# 2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全
# Day 24 – CW Attack

# 大綱

- CW 攻擊
  - 攻擊手法原理
  - 程式實作
- 結論

# CW 攻擊手法原理

https://arxiv.org/abs/1608.04644

- Towards Evaluating the Robustness of Neural Networks (2017)
- 有別於 FGSM 使用梯度作為攻擊原理，CW 偏向去解一個最佳化的問題來產生對抗式攻擊樣本，並且實現指哪打哪的攻擊方式

$$
\begin{aligned}
\text{minimize} \quad & \mathcal{D}(x, x+\delta) \\
\text{such that} \quad & C(x+\delta) = t \\
& x+\delta \in [0,1]^n
\end{aligned}
$$

where $x$ is fixed, and the goal is to find $\delta$ that minimizes $\mathcal{D}(x, x+\delta)$. That is, we want to find some small change $\delta$ that we can make to an image $x$ that will change its classification, but so that the result is still a valid image. Here $\mathcal{D}$ is some distance metric; for us, it will be either $L_0$, $L_2$, or $L_\infty$ as discussed earlier.

# CW 攻擊手法原理

- 從這邊開始數學就很多了，我也未必每個地方都看得懂

*A. Objective Function*

The above formulation is difficult for existing algorithms to solve directly, as the constraint $C(x+\delta) = t$ is highly non-linear. Therefore, we express it in a different form that is better suited for optimization. We define an objective function $f$ such that $C(x+\delta) = t$ if and only if $f(x+\delta) \leq 0$. There are many possible choices for $f$:

$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$
$$f_2(x') = (\max_{i \neq t}(F(x')_i) - F(x')_t)^+$$
$$f_3(x') = \text{softplus}(\max_{i \neq t}(F(x')_i) - F(x')_t) - \log(2)$$
$$f_4(x') = (0.5 - F(x')_t)^+$$
$$f_5(x') = -\log(2F(x')_t - 2)$$
$$\boxed{f_6(x') = (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+}$$
$$f_7(x') = \text{softplus}(\max_{i \neq t}(Z(x')_i) - Z(x')_t) - \log(2)$$

where $s$ is the correct classification, $(e)^+$ is short-hand for $\max(e, 0)$, $\text{softplus}(x) = \log(1 + \exp(x))$, and $\text{loss}_{F,s}(x)$ is the cross entropy loss for $x$.

# CW 攻擊手法原理

Now, instead of formulating the problem as

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) \\ \text{such that} \quad & f(x+\delta) \leq 0 \\ & x+\delta \in [0,1]^n \end{aligned}$$

we use the alternative formulation:

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) + c \cdot f(x+\delta) \\ \text{such that} \quad & x+\delta \in [0,1]^n \end{aligned}$$

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i.$$

Since $-1 \leq \tanh(w_i) \leq 1$, it follows that $0 \leq x_i + \delta_i \leq 1$, so the solution will automatically be valid. [8]
We can think of this approach as a smoothing of clipped gradient descent that eliminates the problem of getting stuck in extreme regions.

# CW 攻擊手法原理

- 最後參考 https://github.com/Harry24k/CW-pytorch 整理的式子比較簡潔乾淨
- 稍微分析一下，可以知道 K 在這邊就是個拘束器的腳色

$$\delta_i = \frac{1}{2}(tanh(w_i) + 1) - x_i$$

$$minmize\|\frac{1}{2}(tanh(w) + 1) - x\|_2^2 + c \cdot f(\frac{1}{2}(tanh(w) + 1))$$

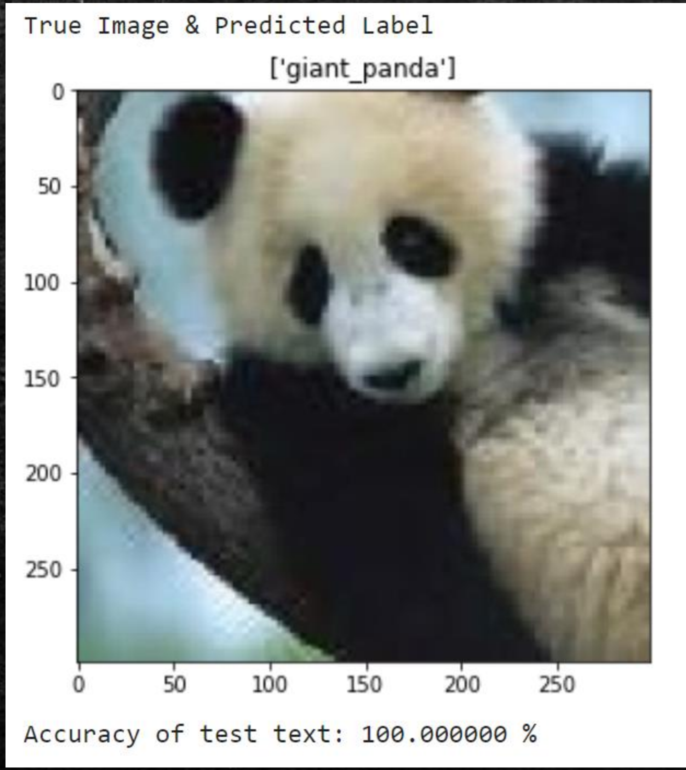
$$f(x') = max(max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$$

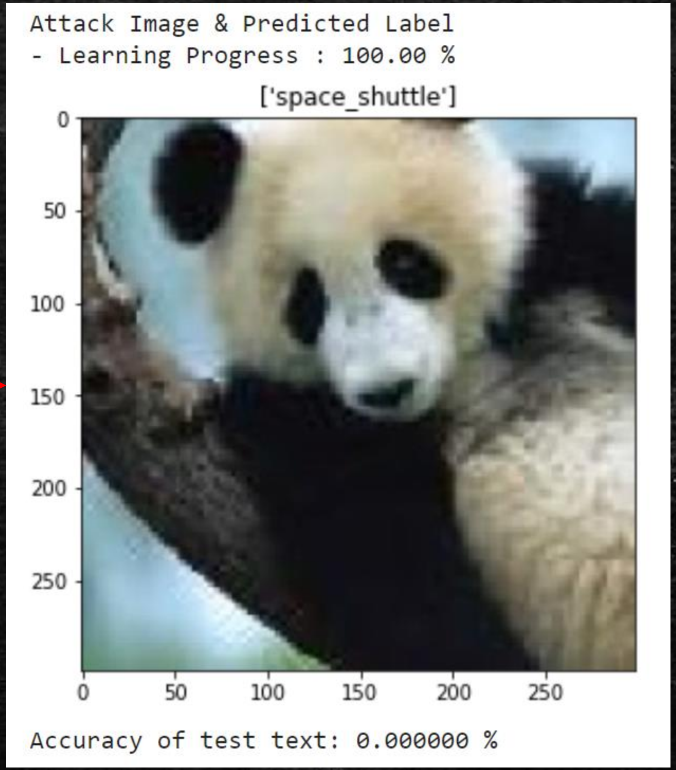
- Notation - $w$ : modifier, $t$ : class that $x'$ will be classified, $\kappa$ : confidence, $Z$ : classifier without last softmax

# 程式實作

https://github.com/Harry24k/CW-pytorch

- 只可惜這個專案是用 pytorch 完成的，但你也可以考慮人工翻成 tensorflow



# 結論

- CW 是一個蠻優秀的演算法，跟 FGSM 相比可以指定攻擊對象，增加了其攻擊的威脅程度
- 只是缺點在於定義出最佳化的問題時用到很多無法解釋的數學