

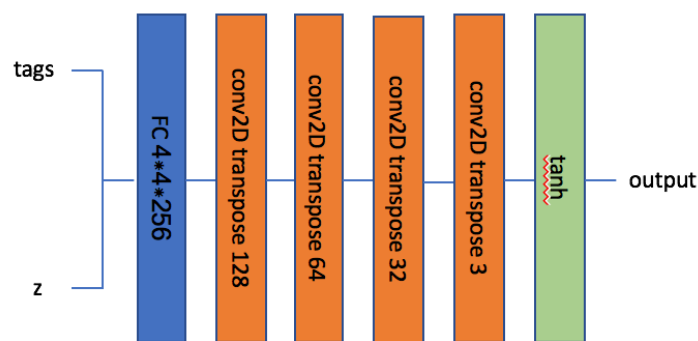
ADL hw4 report

R06922057 資工碩一 梁智泓

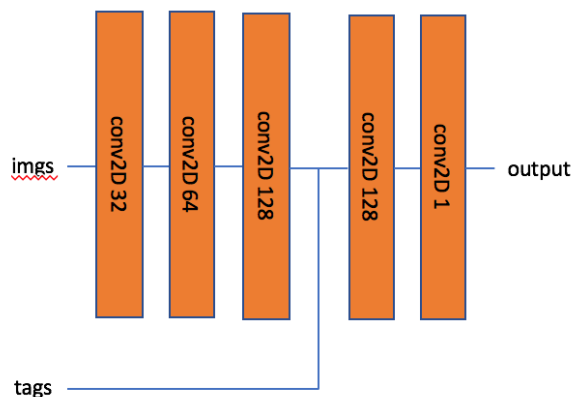
Model description

這次的作業中，我實作了四種不同的 GAN，分別是 improved WGAN, DCGAN, DCGAN ResNet，下面分別是三個 model 的敘述

- **DCGAN**：DCGAN 主要的 model 架構很簡單，分成 generator 與 discriminator，generator 主要是一層的 fully connect 處理 concat 的 tags 及 z，加上四層的 conv2D transpose，每層之間都有加上 batch normalize 以及 relu，最後再接一層的 hyperbolic tangent 做輸出，大致的架構圖如下

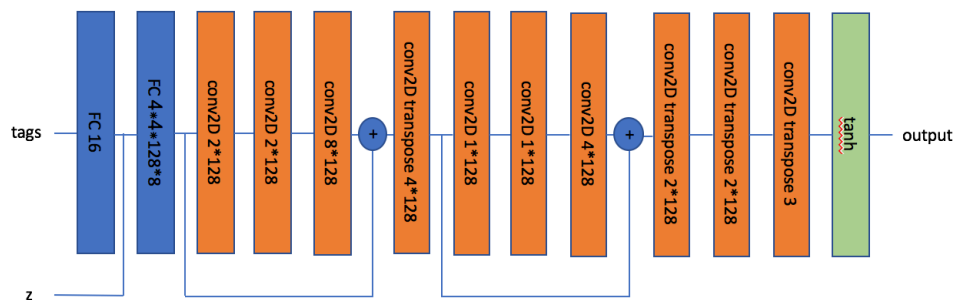


而 discriminator 則是先將 input 的 image 過三層的 conv2D，之後再將 input 的 tags 與第三層的 output concat，再過兩層的 conv2D，最後再輸出，而每層 conv2D 之間都有做 batch normalize 以及 leaky relu。而在 loss function 方面，大致的架構圖如下

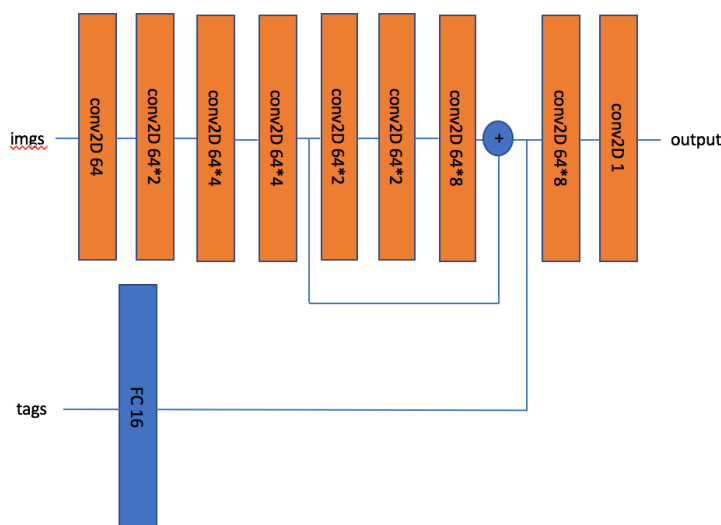


在計算 loss 時，分為四個 case，分別是 1. 真實的 image 配上真實的 tags，2. generator 生成的 image 配上真實的 tags，3. 真實 image 配上錯誤的 tags，4. 錯誤的 image 配上真實的 tags，而 generator loss 就是 case2 過 sigmoid 然後取 mean，而 discriminator loss 則是第一個 case1 + (case2+case3+case4)/3 過 sigmoid 然後取 mean，最後再用 AdamOptimizer 去 training，讓兩個 network 彼此越來越強。

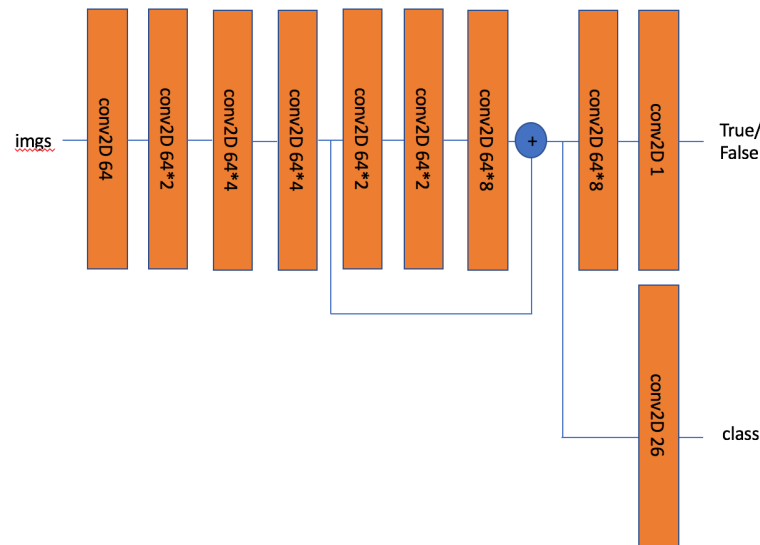
- Improved WGAN** : improve WGAN 的 model 架構跟 DC GAN 幾乎一樣，唯一不同的是在 discriminator 那裡沒有使用任何的 batch normalize，然後 discriminator 更新的速度會是 generator 的 5 倍快，而在 loss function 上，generator 的 loss 變成了負的 case1 取 mean，discriminator 的 loss 則是 $(\text{case2} + \text{case3} + \text{case4})/3 - \text{case1} + \text{Lambda} * \text{gradient penalty}$ ，而 gradient penalty 的公式如下 $E[(\|\nabla D(\alpha x + (1 - \alpha)G(z))\|_2 - 1)^2]$ ，其實就是將真實的 image 與 generator 生成的 image 用一定比例加在一起過 discriminator 之後取 gradient，再將 gradient - 1，最後再取 mean。而 optimizer 使用的是 RMSProp 而不是 Adam。
- DC GAN ResNet** : 與 DC GAN 最大的不同在於 model 的主架構是用 ResNet 所組成，generator 一開始是兩層 fully connected layer 處理 input 的 tags 及 noise z，然後過兩個 residual block 之後再過三層的 conv2D transpose，然後接一層 hyperbolic tangent 輸出，如下圖所示



而 discriminator 則是先將 input 的 image 過三層的 conv2D 之後，再過一層的 residual block，而 input 的 tags 則是先過一層的 fully connected layer 之後再與 image 的 feature concat 在一起，最後再一起過兩層的 conv2D 輸出，如下圖所示，而其餘的部分（包括 loss function）則是跟 DC GAN 一樣。



- **ACGAN ResNet**：與 DCGAN ResNet 最大的不同其實就是在 discriminator，不需要餵 tags 進去，而是讓 discriminator 自己學出 tags，大致的 model 架構如下圖。



D, G loss function 變成

$$D_loss = \lambda * (E[\log P(S = \text{real}) \mid X_{\text{real}}] + E[\log P(S = \text{fake}) \mid X_{\text{fake}}]) + E[\log P(C=c) \mid X_{\text{real}}] + E[\log P(C=c) \mid X_{\text{fake}}]$$

$$G_loss = \lambda * (E[\log P(S = \text{real}) \mid X_{\text{real}}] + E[\log P(S = \text{fake}) \mid X_{\text{fake}}]) + E[\log P(C=c) \mid X_{\text{fake}}]$$

How to improve performance

1. **Tags handling**：為了讓 performance 變好，我一開始便對 training 用的 tags 做了前處理，將無用的 tags 去除，只留下與 hair 跟 eyes 相關的 tags 及敘述。
2. **Image preprocessing**：為了讓 train 的過程更加 robust，我將 training 用的 image 做 flip、左右各 rotate 5 度，然後將其當成新的 training data pair，可以增加 training data 的量之外，更可以讓 discriminator 可以看到更多不同的 image 樣子，更為 robust。
3. **Tags random shuffle**：因為 test 過程中，hair 跟 eyes 的順序是不固定的，甚至可能是只有其中一種，為了讓 GAN 學出這樣的過程，我對處理過的 tags 做 random shuffle，讓它可以看到更多不同排序的 tags，以防止學到錯誤的排序方式。
4. **96 -> 64**：因為原本的照片大小是 96*96，所以我一開始直接使用 96*96 的圖做 train，輸出 96*96 的圖之後，再壓成 64*64，這樣出來的圖比較不會出現模糊的情況
5. **Improved WGAN**：如同上面所敘述，為了學出更好的 GAN 以生出更為精細的圖片，我嘗試實作了 Improved GAN，相關的實驗結果與比較會在最後面。
6. **DCGAN ResNet**：如同上面所敘述，為了學出更好的 GAN 以生出更為精細的圖片，我嘗試時做了 DCGAN ResNet，相關的實驗結果與比較會在最後面。

7. **ACGAN ResNet**：如同上面所敘述，為了學出更好的 GAN 以生出更為精細的圖片，我嘗試時做了 ACGAN ResNet，相關的實驗結果與比較會在最後面。

Experiment and observation






因為看 loss 比較難看出 train 的好壞，所以我直接將三個 model 在不同的 iteration 下，使用相同的 Z 所生成的圖放上來直接比較

DCGAN

iteration	10000	20000	30000	40000	50000
blue hair green eyes					
img					

因為 DCGAN 收斂的比較快，所以只展示了前 50000 iteration 的照片，可以看出大概在 20000-30000 iteration 就已經可以生出不錯的照片，但問題是沒辦法完全的對到 condition，而且也相對的不穩定，時常會生出不像人的照片，而過了 30000 iteration 之後，discriminator 會快速收斂，而 generator 會因為 discriminator 太強，所以開始放棄學習，開始只會生成固定的圖片，也就是 noise 的影響會漸漸變小。

Improved WGAN

iteration	50000	100000	150000	200000	250000
blue hair green eyes					

Improved WGAN 收斂得比 DCGAN 來得明顯慢，要大概 50000 iteration 才可以生出看起來像人臉的圖片，大概要 100000 iteration 才可以學出比較精細的 condition (eyes color)。可以直接從圖片看出，model 生出來的圖片畫質明顯較差，顆粒感很重，雖然可以從縮圖大致看出人臉的樣子，condition 也有學到，但生出來的圖片品質很差。

DCGAN ResNet

iteration	100000	150000	200000	250000	300000
blue hair blue eyes					
green hair blue eyes					

其實 DCGAN ResNet 大概在 10000 iteration 就可以生出很像人臉的圖像，但 condition 一直對不上，大概到了 50000 iteration 才會學到比較準確的 condition，從上面的圖可以看出隨著 iteration 漸漸變多，生出來的圖會越來越精細，會考慮到比較多人臉上面的細節，例如：大小眼的問題，正臉與側臉的不同，嘴巴大小，臉上腮紅等...不過比較不穩的反而是 condition 有時會對不太準，尤其是頭髮的顏色常常會把藍色與紫色搞混，雖然在色系上的確相似，但人的肉眼上卻還是可以清楚分辨。

ACGAN ResNet

iteration	5000	10000	15000	20000	25000
blue hair blue eyes					
blue hair red eyes					

換成 ACGAN 之後，可以看出臉的形狀有明顯的變好，而且在臉的各個細節上也處理得很好，condition 也比較不會出現混淆的情況，臉的邊緣也比較沒有模糊的情況發生，而且收斂的比較快。

比較三個 model，以結果來說，ACGAN ResNet 明顯比較穩定也比較好，所以我將 ACGAN ResNet 作為我最後的 model。