Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

深度神经网络模型窃取防御方法

硕士研究生 张辰龙

2023年09月17日

问题回溯

- 总结反思
  - 对整个领域的宏观介绍不足
  - 算法原理讲解的不够详细，基础概念分析的不足
  - PPT制作存在瑕疵，英文字体格式存在问题
  - 忽略了对TIPO-PCDL的详细讲解
  - 缺乏对内容的总结

- 整改方案
  - 从宏观层面引入问题，对领域进行更加全面的概括
  - 详细讲解算法，干掉算法公式
  - 细化TIPO-PCDL的讲解分析，对每一个算法采用总-分-总形式讲解
  - 尽早完成PPT制作，对格式问题进行详细检查

相关内容

- 2021年01月03日，王琛，深度神经网络对抗样本防御方法
  - 算法1包含了一种类于“算两次”的思想；算法2从神经元激活状态进行分析
  - 具有一定的启发性，但对于攻击方有较为明确的场景限制
- 2023年03月05日，张辰龙，深度神经网络模型窃取检测
  - 算法1包含统计分析思想；算法2完全使用深度神经网络；算法3关注关联信息
  - 受限于模型窃取场景，检测的思想较为简单
- 2023年03月12日，邢凤桐，深度神经网络模型水印保护方法
  - 算法1提升了水印数据和任务数据的纠缠程度，使水印难以被去除；算法2使用哈希算法生成水印，提高水印的复杂程度
  - 水印方法的场景（模型窃取完成后）限制了其应用范围

内容提要

- 背景简介
- 基本概念
- 算法原理
  - GRAD2
- 优劣分析
- 总结
- 参考文献

内容提要


#模型窃取防御
专业
黑盒
Passive
Active
2022
梯度重定向
受害模型
保护模型安全!
替代模型
Proactive
Reactive
扰动
云
最优化
贪心算法

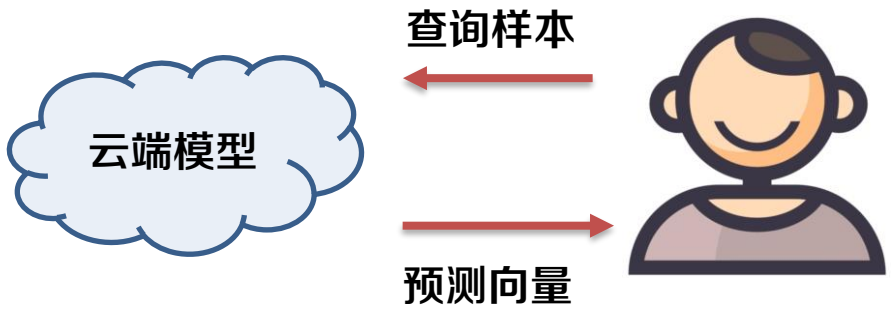
预期收获

- 预期收获
  - 了解深度神经网络模型防御整体框架
  - 理解深度神经网络模型窃取防御的算法原理及其理论问题
  - 理解深度神经网络模型窃取防御发展趋势
  - 了解深度神经网络模型窃取防御的重要意义

背景简介

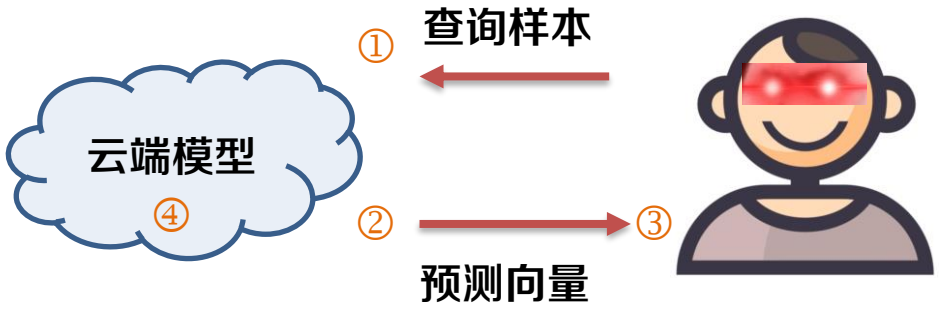
- 三个问题：什么场景下防御、为什么防御、怎么防御?
- 问题1：什么场景下防御
  - 部署在云端的模型（黑盒），向用户提供查询接口
  - 攻击者：构造样本→利用查询接口获得预测向量→利用样本、向量训练本地模型
- 问题2：为什么防御
  - 攻击者训练的本地模型与云端模型“很像”（体现在预测功能、分类边界等信息）
  - 云端模型查询要收费，攻击者利用本地模型避免缴费
  - 攻击者分析本地模型的结构，构造对抗样本，迁移到云端模型
  - 保护模型拥有者权益和数据隐私

反者道之动，弱者道之用!



背景简介

- 三个问题：什么场景下防御、为什么防御、怎么防御?
- 问题3：怎么防御（4个阶段、5类方法）
  - 4个阶段：Passive（测过就行）、Active（主动捣乱）、Proactive（实力证明）、Reactive（秋后算账）
  - 5类方法：
    - Passive：通过对查询样本流分析，判断是否为模型窃取
    - Active：对模型输出的预测向量进行扰动，使攻击者无法高效获取信息
    - Proactive：根据查询带来的信息泄露为查询用户出题（ HashCash ）
    - Reactive：验证模型版权
      - Watermark：向模型中添加认证信息
      - Fingerprint：采用模型固有的唯一认证信息



宏观着眼、微观着手!

基本概念

- 模型窃取
  - 攻击者构造无标签的“攻击者数据集”，利用预测模型的接口对数据集添加标签，利用带标签的“攻击者数据集”训练替代模型
- 模型窃取防御（Active类，后简称为防御）
  - 对模型输出预测向量添加扰动，减少攻击者获得的信息
  - 但仍保持输出预测向量的top-k不变
  - 最简单的例子：砍掉预测向量，仅输出标签（Tramer 2016）

- 攻击者绝对聪明理论
  - 攻击者会利用目标模型输出的一切信息，否则多数防御方法会不起效果

模型窃取的攻防是围绕着模型的输出进行的！

基本概念

- 符号代称
  - g：受害模型，被攻击的模型
  - f：替代模型，攻击者训练的模型
  - x：攻击者的查询样本，如无特殊说明，指代单个样本
  - y：g对x给出的最原始的预测向量（不添加扰动）
  - $\tilde{y}$：攻击者获得的，对y添加过扰动的预测向量

- 问题建模
  - 通过对y添加扰动，让攻击者利用$< x, \tilde{y} >$训练所得f的效果差
  - 流形约束：$y, \tilde{y} \in \Delta^K$，即$\{y \geqslant 0,\ \mathbb{I}^T y = 1\}$
  - 良性用户约束：$top_k(y)$与$top_k(\tilde{y})$的索引相同
    - 暗含扰动不能过大这一约束，即$|\tilde{y} - y|_1 \leq \varepsilon$，式中$\varepsilon$为常量，控制扰动范围

符号化的表示更有助于后续公式的推导！

算法原理——GRAD2

【ICML】

How to Steer Your Adversary: Targeted and Efficient Model Stealing Defenses with Gradient Redirection

算法原理——GRAD2

T	目标	使替代模型准确率降低
I	输入	查询样本（1组）
P	处理	1. 选定梯度重定向方向z 2. 估计替代模型的对数后验矩阵的雅可比矩阵 3. 确保预测向量top-k不变情况下添加扰动，使梯度更新向重定向方向z进行 4. 扰动达距离约束后停止
O	输出	带扰动的预测向量（1组）

P	问题	现有防御方法计算开销大、存在严重的效用权衡、存在信息泄露风险
C	条件	攻击者可以利用一切受害模型返回的信息（“聪明的”攻击者）
D	难点	1. 估计替代模型的log后验雅可比矩阵 2. 解决约束条件下重定向的最优化问题
L	水平	ICML 2022（CCF-A）

算法原理——GRAD2

- 攻击者建模
  - Q：攻击者的查询样本集
  - x：攻击者的查询样本，于Q中随机取样
  - θ：模型f的所有参数
  - f对(x, y)的交叉熵损失值记为$H\big(y, f(x)\big) = -\sum_i y_i log f(x)_i$
  - 攻击者对于f的梯度更新方向为$-\nabla_\theta H\big(y, f(x)\big) = \sum_i y_i \nabla_\theta log f(x;\theta)_i$
  - 记$G = \nabla_\theta log f(x;\theta) \in R^{n\times d}$，即$f$的$log$后验雅可比矩阵
  - 则梯度更新方向简化表示为$G^T y$
  - 参数更新：$\theta = \theta + G^T y$

维度分析：

$\theta \in R^d$

$y \in R^n$

$log f(x;\theta) \in R^n$

$\nabla_\theta log f(x;\theta) \in R^{n\times d}$

为什么要对攻击者建模？

攻击者的建模是白盒情况下进行的，真实防御场景下无法获取攻击参数

算法原理——GRAD2

- 优化问题构建
  - 目标：使攻击者模型梯度更新方向$G^T\tilde{y}$受控于防御者
  - 构造优化问题：$\max_{\tilde{y}} < G^T\tilde{y}, z >$，其中$<,>$表示向量内积，$z$为梯度偏移方向
  - 约束条件：$\tilde{y} \in \Delta^K$与$|\tilde{y} - y|_1 < \varepsilon$，1阶范数可以有效控制单个位的变化
  - 不严谨性：$\tilde{y}$的变化会影响$G^T\tilde{y}$的模，且影响$G^T\tilde{y}$与z的夹角，但期望最小化夹角
  - 优化的难度问题：$G \in R^{n\times d}$，维数较大，计算量大
  - 约束条件深入分析
    - 对y添加的扰动在单个维度上
    - 流形约束使维度的增减相同
    - 简化为线性优化问题，可类比于贪心算法进行求解

利用约束条件对优化问题进行化简，得到可解方案

论文中并没有提及不严谨性，但$\tilde{y}$模的变化幅度较小，所以使用此优化目标仍能得到较好的偏移效果

算法原理——GRAD2

- 优化问题求解（举例讲解）
  - $\max_{\tilde{y}} < G^T\tilde{y}, z >= \max_{\tilde{y}} \tilde{y}^T Gz$，视$Gz$为常量
  - $Gz = (2, 3, 1)^T$，$y = (0.2, 0.3, 0.5)^T$
  - $argsort$函数用于返回列表从小至大排列索引
  - 提问
    - 右图中的s的值是多少?
    - 右图中$\tilde{y}_{s_n}$的值是多少?
  - 思考
    - 右图中算法的截止条件是什么?
    - 右图中算法的主要思想是什么?

Algorithm 1 Gradient Redirection

Input: G, z, y, ϵ
Output: $\tilde{y}$
$\tilde{y} \leftarrow y$
$s \leftarrow \text{argsort}(Gz)$
$\tilde{y}_{s_n} \leftarrow \min(y_{s_n} + \epsilon/2, 1)$
$\lambda \leftarrow 0$
$t \leftarrow 1$
while $t < n$ **do**
  $\tilde{y}_{s_t} \leftarrow \max(y_{s_t} - (\epsilon/2 - \lambda), 0)$
  if $y_{s_t} - (\epsilon/2 - \lambda) > 0$ **then**
    Return $\tilde{y}$
  end if
  $\lambda \leftarrow \lambda + y_{s_t}$
  $t \leftarrow t + 1$
end while

敢于思考、善于思考、勤于思考、主动提问!

算法原理——GRAD2

- 优化问题求解（举例讲解）
  - $\max_{\tilde{y}} < G^T\tilde{y}, z >= \max_{\tilde{y}} \tilde{y}^T Gz$，视$Gz$为常量
  - $Gz = (2,3,1)^T$，$y = (0.2, 0.3, 0.5)^T$
  - $argsort$函数用于返回列表从小至大排列索引
  - 故$s = (3,1,2)^T$，$n = 3$，因此$\tilde{y}_{s_n} = 0.3$
  - 算法思想
    - 使Gz中的“3”对应的“0.3”尽可能增大
    - 使Gz中的“1”对应的“0.5”尽可能减小
  - 截止条件
    - Gz中的“3”对应的“0.3”增大至①1或②$0.3 + \varepsilon/2$
    - y中“0.5”被减至0后，如未截止，继续减小“0.2”

该求解方式的最优性可类比于一般贪心算法证明！

Algorithm 1 Gradient Redirection

Input: G, z, y, ϵ
Output: $\tilde{y}$
$\tilde{y} \leftarrow y$
$s \leftarrow \text{argsort}(Gz)$
$\tilde{y}_{s_n} \leftarrow \min(y_{s_n} + \epsilon/2, 1)$
$\lambda \leftarrow 0$
$t \leftarrow 1$
while $t < n$ **do**
  $\tilde{y}_{s_t} \leftarrow \max(y_{s_t} - (\epsilon/2 - \lambda), 0)$
  if $y_{s_t} - (\epsilon/2 - \lambda) > 0$ **then**
    Return $\tilde{y}$
  end if
  $\lambda \leftarrow \lambda + y_{s_t}$
  $t \leftarrow t + 1$
end while

① 流形约束

② 1阶范数约束

算法原理——GRAD2

- 注意！！！防御者无法看到攻击者训练参数，即G不可知
- 如何有效估计G?
  - 引入代理模型h，计算在h上的梯度偏移能否转移至f中，记为Transfer Performance
  - $TP = \frac{1}{Q}\sum_{x \in Q} \cos\left(\tilde{y}^T \nabla_\theta log f(x;\theta), z\right) - \cos\left(y^T \nabla_\theta log f(x;\theta), z\right)$

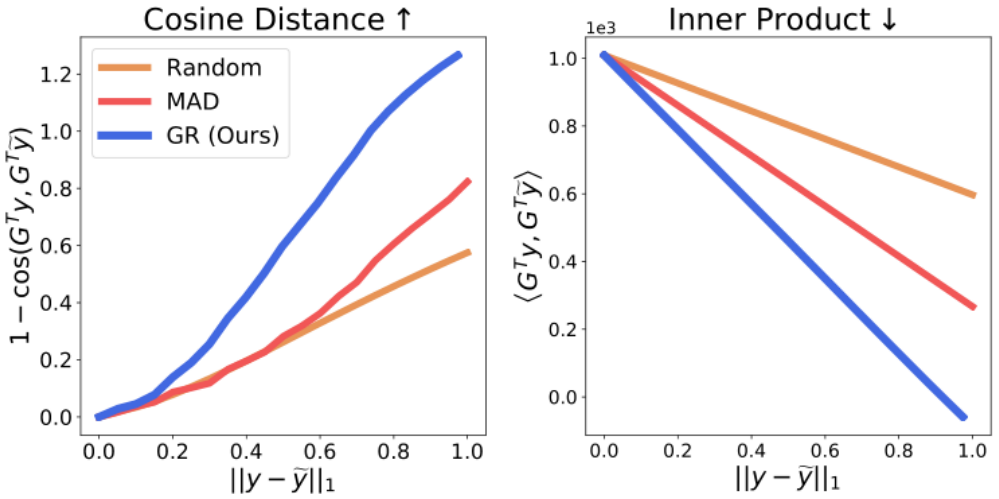
    $\tilde{y}$是defender通过h计算得出
    - $TP > 0$意味着$\tilde{y}^T \nabla_\theta log f(x;\theta)$与$z$的余弦相似度增加，转移有效
- 两个遗留问题
  - z的选取：(1)$z = \nabla_\theta H\left(y, h(x;\theta_h)\right)$；(2)$z = 1$

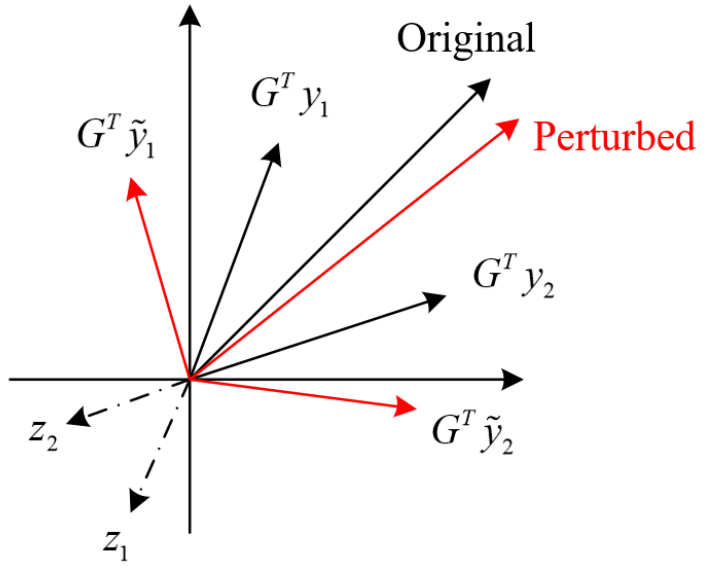
    P12:攻击者对于f的梯度更新方向为$-\nabla_\theta H\left(y, f(x)\right)$
  - h的训练
    - 训练数据集：(1)攻击者的查询数据；(2)受害模型的训练数据
    - 模型的拟合度：epoch在[0, 10, 20, 30, 40, 50]中各训练一次

对h与z采用先假设后验证的方式

算法原理——GRAD2

- z的选取实验验证
  - $z = \nabla_\theta H\left(y, h(x;\theta_h)\right)$，Orekondy 2020
  - $z = 1$，不依赖于任何信息
  - 自变量：1范数扰动范围
  - 因变量：$G^T\tilde{y}$与$G^T y$偏移程度
  - 结论：采用GRAD2产生的偏移最大
- 原因分析
  - 在批处理情况下，具有依赖性的扰动会产生破坏性结果，梯度会互相抵消
  - 固定方向扰动对于批处理具有不变性



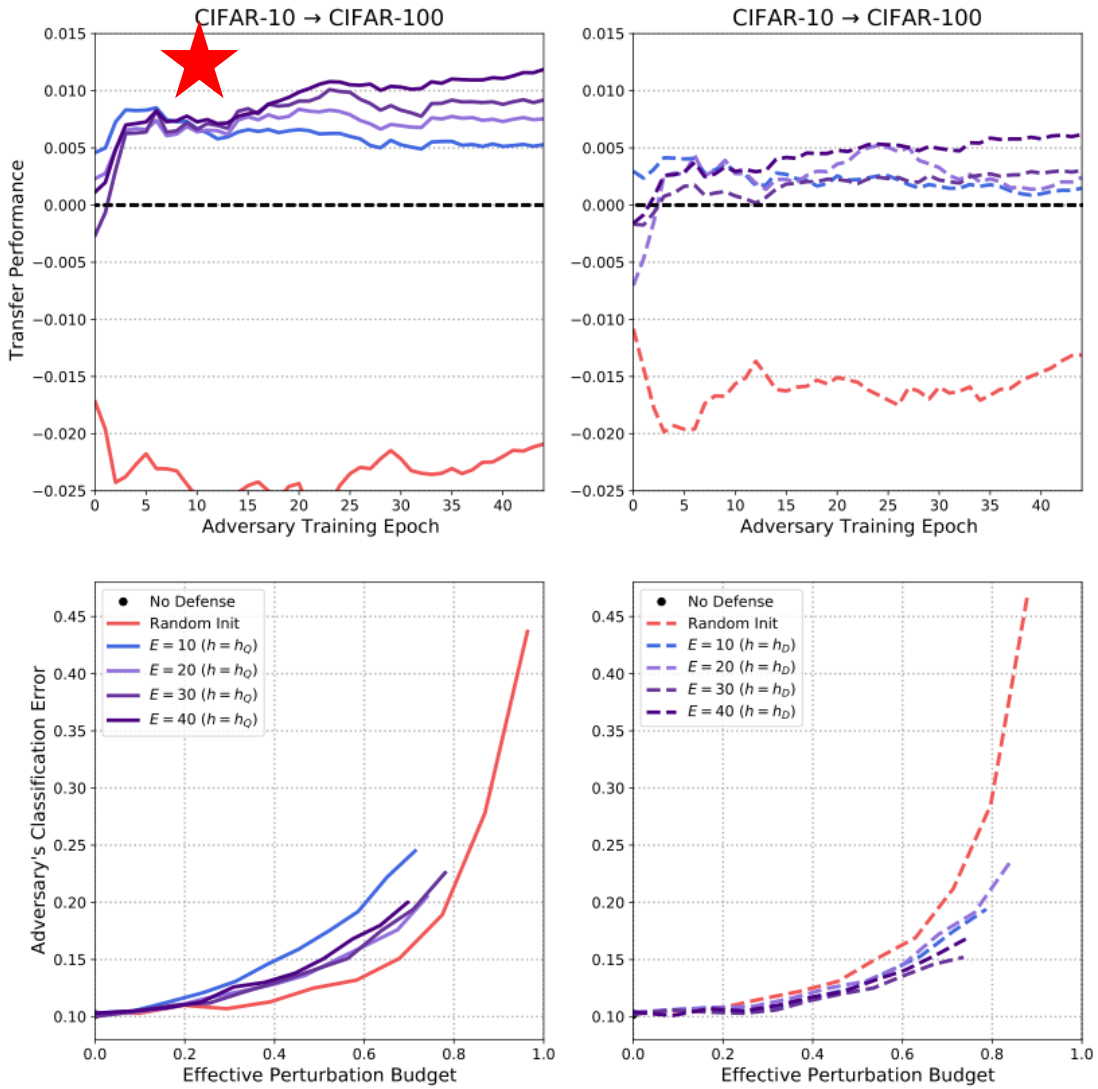


对于扰动方向，选取$z = 1$，计算简单且效果好

算法原理——GRAD2

- 代理模型h的选取实验验证
  - 训练数据集
    - 攻击者的查询数据（实线）
    - 受害模型的训练数据（虚线）
  - 模型的拟合度：epoch在[0, 10, 20, 30, 40, 50]中选取 → 理想的假设
  - 采用攻击者查询数据时，转移效果好
  - 在TP上，攻击者的训练次数与代理模型的训练次数存在相关性 为什么？
  - h训练的epoch为10时，f的分类误差最大

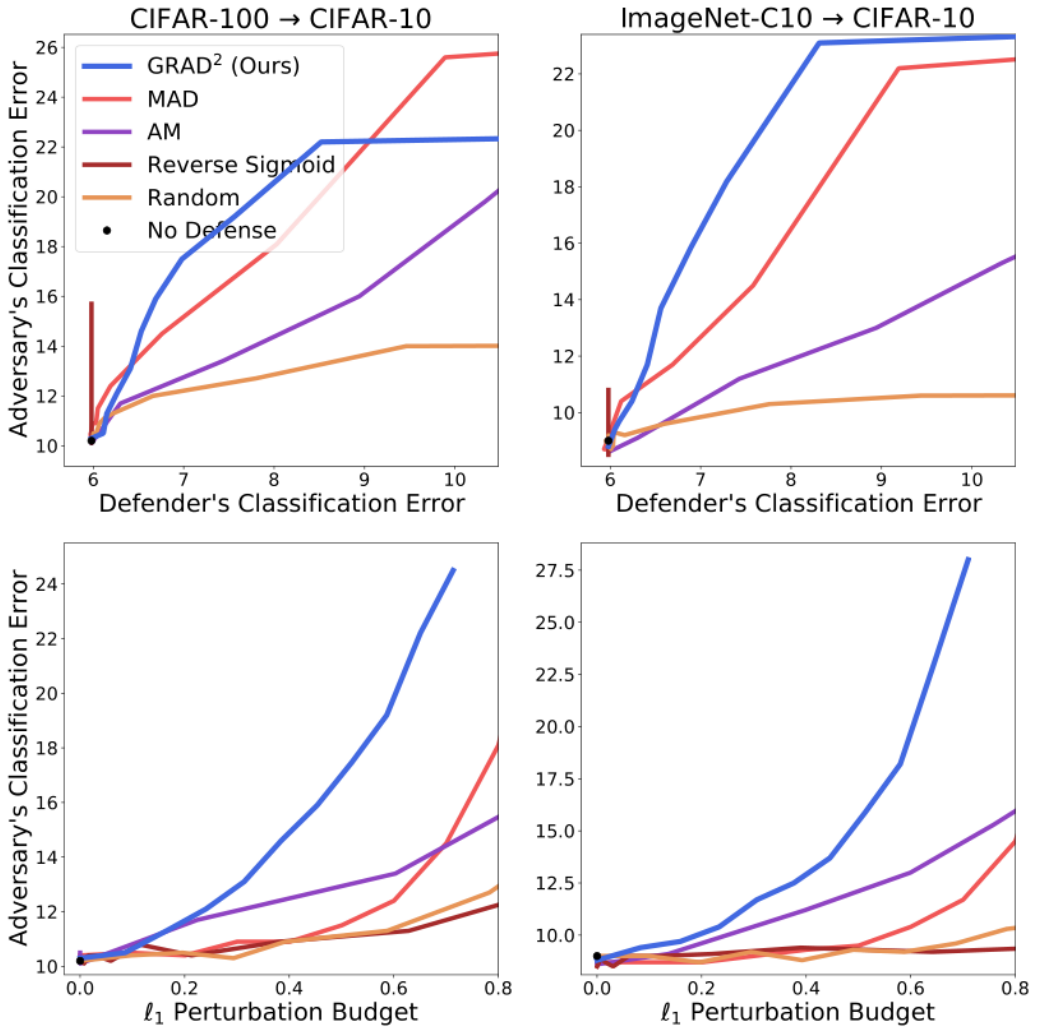
选取h的训练方式为epoch=10，采用攻击者数据集，能够得到更好的下游防御



实验结果——GRAD2

- 对比实验
  - MAD：替代模型f训练梯度重定向，易出现批处理下的梯度抵消
  - AM：利用OOD二分类模型先判断样本分布，以此控制扰动系数
  - RS：确保预测置信值大小顺序不变的情况下，尽可能混淆置信值
  - Random：返回随机预测向量

模型窃取防御本质上就是在考虑正常用户使用的情况下向预测向量添加扰动



实验结果——GRAD2

- 对比实验结果（Distribution-aware）
  - 在不同的受害模型误差范围（$\Delta Clf.Err$）和不同的扰动1范数约束（$l1\ Distance$）下，GRAD2均取得了非常好的下游防御效果
  - 为什么AM算法会在CUB200上表现很好呢？

查询集与受害模型训练集相关

2009). This gives us ImageNet-C10, ImageNet-C100, and ImageNet-CUB200, which are paired with their matching evaluation set and contain 183, 763, 161, 653, and 30, 000 examples respectively.

	ImageNet-C10 → CIFAR-10						ImageNet-C100 → CIFAR-100						ImageNet-CUB200 → CUB200					
	Δ Clf. Err			ℓ_1 Distance			Δ Clf. Err			ℓ_1 Distance			Δ Clf. Err			ℓ_1 Distance		
Method	1	2	5	0.1	0.2	0.5	1	2	5	0.1	0.2	0.5	1	2	5	0.1	0.2	0.5
Random	9.8	10.3	10.6	9.0	8.7	9.3	38.5	38.6	39.8	36.2	36.5	38.5	48.5	51.4	56.0	41.3	42.3	50.7
Reverse Sigmoid	-	-	-	9.0	9.1	9.3	-	-	-	36.3	36.8	38.0	-	-	-	41.2	42.6	45.9
Adaptive Mis.	10.4	11.9	16.3	9.0	9.6	12.1	38.2	40.6	46.6	36.4	37.4	41.8	53.8	**58.6**	**66.8**	**42.8**	**45.6**	53.8
MAD	12.6	16.4	22.6	8.7	8.7	9.5	43.0	46.8	49.2	35.9	36.9	42.6	49.6	52.3	56.0	41.7	42.6	51.7
GRAD^2 (Ours)	**16.4**	**21.5**	**23.4**	**9.5**	**10.1**	**15.5**	**43.4**	**47.6**	**53.0**	**36.5**	**37.7**	**44.1**	**54.1**	56.4	60.7	41.8	44.6	**55.6**

实验结果——GRAD2

- 对比实验结果（Knowledge-limited）
  - 在不同的受害模型误差范围（$\Delta Clf.Err$）和不同的扰动1范数约束（$l1\ Distance$）下，GRAD2均取得了较好的下游防御效果
  - 为什么AM算法会在更多数据集上表现很好呢?

查询集与受害模型训练集无关

Method	CIFAR-100 → CIFAR-10						CIFAR-10 → CIFAR-100						Caltech256 → CUB200					
	Δ Clf. Err			ℓ_1 Distance			Δ Clf. Err			ℓ_1 Distance			Δ Clf. Err			ℓ_1 Distance		
	1	2	5	0.1	0.2	0.5	1	2	5	0.1	0.2	0.5	1	2	5	0.1	0.2	0.5
Random	12.2	12.8	14.0	10.4	10.5	11.1	50.9	52.1	54.5	46.5	47.8	50.6	53.8	58.1	65.1	43.1	45.2	57.1
Reverse Sigmoid	-	-	-	10.7	10.5	11.1	-	-	-	46.0	46.9	50.8	-	-	-	42.7	44.2	49.7
Adaptive Mis.	12.7	14.3	21.7	**10.8**	11.5	12.9	47.6	51.0	60.2	**47.5**	**50.6**	**61.2**	**64.7**	**70.6**	-	43.3	45.6	53.4
MAD	15.1	18.0	**25.9**	10.5	10.4	11.5	52.2	53.6	58.6	45.1	46.7	52.0	55.4	57.7	62.1	**43.4**	**47.6**	57.1
GRAD^2 (Ours)	**17.5**	**20.5**	22.4	10.6	**11.7**	**17.0**	**55.2**	**59.3**	**63.7**	46.3	48.0	56.8	57.9	60.7	**65.2**	42.5	46.1	**58.3**

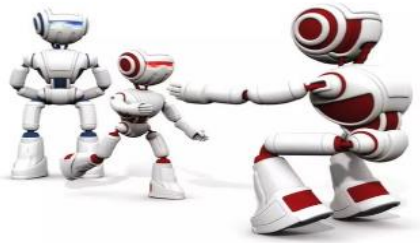
算法总结——GRAD2

- 算法思路
  - 详细分析攻击者训练替代模型的过程，给出白盒计算公式
  - 构造优化问题，利用输出预测标签对替代模型训练梯度进行重定向
  - 考虑现实因素，替代模型参数不可知，因此利用代理模型进行梯度估计
  - 以实验形式确定重定向方向和代理模型训练参数
- 算法优点
  - 以贪心算法形式解决了实际不可解的最优化问题
  - 常量重定向方向极大减少计算量，对批处理具有不变性
- 算法缺点
  - 假设查询样本一次性输入具有不合理性
  - 如果攻击者“不聪明”，只利用输出的硬标签，则不会被重定向

总结

总结

- 防御的发展趋势及原因
  - Passive：算法局限性大，场景限制严格，容易漏检和误检，论文少
  - Active：无论如何扰动，都要有保证正常用户使用的约束，指标提升难，论文渐少
  - Proactive：对于计算量的评估，实质仍依赖于Passive，论文少
  - Reactive：可操作空间大，评价指标多，论文渐多
- 防御的实际场景
  - Passive：对单个用户的检测，面对多用户时计算代价较高
  - Active：通过改变输出向量，保证正常用户使用，使攻击者获取信息变少，能面向不同的实际场景，最具有现实意义
  - Proactive：防御效果依赖于评估效果
  - Reactive：对两个公开的模型进行版权验证，目前没有实际场景



尽管Active指标提升难，但最具有现实意义!

总结

- 模型窃取防御的意义
  - 直接意义
    - 保护深度神经网络模型拥有者的权益，包含经济效益和数据效益
    - 保护深度神经网络数据安全，防止数据泄露，如成员推理攻击等
  - 深远意义
    - 促进深度神经网络领域发展，推动大数据的发展
    - 促进数据交流与共享
    - 提高模型在各类应用中可信度和安全性，使社会愿意去接纳和认可

保护深度神经网络模型安全，任重而道远!

前沿发展

- 模型窃取防御的前沿发展
  - 差分隐私
    - 应用差分隐私可以使攻击者更加难以从模型中推断出敏感信息。可以利用差分隐私技术在模型训练阶段提升其抗攻击能力
  - 硬件保护
    - 在芯片或处理器级别实施硬件隔离，保护模型权重和结构
  - 联邦学习
    - 共享模型更新信息，但不共享原始数据，有助于减少对中央模型的攻击风险
  - 监控和检测
    - 建立更加完善、算力更强的监控系统，快速及时的发现异常行为

防御要跟着攻击方法不断演进，攻与防在螺旋式上升发展!

参考文献

[1] Mazeika M, Li B, Forsyth D. How to steer your adversary: Targeted and efficient model stealing defenses with gradient redirection[C]. International Conference on Machine Learning. PMLR, 2022: 15241-15254.

[2] Orekondy T, Schiele B, Fritz M. Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks[C]. 8th International Conference on Learning Representations. OpenReview. net, 2020.

[3] Lee T, Edwards B, Molloy I, et al. Defending against neural network model stealing attacks using deceptive perturbations[C]. 2019 IEEE Security and Privacy Workshops (SPW). IEEE, 2019: 43-49.

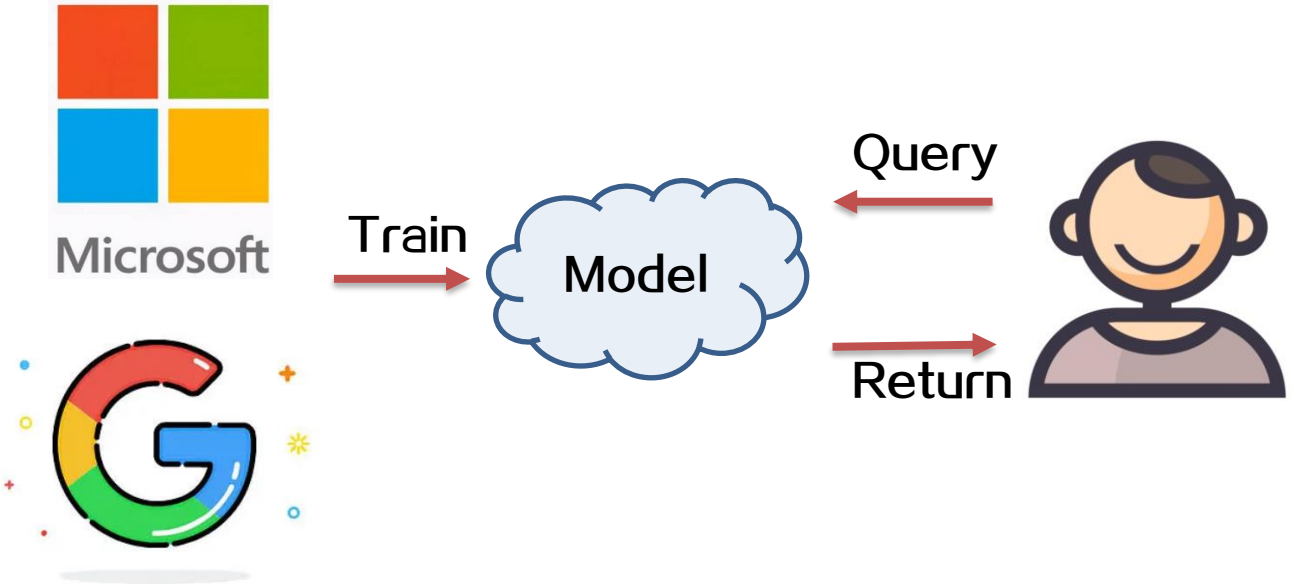
道德经

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！

附录

- 云端模型
  - 深度神经网络技术发展迅速，在图像识别、自动驾驶等领域发挥重要作用
  - 深度神经网络模型训练过程繁琐、花销昂贵
  - 微软、谷歌等大型公司将模型部署在云端服务器，仅向用户提供预测接口
  - 研究表明，预测接口仍能泄露模型的大量信息



附录

- 模型窃取
  - 概念：攻击者利用目标模型查询接口泄露的信息，窃取目标模型的参数或功能
  - 流程：攻击者构造无标签的“攻击者数据集”，利用目标模型的查询接口对数据集添加标签，利用带标签的“攻击者数据集”训练替代模型
  - 目的：
    - 免费使用模型功能
    - 进行白盒对抗攻击
  - 危害：
    - 损害模型拥有者的商业利益
    - 侵犯模型的隐私信息