

Attack on Image Recognition

Attack on Image Recognition

[Todo List](#)

[Explaining and Harnessing Adversarial Examples](#)

[Notes](#)

[Links](#)

[ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training](#)

[Substitute Models](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Synthesizing Robust Adversarial Examples](#)

[Contribution](#)

[Notes](#)

[Links](#)

[NES: Black-box Adversarial Attacks with Limited Queries and Information](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Hybrid Batch Attacks: Finding Black-box Adversarial Examples with Limited Queries](#)

[Contribution](#)

[Notes](#)

[Links](#)

Todo List

1. Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. 2016.
2. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. 2013.
3. Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In IEEE Symposium on Security & Privacy, 2017c.
4. Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., and Song, D. Robust PhysicalWorld Attacks on Deep Learning Models. 2017.
5. Tom B Brown, Dandelion Man' e, Aurko Roy, Mart' in Abadi, and Justin Gilmer. Adversarial patch. arXiv preprint arXiv:1712.09665, 2017.
6. Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. arXiv preprint arXiv:1801.02608, 2018.
7. Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. arXiv preprint arXiv:1910.14667, 2019.
8. Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks, 2019
9. Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Adversarial generative nets: Neural network attacks on state-of-the-art face recognition. arXiv preprint arXiv:1801.00349, 2017.
10. Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint

arXiv:1802.00420, 2018.

11. advbox
12. paddlepaddle 对抗工具箱
13. Stealthy Porn: Understanding Real-World Adversarial Images for Illicit Online Promotion
14. Stealing Hyperparameters in Machine Learning
15. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks
16. Text Captcha Is Dead? A Large Scale Deployment and Empirical Study
17. A Tale of Evil Twins: Adversarial Inputs versus Poisoned Models
18. Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving
19. Privacy Risks of Securing Machine Learning Models against Adversarial Examples
20. Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Convolutional Networks
21. Seeing isn't Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors
22. Model-Reuse Attacks on Learning Systems
23. A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in ICML, 2018.
24. A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in ICLR, 2017.
25. Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, Kai Chen, "Seeing isn't Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors", CCS 2019

Explaining and Harnessing Adversarial Examples

Notes

1. 线性解释：作者提出了对抗样本存在的**线性解释**。如果将神经网络泛化成如下所示的**线性**点积形式：

$$\mathbf{w}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + \mathbf{w}^\top \boldsymbol{\eta}$$

其中 $\tilde{\mathbf{x}}$ 为目标类的一个样本， \mathbf{x} 为原始样本， $\boldsymbol{\eta}$ 为添加的对抗扰动。当输入的维度无限扩大时，很显然我们可以保证 $\|\boldsymbol{\eta}\|_\infty$ 很小的情况下，而点积后的值却变化很大，使得上式两侧的值相等，即生成了一个成功的对抗扰动；

2. 对抗样本生成算法：作者提出了基于对抗样本线性解释的快速对抗样本生成算法，**Fast Gradient Sign Method (FGSM)** 生成对抗样本只需要计算一次梯度，然后在梯度上走一小步：

$$\boldsymbol{\eta} = \epsilon \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$$

3. 对抗训练：作者提出了在深度神经网络中可以通过对抗训练的方法来防御对抗攻击。下面利用 FGSM 进行对抗训练：

$$\tilde{J}(\boldsymbol{\theta}, \mathbf{x}, y) = \alpha J(\boldsymbol{\theta}, \mathbf{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y)))$$

公式的含义为，在训练网络的过程中，不仅要保证现有样本能够被成功分类，对于那些添加了一小步的对抗样本应该同样被正确分类；

4. 泛化原因：作者解释**对抗样本的存在位置并不是一个特定的点，而是一个区域，或称为子空间**。由于子空间的这个属性，以及深度学习模型对训练集拟合的相同效果，可能出现了**对抗样本子空间的重叠**，使得对抗样本可以在不同的模型之间进行迁移；

Links

- 论文链接: [Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples\[J\]. arXiv preprint arXiv:1412.6572, 2014.](#)
- 论文代码: https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/papers/ma_xout
- 前人笔记: <https://zhuanlan.zhihu.com/p/32784766>

ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models

Contribution

1. 提出了第一个基于梯度估计的黑盒优化攻击算法;
2. 针对梯度估计过程中需要大量访问黑盒模型的问题, 提出了 3 中可行的缓解方法 (访问次数依然很大);

Notes

1. 作者提出了一种**针对黑盒的优化攻击 (Optimization Attack) 算法**;
2. Introduction: (文章比较早, 故作者用较多的篇幅介绍了对抗攻击领域的工作)
 - (1) 白盒攻击算法: FGSM (Fast Gradient Sign Method), JSMA (Jacobian-based Saliency Map Attack), DeepFool, Carlini & Wagner (C&W) Attack;
 - (2) 本文攻击场景: 攻击黑盒模型, 攻击者只知道输入和相应的输出;
 - (3) 对抗攻击防御: Detection-based Defense, Gradient and Representation Masking, Adversarial training;

企业更加看重黑盒攻击和防御, 所以如果以工作为目标的同学, 需要多学习、思考这方面的内容

3. ZOO 攻击算法:

- (1) 借鉴 C&W Attack, 将生成对抗样本的过程转换成一个最优化问题:

$$\begin{aligned} & \text{minimize}_{\mathbf{x}} \|\mathbf{x} - \mathbf{x}_0\|_2^2 + c \cdot f(\mathbf{x}, t) \\ & \text{subject to } \mathbf{x} \in [0, 1]^p, \end{aligned}$$

其中 $f(x, t)$ 为损失函数;

- (2) 损失函数:

- 有目标攻击的损失函数如下:

$$f(\mathbf{x}, t) = \max\{\max_{i \neq t} \log[F(\mathbf{x})]_i - \log[F(\mathbf{x})]_t, -\kappa\}$$

- 无目标攻击的损失函数如下:

$$f(\mathbf{x}) = \max\{\log[F(\mathbf{x})]_{t_0} - \max_{i \neq t_0} \log[F(\mathbf{x})]_i, -\kappa\}$$

- (3) 零阶优化:

- 一阶导数估计:

$$\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}$$

◦ 二阶导数估计:

$$\hat{h}_i := \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_{ii}^2} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - 2f(\mathbf{x}) + f(\mathbf{x} - h\mathbf{e}_i)}{h^2}$$

其中 h 为一个极小的固定值, 文章中作者取 0.0001, \mathbf{e}_i 为只有第 i 个值为 1 的矩阵。如果输入的矩阵 (图像) 含有 p 个像素点的话, 那么通过作者的方法需要访问模型 $2p$ 次。

(4) Stochastic Coordinate Descent: (直译过来为“随机坐标下降”) 随机从输入中挑选一个点, 使用梯度下降算法进行修改;

◦ Stochastic Coordinate Descent:

Algorithm 1 Stochastic Coordinate Descent

```

1: while not converged do
2:   Randomly pick a coordinate  $i \in \{1, \dots, p\}$ 
3:   Compute an update  $\delta^*$  by approximately minimizing
        
$$\arg \min_{\delta} f(\mathbf{x} + \delta \mathbf{e}_i)$$

4:   Update  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$ 
5: end while

```

◦ ZOO-ADAM:

Algorithm 2 ZOO-ADAM: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise ADAM

Require: Step size η , ADAM states $M \in \mathbb{R}^p, v \in \mathbb{R}^p, T \in \mathbb{Z}^p$, ADAM hyper-parameters $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

```

1:  $M \leftarrow \mathbf{0}, v \leftarrow \mathbf{0}, T \leftarrow \mathbf{0}$ 
2: while not converged do
3:   Randomly pick a coordinate  $i \in \{1, \dots, p\}$ 
4:   Estimate  $\hat{g}_i$  using (6)
5:    $T_i \leftarrow T_i + 1$ 
6:    $M_i \leftarrow \beta_1 M_i + (1 - \beta_1) \hat{g}_i, \quad v_i \leftarrow \beta_2 v_i + (1 - \beta_2) \hat{g}_i^2$ 
7:    $\hat{M}_i = M_i / (1 - \beta_1^{T_i}), \quad \hat{v}_i = v_i / (1 - \beta_2^{T_i})$ 
8:    $\delta^* = -\eta \frac{\hat{M}_i}{\sqrt{\hat{v}_i + \epsilon}}$ 
9:   Update  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$ 
10: end while

```

◦ ZOO-Newton:

Algorithm 3 ZOO-Newton: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise Newton's Method

Require: Step size η

```

1: while not converged do
2:   Randomly pick a coordinate  $i \in \{1, \dots, p\}$ 
3:   Estimate  $\hat{g}_i$  and  $\hat{h}_i$  using (6) and (7)
4:   if  $\hat{h}_i \leq 0$  then
5:      $\delta^* \leftarrow -\eta \hat{g}_i$ 
6:   else
7:      $\delta^* \leftarrow -\eta \frac{\hat{g}_i}{\hat{h}_i}$ 
8:   end if
9:   Update  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$ 
10: end while
  
```

作者实验中发现，ADAM 比 Newton 生成对抗样本来得更快；

(5) 缩小迭代空间：为了减少 ZOO 的 Query 数量，从而加快算法的运行。大致的思想是进行**对抗扰动特征空间的映射**，定义一个（更小的）扰动特征空间 \mathbb{R}^p 和特征映射函数 $D(\cdot)$ ，那么转换（原特征空间-像素空间的）最优化问题为扰动特征空间的最优化问题：

$$\begin{aligned} & \text{minimize}_y \|D(\mathbf{y})\|_2^2 + c \cdot f(\mathbf{x}_0 + D(\mathbf{y}), t) \\ & \text{subject to } \mathbf{x}_0 + D(\mathbf{y}) \in [0, 1]^P. \end{aligned}$$

其中 y 表示在扰动特征空间的对抗扰动；作者提到的特征映射方法有 Up-Sampling（升采样）和 DCT（时频变换）；

(6) 分层递进攻击：前一个方法可以大大减小对抗样本的搜索空间，但是由于搜索空间的受限，会导致无法生成成功的对抗样本的问题。大致的思想是定义**多个**对抗扰动特征空间的映射 $D_1(\cdot), D_2(\cdot), \dots$ ，攻击过程中首先使用 D_1 生成对抗样本，如果在一定轮数后仍未生成成功的对抗样本，那么将最后一轮的样本转换到 D_2 的特征空间（**后面使用的特征空间应保证比前面的特征空间更广**），继续生成对抗样本。

(7) 重要像素点优先迭代：作者虽然缩小了查询的特征空间（ $32 \times 32 \times 3$ for example），但是在这个空间中生成对抗样本还是需要花费大量的 Query 次数，并且不一定能够生成成功的对抗样本。大致的思想是**将图像切块，分块定义像素点被随机采样的概率，概率的大小和区域中像素值的变化大小成正相关**。作者给出了大致的采样概率变化示意图：

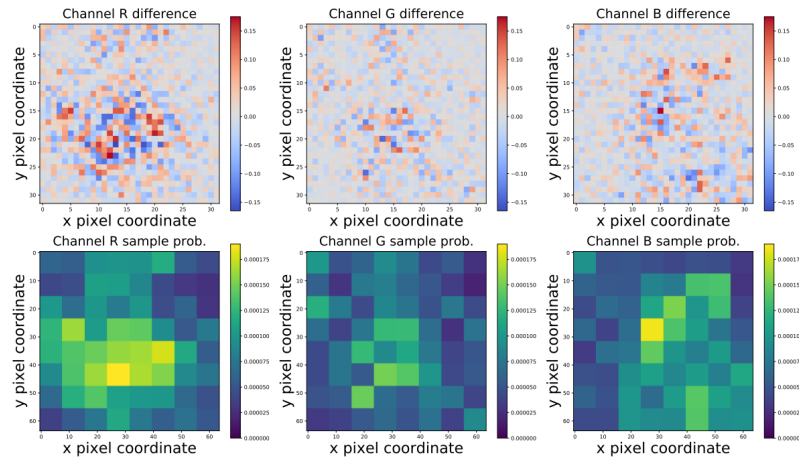


Figure 3: Attacking the bagel image in Figure 1 (a) with importance sampling. Top: Pixel values in certain parts of the bagel image have significant changes in RGB channels, and the changes in the R channel is more prominent than other channels. Here the attack-space is $32 \times 32 \times 3$. Although our targeted attack in this attack-space fails, its adversarial noise provides important clues to pixel importance. We use the noise from this attack-space to sample important pixels after we increase the dimension of attack-space to a larger dimension. Bottom: Importance sampling probability distribution for $64 \times 64 \times 3$ attack-space. The importance is computed by taking the absolute value of pixel value changes, running a 4×4 max-pooling for each channel, up-sampling to the dimension of $64 \times 64 \times 3$, and normalizing all values.

作者指出在小的扰动空间时，并不采用这种优先采样算法；

4. Evaluation 1:

(1) 实验 1 的目标：这是第一个在黑盒模型上做的优化攻击，所以作者的目标是和已有的白盒攻击 (C&W Attack) 和迁移攻击 (CleverHans) 做对比，希望能够达到这样的效果：

- 攻击的成功率和添加的对抗扰动大小能够和白盒攻击算法相近；
- 攻击的成功率应该远优于迁移攻击；

(2) 黑盒模型：

Layer Type	MNIST Model	CIFAR Model
Convolution + ReLU	$3 \times 3 \times 32$	$3 \times 3 \times 64$
Convolution + ReLU	$3 \times 3 \times 32$	$3 \times 3 \times 64$
Max Pooling	2×2	2×2
Convolution + ReLU	$3 \times 3 \times 64$	$3 \times 3 \times 128$
Convolution + ReLU	$3 \times 3 \times 64$	$3 \times 3 \times 128$
Max Pooling	2×2	2×2
Fully Connected + ReLU	200	256
Fully Connected + ReLU	200	256
Softmax	10	10

TABLE I
MODEL ARCHITECTURES FOR THE MNIST AND CIFAR MODELS. THIS ARCHITECTURE IS IDENTICAL TO THAT OF THE ORIGINAL DEFENSIVE DISTILLATION WORK. [39]

Parameter	MNIST Model	CIFAR Model
Learning Rate	0.1	0.01 (decay 0.5)
Momentum	0.9	0.9 (decay 0.5)
Delay Rate	-	10 epochs
Dropout	0.5	0.5
Batch Size	128	128
Epochs	50	50

TABLE II
MODEL PARAMETERS FOR THE MNIST AND CIFAR MODELS. THESE PARAMETERS ARE IDENTICAL TO THAT OF THE ORIGINAL DEFENSIVE DISTILLATION WORK. [39]

(3) 样本数量：

- 有目标攻击，生成 900 个对抗样本；（其他细节见原文）
- 无目标攻击，生成 200 个对抗样本；

(4) 实验结果：

Table 1: MNIST and CIFAR10 attack comparison: ZOO attains comparable success rate and L_2 distortion as the white-box C&W attack, and significantly outperforms the black-box substitute model attacks using FGSM (L_∞ attack) and the C&W attack [35]. The numbers in parentheses in Avg. Time field is the total time for training the substitute model. For FGSM we do not compare its L_2 with other methods because it is an L_∞ attack.

	MNIST					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	1.48066	0.48 min	100 %	2.00661	0.53 min
Black-box (Substitute Model + FGSM)	40.6 %	-	0.002 sec (+ 6.16 min)	7.48 %	-	0.002 sec (+ 6.16 min)
Black-box (Substitute Model + C&W)	33.3 %	3.6111	0.76 min (+ 6.16 min)	26.74 %	5.272	0.80 min (+ 6.16 min)
Proposed black-box (ZOO-ADAM)	100 %	1.49550	1.38 min	98.9 %	1.987068	1.62 min
Proposed black-box (ZOO-Newton)	100 %	1.51502	2.75 min	98.9 %	2.057264	2.06 min
	CIFAR10					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	0.17980	0.20 min	100 %	0.37974	0.16 min
Black-box (Substitute Model + FGSM)	76.1 %	-	0.005 sec (+ 7.81 min)	11.48 %	-	0.005 sec (+ 7.81 min)
Black-box (Substitute Model + C&W)	25.3 %	2.9708	0.47 min (+ 7.81 min)	5.3 %	5.7439	0.49 min (+ 7.81 min)
Proposed Black-box (ZOO-ADAM)	100 %	0.19973	3.43 min	96.8 %	0.39879	3.95 min
Proposed Black-box (ZOO-Newton)	100 %	0.23554	4.41 min	97.0 %	0.54226	4.40 min

5. Evaluation 2:

- (1) 实验 2 目标：作者尝试将这种攻击运用在更大的模型上，并且探讨文章提出的缓解方法的作用；
- (2) 黑盒模型：Inception-V3；
- (3) 实验设定：
- 无目标攻击：
生成150张对抗样本；保证每张对抗样本的大小都大于 $299 * 299$ ；不使用分层递进方法，只使用一个 $32 * 32 * 3$ 的对抗扰动域进行攻击；限制算法的迭代轮数为 1500 轮（ $1500 * 128$ 次 Query）；
 - 有目标攻击：
只选择了一张在无目标攻击中无法成功攻击的样本；扩大对抗扰动域为 $64 * 64 * 3$ 和 $128 * 128 * 3$ ；最大迭代轮数上升为 20000 轮（ $20000 * 128$ 次 Query）；（可以看到，黑盒下面的 Query 数量是十分惊人的）
- (4) 实验结果：
- 无目标攻击：

Table 2: Untargeted ImageNet attacks comparison. Substitute model based attack cannot easily scale to ImageNet.

	Success Rate	Avg. L_2
White-box (C&W)	100 %	0.37310
Proposed black-box (ZOO-ADAM)	88.9 %	1.19916
Black-box (Substitute Model)	N.A.	N.A.

- 有目标攻击：

Table 3: Comparison of different attack techniques. “First Valid” indicates the iteration number where the first successful attack was found during the optimization process.

Black-box (ZOO-ADAM)	Success?	First Valid	Final L_2	Final Loss
All techniques	Yes	15,227	3.425	11.735
No Hierarchical Attack	No	-	-	62.439
No importance sampling	Yes	17,403	3.63486	13.216
No ADAM state reset	Yes	15,227	3.47935	12.111

Links

- 论文链接： [Chen P Y, Zhang H, Sharma Y, et al. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models\[C\]//Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. 2017: 15-26.](#)
- 论文代码： <https://github.com/IBM/ZOO-Attack>
- C&W Attack 代码： https://github.com/carlini/nn_robust_attacks
- CleverHans： <https://github.com/cleverhans-lab/cleverhans>

Synthesizing Robust Adversarial Examples

Contribution

1. 提出了一种增加物理环境下对抗样本鲁棒性的一般化方法 EOT;
2. 不仅在 2D 下测试, 而且在 3D 下测试;
3. 模拟物理变换的想法十分具有借鉴意义, 已被后续的对抗攻击算法广泛使用;

Notes

1. **白盒的、针对物理环境下的、有目标**的对抗攻击算法。攻击的算法不仅在 2D 下可行, 同时在 **3D** 下也可以生成成功的对抗样本;
2. 已有的对抗攻击算法, 训练的目标如下:

$$\begin{aligned} \arg \max_{x'} \quad & \log P(y_t | x') \\ \text{subject to} \quad & \|x' - x\|_p < \epsilon \\ & x' \in [0, 1]^d \end{aligned}$$

但是这样生成的对抗样本, 在视角等物理环境发生改变时**无法保持对抗性**。故作者提出改进后的训练目标 **EOT (Expectation Over Transformation)**:

$$\begin{aligned} \arg \max_{x'} \quad & \mathbb{E}_{t \sim T} [\log P(y_t | t(x'))] \\ \text{subject to} \quad & \mathbb{E}_{t \sim T} [d(t(x'), t(x))] < \epsilon \\ & x \in [0, 1]^d \end{aligned}$$

其含义是, 在保证对抗样本经过物理变换的“感受”修改量在一定范围内时, 使得对抗样本**(经过物理变换)**能够尽可能地被分类为目标类别。这类物理变换可以是 2D/3D 的变换, 包括随机旋转、平移、噪声、视角变化、光照等。作者将公式转换为 [Carlini & Wagner \(2017c\)](#) 的形式, 并使用**二级范数**和 **PGD** (Projected Gradient Descent) 优化器进行计算:

$$\begin{aligned} \arg \max_{x'} \quad & \mathbb{E}_{t \sim T} [\log P(y_t | t(x')) \\ & - \lambda \|LAB(t(x')) - LAB(t(x))\|_2] \end{aligned}$$

其中 *LAB* 代表指的是 [LAB 色域](#)。

3. Distributions of Transformations:

(1) 2D Case

Transformation	Minimum	Maximum
Scale	0.9	1.4
Rotation	-22.5°	22.5°
Lighten / Darken	-0.05	0.05
Gaussian Noise (stdev)	0.0	0.1
Translation	any in-bounds	

(2) 3D Case

Transformation	Minimum	Maximum
Camera distance	2.5	3.0
X/Y translation	-0.05	0.05
Rotation	any	
Background	(0.1, 0.1, 0.1)	(1.0, 1.0, 1.0)

(3) Physical Case

Transformation	Minimum	Maximum
Camera distance	2.5	3.0
X/Y translation	-0.05	0.05
Rotation	any	
Background	(0.1, 0.1, 0.1)	(1.0, 1.0, 1.0)
Lighten / Darken (additive)	-0.15	0.15
Lighten / Darken (multiplicative)	0.5	2.0
Per-channel (additive)	-0.15	0.15
Per-channel (multiplicative)	0.7	1.3
Gaussian Noise (stdev)	0.0	0.1

4. Evaluation:

(1) 攻击基于数据集 ImageNet 的 **Inception V3** 模型 (Top-1 Accuracy = 78.0%), 随机选择目标分类;

(2) **Robust 2D adversarial examples**: 在 2D 下考虑的物理变换有 **缩放、旋转、亮度调节、高斯噪声和平移**。每个样本都在 **1000** 个随机的模拟物理变换上进行测试, 结果如下:

Images	Classification Accuracy		Adversariality		ℓ_2
	mean	stdev	mean	stdev	mean
Original	70.0%	36.4%	0.01%	0.3%	0
Adversarial	0.9%	2.0%	96.4%	4.4%	5.6×10^{-5}

(3) **Robust 3D adversarial examples**: 在 3D 下考虑不同的相机距离、照明条件、对象的平移和旋转以及纯色背景色。挑选了 10 个 3D 模型 —— 木桶、棒球、够、橘子、海龟、小丑鱼、沙发、泰迪熊、汽车和出租车。每个 3D 模型都挑选 20 个随机的目标分类标签; 每个样本都在 100 个随机的模拟物理变换上进行测试, 结果如下:

Images	Classification Accuracy		Adversariality		ℓ_2
	mean	stdev	mean	stdev	mean
Original	68.8%	31.2%	0.01%	0.1%	0
Adversarial	1.1%	3.1%	83.4%	21.7%	5.9×10^{-3}

(4) **Physical adversarial examples**: 在 3D 的基础上, 考虑摄像机的噪声、照明的影响和颜色的失真。作者考虑将“海龟”错误分类成“手枪”、“棒球”错误分类成“咖啡”两种情况, 对抗样本经过 3D 打印后, 拍 100 张照片进行测试, 结果如下:

Object	Adversarial	Misclassified	Correct
Turtle	82%	16%	2%
Baseball	59%	31%	10%

(5) **Perturbation budget**: 在物理环境下越鲁棒, 需要模拟更多的物理变换, 添加的噪声也会更多;

Links

- 论文链接: [Athalye, Anish, et al. "Synthesizing robust adversarial examples." International conference on machine learning. PMLR, 2018.](#)
- 开源代码: [prabhant/synthesizing-robust-adversarial-examples \(github.com\)](#)

NES: Black-box Adversarial Attacks with Limited Queries and Information

Contribution

1. 利用 NES 算法大大减少黑盒攻击的访问次数;

Notes

1. 黑盒设定:
 - Query-limited Setting: 限制访问次数;
 - Partial-information Setting: 只知道 Top-K 的结果 (包括概率);
 - Label-only Setting: 只知道 Top-K 的标签 (不包括概率); (这一项我觉得没必要看)
2. NES (Natural Evolutionary Strategies) 进行梯度估计: 最小化期望的损失大小, 算法伪代码如下 (如何挑选这个参数?)

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**

$u_i \leftarrow \mathcal{N}(\mathbf{0}_N, \mathbf{I}_{N \cdot N})$

$g \leftarrow g + P(y|x + \sigma \cdot u_i) \cdot u_i$

$g \leftarrow g - P(y|x - \sigma \cdot u_i) \cdot u_i$

end for

return $\frac{1}{2n\sigma} g$

看不懂这个式子的话, 在草稿纸上把这两个式子列成求梯度的形式

3. PGD (Projected Gradient Descent) 进行梯度更新:

$$x^{(t)} = \Pi_{[x_0 - \epsilon, x_0 + \epsilon]}(x^{(t-1)} - \eta \cdot \text{sign}(g_t))$$

4. 仅知道 Top-K 的概率:

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**

$g \leftarrow \text{NESTGRAD}(P(y_{adv}|x_{adv}))$

$\eta \leftarrow \eta_{max}$

$\hat{x}_{adv} \leftarrow x_{adv} - \eta g$

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

$\eta \leftarrow \frac{\eta}{2}$

$\hat{x}_{adv} \leftarrow \text{CLIP}(\hat{x}_{adv} - \eta g, x - \epsilon, x + \epsilon)$

end while

$x_{adv} \leftarrow \hat{x}_{adv}$

$\epsilon \leftarrow \epsilon - \delta_\epsilon$

end while

return x_{adv}

当学习率低于最小值时仍未生成对抗样本时，则增大扰动的变化区间。

当学习率仍大于最小值时，不断减小学习率进行探测

- 使用目标分类的样本来初始化扰动，从而减少 query 的数量；
- 在保证目标分类在 Top-K 中的前提下，不断缩小对抗扰动，直至生成对抗样本且满足修改量的限制；

5. Evaluation:

(1) 参数的选择:

General	
σ for NES	0.001
n , size of each NES population	50
ϵ, l_∞ distance to the original image	0.05
η , learning rate	0.01
Partial-Information Attack	
ϵ_0 , initial distance from source image	0.5
δ_ϵ , rate at which to decay ϵ	0.001
Label-Only Attack	
m , number of samples for proxy score	50
μ, ℓ_∞ radius of sampling ball	0.001

(2) On ImageNet: 这里大概的 query 数量级为上万级别的

Threat model	Success rate	Median queries
QL	99.2%	11,550
PI	93.6%	49,624
LO	90%	2.7×10^6

Table 1. Quantitative analysis of targeted $\epsilon = 0.05$ adversarial attacks in three different threat models: query-limited (QL), partial-information (PI), and label-only (LO). We perform attacks over 1000 randomly chosen test images (100 for label-only) with randomly chosen target classes. For each attack, we use the same hyperparameters across all images. Here, we report the overall success rate (percentage of times the adversarial example was classified as the target class) and the median number of queries required.

Links

- 论文链接: [Ilyas, Andrew, et al. "Black-box adversarial attacks with limited queries and information." PRML \(2018\).](#)
- 论文代码: <https://github.com/labsix/limited-blackbox-attacks>

Hybrid Batch Attacks: Finding Black-box Adversarial Examples with Limited Queries

Contribution

Notes

1. 作者结合 迁移攻击 (Transfer Attack) 和 优化攻击 (Optimization Attack) , 并且利用种子优先级策略对模型进行黑盒攻击, 目标是在保证攻击成功率的前提下, 减少 Query 的数量;
2. 现有的优化攻击算法:

Attack	Gradient Estimation	Queries per Iteration	White-box Attack
ZOO [10]	$\hat{\mathbf{g}} = \{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_D\}, \hat{g}_i \approx \frac{f(\mathbf{x} + \delta \mathbf{e}_i) - f(\mathbf{x} - \delta \mathbf{e}_i)}{\delta}$	$2D$	CW [8]
Bhagoji et. al [4]	ZOO + random feature group or PCA	$\leq 2D$	FGSM [17], PGD [32]
AutoZOOM [43]	$\mathbf{u}_i \sim U, \hat{\mathbf{g}} = \frac{1}{N} \sum_i^N \frac{f(\mathbf{x} + \delta \mathbf{u}_i) - f(\mathbf{x})}{\delta} \mathbf{u}_i$	$N + 1$	CW [8]
NES [21]	$\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, I), \hat{\mathbf{g}} = \frac{1}{N} \sum_i^N \frac{f(\mathbf{x} + \delta \mathbf{u}_i)}{\delta} \mathbf{u}_i$	N	PGD
Bandits _{TD} [22]	NES + time/data dependent info	N	PGD
SignHunter [1]	Gradient sign w/ divide-and-conquer method	$2^{\lceil \log(D)+1 \rceil}$	PGD
Cheng et al. [13]	$\mathbf{u}_i \sim U, \hat{\mathbf{g}} = \frac{1}{N} \sum_i^N (\sqrt{\lambda} \cdot \mathbf{v} + \sqrt{1-\lambda} \cdot \frac{(\mathbf{I} - \mathbf{v}\mathbf{v}^T) \mathbf{u}_i}{\ (\mathbf{I} - \mathbf{v}\mathbf{v}^T) \mathbf{u}_i\ _2})$	N	PGD

Links

- 论文链接: [Suya F, Chi J, Evans D, et al. Hybrid batch attacks: Finding black-box adversarial examples with limited queries\[C\]//29th {USENIX} Security Symposium \(USENIX Security 2020\). 2020: 1327-1344.](#)
- 论文代码: <https://github.com/suyeecav/Hybrid-Attack>