

Deep Learning for Computer Vision

Homework 1

蘇楷鈞 電機碩一 r06921062

Problem 1

$$\begin{aligned} p(x|w_1) &= \frac{1}{5} & p(w_1) &= \frac{3}{4} \\ p(x|w_2) &= \frac{1}{3} & p(w_2) &= \frac{1}{4} \\ P_e &= \int_T^5 p(x|w_1)p(w_1)dx + \int_3^T p(x|w_2)p(w_2)dx \\ &= \int_T^5 \frac{1}{5} \times \frac{3}{4} dx + \int_3^T \frac{1}{3} \times \frac{1}{4} dx \\ &= \frac{15-3T}{20} + \frac{T-3}{12} \\ &= \frac{30-4T}{60} = \frac{1}{2} - \frac{T}{15} \\ \min_{3 \leq T \leq 5} P_e &= m \left(\frac{1}{2} - \frac{T}{15} \right) \\ \therefore \text{當 } T=5 \text{ 時 } P_e &= \frac{1}{6} \text{ 為最小值} \\ R_1 &\text{ 為 } 0 \leq x \leq 5 \\ R_2 &\text{ 為 } 5 < x \leq 6 \end{aligned}$$

Problem 2

a) 左圖(Fig.1)為 mean face，右圖(Fig.2)為前三大特徵值的 eigenface。

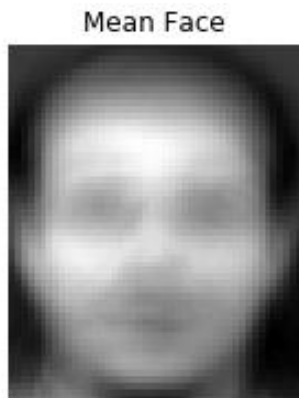


Fig.1



Fig.2

b) 下圖(Fig.3)由左至右為：person1_image1.png 原圖，依序分別為用 n 個 eigenfaces 所還原的，下方為重建後與原圖的 MSE。

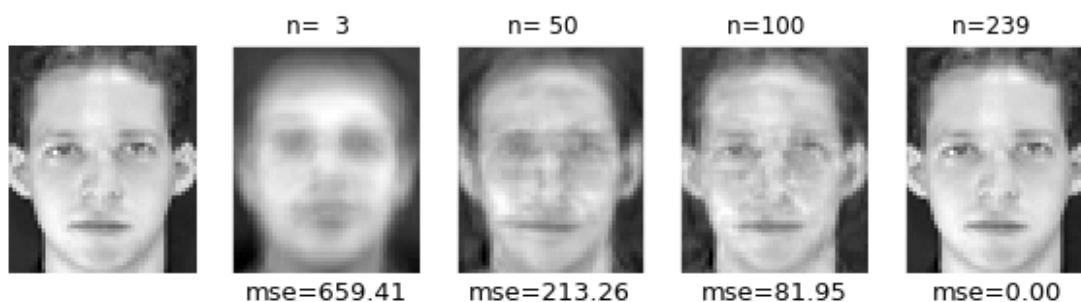


Fig.3

c) 使用 sklearn 中 KNeighborsClassifier 來訓練模型，結果如右圖(Fig.4)，每兩行為一次實驗，其中第一行為 KNeighborsClassifier 參數(n -dim 為使用多少個特 eigenface)，第二行前三個值為 3-fold cross-validation 的實驗結果，最後直接取平均來比較。經多次實驗後發現使用 cross-validation 在每輪 fold 中大部分都不會 overfit，因此直接以平均準確率來選擇最佳模型，又在 $n=50$ 和 159 時效果不相上下，因此選擇少的 eigenface 來時做較為省空間成本，因此 hyperparameter 將用 $k=1$ 、 $n=50$ 、使用歐式距離來訓練模型，而以此模型去測試測試集時，準確率為 0.8875。

```
k-neighbor : 1 , n-dim : 3
['0.600', '0.650', '0.662'] 0.6375
k-neighbor : 1 , n-dim : 50
['0.900', '0.900', '0.900'] 0.9000
k-neighbor : 1 , n-dim : 159
['0.900', '0.875', '0.900'] 0.8917
k-neighbor : 3 , n-dim : 3
['0.600', '0.650', '0.625'] 0.6250
k-neighbor : 3 , n-dim : 50
['0.838', '0.825', '0.775'] 0.8125
k-neighbor : 3 , n-dim : 159
['0.812', '0.812', '0.775'] 0.8000
k-neighbor : 5 , n-dim : 3
['0.600', '0.613', '0.575'] 0.5958
k-neighbor : 5 , n-dim : 50
['0.713', '0.787', '0.762'] 0.7542
k-neighbor : 5 , n-dim : 159
['0.700', '0.713', '0.700'] 0.7042
```

Fig.4

Bonus

當 A 為對稱矩陣，且各個特徵值都不一樣時，可採用 Power 迭代法去逼近最大特徵向量。右圖(Fig.5)為向量更新時，針對每次更新向量計算第 2 步驟的誤差圖。

Algorithm:

給定對稱矩陣 A 、初始向量 v_0

1. For $i = 0$ to k :

a) 計算 $v_{i+1} = Av_i$

b) $v_{i+1} = \frac{v_{i+1}}{\|v_{i+1}\|_2}$

c) 計算 $\lambda^* = v_{i+1}^T A v_{i+1}$

d) End for

2. 檢查 λ^* 是否收斂，收斂時 v_{i+1} 即為第一個特徵值最大之特徵向量。如果沒收斂則回到步驟 1.

3. 判斷 λ^* 大於 0，則 v_{i+1} 即為所求。

如果小於 0 則計算 $A^* = A - \lambda^* I_d$ ， I_d 為單位矩陣。

4. 將 A 替換成 A^* 再執行步驟 1.、2.，求得特徵向量即為所求

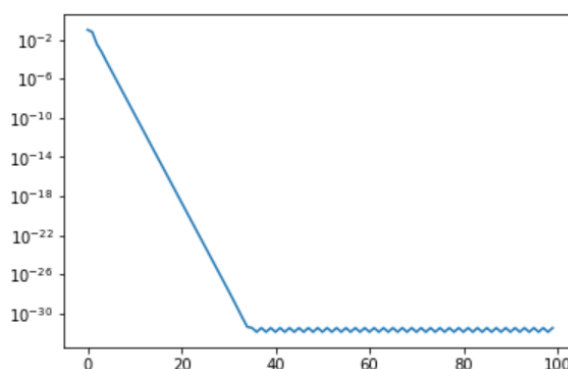


Fig.5

Proof

步驟(1, 2)

給定對稱矩陣 A 維度為 $d * d$ 且其特徵值皆不同及隨機向量 v_0 ，由於 A 為對稱矩陣，因此可正交對角化

$$\forall i = 1 \dots d, x_i \in Ax_i = \lambda x_i \rightarrow \forall i \neq j, x_i \perp x_j$$

因此任意向量皆可分解為以特徵向量為基底的組合且假設 $|\lambda_1| > |\lambda_2| > |\lambda_3| > \dots$ 。

$$v_0 = \sum_{i=1}^d a_i x_i$$

$$\text{則 } v_k = A^k v_0 = \sum_{i=1}^d a_i \lambda_i^k x_i = \lambda_1^k \sum_{i=1}^d a_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i$$

$$\text{隨著 } k \text{ 增大 } \left(\frac{\lambda_i}{\lambda_1}\right)^k \text{ 會趨近於 } 0, \text{ 因此 } v_k = \frac{Av_k}{\|Av_k\|_2} \approx \frac{\lambda^k a_1 x_1}{\|\lambda^k a_1 x_1\|_2} = \pm x_1$$

步驟(3)

$$A^* = A - \lambda^* I_d$$

$$\det(A^* - x I_d) = \det(A - \lambda^* I_d - x I_d) = \det(A - (\lambda^* + x) I_d)$$

令 λ^p 為 A^* 之特徵值，則 $\lambda' = \lambda^p + \lambda^*$

$$\begin{aligned} N(A^* - \lambda^p I_d) &= N(A - \lambda^* I_d - \lambda^p I_d) = N(A - (\lambda^* + \lambda^p) I_d) \\ &= N(A - \lambda' I_d) \end{aligned}$$

因此 A^* 的特徵值都和 A 的特徵值相差 λ^* ，故如果 A 的最大特徵值 λ^* 小於0，則將所有特徵值位移 $-\lambda^*$ ，使得所有特徵值皆大於0，即可由步驟(1, 2)找出 A^* 最大特徵值 λ^p ， A 的最大特徵值即為 $\lambda' = \lambda^p + \lambda^*$ ，且 $N(A^* - \lambda^p I_d)$ 和 $N(A - \lambda' I_d)$ 相等，因此由 A^* 計算出的向量即為最大特徵值所對應的特徵向量。