图像分类模型的对抗攻击和对抗训练

张煌昭 †

I. 对抗攻击 (Adversarial Attack)

假设存在一图像模型（例如图像分类模型），可以接受图像输入，并产生正确的输出（例如预测分类）。**攻击算法**可以通过修改原本的图像数据，使得模型产生和原输出不同的输出。但是这样简单的修改非常容易被人类识破，把猫的图像直接换成狗，输入猫狗分类器，输出也一定就是狗。

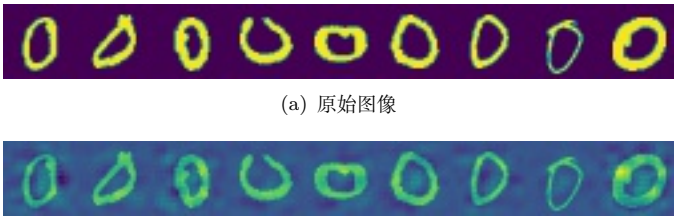
对抗攻击相对而言，更为精巧。对抗攻击算法通过扰动原本的图像数据，即在仅对图像进行非常小的修改的条件下，使得模型产生和原输出不同的输出结果。这种扰动，一般情况下是人类难以发觉的，甚至是肉眼无法分辨的。对抗攻击产生的样本，一般具有如下特征：

- 与原数据相比，变化极小
- 扰动会被人眼视为噪声，甚至人眼不可见
- 引起模型输出错误，但不会使得人类犯错

图1展示的是一组 MSNIT 数据集上的对抗样本的例子。在某个 MNIST 分类器上（该分类器在 test 集上的准确率超过 95%），图1(a)的九幅图像均可以被正确识别为 “0”；经过对抗攻击算法扰动之后，得到对抗样本如图1(b)，此时这九幅图像从左到右会被分类器依次识别为 “1” 到 “9”。

对抗攻击有两种场景–黑盒和白盒。在**黑盒攻击**的场景下，整个待攻击模型都处于黑盒中，其模型框架、模型结构等所有超参数，以及模型各层权重等参数均不可知，仅有一个 “predict” 接口可用。该接口允许输入图像数据，并返回模型输出的各个类别的概率。除此之外的所有操作均被禁止。在**白盒攻击**的场景下，整个待攻击模型均为透明，所有参数和超参数都可见，允许使用 “predict” 和 “gradient” 接口。“predict” 接口与黑盒攻击中相同；“gradient” 接口，允许输入图像数据和期望的输出，计算任意层上的（包括输入数据上的）的梯度并返回。禁止训练和直接给权重赋值的操作。

在这次作业中，我们统一规定黑盒模型有 “predict” 接口可用，白盒模型有 “predict” 和 “gradient” 接口可用。在本次作业的图像分类问题上，对两个接口的定义如下：“predict” 接口接受图像输入 x，返回模型 C 预测的概率向量 $\hat{y}$，$\hat{y}$ 的每一维对应 C 预测的该类别的概率；“gradient” 接口接收图像输入 x 和类别向量 y（y 是一个 One-hot 的概率向量，为 1 的一维表示类别），返回损失函数 $L(x, y|C) = CE(y, \hat{y})$ 对 x 的梯度 $\nabla_x L(x, y|C) = \frac{\partial L(x,y|C)}{\partial x}$。



(a) 原始图像

(b) 对抗样本

图 1. MNIST 数据集上的对抗样本实例–图1(a)为 9 幅原始的 MNIST 数据集中的 “0” 的图像，图1(b)为对抗攻击算法产生的 9 幅经过扰动的 “0” 的对抗样本图像，这九幅 “0” 的对抗样本图像，会被一 MNIST 分类器分别判别为 “1”-“9”。

一般而言，图像模型的白盒攻击非常简单，因为输入数据上的梯度可以明确指出扰动的方向；而黑盒攻击相对困难，因为需要随机尝试扰动的方向。但在 NLP 领域中，白盒攻击很困难，因为句子空间是离散空间，如何计算和利用梯度会很困难；而黑盒攻击相对简单，仅仅需要随机地进行扰动尝试即可。

在现实场景中，黑盒攻击占据大部分，因而仅仅在 “攻击” 这一层面上，黑盒攻击更为重要（例如攻击某个加密系统中的人脸识别系统或者虹膜识别系统）。但是，白盒攻击可以针对模型产生特定的样本，这些对抗样本可以被用来修复模型原本的缺陷。就像 “黑客” 和 “白客”，白盒攻击的目的并不是为了攻破模型，而是发现模型的缺陷，并辅助修复–因此在” 防御 “的层面上，白盒攻击更有意义。

A. 白盒攻击 (White-Box Attack)

分类器 C 接受输入图像 x 后产生概率向量 $\hat{y}$（每一维对应一类），损失函数 $L(x, y|C)$ 表示 $\hat{y} = C(x)$ 与

† 张煌昭，zhang_hz@pku.edu.cn

图像 x 的真实类别 y 的距离。$L(x,y|C)$ 越小说明 C 的预测与真实的类别越接近。在进行白盒攻击时，若我们期望分类器 C 给出的错误输出是 $\tilde{y}$，则需要对 x 扰动得到 $\tilde{x}$，并且使得 $L(\tilde{x},\tilde{y}|C)$ 尽量小。因此白盒攻击实质上是一个优化问题，即需要满足式1，其中 $D(a,b)$ 计算 a 和 b 的距离（例如欧氏距离）。式1表明，对抗样本 $\tilde{x}$ 与原样本 x 被要求尽量相近，而 $\tilde{x}$ 会使得模型 C 的输出为错误类别 $\tilde{y}$。

$$\min_{\tilde{x}} L(\tilde{x},\tilde{y}|C) \ \& \ \min_{\tilde{x}} D(\tilde{x},x) \tag{1}$$

一个最简单的白盒攻击算法就是梯度下降，即“固定模型调数据”。这种方法相对简单粗暴，更为高效的方法请参考相关论文。

- 初始化样本，$x^{(0)}=x$；
- 计算样本梯度，需要使用反向传播算法；

$$\nabla_{x^{(n)}} L(x^{(n)},\tilde{y}|C)=\frac{\partial L(x^{(n)},\tilde{y}|C)}{\partial x^{(n)}},\ n=0,1,2,\cdots$$

- 根据梯度扰动样本，α 为学习速率；

$$x^{(n+1)}=x^{(n)}-\alpha\cdot\nabla_{x^{(n)}} L(x^{(n)},\tilde{y}|C),\ n=0,1,2,\cdots$$

- 重复迭代，直至收敛，返回 $\tilde{x}=x^{(n)}$。

$$arg\max C(x^{(n)})=arg\max\tilde{y} \tag{2}$$

B. 黑盒攻击 (Black-Box Attack)

黑盒攻击的目标仍是解相同的优化问题，但是在求解过程中分类器 C 始终为黑盒，即不能计算任何梯度。因此在第I-B节提到的梯度下降的方法并不适用。

可以使用 MCMC 采样 (Monte Carlo Markov chain sampling) 进行黑盒攻击，MCMC 采样可以被形象地理解为“走一步看一步”。

- 初始化样本，$x^{(0)}=x$；
- 根据 Markov 转移概率 $g(x|x^{(n)})$ 产生候选样本 x'；
- 若 $u\le C(x')[arg\max\tilde{y}]$，则拒绝 x'，其中随机变量 $u\sim U(0,1)$，$C(x')[arg\max\tilde{y}]$ 表示分类器在以 x' 为输入时，在目标类别上的预测概率；否则，接受 x'；
- 若拒绝 x'，则重新产生新的候选样本；否则，接受转移，$x^{(n+1)}=x'$；
- 重复迭代，直至收敛（满足式2），返回 $\tilde{x}=x^{(n)}$。

转移概率 $g(x|x^{(n)})$ 基本可以任意指定（满足 Markov 链的遍历性和非周期性即可），举个例子，我们可以指定 $g(x|x^{(n)})$ 是以 $x^{(n)}$ 为中心，以 σ^2 为方差的正态分布。需要说明的是，为了保证 $\tilde{x}$ 和 x 尽量相近，我们可以设定迭代中单步扰动的变化阈值，一旦单步扰动的变化过大，则直接拒绝该候选样本，即若 $D(x',x^{(n)})>\Delta_{max}$，则拒绝候选样本 x'，其中 Δ_{max} 为单步扰动的变化阈值。

另一种比较简单的黑盒攻击的方法是迁移白盒攻击产生的样本。

- 训练一个与待攻击的黑盒分类器 C 完全独立的白盒分类器 C'；
- 使用第I-A节中的方法，对 C' 进行白盒攻击，得到待迁移样 $\tilde{x}'$；
- 若 $\tilde{x}'$ 可以攻击 C，则返回 $\tilde{x}=\tilde{x}'$；否则，重新产生待迁移样本。

尽管看起来样本迁移的方法过于简单，但实际上，在某个深度神经网络分类器上白盒攻击得到的对抗样本，有大概率可以攻击其他的超参数不同，甚至结构不同的深度神经网络模型。这些对抗样本，甚至可以攻击一些非深度模型（例如决策树等）。

II. 对抗训练 (Adversarial Training

对抗训练首先需要训练一个模型；之后攻击该模型，在**训练集**上产生一批对抗样本；再将这些样本（用原本的正确标签标注）掺入训练集中，得到新的训练集；最后利用新数据集重新训练一个结构和超参数相同的模型。对抗训练可以修复模型原本的缺陷，达到提升性能（例如提升准确率）以及抵抗对抗攻击（例如降低攻击成功率）的效果。

以图像分类举例，简单说明对抗训练为什么可以提升性能。经过在训练集上若干轮的训练，分类器不断调整自己的各个参数，实质上表现为调整在图像超空间里的分类超平面。根据训练集的数据学习到的超平面是不可能和真实数据的分类超平面完全重合的。白盒训练得到的对抗样本的会略微跨过模型的分类超平面，而并未跨过真实数据的分类超平面。因此将这些对抗样本重新加入到训练集中，可以帮助修复模型的分类面偏离了真实数据的分类面的地方。

III. 作业要求

A. 数据集

我们规定使用 Fashion MNIST 数据集。这是一个类 MNIST 数据集，数据集的具体介绍请见 README.md。

Fashion MNIST 数据集位于 data 目录中，数据集的 python3 封装位于 code/fmnist_dataset.py 文件中。

此外，我们还提供一个已经训练好的 CNN 模型（test 集准确率达到 90%）以及 1000 个样本用于黑盒攻击。CNN 模型的代码位于 code/model.py 文件中，模型存放于 model/model.ckpt 断点文件中，用于攻击的样本位于 attack_data 目录下。[1]

B. 白盒攻击

要求对自己训练的模型进行定向白盒攻击，即原本被正确分类的图像，在扰动后被判别为指定类别，原类别与目标类别一一对应，如表I所示。要求：

- 训练一个 Fashion MNIST 上的图像分类模型，要求分类准确率不低于 90%；
- 在 **test 集**中随机选出至少 1000 张可以被分类器正确分类的图像；
- 对选出的图像进行白盒攻击，得到对抗样本。

白盒攻击的算法不限，鼓励阅读相关论文并使用别的算法。需要提交的内容包括：

- 分类器在 test 集上的准确率，以及分类器源码；
- 白盒攻击的成功率，若攻击算法在迭代上限或运行时间上限前产生出可以使得分类器判别为类别 B 的样本，则认为成功，否则认为失败定；
- 在攻击成功的样本中随机选出 10 组，提交 10 张原图像和 10 张对抗样本图像，以及分类器对它们的判别类别。

C. 黑盒攻击

要求对一个黑盒模型进行定向黑盒攻击，原类别与目标类别的对应关系如表I所示。黑盒攻击的待攻击模型以及样本已经提供，见第III-A节。

黑盒攻击的算法不限，可以使用采样的方法直接产生对抗样本进行攻击，也可以使用样本迁移的方法，通过白盒攻击产生一批样本迁移至黑盒模型尝试进行攻击。需要注意，样本迁移时，需要在我们提供的样本上产生对抗样本，而不能直接使用第III-B节中得到的对抗样本。需要提交的内容包括：

- 黑盒攻击的成功率；
- 在攻击成功的样本中随机选出 10 组，提交 10 张原图像和 10 张对抗样本图像，以及分类器对它们的判别类别。

[1]项目位于https://github.com/LC-John/Fashion-MNIST

原正确类别	0	1	2	3	4	5	6	7	8	9
指定的目标类别	1	2	3	4	5	6	7	8	9	0

表 I
原正确类别与目标类别的对应关系

除了上述的黑盒攻击，还需要对第III-B节自己训练的模型，在取出的正确分类的图像上，进行黑盒攻击得到成功率。这一结果需要在第III-D节中进行对比。

注意！黑盒模型不允许被打开！不允许被修改！不允许被用于白盒攻击！

D. 对抗训练

要求使用与第III-B节中的白盒模型，进行对抗训练，并测试模型的性能和鲁棒性是否会有提升。要求：

- 使用第III-B节中自己训练的模型，在**训练集**中随机选出至少 1000 张可以被分类器正确分类的图像，对齐进行白盒攻击，得到对抗样本；
- 将对抗样本掺入训练集中，使用新的训练集重新独立训练一个与第III-B节中的模型设置完全相同的分类器；
- 对比新旧分类器在 test 集上的准确率；
- 在新分类器上重复第III-B节中的白盒攻击，对比相同的白盒攻击在新旧分类器上的攻击成功率；
- 在新分类器上重复第III-B节中的黑盒攻击，对比相同的黑盒攻击在新旧分类器上的攻击成功率。

由于数据集相对很容易，因此可能并不会有明显的性能或鲁棒性的提升。需要提交的内容包括：

- 对抗训练得到的新分类器和第III-B节中的旧分类器在 test 集上的准确率；
- 第III-B节中的白盒攻击在新分类器和就分类器上的攻击成功率；
- 第III-C节中的黑盒攻击在新分类器和就分类器上的攻击成功率；
- 在新分类器上，白盒攻击成功的样本中随机选出 10 组，提交 10 张原图像和 10 张对抗样本图像，以及分类器对它们的判别类别。
- 在新分类器上，黑盒攻击成功的样本中随机选出 10 组，提交 10 张原图像和 10 张对抗样本图像，以及分类器对它们的判别类别。