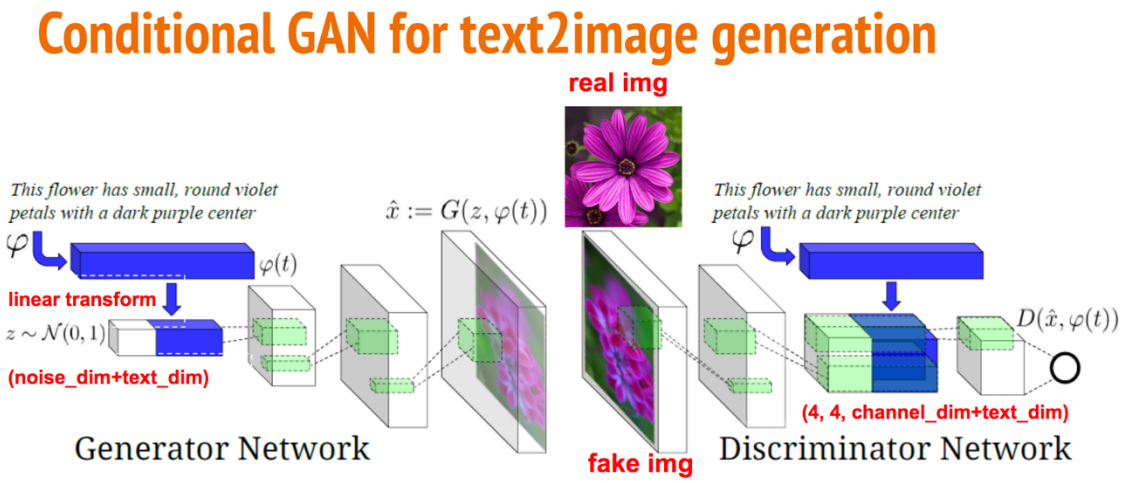
R06725048 資管碩一 陳信豪

MLDS HW4 Report

1. Model description

我的 Conditional GAN 架構如作業 slide 所示



Paper: https://arxiv.org/pdf/1605.05396.pdf

以下是更詳細的說明：

- 圖片前處理

  首先我只保留至少有 hair 或 eyes 其中一個 tag 的圖片

  而圖片在餵進 model 前會先 resize 成 64 x 64 x 3，並且將所有維度除以 127.5 後減 1 （彩色圖片的 normalization）

- Condition tag 處理

  我認為有關 tag 的部分太複雜其實也沒有什麼益處（要花更多時間才能 train 到較好的結果），所以我只保留 12 種顏色的 hair 和 11 種顏色的 eyes 和 unknown hair、unknown eyes，並做成長度為 25 的 one-hot tag vector (其中只有兩個維度是 1，分別代表 hair 與 eyes，其他是 0)

- Generator

  Generator 會先將 tag vector 和一個隨機產生的 noise vector 合併 (concat)，然後再經過一層 Fully-Connected Layer （reshape -1,4,4,256），再經過四層 Convolution Layer 後輸出圖片，其中每層之間都有 batch normalization

- Discriminator

  Discriminator 會先將吃進來的圖片 vector 經過三層 Convolution Layer (Generator 的 Deconvolution) 後，再與 tag vector 合併 (concat)，再經過兩層 Convolution Layer (最後輸出維度是 1) 後，得到一個值，其中每層之間都有 batch normalization

- Conditional GAN Model

  Train Conditional GAN model 時，會先用 Generator 產生 fake img，然後讓 Discriminator 吃四種配對，(right tag, real img)、(right tag, fake img)、(wrong tag, real img)、(right tag, wrong img)，

  - Train Generator

    要讓 (right tag, fake img) 的輸出為 1

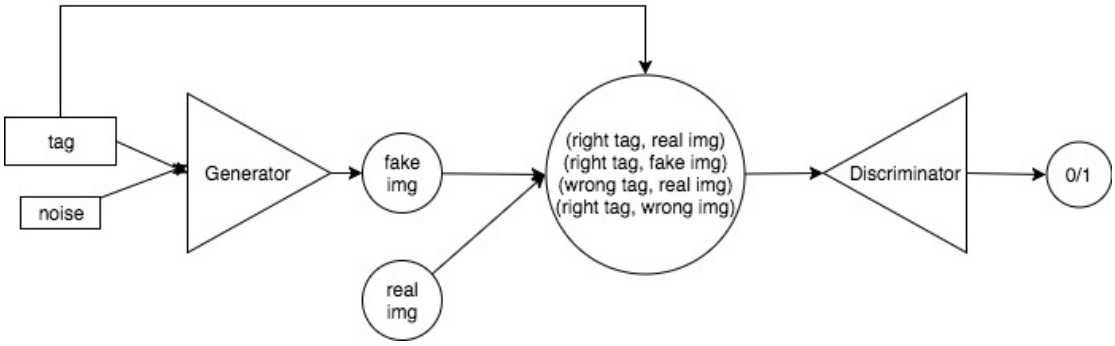
    minimize [loss(right tag-fake img,1)]

  - Train Discriminator

    除了 (right tag, real img) 要輸出 1 以外

    其他三種配對要輸出 0

    minimize [((loss(right tag-fake img,0) + loss(wrong tag-real img,0) + loss(right tag-wrong img,0))/3) + loss(right tag-real img,1)]



- 參數
  - Update-Generator: Update-Discriminator = 1:1
  - z_dim = 100 (noise vector 長度)
  - Adam(lr = 2e-4)
  - Weights random_normal_initializer stddev = 0.02

    (權重初始 random 值平均為 0，標準差為 0.02)
  - Batch size = 64

2. How do you improve your performance
   A. 輸出五張差異較大的圖片的方法

      Train 到最後會發現其實 Model 很容易把 noise vector 帶來的影響給忽略掉，導致產生的圖片都長得非常像，為了解決這個問題，我的做法是 training 到後期時，在五個不同的 epoch (step) 存五份不同的 weight (即保留五份 checkpoint)，然後 testing 時，分別利用這五個 checkpoint，就可以產生出差異較大的五張圖片了。

   B. 權重初始值調整

      我原本將所有參數的初始值設置成零，結果發現完全 train 不起來，圖片都一整片灰，後來改成 random_normal_initializer，(default mean = 0, stddev = 1)，也還是不行，最後將 random_normal_initializer 的 stddev 設成 0.02 才 train 成功

   C. 圖片翻轉旋轉

      由於我只取至少有 hair 或 eyes 的圖片來 train，導致資料變少，為了解決這一個問題，我將取出來後的圖片水平翻轉、順時針轉五度、逆時針轉五度，以此多出三倍的資料

   D. Update Ratio

      我有嘗試調整 Generator 和 Discriminator 的更新比例，因為在 train GAN 時會希望兩者的表現都不要相差太多。一般來說 Discriminator 會比 Generator 的表現好，所以用 1:1 的話，Generator loss 會漸漸的越來越大。(以我的 model 而言，train 到後期，Discriminator loss 大致都維持在 0.1 以內，而 Generator loss 則是高至 10 左右)

      所以我有嘗試過 2:1 和 5:1 和 10:1，甚至是設 loss threshold (若低於該 thresh 才算更新完畢) … 等等。而的確，兩個 loss 之間可以保持抗衡了，但是代價是 training 的時間變得更長，而且從產生出來的圖片來看，表現也沒有變得更好。

      而我認為之所以沒有變得更好的原因是，Generator 在學的時候是會先學簡單的部分再學複雜的部分，而在每個 step 給 Generator 多 train 幾次代表著是讓它去多學複雜的部分，但其實 Discriminator 自己複雜的部分也還沒有學好，因此 Generator 的學習就變得沒有意義了。

      簡單來說，train GAN 就是「教學相長」，兩邊同時進步才會有效率。

3. Experiment settings and observation

基本參數設置皆如第一大點所提

然後由於用 loss 來衡量 GAN 的表現實在是沒什麼意義

所以我這邊就以產出的圖片來做實驗比對了

A. random_normal_initializer 的 stddev

左上：stddev = 1 (tensorflow default)、右上：stddev = 0.5

左下：stddev = 0.05、右下：stddev = 0.02

stddev = 0 (zero initial)

stddev 的設置很重要，如果設太大或設成 0 都會 train 不起來

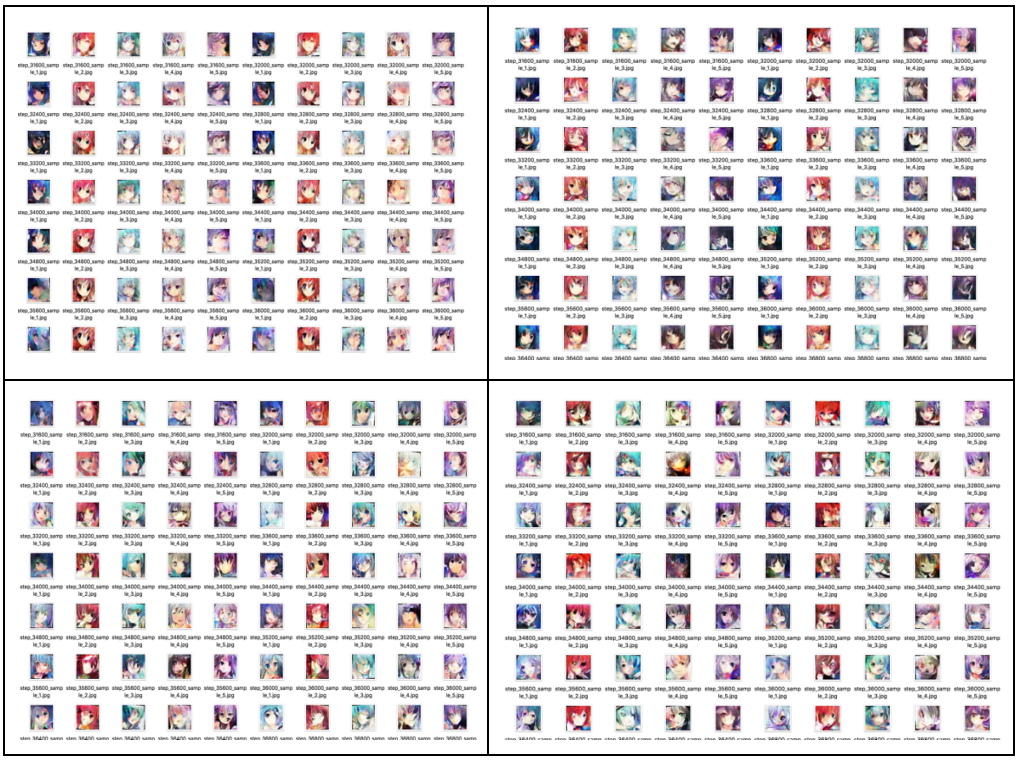
經實驗結果 stddev = 0.02 是不錯的

B. z_dim

由於助教設的 z_dim 長度是 100，但我現在只做 hair 和 eyes 兩個維度的 one-hot (共 25 維)，因此會擔心 z_dim 是否過長而影響結果

左上：z_dim = 1、右上：z_dim = 10

左下：z_dim = 50、右下：z_dim = 100



從實驗結果來看，似乎是多慮了，實驗的結果區別不出好壞

真要說的話，z_dim 越小似乎 generate 的圖片到後期的變化也會越小（即 noise vector 的影響越快被消除掉）