Hw4 Report

Student ID : R05921012, Name : 吳宗澤

- Model Description – Must include model structure, objective function for G and D (2%)

我所嘗試的共有三種不同的 GAN(DCGAN、WGAN、Imporved WGAN)，但在 generator 和 discriminator 都是相同的模型。首先，在 generator，會把 100 維的 noise 和 25 維 one-hot encoding 的 condition 給 concatenate 起來，接著通過一層 Fully connected layer (4*4*1024) with relu(batch_normalization) -> conv2d_transpose (filter_size=4, filter_num=512, stride=2) with relu(batch_normalization) -> conv2d_transpose (filter_size=4, filter_num=256, stride=2) with relu(batch_normalization) -> conv2d_transpose (filter_size=4, filter_num=128, stride=2) with relu(batch_normalization) -> conv2d_transpose (filter_size=4, filter_num=3, stride=2) with sigmoid。

而 discriminator 則是將 64by64 的影像直接通過 conv2d(fileter_size=5,fileter_num=32, stride=2) with lrelu -> conv2d(fileter_size=4,fileter_num=64, stride=2) with lrelu -> conv2d(fileter_size=4,fileter_num=128, stride=2) with lrelu -> conv2d(fileter_size=4,fileter_num=256, stride=2) with lrelu -> 將 25 維 condition 拿進來並 reshape 成(-1,1,1,25)然後 tile 一下複製成 dim＝(-1,4,4,25)在和原本的去做最後一維的 concatenation 並在通過 conv2d(fileter_size=4,fileter_num=256, stride=2) with lrelu -> conv2d(fileter_size=4,fileter_num=1, stride=2) （if DCGAN 則要有 sigmoid）

Generator 的 objective function :

$\mathrm{h} \leftarrow \varphi(\mathrm{t})\{\text{Encode matching text description}\}$

$\mathrm{z} \sim \mathrm{U}(-1,1)\{\text{Draw sample of random noise from uniform distribution}\}$

$\hat{x} \leftarrow G(z,h)\{fake\ image\ from\ generator\}$

$L_G \leftarrow \log(D(\hat{x},h))$

$\mathrm{G} \leftarrow \mathrm{G} - \alpha\, \partial L_G / \partial G$

Discriminator 的 objective function :

$\mathrm{h} \leftarrow \varphi(\mathrm{t})\{\text{Encode matching text description}\}$

$\hat{\mathrm{h}} \leftarrow \varphi(\mathrm{t})\{\text{Encode matching text description}\}$
$\mathrm{z} \sim \mathrm{U}(-1,1)\{\text{Draw sample of random noise from uniform distribution}\}$
$\hat{x} \leftarrow G(z,h)\{fake\ image\ from\ generator\}$
$\mathrm{s}_r \leftarrow D(x,h)\{real\ image, right\ text\}$
$\mathrm{s}_w \leftarrow D(x,\hat{\mathrm{h}})\{real\ image, wrong\ text\}$
$\mathrm{s}_f \leftarrow D(\hat{x},h)\{fake\ image, right\ text\}$
$\mathrm{s}_{wr} \leftarrow D(\hat{x},h)\{wrong\ image, right\ text\}$
$L_D \leftarrow \log \mathrm{s}_r + (\log(1-\mathrm{s}_w) + \log(1-\mathrm{s}_f) + \log(1-\mathrm{s}_{wr}))/3$
$\mathrm{D} \leftarrow \mathrm{D} - \alpha\, \partial L_D / \partial G$

- How do you improve your performance and Experiment settings and observation (2% + 2%)

  首先，我有嘗試過 batch normalization 在 discriminator 每一層除了最後一層或是除了第一層和最後一層，但我發現加入 batch normalization 後的效果不是到太好，讓我的 model 學不太起來，然後嘗試過不同的 noise 像是從 normal distribution (-1, 1)、normal distribution (0, 1)、uniform distribution (-1, 1)、uniform distribution (0, 1) sample 出來，發現 normal distribution 會讓 mode collapse 更嚴重，讓每種 condition 出來的人臉都長很像，而 uniform distribution 則可以保有一定的 diversity。

  再來，如果讓 noise 的維度大一些（我測試過兩種 100 和 200），維度大也可以讓圖片生成的結果更 diverse 不易發生 mode collapse，但學習的效率則會比較慢，因為會常 sample 沒看過的 noise 導致生成的圖片效果不好。另外，我也有用 skip-thought vector 但效果並沒有比 one-hot 還要好，他也會讓 condition 區分的比較慢。

  我也曾測試過 GAN 沒有用 conv2d 到尾就不是 DCGAN 那樣效果會比較差一些。我有試著去嘗試使用 WGAN 和 Improved WGAN 但我下面會直接列 Improved 版的 objective function 因為兩個的差別只在對 gradient 更新速率限制的不同，一個是直接 clip 住另一個則加 gradient penalty

  Generator 的 objective function :

$\mathrm{h} \leftarrow \varphi(\mathrm{t})\{\text{Encode matching text description}\}$
$\mathrm{z} \sim \mathrm{U}(-1,1)\{\text{Draw sample of random noise from uniform distribution}\}$
$\hat{x} \leftarrow G(z,h)\{fake\ image\ from\ generator\}$

$L_G \leftarrow D(\hat{x}, h))$

$\mathrm{G} \leftarrow \mathrm{G} - \alpha\, \partial L_G / \partial G$

Discriminator 的 objective function :

$L_D \leftarrow \mathrm{s}_r - (\mathrm{s}_w + \mathrm{s}_f + \mathrm{s}_{wr})/3$ + gradient penalty

$\mathrm{D} \leftarrow \mathrm{D} - \alpha\, \partial L_D / \partial G$

不過後來我發現假如我這樣去使用的話，會讓我的頭髮顏色都是彩色會讓他學不到如何去區分 condition，眼睛也會常常有區分不好的問題。

而在一般的 DCGAN 會有比較好的效果可以很快速的分出各種髮色和眼睛，並產生相對應的照片，如下圖：

但我發現，train 太久會有一個缺點就是一開始有些照片數量比較少的 condition，產生的照片則會開始有 mode collapse 和模糊化的結果，如上圖第一排 orange hair 的會全部變成一樣的模糊照片，其他有些則會變得黑黑暗淡的照片。這讓我真的很難

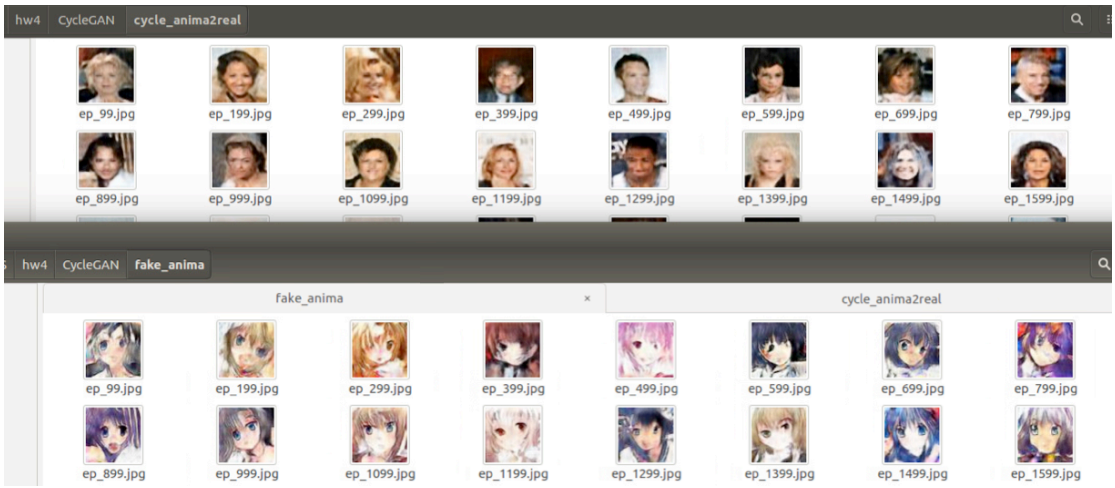
抓到什麼時候該停止訓練，都只能利用眼睛去觀察結果，因為從 loss 觀察其實在每個不同的的 batch 都會有點震盪。

除此之外，我發現若丟相同的 noise 給 generator 他出來的臉都是一樣的，而不同的 condition 會直接影響到髮色和眼色。

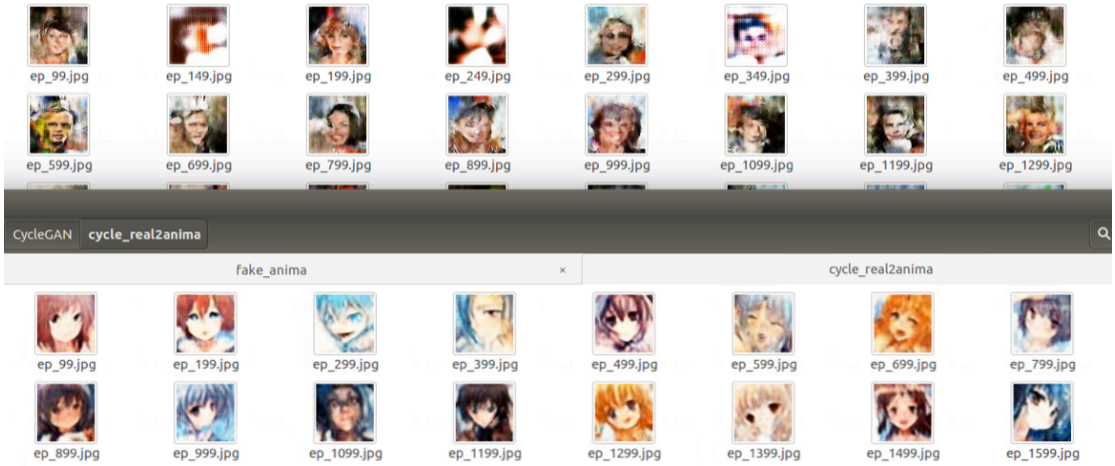
再來，我也有發現若是我沒有把髮色跟眼色 unknown 的資料給拿掉，會影像到我區分 condition 的效果，因此我只保留大概 16000 多張，讓兩個都有或是只有髮色或只有眼色那些 tag 的照片給留著。

BONUS

我的 style transfer 是拿我們的動漫 dataset 跟 celebA 的將真實的臉轉去動漫的臉，而其中的 model 的地方則都跟 cycle GAN 的 model 一樣，generator 的地方是使用 resnet 那種形式並疊了 9 個 block，然後 discriminator 我並沒有使用 patch discriminator，因為我想說原本就都 resize 成 64by64 去做了，就沒有想說要再去利用 patch discriminator 來減少訓練參數，而是使用一般 DCGAN 那種 conv2d 到底的 discriminator。而其中的演算法 objective function 都和 cycle GAN paper 上的一樣。下圖的下方是將真人轉乘的動漫人物，上方則是將動漫人物 cycle transfer 回去真人臉，可以看出效果其實還不錯。他通常都是讓背景色或是頭髮色比較大的色塊轉成動漫人髮色



下圖則是，下方是我轉成的動漫臉而上方則是我轉回去的真臉，我將 anima 的臉轉成真人臉的圖，在過程中我發現，他一開學會的是直接將動漫臉直接換成真人臉，再來則是學怎麼產生真人照片後方那種亂亂的背景。但其實這個效果就沒有這麼好，因為好像圖比較色彩繽紛一點 discriminator 就會認為跟 celebA 是一樣的了。



因為在 paper 上寫說他們使用 batch=1 一張一張轉換，讓訓練的速度其實還滿慢的，一天頂多 10 幾個 epoch（我將 anima 3w 多張全使用,而 celebra 挑出 3w5 千張來用），但我發現他跟我們 DCGAN 在產生動漫臉有一樣的問題，就是 train 太久一樣會壞掉（我後來 train 了大概 75 個 epoch）如下圖。