

# ADLxMLDS - hw4

孫奧 R05922147

## 1 Model description (2%)

### 1.1 model structure

總的架構是這樣的：有一個 Generator，和一個 Discriminator。Generator 的輸入是 noise 的 condition 拼接在一起的 vector，輸出的是一張(64,64,3)的矩陣(我們將這個矩陣理解成圖片)。而 Discriminator 的輸入是(64,64,3)的矩陣，輸出是一個實數值(如果不在 0~1 會讓他通過 sigmoid)，將其解釋成是否屬於真實分佈的幾率的判斷。

### 1.2 objective function

對於 G 而言很簡單，他的 objective function 是想要讓  $D(G)$  更接近與 1，這邊用 cross entropy 來完成。

而對於 D 而言，在這個作業中來自 3 個部分：Discriminator 對於真實資料的判斷、對於 G 生成資料的判斷、對於錯誤的 label(condition)的判斷。

具體實作的是將各個值與相對應的 0/1 做 cross entropy。

## 2 How to improve your performance (2%)

其一就是增加資料的方法。將原本的一張照片做 5 種角度的旋轉，以及把所有的照片做鏡像翻轉。



如上圖所示，一張照片變成了 11 張，這樣資料集的大小也變大了 11 倍。通過增加資料的數量得到的結果

其二在於 sample 的部分。由於這次作業需要餵給錯誤的 label 以及 label 對應錯誤的圖片(這邊想說一下：原始的論文中並不需要錯誤的圖片，而在討論和網上找到的 code 當

中都新增了這一個(正確 label, 錯誤圖片)的 error), 我使用的是每過 10 個 EPOCH 重新 sample 這些 pairs (一開始嘗試每次都重新 sample 訓練的會很滿甚至很難訓練起來)

### 3 Experimental settings and observation (2%)

#### 3.1 Experimental settings

```
hp = {  
    'Z_shape': [100],  
    'R_shape': [64, 64, 3],  
    'C_shape': [23],  
    'beta1': 0.5,  
    'beta2': 0.9,  
    'lr': 0.00001,  
    'batch_size': 128,  
    'g_output_activation': 'sigmoid',  
    'batch_norm': True  
}
```

其中 Z\_shape 代表 noise 的維度, C\_shape 代表 one-hot 的 condition 的維度(這邊一共有 23 維: 12 種頭髮 concat 上 11 種的眼鏡), beta1 和 beta2 是 AdamOptimizer 的參數。

#### 3.2 Observation



這是挑選一下 seed 所產生的結果圖。個人覺得如果說這是人臉我還是相信的。並且顏色之類的也能對上, 效果還是挺不錯的。