对抗样本简介—从基于迁移和基于查询与先验的角度

所有的代码位于https://github.com/deadfffool/ADV，这是我之前学习对抗样本时候的实验仓库

Abstract

如今，深度学习已被广泛应用于图像分类和图像识别的问题中，取得了令人满意的实际效果，成为许多人工智能应用的关键所在.在对于模型准确率的不断探究中，研究人员在近期提出了“对抗样本”这一概念.通过在原有样本中添加微小扰动的方法，成功地大幅度降低原有分类深度模型的准确率，实现了对于深度学习的对抗目的，同时也给深度学习的攻方提供了新的思路，对如何开展防御提出了新的要求.在介绍对抗样本生成技术的起源和原理的基础上，对近年来有关对抗样本的研究和文献进行了总结，主要集中在基于迁移和基于查询的角度。

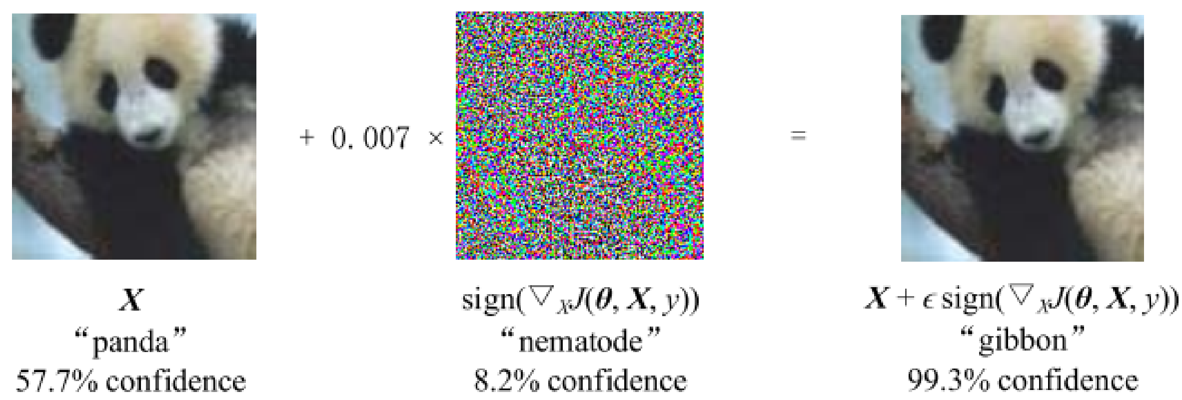
Introduction

随着深度学习的快速发展与巨大成功，深度学习被应用在许多对安全有严格要求的环境中。然而，深度神经网络近来被发现，对于精心设计好的输入样本，其是脆弱的，这种样本就被称为**对抗样本**。对抗样本对人类是很容易分辨的，但却能在测试或部署阶段，很容易的糊弄深度神经网络。当应用深度神经网络到对安全有严格要求的环境中时，处理对抗样本造成的脆弱性变成已成了一个重要的任务。因此对抗样本的**攻击**和**防御**吸引了很大的注意。

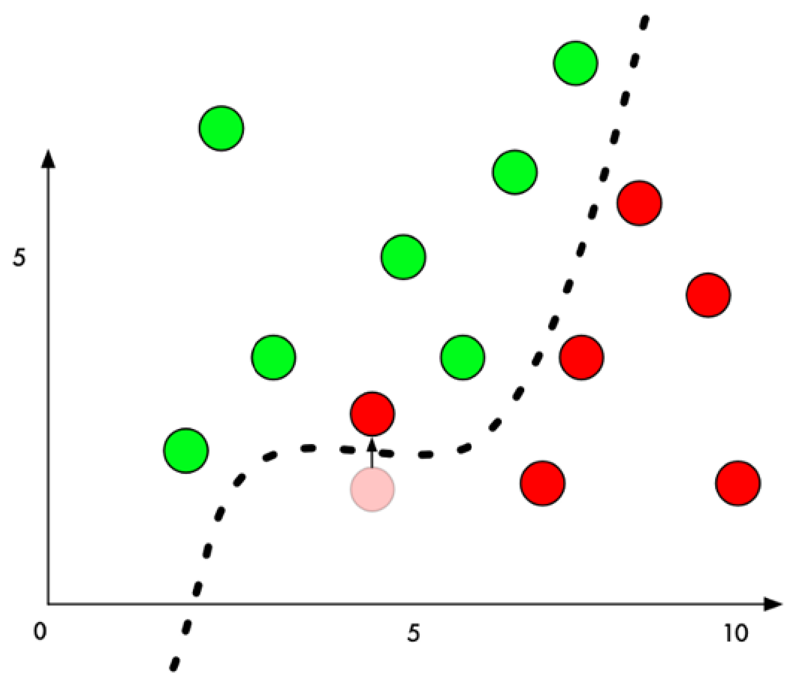
对抗样本由 Christian Szegedy 等人提出，是指在数据集中通过故意添加细微的干扰所形成的输入样本，这种样本导致模型以高置信度给出一个错误的输出。在正则化背景下，通过对抗训练减少原有独立同分布的测试集的错误率，在对抗扰动的训练集样本上训练网络。

Preliminaries

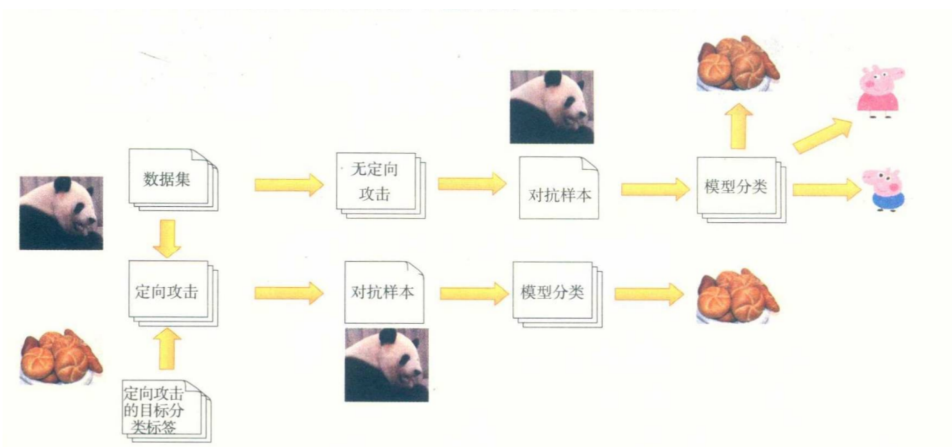
简单地讲，对抗样本通过在原始数据上叠加精心构造的人类难以察觉的扰动，使深度学习模型产生分类错误。以图像分类模型为例，如图所示，通过在原始图像上叠加扰动，对于肉眼来说，扰动非常细微，图像看起来还是能猫，但是图像分类模型却会以很大的概率识别为长臂猿。



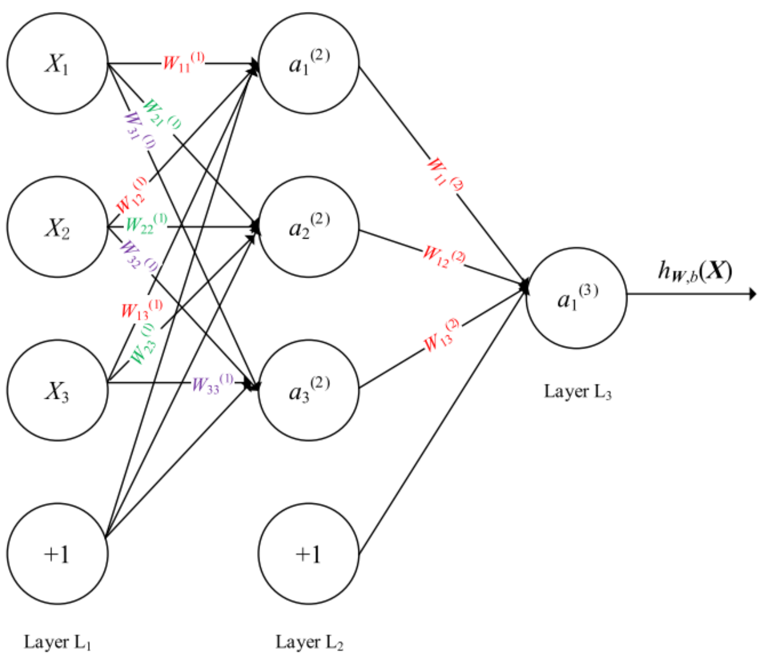
下面以一个图像分类模型为例，更加直接地解释对抗样本的基本原理。通过在训练样本上学习，学到一个分割平面，在分割平面一侧的为绿球，在分割平面另外一侧的为红球。生成攻击样本的过程，就是在数据上添加一定的扰动，让其跨越分割平面，从而把分割平面一侧的红球识别为绿球，如下图所示。



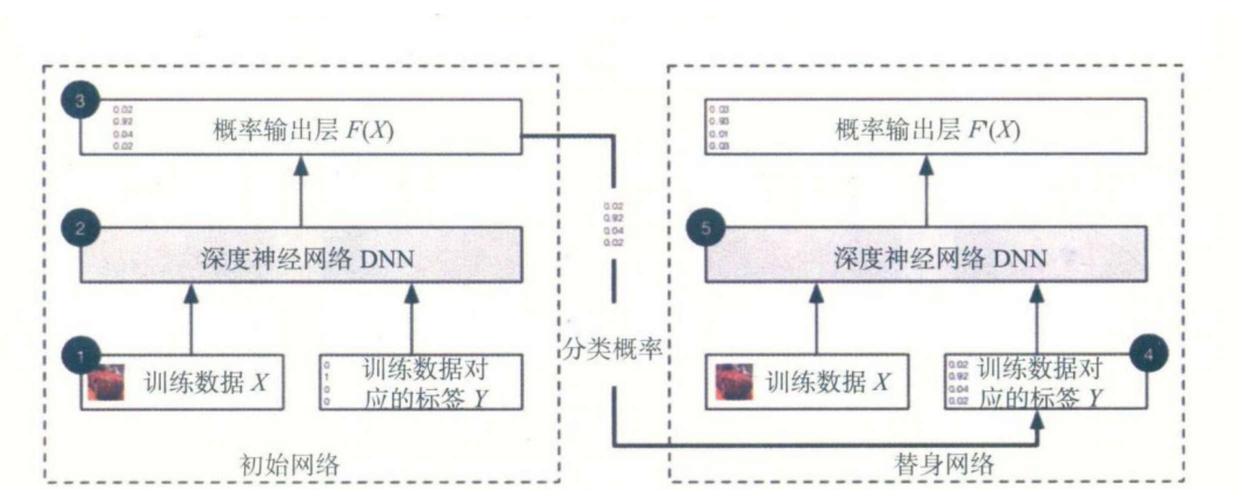
对抗样本按照攻击后的效果分为 Targeted Attack (定性攻击)和 Non-Targeted Attack(无定向攻击)。区别在于 Targeted Attack 在攻击前会设置攻击的目标，比如把红球识别为绿球，或者把面包识别为熊猫，也就是说在攻击后的效果是确定的; Non-Targeted Attack在攻击前不用设置攻击目标，只要攻击后，识别的结果发生改变即可，可能会把面包识别为熊猫，也可能识别为小猪佩琪或者小猪乔治，如图所示。



对抗样本按照攻击成本分为 White-Box Attack(白盒攻击)Black-Box Attack(黑盒攻击)和 Real-World Attack/PhysicalAttack (真实世界/物理攻击)。White-Box Attack是其中攻击难度最低的一种，前提是能够完整获取模型的结构，包括模型的组成以及隔层的参数情况，并且可以完整控制模型的输入，对输人的控制粒度甚至可以到比特级别。由于 White-Box Attack 前置条件过于苛刻，通常作为实验室的学术研究或者作为发起 Black-Box Attack的基础。



Black-Box Attack 相对 White-Box Attack 攻击难度具有很大提高，Black-Box Attack完全把被攻击模型当成一个黑盒，对模型的结构没有了解，只能控制输入，通过比对输入和输出的反馈来进行下一步攻击，如下图所示。



Adversarial Examples

Basic Method

FGM/FGSM

论文为 Explaining and Harnessing Adversarial Examples
https://arxiv.org/abs/1412.6572v3

FGM也被称作FGSM，快速梯度算法，fast gradient method，可以作为无定向攻击和定向攻击算法使用。假设图片原始数据为x，图片识别的结果为y，原始图像叠加上的细微

变化为η，肉眼难以识别，公式如下$\widetilde{x} = x + \eta$

将修改的图像传入模型之后会与参数矩阵和激活函数相作用，我们的目标是追求微小的修改来对分类的结果产生变化，因此我们采用$sign$函数，将变化量与梯度的方向相一致，就可以对分类结果产生较大的变化。当x的维度为n时，模型在每个维度的平均值为m，η的无穷范数为ε，每个维度的微小改变都与函数梯度的方向一致，累计的效果就为$n*m*\varepsilon$，当数据的维度很大时，即使η很小，对最后结果的影响也可能很大。优化公式如下

$$\omega^T x = \omega^T x + \omega^T \eta$$

$$x' = x - \varepsilon * sign(\nabla loss_{F,t}(x))$$

DeepFool

论文为 CVPR 2016 DeepFool: a simple and accurate method to fool deep neural networks

Deepfool的思想就是利用迭代的方式一步步向着分类的决策边界移动，为了找到最小的决策边界，我们利用`while`循环找出最小扰动可以到达的边界，然后通过找出扰动方向，利用梯度不断向边界靠近，实现分类的攻击

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 $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$

JSMA

论文为 IEEE 2016 The Limitations of Deep Learning in Adversarial Settings https://arxiv.org/pdf/1511.07528.pdf

JSMA引入了显著图(Saliency Map)的概念，该算法致力于用扰动较少的像素点来完成定向攻击，所以从Saliency Map中查找需要扰动的像素点

Saliency Map的生成方式为对原始标签和其他标签求梯度，找到是原始标签损失上升而其他标签损失下降最快的点，并执行迭代更新

Algorithm 3 Increasing pixel intensities saliency map
$\nabla \mathbf{F}(\mathbf{X})$ is the forward derivative, Γ the features still in the search space, and t the target class

Input: $\nabla \mathbf{F}(\mathbf{X})$, Γ, t

```
for each pair (p, q) ∈ Γ do
    α = Σ_{i=p,q} ∂F_t(X)/∂X_i
    β = Σ_{i=p,q} Σ_{j≠t} ∂F_j(X)/∂X_i
    if α > 0 and β < 0 and −α × β > max then
        p1, p2 ← p, q
        max ← −α × β
    end if
end for
return p1, p2
```


C&W攻击算法

论文为 Towards Evaluating the Robustness of Neural Networks
https://arxiv.org/pdf/1608.04644.pdf

CW通常被认为是攻击能力最强的白盒攻击算法之一，达到了但是的SOTA，是一种基于优化的算法，CW算法的论文打破的防御蒸馏这种对抗防御的方法，CW的零感来在于原来的Box-Constrained L-BFGS，论文中的优化目标如下所示

$$mimimize \quad D(x, x+\delta)$$

$$such\ that \quad C(x+\delta) = t,\ x+\delta \in [0,1]^n$$

作者尝试了不同的loss，并测试了他们的表现，发现f6在实验中表现最好，在后续许多攻击中也采用$f6$这种loss

$$
\begin{aligned}
f_1(x') &= -\text{loss}_{F,t}(x') + 1 \\
f_2(x') &= (\max_{i \neq t}(F(x')_i) - F(x')_t)^+ \\
f_3(x') &= \text{softplus}(\max_{i \neq t}(F(x')_i) - F(x')_t) - \log(2) \\
f_4(x') &= (0.5 - F(x')_t)^+ \\
f_5(x') &= -\log(2F(x')_t - 2) \\
f_6(x') &= (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+ \\
f_7(x') &= \text{softplus}(\max_{i \neq t}(Z(x')_i) - Z(x')_t) - \log(2)
\end{aligned}
$$

关于box constraints，作者利用变量变换，引入新的变量w，将对抗样本表示为下图所示，这样子可以有效保障我们的优化不会溢出且具有良好的梯度。

$$x+\delta=\frac{1}{2}(tanh(\omega)+1)$$

最后的优化目标如下（采用l2范数攻击）

$$mimimize \quad ||\frac{1}{2}(tanh(\omega)+1)||_2^2+c*f(\frac{1}{2}(tanh(\omega)+1))$$

$$with\ f\ defined\ as \quad f(x')=max(max\{Z(x')_i\ i\neq t\}-Z(x')_t,-\kappa)$$

PGD攻击算法

PGD证明了明确了对抗样本需要解决的问题，将其归结于一个最大最小化问题

$$\min_{\theta}(\theta), \quad where\ \rho(\theta)=E_{(x,y)}[\max_{\delta\in S}L(\theta,x+\delta,y)]$$

这个公式给了一种统一的视角，把这类鞍点问题看作内部最大化和外部最小化问题的组合，内部问题利用给定数据点x来找到一个具有高损失的对抗样本，外部最小化问题是找到合适的模型参数，使内部攻击模型的损失最小化。
PGD攻击的方式为

$$x^{t+1}=\prod_{x+S}(x^t+\alpha sign(\nabla_x L(\theta,x,y)))$$

PGD的特点之一就在于它在Constrain内随机重启，作者发现在对抗样本生成空间内有很多局部的最优解，FGSM类算法并没有完全捕获到攻击空间的丰富度，随机重启可以找到所有空间内的一阶对抗样本（他们基本上是正交的，也没有极端的异常值）
通过Danskin‘s theorem可以知道该鞍点中内部最大化的方向恰好也是外部最小化的方向，我们将对抗样本加入到训练集中，便可以训练出更加鲁棒的神经网络。

Black-Box Attacks using Transferable Adversarial Examples

MI-FGSM攻击算法

论文为 Boosting Adversarial Attacks with Momentum

MI-FGSM 将动量相关的梯度整合进对抗样本的迭代过程中，在训练的过程中，使用动量法可以有效的稳定更新的方向，跳出局部极值，使得对抗样本获得更好的迁移性。

MI-FGSM也可以运用到集成攻击中，进一步提高迁移性

Algorithm 1 MI-FGSM

Input: A classifier f with loss function J; a real example $\boldsymbol{x}$ and ground-truth label y;
Input: The size of perturbation ϵ; iterations T and decay factor μ.
Output: An adversarial example $\boldsymbol{x}^*$ with $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \le \epsilon$.

$\alpha = \epsilon/T$;
$\boldsymbol{g}_0 = 0$; $\boldsymbol{x}_0^* = \boldsymbol{x}$;
for $t = 0$ to $T-1$ **do**
    Input $\boldsymbol{x}_t^*$ to f and obtain the gradient $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)$;
    Update $\boldsymbol{g}_{t+1}$ by accumulating the velocity vector in the gradient direction as

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)\|_1}; \tag{6}$$

    Update $\boldsymbol{x}_{t+1}^*$ by applying the sign gradient as

$$\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \alpha \cdot \mathrm{sign}(\boldsymbol{g}_{t+1}); \tag{7}$$

end for
return $\boldsymbol{x}^* = \boldsymbol{x}_T^*$.

NI-FGSM攻击算法

论文来自我们学校的何琨老师 ICLR 2020 Nesterov accelerated gradient and scale invariance for adversarial attacks

如果说MI-FGSM借鉴了梯度下降中的momentum算法，NI-FGSM便是借鉴了Nesterov Accelerated Gradient（NAG）算法，该算法的公式如下。

$$d_i = \beta_{i-1} + g(\theta_{i-1} - \alpha\beta d_{i-1})$$

$$\theta_o = \theta_{i-1} - \alpha d_i$$

该算法的想法很简单，在momentum项中，我们会利用以前的梯度，那么既然已经知道一定会走$\alpha * \beta * d_{i-1}$,何必还要用原来那一个点的梯度，直接利用走一步之后那个点的梯度。在相关的数学分析后，我们会发现这个操作实际上是利用了部分二阶导数的信息，来达到稳定梯度更新方向的效果，达到更好的收敛效果，即NI-FGSM比MI-FGSM具有更好的前瞻性

Algorithm 1 SI-NI-FGSM

Input: A clean example x with ground-truth label y^{true}; a classifier f with loss function J;
Input: Perturbation size ϵ; maximum iterations T; number of scale copies m and decay factor μ.
Output: An adversarial example x^{adv}

$\alpha = \epsilon/T$
$g_0 = 0$; $x_0^{adv} = x$
for $t = 0$ to $T-1$ **do**
    $g = 0$
    Get x_t^{nes} by Eq.(6)   ▷ make a jump in the direction of previous accumulated gradients
    for $i = 0$ to $m-1$ **do**   ▷ sum the gradients over the scale copies of the input image
        Get the gradients by $\nabla_x J(S_i(x_t^{nes}), y^{true})$
        Sum the gradients as $g = g + \nabla_x J(S_i(x_t^{nes}), y^{true})$
    Get average gradients as $g = \frac{1}{m} \cdot g$
    Update g_{t+1} by $g_{t+1} = \mu \cdot g_t + \frac{g}{\|g\|_1}$
    Update x_{t+1}^{adv} by Eq.(8)
return $x^{adv} = x_T^{adv}$

VM（N）I-FGSM攻击算法

同样来自何琨老师，CVPR 2021 Enhancing the Transferability of Adversarial Attacks through Variance Tuning

VMI-FGSM不再直接使用前一步的梯度进行梯度累计，而是进一步考虑前一步迭代的梯度方差来调整当前梯度，从而稳定梯度方向。

梯度方差为

$$V_{\epsilon'}^{g}(x) = \mathbb{E}_{\|x'-x\|_p<\epsilon'}[\nabla_{x'}J(x',y;\theta)] - \nabla_x J(x,y;\theta).$$

由于输入空间的连续性，我们无法计算周围空间的数学期望，所以采用sample的方式，在N的样本里取平均值达到期望的无偏估计的效果

$$V(x) = \frac{1}{N}\sum_{i=1}^{N}\nabla_{x^i}J(x^i,y;\theta) - \nabla_x J(x,y;\theta). \qquad (7)$$

Here $x^i = x + r_i$, $r_i \sim U[-(\beta\cdot\epsilon)^d, (\beta\cdot\epsilon)^d]$, and $U[a^d, b^d]$ stands for the uniform distribution in d dimensions.

VMI-FGSM受方差缩减方法的启发，这类方法可以有效稳定更新的梯度之间的方差，稳定更新方向，更快更好得达到极值点。经典的SAG和SVRG算法的核心思想就是通过采样构建当前梯度的无偏估计，加速收敛。

Algorithm 1 VMI-FGSM

Input: A classifier f with parameters θ, loss function J
Input: A raw example x with ground-truth label y
Input: The magnitude of perturbation ϵ; number of iteration T and decay factor μ
Input: The factor β for the upper bound of neighborhood and number of example N for variance tuning
Output: An adversarial example x^{adv}

$\alpha = \epsilon/T$
$g_0 = 0$; $v_0 = 0$; $x_0^{adv} = x$
for $t = 0 \to T-1$ **do**
  Calculate the gradient $\hat{g}_{t+1} = \nabla_{x_t^{adv}} J(x_t^{adv}, y; \theta)$
  Update g_{t+1} by variance tuning based momentum

$$g_{t+1} = \mu \cdot g_t + \frac{\hat{g}_{t+1} + v_t}{\|\hat{g}_{t+1} + v_t\|_1} \qquad (5)$$

  Update $v_{t+1} = V(x_t^{adv})$ by Eq. (7)
  Update x_{t+1}^{adv} by applying the sign of gradient

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \mathrm{sign}(g_{t+1}) \qquad (6)$$

end for
$x^{adv} = x_T^{adv}$
return x^{adv}

通过数据增强的算法来提高迁移性

思路是通过input transformation来进行数据增强，利用一定的先验知识来找到更好的对抗样本。

相关的方法有

Diverse Input Method (DIM) Random resizing and padding

Translation-Invariant Method (TIM)

$$x_{t+1}^{adv} = x_t^{adv} + \alpha * sign(W * \nabla_x J(x_t^{adv}, y))$$

Scale-Invariant Method (SIM)

$$g_{t+1} = \frac{1}{m} \sum_{i=0}^{m-1} \nabla_{x_t^{adv}} (J(x_t^{adv}/2^i, y; \theta))$$

Admix Attack Method (Admix)

$$g_{t+1} = \frac{1}{m_1 * m_2} \sum_{x' \in X'} \sum_{i=0}^{m_1-1} \nabla_{x_t^{adv}} J(\gamma_i * (x_y^{adv} + \eta * x'), y; \theta)$$

Black-Box Adversarial Attacks with Priors

NES算法 NATURAL EVOLUTIONARY STRATEGIES

论文为Black-box Adversarial Attacks with Limited Queries and Information

在真实情况下，查询的次数是有限的，作者利用自然演化算法来估计梯度，其中采样的方法是高斯采样分布，而且是对称的，这样出来的是无偏估计，且上界下界不断逼近。

Algorithm 1 NES Gradient Estimate

Input: Classifier $P(y|x)$ for class y, image x
Output: Estimate of $\nabla P(y|x)$
Parameters: Search variance σ, number of samples n, image dimensionality N
$g \leftarrow \mathbf{0}_n$
for $i = 1$ **to** n **do**
  $u_i \leftarrow \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_{N \cdot N})$
  $g \leftarrow g + P(y|x + \sigma \cdot u_i) \cdot u_i$
  $g \leftarrow g - P(y|x - \sigma \cdot u_i) \cdot u_i$
end for
return $\frac{1}{2n\sigma} g$

Partial-Information下的NES算法

Partial-Information setting下，攻击者只能访问top k种分类的概率，甚至无法获得一个准确的softmax分布。
这时算法从目标分类直接出发，投影到x的范围内，进行自然演替，在更新样本的过程中不断调整超参数。

Algorithm 2 Partial Information Attack

Input: Initial image x, Target class y_{adv}, Classifier $P(y|x) : \mathbb{R}^n \times \mathcal{Y} \rightarrow [0,1]^k$ (access to probabilities for y **in top** k), image x

Output: Adversarial image x_{adv} with $||x_{adv} - x||_\infty \leq \epsilon$

Parameters: Perturbation bound ϵ_{adv}, starting perturbation ϵ_0, NES Parameters (σ, N, n), epsilon decay δ_ϵ, maximum learning rate η_{max}, minimum learning rate η_{min}

$\epsilon \leftarrow \epsilon_0$
$x_{adv} \leftarrow$ image of target class y_{adv}
$x_{adv} \leftarrow \text{CLIP}(x_{adv}, x - \epsilon, x + \epsilon)$
while $\epsilon > \epsilon_{adv}$ or $\max_y P(y|x) \neq y_{adv}$ **do**
  $g \leftarrow \text{NESESTGRAD}(P(y_{adv}|x_{adv}))$
  $\eta \leftarrow \eta_{max}$
  $\hat{x}_{adv} \leftarrow x_{adv} - \eta g$
  while not $y_{adv} \in \text{TOP-K}(P(\cdot|\hat{x}_{adv}))$ **do**
    if $\eta < \eta_{min}$ **then**
      $\epsilon \leftarrow \epsilon + \delta_\epsilon$
      $\delta_\epsilon \leftarrow \delta_\epsilon / 2$
      $\hat{x}_{adv} \leftarrow x_{adv}$
      break
    end if
    $\eta \leftarrow \frac{\eta}{2}$
    $\hat{x}_{adv} \leftarrow \text{CLIP}(x_{adv} - \eta g, x - \epsilon, x + \epsilon)$
  end while
  $x_{adv} \leftarrow \hat{x}_{adv}$
  $\epsilon \leftarrow \epsilon - \delta_\epsilon$
end while
return x_{adv}

Label-Only下的NES算法

更极端的情况下，我们甚至无法获得分数，只能获得一个top k分类，此时我们利用随机扰动生成对应输出概率的代理

$$R(x^{(t)}) = k - rank(y_{adv}|x^{(t)})$$

$$S(x_{(t)}) = \frac{1}{n}\sum_{i=1}^{n} R(x^{(}t), +\mu\delta_i)$$

A visual representation of this process is given in Figure 1.

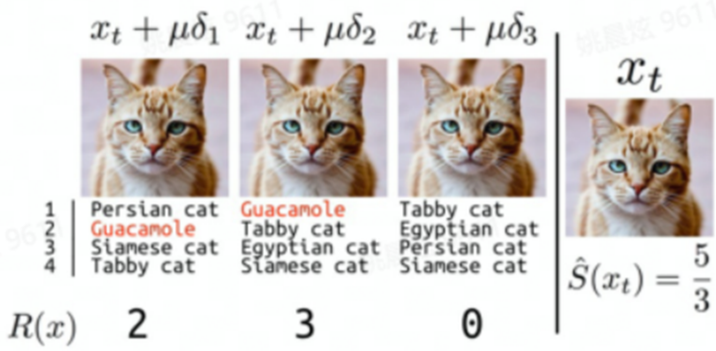


Figure 1. An illustration of the derivation of the proxy score $\hat{S}$ in the label-only setting.

Priors

论文为Prior Convictions:Black-Box Adversarial Attacks with Bandits and Priors

1. **Time-dependent Priors** 作者在实验中发现迭代的步之间梯度是高度相关的，可以将t-1步的梯度作为t步梯度的先验。

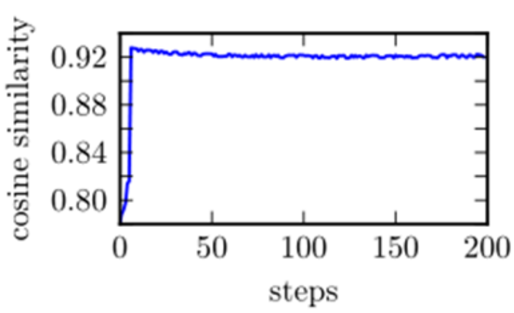


Figure 2: Cosine similarity between the gradients at the current and previous steps along the optimization trajectory of NES PGD attacks, averaged over 1000 random ImageNet images.

2. **Data-dependent Priors** 在图像分类的情况下，图像往往具有空间相似性，在两个非常接近的像素点的位置，梯度是十分相近的

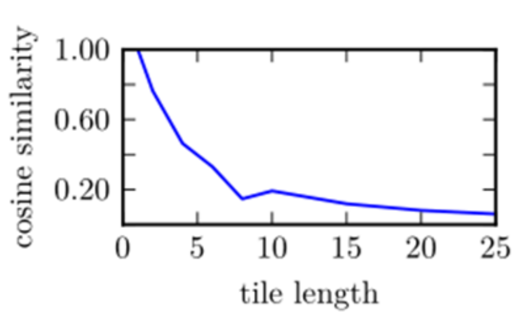


Figure 3: Cosine similarity of "tiled" image gradient with original image gradient versus the length of the square tiles, averaged over 5,000 randomly selected ImageNet images.

Algorithm 3 Adversarial Example Generation with Bandit Optimization for ℓ_2 norm perturbations

1: **procedure** ADVERSARIAL-BANDIT-L2(x_{init}, y_{init})
2: // $C(\cdot)$ returns top class
3: $v_0 \leftarrow \mathbf{0}_{1\times d}$ // If data prior, $d < \dim(x)$; v_t (Δ_t) up (down)-sampled before (after) line 8
4: $x_0 \leftarrow x_{init}$ // Adversarial image to be constructed
5: **while** $C(x) = y_{init}$ **do**
6: $g_t \leftarrow v_{t-1}$
7: $x_t \leftarrow x_{t-1} + h \cdot \frac{g_t}{||g_t||_2}$ // Boundary projection $\frac{g}{||g_t||}$ standard PGD: c.f. [Rig15]
8: $\Delta_t \leftarrow$ GRAD-EST($x_{t-1}, y_{init}, v_{t-1}$) // Estimated Gradient of ℓ_t
9: $v_t \leftarrow v_{t-1} + \eta \cdot \Delta_t$
10: $t \leftarrow t + 1$
return x_{t-1}

Referance

经典的基于梯度的黑盒攻击算法

1. Explaining and Harnessing Adversarial Examples
2. DeepFool: a simple and accurate method to fool deep neural networks
3. The Limitations of Deep Learning in Adversarial Settings
4. Towards Evaluating the Robustness of Neural Networks
5. Towards Deep Learning Models Resistant to Adversarial Attacks

提高黑盒迁移性的相关算法

1. Boosting Adversarial Attacks with Momentum
2. Nesterov accelerated gradient and scale invariance for adversarial attacks
3. Enhancing the Transferability of Adversarial Attacks through Variance Tuning
4. Enhancing the Transferability of Adversarial Attacks through Variance Tuning
5. Admix Enhancing the Transferability of Adversarial Attacks
6. Improving Transferability of Adversarial Examples With Input Diversity
7. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks

基于查询和先验知识的黑盒攻击算法

1. Natural evolution strategies
2. Black-box Adversarial Attacks with Limited Queries and Information
3. Learning Black-Box Attackers with Transferable Priors and Query Feedback
4. Prior Convictions:Black-Box Adversarial Attacks with Bandits and Priors