Shantanu Pande -181060057

Data Science Lab Assignment-2

Problem Statement: We have studied the change of basis problem using Eigen analysis. Write a sample script (preferably user interactive) for N-dimensional (unique) data to implement the same.Perform a comparative analysis for complete and partial reconstruction(s) in terms of error. Comment on the result.Repeat the algorithm for a visual data, its perfect reconstruction and a few samples of partial reconstructions. The submission will contain a single pdf file (neither zipped/linked to drive nor colab) containing the data, script, observation, comparison and results.

Part-1 : Eigen analysis using 1D data

<u>Code</u>:

```
import cv2
import numpy as np
from numpy import linalg as la
#To accept image in normalized form
img=cv2.imread('/home/hp/Downloads/ds_image.jpeg')
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#X = gray_image
X = [[6,5,3,4,4,4,5,5],[5,6,4,3,4,5,5,4]]
N = gray_image.shape[0]
n = gray_image.shape[1]
#print("Covarience is : ")
C = np.cov(X,ddof=0)
print(C)

#Eigen values and vectors
w, v = la.eig(C)
v = v.T
#print("Eigen values")
#print(w)
```

```
#print("Eigen Vectors")
#print(v)
z = [x for _,x in sorted(zip(w,v),reverse=True)]
z = np.array(z)
#print("Index of Eigen value to be discarded: ")
X = np.array(X)
k = int(input("Number of eigen values to be discarded: "))
#k = int((k*X.shape[0])/100)
for i in range (k):
    z = np.delete(z,(z.shape[0]-1),axis=0)
x_ = np.zeros((X.shape[0],X.shape[1]), dtype = int)
y = np.dot(z,X)
y = np.array(y)
x_ = np.dot(z.T,y)
x_ = np.around(x_).astype(np.uint8)
#while True:
#    cv2.imshow("Orig",X)
#    cv2.imshow("Trans",x_)
#    if cv2.waitKey(0):
#        break
#cv2.destroyAllWindows()

E = (X-x_)**2
E = np.sum(E,axis=1)/X.shape[1]
E = np.sum(E,axis=0)
print("Error ", E)

disc_eigen = 0
l = X.shape[0]
for i in range(0,k):
    disc_eigen = disc_eigen + w[l-1-i]
print("Sum of discarded eigen Values " , disc_eigen)
```

Output:

Partial Reconstruction

```
[[0.75  0.375]
 [0.375 0.75 ]]
Number of eigen values to be discarded: 1
Error  0.375
Sum of discarded eigen Values  0.37499999999999999
```

Complete Reconstruction

```
[[0.75  0.375]
 [0.375 0.75 ]]
Number of eigen values to be discarded: 0
Error  0.0
Sum of discarded eigen Values  0
```

Observation and Calculations:

$$x = \begin{bmatrix} 6 & 5 & 3 & 4 & 4 & 4 & 5 & 5 \\ 5 & 6 & 4 & 3 & 4 & 5 & 5 & 4 \end{bmatrix}.$$

No. of data points = 8
dimension = 2

Now

$$E(x) = \begin{bmatrix} (\Sigma\ 1^{st}\ \text{row})/8 \\ (\Sigma\ 2^{nd}\ \text{row})/8 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 4.5 \end{bmatrix}$$

And,

$$x_1 = \begin{bmatrix} 6 \\ 5 \end{bmatrix},\ x_2 = \begin{bmatrix} 5 \\ 6 \end{bmatrix},\ x_3 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \dots$$

$$x_1 - E(x) = \begin{bmatrix} 6 - 4.5 \\ 6 - 4.5 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

Now,

$$(x_1 - E(x))(x_1 - E(x))^T = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \begin{bmatrix} 1.5 & 1.5 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 2.25 \end{bmatrix},\ C_3 = \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$$

$$C_4 = \begin{bmatrix} 0.25 & -0.25 \\ -0.75 & 2.25 \end{bmatrix},\ C_5 = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$$

$$C_6 = \begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix},\ C_7 = C_5,\ C_8 = \begin{bmatrix} 0.25 & -0 \\ -0.25 & 0. \end{bmatrix}$$

Now, covariance matrix

$$C = \begin{bmatrix} \Sigma C_{11}/8 & \Sigma C_{12}/8 \\ \Sigma C_{21}/8 & \Sigma C_{22}/8 \end{bmatrix}$$

$\frac{2.25 + 0.25 + 2.25 + 0.25 + 0.25 + 0.25 + \dots}{8}$

$$= \begin{bmatrix} 0.75 & 0.375 \\ 0.375 & 0.75 \end{bmatrix}$$

Now, to find eigen values,

$$l^2 - 1.5 l + 0.421825 = 0.$$

So,

$$l_1 = 1.125 \quad , \quad l_2 = 0.375.$$

eigen vectors are,

$$e_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \& \quad e_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Normalizing,

$$e_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \& \quad e_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\therefore A = \begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \end{bmatrix}$$

New coordinate system with Basis e_1 & e_2

$$\rightarrow A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{matrix} \rightarrow e_1 : \text{corresponds to highest eigen} \\ \rightarrow e_2 : \text{corresponds to lowest eigen} \end{matrix}$$

$$y = A \cdot x \leftarrow \text{original data}$$

transformed data — transformation or set of rules

Consider a point from a data,

Let, $x_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

so,

$$y = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 7 \\ -1 \end{bmatrix}$$

Now for reconstruction,

$$x' = A^{-1} \cdot y$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 7 \\ -1 \end{bmatrix}$$

$$x' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

So reconstruction is possible & it is [illegible]

→ Partial reconstruction.

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$x = \begin{bmatrix} 5 \\ 4 \end{bmatrix}_{2\times 1}$$ original data

let,

$$A_{new} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$y_{new} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$

$$y_{new} = \frac{1}{\sqrt{2}} \begin{bmatrix} 9 \end{bmatrix}_{1\times 1}$$ in new basis

Now,

$$\hat{x} = A_{new}^{-1} \, y_{new}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 9 \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 9 \\ 9 \end{bmatrix}$$

$$\hat{x} = \begin{bmatrix} 4.5 \\ 4.5 \end{bmatrix}_{2\times 1}$$ partial reconstructed data coming back to orig[inal]

→ Doing this for all points in x

$$x_1 = \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \quad \hat{x}_1 = \begin{bmatrix} 5.5 \\ 5.5 \end{bmatrix}$$

$$(x_1 - \hat{x}_1)^2 = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/4 \end{bmatrix}$$

Page No.
Date / /

$\therefore (x - \hat{x})^2 = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \end{bmatrix}$

Squared error

$E[(x - \hat{x})^2] = \begin{bmatrix} (6 \times 1/4)/8 \\ (6 \times 1/4)/8 \end{bmatrix} = \begin{bmatrix} 0.1875 \\ 0.1875 \end{bmatrix}$

$\therefore$ error $= 0.1875 + 0.1875$
$= 0.375$

This is the error when we ~~dont~~ discarded eigen vector $e_2^T = [1 \ -1]$ where $\lambda_2 = 0.375$.

So, this error after partial reconstruction will be equal to the eigen value of the eigen vector discarded.

If we discard $e_1 = [1 \ 1]$ while reconstructing, then the error,
Error $= \lambda_1 = 1.125$.

Part 2 : Visual Data

Code:

```
import cv2
import numpy as np
from numpy import linalg as la
#To accept image in normalized form
img=cv2.imread('/home/hp/Downloads/ds_image.jpeg')
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
X = gray_image
N = gray_image.shape[0]
n = gray_image.shape[1]
#print("Covarience is : ")
C = np.cov(X,ddof=0)
print(C)

#Eigen values and vectors
w, v = la.eig(C)
v = v.T
print("Eigen values")
#print(w)
print("Eigen Vectors")
#print(v)
z = [x for _,x in sorted(zip(w,v),reverse=True)]
z = np.array(z)
#print("Index of Eigen value to be discarded: ")
X = np.array(X)
k = int(input("Percent for reconstruction: "))
k = int((k*X.shape[0])/100)
for i in range (k):
    z = np.delete(z,(z.shape[0]-1),axis=0)
x_ = np.zeros((X.shape[0],X.shape[1]), dtype = int)
y = np.dot(z,X)
```

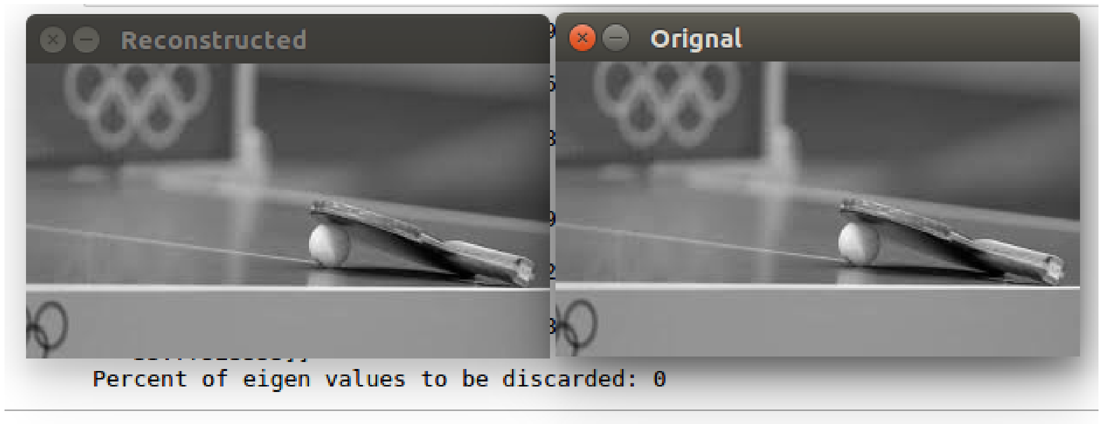
```
y = np.array(y)
x_ = np.dot(z.T,y)
x_ = np.around(x_).astype(np.uint8)
while True:
    cv2.imshow("Orig",X)
    cv2.imshow("Trans",x_)
    if cv2.waitKey(0):
        break
cv2.destroyAllWindows()

E = (X-x_)**2
E = np.sum(E,axis=1)/X.shape[0]
E = np.sum(E,axis=0)
print(“Error”, E)

disc_eigen = 0
l = X.shape[0]
for i in range(0,k):
    disc_eigen = disc_eigen + w[l-1-i]
print("Sum of discarded eigen Values " , disc_eigen)
```

Output:

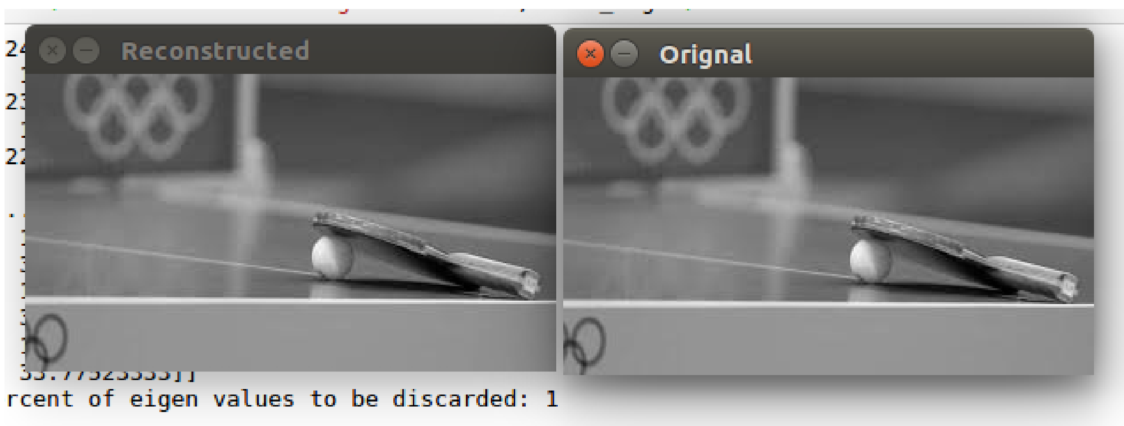
Complete Reconstruction



Error 0.0

Sum of discarded eigen Values 0

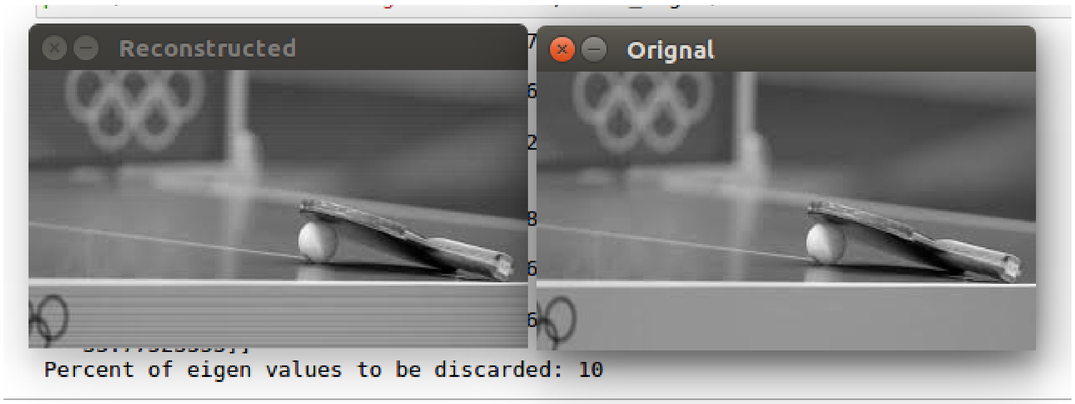
Partial Reconstruction (1% discarded)



Error 0.07666666666666666

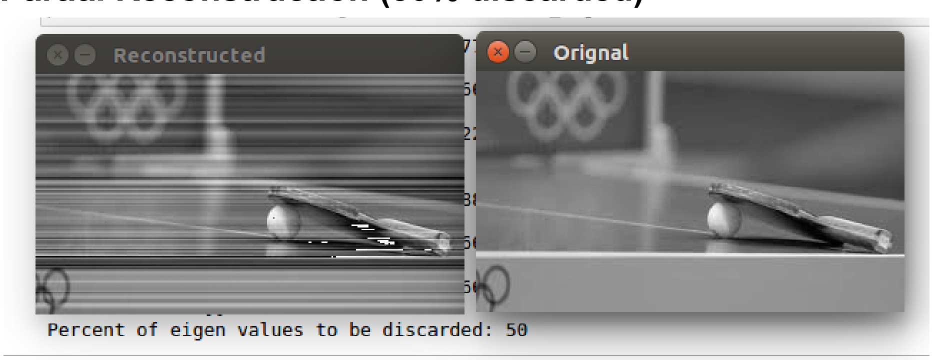
Sum of discarded eigen Values 0.0014156128461653874

Partial Reconstruction (10% discarded)



Error 1553.8500000000001
Sum of discarded eigen Values 0.0575068785782504

Partial Reconstruction (50% discarded)



Error 12795.096666666666
Sum of discarded eigen Values 19.140379501142707

Observation:

From the above experiment we see that the error is not equal to the sum of discarded eigen values as at every point of processing the image right from converting it to grayscale, we are approximating the intensity values and rounding the decimal outputs to uint8 as used by opencv here, which adds to the error at each step.

Also, discarding eigenvalues of about percentage upto 1% of the total does not lead to significant loss of information in the image , so we can reduce the dimensions of the original image by the value of eigenvectors’ percentage discarded from it.