

## Machine Learning HW5 Report

學號：R07943095 系級：EDA 碩一 姓名：劉世棠

1. (1%) 試說明 hw5\_best.sh 攻擊的方法，包括使用的 proxy model、方法、參數等。此方法和 FGSM 的差異為何？如何影響你的結果？請完整討論。(依內容完整度給分)

這邊實作的方式依然為 FGSM，基礎方法一樣是藉由 pytorch 拿輸入資料的梯度，後來有嘗試使用重複此過程可惜效果不佳(L-inf norm 太大)。

Model	Success rate	L-inf norm
BSET FGSM	91.0%	3.00
FGSM modified	97.0%	96.9

2. (1%) 請列出 hw5\_fgsm.sh 和 hw5\_best.sh 的結果 (使用的 proxy model、success rate、L-inf. norm)。

因為我做到後面最好的就是使用 pytorch resnet50 的 FGSM，故我這題將比較其與其他模型的差別:

Model	Success rate	L-inf norm
Keras VGG 16 FGSM	50.5%	9.74
Pytorch resnet16 FGSM	91.0%	3.00
Pytorch resnet16 FGSM modified	97.0%	96.9

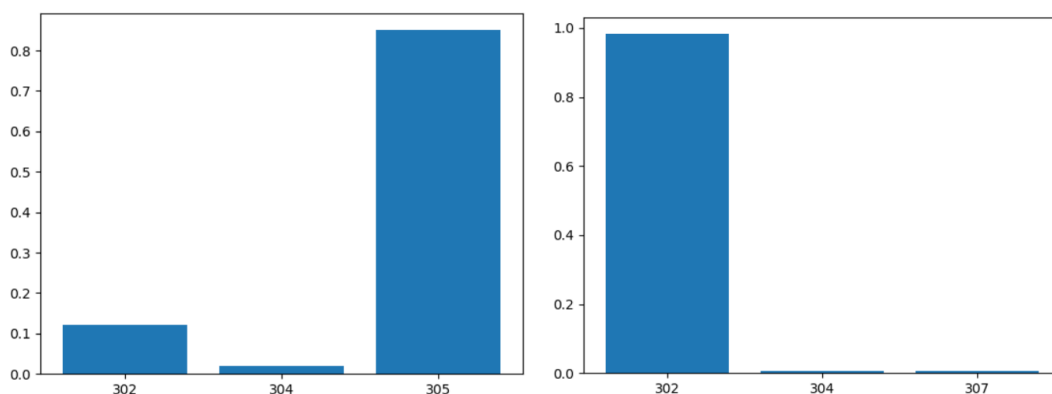
3. (1%) 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型？請說明你的觀察和理由。

⇒ Keras 的成績都不好看，所以採用 Pytorch 做嘗試，結果在 resnet50 有不錯的結果(過 strong based line)

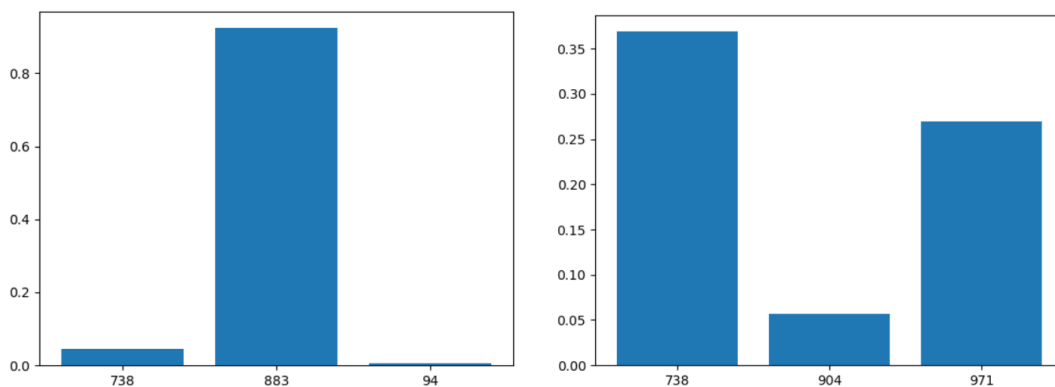
4. (1%) 請以 hw5\_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。

縱軸為機率/橫軸為種類

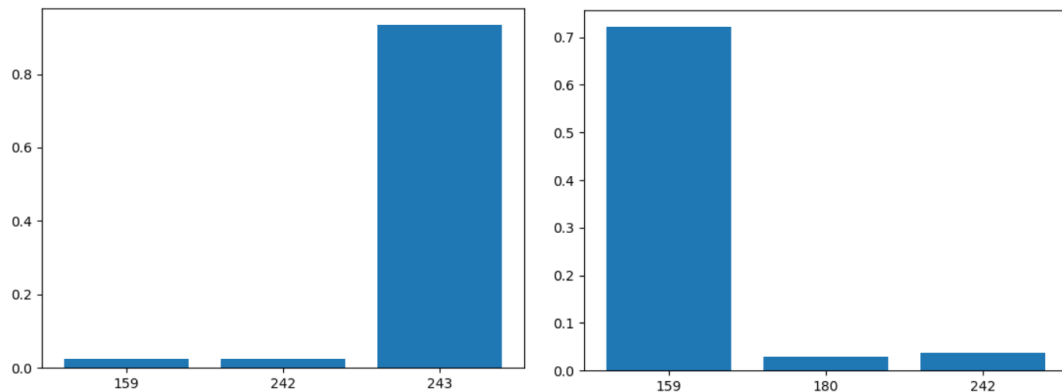
第一張(左為攻擊前;右為攻擊後):



第二張(左為攻擊前;右為攻擊後):



第三張(左為攻擊前;右為攻擊後):



⇒ 從上面圖可以發現攻擊後與攻擊前的機率分布不一樣，而且很有趣的是約莫第二高的機率再攻擊後會變成最高的機率，不過這可能與我的實作方式有關。

5. (1%) 請將你產生出來的 **adversarial img**，以任一種 **smoothing** 的方式實作被動防禦 (**passive defense**)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你防禦前後的 **success rate**，並簡要說明你的觀察。另外也請討論此防禦對原始圖片會有什麼影響。

<input type="checkbox"/>	22	2019-04-26 13:57:06	0.910	3.0000
<input type="checkbox"/>	23	2019-04-26 13:58:38	0.680	131.9250

⇒ 上面是原本用來攻擊的圖片，下面是將圖片經過 **median filter**(手刻)後得出的結果，可以發現攻擊成功率下降了，而且照理來說 **L-infinity** 越大則攻擊稱功率應該越高，可是這邊明顯變大許多而沒有效果，因此可以知道 **filter** 用於防禦有不錯的效果。

⇒ 使用程式碼方式如下

⇒ `python prob_05.py <attackImg dir> <defenseImg dir>`