Tien Li Shen

November 8, 2020

Computer Science 589

Homework 5

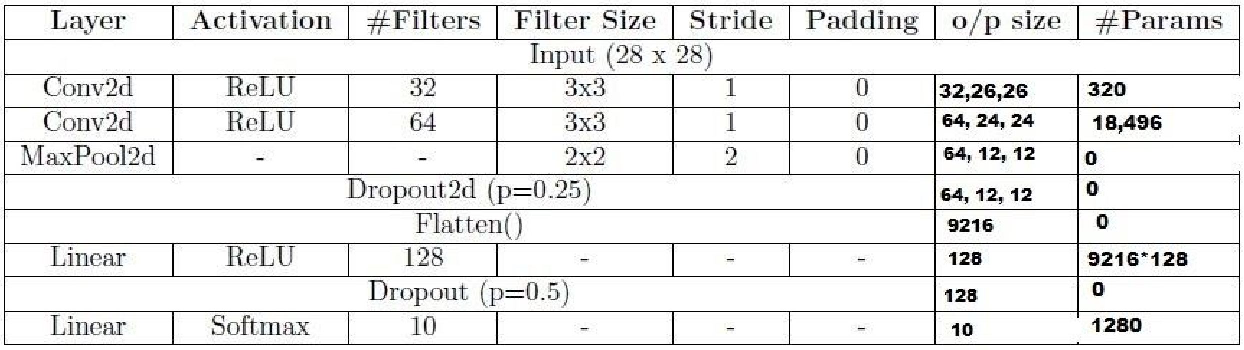
1.1

a. Fully connected layer: parameters = (input layer neurons + 1) * (Fc layer neurons)

b. Convolutional layer: parameters = (K*K+1)*C

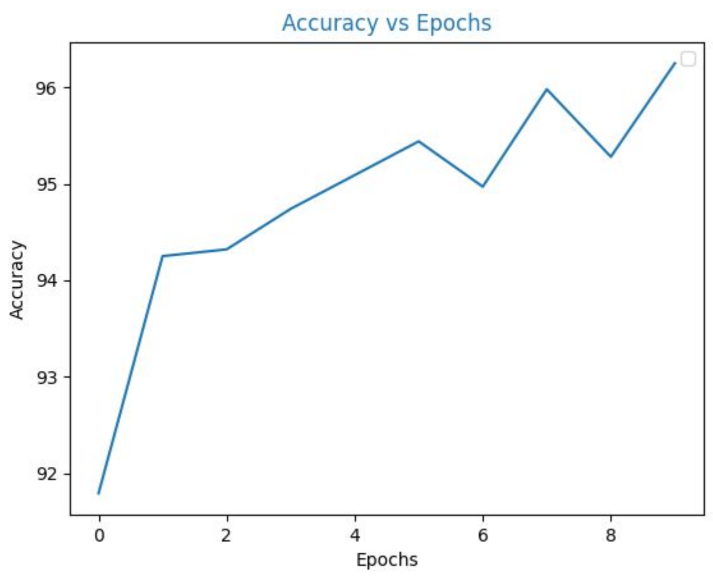
1.2

a.



Layer	Activation	#Filters	Filter Size	Stride	Padding	o/p size	#Params
Input (28 x 28)							
Conv2d	ReLU	32	3x3	1	0	32,26,26	320
Conv2d	ReLU	64	3x3	1	0	64, 24, 24	18,496
MaxPool2d	-	-	2x2	2	0	64, 12, 12	0
Dropout2d (p=0.25)						64, 12, 12	0
Flatten()						9216	0
Linear	ReLU	128	-	-	-	128	9216*128
Dropout (p=0.5)						128	0
Linear	Softmax	10	-	-	-	10	1280

b.



c. Total param = 320+18496+9216*128+1280 = 1,199,744

   Total conv param = 320 + 18496 = 18,816

   Total f_c param = 9216*128+1280 = 1,180,928

   F_c ratio = 1,180,928/ 1,199,744= 0.9843

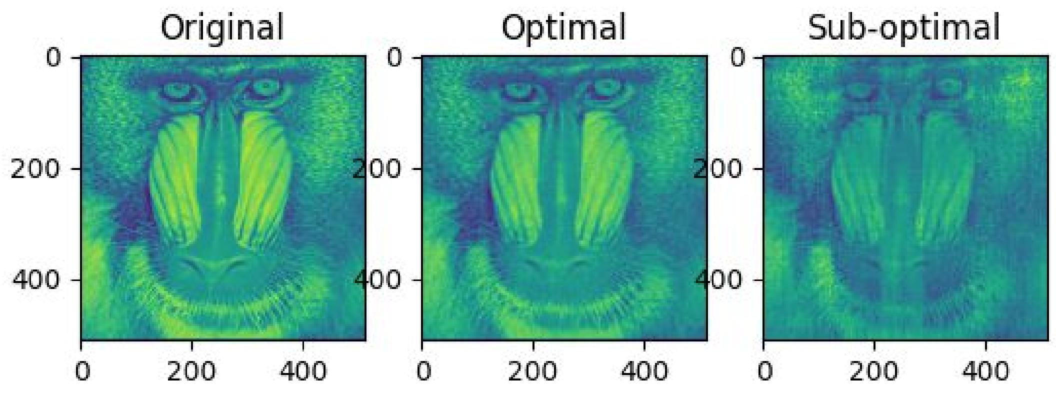
   Conv ratio = 1 - F_c ratio = 0.0157

   This result did surprise me. I would expect fully connected layers to have more parameters since the input was larger into the fully connected layers and there are more filters. However, the realistic ratio is more drastic than I expected.

2.1

See the attached PDF

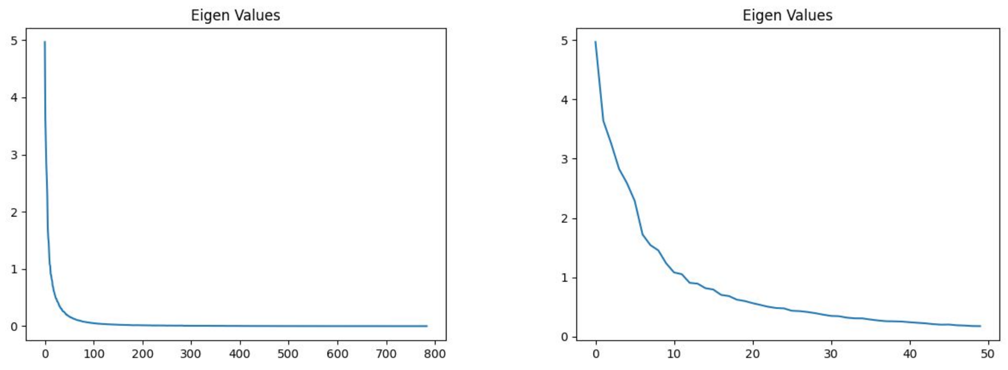
2.2



error for optimal 60 rank approximation is: 8827.53

error for sub-optimal 60 rank approximation is: 78916.34

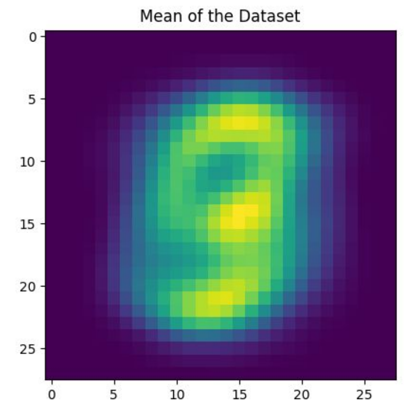
3.1

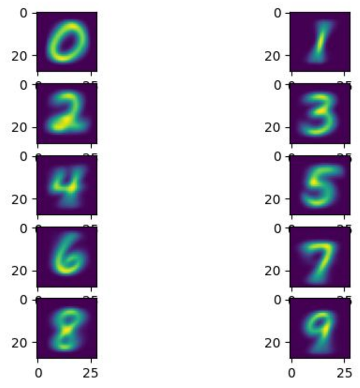


Eigenvalues represent the magnitude of variance for eigenvectors. I think I would use the first 10 eigenvalues. After the first 10, the eigenvalues decrease to less than 1.

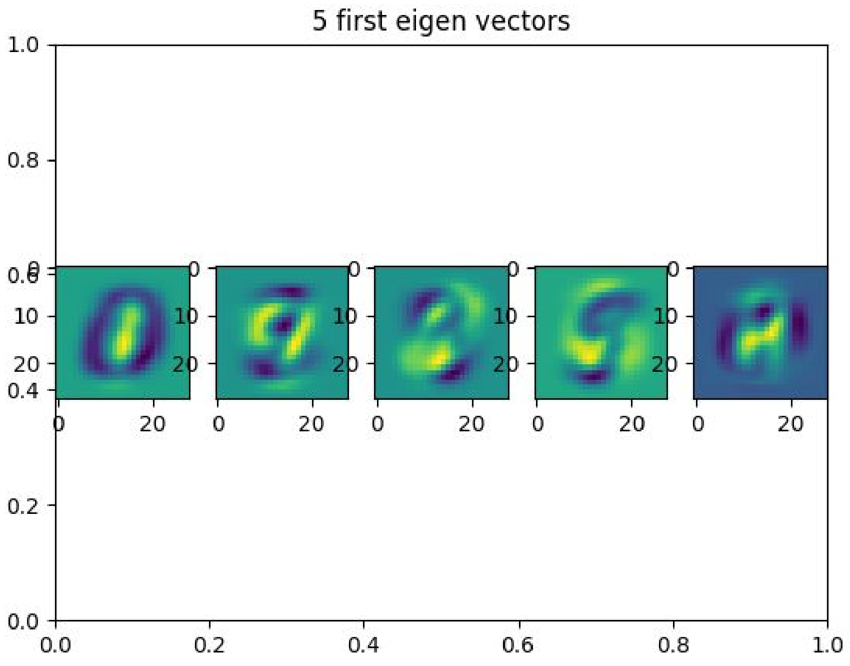
3.2

The first picture shows the mean of the dataset and the second shows the mean of each digit.



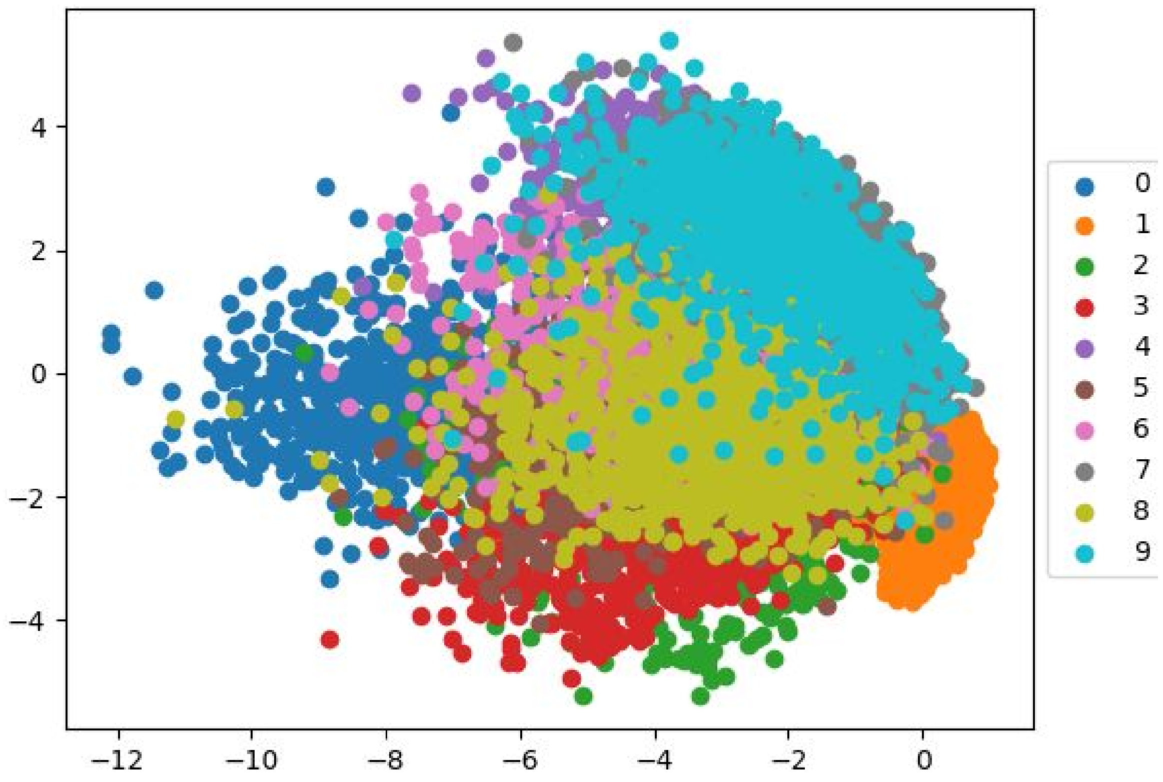


3.3



The pictures look like they represent the digits (0, 9, 3, 5, 3). I think these pictures represent the direction of maximum variance in the dataset. Blue regions represent pixels that see more variance in the dataset. Green regions represent pixels that have low variance in the dataset.

3.4



I think the plot is supposed to show the direction of variance for each digit in the dataset. If two digits were to seem similar to this plot, we can assume the digits are alike. digits (4, 7, 9), (6,8) (3,5) seem to be close to one another.

1 SVD implementation

```python
# Calculate probabilities p_i and use it as probibility distribution for constructing matrix S
n,m = A.shape
p = np.zeros(n)
fro = np.linalg.norm(A[1:], ord = 'fro') ** 2
for i in range(1, n):
    p[i] = (np.linalg.norm(A[i])**2)/fro
# Construct S matrix of size s by m using p prob distribution
S = np.zeros((s,m))
for i in range(1,s):
    j = np.random.choice(n, p = p, replace=False)
    S[i] = A[j]
# Calculate SS^T
sst = np.matmul(S, S.T)

# compute SVD for SS^T
u, s , vh = np.linalg.svd(sst)
lamb = np.sqrt(s[:k])

# Construct H matrix of size m by k
H = np.zeros((k,m))
for i in range(1,k):
    st_w = np.matmul(S.T, vh[i])
    H[i] = st_w/np.linalg.norm(st_w)
H = H.T
print("shape of H: ", H.shape)

# Return matrix H and top-k singular values sigma
return H, lamb
```