

Machine Learning Homework2

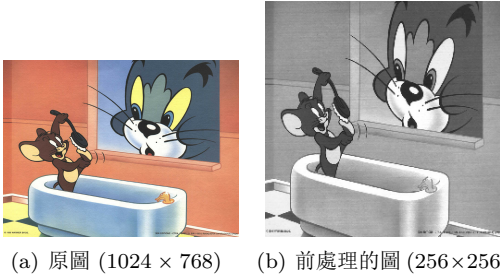
張格恩
應用數學系
國立中興大學

I. DATA PREPROCESSING

A. cartoon 圖的前處理

將所有圖片轉成灰階圖，並將每張圖片 resize 成 256×256 。

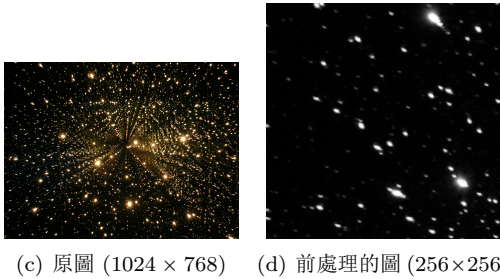
原圖是 1024×768 的彩圖，前處理後的圖是 256×256 的灰階圖。



B. texture 圖的前處理

將所有圖片轉成灰階圖，並先計算出數據中最小的圖片的維度，並以其維度來擷取所有圖片的左上角區塊，最後將擷取後的圖片 resize 成 256×256 。

原圖是 1024×768 的彩圖，前處理後的圖是 256×256 的灰階圖。



B. 實驗設定

Dimension of D: 256×1024 ; Dimension of A: 1024×1 ; λ (1-norm of A 的係數): 0.65; learning rate: 10^{-4} ; steps: 20000。

Objective function 如下:

$$\operatorname{argmin}_{D,A} \left(\frac{1}{2} \|Y - DA\|_2^2 + \lambda \|A\|_1 \right)$$

C. 模型

將所有的切割圖依序用 sparse coding 和 dictionary learning 交替訓練；一張切割圖收斂後，再用下一張圖來訓練 D，直到訓練完所有圖片的切割圖。

D 用標準常態分布做初始值設定。A 用 Xavier 做初始值設定，之後的則是用前一張圖訓練的 A 當做初始值設定。

1) *sparse coding*: 將標準常態分布的 D_{sc} 用 placeholder 放入。將 Xavier 初始值的 A_{sc} 用 variable 放入。用 D_{sc} 和 A_{sc} 計算出切割圖的預測值 (\hat{y}_i^{sc})，用預測值 (\hat{y}_i^{sc})，切割圖 (y_i) 以及 A_{sc} 來求出 loss，並用 gradient descent 更新 A_{sc} ，將 A_{sc} 輸出。

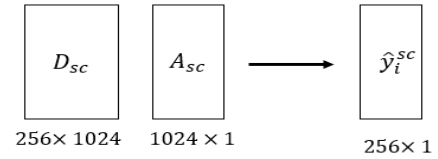


Fig. 1: model of sparse coding

2) *dictionary learning*: 將 sparse coding 的 D 當作 D_{dl} ，並用 variable 放入。將 sparse coding 訓練的 A 當作 A_{dl} ，並用 placeholder 放入。用 D_{dl} 和 A_{dl} 計算出切割圖的預測值 (\hat{y}_i^{dl})，用預測值 (\hat{y}_i^{dl})，切割圖 (y_i) 以及 A_{dl} 來求出 loss，並用 gradient descent 更新 D_{dl} ，將 D_{dl} 輸出並當作 sparse coding 的 D。

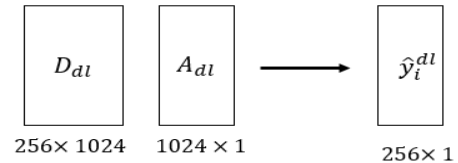


Fig. 2: model of dictionary learning

II. LEARN TWO DICTIONARIES RESPECTIVELY FOR THE DATA IN EACH FOLDER

A. 前處理

將所有 256×256 的灰階圖片做 partition 成 256 個 16×16 的切割圖，並將所有切割圖做 normalization，也就是把每個 pixel 除以 255，最後將其 reshape 成 1×256 。

最後，分別對 cartoon 圖和 texture 圖，重複 sparse coding 和 dictionary learning 來訓練各自的 D。

D. 結果和討論

1) *cartoon* 圖: 所有 256×256 的 *cartoon* 圖訓練的 dictionary。



Fig. 3: dictionary of cartoon 圖

2) *texture* 圖: 所有 256×256 的 *texture* 圖訓練的 dictionary。

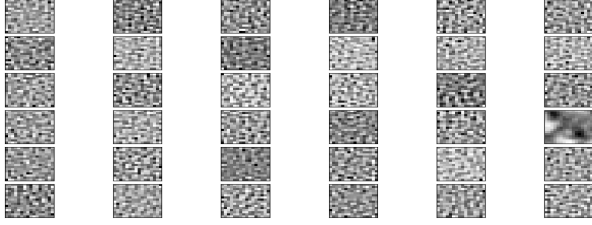


Fig. 4: dictionary of texture 圖

第四張 64×64 的 *texture* 圖訓練的 dictionary。

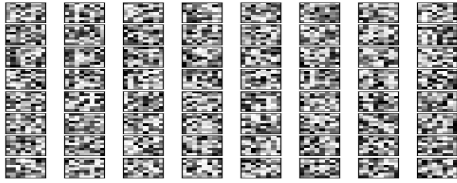


Fig. 5: dictionary of 第四張 texture 圖

我認為同一張圖，其圖片的大小及其包含的圖片細節會影響我們訓練 D 和重建原圖。

我用 64×64 的 *texture* 圖來訓練新的 D，並希望用 D 重建 64×64 的 *texture* 圖。在實作時，用圖片的每個 partition 所訓練的 D_i 和 A_i 來重建圖片每個 partition 的預測值 (\hat{y}_i)，再將所有 \hat{y}_i 重組回 \hat{y} ，並和原圖做比較。有些圖只需較少的訓練就能重建，有些圖訓練很久也只能生成雜訊。

之後，我會實驗用 *cartoon* 圖和 *texture* 圖訓練的 dictionary，嘗試來將 mixture 圖分成 *cartoon* 圖和 *texture* 圖。

III. LEARN TWO CONVOLUTIONAL DICTIONARIES RESPECTIVELY FOR THE DATA IN EACH FOLDER

因為時間關係，因此我會把圖片調整成 64×64 ，並取出前 2 張 *cartoon* 圖來訓練 D of *cartoon* 圖。

A. 前處理和實驗設定

取出前 2 張 *cartoon* 圖，對 64×64 的灰階圖片做 normalization，也就是把每個 pixel 除以 255。

Dimension of D: $8 \times 8 \times 1 \times 128$; Dimension of A: $64 \times 8 \times 8 \times 1$; λ (1-norm of A 的係數): 0.65 ; learning rate: 10^{-5} ; steps: 5000。

Objective function 如下:

$$\operatorname{argmin}_{d_i, a_{i,j}} \sum_l \frac{1}{2} \left\| y_l - \sum_m d_m * a_{l,m} \right\|_2^2 + \sum_m \lambda \|a_{l,m}\|_1$$

B. 模型

將 2 張 *cartoon* 圖依序用 sparse coding 和 dictionary learning 交替訓練；第一張圖收斂後，再用第二張圖來繼續訓練 D。D 用標準常態分布做初始值設定。A 用 Xavier 做初始值設定，之後的則是用前一張圖訓練的 A 當做初始值設定。

1) *Sparse coding*: 將標準常態分布的 D_{sc} 用 placeholder 放入。將 Xavier 初始值的 A_{sc} 用 variable 放入。用 D_{sc} 和 A_{sc} 計算出原圖的預測值 (\hat{y}^{sc})，用預測值 (\hat{y}^{sc})，原圖 (y) 以及 A_{sc} 來求出 loss，並用 gradient descent 更新 A_{sc} ，將 A_{sc} 輸出。

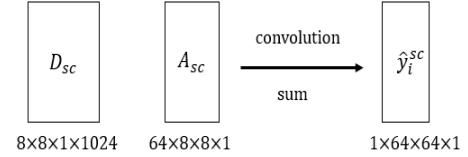


Fig. 6: model of sparse coding

C. Dictionary learning

將 Sparse coding 的 D 當作 D_{dl} ，並用 variable 放入。將 Sparse coding 訓練的 A 當作 A_{dl} ，並用 placeholder 放入。用 D_{dl} 和 A_{dl} 計算出原圖的預測值 (\hat{y}^{dl})，用預測值 (\hat{y}^{dl})，原圖 (y) 以及 A_{dl} 來求出 loss，並用 gradient descent 更新 D_{dl} ，將 D_{dl} 輸出並當作 sparse coding 的 D。

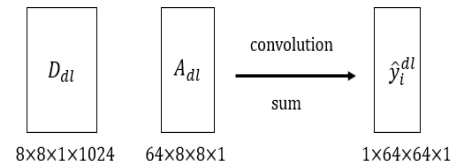


Fig. 7: model of dictionary learning

最後，對這兩張 *cartoon* 圖，重複用 sparse coding 和 dictionary learning 來訓練 D。

D. 結果和討論

前兩張 64×64 的 *cartoon* 圖訓練的 dictionary。

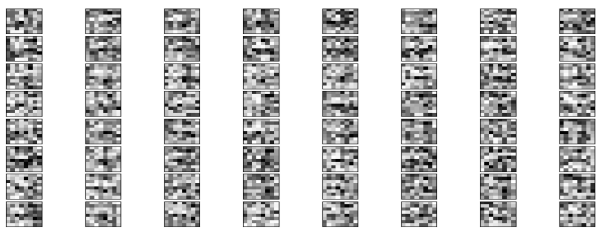


Fig. 8: dictionary of cartoon 圖

之後，我會實驗用同一張圖片但 resize 成不同的大小來做 convolutional dictionary learning，並且利用訓練好的 dictionary 來重建原圖以及將 mixture 圖分成 cartoon 圖和 texture 圖。

IV. SHOW THE SPARSE APPROXIMATIONS OF EACH IMAGE USING THE LEARNED DICTIONARIES

因為時間關係，因此我用第二章的 D 來生成部分的 cartoon 圖和 texture 圖。

A. 前處理

將所有 256×256 的灰階圖片做 partition 成 256 個 16×16 的切割圖，並將所有切割圖做 normalization，也就是把每個 pixel 除以 255，最後將其 reshape 成 1×256 。

B. 實驗設定

Dimension of D: 256×1024 ; λ (1-norm of A 的係數): 0.65 ; learning rate: 10^{-4} ; steps: 10000 ; 限制的值 (δ): 10^{-8} 。

Objective function:

$$\operatorname{argmin}_{a_i} \sum_i \|y_i - Da_i\|_2^2 + \lambda \|a_i\|_1$$

i 代表一張圖的第 i 個切割圖。

C. sparse coding

將第二章訓練好的 D 用 placeholder 放入，將 Xavier 初始值的 A 用 variable 放入。第一步訓練 A 時，不用限制值來讓 A 變成 sparse ($<$ 限制的值變成 0)。第二步之後，我則會讓 A 在 forward 時變成 sparse，接著，做 backward 來更新 A。用第二章的 D 和訓練的 A 計算出切割圖的預測值 (\hat{y}_i)，用預測值 (\hat{y}_i)，切割圖 (y_i) 以及 A 來求出 loss，將 A 和 $loss_{sc}$ 輸出。最後，將所有切割圖的預測值組成原圖的預測值。

D. 結果和討論

1) *cartoon* 圖：原圖和利用第二章 cartoon 圖的 dictionary 所重建的圖。

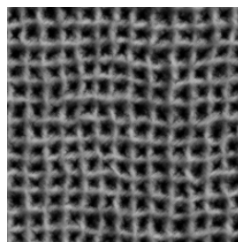


(a) 原圖 (256×256)

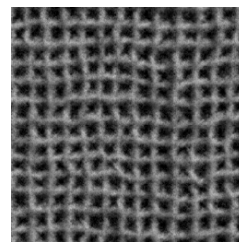


(b) 重建的圖 (256×256)

2) *texture* 圖：原圖和利用第二章 texture 圖的 dictionary 所重建的圖。



(c) 原圖 (256×256)



(d) 重建的圖 (256×256)