CS 464
Homework 2
Report

İlker ÖZGEN
21902719
13.12.2023

QUESTION 1

1.1)
Proportion of Variance Explained (PVE) for the first 10 principal components:

PC1: 0.09704664

PC2: 0.16800588

PC3: 0.22969677

PC4: 0.28359097

PC5: 0.33227894

PC6: 0.37540125

PC7: 0.40812055

PC8: 0.43695950

PC9: 0.46457980

PC10: 0.48814980

Discussion:

PVE allows us to know how much information each component keeps from the original data. In this case, approximately, PC1 explains 9.70% of the variance, PC2 explains 16.80% of the variance, and it goes like this. Observing the results, it can be seen that the PVE increases with each principal component. At first, the increase is high and it gets lower as more principal components added. These PVE values can help in selecting an appropriate number of principal components.

1.2) Number of components needed to explain 70.0% of the data is 26. The PVE values until PC26:

PC1: 0.09704664

PC2: 0.16800588

PC3: 0.22969677

PC4: 0.28359097

PC5: 0.33227894

PC6: 0.37540125

PC7: 0.40812055

PC8: 0.43695950

PC9: 0.46457980

PC10: 0.48814980

PC11: 0.50924170

PC12: 0.52947161

PC13: 0.54662979

PC14: 0.56355091

PC15: 0.57933732

PC16: 0.59416685

PC17: 0.60741246

PC18: 0.62018143

PC19: 0.63205406

PC20: 0.64358090

PC21: 0.65424256

PC22: 0.66430969

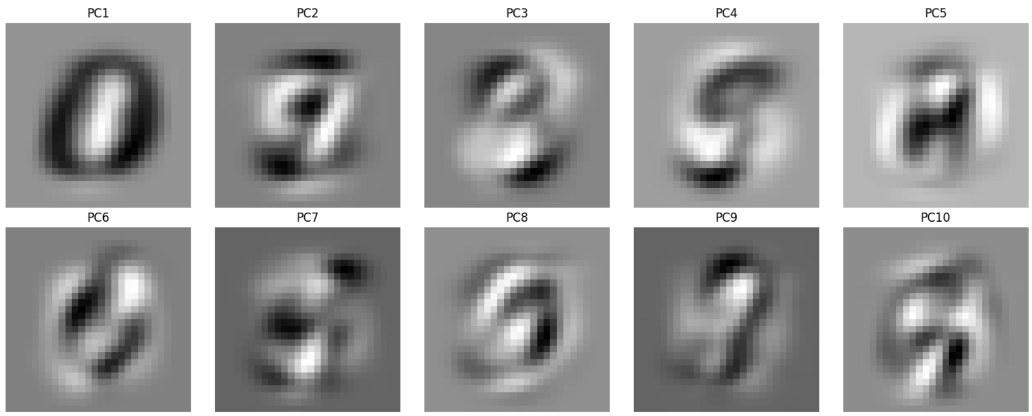
PC23: 0.67384543

PC24: 0.68297086

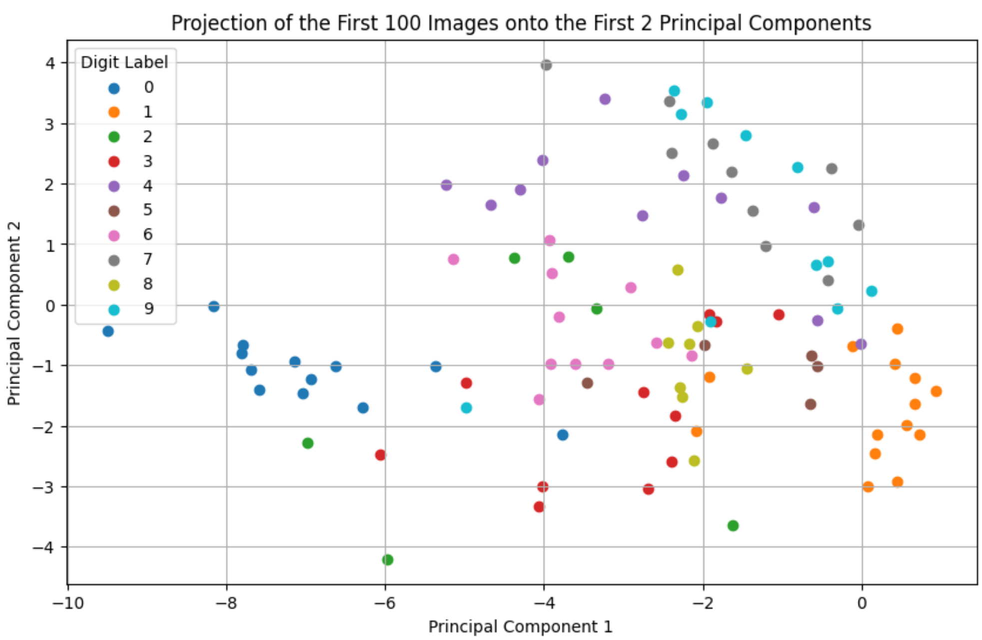
PC25: 0.69180491

PC26: 0.70019810

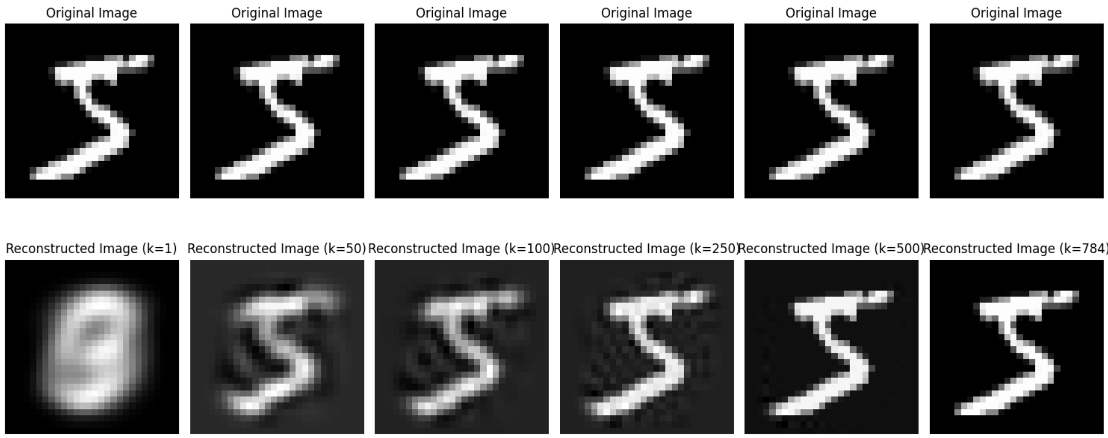
1.3) The output for this question can be seen below. The images show the patterns captured by principal components. These images and patterns help in identifying the essential features to represent the data. Images are expected to be blurry, as it is in the images, since PCA captures the variation in the data.



1.4) The distribution of the data points can be seen below. In the graph, we see that except for 0, all digits are mostly clustered in range [-4, 2] in x-axis and [-3, 3] in y-axis. The digit 0 data points are clustered in range [-8, -6] in x-axis and [-2, 0] in y-axis. This might mean that the digit 0 shows distinct patterns and these patterns are captured successfully by the first two principal components. The graph shows how well PCA preserves the patterns and captures the digits in the dataset, while reducing dimensionality.



1.5) The reconstruction of the image can be seen below. As can be seen, as k increases, the reconstructed image get closer to the original image. For small values of k, the reconstructed images are blurry since a limited number of principal components is used for it. At k = 784, the reconstructed image is the same as the original image. Therefore, we can conclude that as k gets closer to the total number of principal components, the reconstructed image gets closer and closer to the original one.

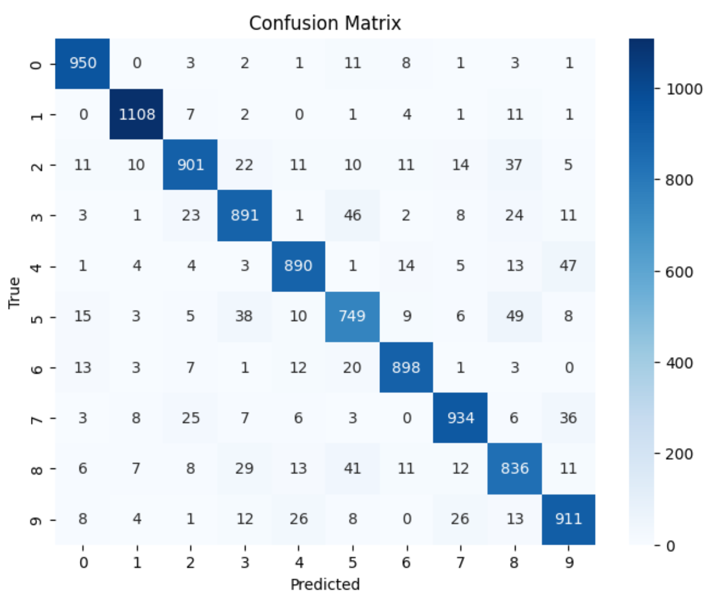


QUESTION 2

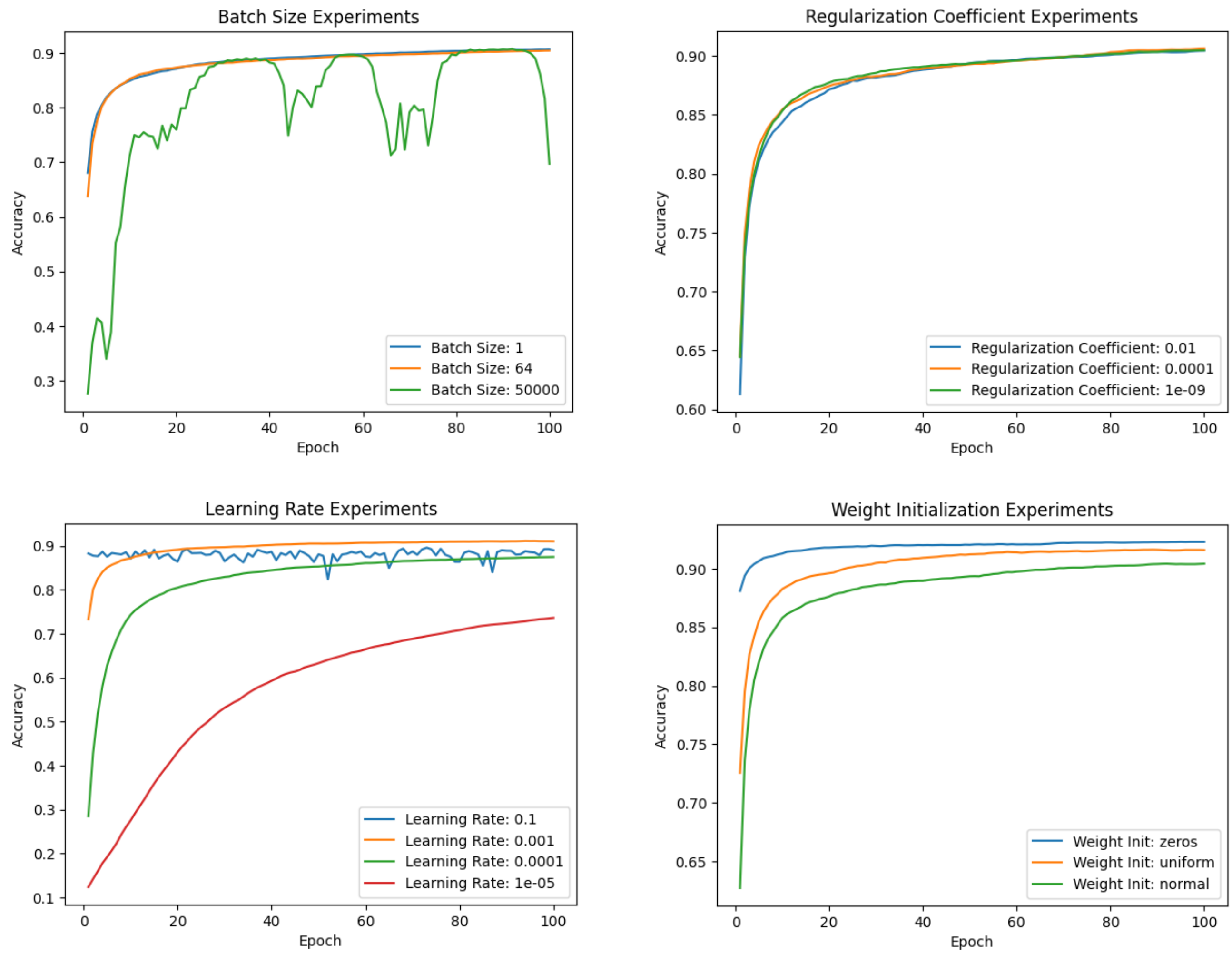
2.1)

Accuracy: %90.68

Confusion Matrix of Default Model:



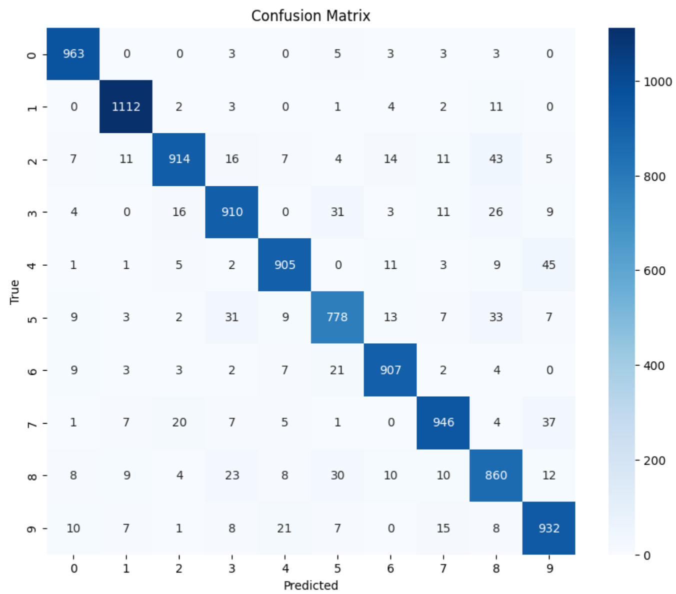
2.2) The results of the experiments for each hyperparameter are given below.



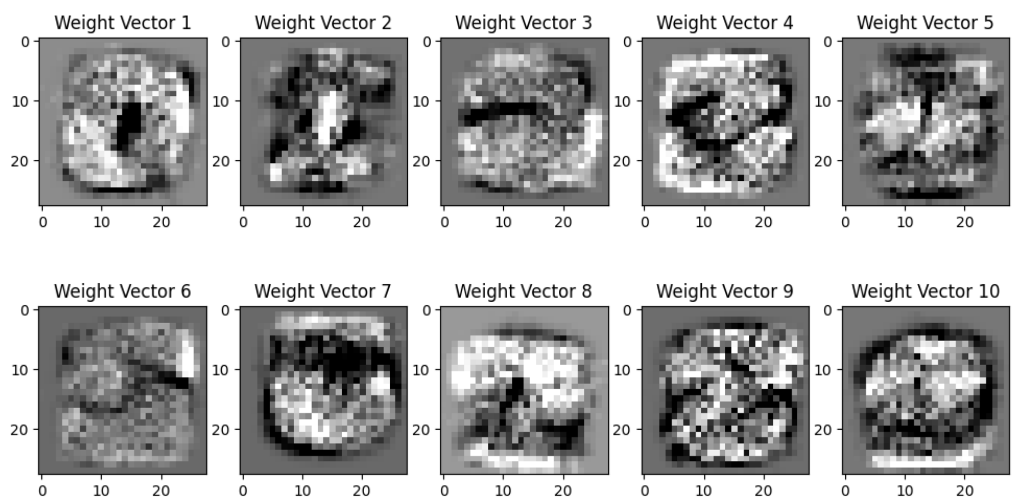
2.3)

Accuracy: %92.27

Confusion Matrix of Optimal Model:



2.4) The weight vectors can be seen below. Each weight vector represents the characteristics of a digit, although they are blurry. The blurriness observed in the images is expected and does not affect their interpretability. Each vector image carry certain features of the digit, for example, there might be loops for digit 6, or straight lines for the digit 1. The complexity of the digits also affect the vector images. By analyzing the weight vector images and the confusion matrix, we can identify the confused patterns. For example, if we look at the confusion matrix above, the most confused one is digit 4 predicted as digit 9. It is misclassified 45 times. Then, when we look at the weight vector images, we can see similarities in the patterns of the digits 4 and 9.



2.5)

Class 0:

Precision = 0.9516

Recall = 0.9827

F1 Score = 0.9669

F2 Score = 0.9763

Class 1:

Precision = 0.9644

Recall = 0.9797

F1 Score = 0.9720

F2 Score = 0.9766

Class 2:

Precision = 0.9452

Recall = 0.8857

F1 Score = 0.9145

F2 Score = 0.8970

Class 3:

Precision = 0.9055

Recall = 0.9010

F1 Score = 0.9032

F2 Score = 0.9019

Class 4:

Precision = 0.9407

Recall = 0.9216

F1 Score = 0.9311

F2 Score = 0.9254

Class 5:

Precision = 0.8861

Recall = 0.8722

F1 Score = 0.8791

F2 Score = 0.8749

Class 6:

Precision = 0.9399

Recall = 0.9468

F1 Score = 0.9433

F2 Score = 0.9454

Class 7:

Precision = 0.9366

Recall = 0.9202

F1 Score = 0.9284

F2 Score = 0.9235

Class 8:

Precision = 0.8591

Recall = 0.8830

F1 Score = 0.8709

F2 Score = 0.8781

Class 9:

Precision = 0.8902

Recall = 0.9237

F1 Score = 0.9066

F2 Score = 0.9168

Discussion of the results:

Overall, it can be seen that the precision and recall are relatively high for all of the classes. This indicates that the model's performance in recognizing digits is good. Analyzing these values, we can say that some classes show a trade-off between precision and recall, so fine-tuning might be needed. If we examine the values and the confusion matrix closely, we can say that

- For classes 0 and 1, high precision and recall indicate strong performance.
- For classes 2 and 4, the decrease in recall indicate an area to improve. It can also be seen from the weight vector images that it captures the patterns relatively weaker. Furthermore, when we look at the confusion matrix the most two misclassified cells are 4 predicted as 9 and 2 predicted as 8. The number of misclassification in them are 45 and 43 respectively.
- For classes 5 and 8, balanced performance of precision and recall indicate good performance.

Therefore, we can conclude that overall, the model has a good performance in predicting the classes, while there are some aspects that can be improved.