

ADLxMLDS2017 HW4 report

姓名：徐有慶

學號：R05922162

1. Model description

● Preprocess

每筆 training data 只取跟髮色還有眼色有關的資訊出來，如果該筆 training data 沒有任何跟髮色或眼色有關的資訊，就捨棄這筆 data，髮色或眼色也要包含在 testing 時會出現的顏色之中，如果沒有的話則當作沒出現。過濾完畢後，每筆 training data 都能得到「color hair color eyes」的資訊，對於髮色還有眼色分別做 one-hot encoding 再串接起來，便可以形成一個 25 維的向量(12 種髮色、11 種眼色、眼色及髮色各有 1 維代表沒出現該資訊)，再將該 25 維向量和產生出來的 noise vector 串接起來，丟進 Generator 裡面產生一張大小為 64*64 的假圖片。

將每張圖片先 resize 成 64*64 的圖片，並將每個像素都正規化到-1 到 1 之間

● Generator

■ Model structure

Dense, neuron = 4*4*256, activation = relu
Reshape(4, 4, 256)
Conv2d_transpose, output_channels = 128, kernel_size = 5, strides = 2, padding = 'same', activation = relu
Conv2d_transpose, output_channels = 64, kernel_size = 5, strides = 2, padding = 'same', activation = relu
Conv2d_transpose, output_channels = 32, kernel_size = 5, strides = 2, padding = 'same', activation = relu
Conv2d_transpose, output_channels = 3, kernel_size = 5, strides = 2, padding = 'same', activation = tanh

除了最後一層外，每一層都有先經過 batch_normalization 才過 activation function

■ Loss function

假設(fake image, right text)經過 discriminator 出來的值為 d1

$\text{loss} = \text{mean}(-\log(\text{sigmoid}(d1)))$

● Discriminator

■ Model structure

Conv2d, output_channels = 32, kernel_size = 5, strides = 2, padding = 'same', activation = leaky relu
Conv2d, output_channels = 64, kernel_size = 5, strides = 2, padding = 'same', activation = leaky relu
Conv2d, output_channels = 128, kernel_size = 5, strides = 2, padding = 'same', activation = leaky relu
Concatenate
Conv2d, output_channels = 128, kernel_size = 1, strides = 1, padding = 'same', activation = leaky relu
Conv2d, output_channels = 1, kernel_size = 8, strides = 1, padding = 'valid', activation = None

除了最後一層外，每一層都有先經過 batch_normalization 才過 activation function。Concatenate 做的就是將前一層的輸出和「color hair color eyes」所形成的 25 維向量串接起來，形成一個(8, 8, 128 + 25)的 tensor。

■ Loss function

假設(right image, right text) 經過 discriminator 出來的值為 d

(fake image, right text) 經過 discriminator 出來的值為 d1

(right image, wrong text) 經過 discriminator 出來的值為 d2

(wrong image, right text) 經過 discriminator 出來的值為 d3

$\text{loss_d} = \text{mean}(-\log(\text{sigmoid}(d)))$

$\text{loss_d1} = \text{mean}(-\log(1-\text{sigmoid}(d1)))$

$\text{loss_d2} = \text{mean}(-\log(1-\text{sigmoid}(d2)))$

$\text{loss_d3} = \text{mean}(-\log(1-\text{sigmoid}(d3)))$

$\text{loss} = \text{loss_d} + (\text{loss_d1} + \text{loss_d2} + \text{loss_d3}) / 3$

2. How to improve performance

主要還是從 training data 還有 model 的架構下做改善，將原先的圖片做旋轉、鏡像，藉此獲得更多的 training data。還有將 Generator 最後一層的 activation function 改成 sigmoid 而不是 tanh，直觀上來說，會把圖片的每個像素正規化到 0 到 1 之間，而網路上蠻多資料反而是把它正規化到 -1 到 1 之間，然後使用 tanh 當作 activation function，原因好像是因為這樣平均值為 0，可以收斂的比較快，但還是想試試，如果正規化到 0 到 1 之間出來的效果會是如何。也有試著讓 Generator 先輸出 96*96 的影像，之後再轉成 64*64 的影像做輸出，聽說這樣可以讓影像看起來比較清楚。

3. Experiment settings and observation

● Hyperparameters

Updates between Generator and Discriminator 1:1

Normal noise dimension: 100

Batch size: 64

Optimizer: Adam, learning rate = 0.0002

Training step: 30000

● Experiment

總共做了三種不同的實驗，分別是

Test1: 將圖片左右旋轉 5 度、鏡像

Test2: 將圖片左右旋轉 5 度、鏡像，Generator 最後一層接 sigmoid 而不是 tanh

Test3: Generator 輸出 96*96 的影像，最後再 resize 到 64*64

以下圖片皆為 training step = 30000 時所產出的影像，由左至右，由上到下分別:

1. blue hair blue eyes
2. blue hair green eyes
3. blue hair red eyes
4. green hair blue eyes
5. red hair green eyes
6. gray hair green eyes
7. green hair red eyes
8. orange hair brown eyes
9. blonde hair

Original:



Test1: 旋轉、鏡像



Test2: 旋轉、鏡像、sigmoid



Test3: 96*96 -> 64*64



● Observation

將圖片做旋轉、鏡像之後，可以發現邊緣會變成黑色的，而出來的顏色看起來比較一致，較不會有漸層的感覺，但相對的，眼睛還有嘴巴的部分可能會消失或變得較小。將 **activation function** 改成 **sigmoid** 後，感覺顏色上沒有原先的那麼深，也沒那麼明顯。而將圖片先輸出成 96*96 再 **resize** 成 64*64 出來的結果看起來大部分是壞掉的，但最後的兩張看起來較其他的真實一些，感覺可能要再改一下 **model** 的架構才會有比較好的結果。