

TAICA_生成式 AI：文字與圖像生成的原理與實務

第三週作業 03/04: (主題二) 研究 GAN 背後原理，試著用自己的方式解釋 Cross Entropy、KL divergence

臺師大 613K0019C AI 所碩一 劉思廷

簡介

生成式對抗網絡 (Generative Adversarial Networks, GAN) 是由 Ian Goodfellow 等人於 2014 年提出。GAN 的核心思想是透過生成器 (Generator) 和判別器 (Discriminator) 的對抗性訓練來生成逼真的數據。本作業將簡單探討 GAN 的運作原理，並聚焦於損失函數中常用的 Cross Entropy (交叉熵) 和 KL Divergence (KL 散度)，以自己的方式解釋它們的定義及關係。此外，我將透過實際計算、程式實驗來進一步闡述這兩個函數的差別。

一、GAN 的基本原理

- 生成器 (Generator)：從隨機噪聲 z 生成數據 $G(z)$ ，目標是使生成的數據分布 p_g 盡可能接近真實數據分布 p_{data} 。
- 判別器 (Discriminator)：接收真實數據 $x \sim p_{data}$ 和生成數據 $G(z)$ ，並判斷輸入數據是真實的還是生成的，輸出一個機率值 $D(x)$ 。

訓練過程中，生成器試圖生成逼真的數據欺騙判別器，而判別器則努力正確區分真假數據，這形成了一個對抗遊戲，其目標函數為：

$$\min G \max D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log(D(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]$$

當訓練達到平衡時， $p_g \approx p_{data}$ ，判別器無法區分真假數據（即 $D(x) \approx 0.5$ ）。損失函數在這一過程中非常重要，而 Cross Entropy 是 GAN 中常用的損失函數形式，KL Divergence 則在理論分析中扮演重要角色。

二、Cross Entropy 和 KL Divergence 的定義與解釋

在介紹 Cross Entropy 之前，我想先介紹資訊量 (Information Content, $I(x)$) 以及 Entropy (熵)。資訊量被定義為：

$$I(x) = -\log_2 P(x)$$

其中， $P(x)$ 表示事件 x 發生的機率。若某事件發生的機率是 $1/8$ ，則此事件的資訊量為 $I(x) = -\log_2 \frac{1}{8} = 3$ (bits)。這表示描述此事件至少需要 3 個 bit。我們也可以觀察到，當事件發生的機率越小，對應的資訊量將會越大。直觀上來說，越是不容

易發生的事發生了，將帶給我們更大的資訊量，因此，資訊量的大小也可以看作是事件帶給我們的驚訝程度。針對整個系統的資訊量，則使用 Entropy 來描述：

$$H(P) = - \sum_x P(x) \log P(x) = \mathbb{E}_{x \sim p} I(x)$$

Entropy 代表系統資訊量的期望值 (平均資訊量)，表示對該系統進行最佳編碼時，每個樣本平均需要的 bit 數。

1. Cross Entropy (交叉熵)

Cross Entropy 用於衡量兩個機率分佈之間的差異，在機器學習中常用作損失函數。假設 $P(x)$ 是真實分佈， $Q(x)$ 是預測分佈，則 Cross Entropy 定義為：

$$H(P, Q) = \mathbb{E}_{x \sim p} [-\log Q(x)] \geq \mathbb{E}_{x \sim p} [-\log P(x)] = H(P, P) = H(P)$$

直觀上來說，Cross Entropy 表示如果我們使用 Q 來描述 P 的數據，平均需要多少資訊量。當 Q 越接近 P 時，Cross Entropy 越小，且當 $Q = P$ 時， $H(P, Q) = H(P)$ 有最小的 Cross Entropy。因此，Cross Entropy 除了衡量預測分佈以及真實分佈之間的差距，還包含了真實分佈的本身亂度。

2. KL Divergence (KL 散度)

KL Divergence (Kullback-Leibler Divergence) 同樣用於衡量兩個機率分佈的差異，但它是非對稱的。其定義為：

$$\begin{aligned} \mathcal{D}_{KL}(P||Q) &= - \sum_x P(x) \log\left(\frac{Q(x)}{P(x)}\right) = \mathbb{E}_{x \sim p} [-\log\left(\frac{Q(x)}{P(x)}\right)] \\ &= \mathbb{E}_{x \sim p} [-\log Q(x)] - \mathbb{E}_{x \sim p} [-\log P(x)] = H(P, Q) - H(P) \end{aligned}$$

這表示 KL Divergence 等於模型預測的 Cross Entropy 減去真實分佈的 Entropy，反映了模型實際上兩分佈之間的差距。且當 $Q = P$ 時， $\mathcal{D}_{KL}(P||Q) = 0$ ，表示模型已完全匹配真實分佈，無法再進一步優化。

3. Cross Entropy 與 KL Divergence 的關係

由於真實分佈的 Entropy $H(P)$ 是固定常數，最小化 Cross Entropy $H(P, Q)$ 等價於最小化 KL Divergence $\mathcal{D}_{KL}(P||Q)$ 。因此，兩者皆可作為機器學習中的損失函數，目的是讓模型的預測分佈 Q 盡可能接近真實分佈 P 。而 Cross Entropy 主要用於衡量模型預測分佈與真實分佈的總亂度；KL Divergence 則是用於反映模型的預測結果與真實分佈之間的差距，也可視為模型還能進步的空間。

三、程式 Demo

Colab 連結:

https://colab.research.google.com/drive/19Mn9NhI4prasko64rzf9nV7N4al0jg_V?usp=sharing

1. 套件引入

```
""" 啟用內嵌繪圖模式，讓 matplotlib 圖表直接顯示在輸出區塊 """
%matplotlib inline

""" 固定四行引用套件 """
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

2. 定義 Cross Entropy 和 KL Divergence 函數

```
""" 定義 Cross Entropy 和 KL Divergence 函數 """
# 參數: "P", "Q" 分別為 "真實" 以及 "生成" 的數據
# 返回值: CE_value or KL_value (浮點數)
def cross_entropy(P, Q):
    return -sum(p * np.log(q) for p, q in zip(P, Q))

def kl_divergence(P, Q):
    return sum(p * np.log(p/q) for p, q in zip(P, Q))
```

3. 定義生成隨機列表的函數

```
""" 定義函數: generate_normalized_list """
# 功能: 生成一個長度與 "P" 相同的隨機數列表，並將其正規化使總和為 1
# 參數: "length" 為欲生成的隨機數個數
# 返回值: "normalized_list": 正規化後的隨機數列表，元素總和為 1
def generate_normalized_list(length):
    random_numbers = np.random.rand(length) # 生成長度與 "P" 相同的隨機數 (範圍在0-1之間)
    normalized_list = random_numbers / np.sum(random_numbers) # 正規化
    return normalized_list
```

4. 定義數據

```
""" 定義 Demo 數據 """
# 建立固定數據 P，並生成 N 個相同長度的數據
N = 10
P = [0.1, 0.2, 0.3, 0.4] # 固定數據
Q_list = [generate_normalized_list(len(P)) for _ in range(N)] # 10 個隨機生成數據
print("P:", P)
for i in range(N):
    print(f"Q[i+1]: {Q_list[i]}") # 印出生成的隨機數列表
```

5. 計算並輸出結果

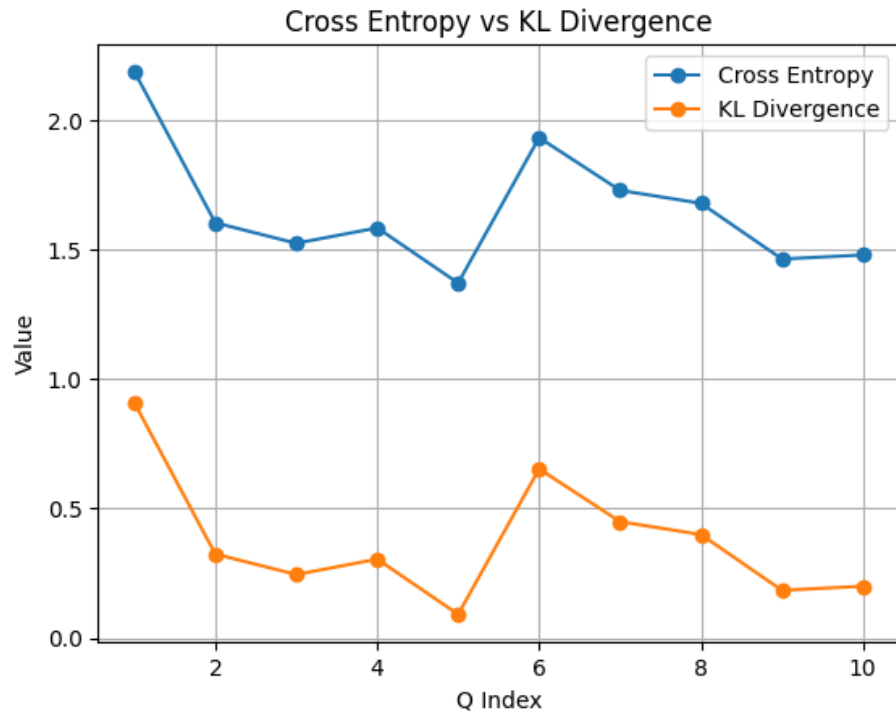
```
""" 計算結果 """
ce_values = [cross_entropy(P, Q) for Q in Q_list]
kl_values = [kl_divergence(P, Q) for Q in Q_list]

""" 輸出結果 """
for i, (ce, kl) in enumerate(zip(ce_values, kl_values)):
    print(f"Q{i+1}: Cross Entropy = {ce:.4f}, KL Divergence = {kl:.4f}")

P: [0.1, 0.2, 0.3, 0.4]
Q1: [0.5101243 0.146313 0.31238896 0.03117374] Q1: Cross Entropy = 2.1880, KL Divergence = 0.9082
Q2: [0.33838008 0.16273734 0.07120858 0.42767399] Q2: Cross Entropy = 1.6039, KL Divergence = 0.3240
Q3: [0.32688522 0.23702942 0.10110463 0.33498072] Q3: Cross Entropy = 1.5247, KL Divergence = 0.2448
Q4: [0.02412828 0.398413 0.08324485 0.49421387] Q4: Cross Entropy = 1.5842, KL Divergence = 0.3043
Q5: [0.08890244 0.0710507 0.36040805 0.4796388 ] Q5: Cross Entropy = 1.3709, KL Divergence = 0.0911
Q6: [0.43421764 0.19248258 0.3207874 0.05251238] Q6: Cross Entropy = 1.9327, KL Divergence = 0.6529
Q7: [0.42255273 0.02616347 0.30103667 0.25024713] Q7: Cross Entropy = 1.7291, KL Divergence = 0.4492
Q8: [0.1938184 0.01329389 0.40811308 0.38477463] Q8: Cross Entropy = 1.6791, KL Divergence = 0.3992
Q9: [0.0190635 0.09744091 0.52188875 0.36160684] Q9: Cross Entropy = 1.4637, KL Divergence = 0.1838
Q10: [0.31807745 0.15836441 0.3358525 0.18770564] Q10: Cross Entropy = 1.4796, KL Divergence = 0.1997
```

6. 視覺化結果

```
""" 視覺化結果 """
plt.plot(range(1, N+1), ce_values, label="Cross Entropy", marker='o')
plt.plot(range(1, N+1), kl_values, label="KL Divergence", marker='o')
plt.xlabel("Q Index")
plt.ylabel("Value")
plt.title("Cross Entropy vs KL Divergence")
plt.legend()
plt.grid(True)
```



結論

生成對抗式網絡 (GAN) 透過生成器與判別器的對抗性訓練，提供了一種創新的生成模型框架，其核心在於生成逼近真實數據分佈的數據。在本作業中，探討了 GAN 的運作原理，並聚焦於損失函數中常用的 Cross Entropy 和 KL Divergence。透過資訊論的視角，我將 Cross Entropy 解釋為使用預測分佈 Q 描述真實分佈 P 時所需的平均資訊量，而 KL Divergence 則被視為以 Q 描述 P 所需的額外資訊量。也就是說，Cross Entropy 可表示預測分佈與真實分佈之間的總亂度，而 KL Divergence 則可直接反映兩分佈之間的差距。兩者的關係進一步揭示了它們在優化過程中的互補性：最小化 Cross Entropy 等價於最小化 KL Divergence，從而推動 Q 趨近於 P 。

程式實驗透過定義 Cross Entropy 和 KL Divergence 的計算函數，並生成固定數據 P 以及測試數據 Q 進行比較，我們可以觀察到，當隨機生成的測試數據 Q 與固定數據 P 相差較大時，兩者的值均會增加，而 KL Divergence 更直接反映了分佈間的差異，Cross Entropy 則因包含真實分佈本身的亂度，而具有更高的基準線。在應用層面上，因為 Cross Entropy 的計算較為簡單，而 KL Divergence 則包含了分數項，因此 Cross Entropy 較常被選用為訓練的損失函數。