

機器學習 Machine Learning LAB 4

系所：電機所

學號：609415074

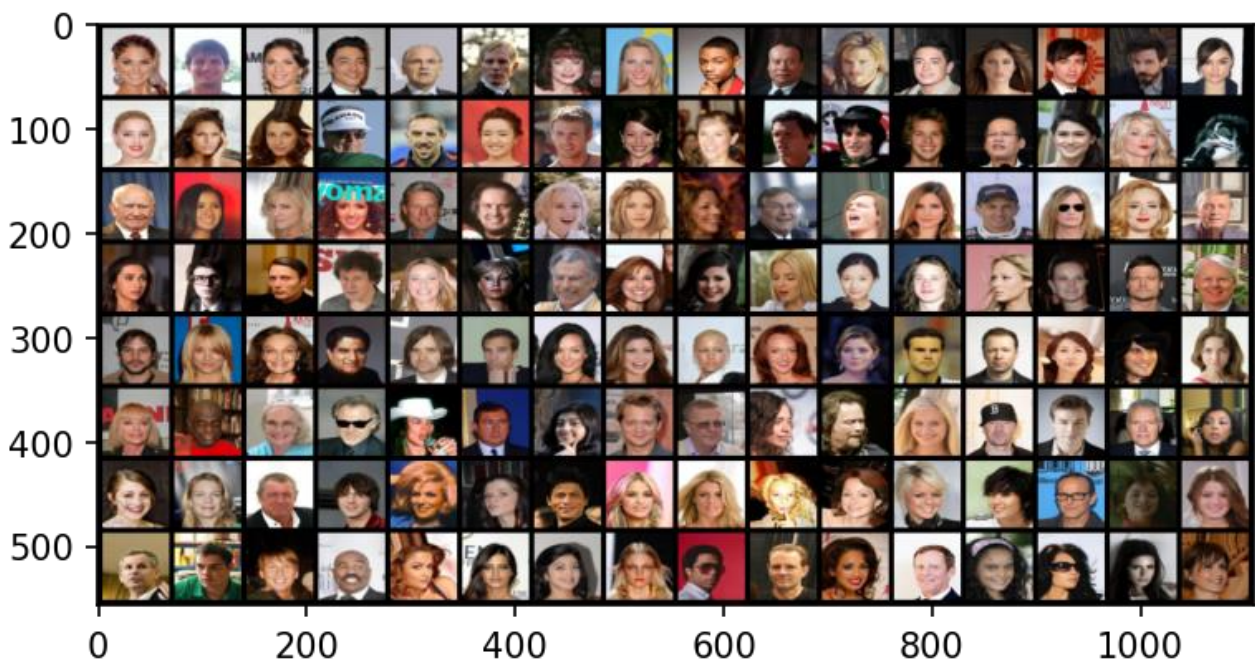
姓名：蔡承宏

1. colab 的網址(colab URL)：

網址：

<https://colab.research.google.com/drive/1GOMAJdMmiONHOGZj2e1asy5c9NoJMj4U?authuser=1#scrollTo=6yMlbSnA2mzL>

2. 實驗結果(Execution results)：

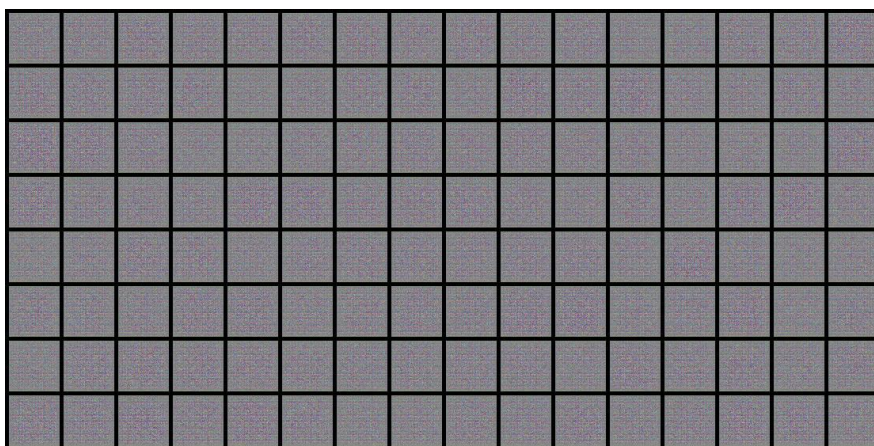


圖一、結果圖片

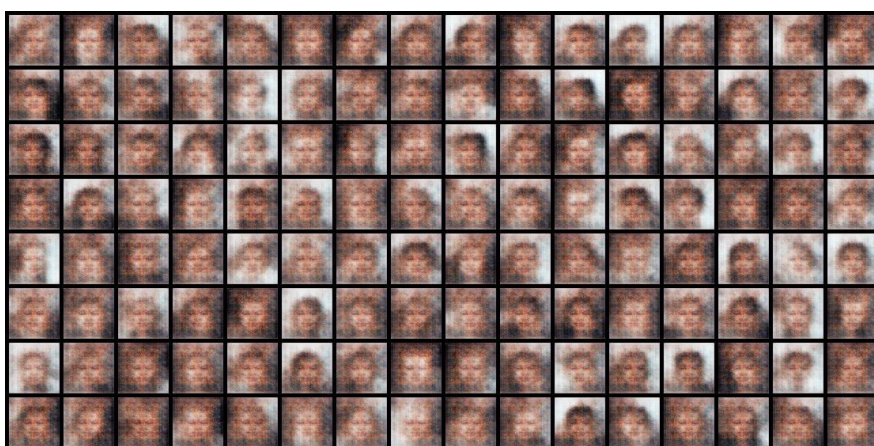
3. QA 問答(Answers of each question)：

Q1. Compare the visualization images after step 0, 200, 1000, 2500, 5000.

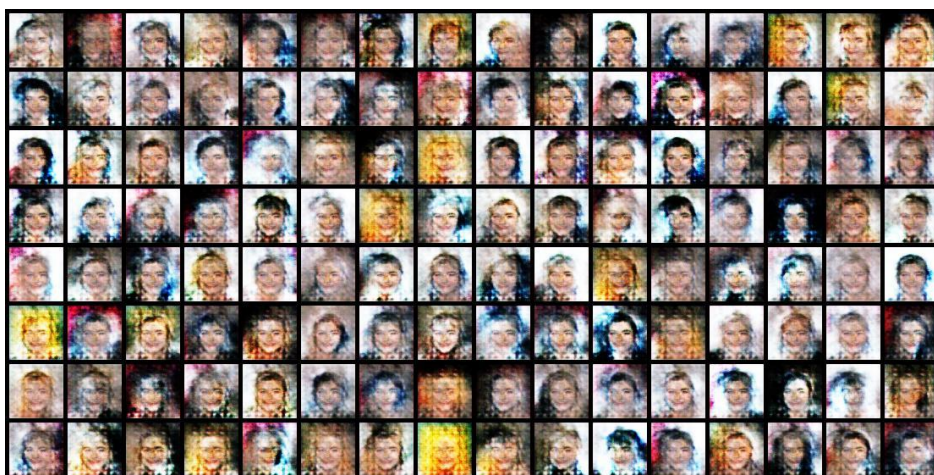
A1. 首先貼上 0 / 200 / 1000 / 2500 / 5000 的結果圖



圖二、0 step



圖三、200 step



圖四、1000 step



圖五、2500 step



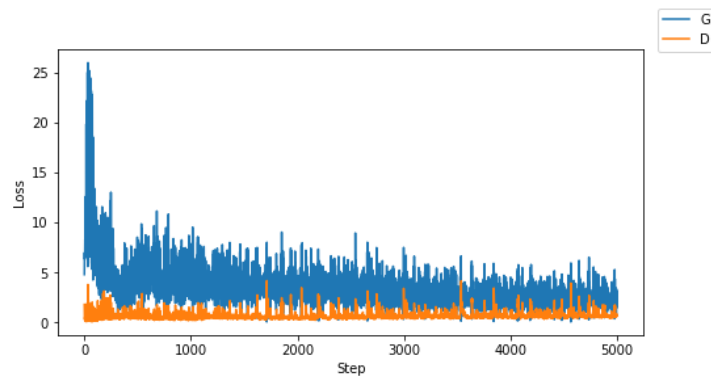
圖六、5000 step

Step	Describe Image
0	雜訊(Noise)，看不懂內容。
200	色彩較為豐富，有人影。
1000	鮮豔的色彩，可以看出人的輪廓，但模糊。
2500	圖片較清晰，但大概是影片的 144p→240p 的感覺。
5000	圖片更加清晰，從 240p→360p 了。

表 1.各 Step 的比較

Q2. Why can't we see the typical loss decreasing in generator/discriminator's loss curve?

A2. 首先放上 Loss curve 的圖片



圖七、Loss curve

GAN 的 Loss curve 跟 CNN 的完全不一樣，CNN 的是一條慢慢往下的線，而 GAN 前面的部分可以看到落差非常大，那是因為 GAN 的 Generator 和 Discriminator 是一種互相競爭、互相拔河的神經網路，所以會造成此現象。

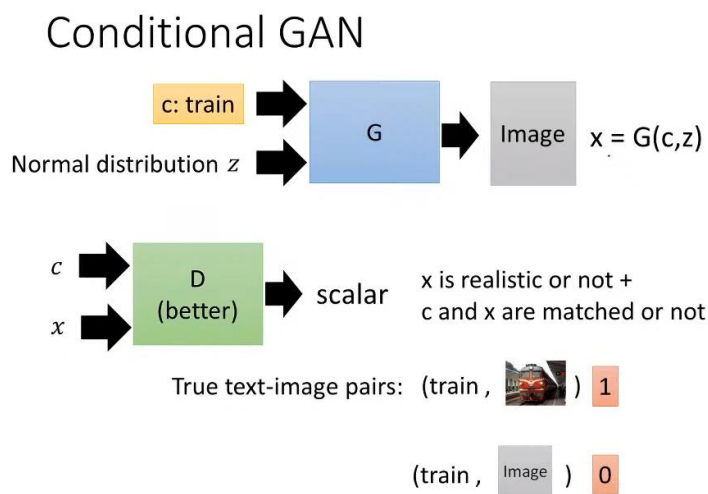
Q3. How to generate fake images after training? Please use our trained generator to generate fake images using following distributions:

1. $N(0,1)$: same as our training, the standard normal distribution
2. $N(-10,1)$: normal distribution with mean=-10, std=1
3. $U(0,1)$: uniform distribution in range $[0, 1]$

Each distribution should generate 128 fake images (a mini-batch). The fake images should be embedded in your notebook. You need to write some code in this problem.

(Hint: The last cell of this notebook is the solution to Q3.1)

A3.生成影像的流程圖如下



圖八、GAN 生成流程圖

我們透過一開始給的條件(C)和雜訊(Z)輸入到 Generator 裡面，即可生成出一張影像，而這張影像會再由 Discriminator 進行鑑定，判斷其是否為假的照片，假如是假的照片則會告訴 Generator 少了什麼，如:色彩、人物、解析度…等，藉由兩者不斷的競爭來增強模型生成出來的圖片。

A3-1. Image of Normal distribution(0,1) :



圖九、Normal distribution(0,1)的圖片

A3-2. Image of Normal distribution(-10,1) :



圖十、Normal distribution(-10,1)的圖片

A3-3 Image of Normal distribution(0,1) :



圖十一、Uniform distribution(0,1)的圖片

這題其實我不是很確定，但是查了網路上的資料說 Uniform Distribution 是用 torch.rand 來執行，而執行出來的結果如上圖。我參考的參考的資料如下圖

均勻分佈

`torch.rand(*sizes, out=None) → Tensor`

返回一個張量，包含了從區間[0, 1)的均勻分佈中抽取的一組隨機數。張量的形狀由參數sizes定義。

參數:

- sizes (int...) - 整數序列，定義了輸出張量的形狀
- out (Tensor, optional) - 結果張量

例子:

```
torch.rand(2, 3)
0.0836 0.6151 0.6958
0.6998 0.2560 0.0139
[torch.FloatTensor of size 2x3]
```

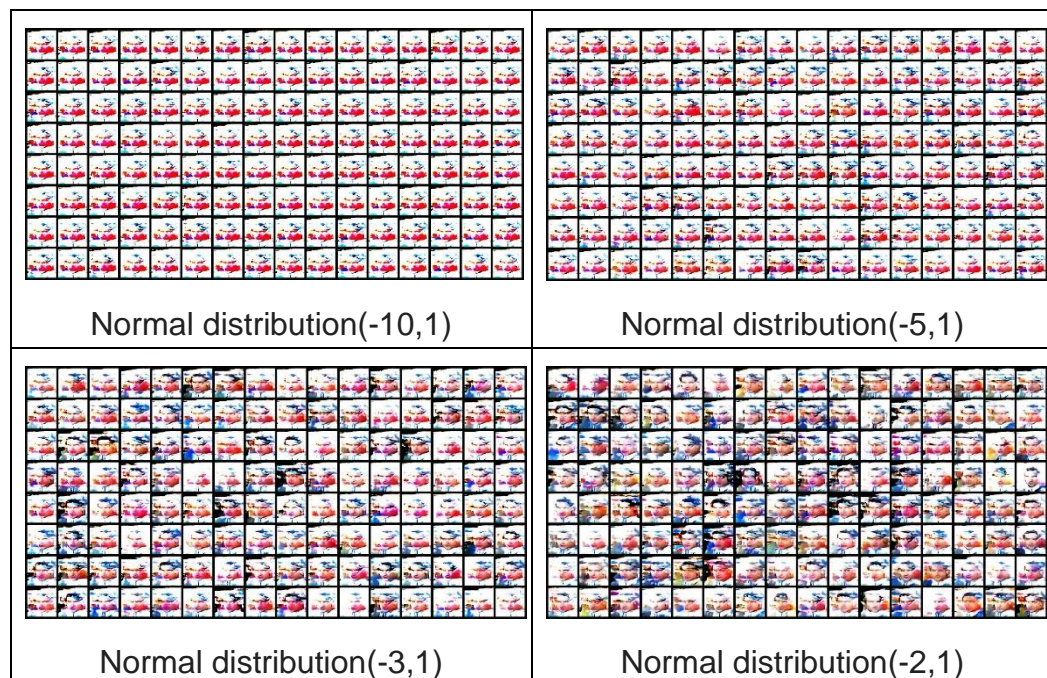
圖十二、Uniform distribution 參考圖片

Q4. Why the result of Q3.2 looks so weird, i.e., Why our generator fails at generating images using latent vector following distribution $\mathbf{N}(-10,1)$?

A4. 通常常態分佈訊號輸入的範圍越大，則輸出的質量則會較低。

這邊我額外又多做了四次數據驗證分別是 Normal distribution(-5,1)、Normal distribution(-3,1)、Normal distribution(-2,1)、Normal distribution(-1,1)。

比較表如下:



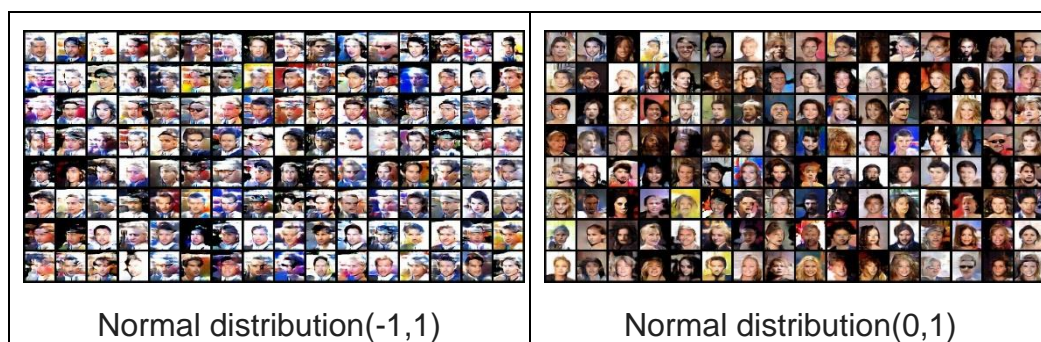


表 2. 常態分佈比較圖

Q5. Why the result images of Q3.3 looks so similar to each other?

A5. 因為 Q3.3 使用的是均勻分布，均勻分布的函式為 $f(x) = \frac{1}{b-a}$ ，因此能夠輸入的訊號差異較大，而常態分佈的函式為 $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ，因此輸入的訊號基本上都會落在中心點，其比較圖如下。

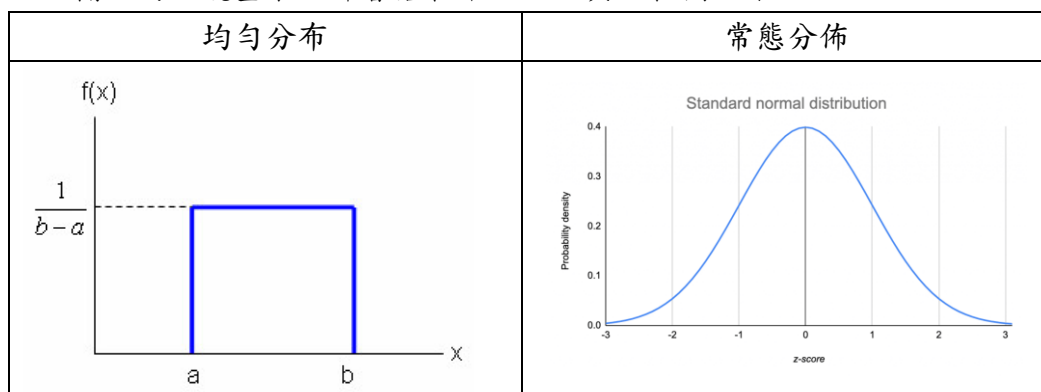


表 3. 均勻分布跟常態分佈比較圖

4. 結論(Conclusion)：

這次的 Lab 十分有趣，可以使用 GAN 來生成各種不同的圖片，而且只要將 GAN 模型重新訓練一次，產生出來的圖片又將會是不一樣的圖片，我在做 LAB 的過程中，有不小把 colab 關閉，導致需要重新訓練，因此才發現這件事，不過關於最後面的均勻分布跟常態分佈其實還不是很了解，在這部分還需要助教或老師幫忙解答，而最後生成出來的 fake image，我認為完成度相當高，完全看不出來這是假的影像，ML 真的太強大了 > /// <。

5. 討論(Discussion)：

這次做 Lab 的過程中基本上不太會有什麼問題，只有均勻分佈不知道該怎麼改寫而已，不過這部分只要上網尋找資料即可找到答案，關於最好奇的一點當然還是在結論所說的，均勻分佈跟常態分佈的差別，這部分查了很多資料，有看到一篇文章有講說有部分的 GAN 不適合使用均勻分佈，但是內文並沒有詳細說明，因此希望這次 LAB 公布答案時，老師跟助教也能說明這個問題，也很感謝助教寫了那麼複雜的 Code，GAN 的程式

碼跟 CNN 真的差距很大，很難看懂，因此未來會更加努力的去研究!!