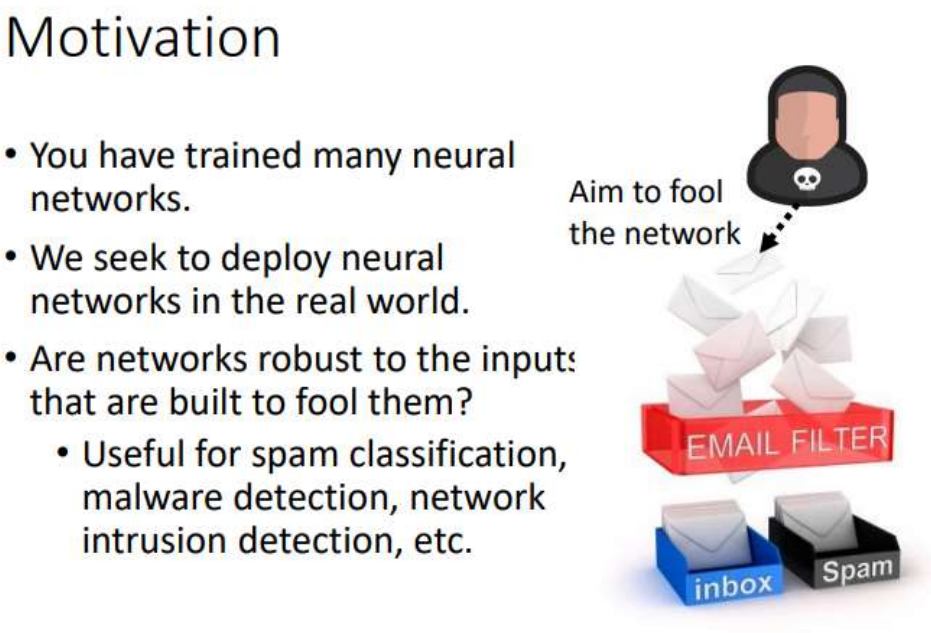
來自人類的惡意攻擊 (Adversarial Attack)

Create at 2022/06/25

- 上課資源：
  1. 來自人類的惡意攻擊 (Adversarial Attack) (上) – 基本概念 (https://www.youtube.com/watch?v=xGQKhbjrFRk)
  2. 來自人類的惡意攻擊 (Adversarial Attack) (下) – 類神經網路能否躲過人類深不見底的惡意？ (https://www.youtube.com/watch?v=z-Q9ia5H2Ig)

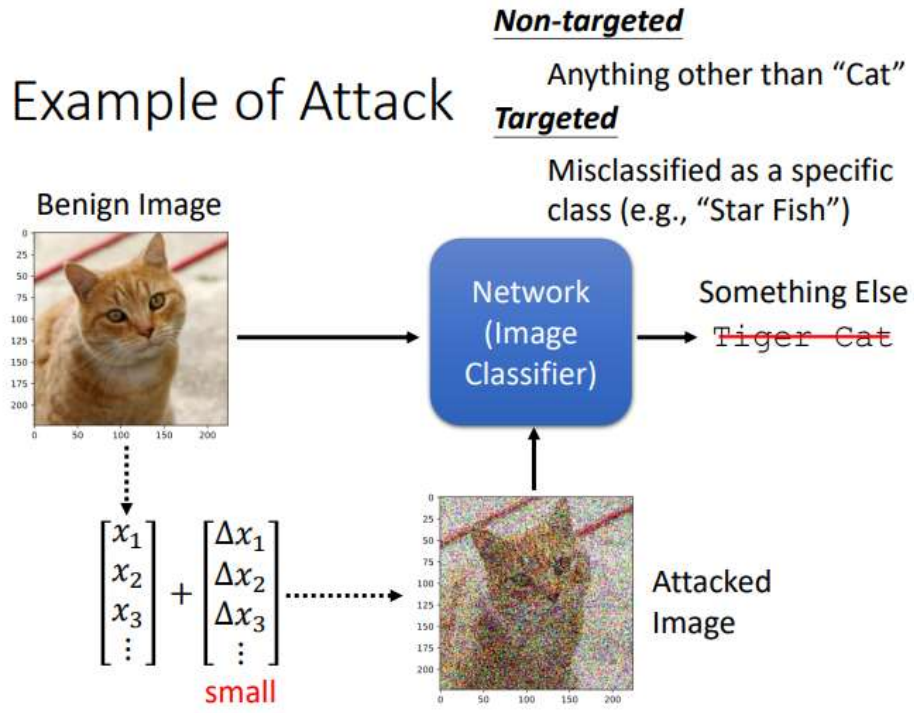
Motivation



- 我們已經訓練了非常多的 neural networks，我們期待可以把這些技術用在真正的應用上
- 光是正確率高是不夠的，還需要能夠應付來自人類的惡意
- 有時候 network 的工作是為了要偵測一些有惡意的行為
  - 但那些被偵測的對象，會去想辦法騙過 network
  - 所以必須要在有人試圖要欺騙它的情況下也得到高的正確率

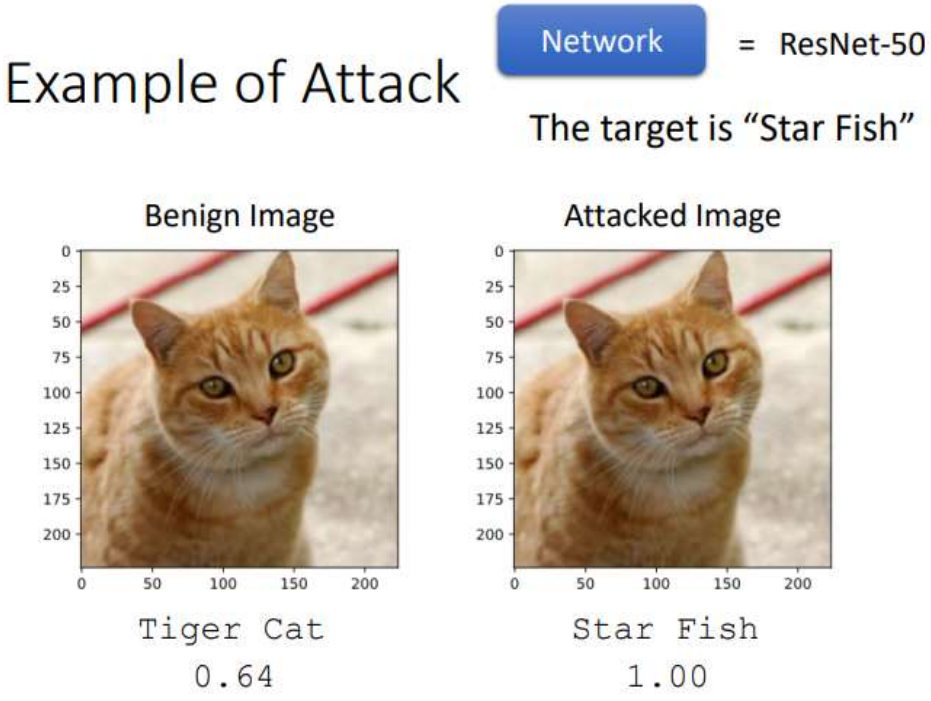
- 人類的惡意是什麼樣子？

Example of Attack

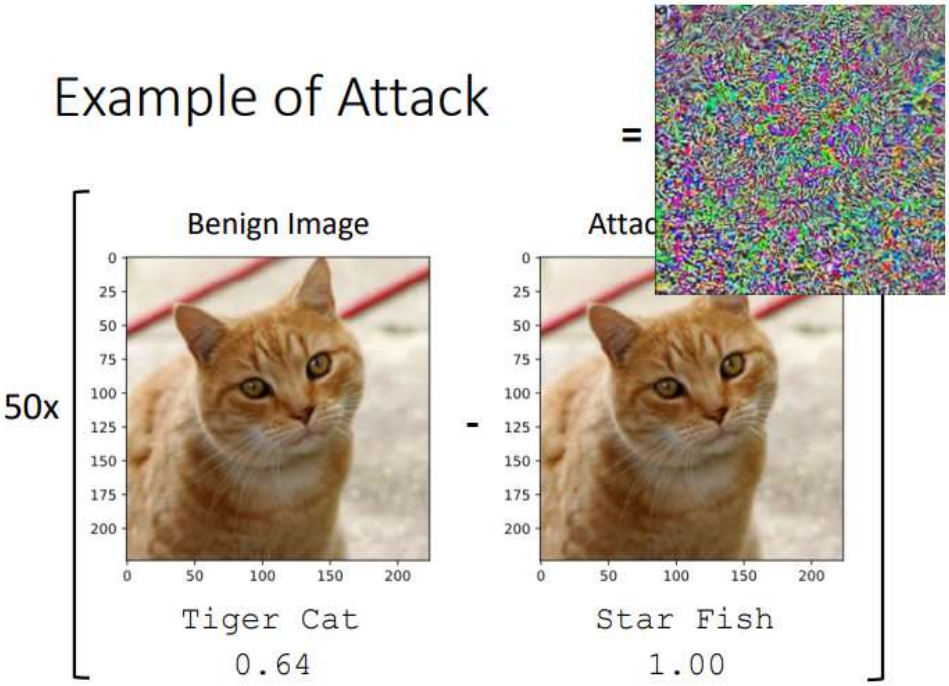


- 我們已經訓練了影像辨識的模型
- 現在把 image 的每個 pixel 都加上小小的雜訊之後，丟到 network
- 有被攻擊的照片稱為 Attacked Image
- 還沒有被加雜訊的照片稱為 Benign Image

- 攻擊可以被分為兩種類型
  - 沒有目標的攻擊 (只要輸出不是貓就好)
  - 有目標的攻擊 (要輸出不是貓，還要是一個特定的東西)



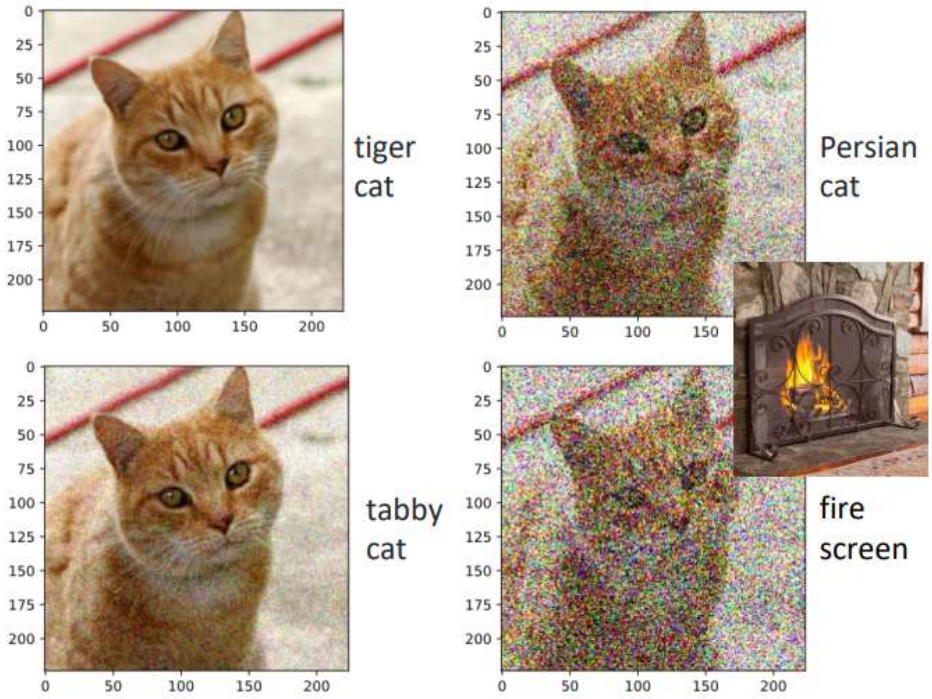
Example of Attack



Example of Attack

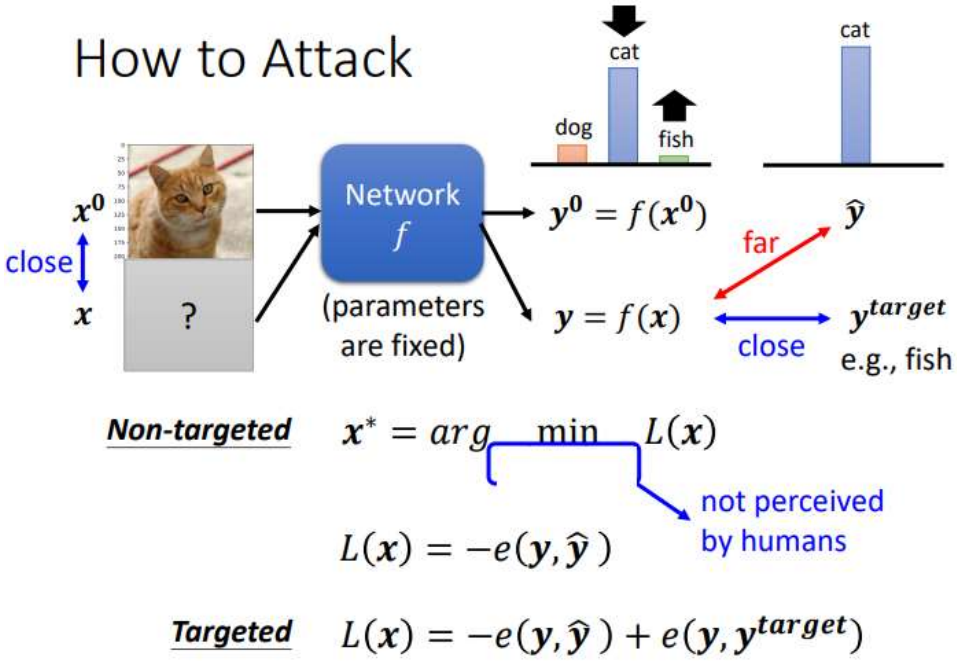
Network = ResNet-50

The target is “Keyboard”

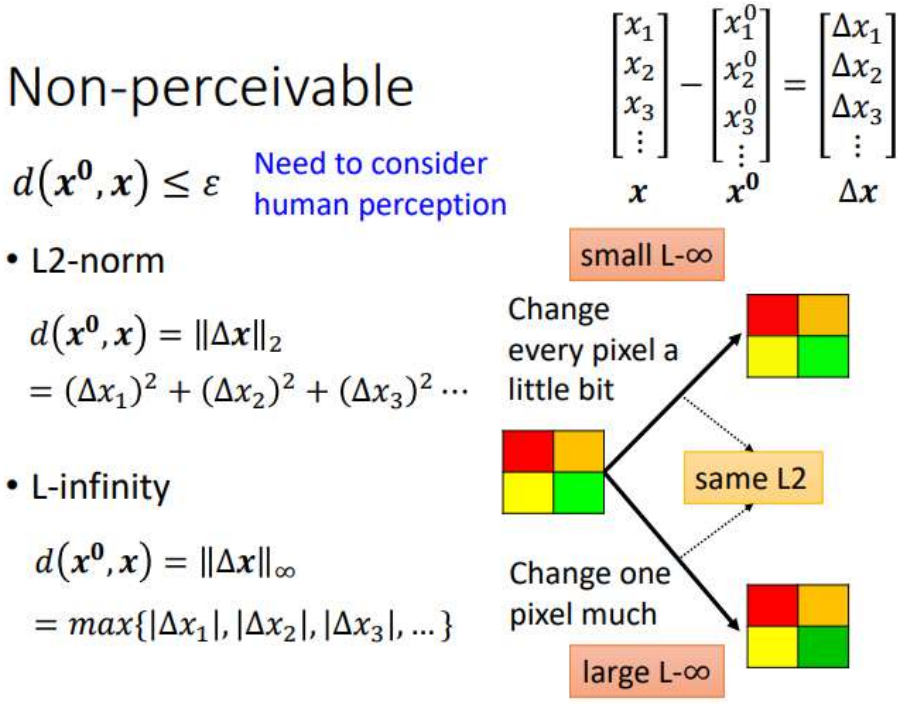


- 如果加入的只是一般的雜訊，那機器不一定會犯錯
- 但加入一個人肉眼看不到的雜訊之後，卻產生天差地遠的結果

How to Attack



- 如何加入非常微小的雜訊，而這個雜訊能讓 network 產生非常錯誤的結果呢？
  - 要怎麼找出 Non-Targeted Attack 的雜訊呢？
    - 現在的目標要找出一張新的圖片，用 x 表示
    - 把 x 丟到 network f 輸出 y，我們希望 y 跟正確答案 $\hat{y}$ 差距越大越好
    - 要解一個 optimization 的問題，先定一個 loss function L 是 y 跟 $\hat{y}$ 之間的差距取一個負號 (因為現在希望 cross entropy 越大越好)，同時期望 loss 越小越好
  - 如果是 Targeted Attack
    - 會先設定好我們的目標，用 y^{target} 代表我們的目標
    - 現在不只希望 y 跟 $\hat{y}$ 越遠越好，還要跟 y^{target} 越接近越好
- 希望 x 跟 x^0 的差距小於某一個 threshold (根據人類的感知能力來決定)
- 人類看起來 x 跟 x^0 是一模一樣的，但是產生結果對 network 來說是非常不一樣的



- 怎麼計算 x 跟 x^0 之間的差距
  - L2-norm
  - L-infinity
- 在決定要使用哪一種方法的時候，要把人類的感知考慮進來
  - L2-norm 跟 L-infinity 哪一個在 attack 的時候是比較好的距離呢？
    - 以右邊的例子為例
    - 要避免被人類發現，光是 L2 小是不夠的
    - 要讓 L-infinity 小才是最好的，才不會被發現

Attack Approach

$w^*, b^* = arg \min_{w,b} L$ Difference?

Update ***input***, not ***parameters***

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

Gradient Descent

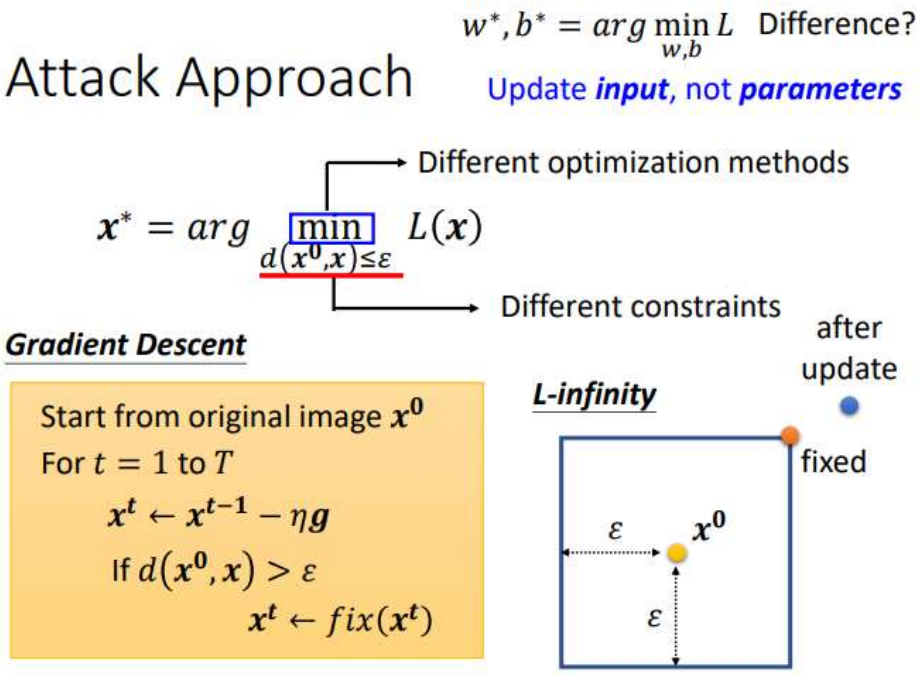
Start from original image x^0

For $t = 1$ to T

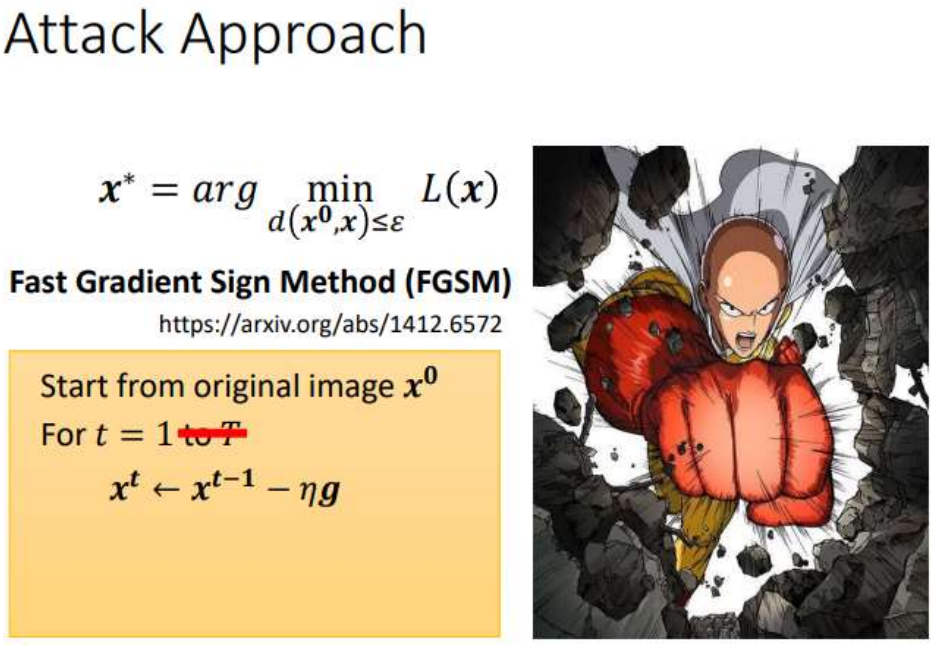
$x^t \leftarrow x^{t-1} - \eta g$

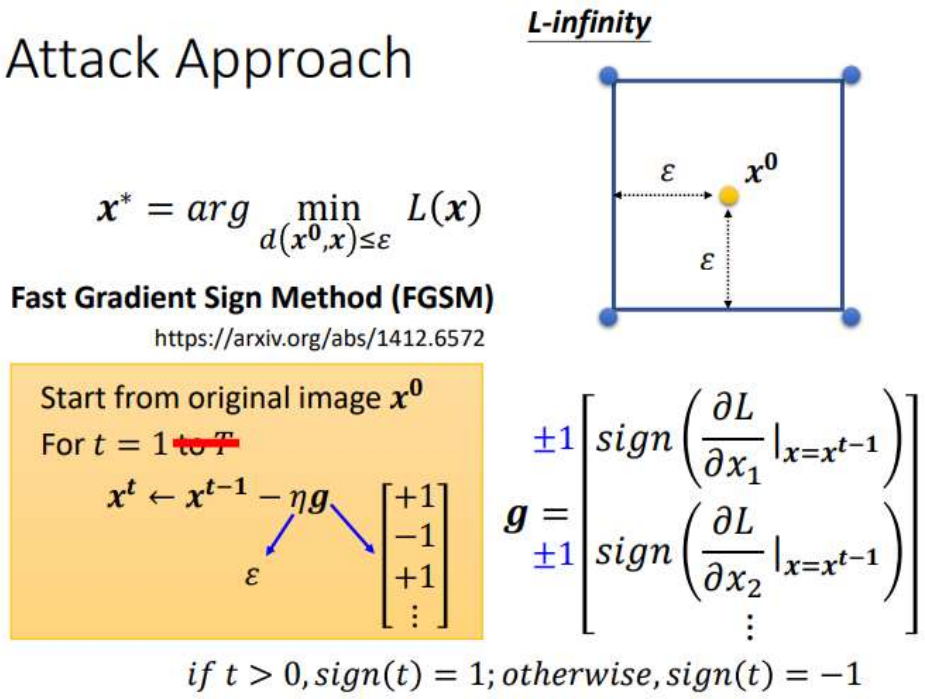
$$g = \begin{bmatrix} \frac{\partial L}{\partial x_1}|_{x=x^{t-1}} \\ \frac{\partial L}{\partial x_2}|_{x=x^{t-1}} \\ \vdots \end{bmatrix}$$

- 現在我們要找一個 x 去 Minimize Loss L
- 但是 x 跟 x^0 它們之間要小於等於某一個 threshold
  - 現在先把這個限制拿掉，就跟我們 train 一個模型沒什麼差別
  - 今天只是把調整參數這部分，改成調整 network 的 input 而已
  - 現在把 input image 看作是 network 參數的一部分
  - 然後去 minimize loss function 就結束了
  - 現在 network 參數固定 (這堂課把 network 固定參數)，只去調 input 的部分，讓 input 改變去 minimum Loss 就結束了，用的一樣是 Gradient Descent
- 怎麼做呢？
  - 先 initialize 一個 image，跟 x^0 越接近越好，所以直接設 x^0
  - 接著就跟一般的 gradient descent 一樣
    - iterative 去 update 參數
    - 每個 iteration 都會計算 gradient
    - 只是這個 gradient 不是 network 參數對 loss 的 gradient
    - 是 input image x 對 loss 的 gradient
    - 算出這個 gradient 去 update image 就結束了
      - 原本 image x^0 - ηg 得到新的 image

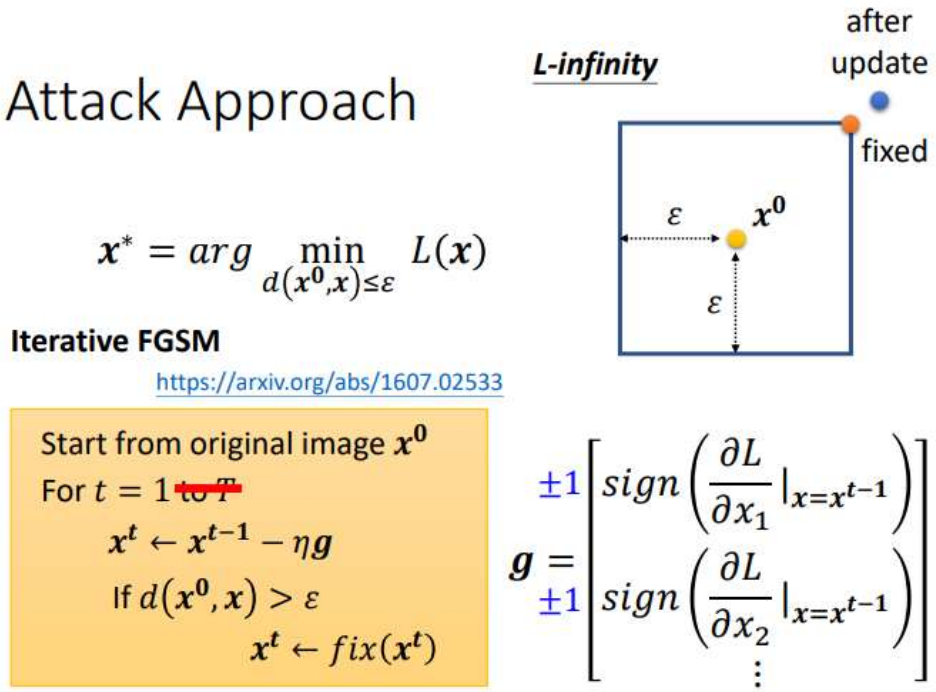


- 接下來把 constraint 加進去
- 現在限制 x 跟 x^0 要小於某個 threshold
- 現在需要在 gradient descent 裡面，再加一個 module
- update 完參數之後，發現 x^t 跟 x^0 大於 threshold 之後，就把 x^t 做修改，把它改回符合限制就結束了





- Fast Gradient Sign Method (FGSM)
  - 它只 update 一次參數
  - 而且不要直接用 gradient descent 的值，給它取一個 Sign
    - 如果括號裡面的值大於 0，就輸出 1
    - 如果括號裡面的值小於 0，就輸出 -1
  - 而且 learning rate η 設跟 threshold 一模一樣
    - 攻擊完之後，一定落在藍色框框的四個角落的地方
- 一擊必殺



- 多跑幾個 iterative
- 但是多跑幾個 iterative 的壞處是，有可能一不小心就出界，跑出四方形的範圍
- 解決方法：把它拉回來就結束了

White Box v.s. Black Box

- In the previous attack, we know the network parameters θ
  - This is called ***White Box Attack***.
- You cannot obtain model parameters in most online API.
- Are we safe if we do not release model? ☺
- No, because ***Black Box Attack*** is possible. ☹

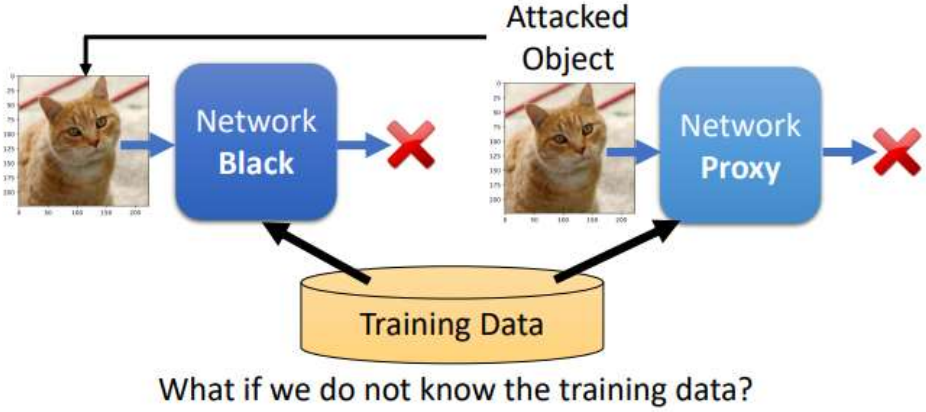
$$g = \begin{bmatrix} sign\left(\frac{\partial L}{\partial x_1}|_{x=x^{t-1}}\right) \\ sign\left(\frac{\partial L}{\partial x_2}|_{x=x^{t-1}}\right) \\ \vdots \end{bmatrix}$$

- 知道模型參數的攻擊，叫做 White Box Attack
- 不知道模型參數的攻擊，叫做 Black Box Attack

Black Box Attack

Black Box Attack

If you have the training data of the target network
Train a proxy network yourself
Using the proxy network to generate attacked objects



What if we do not know the training data?

- 假設我們知道這個 network 是用什麼訓練資料訓練出來的話
- 那我們可以去訓練一個 proxy network
- 讓這個 network 來模仿我們要攻擊的對象
- 我們要攻擊的對象跟 proxy network 如果都是用同樣的訓練資料訓練出來的話，也許它們就會有一定的相似程度
- 如果要攻擊的對象跟 proxy network 有一定的相似程度的話，我們只要對 proxy 的 network 進行攻擊，也許這個有被攻擊過的 image 拿去丟到我們不知道參數的 network 上攻擊也會成功

Black Box Attack

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

Be Attacked

Proxy

	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
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%

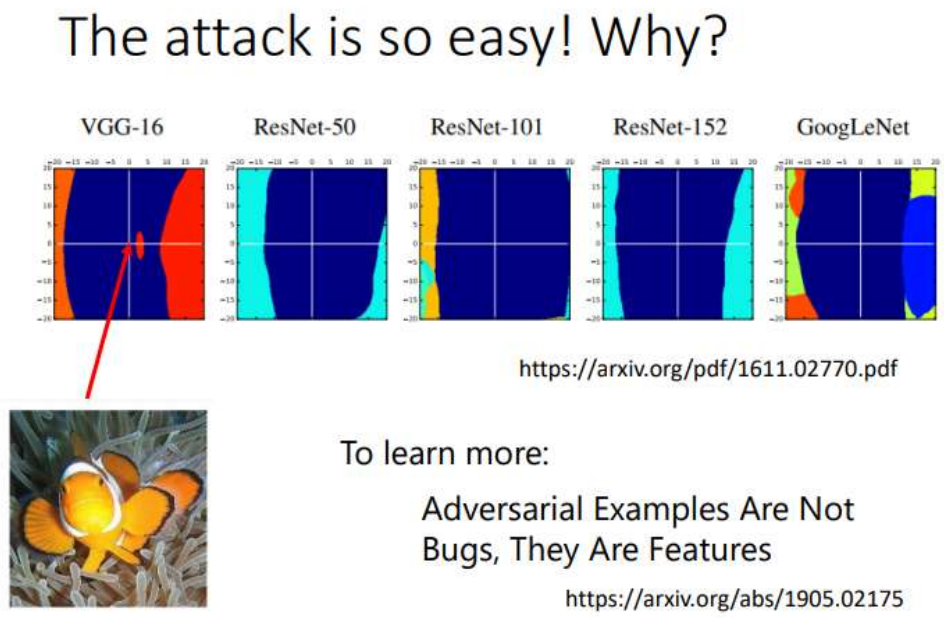
(lower accuracy → more successful attack)

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%

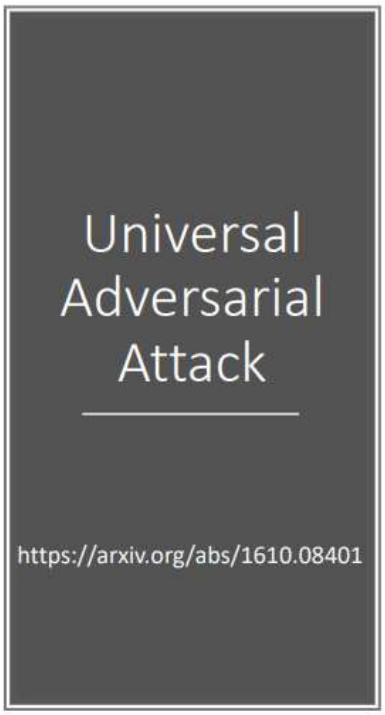
- Column : 要被攻擊的 network
- Row : proxy network
- 對角線代表 proxy network 根要被攻擊的 network 是一樣的，是白箱攻擊
- 越低的正確率代表攻擊越成功，我們現在是站在攻擊方
- 黑箱攻擊也有成功的可能性
- 在黑箱攻擊的時候，Target Attack 比較難成功，但是 Non-Target Attack 還是比較容易成功的

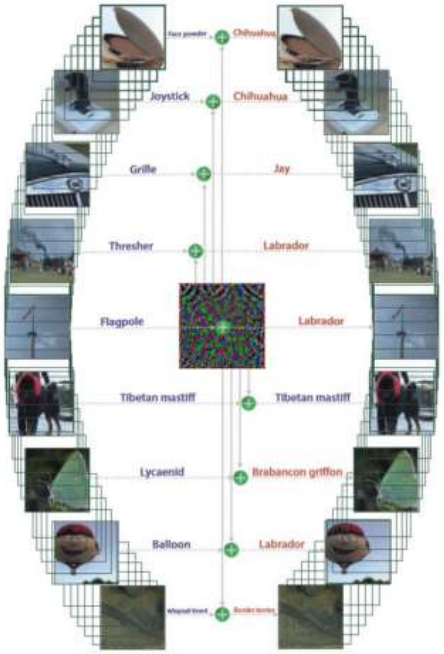
- 要如何增加 Black Box Attack 的成功率？
  - Ensemble Network Attack
    - Column : 要被攻擊的 network
    - Row : 把 Column 的五個模型都集合起來，但是拿掉 Row 欄位的，用另外四個 network
    - 找一張 image 攻擊 4 個 network
    - 非對角線的是白箱攻擊
    - 對角線的才是黑箱攻擊



- 為什麼攻擊這件事這麼容易成功？
- 圖上的紅色圓點代表小丑魚的圖片
- 橫軸跟縱軸分別是把這張圖片往兩個不同的方向移動
- 一張圖片是一個非常高維的向量
- 最左邊的圖
  - 橫軸：VGG-16 上可以攻擊成功的方向
  - 縱軸：一個隨機的方向
- 五張圖中間的深藍色區域都蠻相近
  - 深藍色區域：會被辨識成小丑魚圖片的範圍
- 也許 Adversarial Attack 會成功的原因是來自於資料上的問題，當我們有足夠的資料，也許就有機會避免 Adversarial Attack

- Attack 的 Signal 希望它越小越好
- One pixel attack
  - 只能動圖片裡面的一個 pixel 而已
  - 希望影像辨識系統的判斷就必須要有錯誤
  - 黑色是攻擊前的正確結果
  - 藍色是攻擊後的影像辨識結果





Black Box Attack is also possible!

- 更厲害的攻擊方式
- Universal Adversarial Attack
- 找到一個 Attack Signal 加在非常多不同的圖片上都可以讓影像辨識系統辨識錯誤

Beyond Images

感謝吳海濱同學提供實驗結果

- Speech processing

Detect synthesized speech

- Natural language processing

https://arxiv.org/abs/1908.07125

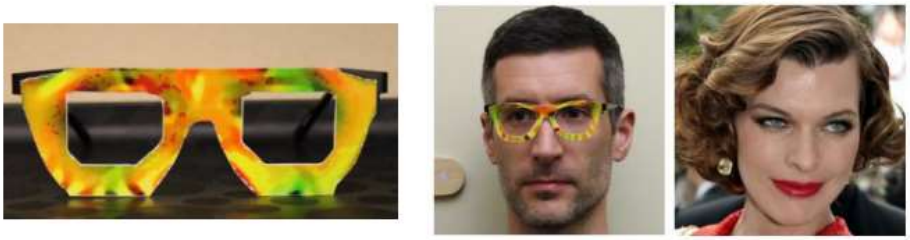
Question: Why did he walk?
For exercise, Tesla walked between 8 to 10 miles per day. He squished his toes one hundred times for each foot every night, saying that it stimulated his brain cells. ' exercise

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 ..... crime and poverty

- 偵測一段聲音是不是合成的系統，也會被輕易的攻擊
- 文字也會被攻擊

https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf

Attack in the Physical World



- An attacker would need to find perturbations that generalize beyond a single image.
- Extreme differences between adjacent pixels in the perturbation are unlikely to be accurately captured by cameras.
- It is desirable to craft perturbations that are comprised mostly of colors reproducible by the printer.

- 在真實世界中，可以用多個角度去看一個人，也許雜訊騙過了某一個角度，但沒辦法在所有的角度都騙過影像辨識系統
- 把攝像頭本身解析度能力的極限考慮進去
- 這個眼鏡能不能真的被做出來，因為某些顏色在電腦裡面跟在真實的世界看起來會有差異

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%

- 對車牌辨識系統進行攻擊

Attack in the Physical World

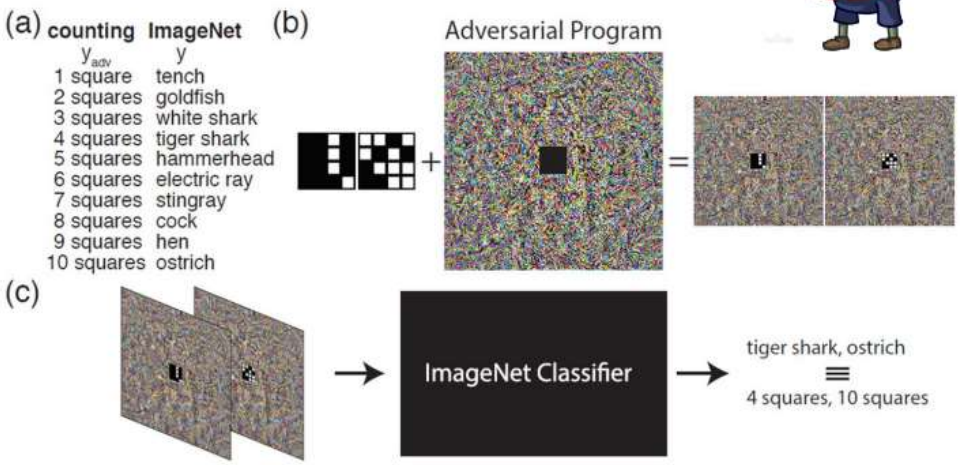
read as an 85-mph sign

https://youtu.be/4uGV_fRj0UA

https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles/

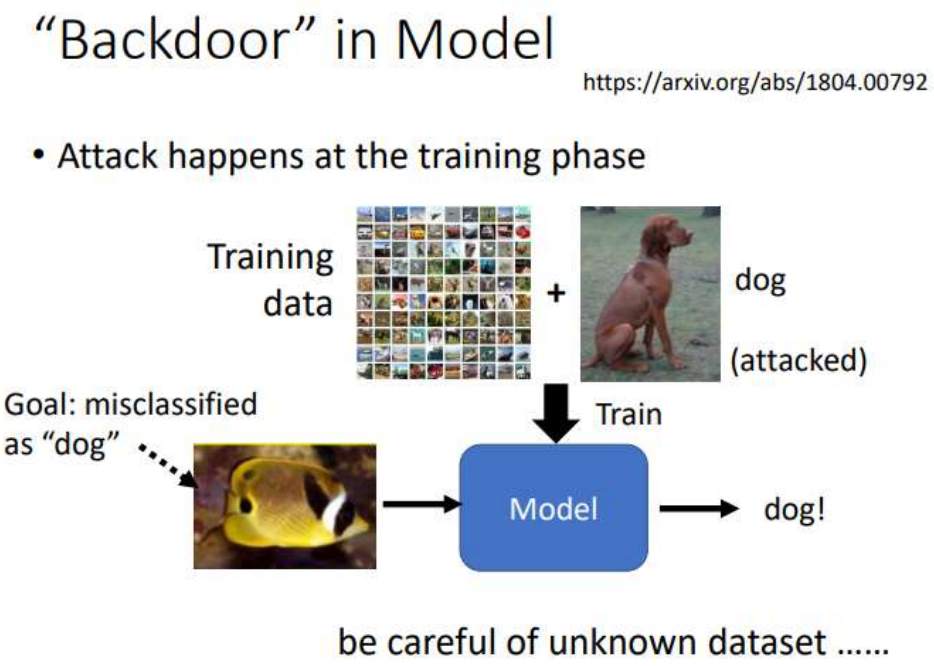
- 把 3 中間拉長，會辨識成 8

Adversarial Reprogramming



https://arxiv.org/abs/1806.11146

- **Adversarial Reprogramming**
- 想做一個方塊辨識系統，去數圖片裡面有幾個方塊 1~10 個
- 但它不想 train 自己的模型，它想要寄生在某一個已有的 train 在 ImageNet 模型上面
- ImageNet 就給它圖片，然後辨識裡面有什麼樣的東西
- 希望輸入一張圖片，如果圖片裡面有兩個方塊的時候，ImageNet 模型就要輸出 goldfish，如果三個方塊，就輸出 White Shark
- 可以操控 ImageNet train 出來的模型，做它本來不是訓練要做的事情
- 目標
  - 把 4 個方塊的圖片嵌在雜訊的中間，丟到 ImageNet 輸出 tiger shark
  - 把 10 個方塊的圖片嵌在雜訊的中間，丟到 ImageNet 輸出 ostrich
- 怎麼做？
  - 把圖片外面加雜訊，再把圖片丟進 image classifier 裡面，就會照你的操控，做它本來不是訓練來要做的事情

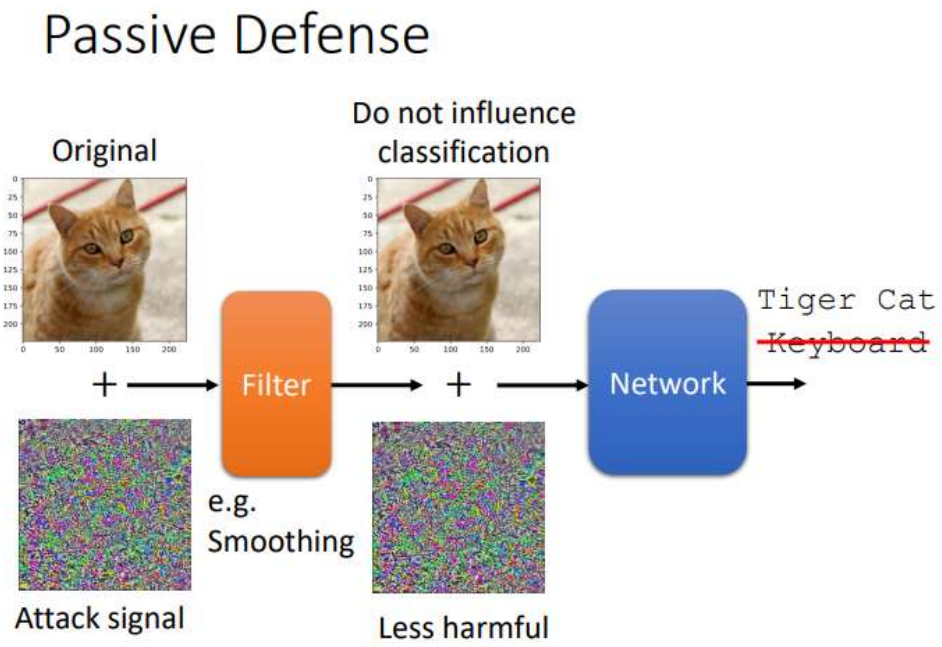


- 在模型裡面開一個後門
- 目前為止，攻擊都是在測試的階段才展開
- 有沒有可能在訓練的階段就展開攻擊呢？
  - 之前我們都是在測試階段，模型已經訓練好之後才在圖片上面加入雜訊去騙過模型
  - 可能在訓練資料裡面加一些，人看起來沒有問題但實際上有問題的資料，讓模型訓練完之後，模型就開了後門，在測試階段它就會辨識錯誤
  - 而且只會對某一張圖片辨識錯誤，對其他圖片是沒有問題的
  - 所以不會覺得你的模型訓練完以後有什麼不對的地方
  - 直到有人拿這張圖片來攻擊你的模型的時候，才會發現這個模型是有被下毒的

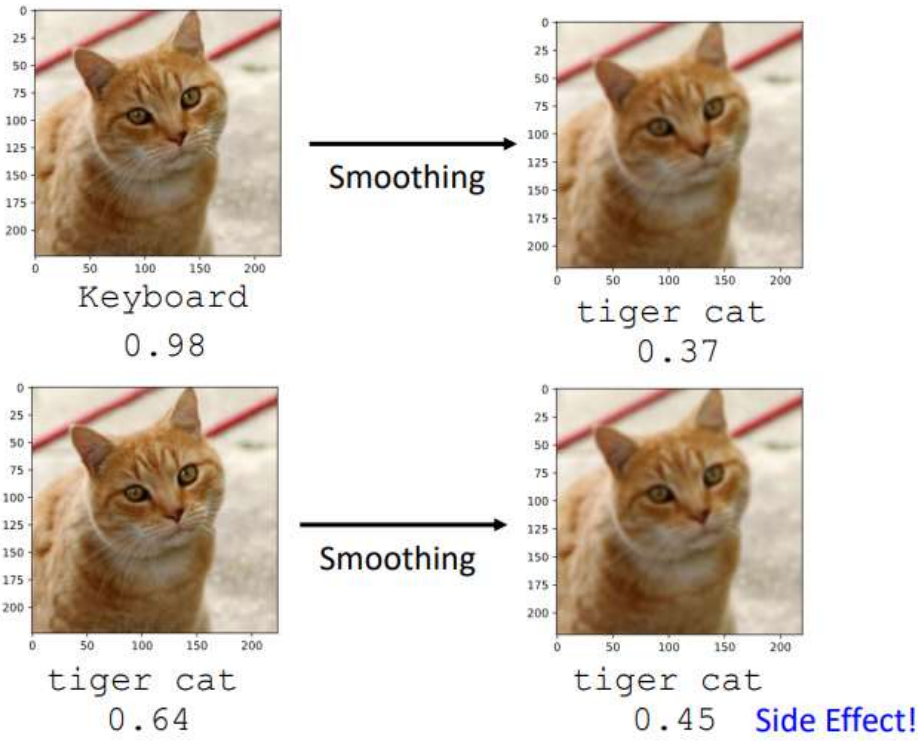
Defense

- 防禦的方式分成兩類
  1. 主動防禦
  2. 被動防禦

被動防禦



- 被動防禦怎麼做呢？
  - network 不動，在模型 (network) 前面加一個盾牌 (Filter)
  - 本來圖片加上雜訊 signal 可以騙過 network，但是 filter 可以削減 Attack Signal 的威力
  - 一般的圖片通過 filter 的時候，不太會受到影響
  - 但是 attack signal 通過 filter 之後，會失去它的威力，讓你的 network 不會辨識錯誤



- filter 可以當成是做 smoothing，就可以達到防禦的效果
- Adversarial Attack 的 signal 其實只有某一個方向上的某一種攻擊的 signal 才能夠成功，不是隨便 sample 一個 noise 都可以攻擊成功
- 所以會攻擊成功的訊號，是非常特殊的
- 當加上 smoothing 之後，攻擊成功的訊號就失去了攻擊的威力，而且 smoothing 對原來的圖片影響很小，不太會影響影像辨識的結果
- 但是 smoothing 也有一些副作用，可能辨識的 confidence 下降
- 所以不能把 smoothing 做得太過頭，可能導致原本正常的影像辨識錯誤

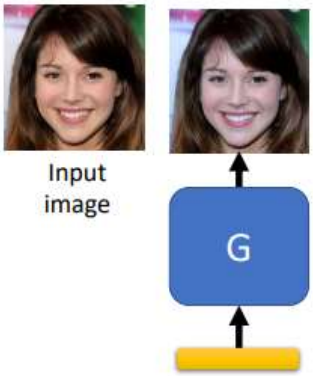
Passive Defense

Image Compression

https://arxiv.org/abs/1704.01155
https://arxiv.org/abs/1802.06816

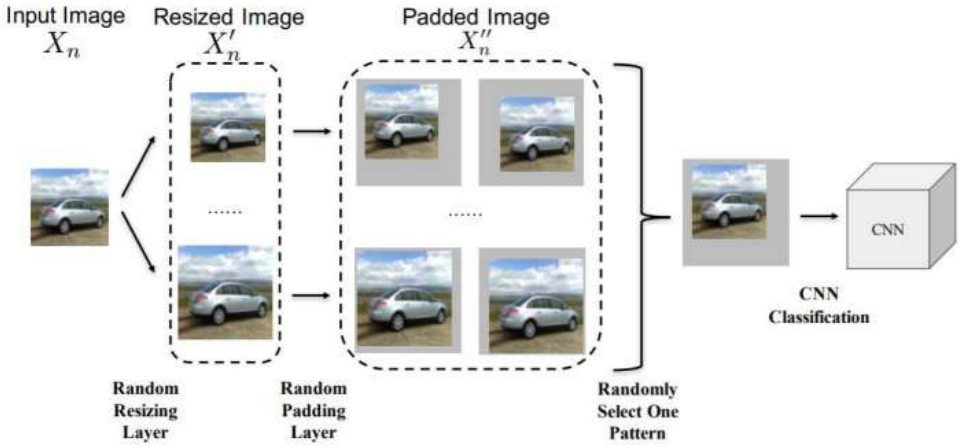
Generator

https://arxiv.org/abs/1805.06605



- **Image Compression**：對影像做壓縮再解壓縮，把一張圖片存成 JPEG 檔之後會失真，失真這件事情就可以讓被攻擊的圖片失去它攻擊的威力
- **Generator**：把輸入的圖片用 generator 重新畫過

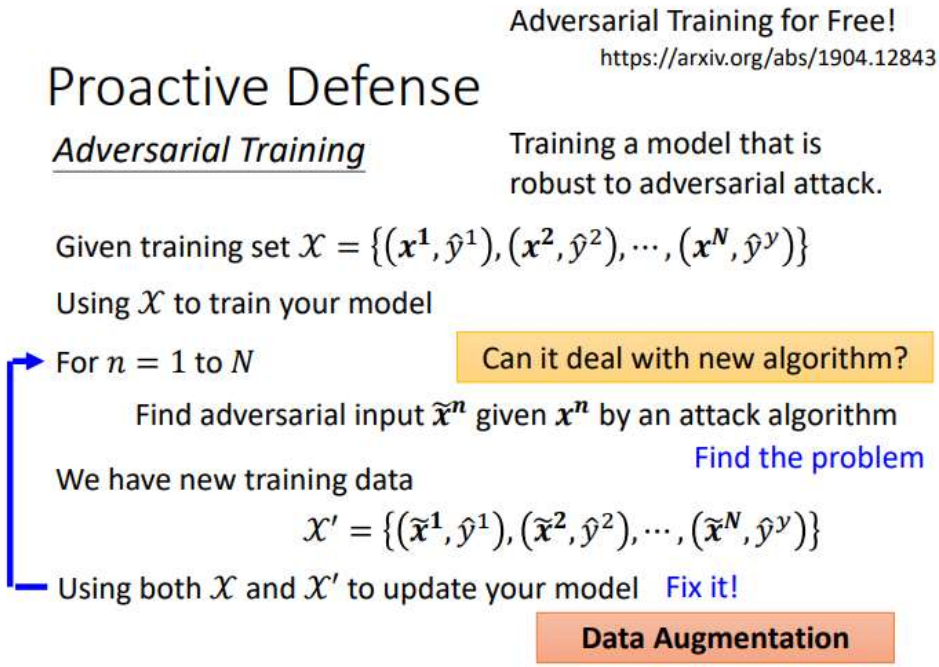
Passive Defense - Randomization



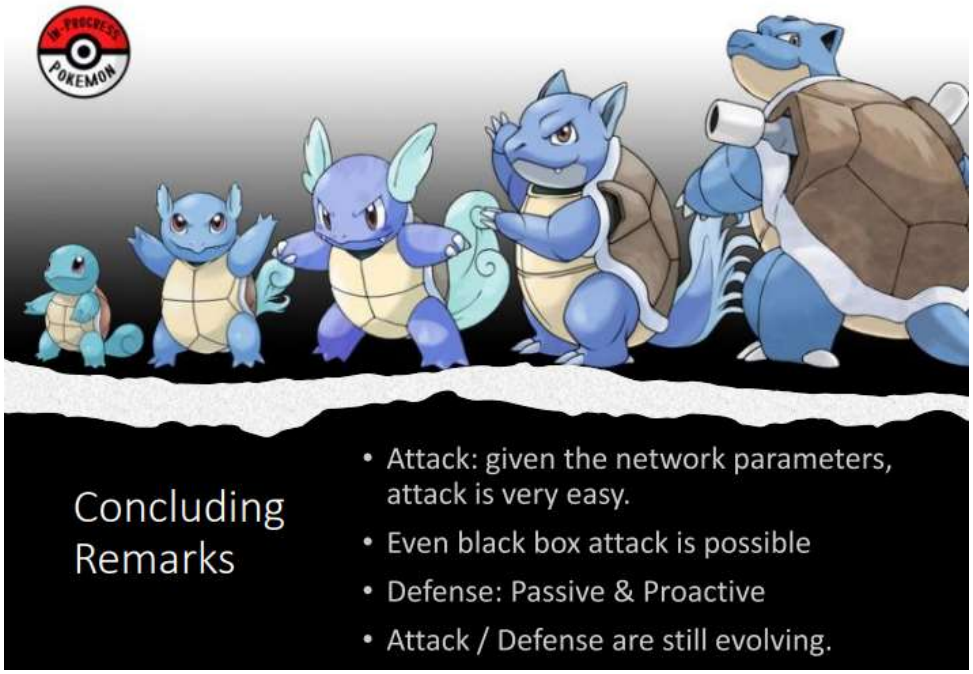
https://arxiv.org/abs/1711.01991

- 被動防禦有一個非常大的弱點
  - smoothing 這個方法只要一旦被別人知道你會做這件事情，馬上就失去效用
  - 可以完全把 smoothing 這件事情想成是 network 的第一層
  - 所以 smoothing 這件事等於是在 network 前面多加了一層
  - 別人如果知道你在 network 前面多加這一層，就多加這一層放到攻擊的過程中，就可以產生一個 signal，是可以躲過 smoothing 這種防禦方式的
- 更強大的防禦方法就是 - 加上隨機性
  - 不要被別人猜中你的下一招

主動防禦



- 一開始就訓練一個比較 Robust(不會被攻破) 的模型
- 這種訓練方式叫做 Adversarial Training
  - 先訓練好一個模型，然後看這個模型有沒有漏洞
  - 把漏洞找出來
  - 再把漏洞填起來
  - 不斷地找漏洞，找到就把它填起來
  - 可以看成是 Data Augmentation 的方法，因為我們產生了更多的圖片，等於是做了資料增強
  - 所以有人也會把 Adversarial Training 當作一個單純的資料增強的方式



- 目前攻擊跟防禦都仍然不斷地在進化中

Attack Approaches

- FGSM (https://arxiv.org/abs/1412.6572)
- Basic iterative method (https://arxiv.org/abs/1607.02533)
- L-BFGS (https://arxiv.org/abs/1312.6199)
- Deepfool (https://arxiv.org/abs/1511.04599)
- JSMA (https://arxiv.org/abs/1511.07528)
- C&W (https://arxiv.org/abs/1608.04644)
- Elastic net attack (https://arxiv.org/abs/1709.04114)
- Spatially Transformed (https://arxiv.org/abs/1801.02612)
- One Pixel Attack (https://arxiv.org/abs/1710.08864)
- ..... only list a few

tags: 2022 李宏毅_機器學習