

Homework3 - Generative Models

- **DCGAN**

直接採用 <https://github.com/carpedm20/DCGAN-tensorflow> 的 code，沒有任何改良。Train 了兩天，結果如 figure 1



figure 1

- **Generated images**

在沒修改 network 架構的情況下，估計會影響品質的幾個地方:

1. *Dataset distribution*

我們希望模擬 dataset 的 distribution，但雖然這次主題是 indoor，但其實十個類別裡面的圖片差異頗大，就算是同個類別下類型也會多元，是否真的在高維度存在可捕捉的 manifold 值得研究。

如果希望比較好的結果，可能可以先跑 classifier，事先判斷哪些是被 classifier 認定為 indoor 的圖片，在使用這些圖片做為 training 的資料，這樣的篩選應該有助於減少原本 dataset 過於紛亂的分布。但這樣就有點是對 dataset 動手腳了，所以並沒有使用。figure 2 是在 category bedroom 下的三張圖片。



figure 2

2. Mini-Batch

為了時間跟空間上的考量，一次為一個 Batch 的 update，而原本 batch size 是 64，我們又用了更少只有 16。這樣可能會導致效果不好，而且受到當前 Batch 的影響非常大。figure 3 是同樣的 z 在不同 batch update 後產生的圖片。

推測 batch 數目大一點會比較 general，也比較不會被 batch 裡面出現一些奇怪的圖就受到影響，像是某種色調、光影還有 texture (線條、紋路...)

不過怎麼知道當前 batch 有沒有奇怪、落在 distribution 外的圖片，就需要 supervise 介入看 training 圖片，所以我們最後也沒使用



figure 3

3. Selected images

理論上生出的無限圖片中，只要交出能夠騙過 classifier 跟 discriminator 的圖片，應該就能滿足又逼真又跟 dataset 相同 distribution 的圖片。但這有點像後處理，為了測試 generator 是否有效的學習到如何轉換，我們單純隨機在 latent space z 選了 500 個生成圖片就上傳了

● Discriminator

直接測試我們的 DCGAN 中的 discriminator，理論上原本對於自己的 generated image 應該有 50%的鑑別率。

不改 network 架構的情況下，觀看 14 組的 generated images，加強訓練 discriminator，應該能大幅提升準確度。但這就是偷看 testing data 了，所以也沒使用。

Contribution: 簡瑞霆