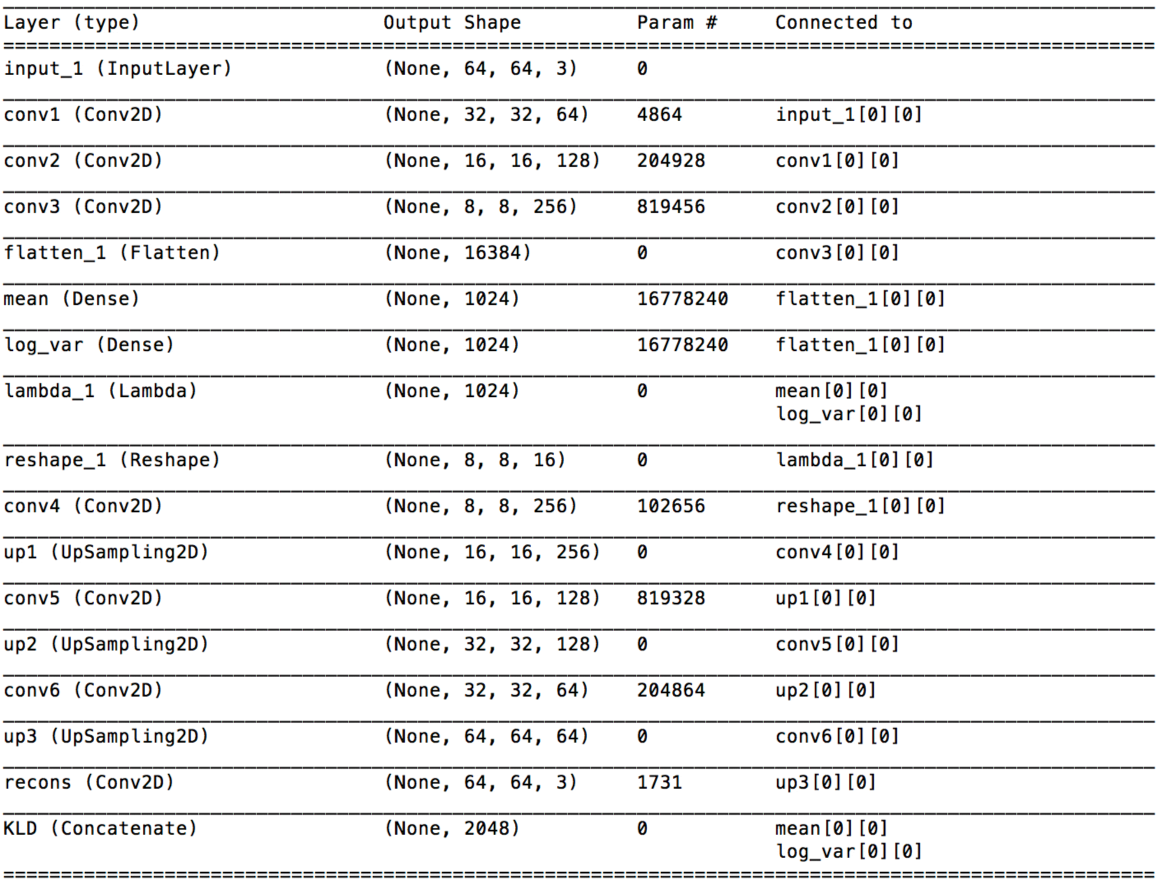
DLCV HW4 Report 物理四 B03202017 李漪莛

Problem 1 VAE

1-1

- Architecture



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 64, 64, 3)     0
____________________________________________________________________________________________________
conv1 (Conv2D)                   (None, 32, 32, 64)    4864        input_1[0][0]
____________________________________________________________________________________________________
conv2 (Conv2D)                   (None, 16, 16, 128)   204928      conv1[0][0]
____________________________________________________________________________________________________
conv3 (Conv2D)                   (None, 8, 8, 256)     819456      conv2[0][0]
____________________________________________________________________________________________________
flatten_1 (Flatten)              (None, 16384)         0           conv3[0][0]
____________________________________________________________________________________________________
mean (Dense)                     (None, 1024)          16778240    flatten_1[0][0]
____________________________________________________________________________________________________
log_var (Dense)                  (None, 1024)          16778240    flatten_1[0][0]
____________________________________________________________________________________________________
lambda_1 (Lambda)                (None, 1024)          0           mean[0][0]
                                                                   log_var[0][0]
____________________________________________________________________________________________________
reshape_1 (Reshape)              (None, 8, 8, 16)      0           lambda_1[0][0]
____________________________________________________________________________________________________
conv4 (Conv2D)                   (None, 8, 8, 256)     102656      reshape_1[0][0]
____________________________________________________________________________________________________
up1 (UpSampling2D)               (None, 16, 16, 256)   0           conv4[0][0]
____________________________________________________________________________________________________
conv5 (Conv2D)                   (None, 16, 16, 128)   819328      up1[0][0]
____________________________________________________________________________________________________
up2 (UpSampling2D)               (None, 32, 32, 128)   0           conv5[0][0]
____________________________________________________________________________________________________
conv6 (Conv2D)                   (None, 32, 32, 64)    204864      up2[0][0]
____________________________________________________________________________________________________
up3 (UpSampling2D)               (None, 64, 64, 64)    0           conv6[0][0]
____________________________________________________________________________________________________
recons (Conv2D)                  (None, 64, 64, 3)     1731        up3[0][0]
____________________________________________________________________________________________________
KLD (Concatenate)                (None, 2048)          0           mean[0][0]
                                                                   log_var[0][0]
====================================================================================================
```

- Implementation details:

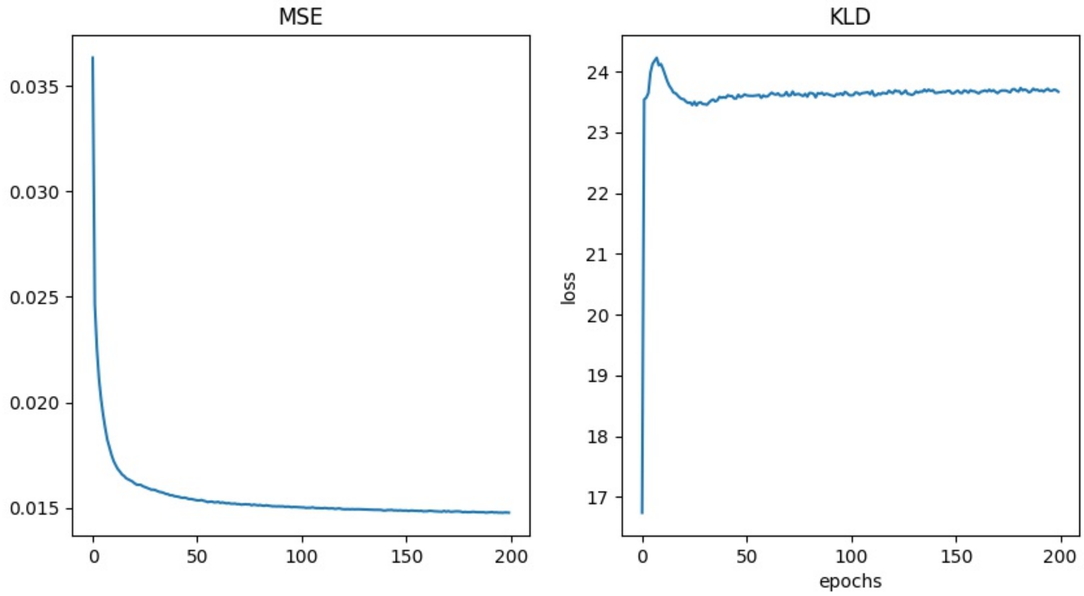
  Loss = Reconstruction_loss + lambda * KL_loss

  Latent dim = 1024

  Sampling 的 code 如下：

```
def sampling(z_mean, z_log_var):
    epsilon = K.random_normal(shape=(K.shape(z_mean)[0], latent_dim), mean=0., stddev=epsilon_std)
    return z_mean + K.exp(z_log_var / 2) * epsilon
```

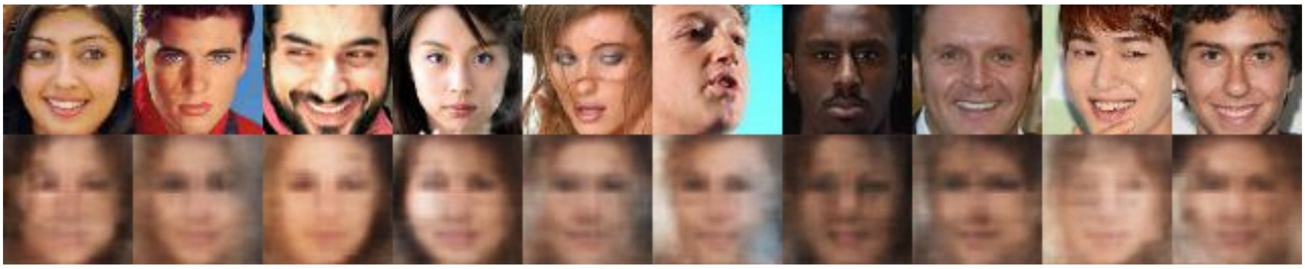
1-2



- 每個 epoch 紀錄一次

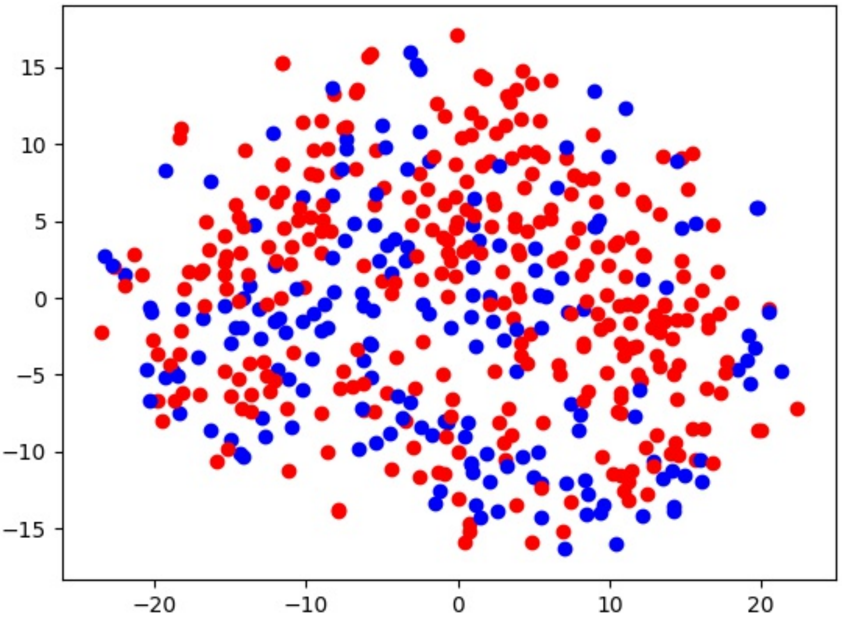
1-3

- MSE = 0.0336532



1-4 Random plot

1-5 TSNE



- 我畫了其中五百個點，紅色是 female，藍色是 male

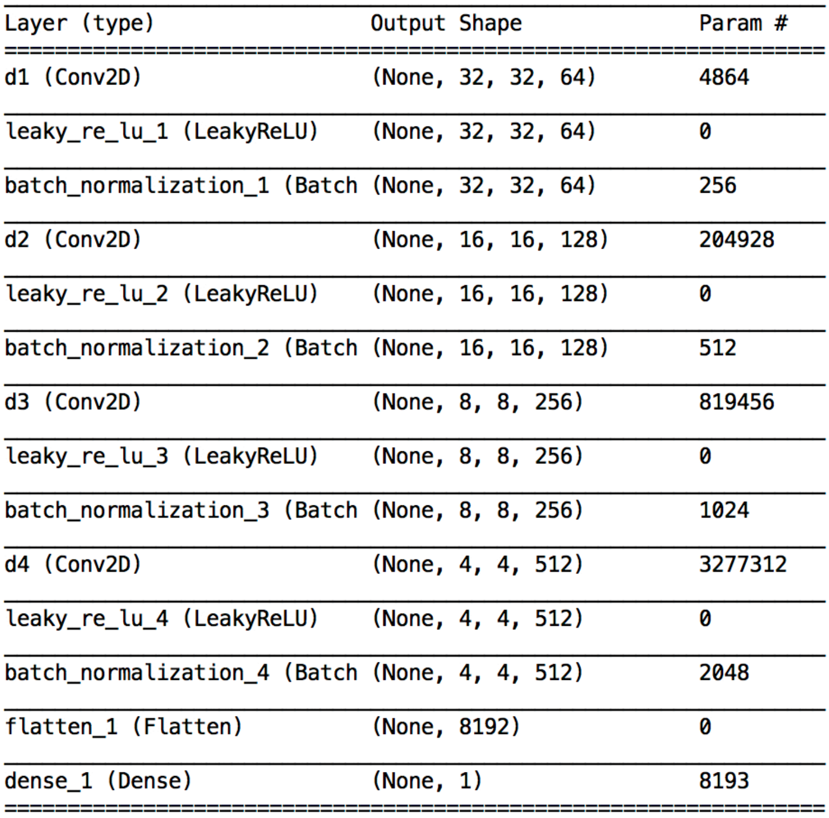
1-6 Observed and learned

- VAE 的 lambda（KL loss weight）若太大，則所有圖片都會看起來很像，但太小的話，在 testing data encode 完的 noise 上會有清楚的圖片，但 random noise 就會非常模糊，甚至都是雜訊。
- MSE 的 loss 通常都會穩定下降，而 KL loss 都會先急速下降，再緩緩上升直到穩定。
- 我這次花了很多時間在改架構跟調參數，看了很多人的 code，不過做出來效果還是很差，花比 GAN 還多的時間，decoder 試過 deconvolutional 或 conv+upsampling，kernel size 設過 3, 4, 5，batch normalization，leakyReLu 或 ReLu，把 capacity 調大，調整 lambda 都好像沒什麼用，只要 test 變清楚，random 就會變模糊，所以只好交出兩個都有點模糊的圖片。

Problem 2 GAN

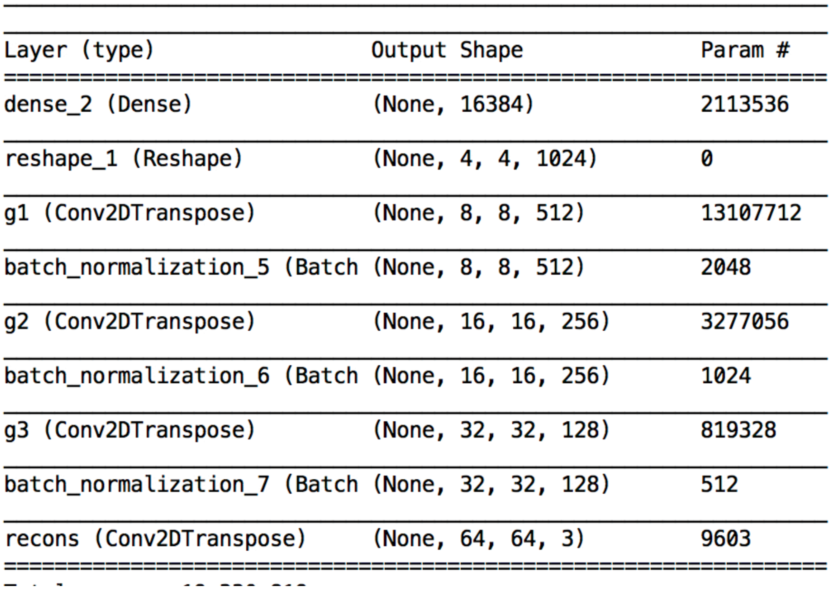
2-1 Architecture

- Discriminator



```
Layer (type)                 Output Shape              Param #
=================================================================
d1 (Conv2D)                  (None, 32, 32, 64)        4864
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 32, 32, 64)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 64)        256
_________________________________________________________________
d2 (Conv2D)                  (None, 16, 16, 128)       204928
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 16, 16, 128)       0
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 128)       512
_________________________________________________________________
d3 (Conv2D)                  (None, 8, 8, 256)         819456
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 8, 8, 256)         0
_________________________________________________________________
batch_normalization_3 (Batch (None, 8, 8, 256)         1024
_________________________________________________________________
d4 (Conv2D)                  (None, 4, 4, 512)         3277312
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 4, 4, 512)         0
_________________________________________________________________
batch_normalization_4 (Batch (None, 4, 4, 512)         2048
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 8193
=================================================================
```

- Generator



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_2 (Dense)              (None, 16384)             2113536
_________________________________________________________________
reshape_1 (Reshape)          (None, 4, 4, 1024)        0
_________________________________________________________________
g1 (Conv2DTranspose)         (None, 8, 8, 512)         13107712
_________________________________________________________________
batch_normalization_5 (Batch (None, 8, 8, 512)         2048
_________________________________________________________________
g2 (Conv2DTranspose)         (None, 16, 16, 256)       3277056
_________________________________________________________________
batch_normalization_6 (Batch (None, 16, 16, 256)       1024
_________________________________________________________________
g3 (Conv2DTranspose)         (None, 32, 32, 128)       819328
_________________________________________________________________
batch_normalization_7 (Batch (None, 32, 32, 128)       512
_________________________________________________________________
recons (Conv2DTranspose)     (None, 64, 64, 3)         9603
=================================================================
```

- Details

  一開始先將圖片的 pixels 值縮到-1~1 之間，

  Batch size = 128

  Noise size = 128

  Training epochs = 20000

2-2

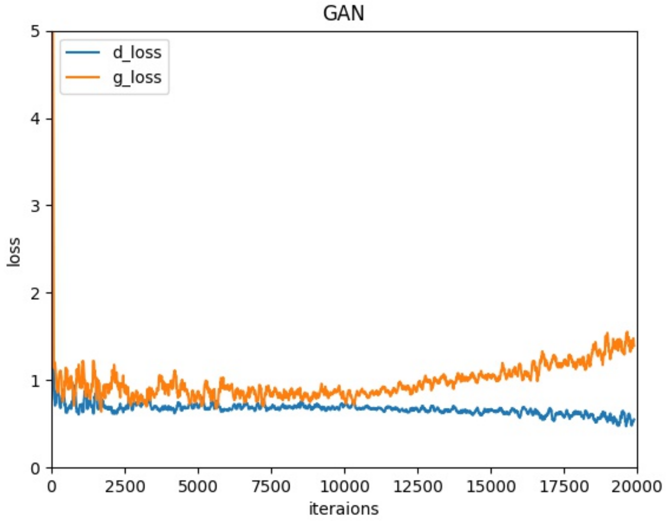


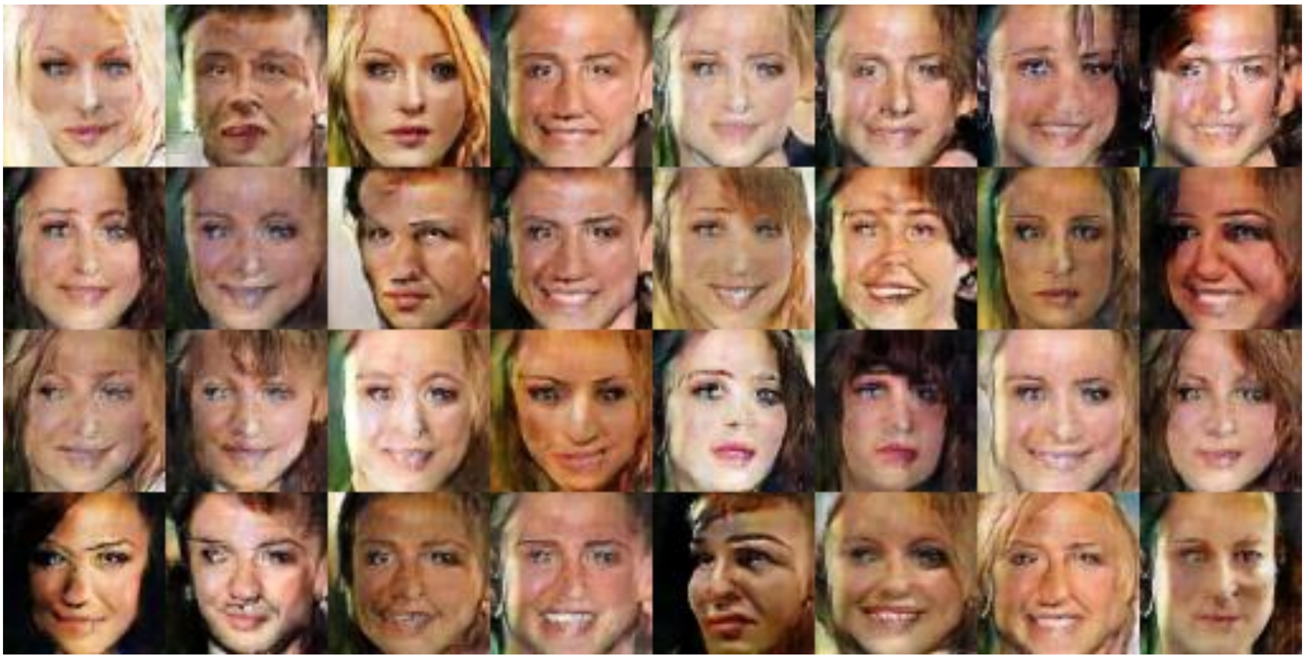
fig2_2

- Smooth 方式：每十個 iteration 紀錄一次，每次和前十個點取平均。
- What do you think it represents?

  D 和 G 的 loss 雖然不穩定，不過保持在一定範圍內，互相增強自己的能力，且通常，一方 loss 突然變高時，另一方就會下降，表示兩者強度開始有差距。

  其實到後期，Discriminator 好像有點稍強，有影響到一點輸出品質。

2-3 Random generated images

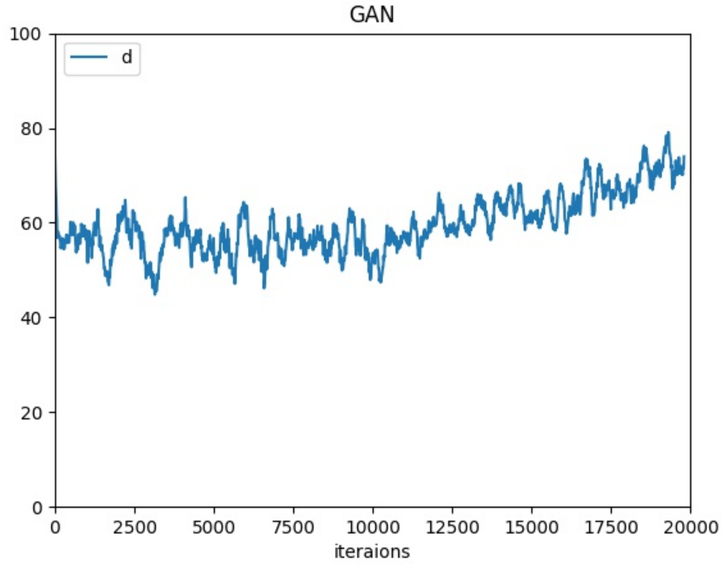


2-4 Observed and learned

- GAN 的兩種 loss 的震盪都很大，不過只要能維持在一定的範圍之內，且兩方的 loss 不要差太多，都能達到訓練效果；若有一方增大到爆掉，會輸出接近雜訊的圖片 。
- 通常在第 300 個 iterations 時，可以看到圖片中央有皮膚色的圓形輪廓。我跑了 20000 個

iterations 約花費了六個小時。

- 並且，合理的 discriminator accuracy 通常介在 60~75%之間。(如下圖)



Accuracy of Discriminator（補充）

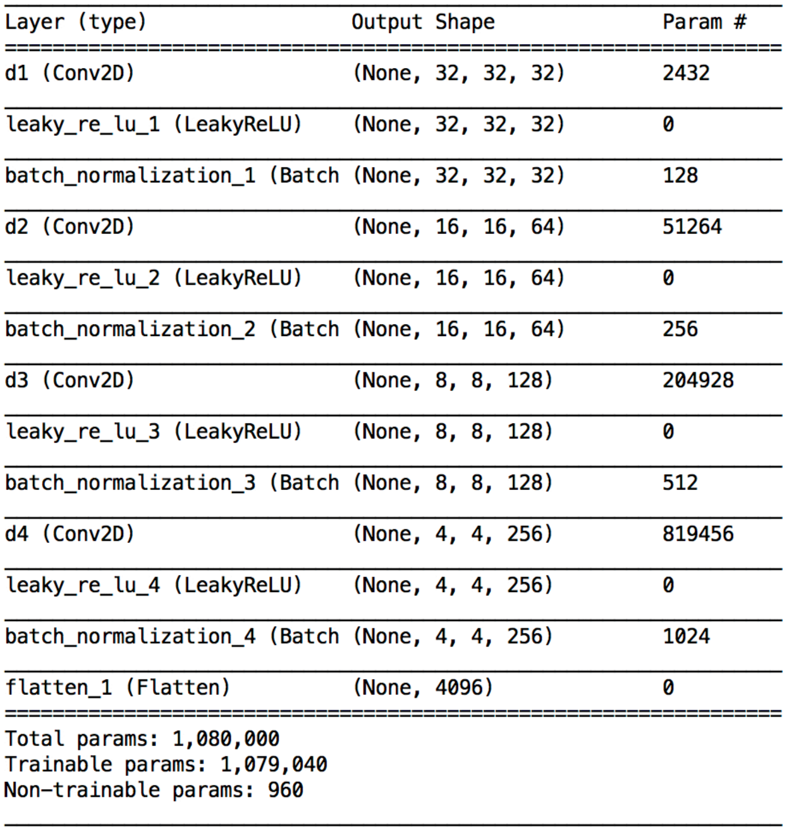
2-5 VAE & GAN

- GAN 訓練出來的圖片通常比較鮮艷，也比較不模糊，不過失敗的話，就會有直接毀容的感覺（還好 training data 的臉都長得滿好看的，不然可能會更慘）。
- GAN 使用的參數量較多，訓練過程比較長，也比較不穩定。
- 雖然理論上 VAE 應該比較簡單，但看起來我的 VAE 比較失敗 QQ，一般成功的 VAE 雖然比 GAN 模糊，但應該還是會比我的輸出清楚。

Problem 3 ACGAN

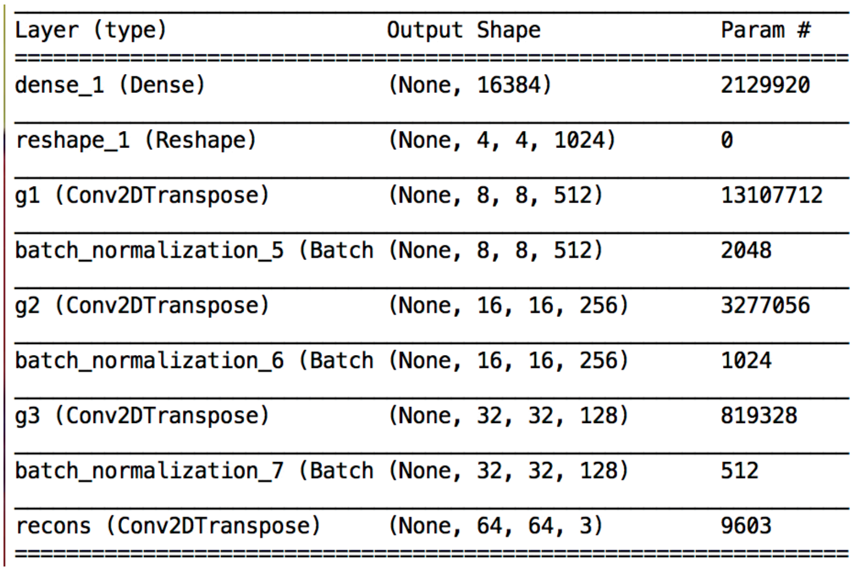
3-1

- Discriminator



```
Layer (type)                 Output Shape              Param #   
=================================================================
d1 (Conv2D)                  (None, 32, 32, 32)        2432      
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 32, 32, 32)        0         
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32)        128       
_________________________________________________________________
d2 (Conv2D)                  (None, 16, 16, 64)        51264     
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 16, 16, 64)        0         
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 64)        256       
_________________________________________________________________
d3 (Conv2D)                  (None, 8, 8, 128)         204928    
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 8, 8, 128)         0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 8, 8, 128)         512       
_________________________________________________________________
d4 (Conv2D)                  (None, 4, 4, 256)         819456    
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 4, 4, 256)         0         
_________________________________________________________________
batch_normalization_4 (Batch (None, 4, 4, 256)         1024      
_________________________________________________________________
flatten_1 (Flatten)          (None, 4096)              0         
=================================================================
Total params: 1,080,000
Trainable params: 1,079,040
Non-trainable params: 960
_________________________________________________________________
```

- 從 flatten 後再接 Dense(2)，使得輸出為 real/fake, class 1/0 兩個 classifier
- Generator

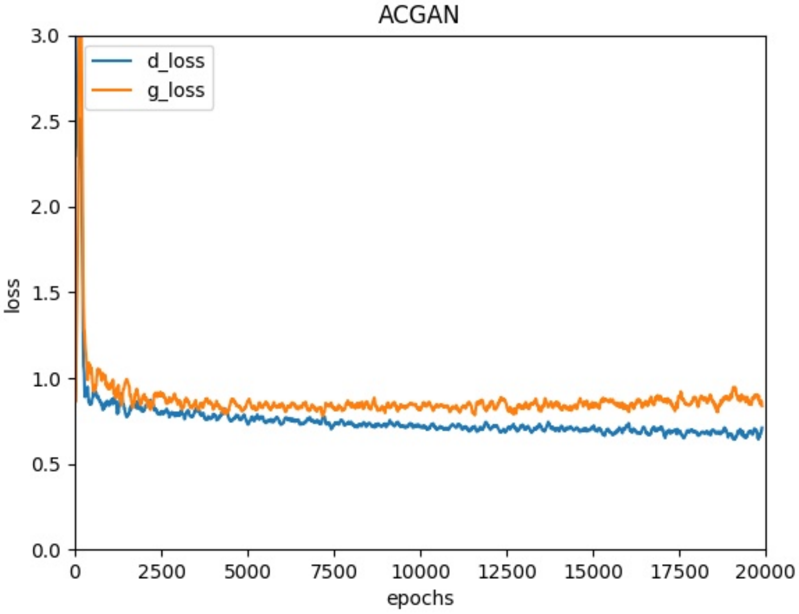


```
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 16384)             2129920   
_________________________________________________________________
reshape_1 (Reshape)          (None, 4, 4, 1024)        0         
_________________________________________________________________
g1 (Conv2DTranspose)         (None, 8, 8, 512)         13107712  
_________________________________________________________________
batch_normalization_5 (Batch (None, 8, 8, 512)         2048      
_________________________________________________________________
g2 (Conv2DTranspose)         (None, 16, 16, 256)       3277056   
_________________________________________________________________
batch_normalization_6 (Batch (None, 16, 16, 256)       1024      
_________________________________________________________________
g3 (Conv2DTranspose)         (None, 32, 32, 128)       819328    
_________________________________________________________________
batch_normalization_7 (Batch (None, 32, 32, 128)       512       
_________________________________________________________________
recons (Conv2DTranspose)     (None, 64, 64, 3)         9603      
=================================================================
```

- Details

  大致架構都和 GAN 相同，只有 discriminator 輸出變成兩維，都是 sigmoid（因為只有 binary feature），且 generator 輸入的 noise 多一維，紀錄 features。

3-2



ACGAN 的 loss

- Smooth 方式：每十個 iteration 紀錄一次，每次和前十個點取平均。
- 因為我在 GAN 的 discriminator 稍強，因此我這次 ACGAN 的 discriminator 中每一層 Conv2d 的 filter 數都只留一半（總參數量只剩一半），發現訓練過程的 loss 較穩定，訓練出來的圖片也比較清楚。

3-3

- 我選定的 feature 是 smiling。(上排：有笑，下排：沒笑)