

# MLDS HW3

---

R05921035 陳奕安  
R04921055 劉叡聲  
R05921043 林哲賢  
R05548020 吳侑學

## Environment

---

OS Ubuntu 14.04  
CPU Intel® Core™ i7-3770K CPU @ 3.50GHz  
GPU GeForce GTX TITAN  
Memory 32GB  
libraries

- cv2
- theano
- nltk3
- gensim

## Model Description

---

本次作業使用的model與[1]中使用的model大致上相同。

### Input

- Image使用cv2套件將size縮小至(64,64,3)
- z\_vector為100維在(-1.,1.)的uniform distribution
- Tag部份使用[2][3]的skip-thoughts套件encode至4800維的sentence vector。

### Generator

- 將4800維的tag vector embedding到50維，並與z\_vector concat後經過下列layers
- Full-connected : 4 \* 4\*1024 + batch\_norm + relu
- conv2d\_transpose : 512個 5\*5 filter, stride=2 + batch\_norm + relu
- conv2d\_transpose : 256個 5\*5 filter, stride=2 + batch\_norm + relu
- conv2d\_transpose : 128個 5\*5 filter, stride=2 + batch\_norm + relu
- conv2d\_transpose : 3個 5\*5 filter, stride=2 + batch\_norm + tanh
- 得(64,64,3)的輸出

## Discriminator

- 以(64,64,3)的image作為輸入經過下列layers
- conv2d : 64個 5\*5 filter, stride=2 + batch\_norm + lrelu
- conv2d : 128個 5\*5 filter, stride=2 + batch\_norm + lrelu
- conv2d : 256個 5\*5 filter, stride=2 + batch\_norm + lrelu
- conv2d : 512個 5\*5 filter, stride=2 + batch\_norm + lrelu
- 將tag embedding至50維後的每一個維度變成4\*4\*1的filter後與上一層輸出concat在一起。
- conv2d : 512個 1\*1 filter, stride=1 + batch\_norm + lrelu
- Full-connected : 1 neuron
- 得1個scalar的輸出

## Loss

- generator loss : sigmoid\_cross\_entropy\_with\_logits(1, D(G(z,t)))
- discriminator\_loss :  
 $\text{sigmoid\_cross\_entropy\_with\_logits}(1., \text{D}(\text{real image}, \text{correct tag})) + \text{sigmoid\_cross\_entropy\_with\_logits}(0., \text{D}(\text{real image} + \text{wrong tag})) + \text{sigmoid\_cross\_entropy\_with\_logits}(0., \text{D}(\text{G}(z), \text{correct tag}))$

## How do you improve your performance

---

### tag encoding

由於這次要implement的特徵是髮色和眼睛顏色，相對於[1]要針對句子做encode是簡單很多的情況，我們認為使用skiptokens vector做encdoe成4800維似乎有點多餘。因此嘗試使用one hot encoding對不同顏色encode，最後只以兩個各是11維的one of k來代表兩種顏色，作為tag的embedding。然而實際train了之後，雖然也有部份圖片train出對應特徵，但卻沒有skiptokens encoding好。

### embedding dimension

將4800維encode後的結果在D跟G中分別會做embedding降至較低維度，[1]中使用256維，但我們發現使用256維時train出來的圖片容易糊掉且條件也不明顯。降至50維反而讓圖片較sharp且條件也更明顯。

### GAN -> WGAN -> GAN-gp

嘗試使用改進後的GAN的loss function來train，但training的時間比GAN多上許多，且圖片雖然不容易出現雜訊點，但五官位置、比例有些許怪異，condition也並未很明顯得顯示，或許是在real image+wrong tag的loss部份要有不同的處理方法。

### different conditional GAN structure

除了[1]的結構之外，我們也使用類似ACGAN[4]的結構來train，與上述的model相比，generator產生的image另外丟進一個classifier network來分類髮色與眼色。training步驟如下：

- 用real data與 true tag 來train Classifier
- 以real image與 $G(z, tag)$ 產生的image來train D。
- 以 $D(G(z, tag))$ 來train G。
- 以Classifier( $G(z, tag)$ )來train G。

然而產生出來的圖效果極差，根據產生圖片結果，我們認為應該是用classifier更新G時，反而使圖片產生雜訊。

## Experiment settings and results

conditional GAN : 50維 tag embedding @ iteration 5000



conditional GAN : 50維 tag embedding @ iteration 11000



conditional GAN : 256維 tag embedding @ iteration 5000 (and broken @ iteration 6000)



AC-WGAN : one-hot 22維 tag embedding @ iteration 10000



## Team division

- 陳奕安 (R05921035) : implement model
- 林哲賢 (R05921043) : implement model

- 劉叡聰 (R05921043) : implement model
- 吳侑學 (R05548020) : implement model

## Reference

---

- [1] Generative Adversarial Text to Image Synthesis <https://arxiv.org/abs/1605.05396> (<https://arxiv.org/abs/1605.05396>)
- [2] Skip–Thought Vectors <https://arxiv.org/abs/1506.06726> (<https://arxiv.org/abs/1506.06726>)
- [3] ryankiros/skip-thoughts <https://github.com/ryankiros/skip-thoughts> (<https://github.com/ryankiros/skip-thoughts>)
- [4] Conditional Image Synthesis With Auxiliary Classifier GANs <https://arxiv.org/abs/1610.09585> (<https://arxiv.org/abs/1610.09585>)