ADLxMLDS HW4: Generative Adversarial Networks

學號：R06922030　系級：　資工碩一　姓名：傅敏桓

I.　模型描述

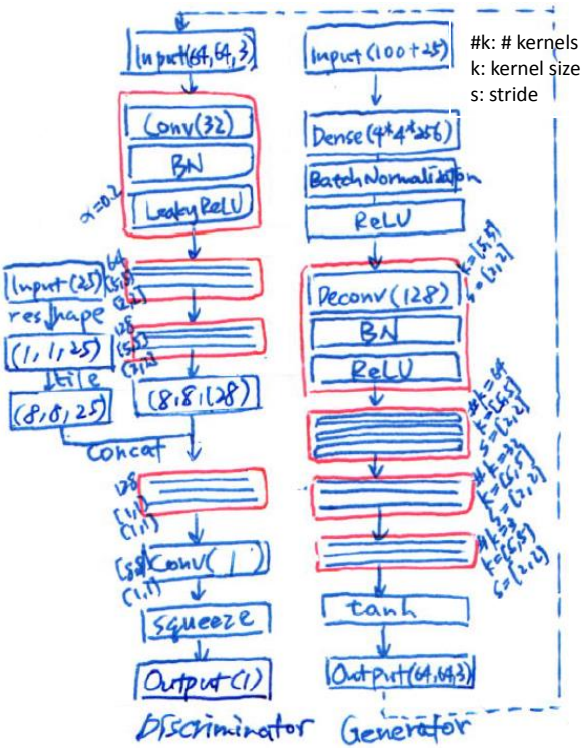
本次作業透過 cDCGAN（條件式的深度卷積生成對抗網路）實作動漫角色的自動生成模型。關於各種 GAN 模型的基本概念在此不多做說明，本篇報告的重點會放在模型實作的細節。

1.　資料前處理

實際測試時只會給髮色和瞳色的條件限制，所以把缺兩種標記的影像捨棄。若影像只有髮色或只有瞳色的標記，給缺的部分補上 unk 標記。如果同種類的標記出現兩個以上，只取最前面的標記。實際資料集包含的標記種類加入 unk 標記後為 13 種髮色、12 種瞳色（異色瞳不納入考量），分別以 one-hot 編碼表示特徵、串接（concatenate）後得到 25 維的條件向量。影像的部分會先將讀入的數據壓縮至（64, 64, 3）。

2.　模型架構

GAN 的模型包含生成網路(Generator, G)和鑑別網路(Discriminator, D)兩個部分，生成網路負責產生影像、鑑別網路負責辨別影像是否為生成網路生成之影像。本次作業中依照［1］所述之架構實作含深層卷積神經網路的 DCGAN 模型，模型架構如右圖所示。



生成網路接收 100 維的雜訊和 25 維的條件向量作為輸入，先將兩個向量串接後先通過全連接層攤開到高維空間，再通過數層轉置卷積層，最後以 tanh 將輸出值壓到(0, 1)之間得到影像輸出。由於需要將低維度的雜訊-條件向量攤開到（64, 64, 3）的高維空間中，我們可以透過轉置卷積(transpose convolution,或稱 deconvolution）的方法來實作數據的 upsampling，並在層與層之間實施批標準化。習慣上設計生成網路時讓轉置卷積核的數目逐層減半可以得到不錯的效果，這裡的生成網路使用的轉置卷積層的核分別是 128、64、32、再對應到 RGB 的 3 個通道。

鑑別網路以大小（64, 64, 3)的影像資料為輸入，通過數層卷積層後先得到（8, 8, 128）的特徵張量，比照［1］的作法將條件向量堆疊後串接在特徵向量的深度軸上（如此在進行判別時可以同時將影像的強度資訊和條件向量納入考慮），最後再通過幾層卷積層後，可以得到表示判別結果的 1 維輸出（之後使用的目標函數已經包含 sigmoid，這裡就直接輸出結果）。卷積網路就是用比較典型的設計，越靠近輸出的卷積層的核就越多，只是這裡並沒有將卷積層的輸出攤開通過全連接層、而是直接利用卷積層降採樣到 1 維的輸出；生成網路的設計也可以大致想像成鑑別網路的逆向操作，兩者架構有相似之處。

3. 訓練方法與設定：

根據上課的投影片，D 的目標函數可以寫成以下的形式，並且希望可以最大化目標函數：

$$\tilde{V}_D = \frac{1}{m}\Sigma_{i=1}^{m}\log D\left(x^i\right) + \frac{1}{m}\Sigma_{i=1}^{m}\log\left(1 - D\left(\tilde{x}^i\right)\right),$$

其中 m 代表每次取樣的數量，$\tilde{x}^i$是生成器產生的資料。另外要加入條件判斷的部分，D 的目標函數還需要包含錯誤的影像標記配對資訊；加入兩個代表錯誤配對的項後，目標函數變成：

$$\frac{1}{m}\Sigma_{i=1}^{m}\log D\left(x^i, h^i\right) + \frac{1}{3}\left(\frac{1}{m}\Sigma_{i=1}^{m}\log\left(1 - D\left(\tilde{x}^i, h^i\right)\right) + \frac{1}{m}\Sigma_{i=1}^{m}\log\left(1 - D\left(\hat{x}^i, h^i\right)\right) + \frac{1}{m}\Sigma_{i=1}^{m}\log\left(1 - D\left(x^i, \hat{h}^i\right)\right)\right),$$

其中h^i是x^i對應的標記、$\hat{x}^i$和$\hat{h}^i$分別代表對應錯誤的影像和標記，另外因為不希望真實影像的影響被稀釋，後面三項各只取 1/3。G 的目標函數仿照投影片的公式，不再加入條件資訊：$\tilde{V}_G = \frac{1}{m}\Sigma_{i=1}^{m}\log\left(1 - D\left(\tilde{x}^i\right)\right)$，希望 G 的目標函數可以越小越好。訓練 G 時會固定 D 的參數，反之亦然。

實作上可以透過 sigmoid_cross_entropy_with_logits 改寫上述的目標函數；根據 tensorflow 文件的定義，給定輸入 x 和目標 z，sigmoid_cross_entropy_with_logits 函數之回傳值為 z * -log(sigmoid(x)) + (1 - z) * -log(1 - sigmoid(x))，最大化 $D\left(x^i, h^i\right)$ 的項可以改寫成最小化 $D\left(x^i, h^i\right)$ 和 1 之間的 sigmoid 交叉熵，然後就可以用 tennsorflow 提供的優化器來進行參數更新，其餘皆可依此類推。以下條列模型的訓練過程的其他細節：

- 交互更新頻率 G : D = 1 : 1
- 以常態分佈(μ=0.0, σ=0.02)初始化參數
- Adam 優化器：學習率 2e-4、beta1=0.5、beta2=0.9
- 批(每次取樣)大小: 64

II. 改進方法

0. 多跑幾次訓練

在訓練 DCGAN 的時候發現，即便用相同的參數設定，得到的結果差異可能也很大，推測除了參數初始化的影響之外，也可能和訓練初期採樣到的資料有關，其差異可能會影響模型訓練的走向，所以多做幾次實驗取比較好的模型來用。以下是在完全相同的實驗設定下得到的兩次實驗的結果，固定輸入的雜訊和條件向量，仍可以看出兩者有明顯的差異。以下在探討其他變因的影響時，用來比較的圖像可能也因此不同。

1. 數據增強 (Data Augmentation)

前處理後的資料只有 17000 多張，所以試著在不改變影像的重要特徵下對影像作一些操作，希望可以增加訓練資料的變化性。對於每次取樣出來的影像資料，都隨機挑選部分影像進行左右翻轉、以及左右的微幅旋轉 (有可能兩者同時發生)。實驗的結果如下所示，左邊是直接用原始圖片、右邊是有處理過的結果，但在這次實驗中並沒有看出顯著的差異。

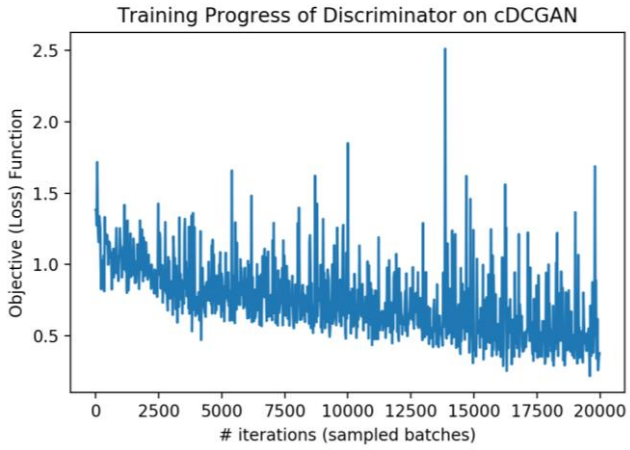
2. 修改神經網路架構

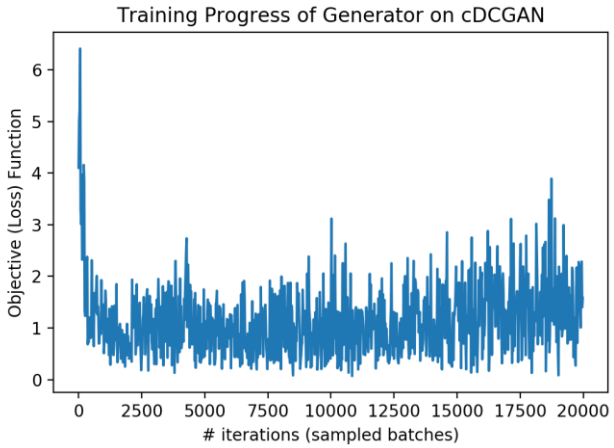
不同的生成網路/鑑別網路架構也應該會影響實驗結果，因此稍微嘗試了不同架構的神經網路，差異在於卷積層的層數不同，分別是在原本的模型架構上增加或是減少一層卷積層。少一層的模型如預期產生出較模糊的結果，但多一層的也沒有顯著的進步，似乎還是原本的架構比較穩定。實驗的結果如下所示，左邊是比原本少一層的結果、右邊是多一層的結果。

3. WGAN

這次也嘗試實作 weight clipping 的 WGAN，按照投影片的方法去除目標函數的 log 和 D 的 sigmoid，並在每次更新參數後進行 weight clipping(範圍在±0.01 之間)，但結果看起來並沒有比較好，可能是因為上面實作的 DCGAN 架構還可以，而 WGAN 的優勢主要在於減少模型架構的影響，以致結果沒有什麼進步；且修改 cDCGAN 的目標函數後模型一直沒辦法收斂，目前還找不出原因在哪裡，因此最後沒有採用 WGAN。以下是改用 WGAN 實驗的結果，除了目標函數和 weight clipping 以外沒有更動其他實驗設定。

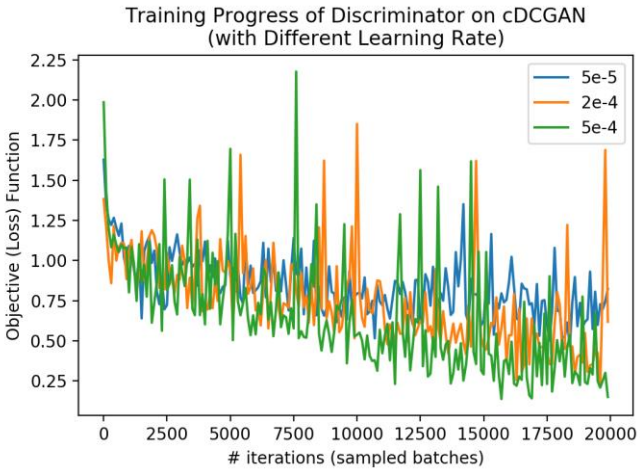
III. 實驗設定與觀察

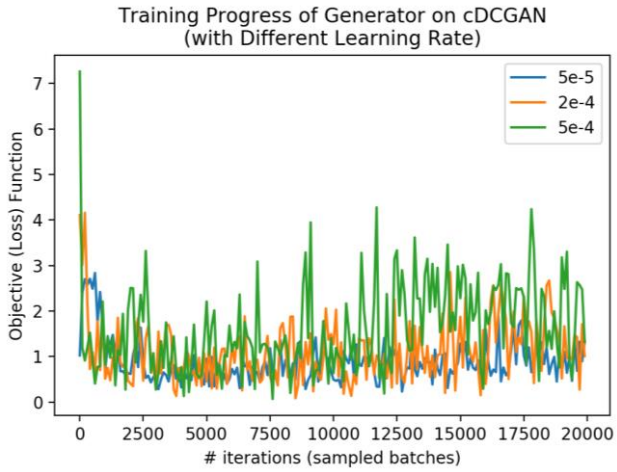


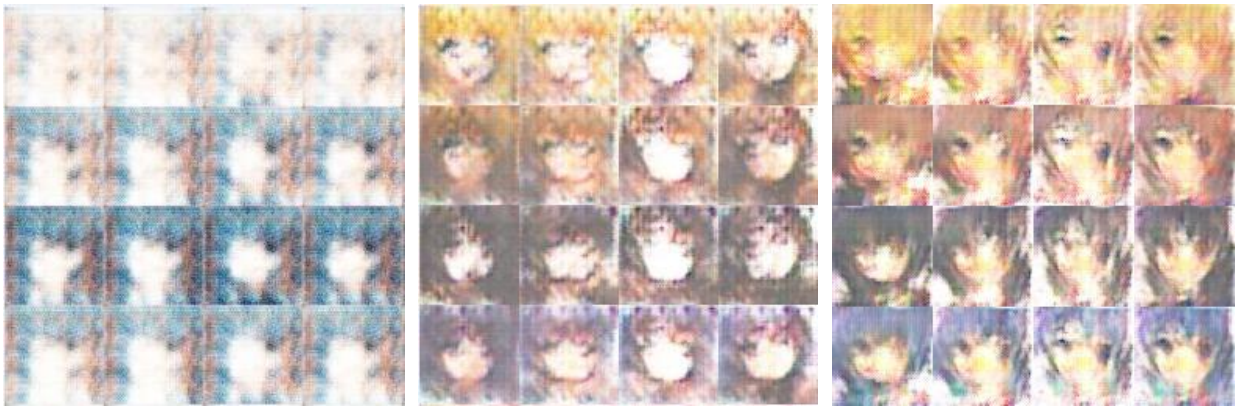


實驗設定如上所述，訓練過程如上圖所示。以下列出幾種不同參數設定的實驗結果比較。

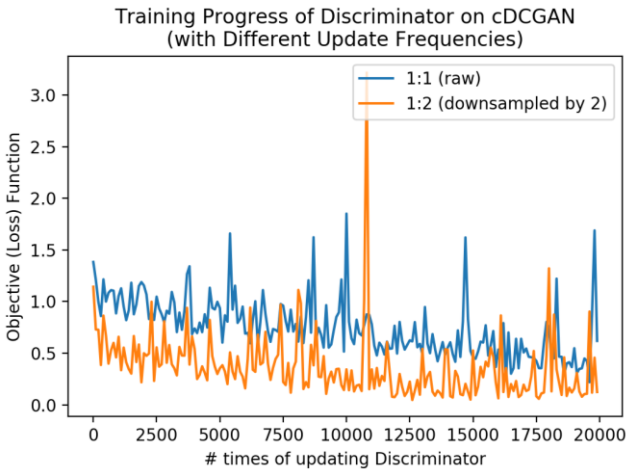
1. 學習率：以不同學習率實驗，訓練過程的比較如下所示。學習率的變化在 D 的目標函數上的影響似乎比較明顯，G 的目標函數的變化看不出什麼端倪。以下是模型在第 1500 次迭代的輸出，學習率由左至右分別是 5e-5、2e-4 和 5e-4，學習率高的模型在前期進步較快也符合我們的預想（但訓練更久結果不一定會比較好，只是收斂得比較快）。

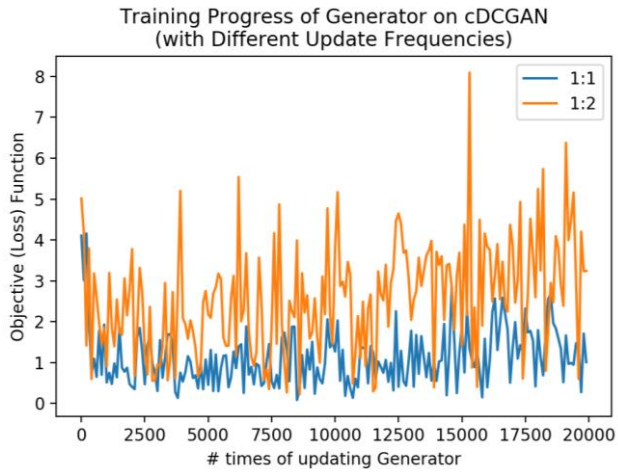






2. 交互更新頻率 G ： D = 1 ： 1 或 1 ： 2 進行兩次實驗，訓練過程如下圖（D 的訓練過程因為更新頻率不同，有先把資料點做 downsampling 後再畫圖）。整體看來 D 的目標函數的部分是 1 ： 2 的比較低，G 的則是相反的情況；生出來的圖片品質覺得沒有太顯著的差異，不過似乎是 1：1 的結果看起來好一點（左邊是 1 ： 1、右邊是 1 ： 2）。





3. 參數初始化：聽說蠻多人用 Xavier 做 GAN 的參數初始化，所以也在這個部分做了實驗，但結果不如想像的好，而且到後期似乎會出現 mode collapse 的問題，以下是實驗結果。

[1] Scott Reed et al., “Generative Adversarial Text to Image Synthesis.”