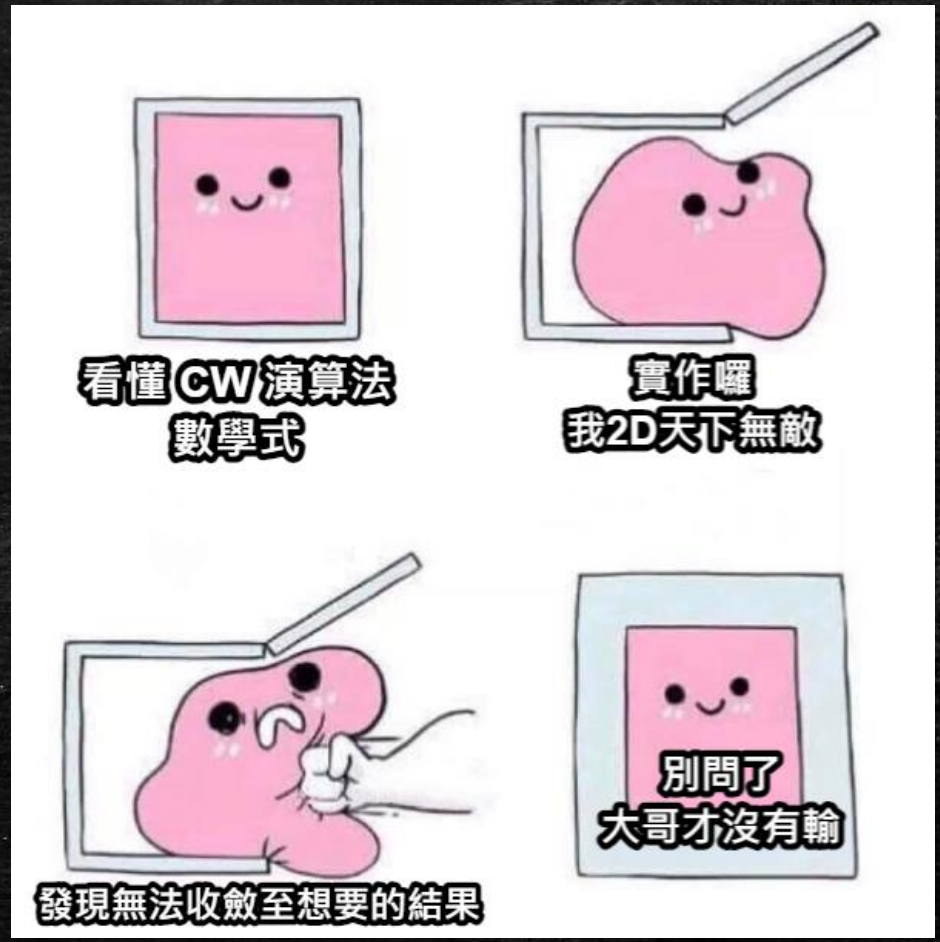
2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全
Day 25 – CW Attack

大綱

- CW 攻擊
  - 前情提要
  - 程式實作
- 結論



前情提要

- 最後參考 https://github.com/Harry24k/CW-pytorch 整理的式子比較簡潔乾淨

5. Adversarial Attack

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

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

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

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

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

程式實作

https://medium.com/@shuangzizuobh2/using-tensorflow-optimizers-to-minimize-a-simple-function-39681c5b6e72

- 工人智慧開始把它轉成 tensorflow

```
# 想辦法透過 CW 演算法針對載入的圖檔做出生成式對抗樣本，看能不能把這張圖變成 0~9

# 定義 f 函式
def f(input_x,attack_target,kappa=0) :
    outputs = load_model(input_x)
    # tf.eye 產生單位矩陣用的函式
    one_hot_label = (tf.eye(len(target_label[0])))[attack_target]
    i = tf.reduce_max((1 - one_hot_label)*outputs)
    j = (one_hot_label*outputs)[0][attack_target]
    return tf.reduce_max([i-j,-kappa])
```

```
for i in range(1000) :
    with tf.GradientTape() as tape:
        a = 1/2*(tf.math.tanh(w) + 1)
        loss1 = tf.keras.losses.MeanSquaredError(reduction='sum')(a,hack_data)
        loss2 = c*f(a,attack_target)
        loss = loss1 + loss2

    gradients = tape.gradient(loss, w)

    # 參考 https://medium.com/@shuangzizuobh2/using-tensorflow-optimizers-to-minimize-a-simple-function-39681c5b6e72
    tf.optimizers.Adam().apply_gradients(zip([gradients],[w]))
```

結論

- 感覺好像實作了 CW 演算法但是又缺了甚麼?
- 當初在這邊也卡很久，但後來檢查過論文發現有個地方我原本以為沒有影響而沒有照著實作，但事後發現還居然真的有影響
- 可以試著回去翻論文看看並對照程式碼，感受一下數學的魔力