

Machine Learning HW5 Report

學號：B06901007 系級：電機二 姓名：戴子宜
2019.4.26

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)

我的 hw5_best.sh 是用 resnet50 當作 proxy model，並且使用 FGSM 的方法，epsilon 設為 10/255，loss 為 $(-1) * \text{true label 和 predict label 的 cross entropy}$ 。這樣做出來的 attack success rate 是 0.895，L-infinity 是 20.87。

我試過其他的方法代替 FGSM 但最好的結果都沒有比 FGSM 的好。我試過下面幾種方法：

- 1) 和 FGSM 很像，但為了降低 L-infinity 所以把 epsilon 調小，然後重複一樣的動作 10-50 個 epoch，這樣的方法確實可以降低 L-infinity 到 4.28 左右，但得到的 attack success rate 都只在 0.4 和 0.5 之間。
- 2) 使用 gradient descent，Adagrad 將 lr 設為 0.005，將原本的 loss 改成 $(-1) * \text{Cross Entropy}(\text{true label}, \text{predict label}) + \mu (x - x_0)^2$ ，其中 x 是目前的圖片， x_0 是最原本的圖片， μ 是調整兩者比重的係數，設 10000 因為兩個的數量級差很多，這個方法 attack success rate 可以到 0.895（和 FGSM 一樣），但 success rate 高的時候 L-infinity 也都很 20 左右。可能是因為 loss 的算法用的是 L_2 ，所以求出 L-infinity 不一定很小。
- 3) 和 2) 類似的方法，把 loss 改為 $(-1) * \text{Cross Entropy}(\text{true label}, \text{predict label}) + \mu \|x - x_0\|_\infty$ ，但這樣第二項 loss 的影響很小，因為去更新 gradient 的時候第二項只對目前 max 的 pixel 有影響。

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)。

FGSM (resnet50 是 best 的結果)

	Attack success rate	L-infinity
vgg16	0.460	20.85
vgg19	0.47	20.91
resnet50	0.895	20.87
resnet101	0.58	20.91
densenet121	0.570	20.765
densenet169	0.615	20.85


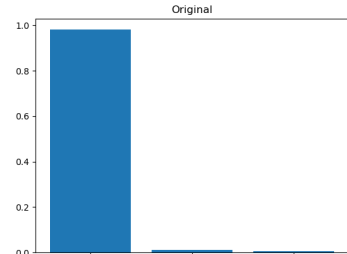
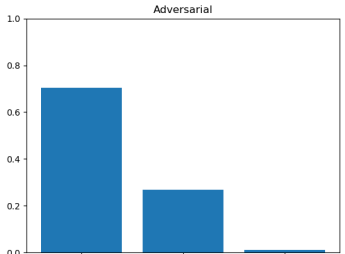
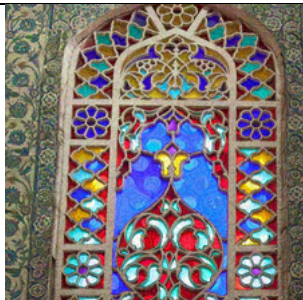
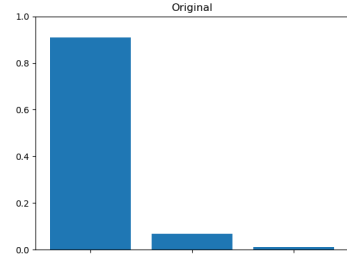
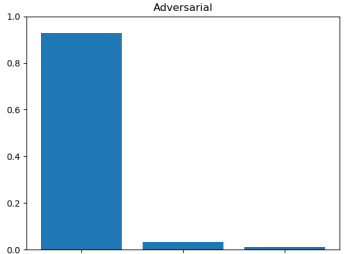

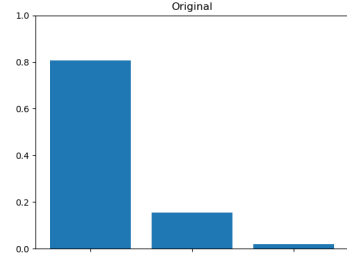
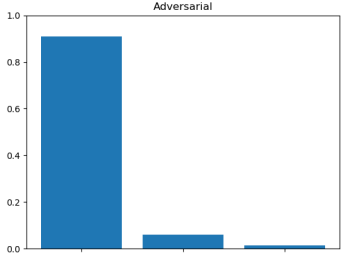
利用 1) 的方法

	Attack success rate	L-infinity
vgg16	0.430	19.99
vgg19	0.445	20.6
resnet50	0.895	20.01
resnet101	0.54	20.65
densenet121	0.55	20.23
densenet169	0.6	20.1

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。

背後的 model 應該為 resnet50，因為從上一題可以看出，一樣的參數和方法，但 resnet50 的 attack success rate 越高。



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)。

Image	Original	Adversarial																
	 <table><caption>Original Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>915</td><td>0.98</td></tr><tr><td>669</td><td>0.02</td></tr><tr><td>672</td><td>0.01</td></tr></tbody></table>	Label	Probability	915	0.98	669	0.02	672	0.01	 <table><caption>Adversarial Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>669</td><td>0.70</td></tr><tr><td>794</td><td>0.28</td></tr><tr><td>905</td><td>0.02</td></tr></tbody></table>	Label	Probability	669	0.70	794	0.28	905	0.02
Label	Probability																	
915	0.98																	
669	0.02																	
672	0.01																	
Label	Probability																	
669	0.70																	
794	0.28																	
905	0.02																	
	 <table><caption>Original Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>741</td><td>0.90</td></tr><tr><td>887</td><td>0.08</td></tr><tr><td>668</td><td>0.02</td></tr></tbody></table>	Label	Probability	741	0.90	887	0.08	668	0.02	 <table><caption>Adversarial Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>887</td><td>0.92</td></tr><tr><td>669</td><td>0.05</td></tr><tr><td>884</td><td>0.03</td></tr></tbody></table>	Label	Probability	887	0.92	669	0.05	884	0.03
Label	Probability																	
741	0.90																	
887	0.08																	
668	0.02																	
Label	Probability																	
887	0.92																	
669	0.05																	
884	0.03																	
	 <table><caption>Original Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>299</td><td>0.80</td></tr><tr><td>138</td><td>0.15</td></tr><tr><td>9</td><td>0.05</td></tr></tbody></table>	Label	Probability	299	0.80	138	0.15	9	0.05	 <table><caption>Adversarial Probabilities</caption><thead><tr><th>Label</th><th>Probability</th></tr></thead><tbody><tr><td>138</td><td>0.90</td></tr><tr><td>84</td><td>0.08</td></tr><tr><td>134</td><td>0.02</td></tr></tbody></table>	Label	Probability	138	0.90	84	0.08	134	0.02
Label	Probability																	
299	0.80																	
138	0.15																	
9	0.05																	
Label	Probability																	
138	0.90																	
84	0.08																	
134	0.02																	

Adversarial image 會增加原本排第二的 label 的機率。

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。

我利用 `gaussian filter (sigma=2)` 的方式，去改變 `adversarial img`。做出來的誤判比例(`success rate`)從原本的 0.895 下降到了 0.655，這種防禦會讓圖片變得比較模糊，所以之前特別產生出來攻擊模型的部分就會變得不明顯。

	
Adversarial image	加了 gaussian filter 後的 image