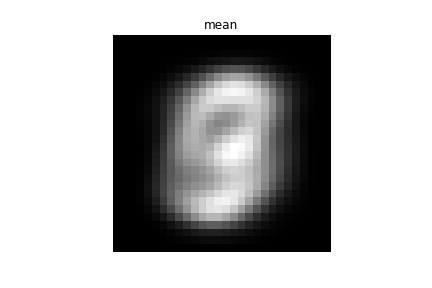
dsp2021\_phw1

R10922129 劉旭庭

Q1:

把70000張image累加在一起然後除以70000



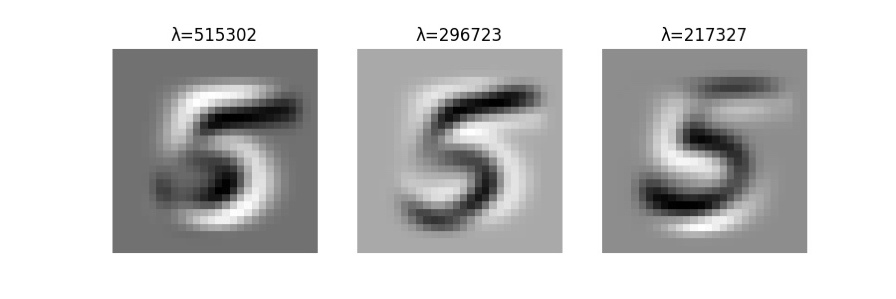
Q2:

使用到手刻cpca(centered pca) 函式，傳入要作pca的資料x，會先算一次mean，把x的每一筆資料都減掉mean，然後找出eigenvalue, eigenvector，隨後把eigenvalue轉正(對應的eigenvector也會隨之作sign的處理)

另外會對eigenvalue作argsort輸出成vector m，越大的eigenvalue index會排越前面。

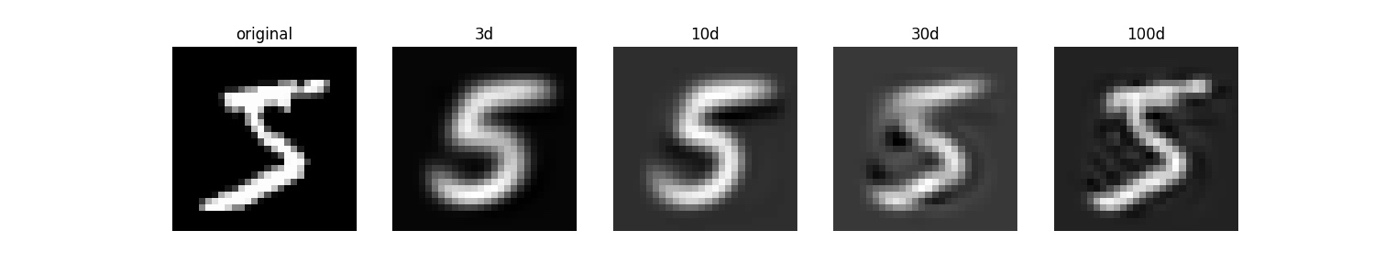
而x在處理完後會變成減去mean的形式。

以下是找出來eigenvalue最大的三個eigenvector



Q3:

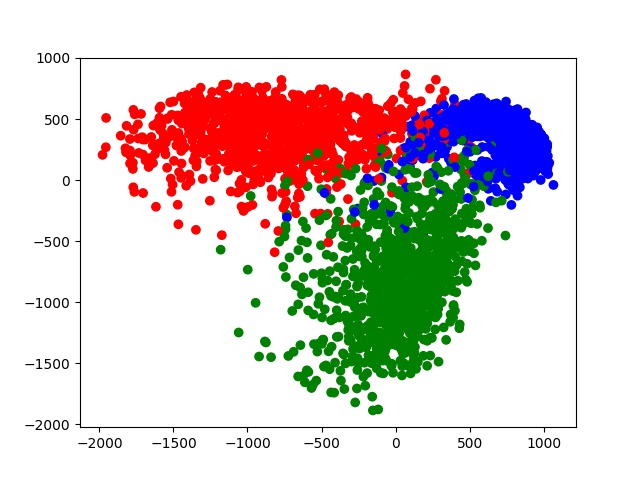
類似Q2，在重建時會利用vector m，找出前N大的eigenvector，並且把要重建的target image投影到eigenvector space 上，最後再把mean加回去。



Q4:

把1,3,6的data搜集起來後，作cpca，找出前2大的eigenvalue對應的eigenvector，分別計算每個資料點對應到的係數，隨後畫成圖。

1:blue, 3:green, 6:red



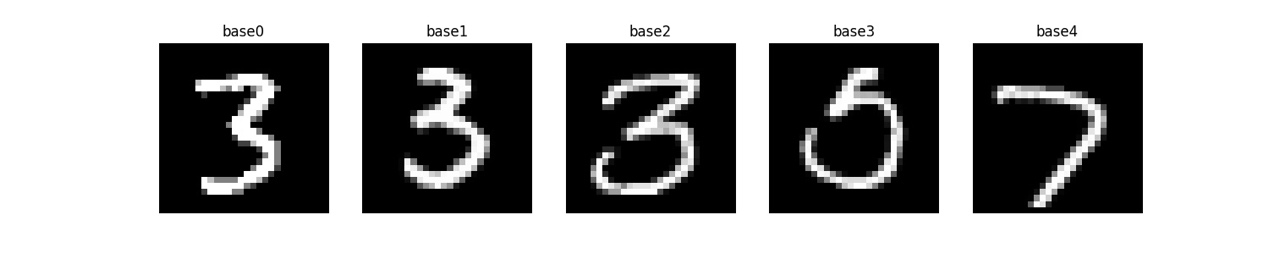
Q5:

使用手刻omp函式，會先對training data作歸一化，建立use vector，表示data是否已經被選作basis，已經被選到的use會是False

依照內積絕對值的大小找出k個basis會依序append到b，用來算每一輪的r

r=x-(projection x on b)。

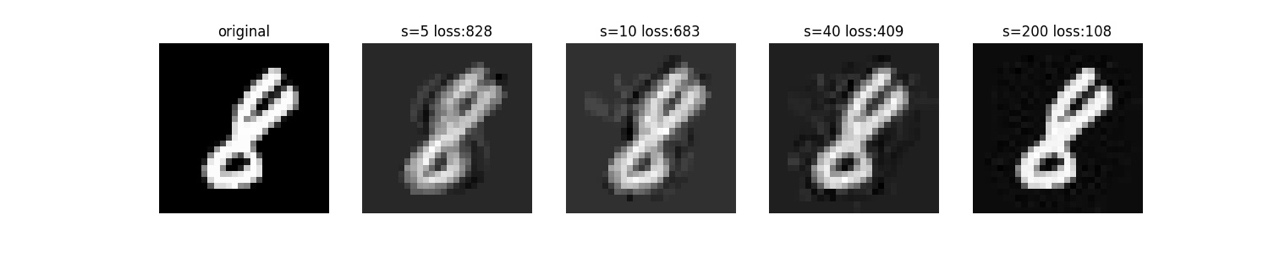
最後函式只會return basis的index。 在函式外透過index找basis



Q6:

原理同Q5，只是重建時將target image投影到basis的space上

s為sparsity縮寫，loss 為L-2 norm



Q7:

1. 作法如Q3
2. 做法如Q6
3. Call sklearn的inear\_model.Lasso

參數: alpha=2.3, normalize=True, copy\_X=True

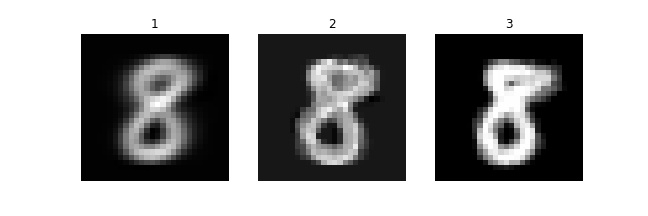
fit完之後，對coef\_作argsort，依據係數大到小對index作排列，利用排列後的index重建target image進行投影。

1. 對於參數的實驗，隨著alpha值越大，regularization的penalty越大，非零細數也越來越少

Alpha :1, 2, 2.3

非零係數:18, 8, 5

下圖分別對應Q7-1,2,3的重建結果



Bonus:

code fragment:

*def* soft(*a*,*c*,*l*):

if c < -l:

return (c+l)/a

elif c>l:

return (c-l)/a

else:

return 0

*def* lasso(*orix*,*oriy*,*l*):

x=orix\*1

y=oriy\*1

ii,jj=x.shape

for j in range(jj):

x[:, j] /= (np.inner(x[:, j], x[:, j])\*\*0.5)

y/=(np.inner(y,y)\*\*0.5)

tempw=np.zeros(jj)

a = np.zeros(jj)

c = np.zeros(jj)

s=x.T@x

invs=np.linalg.pinv(s)

w=invs@x.T@y

while np.all(tempw!=w):

tempw=w\*1

for j in range(jj):

a[j] = 2\*np.inner(x[:,j], x[:,j])

c[j] = (2)\*np.sum(x[:, j]@(y-(x@w)+(x[:, j]\*w[j])))

w[j]=soft(a[j],c[j],l)

return w

z = lasso(vec8.T, last8, 1.5)

idx = np.argsort(-abs(z))

space = []

for i in range(5):

space.append(vec8[idx[i]])

space = np.array(space)

draw(project(last8, space.T),'handcraft')

Implementation explaining:

soft函式講義有提到，單純實作出來而已

lasso function裡面會先對資料以及target image作歸一化，並且對係數vector w初始，這邊初始為target image投影到training data的係數，初始化其實蠻影響結果的。

隨後依照用講義補充的內容去更新，在w沒有更動後認定為收斂，return w

重建的部分也是對w作argsort找到係數前五大的index，然後將target image投影上去做重建。

