DLCV HW4 Report

B04901069 電機三 林志皓

Problem 1 VAE

1. Architecture & Implementation Details

```
VAE:
Layer (type)                    Output Shape          Param #     Connected to
==================================================================================================
input 1 (InputLayer)            (None, 64, 64, 3)     0

conv2d 1 (Conv2D)               (None, 64, 64, 128)   1664        input 1[0][0]

conv2d 2 (Conv2D)               (None, 64, 64, 256)   131328      conv2d 1[0][0]

max pooling2d 1 (MaxPooling2D)  (None, 16, 16, 256)   0           conv2d 2[0][0]

conv2d 3 (Conv2D)               (None, 16, 16, 512)   524800      max pooling2d 1[0][0]

conv2d 4 (Conv2D)               (None, 16, 16, 512)   1049088     conv2d 3[0][0]

max pooling2d 2 (MaxPooling2D)  (None, 4, 4, 512)     0           conv2d 4[0][0]

flatten 1 (Flatten)             (None, 8192)          0           max pooling2d 2[0][0]

dense 1 (Dense)                 (None, 1024)          8389632     flatten 1[0][0]

dense 2 (Dense)                 (None, 1024)          8389632     flatten 1[0][0]

lambda 1 (Lambda)               (None, 1024)          0           dense 1[0][0]
                                                                  dense 2[0][0]

reshape 1 (Reshape)             (None, 1, 1, 1024)    0           lambda 1[0][0]

conv2d transpose 1 (Conv2DTrans (None, 4, 4, 1024)    16778240    reshape 1[0][0]

conv2d transpose 2 (Conv2DTrans (None, 16, 16, 512)   8389120     conv2d transpose 1[0][0]

conv2d 5 (Conv2D)               (None, 16, 16, 256)   524544      conv2d transpose 2[0][0]

conv2d transpose 3 (Conv2DTrans (None, 32, 32, 128)   131200      conv2d 5[0][0]

conv2d transpose 4 (Conv2DTrans (None, 64, 64, 3)     1539        conv2d transpose 3[0][0]
==================================================================================================
Total params: 44,310,787
Trainable params: 44,310,787
Non-trainable params: 0
```

Structure of VAE

在 encoder 的部分，我用 convolution & maxpooling layers 讓影像越來越小，深度越來越深，接近 latent space 的時候 flatten 成兩條分別為 1024 維的 mean & variance layer，再依據公式合成為 latent space。

而在 decoder 的部分，我先將 latent space reshape 為(batchs,1,1,1024)在使用許多層的 convolutionTranspose 將其放大至原圖大小(64,64,3)

Optimizer 我使用 Adam(lr = 1e-4)，在第一個 epoch 時就能達到還不錯的水準，而我在 KL lambda 值得選擇上做過許多嘗試，最後選擇 1e-5 這個數值，最後 reconstruct & random generate 的效果都不算差。

2.Learning Curve

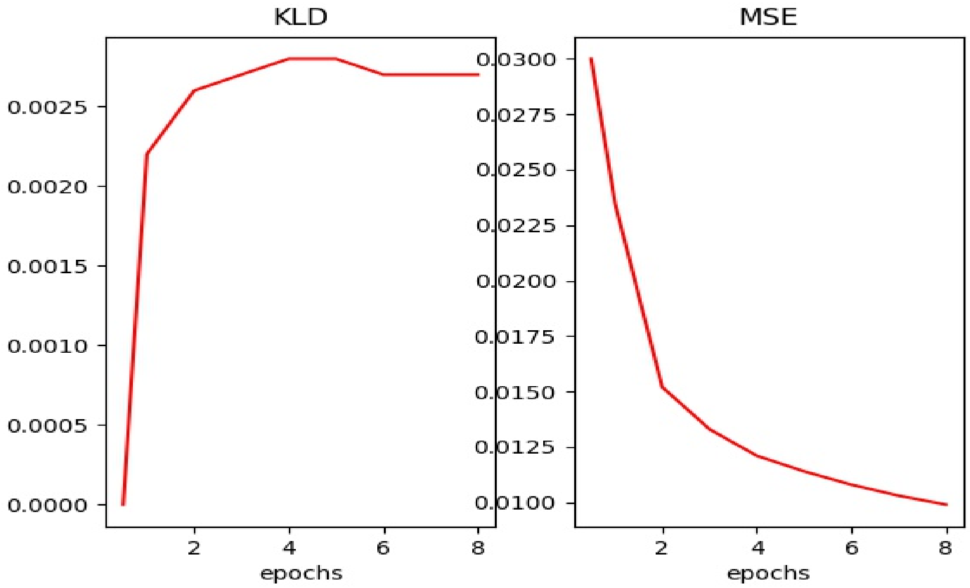

Fig1_2.jpg

3. 10 reconstruction Images of test images

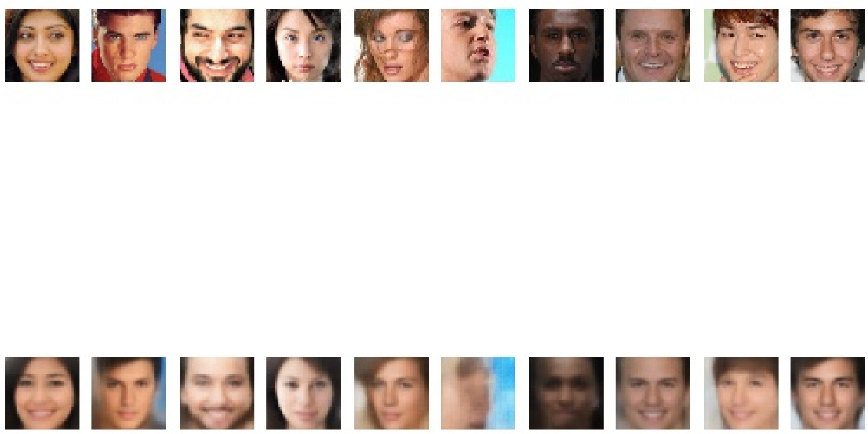

Fig1_3.jpg

4. 32 Random generated Images

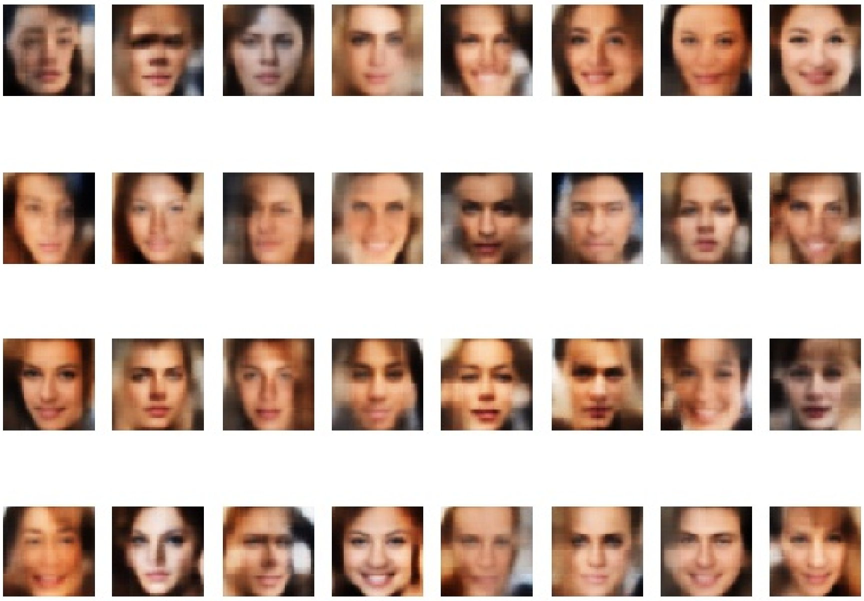

Fig1_4.jpg

5.tSNE

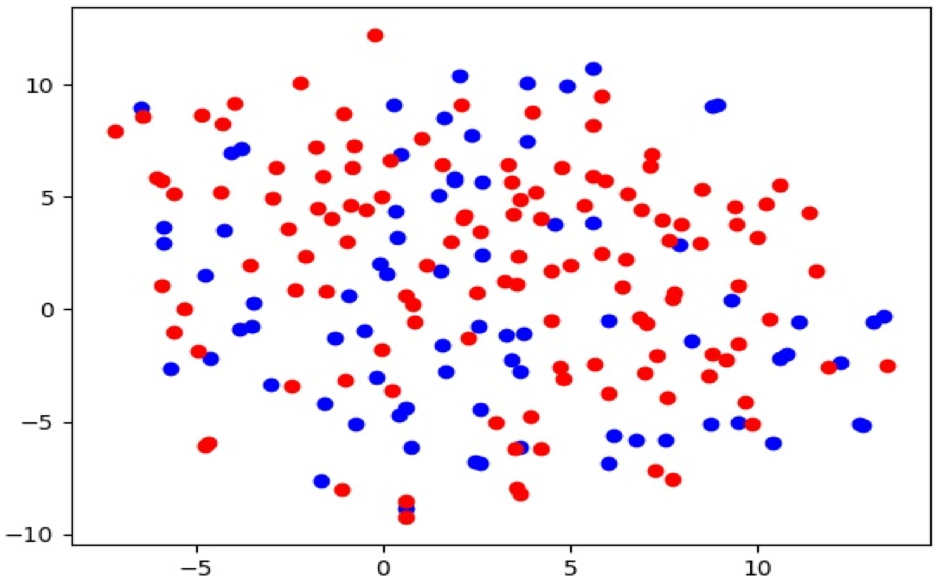

Fig1_5.jpg

（藍色為男性，紅色為女性）

6.What I observe & learn from implementing VAE

在訓練的過程中，我認為最關鍵的是 KL lambda 值的選擇。這個值是 reconstruction & random generation 之間的平衡取捨，我測試過許多不同的值，甚至有設過 0，generate 出來的東西跟人臉還是有一定的相似度，那時我覺得滿神奇的，後來想了想，推測是就算沒有 variance 算進去，mean 的 distribution 還是有一定程度的規則可以產生出圖(雖然 quality 不好就是了)經過實驗，我最終選擇 1e-5 這個數值，產生如上的結果。

Problem 2 GAN

1. Architecture & Implementation Details

```
Layer (type)                 Output Shape              Param #
=================================================================
input 1 (InputLayer)         (None, 1024)              0

reshape 1 (Reshape)          (None, 1, 1, 1024)        0

conv2d transpose 1 (Conv2DTr (None, 4, 4, 1024)        16778240

conv2d transpose 2 (Conv2DTr (None, 16, 16, 512)       8389120

conv2d 1 (Conv2D)            (None, 16, 16, 256)       524544

conv2d transpose 3 (Conv2DTr (None, 32, 32, 128)       131200

conv2d transpose 4 (Conv2DTr (None, 64, 64, 3)         1539
=================================================================
Total params: 25,824,643
Trainable params: 25,824,643
Non-trainable params: 0
```

Generator

```
Discriminator:

Layer (type)                 Output Shape              Param #
=================================================================
input 2 (InputLayer)         (None, 64, 64, 3)         0

conv2d 2 (Conv2D)            (None, 64, 64, 128)       1664

conv2d 3 (Conv2D)            (None, 64, 64, 256)       131328

max pooling2d 1 (MaxPooling2 (None, 16, 16, 256)       0

conv2d 4 (Conv2D)            (None, 16, 16, 512)       524800

conv2d 5 (Conv2D)            (None, 16, 16, 512)       1049088

max pooling2d 2 (MaxPooling2 (None, 4, 4, 512)         0

flatten 1 (Flatten)          (None, 8192)              0

dense 1 (Dense)              (None, 1024)              8389632

dense 2 (Dense)              (None, 1)                 1025
=================================================================
Total params: 10,097,537
Trainable params: 10,097,537
Non-trainable params: 0
```

Discriminator

我使用上一題的結果，將 VAE 的 encoder 當作 GAN 的 discriminator，因為 encoder 已被訓練，可萃取出影像的特徵(但這邊拿掉 variance 的那層，留下 mean 的)，因此再加上一個一維 Full connected layer，即是一個還不錯的 discriminator。

而 generator 的方面，我也是用 VAE 的 decoder 進行實作，因此在訓練初期即能產生一定程度的影像。

而我的 generator 進步速度較跟不上 discriminator，因此在訓練 generator 時我會用較多張的 image，比例跟訓練 discriminator 大概是 5 倍左右，才比較不會有其中一方爛掉的狀況發生。

2. Learning curve

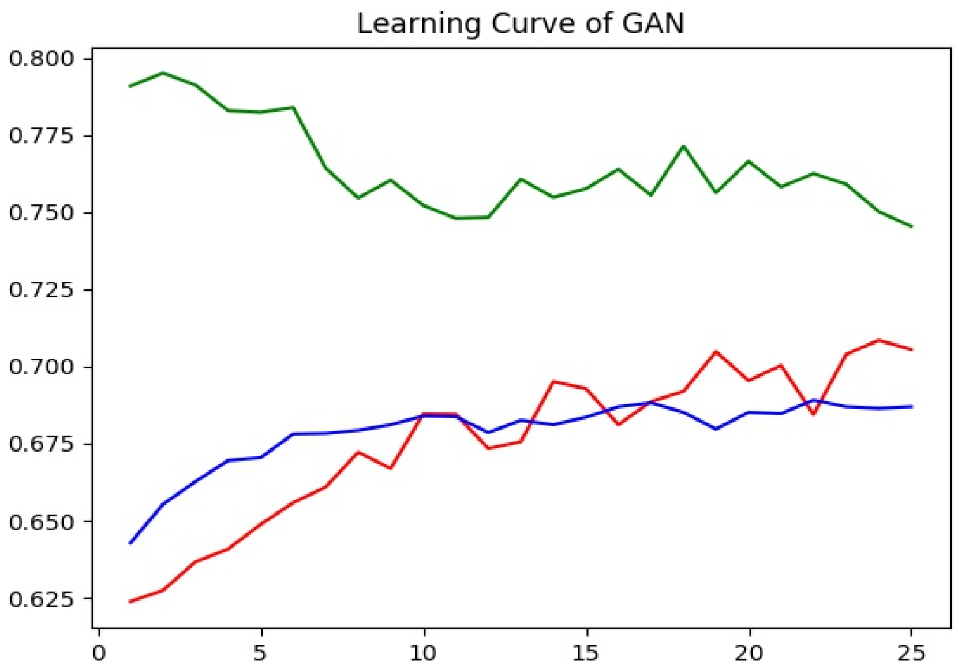

Fig2_2.jpg

綠色: Loss of Discriminator when feeding fake images

藍色: Loss of Generator

紅色: Loss of Discriminator when feeding valid images

依據圖中可看出 Discriminator 對於 fake image 能越來越分辨出來，而 generator 的 loss 並無明顯下降，甚至有點上升，個人推測是因為 generator 的進步速度跟不太上 discriminator 的關係。

3. 32 Random generated images

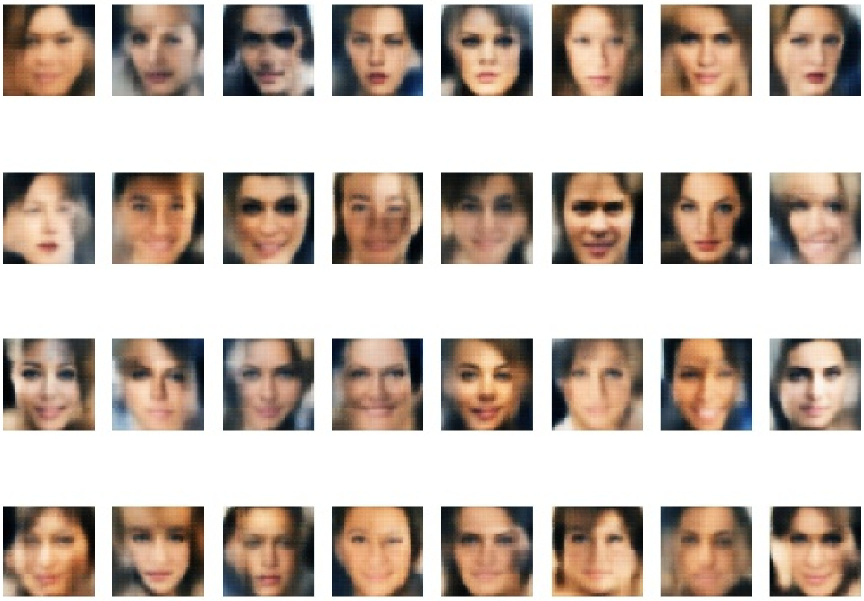

Fig2_3.jpg

4. What I've observed and learned from implementing GAN

我認為 GAN 比起 VAE 更加難以訓練，因為希望 discriminator 和 generator 能同時互相督促對方進步，因此每次訓練的張數、兩者之間訓練量的比例都必須多加嘗試，才能讓兩者都有在訓練的狀態。我有很多次失敗的例子，是參數沒調好，導致 discriminator 因為 fit 太多張 valid image 而導致幾乎只會預測出 1 的結果，對於 generator 的訓練就沒有效果；因此試過許多參數的選擇，才有讓兩者之間呈現互相競爭的關係。

5. Compare difference between image generated by VAE & GAN

在訓練 GAN 的過程中，很常會看到圖片的結果越來越不平滑而出現一塊塊的顆粒，或是略有方塊格線的痕跡。然而我不確定此為 Model 本身的特性，或是我沒有把他訓練好。

Problem 3 ACGAN

1. Architecture & implementation Details

```
Generator:

Layer (type)                 Output Shape              Param #
=================================================================
input 1 (InputLayer)         (None, 1024)              0

reshape 1 (Reshape)          (None, 1, 1, 1024)        0

conv2d transpose 1 (Conv2DTr (None, 4, 4, 1024)        16778240

conv2d transpose 2 (Conv2DTr (None, 16, 16, 512)       8389120

conv2d 1 (Conv2D)            (None, 16, 16, 256)       524544

conv2d transpose 3 (Conv2DTr (None, 32, 32, 128)       131200

conv2d transpose 4 (Conv2DTr (None, 64, 64, 3)         1539
=================================================================
Total params: 25,824,643
Trainable params: 25,824,643
Non-trainable params: 0
```

Generator

```
Discriminator:

Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input 2 (InputLayer)            (None, 64, 64, 3)    0

conv2d 2 (Conv2D)               (None, 64, 64, 128)  1664        input 2[0][0]
                                                                 input 2[0][0]

conv2d 3 (Conv2D)               (None, 64, 64, 256)  131328      conv2d 2[0][0]
                                                                 conv2d 2[1][0]

max pooling2d 1 (MaxPooling2D)  (None, 16, 16, 256)  0           conv2d 3[0][0]
                                                                 conv2d 3[1][0]

conv2d 4 (Conv2D)               (None, 16, 16, 512)  524800      max pooling2d 1[0][0]
                                                                 max pooling2d 1[1][0]

conv2d 5 (Conv2D)               (None, 16, 16, 512)  1049088     conv2d 4[0][0]
                                                                 conv2d 4[1][0]

max pooling2d 2 (MaxPooling2D)  (None, 4, 4, 512)    0           conv2d 5[0][0]
                                                                 conv2d 5[1][0]

flatten 1 (Flatten)             (None, 8192)         0           max pooling2d 2[0][0]
                                                                 max pooling2d 2[1][0]

dense 1 (Dense)                 (None, 1024)         8389632     flatten 1[0][0]
                                                                 flatten 1[1][0]

vadility (Dense)                (None, 1)            1025        dense 1[0][0]

lebel (Dense)                   (None, 1)            1025        dense 1[1][0]
==================================================================================================
Total params: 10,098,562
Trainable params: 10,098,562
Non-trainable params: 0
```

Discriminator

在 input 的部分，是一條長度 1024 的 vector，最後一個數值為 0 或 1，分別代表”無” 和”有”某個 attribute，期望 model 能因此產生相對應的 image。

而 Discriminator 的部分，output 有兩個長度為 1 的 Full-connected layer，分別判斷”是否為真圖?”與”是否具有某向指定特徵?” ，分別都用 softmax 為 activation function 以及使用 binary cross-entropy 作為 loss function。

在訓練 discriminator 時，餵 fake image 的方式是隨機製造 fake image，而 attribute 的部分則是隨機產生 0/1 讓 model 去 fit；分配訓練的比例則是:
Train Discriminator with 500 valid image
Train Discriminator with 500 fake image
Train Generator with 1000 noise

2. Learning Curve

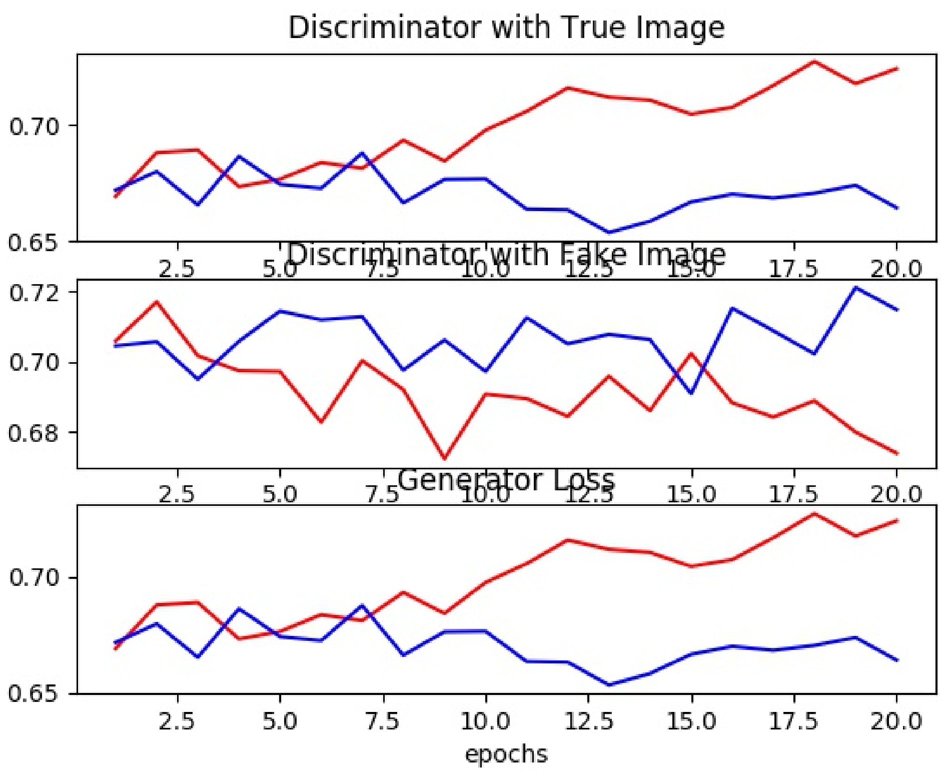

Fig3_2.jpg
上圖中藍色折線都代表關於 attribute 是否正確的 loss，
而紅色折線代表關於是否為真圖的 loss

關於是否為真圖，Discriminator 預測假圖的能力看的出來有隨時間增強，而 generator 的表現或許沒進步那麼多因此 loss 卻有微升的狀況；關於是否具有該 attribute 的部分，由圖中可以看出 discriminator 與 generator 都有隨時間增強的狀況，然而實際情形並不如預期順利(詳見下圖)

3. plot 10 pair generated images

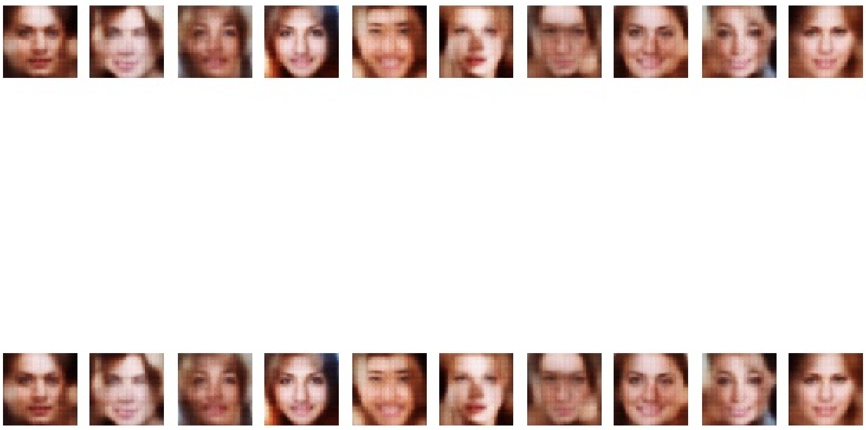

Fig3_3.jpg

上排為’具有’attribute 的圖
下排為’不具有’attribute 的圖片
Attribute:smile