HW3

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？

模型逆向攻击时通过模型的输出， 反推出训练集中的某条目标数据的部分或全部属性值。 白盒模型逆向攻击是指攻击者指导模型的结构和参数的场景。 它适用于绝大部分分类器的模型攻击， 当我们知道标签`label`和 置信度向量， 我们可以通过攻击基本还原它在训练集中的输入。 而黑盒模型反演攻击的情况相对苛刻一点， 攻击者并不知道目标模型的结构， 也不知道参数和训练集。 攻击者实际的操作是通过训练一个反演模型将从目标模型训练的向量作为输入， 而得到一个最终重构的样本与目标模型的输入相似。 所以它作用的场景需要是比较容易获得目标模型训练集的辅助集的场景， 否则黑盒模型反演攻击难以部署。

若仅获得标签， 那么模型逆向攻击很难实施， 仅仅获得标签相当于得到了一个得到了一个分类$f_i(x)$，其中x为输入图像或者向量， 而`i`是输出的n维向量中唯一为1的元素， 其他维都是0。 （比如（0， 0， 0， 1， 0， 0， 0））。 对于攻击者来说这样的信息太过匮乏， 只可能在模型本身数据集很少且分类类别很少的情况下有可能攻击成功。 （比如一个具有少量数据集的而分类器。 ）

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。

模型窃取攻击的场景下， 攻击者攻击的模型参数不公开， 攻击者也不知道或者只知道部分模型结构信息和标签信息。 而基于替代模型的模型窃取攻击的攻击原理是在本地训练一个与目标模型任务相同的替代模型， 来达到与目标模型相似的性质。 因为这种攻击场景下， 攻击者对于目标模型的具体结构并不了解， 仅有一个提供模型访问的黑盒接口， 所以在本地训练模型的时候， 训练集的存在是必要的， 并且显然训练集量越大得到的效果越好， 若想解决查询过多的问题， 只能选择一个得到结果和查询次数的tradeoff， 1）可以不苛求极佳的替代模型而是仅获得一个功能相似的模型， 减少训练集的量， 2）在构造迁移集的步骤中， 采用自适应策略， 选择一些好的样本进行查询从而达到比较好的训练效果， 3）不同的替代模型结构达成的效果并不相同，虽然一般来说是复杂的结构得到的效果越好但也不绝对， 可以牺牲本地的算力同时构建国歌不同的替代模型来选择最好的替代模型， 减少查询次数。

（后面的问题就自己回答一下， 顺便复习一遍）

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， 你认为防御者改如何应对？

我认为这需要分情况讨论， 首先攻击者的扰动如果是随机扰动， 它可能本身就对自身得到的样本有了一定干扰作用。 如果攻击者真的能力很强加入可以破坏单词随机的设定的扰动并能进行模型攻击， 可以1）人工检测模型被异常行为查询的次数， 掌握攻击者数据以及加入的扰动并在本地进行学习对于攻击者对抗样本的对抗扰动。 2）也可以换一个针对攻击者换一个随机数生成算法， 只要保证相同图片随机数相同就好。

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。

借助在其他课上学过的： 多媒体设备中，有关于`copy-once`，`no-copy` 和 `copy-free` 三种不同水印，与生产播放设备的厂商相互作用， 遇到被copy的`no-copy` 载体则不予解码。 那么在人工智能模型的场景中， 如果可以联合所有模型的发明者或者组织， 共同在最后一层加一层水印检测的神经网络， 如果检测到水印并有个人私钥签名相对应， 那么给予正确的输出， 若是窃取模型则水印和私钥不对应， 则对输出加入随机扰动让结果趋于混乱。 除非攻击者自己从零写模型自己排除验证层的神经网络否则得到的就是随机的输出结果。 话说回来， 如果他有能力自己写模型， 也不至于成为模型的窃取者。

实作场景中以上策略部署难度过大， 首先它现式地增加了计算负担， 让本身的模型训练花费的算力更多， 其次人工智能发展至今， 模型的创作者太多了， 很难形成统一的社区文化和管理者， 整体部署的难度大。