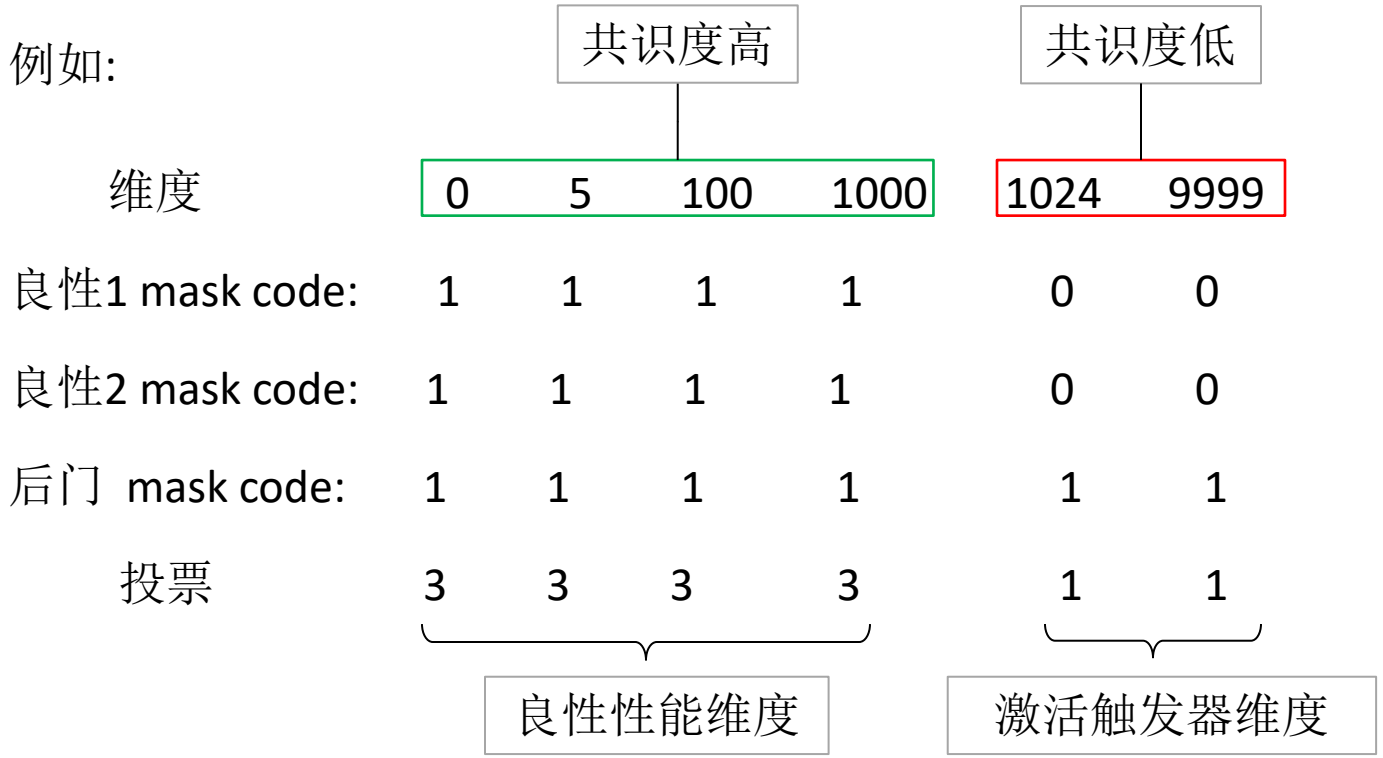
训练结束后:

共识融合(Consensus fusion, CF) 剪枝: 进一步清除残留后门参数, 增强模型鲁棒性。

由于每轮训练存在独立的 Subspace pruning 和 Subspace recovery 过程, 使得客户端之间的 mask code 不再一致:

直觉上是良性客户端和后门客户端训练任务上存在差异, 那么在训练结束之后最终的mask code 经过多轮累积有较大差别。

例如:



0, 5, 100, 1000维度的编码为1, 共识度高, 潜在和客户端之间的共享任务(良性任务)相关。

1024和9999维度共识度低, 可能与后门客户端的独立任务(后门任务)相关。Server将对1024和9999 维度全局模型参数进行剪枝。

Pseudocode

Algorithm 1 Lockdown defense

input Training iteration T; Local steps K; Learning rate η; Pruning/recovery rate α_t decayed by cosine annealing (Specially, $\alpha_{-1} = 0$, i.e., no recovery for first round); Random seed $seed$; Initial model $\boldsymbol{w}_0$.

output Clean model for deployment $\tilde{\boldsymbol{w}}_T$

main Server's Main Loop
  $\boldsymbol{m}_{i,0}$ = **SubspaceInit**($seed$) for $i \in \mathcal{M}$
  for $t = 0, 1, \ldots, T-1$ **do**
    for $i \in \mathcal{M}$ **do**
      Send $\boldsymbol{w}_{i,t,0} = \boldsymbol{m}_{i,t} \odot \boldsymbol{w}_t$ to client i
      Call Client i's main loop for training
      Received $\boldsymbol{w}_{i,t,K}$ and $\boldsymbol{m}_{i,t+1}$
    end for
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{m}_{i,t} \odot (\boldsymbol{w}_t - \boldsymbol{w}_{i,t,K})$
  end for
  $\tilde{\boldsymbol{w}}_T = \boldsymbol{w}_T \odot \mathcal{T}_\theta(\boldsymbol{m}_{1,T}, \ldots, \boldsymbol{m}_{M,T})$
  Deploy $\tilde{\boldsymbol{w}}_T$ for serving/inference.
end main
main Client's Main Loop
  Obtain local gradient $\nabla f_i(\boldsymbol{w}_{i,t,0})$
  $\boldsymbol{m}_{i,t+\frac{1}{2}} = \boldsymbol{m}_{i,t} + \text{ArgTopK}_{\alpha_{t-1}}(|\nabla f_i(\boldsymbol{w}_{i,t,0})|)$
  for $k = 0, 1, \ldots, K-1$ **do**
    $\boldsymbol{w}_{i,t,k+1} = \boldsymbol{w}_{i,t,k} - \eta \boldsymbol{m}_{i,t+\frac{1}{2}} \odot \nabla f_i(\boldsymbol{w}_{i,t,k}; \xi)$
  end for
  $\boldsymbol{m}_{i,t+1} = \boldsymbol{m}_{i,t+\frac{1}{2}} - \text{ArgBottomK}_{\alpha_t}(|\boldsymbol{w}_{i,t,K}|)$
  Send $\boldsymbol{w}_{i,t,K}$ and $\boldsymbol{m}_{i,t+1}$ to server
end main

1. 稀疏后的全局模型参数和 mask code发送给各个客户端

5. 全局聚合

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{m}_{i,t+1} \odot (\boldsymbol{w}_t - \boldsymbol{w}_{i,t,K})$$

6. CF共识全局模型剪枝

2. 本地数据获得梯度, 恢复部分掩码位置（Subspace recovery）

3. Subspace Training

4. 本地训练结束, 去掉参数较小的位置(Subspace pruning)

Experimental Results

Table 4: Defense efficacy with varying poison ratio p under CIFAR10.

Methods	Benign Acc (%) ↑					ASR (%) ↓					Backdoor Acc (%) ↑				
(IID)	clean	p=.05	p=.2	p=.5	p=.8	clean	p=.05	p=.2	p=.5	p=.8	clean	p=.05	p=.2	p=.5	p=.8
FedAvg	**91.0**	**91.4**	**91.1**	**91.0**	**90.8**	**1.6**	12.4	19.9	66.1	94.8	**88.5**	79.6	73.4	32.9	5.1
RLR	86.8	86.7	86.6	86.3	85.5	2.3	**2.4**	**2.4**	**4.3**	25.1	84.6	84.3	83.4	81.7	65.2
Krum	76.3	78.0	75.6	76.4	75.8	4.7	3.9	4.3	4.3	4.9	73.8	74.9	72.9	73.9	73.2
RFA	90.9	91.2	91.1	90.8	90.7	1.6	15.8	20.7	83.7	99.3	88.8	76.8	72.4	15.9	0.7
Trimmed mean	91.0	90.6	91.1	90.9	90.8	1.7	5.0	20.7	61.7	96.2	88.5	84.7	72.0	36.6	3.6
Lockdown	90.0	90.0	89.9	90.1	90.0	1.8	3.6	2.5	7.1	**4.0**	87.9	**85.8**	**86.6**	**83.7**	**85.6**

Methods	Benign Acc (%) ↑					ASR (%) ↓					Backdoor Acc (%) ↑				
(Non-IID)	clean	p=.05	p=.2	p=.5	p=.8	clean	p=.05	p=.2	p=.5	p=.8	clean	p=.05	p=.2	p=.5	p=.8
FedAvg	**89.0**	**89.2**	**89.3**	**88.8**	**88.7**	1.7	17.3	54.4	86.4	96.7	**85.9**	74.0	42.5	13.0	3.2
RLR	74.4	74.4	73.6	72.9	72.5	5.8	15.0	40.2	29.5	82.5	69.0	63.1	46.2	51.4	15.3
Krum	42.7	37.4	45.2	43.4	45.1	10.0	**5.2**	10.4	11.1	10.6	38.6	33.8	40.7	39.3	40.5
RFA	88.8	88.8	88.8	88.3	88.3	2.0	21.4	52.8	90.8	98.7	85.7	70.6	44.3	8.9	1.2
Trimmed mean	88.5	88.4	88.2	88.3	88.3	1.9	25.2	48.4	84.6	96.0	85.4	67.5	47.7	14.7	3.9
Lockdown	85.6	86.2	86.7	86.1	86.6	**0.9**	7.6	**3.6**	**3.4**	**3.3**	84.1	**79.5**	**82.3**	**82.2**	**82.8**

Fed-NAD: Backdoor Resilient Federated Learning via Neural Attention Distillation

1st Hao Ma
School of Computer Science and Technology
Shandong University
Qingdao, China
haoma@mail.sdu.edu.cn

2nd Senmao Qi*
School of Computer Science and Technology
Shandong University
Qingdao, China
senmao_qi@mail.sdu.edu.cn

3rd Jiayue Yao
School of Computer Science and Technology
Shandong University
Qingdao, China
jyyao@mail.sdu.edu.cn

4th Yuan Yuan
School of Computer Science and Technology
Shandong University
Qingdao, China
yyuan@sdu.edu.cn

5th Yifei Zou
School of Computer Science and Technology
Shandong University
Qingdao, China
yfzou@sdu.edu.cn

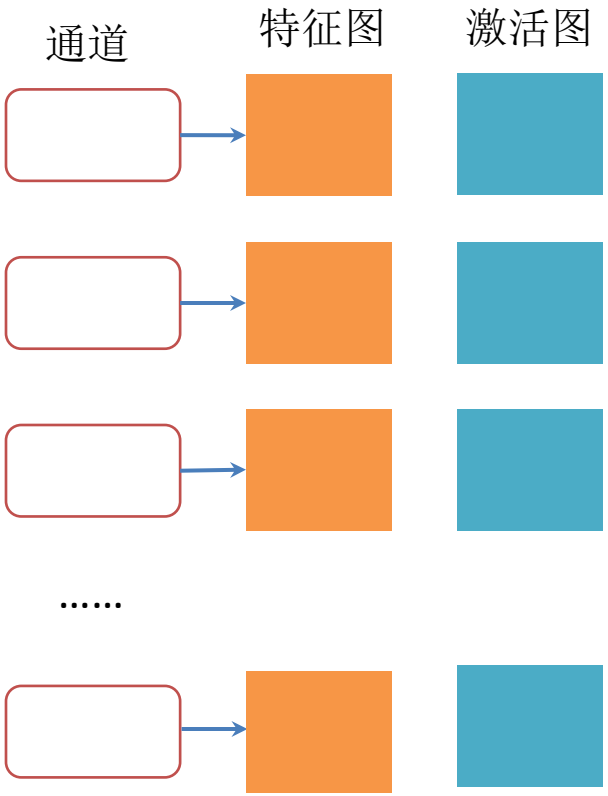
6th Dongxiao Yu
School of Computer Science and Technology
Shandong University
Qingdao, China
dxyu@sdu.edu.cn

IDS 2024

Method

利用客户端本地模型作为教师模型, 通过蒸馏其注意力/特征表示来指导全局(学生)模型更新, 从而削弱后门影响, 需要假设服务器拥有辅助数据。

1. 层级的注意力图(layer-level attention)



$$\mathcal{A}^{p}_{\text{mean}}\left(F^{l}\right)=\frac{1}{C}\sum_{i=1}^{C}\left|F_{i}^{l}\right|^{p}$$

C个通道平均，获得 L 层的注意力图

服务器使用辅助数据获得L层各通道激活图并逐元素取绝对值。

调节 p 值，使不同通道的激活分布差异更加显著

Method

$$L_{\mathrm{NAD}} = \sum_{l=1}^{L} \left\| \frac{\mathcal{A}_T^l}{\|\mathcal{A}_T^l\|_2} - \frac{\mathcal{A}_S^l}{\|\mathcal{A}_S^l\|_2} \right\|_2$$

归一化

教师模型和学生模型之间所有层之间的注意力差异之和

2. 输出分布的知识蒸馏(Output Distribution Knowledge Distillation)

KL 散度度量本地模型与全局模型的输出差异, 把所有客户端的知识传递给全局模型。

单教师模型与学生模型输出分布差异

$$L_{\mathrm{local}}^{k} = D_{\mathrm{KL}}(p_{\mathrm{local}} || p_{\mathrm{gloal}})$$

所有教师模型与学生模型平均输出分布差异

$$D_{\mathrm{KL}} = \frac{1}{N} \sum_{k=1}^{N} L_{\mathrm{local}}^{k}$$

Method

全局优化目标:

$$L_{\text{global}} = \beta(\mu L_{\text{CE}} + (1-\mu)D_{\text{KL}}) + \lambda L_{\text{NAD}}$$

1. Server使用有限数据对关键良性参数维度进行探查:

假设 Server 有辅助数据(clean data), 模拟客户端训练过程(服务器作为良性客户端提供模型信息参考)。评估与良性性能关联度高的维度, 参数维度重要性判断(Fisher score):

在分类模型 (softmax 输出) 下，某个参数 θ_i 的 Fisher 信息近似为：

$$F(\theta_i) = \mathbb{E}_{(x,y)\sim D}\left[\left(\frac{\partial \log p(y|x;\theta)}{\partial \theta_i}\right)^2\right]$$

其中：

- $p(y|x;\theta)$ 是模型的预测概率分布；
- 期望通常在数据集上取平均；
- Fisher 信息越大 → 参数越关键。

- **二阶 Fisher Information 方法**
- 本质：用 **梯度平方的期望**（近似 Hessian 的对角项）：

$$F(\theta) \approx \mathbb{E}_x\left[\left(\nabla_\theta \log p(y|x;\theta)\right)^2\right]$$

- **优势**：
  1. **稳定性更强**：平方期望相当于统计了梯度的方差，弱化了单次噪声影响。
  2. **全局重要性**：Fisher 衡量的是参数对预测分布整体的影响，而不是一次性损失。
  3. **理论支撑**：Fisher 来自信息论，直接刻画了"丢失该参数会导致模型预测分布信息损失多少"。
  4. **参数筛选更精准**：比单纯梯度大小更能识别哪些参数是关键（适合剪枝 / 防御）。

2. 聚合: 聚合时良性维度和非良性维度使用不同的聚合方式。

THANKS