MLDS HW3 – Generative Adversarial Network

TensorJoe

r05922063 陳啓中 r05921032 陳昱維 r05943093 蔣君涵 d04921018 艾弗里

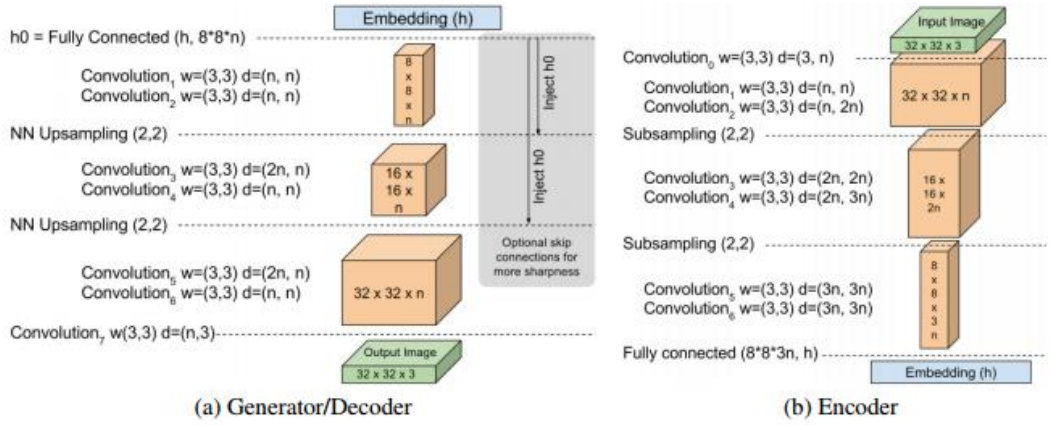
Environment

OS:	Ubuntu 14.04
CPU:	Intel Core i7-4790
GPU:	NVIDIA GTX970
Python Library:	Tensorflow r1.0
	Numpy 1.11.0
	Scikit-image 0.13.0
	Scipy 0.19.0

Model description

我們實作的是上個月才發表的 BEGAN model:
https://arxiv.org/abs/1703.10717

BEGAN 中的 generator 是由底下圖中左邊的 network 所構成，而 discriminator 則是一個 AutoEncoder，由圖中右邊的 encoder 及左邊的 decoder 所構成。圖中的 upsampling 及 downsampling 我使用 bilinear interpolation 來做。



對於 discriminator 而言，他一方面會將 real image 的 reconstruction loss 盡可能降到最低，另一方面要將 generator 所產生的 fake image 的 reconstruction loss 升高。而 generator 的目標則是盡可能讓 discriminator 的 reconstruction loss 越低越好。這篇 paper claim real image 跟 fake image 的 reconstruction loss 差其實就是種 Wasserstein distance。

Objective 如下所示：

The BEGAN objective is:

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases}$$

不過這篇 paper 並沒有實作 condition 的部分，我會在下一個段落說明我怎麼讓 BEGAN 做 conditioning。

How do you improve your performance

- 如何實作 condition

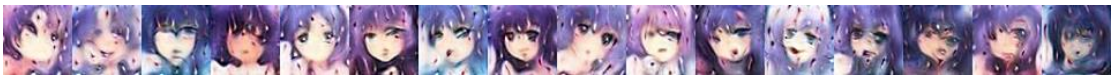
  我們利用類似 text2image 的作法，將 condition embedding 安插在 generator 的 embedding 旁邊，以及 discriminator 最中間 code 的旁邊。除此之外，在這兩層後我各加了一層 fully-connected layer 來讓 condition 的部分充分 apply 到 random 產生的 embedding 中。

  Loss function 的部分只有更改 discriminator 而已，我讓 discriminator 在看到 real image with wrong condition 時會讓 loss 升高，但由於這樣做會使得 network 自己失去 autoencoding 的能力，因此我們將 loss 控制在 real image loss 的 1.5 倍。Generator 在 minimize loss 時也會逐漸學習到 discriminator 在 right condition 的時候 loss 會比較低，因而產生正確的圖象。

  不過這個方法卻會產生很嚴重的 mode collapse，推測是因為描述圖象的部分通通被 encode 進 condition embedding 裡面了，而 condition embedding 是沒有隨機性的。

- Out of bound loss

  在一開始實作完 BEGAN 時所產生的圖象常常會有白斑點或黑斑點，經過分析發現這些斑點的值已經超出範圍(0~255)。因此我在 generator 的 loss 中加入一項 out of bound loss，實作上就是簡單的二次函數，只有在有超出範圍的時候才有這個 loss，超出越多 loss 越大。用這個方式可使斑點消失。



- Real Image Threshold

  因為在觀察 real image 時發現有很多 noise data，譬如說不是臉的圖片、以及色調很淡的圖片，因此我在每個 epoch 前會設一個 real image

threshold，這個值是由上一個 epoch 的平均 real loss 加上兩倍標準差所得到的，在這個 epoch 中會將 real loss 大於 threshold 的圖片直接移除，以避免 noise data 影響 AutoEncoder/discriminator。

Experiment settings and observation

Batch size:	16
Filter base dim:	128
Latent dim:	64
Condition emb dim:	16
Gamma:	0.4
Lambda:	0.001
Learning rate:	0.00005 (divided by 0.5 when no improvement)
Epoch:	127 (let it go for 3 days…)

實驗結果發現 BEGAN 產生的圖片較為模糊，但在原始 paper 中所呈現的人臉生成的效果又差的多，推測是因為動漫人物的 diversity 比較高，因此 AutoEncoder 無法 train 的很好。這個推測可由最終收斂的 convergence measure(這篇 paper 所定義的一個 metric) 來觀察到，原 paper 最後的 convergence measure 落在 0.6 左右，但是我們的 model 最後 convergence measure 只有 0.11，代表我們的 autoencoder 其實沒有 train 起來。

我們所設計的 condition 成功的讓生成的圖象正確，但也產生嚴重的 mode collapse 問題。目前推測是因為原本的 z 直接被 model 當成 noise 了，而 condition embedding 卻又是 deterministic。如果有 future work 的話，我們應該會嘗試直接用 condition embedding 去取代原本的 z，並在後面加一層 noise layer，如此一來不管是 condition 還是 prior distribution 都不能被忽視掉。

Team Division

討論：陳啟中、艾弗里、蔣君涵、陳育維
程式：陳啟中、陳育維
報告撰寫：陳啟中