2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全
Day 26 – CW Attack

大綱

- CW 攻擊
  - 前情提要
  - Softmax 函式
  - 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

細節藏在細節裡

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

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

- 論文有說，式子有寫，可惜我看不見

In this paper we focus on neural networks used as an m-class classifier. The output of the network is computed using the softmax function, which ensures that the output vector y satisfies $0 \leq y_i \leq 1$ and $y_1 + \cdots + y_m = 1$. The output vector y is thus treated as a probability distribution, i.e., y_i is treated as the probability that input x has class i. The classifier assigns the label $C(x) = \arg\max_i F(x)_i$ to the input x. Let $C^*(x)$ be the correct label of x. The inputs to the softmax function are called *logits*.

We use the notation from Papernot et al. [39]: define F to be the full neural network including the softmax function, $Z(x) = z$ to be the output of all layers except the softmax (so z are the logits), and

$$F(x) = \text{softmax}(Z(x)) = y.$$

Softmax 函式

https://zh.wikipedia.org/zh-tw/Softmax%E5%87%BD%E6%95%B0

- 當初沒想到是因為我覺得 Softmax 只是單純把數值壓縮到另一個空間範圍，數字彼此之間的大小關係不變，應該不影響機器學習的訓練過程

Softmax函式 [編輯]

17 種語言

條目 討論 汉 漢 臺灣正體 閱讀 編輯 檢視歷史 工具

在數學，尤其是機率論和相關領域中，**Softmax函式**，或稱**歸一化指數函式**[1]:198，是邏輯斯諦函式的一種推廣。它能將一個含任意實數的K維向量 $\mathbf{z}$ 「壓縮」到另一個K維實向量 $\sigma(\mathbf{z})$ 中，使得每一個元素的範圍都在$(0, 1)$之間，並且所有元素的和為1(也可視為一個 (k-1)維的hyperplane或subspace)。該函式的形式通常按下面的式子給出：

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

```python
import math
z = [1.0, 2.0, 3.0, 4.0, 1.0, 2.0, 3.0]
z_exp = [math.exp(i) for i in z]
print(z_exp)  # Result: [2.72, 7.39, 20.09, 54.6, 2.72, 7.39, 20.09]
sum_z_exp = sum(z_exp)
print(sum_z_exp)  # Result: 114.98
softmax = [round(i / sum_z_exp, 3) for i in z_exp]
print(softmax)  # Result: [0.024, 0.064, 0.175, 0.475, 0.024, 0.064, 0.175]
```

程式實作

https://stackoverflow.com/questions/78413637/how-to-get-the-output-of-the-model-before-the-softmax-without-chaning-the-mode

- 如何拿掉最後一層的 softmax?

- 宣告一個結構一模一樣的模型，然後最後一層定義激勵函數時使用 linear ，然後再把原本模型的參數載入

程式實作

- 其實也就那麼關鍵的一行

```
# 宣告一個結構一樣的模組，但是最後一層的激勵函數改掉
new_model = Sequential()
new_model.add(Dense(128, input_dim=784, activation=tf.nn.relu))
new_model.add(Dense(10, activation='linear'))
new_model.load_weights('mnist_basic_model.h5')
```

結論

- 想不到一個小小的 softmax 居然影響了整個模型宇宙
- 看到這邊只能說沒辦法數學太爛了無法解釋，不過看起來關於這件事情論文有給 reference，所以有興趣的人可以繼續深入研究看看