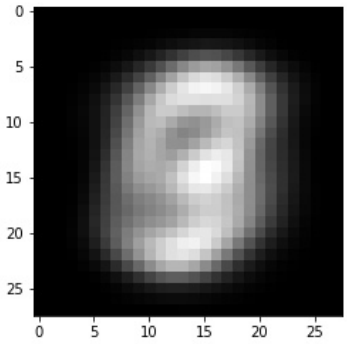
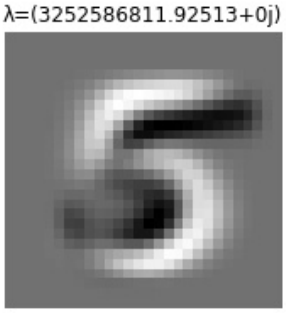
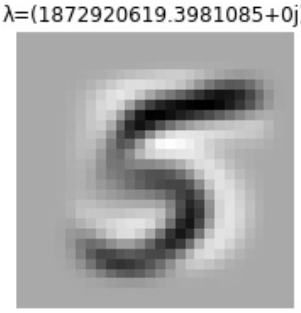
Part 1 : PCA

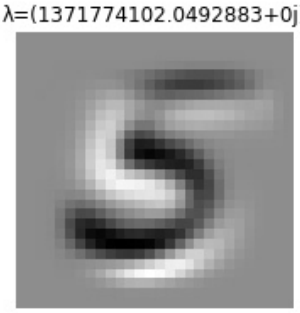
Q1 : Show the mean of all 70,000 images.



Q2 : Extract all the '5' images (6313 vectors). Use centered-PCA (5's center) to decompose. Show eigenvectors with the 3 largest eigenvalues. Show the corresponding eigenvalues as well.



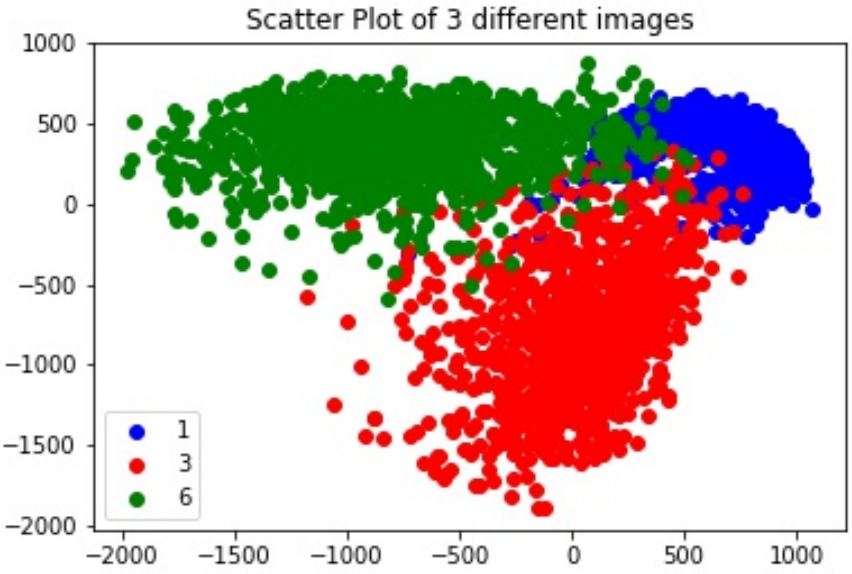




Q3 : Extract all the '5' images. Use centered PCA and the top {3,10,30,100} eigenvectors to reconstruct the first '5' image. Explain your results.

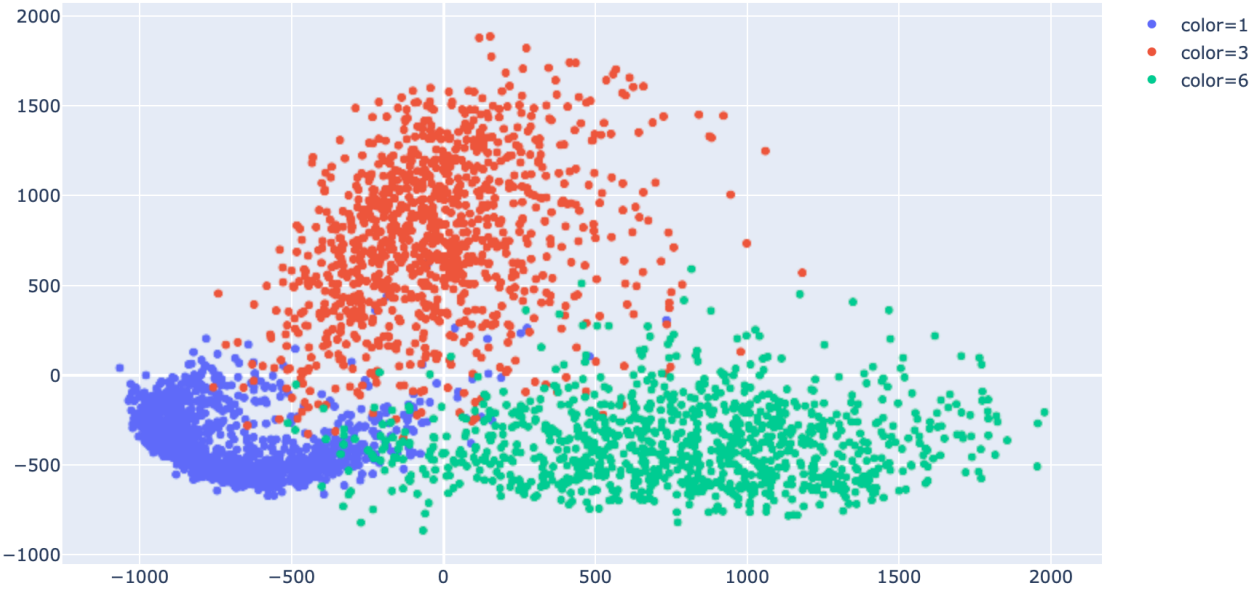
從六千多筆的 dictionary 中選取 eigenvalue 最大的三個基底對影像進行重建，顯示的圖像跟 Q2 的重合度很高。選取前十個基底時，還是顯示的 ‘5’ ，但與原始圖像的形仍不像。選取前三十個基底之後， ‘5’ 的形開始有了明顯的改變。直到基底數增加到一百個，重建的圖像與原始圖像非常相似。

Q4 : Extract the first 10,000 images. Next, extract all the '1,3 and 6' (from 10k images). Use centered-PCA to reduce the dimensions from 784 to 2 (the 2 largest eigenvalues). Plot those points in a 2-D plane using plt.scatter function with different colours. Explain your results.



Q4 result (created from my handcraft PCA)

上圖是用我自己寫的 PCA 進行降維之後畫出來的圖。下圖是我呼叫 sklearn 的 PCA 進行降維畫出來的圖。兩張圖標記為 1、3 和 6 的顏色剛好都對得上，我覺得我的 PCA 可能跟現有公開的 PCA 找出來的的 first and second principle axes 都差了負號。需要（1）左右 + 上下翻轉，或（2）從原點 (0,0) 旋轉 180 度。從這兩張圖中，不難發現原本 784 維的資料經 PCA 降成兩維之後，同一類數字呈現在平面上是群聚的。

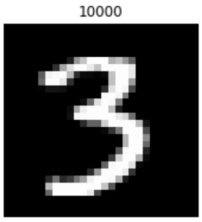


Reduce 1, 3 & 6 to 2D (using sklearn.decomposition.PCA)

Part 2 : OMP

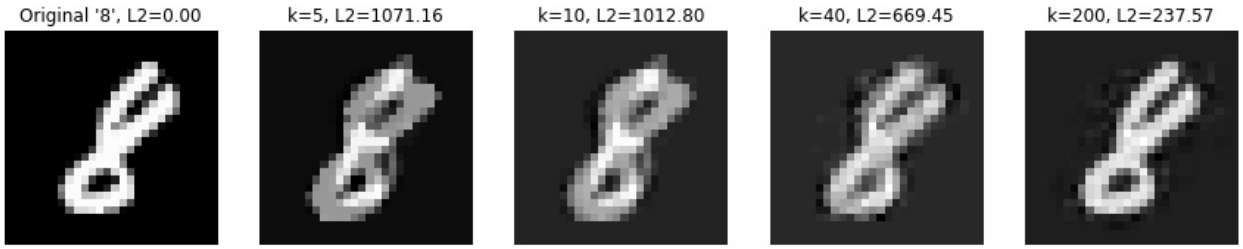
Define the first 10,000 images as training set.

Q5 : Find the 5 bases of the #10001 image ('3') with sparsity=5. Show the 5 bases. What do you observe?



上圖是從不同數字的 1 萬筆 dictionary 中找與 #10001 （索引：10000）內積之後取絕對值最大的五個基底（圖像上標的是基底的索引）的原始圖像。觀察發現這五個基底至少都跟 #10001 屬於同類標記，都是數字 '3'。他們之間沒有巨明顯的差異，肉眼看都不會跟其他數字混淆。

Q6 : Find the bases of the #10002 image ('8') with sparsity={5,10,40,200}. Show the reconstruction images. Calculate their reconstruction errors using L2-norm (Euclidean distance). Explain your results.



PCA 考慮 m<n （基底數量 < 資料維度）的情況，所以在假設訓練資料集固定的情況下，選擇哪些基底來表示訊號的優先順序是不變的（按照 eigenvalue 的大小進行選取）。而 OMP 是考慮 m>n（基底數量 > 資料維度）的情況，在眾多經過 normalized 的基底當中選擇與原始訊號相似度最高（內積取絕對值，可以把資料表達得最好）的 k 個作為基底。即使在訓練資料集固定的情況下，不同的訊號使用 OMP 選擇激活的基底也各不相同。

個人觀點：觀察 Q3 的結果，PCA 用前三、十個基底來表達 #0 的 '5' 時，其重建圖像的形狀與原始圖像有一定差距；反觀這一小題的 OMP 用五或十個基底表達 '8' 就能達到 PCA 前三十個基底表達 '5' 的程度，有些模糊但線條形狀較為相似。但你也可以認為他們之間沒有可比性，因為 Q3 PCA 的基底全是 '5'，而 Q6 基底雖有一萬張，但都不同類。

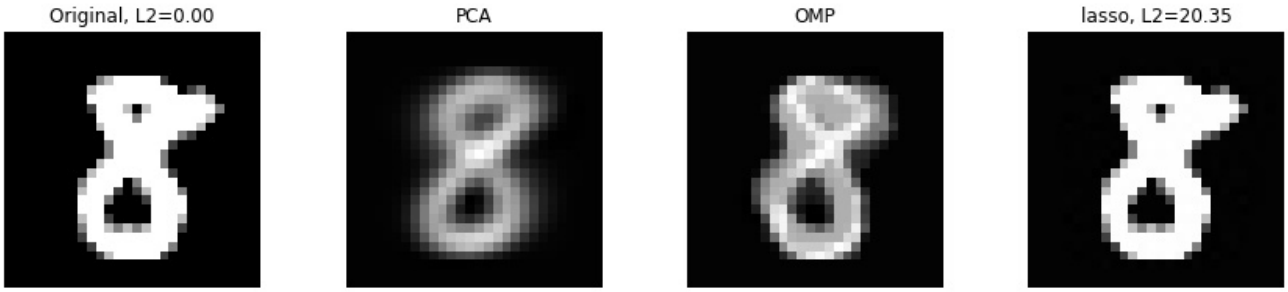
單單就 OMP 而言，隨著可激活的基底數量 k 的增加，重建訊號與原始訊號的誤差（L2norm）明顯地減小。但整體的誤差也不小，因為 OMP 的基底必須經過 normalization，

reconstruct image （normalized dictionary B 和 coefficient c 的線性組合）與原始訊號（沒有 normalized）的誤差自然不小。

Part 3 : lasso

Q7 : Extract all the '8' images from the dataset (6825 vectors).

1. Use centered-PCA to reconstruct the last '8'. (Remain 5 largest eigenvalues)
2. Use the first 6824 images as the base set. Use OMP to find the bases and reconstruct the last '8'. (Sparsity=5)
3. Use the first 6824 images as the base set. Use lasso to find the bases and reconstruct the last '8'.



PCA 和 OMP 都只用了五個基底重構訊號，而 Lasso 用了九百二十個，所以其重構的圖像與原圖像最相近也是情有可原。

4. Adjust the lasso parameters. Explain your experiments and results.

Lasso objective function : $$\min_{c} \frac{1}{2}||x - Ac||_2^2 + \alpha||c||_1$$

Q7 的第三小題跑 lasso 使用預設的 alpha 為 1，其 L2-norm reconstruction error 近似 20.35（於下表以**藍色**標示）。我在這一小題的實驗設計：（1）未進行預處理的資料，設定不同的 alpha 值對最後一筆 '8' 進行 reconstruction；（2）經過 sklearn.preprocessing.StandardScaler 進行 normalise 預處理，設定不同的 alpha 值進行 reconstruction。結果如下表所示：

	α	**0.001**	**0.01**	**0.1**	**1**	**10**	**100**
No pre-processing	L2norm	4.982155	4.725386	3.807011	**20.345078**	127.236957	497.163460
	No. of Coef != 0	6824	6661	2743	**920**	333	100
Standard Scaler()	L2norm	1.080250	4.307277	8.056076	28.000000	28.000000	28.000000
	No. of Coef != 0	365	119	25	0	0	0

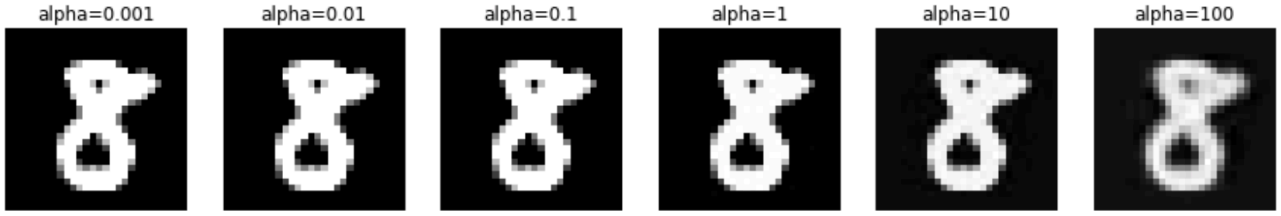
* L2-norm 為原訊號與重構訊號之間每一維度的差，取絕對值的平方總和，再開根號。

* StandardScaler normalisation 則是計算 normalized 原始訊號與 normalized 重構訊號間的 L2norm。

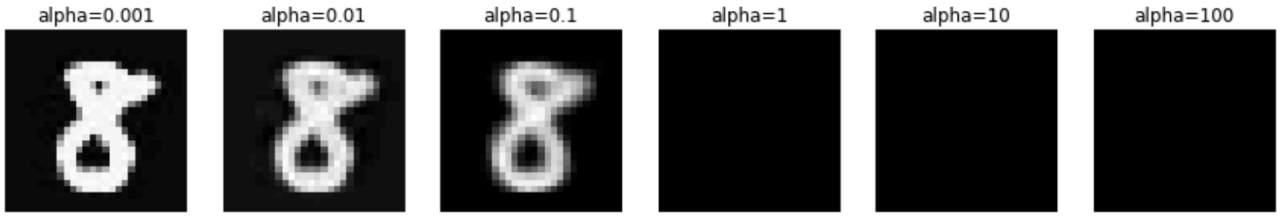
從上表，觀察到無論是否經過預處理，隨著 alpha 值的上升（加重懲罰），被激活的基底數量都呈下降的趨勢。Lagrange multiplier - alpha 的值越大，給予 c 的 L1norm 的懲罰越重，導致被激活的基底數越少，大多數的基底係數為零。

實驗（2）由於所有資料和需要重構的最後一個 ‘8’ 都經過 normalized，因此計算得出的 L2-norm 都較低。但（2）整體而言，激活的基底數都比（1）少，乃至 alpha 值上升到 1 以後，選擇的基底皆為零。

實驗（1）不同 alpha 值重構最後一筆 ‘8’ 的結果圖：



實驗（2）不同 alpha 值重構最後一筆 ‘8’ 的結果圖：



實驗（2）取 alpha = 0.1 只用了二十五個（實驗中最少的）基底來重構訊號，至少看得出是 ‘8’。但若與 Q7 第一小題相比較的話，兩者圖像的重構效果差異不大，但 PCA 就只用了五個基底。

實驗（1）alpha = 100 與實驗（2）alpha = 0.01 ，前者動用了一百個基底·，後者動用了一百一十九個，兩者重構出來的圖像效果以肉眼來衡量是差不多的。