人工智能安全作业 III

3210105703 黄鸿宇

Question I

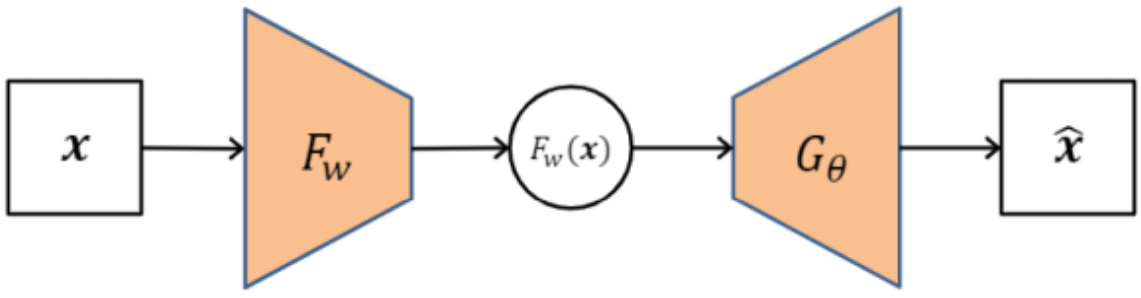
> 白盒模型逆向攻击和黑盒模型反演攻击各适用于什么样的场景？课程中的模型逆向攻击都需要最终的输出向量，是否有可能进行仅获得标签的模型逆向攻击？

两种攻击场景的特征和适用场景如下：

- 白盒模型逆向攻击：攻击者知道模型的结构和参数，将模型逆向供给问题转变为一个优化问题，优化目标为使逆向数据的输出向量与目标数据的输出向量的差异尽可能地小，攻击者通过获取输出向量，逆向出用户的输入向量。当攻击者了解模型的参数，结构等全面知识或者部分知识，才可以进行白盒攻击。
- 黑盒模型反演攻击：攻击者不知道目标模型结构，参数和训练集，但知道模型数据集的大致分布和输入输出结构，访问得到的输出不是完整向量，是截断向量，在这种情况下对模型进行攻击。该方法通过训练一个反演模型，将获得的截断向量反演推导出样本。该方法适用于攻击者在不能获取目标内部结构，但是能通过访问获得目标模型对于样本的推断结果时对模型进行攻击。

如果无法获得最终的输出向量，只能获得输入样本的标签，模型逆向攻击将会变得十分难以进行，因为这本质是一个通过输出信息“推导”出输入信息的过程，输出信息的信息量越大，通过输出信息对样本进行重建的效率也就越多，如果输出信息只有样本的标签，那么可能可以通过大量的模型访问，在样本上有意识的加入扰动来逐步逼近，探索模型的决策边界从而实现对模型参数等隐私的窃取，进而利用获得的模型隐私，推断模型的结构并反演重建样本，但是由于标签所含信息量过低，这种攻击需要大量的访问，在实践上会变得十分的低效，在现实场景中几乎不可完成。

同时，对于一个成熟的模型，由于许多不同样本，但是含有相同特征的图像都会被归类成同一个标签，当我们知道标签的时候，就大概知道了图像的内容（比如图像分类模型，当我们知道一张图片被分类为猫时，就知道图片里含有一只猫），在无法获得更多信息的前提下，我们最多也就只能做到这点了（在只知道两张图片都被归类为同一标签，而没有其他信息，连两者最后的输出向量都不知道的情况下怎么将他们区分开呢？），**再进行反演攻击没有意义。**



攻击原理示意图

Question 2

> 模型窃取攻击中，替代模型方法异常的大量查询不仅仅会增加窃取成本，更会被模型拥有者检测出来，你能想到什么解决方法来避免过多的向目标模型查询？

我们可以从以下两个角度来避免过多的向目标模型查询

1. 针对特化目标的模型查询

在某些查询场景中，我们并不需要获取模型的全部知识，比如在针对NLP模型ChatGPT的查询中，我们可能仅仅需要获得其在某个场景，比如写代码等任务中的回答能力，然后通过一个中等大小的模型去窃取大模型中的部分能力，在这种情况下，我们可以提高查询样本的集中程度，着重于输入服务我们查询目标的样本而删除无关的样本，提高我们单词查询的有效性。

2. 利用聚类算法筛选输入样本提高查询覆盖广度

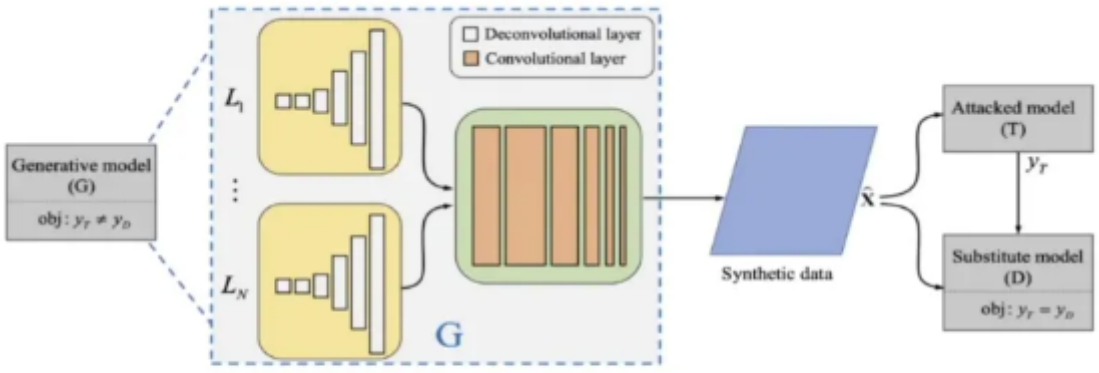
我们可以在查询之前，通过聚类算法对即将输入的样本进行聚类分析，然后选取每个聚类中靠近中心的样本输入，经过如此操作，我们的输入样本会变得具有更强的"代表性"，从而能够达到一个样本覆盖多个样本的效果，进而在给定数量的查询中，我们能够覆盖更多可能的样本空间，从而提高对目标模型窃取的效率。

3. 利用已知模型信息信息提高窃取效率

有的时候，当我们了解部分被攻击模型的有关信息，对该模型进行灰盒攻击时，我们可以利用已知的部分信息，如决策边界，模型超参数等，构造更加有价值的样本，仔细选择涵盖模型决策比娜姐和关键特征的代表性输入，并根据输入样本的反馈结果重新选择下一个样本的扰动方向，构造方式，充分利用已有信息构造更加有价值的样本，优化偷窃能力，最大限度减少整体的查询量。

4. 利用GAN生成模型

对抗生成网络这个Brilliant idea也可以用在模型窃取攻击的样本生成中，其原理可以通过下图说明，下图左边部分是生成器，右边是替身的模型，T是被攻击模型。这一对抗机制在于利用生成模型去生成一些人造数据，和当前训练替身模型的方法一样，将这些人造数据输入被攻击的模型，得到它的输出，利用输入输出对让替身模型模仿被攻击模型的输出。替身模型的目标就在于保证输出与T的输出是一致的，即 ，对于生成器来讲，目标是寻找人造数据，去寻找能够使得T与D输出不一致的人造数据，作用在于挖掘T和D，也就是被攻击模型和替身模型之间的不同之处。找到样本以后，再将信息给替身模型，进而不断地去逼近被攻击模型，形成了一种对抗的形式，一种博弈。



the objective of G: generate samples $\widehat{\mathbf{X}} = G(\mathbf{X})$ and let $y_D(\widehat{\mathbf{X}}) \neq y_T(\widehat{\mathbf{X}})$
the objective of D: guarantee $y_D(\widehat{\mathbf{X}}) = y_T(\widehat{\mathbf{X}})$

除了上述方法，我们还可以通过对样本进行打乱，随机化，分布式多点查询，模仿人类正常用户的行为进行查询等方法，降低我们的查询被发现的可能性。

Question 3

> 在MemGuard 的防御场景下，如果攻击者在输入图像上添加扰动可以破坏单次随机的设定，你认为防御者应该如何应对?

如果攻击者在输入图像上添加扰动，可以破坏单次随机的设定，这是因为我们通过哈希算法来判断每张图片是否应该被添加随机向量，但是如果攻击者在输入图像上进行扰动，将会导致图像的哈希值发生较大的改变，从而使得生成的随机数发生变化，导致攻击者发现随机向量的存在或者进一步了解其生成规律，从而破坏MemGuard的整体防御效果。

作为防御者，我们可以通过以下几种手段应对：

1. 对图像进行滤波处理等图像处理技术，降低微小扰动的影响

通过中值滤波等方式对图像进行处理，可以降低甚至消除单个像素微小扰动带来的影响，从而使得经过处理后的图像的哈希值保持较高的一致性，为了使该方法的效果更为显著，可综合利用滤波处理（降低部分像素更改带来的影响），分块处理（降低图片巨部位置更改带来的影响），甚至综合运动深度学习算法等处理技术，可以有效使同样特征，类似的图片的哈希值保持稳定。

2. 模糊哈希算法

从生成哈希值的算法角度来讲，普通的哈希算法如md5,SHA-1等当碰到极少的像素更改就会使最后生成的结果完全不一样，这固然提高了安全性但是不符合我们在这里要求的"减少像素扰动对结果的影响"的规定，因此，模糊哈希算法，即基于内容分割的分片哈希算法，在特定条件下对文件进行分片，然后使用一个强哈希对文件每片计算哈希值，取这些值的一部分并连接起来，与分片条件一起构成一个模糊哈希结果，该方法在将输入内容压缩的同时，也保留了两个文件的相似度信息（即比较模糊哈希结果可以了解两个原始输入的相似度），从而少量的扰动并不会大幅度改变模糊哈希结果。

3. 用中间层进行随机种子生成，而非原始输入

虽然少量扰动即可大幅改变图片的哈希值，但我们知道，对于同样特征的神经网络，其中间层的相似度（如特定神经元信号大小的比例）等，在扰动不混淆原模型的前提下（由于本章节研究的是通过扰动来破坏"单次随机"，而非混淆原模型，所以我们做出此前提假设），中间层的相似度会高于图像直接哈希的相似度，因此，我们可以通过中间层的有关数据，神经元的强度比例等数据利用特定算法综合生成随机种子，而非直接运用输入图像，这样会使得相似特征，相似图片在添加扰动方面尽量保持一定的一致，提高防御的效率。

Question 4

> 数字水印可以保护模型版权，但是无法防御攻击者窃取模型的过程，是否有方法可以直接防止模型被窃取?

由于模型的训练，部署，提供服务，保存等是一个包括商业，技术，法律等多种领域的综合性过程，所以防止模型被窃取，不仅仅可以从技术角度出发，也可以从管理等多个角度入手。同时，我们可以将模型窃取步骤中与原模型交互的部分分为两类：获取原模型知识和对原模型查询，前者可以通过模型信息保密而避免，而后者只要模型是对外开放，被外界所运用就有可能会发生，所以从纯技术角度来说，没有完全可靠的方法防止模型被窃取（毕竟查询时不可避免的，查的够多总能把模型还原出来），我们能做的，只有尽量识别出攻击性的查询，或者对查询结果做出一定的混淆，从而降低单次攻击能够获得的模型知识。

1.对模型的访问进行一定的控制

对于模型部署者来讲，控制好外界对模型的访问至关重要，尽可能减少外界获得模型内部参数或者中间层的可能性，比如，减少模型输出的信息量，对于传统的分类模型，只输出分类标签就可以了，没必要让外界获得输出向量，同时，通过多种手段，如限制访问次数，设置验证码，限制访问权限等控制攻击方能获得的模型有关数据，从而减少模型知识泄露的风险。

2.对模型的输出内容进行扰动

如果我们能够对模型的输出进行一定扰动的同时又保留模型特征与模型输出的相关性，那么就可以做到在输出向量是有意义的情况下，最大程度增加攻击者窃取模型的难度和成本，课堂上讲到的针对成员推理共计的MemGuard算法即属于此类，我们也可以在神经网络添加一个附加层，在接受前面特征的输出并根据输出生成对应的可控扰动，最大限度减少攻击者一次查询从模型中获得的有效信息，并混淆攻击者用来窃取知识的模型。

3. 识别攻击性样本

为了提高攻击效率，选择一个好的迁移集生成策略和knockoff选取策略至关重要，而这样生成的用于输入原模型获得伪标签的数据样本也有一定的可区分性，有的学者提出了**广义OOD检测**这一方法来识别攻击性的查询数据（该方法一开始用在了对抗样本的辨别上，后面发现在攻击性样本的识别中也同样适用）因此我们可以首先训练一个攻击性样本分类器，对于识别为攻击性类样本最后的输出进行一定的混淆，从而使得攻击者的模型难以训练成功。