

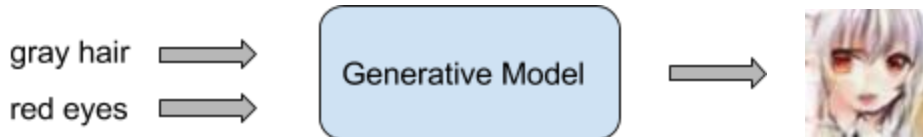
MLDS 2017 Assignment 3

Problem Description

Text to Image Generation

給予一個頭髮顏色的tag和一個眼睛顏色的 tag，我們要產生符合此描述的動漫人物的臉的圖。

Example:



Environment

- OS: Linux
- CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, Memory: 64GB
- GPU: GeForce GTX 1080, Memory: 8GB
- Libraries:
 - Python - 3.5
 - Tensorflow - 1.0
 - NumPy - 1.12.0
 - Progressbar2 - 3.18.1
 - Scipy - 0.19.0
 - Scikit-image - 0.13.0

Model Description

Data Preprocessing

只使用有包含<顏色頭髮>和<顏色眼睛> tag的圖來訓練，且若其中一個tag超過一個以上，該圖就捨棄掉。經過統計，顏色總共有15種，包含 'blonde', 'brown', 'black', 'blue', 'pink', 'purple', 'red', 'green', 'gray', 'aqua', 'white', 'orange', 'yellow', 'bicolored'，我們使用one-hot encoding來encode頭髮的顏色和眼睛的顏色，所有顏色加上 <UNK>（沒有 tag），頭髮和眼睛各為十五維的 vector，串接成為三十維的 feature vector。此外，透過旋轉與水平翻轉圖，增加了十倍量的訓練資料。

Model Structure

- Generator: CNN（採用 DCGAN 的 generator 設計）
Objective Function: 採用一般GAN 的objective function

$$\min \mathbb{E}_{h \sim p_h, z \sim p_z} [-\log(D(G(z, h)))]$$

- Discriminator: CNN（採用 DCGAN 的 discriminator 設計）
Objective Function: 採用一般 conditional GAN 的 objective function

$$\min - \{ \mathbb{E}_{x, h \sim p_{data}(x, h)} [\log D(x, h)] + \mathbb{E}_{x \sim p_{data}(x, h), \hat{h} \sim p_h, h \neq \hat{h}} [\log(1 - D(x, \hat{h}))] \\ + \mathbb{E}_{h \sim p_{data}(x, h), \hat{x} \sim p_x, x \neq \hat{x}} [\log(1 - D(\hat{x}, h))] + \mathbb{E}_{h \sim p_h, z \sim p_z} [\log(1 - D(G(z, h)))] \}$$

- 參數說明
x: 圖片
h: 文字的 vector
z: noise

(x, h) : (正確的照片, 正確的文字)

(x, \hat{h}) : (正確的照片, 錯誤的文字)

(\hat{x}, h) : (錯誤的照片, 正確的文字)

Model Details

- Generator 和 Discriminator 參數更新次數比 : 1:1
- AdamOptimizer with lr = 0.0002, momentum = 0.5
- z: normal distribution $N(0,1)$, dim = 100
- batch size: 256
- epoch: 60

Improvement

Filter Tags

原本使用全部 tag 的文字訓練一個 rnn 模型, 並使用該 rnn 將文字 encode, 再送進我們的模型裡, 但發現這樣產生的圖片會模糊, 推測可能是其他的 tag 數目太多, 影響模型學習好頭髮和眼睛的顏色。因此我們將 tag 過濾到只剩一個 <顏色頭髮> 的 tag 和一個 <顏色眼睛> 的 tag。若一張圖的 <顏色頭髮> 的 tag 和 <顏色眼睛> 的 tag 其中一個超過一個以上, 直接丟棄該筆訓練資料; 若缺少 tag, 補上 <UNK>。在此設定下, 我們發現圖片變得比較清晰, 且符合文字的描述。

Data Augmentation

經過上述 filter tags 的步驟後, 訓練資料量變得很少, 因此藉由對每張圖旋轉和水平翻轉, 產生原本十倍量的訓練資料, 幫助我們的模型學習。經過實驗, 我們發現確實有增加圖片的清晰度和文字的準確率。

Improved Negative Sampling

訓練 discriminator 的時候, 要給它看四種組合: (真實的照片, 正確的文字)、(假的照片, 正確的文字)、(正確的照片, 錯誤的文字)、(錯誤的照片, 正確的文字)。但在取得(假的照片, 正確的文字)、(正確的照片, 錯誤的文字)組合時, 要注意不能隨機選取, 否則容易取到相同 tag 的圖片。在前處理時, 我們就將每筆訓練資料的所有 negative samples 存起來, 訓練模型時, 從該筆資料的 negative sample set 裡隨機選取。

Different Loss Functions

我們嘗試了一般的 DCGAN、Least Squares GAN (LSGAN) 和 Improved Wasserstein GAN (Improved WGAN) 三種模型的 loss。在下面實驗中, 我們呈現了不同模型下圖片生成的結果, 可見 DCGAN 和 LSGAN 產生的圖片較清晰。我們選擇 DCGAN 當作我們最好的模型。

Experiments

Basic Settings

G: CNN (activation function for output layer: tanh)

D: CNN

Epoch: 60

Samples from Different Models











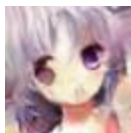



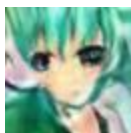


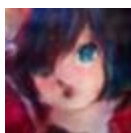

我們比較了以下四種不同的模型:

DCGAN: 一般的 Conditional GAN

DCGAN(G:sigmoid): 將 generator output layer 的 activation function 從 tanh 換成 sigmoid

LSGAN: generator 和 discriminator 的 objective function 改成 L2 loss

Improved WGAN: 使用一般 WGAN 的 loss, 但是不做 weight clipping, 改使用 gradient penalty

	DCGAN	DCGAN (G:sigmoid)	LSGAN	Improved WGAN
aqua hair brown eyes				
pink hair aqua eyes				
gray hair purple eyes				
green hair green eyes				
black hair blue eyes				

Observation:

若以圖片清晰度來比較，DCGAN > LSGAN > DCGAN (G:sigmoid) > Improved WGAN。若只針對頭髮顏色和眼睛顏色的正確性，這四種 model 的準確度都相當高，因此我們認為只要在前處理時將 tag 過濾乾淨，就可以達到頭髮和眼睛顏色的正確性。

Team Division

組員	分工
江東峻 r05922027	data preprocessing, DCGAN, LSGAN, Improved WGAN
陳翰浩 r05922021	data preprocessing, DCGAN, Improved WGAN
鄭嘉文 r05922036	experiments, report