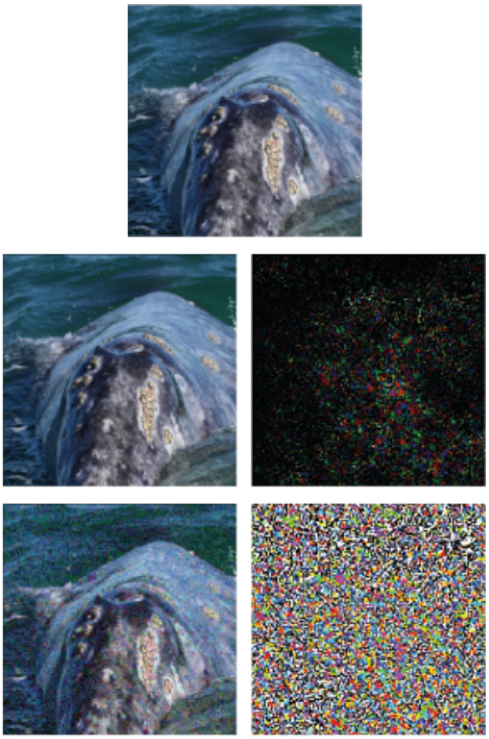
DeepFool算法论文导读

简介

假设有原始样本 x，其标签为 y，我们期望通过对原始样本添加人类不容易发觉的噪声 r，生成对抗样本 x^*，使分类器所生生成的对抗样本的标签判定为 y^*，其中 $y \neq y^*$。本篇论文则是期望可以找到最小的 r。



如上图所示：

第一行是原始样本，分类器将其分类为鲸。第二行和第三行分别为Deepfool算法和FGSM算法所生成的对抗样本及它们添加的噪声。虽然这两种算法所添加的噪声都让分类器将样本误判为乌龟，但是可以明显看出Deepfool算法所添加的噪声是比较小的。

鲁棒性定义

给定一个分类器，样本鲁棒性是使得模型出现误分类的最小扰动，具体形式如下：

$$\Delta(\boldsymbol{x};\hat{k}) := \min_{\boldsymbol{r}} \|\boldsymbol{r}\|_2 \text{ subject to } \hat{k}(\boldsymbol{x}+\boldsymbol{r}) \neq \hat{k}(\boldsymbol{x})$$

其中，x为干净的样本，$\hat{k}(x)$为模型预测的标签。 $\Delta(x;\hat{k})$为样本 x 在模型分类器 $\hat{k}$ 的鲁棒性。进而作者又定义出了模型在整个数据集上的鲁棒性，这是一种期望的形式，具体形式为:

$$\rho_{\text{adv}}(\hat{k}) = \mathbb{E}_{\boldsymbol{x}} \frac{\Delta(\boldsymbol{x};\hat{k})}{\|\boldsymbol{x}\|_2}$$

模型鲁棒性是更好地理解当前网络体系结构的局限性和设计增强健壮性的方法的关键。

DeepFool攻击二分类器

原理推导

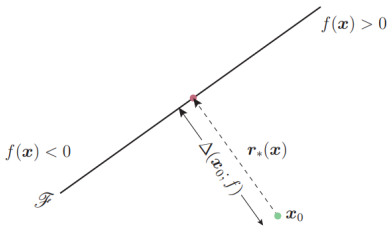


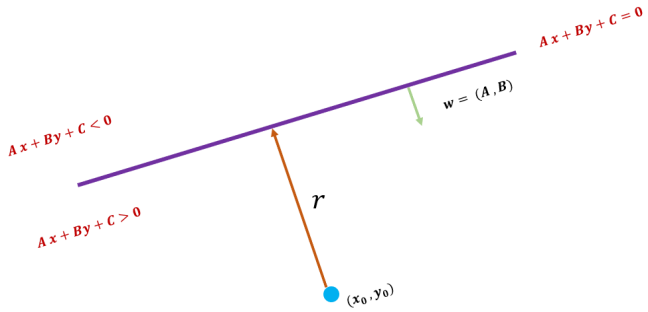
Figure 2: Adversarial examples for a linear binary classifier.

上图为对抗样本攻击线性分类器的图示，其中 $f(x) = w^\intercal + b$ 为一个二元分类器。$\Delta(x_0; f)$为干净样本点x_0的最短距离，即为样本点x_0在分类器f中的鲁棒性。$F = \{x : w^\intercal x + b = 0\}$ 为分类的超平面。

具体有如下目标函数：

$$\begin{aligned}\boldsymbol{r}_*(\boldsymbol{x}_0) &:= \arg\min \|\boldsymbol{r}\|_2 \\ &\text{s.t. } \operatorname{sign}(f(\boldsymbol{x}_0+\boldsymbol{r})) \neq \operatorname{sign}(f(\boldsymbol{x}_0)) \\ &= -\frac{f(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}\end{aligned}$$

> 为了更好的理解上述结论，补充以下内容：



如上图所示，在二维平面中，有一条直线 $Ax+By+C=0$ 和一个点 (x_0,y_0)，其中直线的法向量为 $w=(A,B)$，由点到直线距离知识可知点(x_0,y_0)到直线 $Ax+By+C=0$为：

$$||r||=\frac{Ax_0+By_0+C}{\sqrt{A^2+B^2}}$$

点 (x_0,y_0)移动到直线的位移为（需要注意的是位移带方向）：

$$r=-\frac{Ax_0+By_0+C}{\sqrt{A^2+B^2}}e$$

因此二元分类器的公式为：

$$r(x)=-\frac{f(\boldsymbol{x})}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}$$

上面目标函数可以通过迭代的方式来进行求解以获得最小对抗扰动，可以重新转换成如下的优化形式：

$$\arg\min_{\boldsymbol{r}_i}\|\boldsymbol{r}_i\|_2 s.t.\ f(\boldsymbol{x}_i)+\nabla f(\boldsymbol{x}_i)^T\boldsymbol{r}_i=0$$

“

下面为具体的推导过程：

$$r_i=-\frac{f(x_i)}{||w||_2^2}w$$

$\nabla f(x_i)=\frac{w}{||w||^2}$，所以有：

$$r_i=-f(x_i)\nabla f(x_i)$$

又因为梯度的模长为 1，所以两边同乘以 $\nabla f(x_i)^\intercal$：

$$\nabla f(x_i)^T r_i=-\nabla f(x_i)^T f(x_i)\nabla f(x_i)$$

最后，移项可得到最终论文中给出的公式:

$$f(\boldsymbol{x}_i) + \nabla f(\boldsymbol{x}_i)^T \boldsymbol{r}_i = 0$$

• 算法流程

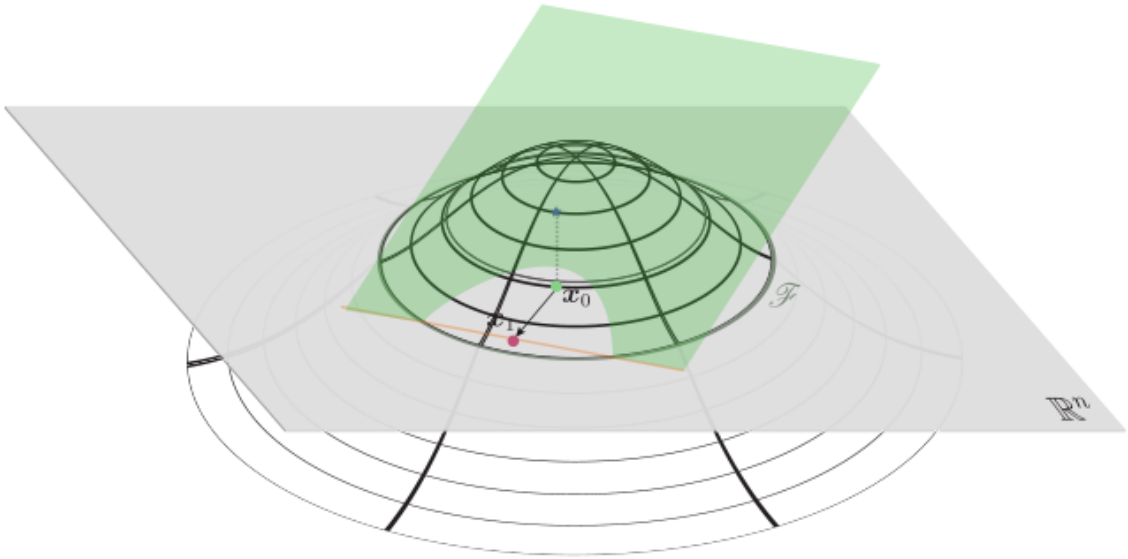
根据这个迭代公式有如下算法：

Algorithm 1 DeepFool for binary classifiers

input: Image $\boldsymbol{x}$, classifier f.
output: Perturbation $\hat{\boldsymbol{r}}$.
Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
while $\text{sign}(f(\boldsymbol{x}_i)) = \text{sign}(f(\boldsymbol{x}_0))$ **do**
    $\boldsymbol{r}_i \leftarrow -\frac{f(\boldsymbol{x}_i)}{\|\nabla f(\boldsymbol{x}_i)\|_2^2} \nabla f(\boldsymbol{x}_i)$,
    $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$,
    $i \leftarrow i + 1$.
end while
return $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$.

下面是对该算法更形象的解释：

以 $n = 2$ 为例，$x_0 \in R^n$,绿色平面是 x_0 点处的切平面为 $f(x_0) + \nabla f(x_0)^{\intercal}(x - x_0)$，此时会投影出现迭代点 x_1，然后以此类推，最终将所有的扰动综合起来即为所求的对抗扰动：



DeepFool攻击多分类器

• 原理推导

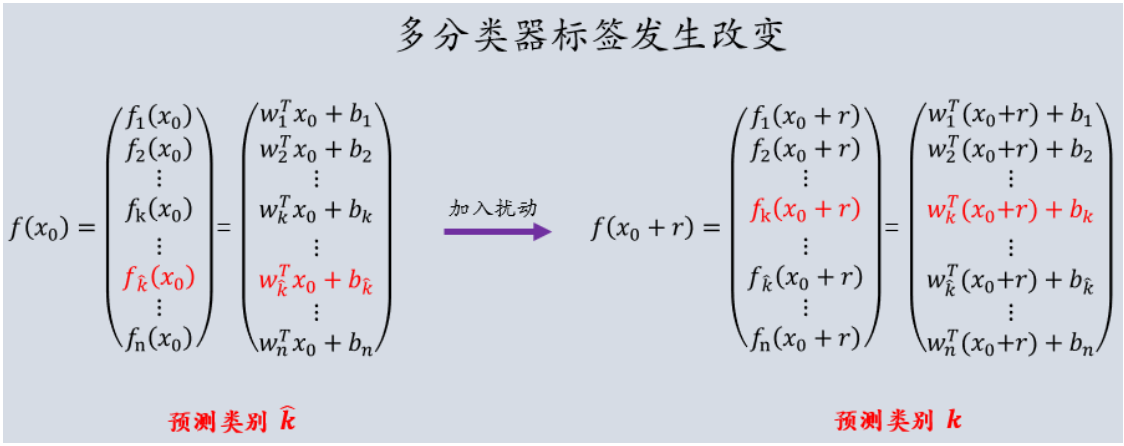
分类器预测标签如下公式所示：

$$\hat{k}(\boldsymbol{x}) = \arg\max_{k} f_k(\boldsymbol{x})$$

其中，$f_k(x)$ 是预测概率向量 $f(x)$ 的第 k 类的概率分量。多分类器模型添加扰动的优化函数如下所示：

$$\arg\min_{r} \|\boldsymbol{r}\|_2$$
$$\text{s.t. } \exists k : \boldsymbol{w}_k^{\top}(\boldsymbol{x}_0 + \boldsymbol{r}) + b_k \geq \boldsymbol{w}_{k(\boldsymbol{x}_0)}^{\top}(\boldsymbol{x}_0 + \boldsymbol{r}) + b_{k(\boldsymbol{x}_0)}$$

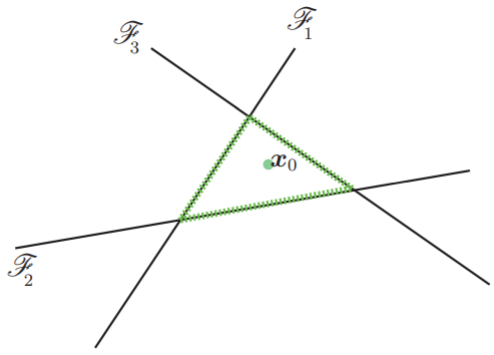
> 为了更好理解上述优化形式的含义，见下图：



> 左半部分是干净样本的概率向量的输出，预测的类别是 $\hat{k}$,加入对抗扰动之后，预测类别变成了 k

举个栗子

如下图实例所示，已知一个四分类器，干净样本 x_0 被分为第四类中。 F_k 是分类的边界超平面。要寻找最小扰动，只需要分别计算 x_0 到三个分类面的距离，并取其中最小的一个值作为最小扰动。



绿色区域为一个凸区域可以表示为：

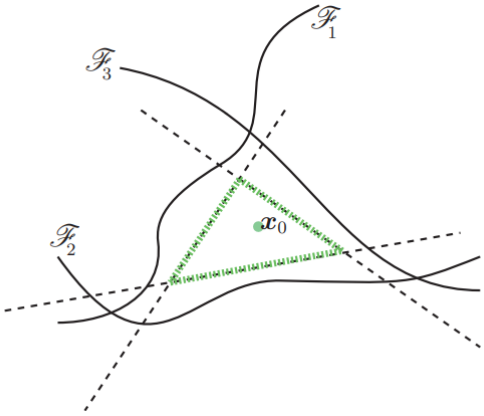
$$P = \bigcap_{k=1}^{c} \left\{ \boldsymbol{x} : f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}) \geq f_k(\boldsymbol{x}) \right\}$$

根据点到直线的距离公式，可以推算得知点 x_0 到三条边界最短的距离的计算公式为:

$$\hat{l}\left(\boldsymbol{x}_0\right)=\arg\min_{k\neq\hat{k}(\boldsymbol{x}_0)}\frac{\left|f_k(\boldsymbol{x}_0)-f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\left\|\boldsymbol{w}_k-\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\right\|_2}$$

与 DeepFool 攻击二分类器相似，则多分类器的对抗扰动为：

$$\boldsymbol{r}_*(\boldsymbol{x}_0)=\frac{\left|f_{\hat{l}\,(\boldsymbol{x}_0)}(\boldsymbol{x}_0)-f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\left\|\boldsymbol{w}_{\hat{l}\,(\boldsymbol{x}_0)}-\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\right\|_2^2}\left(\boldsymbol{w}_{\hat{l}\,(\boldsymbol{x}_0)}-\boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\right)$$

> 更加通用的多分类器如下：
>
> 
>
>
> 其中绿色区域可以表示为：
>
> $$\tilde{P}_i=\bigcap_{k=1}^{c}\Big\{\boldsymbol{x}:f_k(\boldsymbol{x}_i)-f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)+\nabla f_k(\boldsymbol{x}_i)^{\top}\boldsymbol{x}-\nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)^{\top}\boldsymbol{x}\leq 0\Big\}$$
>
> 图中实线表示分类器真实的分类超平面，而虚线则代表近似的线性分类超平面，在每次迭代过程中，总是基于当前的迭代值，计算一组近似的线性分类超平面，并根据这组近似超平面，计算扰动，并进行迭代得到下一次的迭代值。

算法流程

综合以上 DeepFool 攻击多分类器模型的原理获得如下的算法流程：

Algorithm 2 DeepFool: multi-class case

input: Image $\boldsymbol{x}$, classifier f.
output: Perturbation $\hat{\boldsymbol{r}}$.

Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
while $\hat{k}(\boldsymbol{x}_i) = \hat{k}(\boldsymbol{x}_0)$ **do**
    for $k \neq \hat{k}(\boldsymbol{x}_0)$ **do**
        $\boldsymbol{w}'_k \leftarrow \nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
        $f'_k \leftarrow f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
    end for
    $\hat{l} \leftarrow \arg\min_{k \neq \hat{k}(\boldsymbol{x}_0)} \frac{|f'_k|}{\|\boldsymbol{w}'_k\|_2}$
    $\boldsymbol{r}_i \leftarrow \frac{|f'_{\hat{l}}|}{\|\boldsymbol{w}'_{\hat{l}}\|_2^2} \boldsymbol{w}'_{\hat{l}}$
    $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$
    $i \leftarrow i + 1$
end while
return $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$