中国科学: 信息科学 2023 年 第 53 卷 第 5 期: 931–945
SCIENTIA SINICA Informationis
论文

基于采样和加权损失函数的模型窃取攻击方法

王熠旭[1], 李杰[1], 刘弘[2], 王言[5], 徐明亮[3], 吴永坚[4], 纪荣嵘[1*]

1. 厦门大学信息学院, 厦门 361005, 中国
2. National Institute of Informatics, Tokyo 101-8430, Japan
3. 郑州大学计算机与人工智能学院, 郑州 450001, 中国
4. 腾讯优图实验室, 上海 200235, 中国
5. Pinterest, Seattle 98101, USA
* 通信作者. E-mail: rrji@xmu.edu.cn

收稿日期: 2022–01–17; 修回日期: 2022–09–05; 接受日期: 2022–09–13; 网络出版日期: 2023–05–12

国家杰出青年科学基金 (批准号: 62025603)、国家自然科学基金 (批准号: 62176222, 62176223, 62176226, 62072386, 62072387, 62072389, 62002305) 和广东省基础与应用基础研究基金 (批准号: 2019B1515120049)、福建省自然科学基金计划 (批准号: 2021J01002) 资助项目

摘要 模型窃取攻击旨在获得一个和目标受害模型功能相似的替代模型. 现有的方法主要采用数据生成或数据选择方法和交叉熵损失函数去获得一个较好的攻击效果. 据此, 本文着重研究了攻击过程中这两个极为重要的模块: 数据采样和损失函数. 同时, 本文提出了一个新颖的模型窃取攻击方法 S&W, 其包含了一种新的采样策略和一个精心设计的加权损失函数. 首先, 新的采样策略更加关注于从受害者模型中获得更多信息的重要样本. 与此同时, 本文通过引入 k-Center 算法达到选择样本的多样性的目的. 其次, 受到经典 Focal 损失函数的启发, 本文设计了一种新的加权损失函数. 该损失函数主要关注于受害者模型和替代模型对于相同输入所给出的输出之间的差异, 从而促使替代模型模拟受害者模型. 在 4 个常用的数据集上, 我们通过实验证明了本文提出的方法的有效性. 相比于之前最好的方法, 本文方法最高有 5.03% 的性能提升.

关键词 计算机视觉, 模型窃取攻击, 对抗攻击, 主动学习, 知识蒸馏

1 引言

云端部署 (cloud deployment) 是当前流行的利用机器学习模型为用户提供服务的一种方式. 基于云端服务, 用户无需拥有模型便可以通过相应的应用程序接口 (application programming interface, API) 享受云端部署的模型所提供的服务. 然而, 最近的研究表明 [1~4]: 通过恶意的查询请求, 攻击者可以威胁到这些应用程序接口背后模型的知识产权, 这些威胁包括成员推断攻击 [3]、模型结构提取攻

引用格式: 王熠旭, 李杰, 刘弘, 等. 基于采样和加权损失函数的模型窃取攻击方法. 中国科学: 信息科学, 2023, 53: 931–945, doi: 10.1360/SSI-2022-0029
Wang Y X, Li J, Liu H, et al. Model stealing attack based on sampling and weighting (in Chinese). Sci Sin Inform, 2023, 53: 931–945, doi: 10.1360/SSI-2022-0029

击 [1] 和模型窃取攻击 [2] 等. 其中, 模型窃取攻击是上述方法中最具威胁的攻击方法之一. 它旨在去创造一个和受害模型具有相似功能的替代模型. 随后, 攻击者可以利用这个替代模型去给其他用户提供服务或者产生具有更强迁移性的对抗样本, 从而侵犯云端服务商的知识产权. 为了更好地探究这种类型攻击的内在机理, 从而为云端服务应对这种攻击提供防御性指导与帮助, 本文主要关注如何实现 API 查询次数最小化的模型窃取攻击任务.

为了达到这个目的, 我们首先回顾了之前的工作, 发现它们大都由两个重要阶段组成, 即构建迁移集合和利用迁移集合训练替代模型. 在第一个阶段中, 攻击者通常会构建一个近似拟合受害者训练数据集数据分布的子数据集, 这个子数据集常常被称为迁移集合. 目前主要有两种构建迁移集合的方法, 分别是基于数据生成的方法和基于样本选择的方法. 其中, 基于数据生成的方法 [5,6] 通常使用一些专门设计的对抗生成网络 (generative adversarial network, GAN) [7] 来合成用于查询的样本. 虽然这种类型的方法通常能够取得较好的攻击效果, 但是目前对抗生成网络的训练仍比较困难, 即需要较多的迭代次数, 这也就意味着较大的查询开销. 而基于样本选择策略的方法 [2,8] 往往需要先收集一个有标签或无标签的样本池, 之后从这个样本池中选择重要的样本查询获得输出. 在第二个阶段中, 攻击者会使用从第一个阶段获取的迁移集合训练替代模型, 之前的工作都会通过最小化交叉熵损失函数 (cross-entropy loss function, CE) 来使替代模型拟合迁移集合.

在回顾了模型窃取攻击的两个关键阶段之后, 本文认为传统的交叉熵损失函数不够有效, 应该在计算总损失时给不同的样本分配不同的权重, 这有助于更好地利用迁移集合, 从而在较少的查询次数下获得更好的结果. 为此, 本文提出了一种简单但有效的方法, 由于这种方法同时考虑了样本选择 (sampling) 和加权损失函数 (weighting), 我们将其命名为 S&W. 具体而言, 对于模型窃取攻击的第一个阶段, 本文提出了一种新颖的基于深度主动学习框架 (deep active learning framework) 的样本选择策略, 这个样本选择策略更加关注包含更多受害者模型信息的重要样本. 同时, 该样本选择策略中进一步采用 k-Center 算法来保证所选择样本的多样性. 对于第二个阶段, 本文提出了一种加权交叉熵损失函数, 命名为差异损失函数 (differ-loss function). 这个差异损失函数受启发于经典的暗知识理论 (dark knowledge) 和 focal 损失函数, 使用受害者模型和当前替代模型输出之间的统计差异来计算样本的权重值. 更高的权重意味着对于同一张输入图片, 两个模型的输出有更大的差异, 差异损失函数将这样的图片样本看作难样本. 通过设计这样的权重计算方式, 差异损失函数逐步增加难样本在训练中所起到的作用, 从而使得替代模型更好地模拟受害者模型. 最后, 本文将这两个部分整合为 S&W 算法, 并且在 4 个常用的数据集上展示了 S&W 算法的效果. 实验证明, S&W 算法可以在仅使用 3 万次查询的条件下获得具有 77.42%~98.90% 的相似度的替代模型. 与现有最好的方法相比, S&W 算法最高有 24.18% 的性能提升. 同时, 在 S&W 算法获得的替代模型上生成的对抗样本 (adversarial example) 取得了 76.76% 的迁移攻击成功率, 相比于其他对比方法最高提升了 14.48%.

本文结构如下: 第 2 节介绍模型窃取攻击的相关方法和研究现状. 第 3 节介绍本文所需的基础知识. 第 4 节介绍本文提出的基于采样和加权损失函数的模型窃取攻击方法. 第 5 节实验证明我们方法的有效性. 第 6 节对本文的工作进行了总结.

2 相关工作

攻击者进行模型窃取攻击的主要目的是在只能通过在线应用程序接口获得受害者模型输出的情况下, 获得一个与受害者模型功能尽可能相似的替代模型. Papernot 等 [5] 最先发现可以通过多次查询获得黑盒模型的输出的形式来窃取黑盒模型的功能. 之后, 研究人员便开始关注这个问题并不断提

出新的方法. 回顾这些方法, 我们发现它们大都由两个关键阶段组成: (1) 构建一个迁移集合; (2) 使用迁移集合训练替代模型. 其中, 构建迁移集合的方法分为两类, 即基于数据合成的方法和基于数据选择的方法. 基于数据合成的方法 [5,6] 通常使用对抗生成网络 [7] 等生成方法合成虚拟的数据集, 查询受害者模型获得虚拟数据集中样本对应的标签, 之后在虚拟数据集上交替训练替代模型和生成模型. 而基于数据选择的方法则首先收集大量的自然图片样本构建一个数据池, 之后根据一定数据选择策略从中选择出更有价值的样本. 例如, Knockoff-Nets [2] 算法使用强化学习 [9](reinforcement learning, RL) 作为数据选择策略; ActiveThief [8] 使用经典的主动学习方法 (基于不确定性的策略 [10]、k-Center 策略 [11] 和 DFAL 策略 [12]) 作为数据选择策略. 在挑选出用于迁移的数据样本之后, 攻击者通过遍历受害者模型的输出获得这些样本对应的标签, 并利用这些样本训练替代模型.

由于目前对抗生成网络的训练开销较大, 需要较多的查询迭代次数, 进而导致这种攻击方法的整体开销很大. 因此, 本文重点关注基于数据选择的模型窃取攻击方法. 当前的基于数据选择的模型窃取攻击方法大部分可以视为一种特殊的基于数据池 [13] (pool-based) 的主动学习 (active learning, AL) 策略. 攻击者从未标注数据池中选择样本, 之后查询受害者模型获得样本的标注信息. 目前广泛使用的主动学习策略包括基于不确定性的策略、k-Center 策略和 DFAL (margin-based) 的策略. 主动学习策略的目的是尽可能减少标注数据的代价, 即使用较少的标注数据训练出更高测试准确率的模型. 这虽然与模型窃取攻击的目标有一定的相似, 但是本文认为这两个问题有着根本的不同: 在模型窃取攻击任务中, 未标注数据池中数据的分布与受害者模型训练集中的数据分布存在一定差异, 而传统的主动学习任务中不存在这个问题.

模型窃取攻击任务与知识蒸馏 (knowledge distillation, KD) 任务也存在一定关联. 知识蒸馏目前在机器学习领域中得到了广泛的应用 [14~16], 它将知识从一个更强大的教师网络 (teacher model) 迁移到一个小的学生网络 (student model). 本文设计的攻击方法受到了知识蒸馏工作的启发但是也存在一些差异, 差异主要体现在: 在知识蒸馏任务中, 教师模型的信息都是公开的, 即我们知道它的模型结构、训练数据集和训练时使用的超参数等, 而在模型窃取攻击任务中, 我们只能获取受害者模型的输出, 其他一无所知. 此外, 选择合适的替代模型结构对于更好的模拟受害者模型的功能是很重要的, 我们会在本文的实验部分讨论不同的模型结构对最终结果的影响.

模型窃取攻击获得的替代模型除了直接给其他用户提供服务外, 还可以被用来生成具有更强迁移能力的对抗样本. 对抗样本的概念最早由 Szegedy 等 [17] 提出, 他们观察到在输入样本上添加人眼无法察觉的微小噪声能够导致模型对样本的错误分类. 在此之后, Goodfellow 等 [18] 观察到对抗样本具备可迁移的特性, 即在一个模型上生成的对抗样本可以攻击其他模型. 因此, 在黑盒攻击的场景下, 可以在白盒模型上生成对抗样本, 然后使用其攻击其他黑盒模型. 与此同时, 相关工作 [5,6,8] 也验证了在模型窃取攻击中获得的替代模型上生成的对抗样本, 攻击受害者模型时会有更强的迁移性. 因此, 本文也将对比我们的方法和其他方法获得的替代模型产生的对抗样本的迁移能力.

3 预备知识

本节简单介绍模型窃取攻击中的基础概念以及常规的做法. 给定一个受害者模型 $f:[0,1]^d \mapsto \mathbb{R}^N$, 它将 d 维输入图片 x 映射为 N 个类别的置信度值 (confidence score). 模型窃取攻击的目的是优化一个替代模型 $\hat{f}:[0,1]^d \mapsto \mathbb{R}^N$ 使它和受害者模型 f 有着相似的功能, 即对于任意测试集中的样本, 找到

能够最小化替代模型和受害者模型输出差异的模型参数 θ_S:

$$\text{argmin}_{\theta_S} \mathcal{P}_{x\sim D_t}\left(\text{argmax}_i\, f_i(x) \neq \text{argmax}_i\, \hat{f}_i(x)\right), \tag{1}$$

其中 D_t 代表测试数据集 (test dataset), 通常, 测试数据集和受害者训练数据集 (training dataset) D_V 遵从相同的分布. 对于攻击者来说, D_V 通常是无法接触到的. 为了解决这个问题, 之前的方法大都是基于生成模型的生成方法和基于样本选择的选择方法. 其中, 基于生成模型的方法使用一些常见的生成模型 (GAN, VAE 等) [7, 19] 生成样本并查询受害者模型, 利用查询结果交替更新替代模型和生成模型. 然而由于当前生成模型训练需要较多的迭代次数, 导致这种类型的方法往往查询次数很大, 这就意味着很大的攻击开销. 因此本文只考虑基于样本选择的方法.

基于样本选择的方法首先需要收集大量未标注的图片作为未标注数据池 (unlabeled pool) (也被称为攻击数据集 (attack dataset)). 这些图片可以从互联网中收集获得并且不需要人工进行标注, 因此不会花费很大的代价. 之后, 需要使用样本选择算法从数据池中选择出更有价值的样本, 查询获得这些样本对应的受害者模型的输出, 构成样本对 (也被称为迁移集合 (transfer set)), 之后使用迁移集训练替代模型. 通常, 样本选择的算法会迭代得选择, 即训练和选择交替进行. 例如, 在第 k 次迭代时选择 b_k 数量的样本, 获得迁移集 $\{(x_i, f(x_i))|i = 1, \ldots, b_k\}$. 替代模型在迁移集合上通过最小化损失函数 $\mathcal{L}$ 进行优化:

$$\mathbb{E}_{x\sim D_T}[\mathcal{L}(f(x), \hat{f}(x))], \tag{2}$$

其中 D_T 代表迁移集. 常用的损失函数为交叉熵损失函数 (CE), 形式如下:

$$\text{CE}(f(x), \hat{f}(x)) = -\sum_{i}^{N} f(x)_i \cdot \log\left(\hat{f}(x)_i\right). \tag{3}$$

4 基于采样和加权损失函数的模型窃取攻击方法

4.1 迁移集合的构建方法

本小节主要介绍本文提出的模型窃取攻击方法中迁移集合的构建方法. 由第 3 节的背景介绍可以知道, 未标记池中的样本和受害者模型训练数据集中的样本往往是服从不同分布的, 因此难以保证其中的每个样本都有助于迁移受害者模型的知识. 那么便引出了一个关键的问题: 什么样的样本是更重要或者是更有助于迁移受害者模型的知识?

为了解决这个问题, 本文的核心观点之一是: 含有更多受害者模型信息的样本更有助于知识的迁移. 这需要分别考虑两种情况: (1) 使用随机噪声; (2) 使用受害者模型的训练数据集作为攻击数据集, 这两种情况分别代表了攻击数据集中的样本所能包含的受害者信息的上界和下界. 显然, 选择的样本中包含受害者的信息越多, 越能够帮助替代模型获得受害者模型的功能. 那么, 如何衡量样本中所包含的信息含量就显得十分重要. 根据暗知识理论 (dark knowledge theroy) [16], 模型输出的概率值中暗含了模型的知识, 并且这些知识有助于获得模型的功能. 因此, 概率向量中包含更多信息的样本对攻击过程更加重要. 然而, 受害者模型的输出向量只有查询后才能获得, 如果查询所有的样本并计算它们对应的信息含量将带来巨大的开销. 我们选择使用替代模型的输出来近似计算样本的重要程度.

为了判断样本的重要程度, 本文使用 Zhang 等 [20] 提出的度量指标. 该度量指标已被证明比广泛使用的熵 (entropy) 更加有效, 其具体形式如下:

$$\text{Importance}(v) = 1 - \frac{\text{MIN}\,\text{Var}(v)}{\text{Var}(v)} \times \max(v), \tag{4}$$

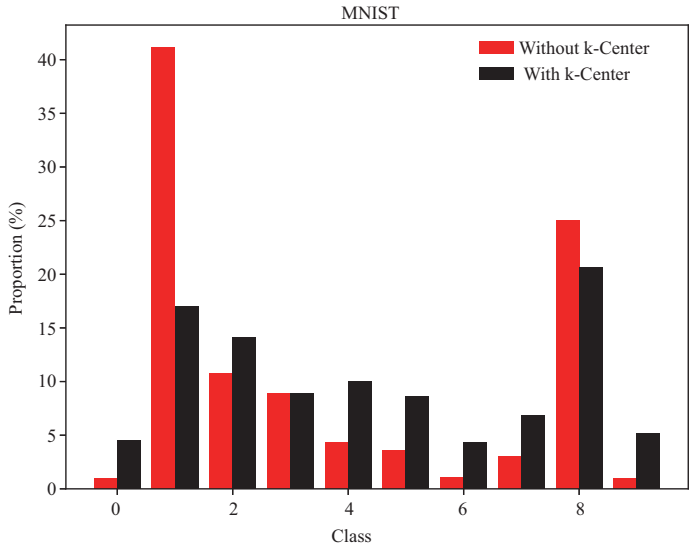

图 1 (网络版彩图) 使用/不使用 k-Center 算法所选择的样本的伪标签分布

Figure 1 (Color online) Distribution of pseudo-label's class of samples selected with/without k-Center algorithm

其中, $\mathrm{MIN\,Var}(v)$ 是向量 v' 的方差, 并可以通过下式计算得到:

$$\mathrm{MIN\,Var}(v) = \mathrm{Var}\left(v'\right) = \frac{1}{N}\left(\left(\frac{1}{N} - \max(v)\right)^2 + (N-1)\left(\frac{1}{N} - \frac{1-\max(v)}{N-1}\right)^2\right). \tag{5}$$

向量 v' 的最大值和向量 v 相同, 其他元素则都为 $\frac{1-\max(v)}{N-1}$. 式 (4) 包含如下 3 个性质: (1) 它的值域为 [0, 1); (2) 它和概率向量最大值是负相关的; (3) 它和概率分布的集中程度是正相关的. 其中, 值域为 [0, 1) 使得这个指标可以很方便地选择出更加重要的样本, 与集中程度的正相关性可以反映出不同类别中的相似性中所包含的信息多少. 基于上述的性质, 这个度量指标可以更好地反映样本的重要程度.

此外, 我们注意到了另一个可能影响算法性能的关键问题. 如果直接选择重要性最高的样本, 那么所选择的样本很可能会过于集中在少数几个类别. 如图 1 所示, 直接按照最大重要性原则选择的样本在类别分布上较为不均匀, 例如, 属于第 2 类的样本占到了 40% 的比例, 而属于第 7 类的样本只占到了 1% 的比例. 这种类别上的不均匀会导致在这个迁移集合上训练的替代模型在样本数较少的类别上表现很差, 从而影响了算法的整体性能. 为了解决这个问题, 本文采用 k-Center 算法 [11] 以保证选择样本的多样性. 相比于直接按照预算 (budget) 选择重要性最高的样本, 我们的算法会先选择预算一定倍数的样本作为候选集. 具体来说, 本文引入了一个超参数 $\mathcal{M} \geqslant 1$, 并且选择 $\mathcal{M} \times$ budget 数量具有最高重要性的样本, 这部分样本被称为候选集 D_C. 在此之后, 本文提出的算法将使用 k-Center 算法计算 D_C 的核心集 (core-set), 这个核心集便是算法当前迭代的迁移集 D_T. 这个核心集可以看作是重要样本一个具有代表性的子集, 从而保证最终算法选择的样本既是重要的, 同时也足够分散和多样. 通过 k-Center 算法迭代地选出 budget 数量的样本应满足下式:

$$\mathrm{argmax}_{x_c \in D_C} \min_{x_t \in D_T} \left\| \hat{f}\left(x_c\right) - \hat{f}\left(x_t\right) \right\|^2. \tag{6}$$

具体的 k-Center 算法如算法 1 所示, 其中需要注意的是, 在 k-Center 算法过程中 D_C 和 D_T 是不断迭代更新的.

算法 1 k-Center 算法

输入: 重要样本子集 D_C, 之前迭代的迁移集 D_T, 当前迭代的替代模型 $\hat{f}$, 当前的预算 budget;

```
while i < budget − 1 do
    根据式 (6) 挑选样本 x_i;
    D_T ← D_T ∪ x_i;
    D_C ← D_C/x_i;
    i = i + 1;
end while
```

输出: 更新后的迁移集合 D_T.

图 1 中对比了是否使用 k-Center 算法所选择样本的类别分布, 可以很明显地观察到使用 k-Center 算法所选择的样本更加多样, 即类别分布上更加均匀. 此外, 超参数 $\mathcal{M}$ 的选择也是很重要的. 如果 $\mathcal{M}=1$, 那么我们的采样策略便退化为了初始的基于最大重要性原则的选择方法, 而如果 $\mathcal{M}$ 过大使得 budget $\times\,\mathcal{M}$ 近似等同于整个未标记数据池的大小, 采样策略就又变为了传统的 k-Center 算法. 基于重要性原则的选择方法的方法和 k-Center 算法可以看作我们的采样策略的特例, 这也就意味着这个策略结合了之前两种方法的优点. 本文的实验部分对超参数 $\mathcal{M}$ 的选择对实验结果的影响进行了具体的分析. 综上所述, 将我们的采样策略命名为多样化的重要性采样策略 (diversity-importance). 这个采样策略关注更加重要的样本并且使用 k-Center 算法保证所选样本的多样性, 并且很好地结合了之前策略的优点, 能够更加有效地选择出更有助于获得受害者模型功能的重要样本.

4.2 替代模型的训练方法

在每次迭代攻击过程中, 使用多样化的重要性采样策略构建迁移集合之后, 攻击者会使用迁移集合训练替代模型. 之前的工作通常使用交叉熵 (CE) 作为训练时的损失函数. 交叉熵损失函数是分类任务中常用的损失函数. 然而, 常规分类任务中的标签通常是热独编码 (one-hot) 的形式, 而本文所关注的模型窃取任务中查询受害者模型所获得的伪标签为概率向量的形式, 对于不同的样本, 受害者模型和替代模型的输出之间可能会有较大的差异, 传统的交叉熵损失函数难以关注到训练过程中不同样本之间的差异, 因而不够有效. 为了增强替代模型从迁移集合学习受害者模型功能的能力, 本小节试图设计一种能够更多关注困难样本的损失函数. 受到 focal 损失函数 [21] 的启发, 同时考虑输出中每个类别概率值所起到的不同作用, 这里将受害者模型和替代模型的输出之间的差值作为权重. 换而言之, 本文提出了一个基于类别统计的加权损失函数, 并将其命名为差异损失 (differ-loss, DL), 定义为

$$\mathrm{DL}=-\sum_{i}\sum_{j}N\cdot\mathrm{softmax}\left(\left|f(x_i)-\hat{f}(x_i)\right|\right)_j\cdot f(x_i)_j\cdot\log\left(\hat{f}\left(x_i\right)_j\right),\tag{7}$$

其中, f 代表受害者模型, $\hat{f}$ 代表替代模型, N 代表类别数. 因此, softmax 函数的定义如下:

$$\mathrm{softmax}\left(\left|f(x_i)-\hat{f}(x_i)\right|\right)_j=\frac{\mathrm{e}^{|f(x_i)-\hat{f}(x_i)|_j}}{\sum_{k=1}^{N}\mathrm{e}^{|f(x_i)-\hat{f}(x_i)|_k}}.\tag{8}$$

值得注意的是, 这个差异损失函数具有如下的 3 个性质: (1) 在训练的开始阶段, 由于替代模型对所有的样本都不能做出很好的预测, 此时伪标签中概率值较大的类别会占据较大的比例, 从而加快模型的收敛速度; (2) 在训练过程中, 替代模型会更多地关注差异较大的类别, 从而增强替代模型模仿受害者模型的能力; (3) 当各个类别上的差异比较小时, softmax 函数的输出会趋于均匀分布 (uniform distribution), 避免了后期训练过程中损失函数的曲线过于振荡.

算法 2 S&W 算法

输入: 未标记样本池 D_U, 受害者模型 f, 最大查询次数 Q, 查询次数序列 S, 超参数 $\mathcal{M}$;
主迭代: 查询次数 $q \leftarrow 0$, $D_T \leftarrow \emptyset$;
注: D_U 为未标记样本池, D_C 为重要样本子集合, D_T 为迁移集合;
while $q < Q$ **do**
  //构建迁移集合 (sampling)
  根据查询序列 S 决定查询预算 budget;
  $D_C = \text{topk}(\text{Importance}(\hat{f}(x_i))|x_i \in D_U, \text{budget} \times \mathcal{M})$;
  根据算法 1 更新迁移集合 D_T;
  //使用加权损失函数训练替代模型 (weighting)
  $D_T \leftarrow D_T \cup x_i$;
  $D_C \leftarrow D_C / x_i$;
  $i = i + 1$;
end while
输出: 替代模型 $\hat{f}$, 迁移集合 D_T.

本文提出的损失函数更好地关注了困难样本, 从而提高了替代模型模拟受害者模型功能的能力. 这种提高意味着替代模型可以在更少的查询次数下和受害者模型有着更高的相似度, 从而有助于我们的方法在第一阶段利用替代模型近似计算出的样本对于受害者模型的重要性更加准确. 本文的实验部分进一步展示了分别在使用和不使用差异损失函数的情况下所选出样本的估算重要性和真实重要性之间的差异. 结合多样化的重要性采样策略和差异损失函数, 本文提出的方法更有助于提升模型窃取攻击成功率. 综上所述, 本文所提出的模型窃取方法被命名为 S&W, 并在算法 2 中详细展示了 S&W 算法的整体流程.

5 实验分析

5.1 实验设置

本小节主要在真实数据集上对 S&W 算法进行评估. 首先主要介绍本文的实验设置, 包含了使用的受害者模型、替代模型结构的选择、评价指标、攻击数据集和训练过程中超参数的设置.

5.1.1 受害者模型

实验中使用的受害者模型为在 4 个常用数据集上训练的 ResNet-34 结构模型 [22]. 数据集包括 CIFAR10 (91.56%) [23], MNIST (99.59%) [24], Caltech-256 (78.40%) [25] 和 CUBS200 (77.10%) [26]. 实验遵循了这些数据集原始作者所建议的训练集 – 测试集划分方式, 测试集随后会被用于评价替代模型的性能. 受害者模型使用基于动量的随机梯度下降优化器进行参数学习, 其中优化器的动量值 (momentum) [27] 设置为 0.5, 初始学习率设置为 0.1, 并且学习率在每 30 次后衰减为之前的 0.1, 一共训练 200 次. 为了模拟真实的在线部署场景, 这些模型全都以黑盒设定进行实验评估 (即仅知道输入图片和输出预测值的信息). 仿照之前的工作 [2,6,8], 本文中替代模型的结构同样选择 ResNet-34, 但是会分析不同的替代模型结构对实验结果的影响.

5.1.2 评价指标

测试准确率 (test accuracy) 是该任务中最为常用的评价指标. 同时, 相关工作 [2,5] 使用替代模型和受害者模型的测试准确率的比值作为评测指标. 然而, 这样的评测指标不能够真实地反映两个模型

之间的相似程度, 例如, 在这个评价标准下, 两个差异很大的模型具有相近的测试准确率也会被认为是很相似的模型. 因此, 本文沿用了 ActiveThief [8] 中提出的评价指标, 它计算两个模型所作出的相同预测占总预测的比例, 具体形式如下:

$$\text{Similarity} = \frac{1}{|D_{\text{test}}|} \sum_{x \in D_{\text{test}}} \mathbb{I}(\text{argmax}(f(x)) = \text{argmax}(\hat{f}(x))), \tag{9}$$

其中 D_{test} 表示测试集, $\mathbb{I}(\cdot)$ 为指示函数. 这个评价指标可以更好地反映出两个模型之间的相似性. 此外, 本文还使用了相似性曲线的曲线下面积 (the area under similarity, AUC) 作为另一个评价指标. AUC 代表了迭代攻击过程中相似性随着查询次数的增长速率. 一个较大的 AUC 值意味着方法可以用较少的查询次数获得更好的效果.

5.1.3 攻击数据集

为了方便起见, 本文使用 ILSVRC-2012 挑战赛 [28] 中提出的 120 万张图片作为攻击数据集, 并且仅使用了图片本身, 而没有使用原始数据集中提供的标签信息. 在真实的攻击场景中, 攻击者很可能会使用互联网上收集的大量无标注图片作为攻击数据集, ILSVRC 数据集可以很好地模拟这一场景. 当攻击在 MNIST 数据集上训练的受害者模型时, S&W 算法会首先将图片转换为单通道灰度图, 之后将图片降采样为 28×28 像素大小. 同样, 当攻击 CIFAR10 数据集上训练的受害者模型时, S&W 算法会将图片降采样为 32×32 像素大小.

5.1.4 替代模型的训练过程

替代模型使用动量设置为 0.9 的 SGD 优化器训练 200 次. 其中, 初始学习率设置为 $0.02 \times \frac{\text{batchsize}}{128}$, 并且每 60 代衰减为之前的 0.1, 权重衰减 (weight decay) 设置为 5×10^{-4}. 本文实验事先设置了一个查询序列: {0.1k, 0.2k, 0.5k, 0.8k, 1k, 2k, 5k, 10k, 20k, 30k} 作为每次迭代的最大查询次数限制. 在每次迭代中, 如果达到了当前迭代的最大查询次数限制, 算法就会停止样本选择或生成策略. 为了公平对比, 所有的实验都会在相同的查询序列下进行.

5.1.5 对比的方法

本文的实验主要对比了 S&W 方法和之前的一些最优秀的基于数据选择的方法 (Knockoff-Nets [2] 和 ActiveThief [8]) 的性能表现. 此外也验证了一些基于生成的方法 (DaST [6] 和 JBDA [5]) 的性能, 由于这种类型的方法需要较大的计算和时间开销, 因此只在 CIFAR10 和 MNIST 数据集上验证了这两个方法的性能. 为了对比公平, 本文对比的 ActiveThief [8] 方法为使用 PyTorch 复现的版本, PyTorch 版本取得了和作者提供的 TensorFlow 版本相近的性能. 除了 ActiveThief [8], 其他方法 [2, 5, 6] 都使用了由作者提供的原始的代码.

5.2 实验结果

本小节首先对比 S&W 算法和之前方法在 4 个常用数据上的性能差异. 之后分析不同模型结构对结果的影响. 更进一步, 讨论超参数 $\mathcal{M}$ 对实验结果的影响, 并进行消融实验. 最后, 比较 S&W 算法和之前方法获得的替代模型上产生的对抗样本的迁移能力强弱.

5.2.1 性能对比

这里比较 S&W 算法和之前方法的性能差异. 如表 1 所示, S&W 算法在 3 万查询次数下的 Similarity 和 AUC 指标都优于其他的方法. 在 CIFAR10 数据集上, S&W 算法所取得的 Similarity

表 1 各个方法 3 万查询次数下获得的替代模型的相似性 (similarity) 和曲线下面积 (AUC)

Table 1 Similarity and area under the curve (AUC) of the substitute model of each method under 30k queries[a)]

Method	CIFAR10		MNIST		Caltech-256		CUBS200	
	Similarity (%)	AUC	Similarity (%)	AUC	Similarity (%)	AUC	Similarity (%)	AUC
Knockoff-Nets(Random)	81.59	19344.48	92.91	22499.04	76.42	19520.95	65.48	14684.95
Knockoff-Nets(Adaptive)	85.17	19388.41	98.48	26065.39	79.08	20705.74	64.53	15352.14
ActiveThief(Entropy)	81.61	18624.32	93.12	23344.51	77.38	20698.33	68.12	16335.49
ActiveThief(k-Center)	82.98	18451.83	98.70	27321.89	78.66	21198.21	73.71	18391.34
ActiveThief(DFAL)	80.42	17945.05	93.20	24226.05	64.56	18162.48	53.24	13362.67
ActiveThief(DFAL+k-Center)	82.05	18926.38	96.87	27385.54	67.27	18847.35	61.39	15192.22
Ours	**86.93**	**21789.20**	**98.90**	**28021.78**	**80.64**	**22214.30**	**77.42**	**19439.42**

a) Boldface: the best value.

表 2 DaST 和 JBDA 的性能

Table 2 Performance of DaST and JBDA[a)]

Method	CIFAR10		MNIST	
	Similarity (%)	AUC	Similarity (%)	AUC
DaST	10.00	2990.00	16.41	3459.35
JBDA	10.01	3364.53	18.55	4595.63
Ours	**86.93**	**21789.20**	**98.90**	**28021.78**

a) Boldface: the best value.

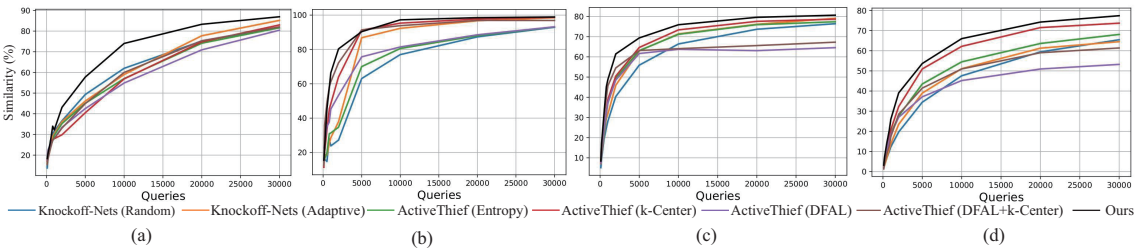

图 2 (网络版彩图) S&W 算法, Knockoff-Nets 和 ActiveThief 的相似性随查询次数的变化曲线

Figure 2 (Color online) Curves of the similarity versus the number of queries for the S&W algorithm, Knockoff-Nets, and ActiveThief. (a) CIFAR10; (b) MNIST; (c) Caltech-256; (d) CUBS200

比之前次优方法 Knockoff-Nets(Adaptive) 提高了 1.76%, AUC 高出了 12.38% (2400.79). 在 MNIST 数据集上, S&W 算法取得了高达 98.90% 的 Similarity, 这意味着算法取得的替代模型与受害者模型的功能几乎完全一致. 在 Caltech-256 数据集上, S&W 算法所取得的 Similarity 比之前次优方法 Knockoff-Nets(Adaptive) 提高了 1.56%, AUC 高出了 7.29% (1508.56). 在 CUBS200 数据集上, S&W 算法所取得的 Similarity 比之前次优方法 ActiveThief(k-Center) 提高了 3.71%, AUC 高出了 5.70% (1048.08). 由于基于数据生成的方法往往需要很大的计算和时间开销, 这里只在 CIFAR10 和 MNIST 两个小型数据集上测试了这一类型的方法的性能. 此外, 在文献 [5] 中 JBDA 方法假设攻击者可以接触到受害者模型训练数据集的部分信息, 但在本文的设置中, 这部分信息是完全未知的. JBDA [5] 和 DaST [6] 的性能如表 2 所示, 显然, 这种基于数据生成的方法当查询次数较小时性能很差. 在文献 [5,6] 中, 使用的查询次数通常是 10^7 数量级的, 这是非常巨大的开销, 这种类型的方法是很无效的. 完整的 Query/Similarity 曲线如图 2 所示, 综上所述, 在 4 个常用数据集上进行的实验展示了 S&W 算法的性能均优于其他的

表 3 S&W 算法和之前的方法在不同的模型结构下在 CIFAR10 数据集上的性能 (%). 其中, 左侧第一列表示替代模型的结构, 受害者模型的结构仍使用 ResNet-34

Table 3 Performance of the S&W algorithm and previous methods under different architectures on CIFAR10 dataset. The first column on the left represents the structure of the substitute model, and the victim model still uses ResNet-34[a)]

Architecture	Ours	Knockoff-Nets(Random)	ActiveThief(k-Center)
ResNet-34	***86.93***	85.17	82.98
ResNet-18	**86.23**	85.00	*84.66*
ResNet-50	**86.86**	*86.13*	83.64
VGG-16	**83.83**	75.54	71.89
DenseNet	81.32	**83.50**	79.96

a) Boldface: the best value. Italics: the second-best value.

方法.

5.2.2 模型结构对结果的影响

相比于第 5.2.1 小节替代模型和受害者模型使用相同的模型结构, 本小节将在 CIFAR10 数据集上展示不同的模型结构对结果的影响. 保持受害者模型的结构仍然是 ResNet-34, 分别选择 ResNet-18, ResNet-50 [22], VGG-16 [29] 和 DenseNet [30] 作为替代模型的模型结构, 并测试攻击的效果. 同时也对比了之前具有代表性的方法, Knockoff-Nets(Random) 和 ActiveThief(k-Center), 在这几种模型结构下的表现. 结果如表 3 所示, 可以看到, S&W 算法在所有的模型结构上的效果几乎都优于之前的方法. 同时, 还可以从表 3 中发现: (1) 选择和受害者模型不同的模型结构确实会影响算法的性能, 并且相似的结构会取得更好的结果, 比如 ResNet 的效果要优于 VGG 和 DenseNet; (2) S&W 算法受到模型结构的影响更小. S&W 算法最差与最好的结果之间的差值为 5.61%, 相比于其他两个方法的差值 10.59% 和 12.77%, S&W 算法有着更好的适用性, 能够胜任各种不同的模型结构.

5.2.3 超参数对结果的影响

在 S&W 算法中, 每次迭代会先选择 budget $\times\,\mathcal{M}$ 数量有着最高 Importance 值的样本, 之后会使用 k-Center 算法从中选择出 budget 数量的样本. 因此, 超参数 $\mathcal{M}$ 选择不同的值会对实验的结果产生影响. 我们在 CIFAR10 数据集上测试了不同的 $\mathcal{M}$ 取值下的实验效果, 结果如图 3 所示. 随着 $\mathcal{M}$ 取值的变化, 实验结果也有变化, 其中最好的结果与最差的结果之间相差 1.09%. 这是因为当 $\mathcal{M}$ 取值较小时, 开始选择的样本数较少, k-Center 算法无法保证最终所选样本的多样性, 影响了算法的性能. 而当 $\mathcal{M}$ 取值较大时, 开始选择的样本中会有很多包含较低信息的样本, 之后再使用 k-Center 算法则不可避免地将这些低信息样本引入了迁移集, 这些样本对获得受害者模型功能的帮助较小, 浪费了查询次数, 从而影响了算法的性能.

5.2.4 面对防御方法的效果

目前已经有了一些防御模型窃取的方法, PRADA [31] 和 Prediction poisoning [32] 便是常见的防御方法. PRADA 是一种基于检测的防御方法, 这种方法认为现实场景中正常用户的查询样本之间的距离分布通常为正态分布, 而攻击者的查询样本的距离分布通常偏离这一分布. 图 4 中统计了在攻击 CIFAR10 模型时 S&W 算法使用的查询样本之间的距离分布, 可以看到这个分布完全符合正态分布. 由于 S&W 算法使用的样本为从收集的自然样本中选择出来的, 因而可以很轻易地躲避这种防御方法的检测.

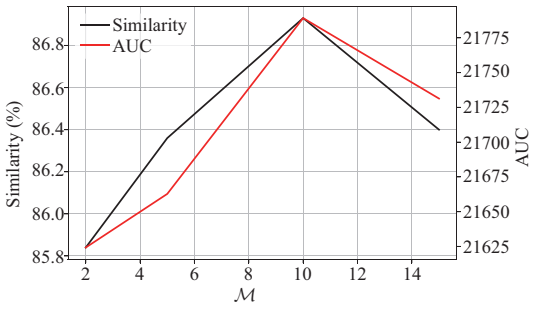

图 3 (网络版彩图) Similarity 和 AUC 随着超参数 $\mathcal{M}$ 的变化曲线

Figure 3 (Color online) Curves of similarity and AUC versus different $\mathcal{M}$

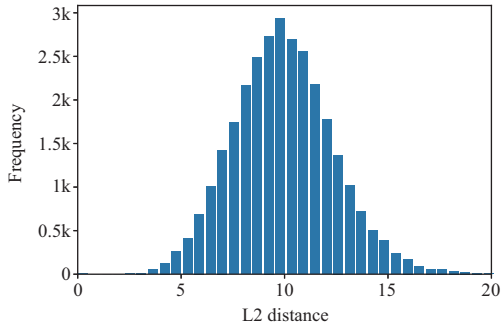

图 4 (网络版彩图) S&W 算法查询样本之间距离的分布

Figure 4 (Color online) Distribution of distances between queries of the S&W algorithm

表 4 S&W 算法和之前的方法面对 Prediction poisoning 防御方法在 CIFAR10 数据集上的性能 (%)

Table 4 Performance of the S&W algorithm and previous methods against Prediction poisoning on the CIFAR10 dataset[a)]

Method	Threshold = 0	Threshold = 0.5	Threshold = 0.8
Knockoff-Nets(Random)	81.59	70.56	68.97
ActiveThief(Entropy)	81.61	73.70	63.29
ActiveThief(k-Center)	82.98	72.45	71.73
Ours	**86.93**	**75.88**	**73.27**

a) Boldface: the best value.

Prediction poisoning 则是基于扰动的防御方法, 这种方法会给后验概率 y 增加噪声, 使增加噪声后的梯度信号最大程度地偏离原始梯度. 这里我们选择了两个不同的阈值来测试 S&W 算法面对这种防御方法时的效果, 阈值越大代表防御效果越好, 但也会损失一定的受害模型精度. 表 4 中展示了不同方法面对这种防御方法时的效果. 对比其他方法, S&W 算法在不同的阈值下都有较好的效果, 这也进一步说明了 S&W 算法在面对防御方法时更加鲁棒.

5.2.5 差异损失函数对数据选择所起到的帮助

第 4.2 小节讨论了差异损失函数可以帮助模型在更小的查询次数下获得和受害者模型更加相似的输出, 这有助于 S&W 算法利用替代模型的输出更精确地计算样本的 Importance 值. 图 5 统计了使用和不使用差异损失函数所选择的样本的真实 Importance 值的分布情况. 很明显, 差异损失函数的加入改变了所选择样本的真实 Importance 值的分布, 所选择的样本大部分都倾向于具有较高 Importance 值. 这是由于差异损失函数更快地缩小了替代模型和受害者模型输出上的差异, 从而使得利用替代模型输出近似计算的 Importance 值更加准确.

5.2.6 对抗样本的迁移能力

之前的一些工作 (例如, JBDA [5], DaST [6] 和 ActiveThief [8]) 在替代模型上产生对抗样本, 之后利用这些对抗样本攻击受害者模型. 值得注意的是, 在更加相似的替代模型上产生的对抗样本会有更强的迁移性. 本小节测试了利用不同方法获得的替代模型在 CIFAR10 测试集上产生的对抗样本的迁移能力. 这里使用投影梯度下降 (projected gradient descent, PGD) 攻击法 [33] 生成对抗样本, 并限制

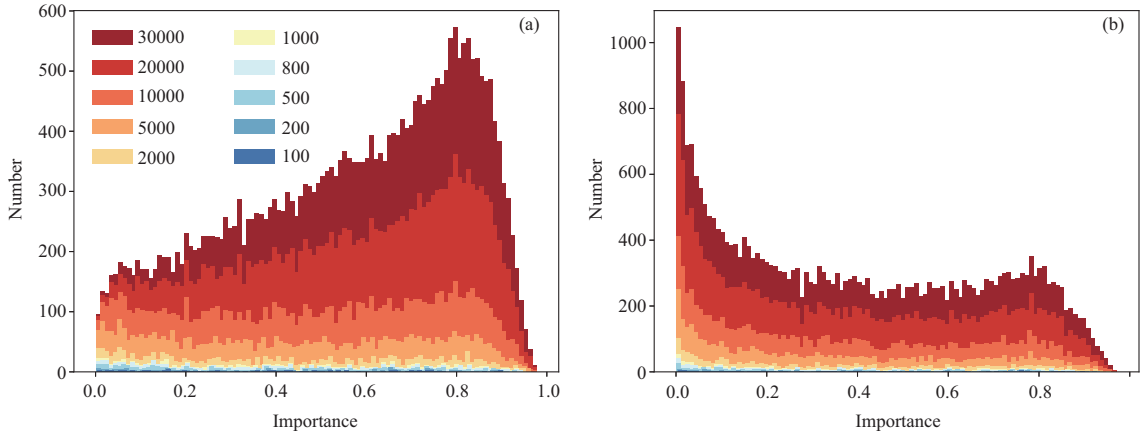

图 5 (网络版彩图) 使用 (a)/不使用 (b) 差异损失函数所选择的样本的 Importance 值的分布, 其中图例代表查询次数

Figure 5 (Color online) The distribution of selected samples' Importance values with (a)/without (b) differ-loss, where the label represents the number of queries

表 5 在替代模型上使用 PGD 攻击方法产生的对抗样本的迁移能力

Table 5 Transferability of adversarial samples generated by the PGD attack on the substitute models[a)]

Method	Transferability (%)
Knockoff-Nets(Random)	63.09
Knockoff-Nets(Adaptive)	68.33
ActiveThief(Entropy)	62.51
ActiveThief(k-Center)	64.15
ActiveThief(DFAL)	62.28
ActiveThief(DFAL+k-Center)	67.71
Our method	**76.76**

a) Boldface: the best value.

扰动的无穷范数 (L_∞-norm) 上限为 8/255. 结果如表 5 所示, 在 S&W 算法所获得的替代模型上生成的对抗样本有着更强的迁移性, 迁移成功率至少提高了 8.43%.

5.2.7 消融研究

表 6 展示了本文所进行的消融实验, 主要研究了 S&W 算法中两个关键模块, 即数据选择策略和差异损失函数, 分别起的作用大小. 消融实验在 CIFAR10 数据集上进行, 分别考虑如下两种情况: (1) 只使用数据选择策略; (2) 同时使用差异损失函数和数据选择策略. 第一种情况比较了 S&W 算法中的数据选择策略和之前的一些数据选择策略, 结果如表 6 中前 4 行所示. 很显然, 使用 Importance 度量重要程度要优于使用熵度量, 并且, 结合 k-Center 算法有助于进一步提升算法的性能, 表现在加入 k-Center 算法后有着更大的 AUC 值, 这意味着可以用更小的查询次数获得更好的结果. 之后, 验证了差异损失函数的影响, 如表 6 中后 2 行所示, 加入差异损失函数进一步带来了 1.36% 的性能提升, 这说明了差异损失函数所起到的重要作用.

表 6 分析 S&W 算法中的数据选择策略和差异损失函数的消融实验的结果

Table 6 Ablation results on analyzing the effect of our data selection strategy and differ-loss function[a)]

Method	Similarity (%)	AUC
Entropy	81.61	18624.32
Importance	**85.57**	17982.29
k-Center	82.98	18451.83
Importance+k-Center	**85.57**	**19934.36**
Entropy+k-Center+differ-loss	85.49	21569.42
Importance+k-Center+differ-loss	**86.93**	**21789.20**

a) Boldface: the best value.

6 总结与展望

本文提出了一种新的模型窃取攻击方法 S&W, 包含了一种简单而有效的数据选择方法和一个新颖的加权损失函数. 本文研究了有助于模型窃取攻击的样本数据特征, 并测试了在不同方法获得的替代模型上产生的对抗样本的迁移能力, 这证明了模型窃取攻击有着对现实世界的真实潜在威胁. 实验部分比较了 S&W 算法和之前一些方法的性能, 并通过消融实验分析了 S&W 算法中各个模块所起的作用, 展示了 S&W 算法的优越性, 证明了可以使用更少的查询次数获得更好的攻击效果. 希望本文的工作可以为模型窃取攻击提供一些新思路, 从而引起人们对模型隐私保护的重视.

虽然本文主要针对图像分类模型进行了研究, 然而我们认为 S&W 方法同样适用其他类型的模型. 目前已经有了较多的在其他任务模型上进行样本重要性度量的指标, 例如, 在 BERT 模型上 [34] 和在图像分割任务上 [35]. 而 S&W 方法中的差异损失函数为基于交叉熵损失函数的改进, 因而也适用于大多数模型. 综上所述, 我们认为 S&W 方法可以很容易地扩展到其他任务中, 这将在我们以后的工作中进行验证.

参考文献

1 Oh S J, Schiele B, Fritz M. Towards reverse-engineering black-box neural networks. In: Proceedings of Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Cham: Springer, 2019. 121–144

2 Orekondy T, Schiele B, Fritz M. Knockoff Nets: stealing functionality of black-box models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4954–4963

3 Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models. In: Proceedings of IEEE Symposium on Security and Privacy, 2017. 3–18

4 Ji S L, Du T Y, Li J F, et al. Security and privacy of machine learning models: a survey. J Softw, 2021, 32: 41–67 [纪守领, 杜天宇, 李进锋, 等. 机器学习模型安全与隐私研究综述. 软件学报, 2021, 32: 41–67]

5 Papernot N, McDaniel P, Goodfellow I, et al. Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017. 506–519

6 Zhou M, Wu J, Liu Y, et al. DaST: data-free substitute training for adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 234–243

7 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. Commun ACM, 2020, 63: 139–144

8 Pal S, Gupta Y, Shukla A, et al. ActiveThief: model extraction using active learning and unannotated public data. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 34: 865–872

9 Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. In: Proceedings of Conference and Workshop on Neural Information Processing Systems Deep Learning Workshop, 2013

10 Lewis D D, Gale A W. A sequential algorithm for training text classifiers: corrigendum and additional data. ACM

SIGIR Forum, 1995, 29: 13–19
11 Sener O, Savarese S. Active learning for convolutional neural networks: a core-set approach. In: Proceedings of International Conference on Learning Representations, 2018
12 Ducoffe M, Precioso F. Adversarial active learning for deep networks: a margin based approach. In: Proceedings of the 35th International Conference on Machine Learning, 2018
13 Cohn D, Ghahramani Z, Jordan M. Active learning with statistical models. In: Proceedings of Advances in Neural Information Processing Systems, 1994. 7
14 Anil R, Pereyra G, Passos A, et al. Large scale distributed neural network training through online distillation. 2018. ArXiv:1804.03235
15 Furlanello T, Lipton Z, Tschannen M, et al. Born again neural networks. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 1607–1616
16 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. In: Proceedings of Conference and Workshop on Neural Information Processing Systems Deep Learning Workshop, 2015
17 Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks. In: Proceedings of International Conference on Learning Representations, 2014
18 Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of International Conference on Learning Representations, 2015
19 Kingma D P, Welling M. Auto-encoding variational Bayes. 2014. ArXiv:1312.6114
20 Zhang B, Li L, Yang S, et al. State-relabeling adversarial active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 8756–8765
21 Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 2980–2988
22 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016. 770–778
23 Krizhevsky A, Hinton G. Learning Multiple Layers of Features from Tiny Images. Technical Report TR-2009, Toronto: University of Toronto, 2009
24 Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE, 1998, 86: 2278–2324
25 Griffin G, Holub A, Perona P. Caltech-256 object category dataset. 2007. https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001
26 Wah C, Branson S, Welinder P, et al. The Caltech-UCSD birds-200-2011 dataset. 2011. https://www.vision.caltech.edu/datasets/cub_200_2011/
27 Qian N. On the momentum term in gradient descent learning algorithms. Neural Networks, 1999, 12: 145–151
28 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252
29 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of International Conference on Learning Representations, 2015
30 Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017. 4700–4708
31 Juuti M, Szyller S, Marchal S, et al. PRADA: protecting against DNN model stealing attacks. In: Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P), 2019. 512–527
32 Orekondy T, Schiele B, Fritz M. Prediction poisoning: towards defenses against DNN model stealing attacks. In: Proceedings of the 36th International Conference on Machine Learning, 2019
33 Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks. In: Proceedings of International Conference on Learning Representations, 2018
34 Dor L E, Halfon A, Gera A, et al. Active learning for BERT: an empirical study. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. 7949–7962
35 Casanova A, Pinheiro P O, Rostamzadeh N, et al. Reinforced active learning for image segmentation. In: Proceedings of International Conference on Learning Representations, 2020

Model stealing attack based on sampling and weighting

Yixu WANG[1], Jie LI[1], Hong LIU[2], Yan WANG[5], Mingliang XU[3], Yongjian WU[4] & Rongrong JI[1*]

1. *Department of Artificial Intelligence, Xiamen University, Xiamen* 361005*, China*;
2. *National Institute of Informatics, Tokyo* 101-8430*, Japan*;
3. *Department of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou* 450001*, China*;
4. *Youtu Laboratory, Tencent, Shanghai* 200235*, China*;
5. *Pinterest, Seattle* 98101*, USA*

* Corresponding author. E-mail: rrji@xmu.edu.cn

Abstract A model stealing attack aims to create a substitute model that steals the task completion ability of the target victim model. Popular approaches have used data generation/selection and entropy loss to achieve promising attack performance. In this paper, we explore two overlooked yet effective components of the attack, data sampling and weighting. We propose a novel method named S&W that provides a sampling scheme and a soft-label weighted loss function. First, we propose a data selection strategy that pays more attention to important samples for stealing more information from the victim model. Then, we introduce the k-Center algorithm to guarantee the selected subset's diversity, aiming to make the core-set selection tractable. Second, we propose a weighted entropy loss inspired by the focal loss that mainly focuses on the difference in outputs of the victim and the stealing models, allowing the substitute model to better simulate the victim model. Extensive experiments on four widely used datasets consistently show that our proposed method outperforms state-of-the-art methods, with a maximum improvement of 5.03% over the next best method.

Keywords computer vision, model stealing attack, adversarial attack, active learning, knowledge distillation