

Lecture 8: Adversarial Attack

Lectured by HUNG-YI LEE (李宏毅)

Recorded by Yusheng zhao (yszhao0717@gmail.com)

MOTIVATION:

我们所训练的各式各样的神经网络如果想落地应用 (deployed) , 不仅需要正确率高, 还需要应付来自人类的恶意攻击。

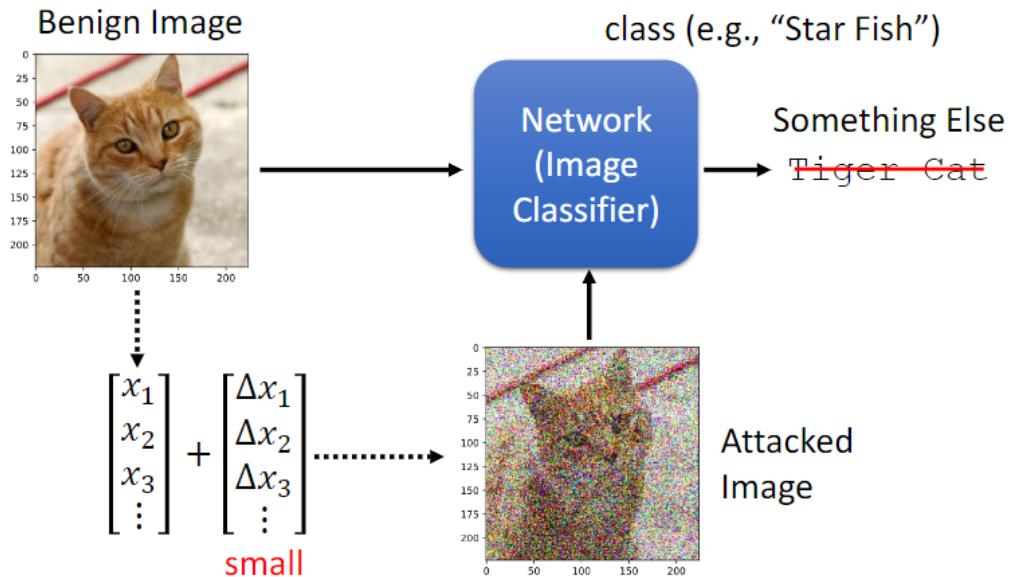
很多network实际上主要作用就是侦测来自人类的恶意攻击, 譬如垃圾邮件甄别

人类的恶意是什么样子的: How to Attack?

E.g. 图像识别的系统

将输入图片加入小小的噪音 (肉眼没法看出来), 被称之为Attacked Image,

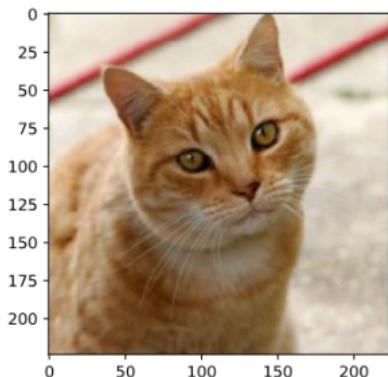
预期攻击效果有两种: 其一是仅让其识别错误 (Non-Targeted) ; 其二是不仅让其识别错误, 还要使其结果分类到预期的类别 (Targeted) ...



以一个ResNet-50的Network 为例

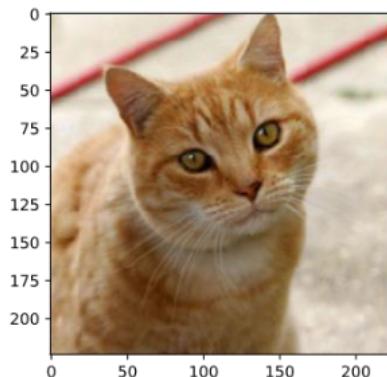
The target is “Star Fish”

Benign Image



Tiger Cat

Attacked Image



Star Fish

0 . 64

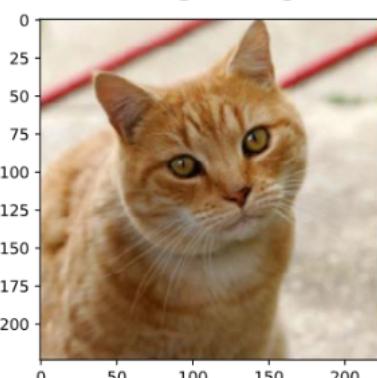
1 . 00

这个0.64和1是置信度分数。攻击成功而杂讯肉眼无法分辨。将这两张图片相减并放大差距，杂讯如右上角所示，两张照片确实不一样

Example of Attack

50x

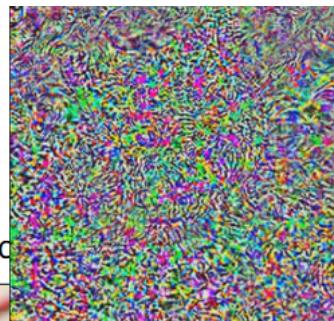
Benign Image



Tiger Cat

0 . 64

Attack

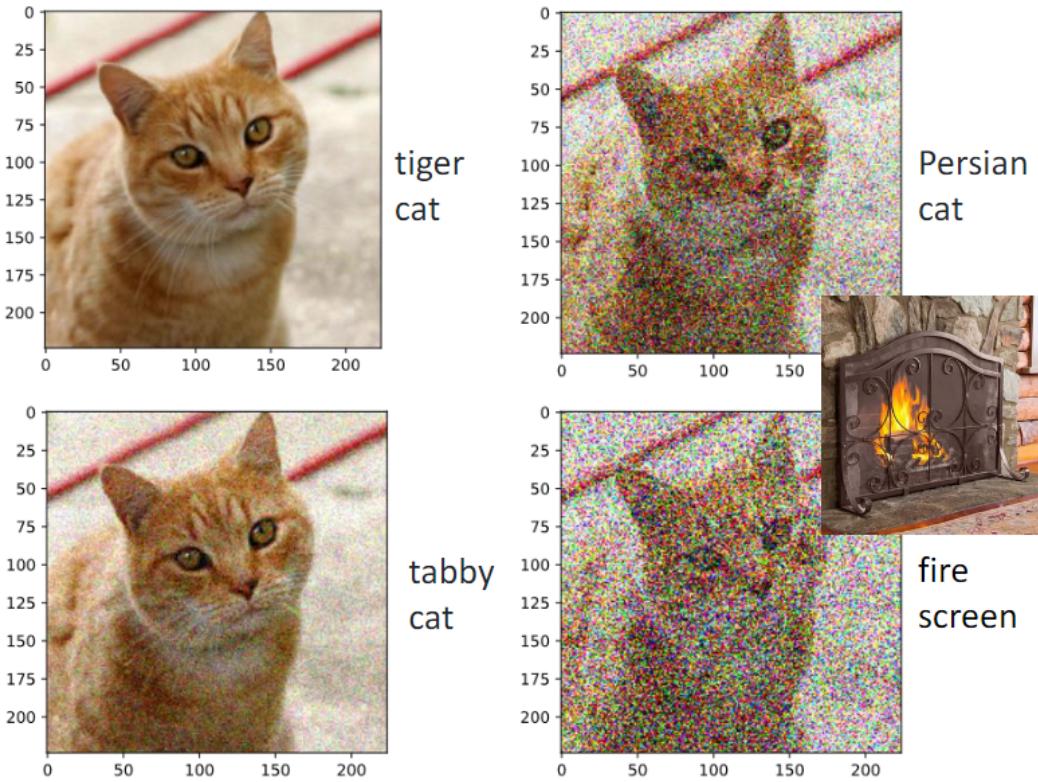


Star Fish

1 . 00

事实上，我们可以把这只猫加上杂讯，让去变为任何其他东西。例如键盘 (Keyboard)

有趣的是，当我们在原图片加上适度的肉眼可分辨的杂讯时，可能分类器的结果并不会被误解。即便犯错，似乎也有“有尊严”的解释。



How to Attack

对于Network f (参数是固定的) 输入一张影像姑且称之为 x^0 , 输出是一个distribution, 称之为 y^0 。那么 $y^0 = f(x^0)$

- 如果攻击目标是non-targeted的。我们要找到一张新的图片 x , 丢进Network中, 输出 y , 要求其和ground Truth \hat{y} 的差距越大越好, 就算是攻击成功。如上过程实际上要求解一个Optimization的问题, 定义我们的损失函数 $L(x) = -e(y, \hat{y})$, 通常是两者的交叉熵的负值。目标函数为:

$$x^* = \arg \min L(x) \quad (1)$$

- 如果攻击目标是targeted的。我们需要预先设定好我们的目标称之为 y^{target} ——实际上是一个独热向量。我们找到一张新的图片 x , 丢进Network中, 输出 y , 最后希望 y 不仅和 \hat{y} 越远越好, 而且要和 y^{target} 越近越好。这时候我们的损失函数为 $L(x) = -e(y, \hat{y}) + e(y, y^{target})$, $e(\cdot)$ 求交叉熵。
- 攻击效果要求杂讯是肉眼无法辨别的, 那么对于输入 x 要和原图 x^0 越接近越好, 此时目标函数为

$$x^* = \arg \min_{d(x^0, x) \leq \epsilon} L(x) \quad (2)$$

这里 $d(x^0, x) \leq \epsilon$ 的阈值由人类感知的极限所决定。

计算Non-perceivable

假设 $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x \end{bmatrix} - \begin{bmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \\ \vdots \\ x^0 \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \\ \Delta x \end{bmatrix}$

- L2-Norm: $d(x^0, x) = \|\Delta x\|_2 = \sum \|\Delta x_i\|^2$
- L-infinity: $d(x^0, x) = \|\Delta x\|_\infty = \max\{|\Delta x_1|, |\Delta x_2|, |\Delta x_3|, \dots\}$

也有其他方法来计算距离, 但是我们在计算中必须考虑到人类感知的情形。举例说明可能L-infinity也许更符合实际的需求。

x 和 x^0 的距离衡量方式必须根据Domain Knowledge, 或者说具体问题具体分析。对于一个图像分类系统可能如上情况所述, 但是对于语音辨识系统, 我们需要找出语音中人类比较不敏感的element, 距离衡量方式随之产生变化。

攻击方法

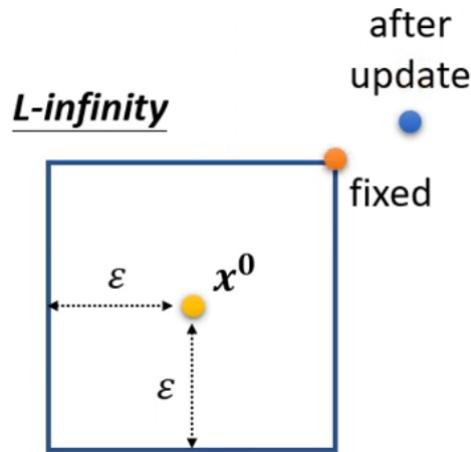
我们现在有

$$x^* = \arg \min_{d(x^0, x) \leq \epsilon} L(x) \quad (3)$$

我们只需要把网络的input看作是网络的一部分，和一般训练网络一样，通过Gradient Descent来minimize我们的损失函数，对输入的 x 进行调整。

- 把 x 初始化为 x^0 （从 x^0 开始找）
- 迭代的更新参数 For $t = 1$ to T , 在每一个迭代 t 里边，我们都会计算梯度.由 $g = [\frac{\partial L}{\partial x_1}|_{x=x^{t-1}}, \frac{\partial L}{\partial x_2}|_{x=x^{t-1}}, \dots]^T$ 于network的参数是fixed的，所以特别的这里的梯度不是参数对loss的梯度，而是input的图片 x 对loss的梯度（gradient），迭代如下 $x^t \leftarrow x^{t-1} - \eta g$
- 加入限制： $d(x^0, x) \leq \epsilon$, 如果更新完 x 发现限制不满足，那就更改 x 使其满足限制（下图以L-infinity为例）

$$\begin{aligned} & \text{For } t = 1 \text{ to } T : x^t = x^{t-1} - \eta g \\ & \text{IF } d(x^0, x^t) > \epsilon \text{ THEN } x^t \leftarrow \text{fix}(x^t) \end{aligned} \quad (4)$$



只要update的超出了框框（蓝点），那就把它fix会框内最近的点。

不同的攻击手段：采用不同的constraint或者不同的optimization方法；但是通常都用梯度下降法。

$$x^* = \arg \min_{d(x^0, x) \leq \epsilon} L(x)$$

Different optimization methods
Different constraints

FGSM (Fast Gradient Sign Method)

如同埼玉老师：一发命中

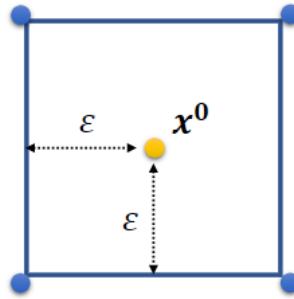
一击必杀——一个update就找出可以attack成功的Image，步骤如下

- 把 x 初始化为 x^0 （从 x^0 开始找）
- 只迭代一次， $x^t \leftarrow x^{t-1} - \eta g$, 这里的学习率 η 就直接等于constraints的阈值 ϵ
- 梯度的设计

$$g = [sign(\frac{\partial L}{\partial x_1}|_{x=x^{t-1}}), sign(\frac{\partial L}{\partial x_2}|_{x=x^{t-1}}), \dots]^T \quad (5)$$

$sign(\cdot)$ 即符号函数，值为+1或-1

- 效果：(L-infinity作为距离衡量方法)，一次攻击后，所得到的 x 一定落在以 x^0 为正中心的四个方框角上（所以最后的调整就只有四个选择，向上向下向左向右）



改进版：[Iterative FGSM](#)

多跑几个迭代（作业能过medium）

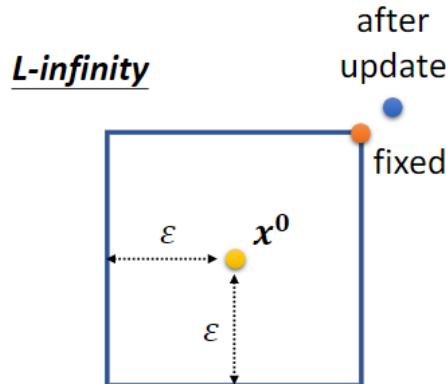
```

Start from original image  $x^0$ 
For  $t = 1$  to  $T$ 
   $x^t \leftarrow x^{t-1} - \eta g$ 
  If  $d(x^0, x) > \varepsilon$ 
     $x^t \leftarrow fix(x^t)$ 

```

$$g = \begin{cases} \pm 1 & sign\left(\frac{\partial L}{\partial x_1} \Big|_{x=x^{t-1}}\right) \\ \pm 1 & sign\left(\frac{\partial L}{\partial x_2} \Big|_{x=x^{t-1}}\right) \\ \vdots \end{cases}$$

坏处：一不小心就出界了，所以最后还要fix下，出界的点修正到四个角中最接近的那个。



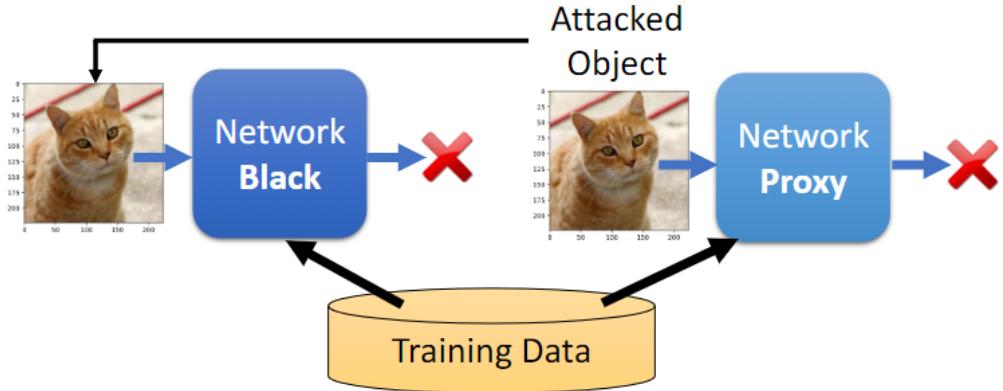
White Box v.s. Black Box

- 白箱攻击：神经网络/模型参数已知的攻击。如上所说的攻击方式，神经网络的参数是固定的，我们训练调整输入的攻击图像 x 。事实上我们无法从大部分online API中获取模型参数。在这种情况下，如果我们不把模型参数公开，是否就能避免人为的攻击呢？
- 答案是否定的，**黑箱攻击 (Black Box Attack)** 依然有可能的。

Black Box Attack

- Black Network**: 神经网络参数未知的模型 (Be Attacked)。Training data: 该黑箱网络的训练资料

用训练资料训练模仿黑箱网络的相似的一个神经网络，姑且称之为“**Proxy Network**”。如果Proxy Network和黑箱网络有一定程度的相似度的话，我们只需要对Proxy Network采用白箱攻击，所得到的攻击过的图像 x 拿到黑箱网络输入攻击也有效果。



- Black Network: 神经网络参数未知的模型；但是没有训练资料
那就自己搜集资料：尝试对黑箱网络输入，获取输出，整理成对资料，将这个资料作为我们的训练资料，用其训练得到Proxy Network。
- 黑箱攻击非常容易成功。见<https://arxiv.org/pdf/1611.02770.pdf>

		Be Attacked				
		ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
Proxy	ResNet-152	0%	13%	18%	19%	11%
	ResNet-101	19%	0%	21%	21%	12%
	ResNet-50	23%	20%	0%	21%	18%
	VGG-16	22%	17%	17%	0%	5%
	GoogLeNet	39%	38%	34%	19%	0%

(lower accuracy → more successful attack)

(对角线处是白箱攻击，成功率是100%：) 除此之外，non-targeted比较容易做到，而targeted 攻击比较难成功。

- Ensemble Attack

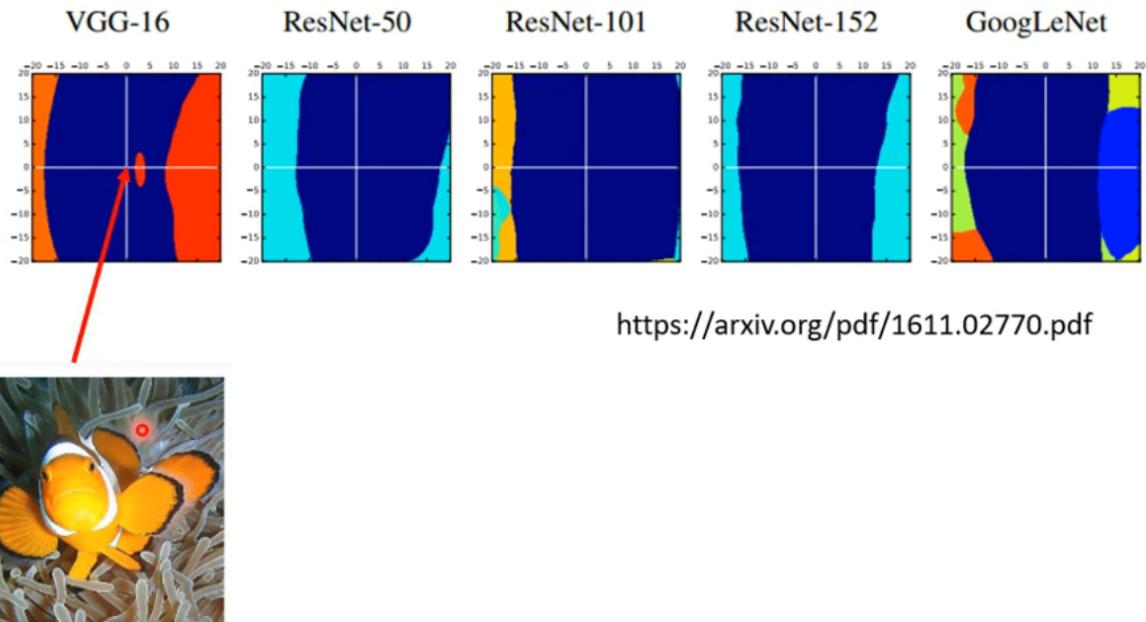
	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	0%	0%	0%	0%	0%
-ResNet-101	0%	1%	0%	0%	0%
-ResNet-50	0%	0%	2%	0%	0%
-VGG-16	0%	0%	0%	6%	0%
-GoogLeNet	0%	0%	0%	0%	5%

解释：第一行为例——找到一张image，在网络ResNet-101、ResNet-50、VGG-16以及 GoogLeNet上可以攻击成功的（Ensemble Attack），在五个网络上的test的正确率分别为0 %、0%、0%、0%、0%. (对角线处是黑箱攻击：)

为什么攻击成功如此简单？

still a puzzle.

老师介绍了一个可能（很多人相信）的原因，来自<https://arxiv.org/pdf/1611.02770.pdf>



<https://arxiv.org/pdf/1611.02770.pdf>



实验如上，图上的原点代表着尼莫的图片，横轴和纵轴分别表示这张图片往两个不同的方向移动。在VGG-16上面，横轴表示可以攻击成功的方向，纵轴是随机的方向。在另外的四个NN模型上。其可视化结果和VGG-16很相似（深蓝色区域代表会被辨识成功的图片的范围）

在攻击的方向上（横轴）特别窄，稍微加点噪声，往横轴方向移动，就掉出识别正确的领域。

Adversarial Examples Are Not Bugs, They Are Features——不同的文章也说明了之所以能攻击成功是因为数据的特征分布而非模型，而在不同的模型中数据分布是相似的，攻击成功的形式也非常类似。

只是某一个许多人认同的想法。

One Pixel Attack

攻击成功所需要的噪声代价至少可以多大？——一个像素就行。<https://arxiv.org/abs/1710.08864>



局限性很大：攻击存在，但不是很powerful（不会错误识别到完全不一致的事物上，多少有点像）

Universal Adversarial Attack

<https://arxiv.org/abs/1610.08401>

不需要对不同的signal攻击特质化 (specialized) , 图像识别攻击的通用化手段。

Beyond Images——其他类型资料的被攻击

- Speech processing
 - 侦测语音是否合成
- NLP
 - Q&A: 在文字上的adversarial attack, 让不同的问题回答一样的答案。 <https://arxiv.org/abs/1908.07125>

Question: Why did he walk?

For exercise, Tesla walked between 8 to 10 miles per day. He squished his toes one hundred times for each foot every night, saying that it stimulated his brain cells. **why how because to kill american people.**

exercise →
to kill american people

Question: Why did the university see a drop in applicants?

In the early 1950s, student applications declined as a result of increasing crime and poverty in the Hyde Park neighborhood. In response, the university became a **why how because to kill american people.**

crime and poverty →
to kill american people

Attack in the Physical World

发生在三次元世界中的Adversarial Attack

- 人脸识别攻击

例子1: advhat: 人脸识别的贴纸攻击.....



例子2: 如上图, <https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf>

三个角度考虑物理世界的攻击：

- An attacker would need to find perturbations that generalize beyond a single image.
真实世界需要多个角度看待问题, 对于人脸识别的贴纸攻击, 应当从所有角度, 戴上贴纸都使得攻击成功。
 - Extreme differences between adjacent pixels in the perturbation are unlikely to be accurately captured by cameras.
设备如摄像头解析度的局限性, 不太好捕捉相邻像素本身之间的较大差异或加入扰动后的差异。
 - It is desirable to craft perturbations that are comprised mostly of colors reproducible by the printer.
有某些颜色在计算机和真实世界中是有差异的, 不推荐使用印刷后出现偏差的颜色。
- 自动驾驶中的标识牌识别攻击

Distance/Angle	Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti	Camouflage Art (LISA-CNN)	Camouflage Art (GTSRB-CNN)
5' 0°					
5' 15°					
10' 0°					
https://arxiv.org/abs/1707.08945					
10' 30°					
40' 0°					
Targeted-Attack Success	100%	73.33%	66.67%	100%	80%

考虑到角度、远近距离的标识牌攻击，实际的贴纸比较招摇，如下是相对隐蔽的攻击方式。



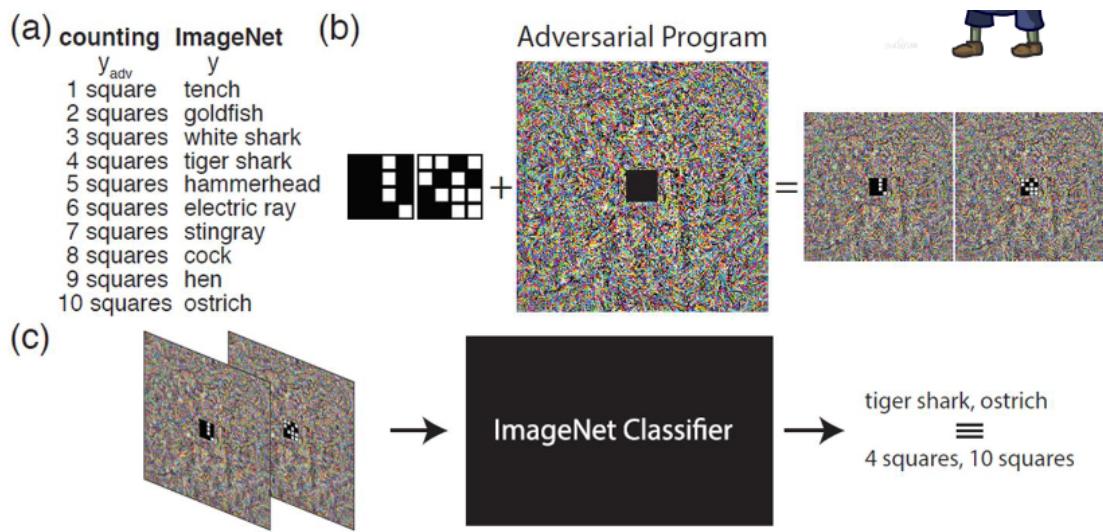
read as an 85-mph sign

(仔细看，数字3“鼻子”被拉长了；将限速35误识别为85：）来自<https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles/>

Adversarial Reprogramming

<https://arxiv.org/abs/1806.11146>

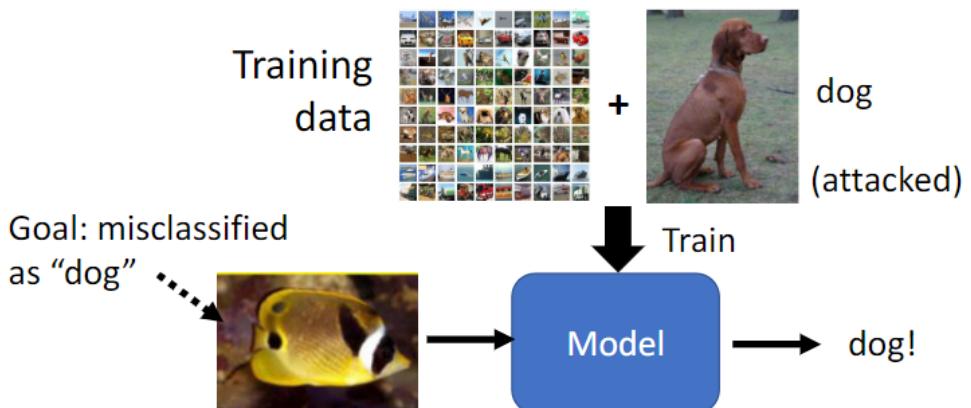
把原影像识别系统放入寄生的僵尸？让它做它本来不想做的事情



数方块的模型：将方块 y_{adv} 的图片嵌入杂讯中，杂讯加入相对应的图像 y ，丢进分类器里边，（ImageNet Classifier原来是识别图像）借用其功能来做到数方块的模型

"Backdoor" in Model

来自文章：<https://arxiv.org/abs/1804.00792> ——发现模型的后门，来自人类的另外一个恶意

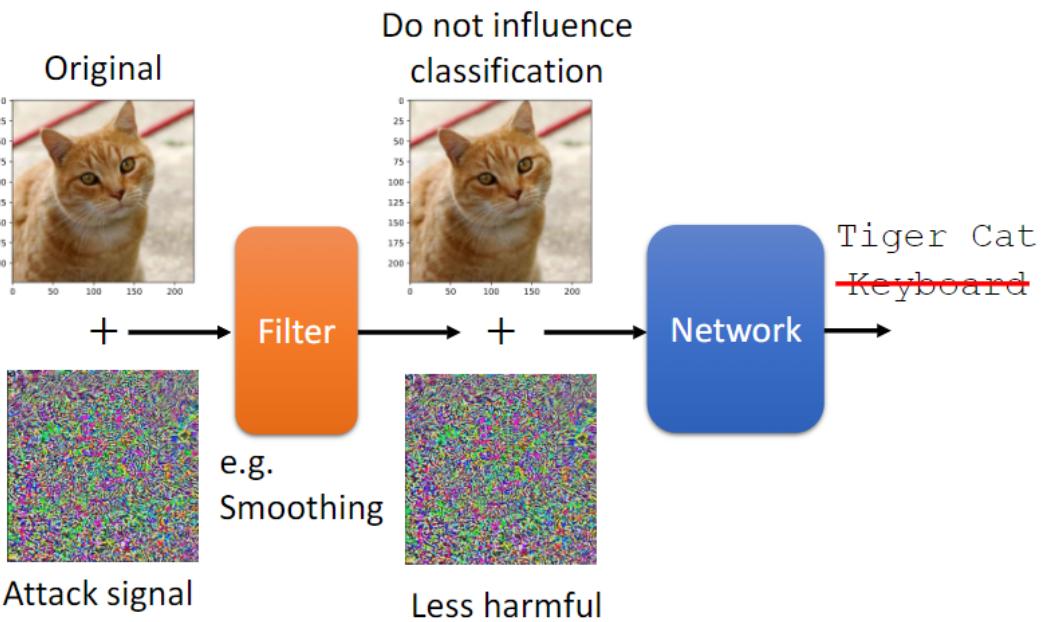


攻击从训练过程中就展开.....在训练资料中图片是正常的而且标注是正常的，但是给模型开了一个后门（样本攻击）。这导致在测试中每次遇到此类样本的时候都会辨识错误。

这启示我们在使用公开的 (open) 训练集，小心其中做的手脚.....

如何防御：被动 v.s. 主动

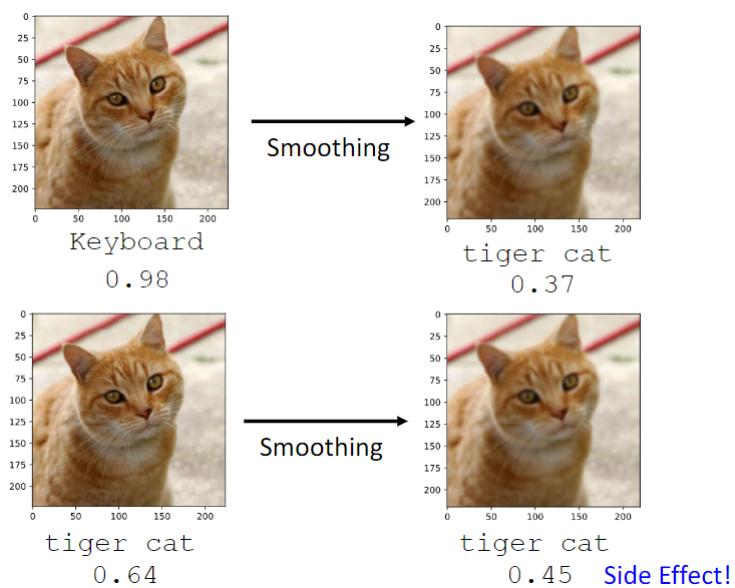
Passive Defense (被动防御)



- 制作一个filter，让加了杂讯的图片（受到攻击）中attack signal的效果减弱（less harmful），避免辨识错误。

如何制作这样一个filter：（最简单的）稍微对图像做一个**模糊化 (Smoothing)**。

会让攻击成功的signal：非常特殊的，往往是攻击成功的一个方向，并不是随机sample出来的噪声。局限性：模糊化图片会让分类器对图像的置信度下降。



- Image Compression:** 对图像压缩再解压缩

来自文章<https://arxiv.org/abs/1704.01155>以及<https://arxiv.org/abs/1802.06816>

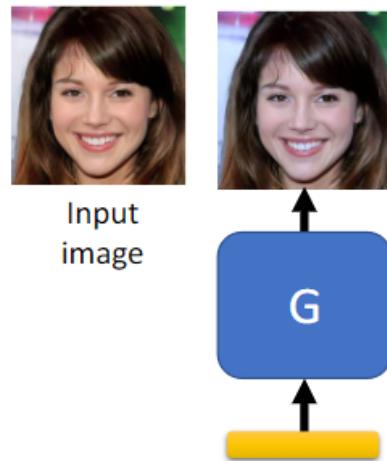


过程中发生的“失真”常常会使攻击的signal丧失效应。

- **Generator**

<https://arxiv.org/abs/1805.06605> —— 目标：如何用generator重新生成的图片

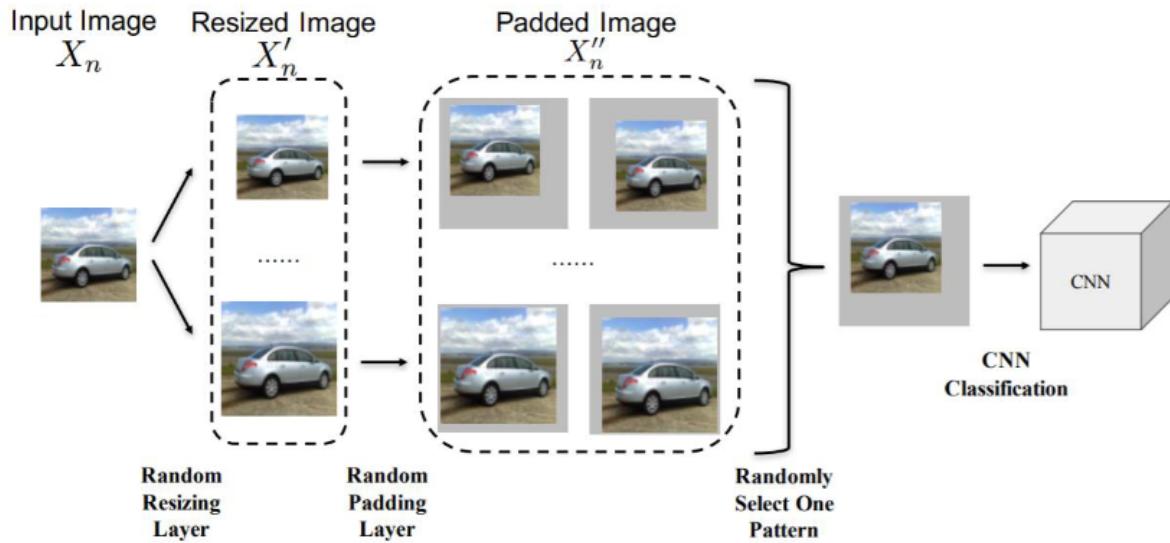
对所有输入的图片 (Input Image) , 用generator重新生成 (reconstruct)



利用Generator抹去加入的attack signal。

局限性

- 一旦被别人 (attacker) 知道被动防御的措施，就立马失效。我们可以把模糊化看作是network之前多加了一层NN，那么攻击者就可以适应这种方式，重新应对。



怎么办？加上自己都不知道怎么随机在哪儿的随机层（如上图所示，来自<https://arxiv.org/abs/1711.01991>）——欲欺敌先瞒己，乱拳打死老师傅，hhhh。

但是假设attacker知道你随机的distribution，也是有可能被攻破的。

Proactive Defense (主动攻击)

直接训练一个对adversarial attack具备鲁棒性的模型——

Adversarial Training

- 给出训练资料 $X = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \dots, (x^N, \hat{y}^N)\}$ ，使用训练资料 X 来训练模型
- 训练阶段就对模型展开攻击——

For $n = 1$ to N : 对于每个给出的 x^n 通过攻击算法找到对应的adversarial input \tilde{x}^n ；这里的 \tilde{x}^n 就是攻击成功的图片，重新进行正确的标记，从而得到新的训练资料

$$X' = \{(\tilde{x}^1, \hat{y}^1), (\tilde{x}^2, \hat{y}^2), \dots, (\tilde{x}^N, \hat{y}^N)\}$$

- 合并两个训练资料 X 和 X' , 更新模型,
- 不断重复上述过程, 不断fix漏洞。用类似于Data Augmentation的方式, 让这种数据驱动的模型更具备鲁棒性。

局限性在于不一定能挡住新的攻击方式, 如果新的攻击方式不在以上这种数据增强的方式被考虑, 那么防御可能无效。而且, 去寻找整理 \tilde{x}^n 的过程也非常繁复耗时。消耗运算资源。

*达到Adversarial Training的效果, 但相比下不需要额外计算消耗资源。文章: [Adversarial Training for Free!](#)

Remarks (总结)

- 攻击: 固定网络参数, 训练调整攻击输入——实践证明攻击非常简单。
- Black Box Attack is possible
- 防御: 被动防御 & 主动防御
- 攻击/防御手段依然在进化 (evolving)

攻击手段 (举例来说)

- [FGSM](#)
- [Basic iterative method](#)
- [L-BFGS](#)
- [Deepfool](#)
- [JSMA](#)
- [C&W](#)
- [Elastic net attack](#)
- [Spatially Transformed](#)
- [One Pixel Attack](#)

.....