CS464 Introduction to Machine Learning

Homework 2

Name: Ceren Akyar

Student ID: 22003158

Question 1.1

```
Proportion of Variance Explained (PVE) for the first 10 principal components:
PC1: 0.0970
PC2: 0.0710
PC3: 0.0617
PC4: 0.0539
PC5: 0.0487
PC6: 0.0431
PC7: 0.0327
PC8: 0.0288
PC9: 0.0276
PC10: 0.0236
```

Fig. 1: PVE for First 10 Principal Components

PVE shows the amount of information captured by each principal component relative to the total variance. In our case, PC1 explains 9.7% of the variance. This means that, PC1 captures a significant amount of information in the dataset. Hence, it could possibly capture the most dominant patterns in the dataset. The following principal components slowly decrease, but they still contribute in a significant amount, too. In total, they approximately explain 45% of the variance.

Question 1.2

```
Cumulative Proportion of Variance
PC1: 0.0970
PC2: 0.1680
PC3: 0.2297
PC4: 0.2836
PC5: 0.3323
PC6: 0.3754
PC7: 0.4081
PC8: 0.4370
PC9: 0.4646
PC10: 0.4881
PC11: 0.5092
PC12: 0.5295
PC13: 0.5466
PC14: 0.5636
PC15: 0.5793
PC16: 0.5942
PC17: 0.6074
PC18: 0.6202
PC19: 0.6321
PC20: 0.6436
PC21: 0.6542
PC22: 0.6643
PC23: 0.6738
PC24: 0.6830
PC25: 0.6918
PC26: 0.7002
```

Fig. 2: Cumulative Proportion of Variance

As it can be seen from the figure above, at least 26 principal components should be used to explain the 70% of the data.

Question 1.3

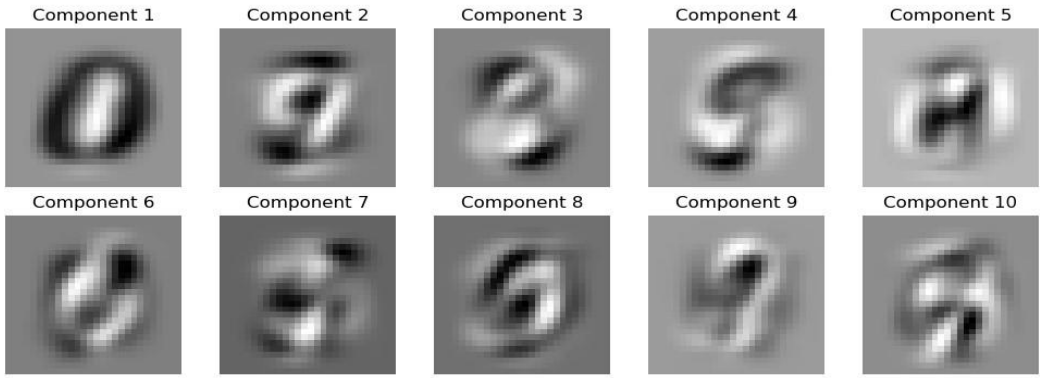


Fig. 3: Reshaped Images

The above figure shows the first ten principal components, each scaled to the range [0, 1]. The images show the most significant patterns captured. There are some patterns that appears similar to some digits. For instance, the first image contains circular patterns, indicating the 0 digit. Also, the third component shows a curvy pattern with an intersection that has some similarities with digit 3 or 8.

The first images show patterns that are more visible to human eye. However, when we come to the 7th component, the patterns become less interpretable. This can be due to the fact that these components possibly represent variations that are relatively less occurred in the dataset.

In short, it can be said that these components are the aggregation of features (or possibly, patterns) that define the structure of the dataset.

Question 1.4

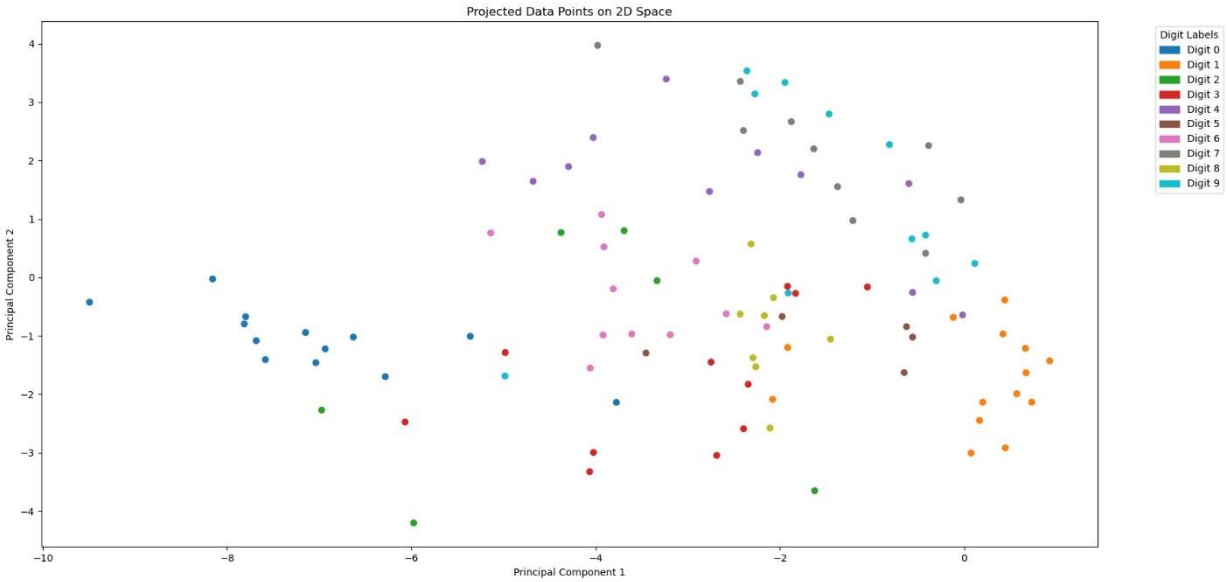


Fig. 4: Projected Data Points

As it can be seen from the above figure, there are some clustering. As aforementioned in the question 1.3, the first two principal components capture key features. These clustering are a proof of this. Both 0 and 9 contain some curvy patterns. From the figure, it can be seen that the shapes of 0 and 9 match with the first two principal components. Also, there are some other clustering that are not discovered yet. Those digits share common features that we might possibly need more dimensions to make a clearer distinction between.

Question 1.5

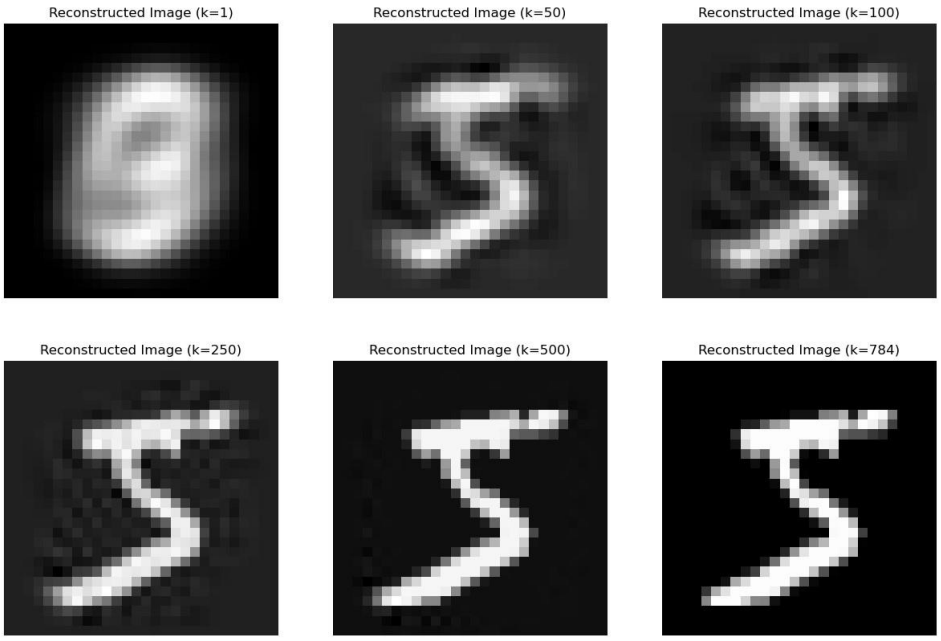


Fig. 5: Reconstructed Image With Various k Values

When k is 1, it can be seen that the image is undisguisable. Hence, it can be said that there must be more than one principal component to distinguish a digit. As k increases, we start to get better reconstruction quality. When k=100, it becomes very distinguishable. When we use all principal components, the reconstructed image is the same as the original image. However, using principal components is costly and we must use the resources efficiently.

Question 2.1

Test Accuracy: 0.9049

	Actual										
Prediction		0	1	2	3	4	5	6	7	8	9
	0	944	0	3	2	0	17	8	4	1	1
	1	0	1100	7	4	1	4	4	1	14	0
	2	10	13	898	27	11	6	11	15	35	6
	3	5	1	21	899	1	41	4	12	19	7
	4	2	3	9	4	890	2	14	5	11	42

	5	13	4	5	35	12	772	14	5	25	7
	6	9	2	16	2	11	19	895	2	2	0
	7	2	13	18	8	11	3	0	937	0	36
	8	13	13	9	29	13	44	12	10	818	13
	9	9	4	2	11	32	8	0	30	17	896

Table 1: Confusion Matrix for Default Model

Question 2.2

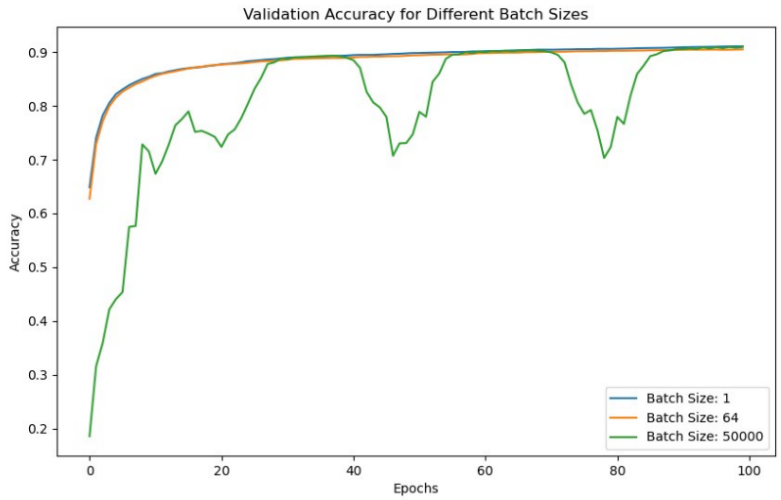


Fig. 6: Validation Accuracy for Batch Sizes 1, 64, 50000

When the batch size is 1, it adapts quickly to the changes. When the batch size is 64, it compromises between the efficiency of large batch sizes and the adaptability of small batch sizes, and gave us a high accuracy. It can also be seen that when batch size is 50000, it did not perform well compared to sizes 1 and 64.

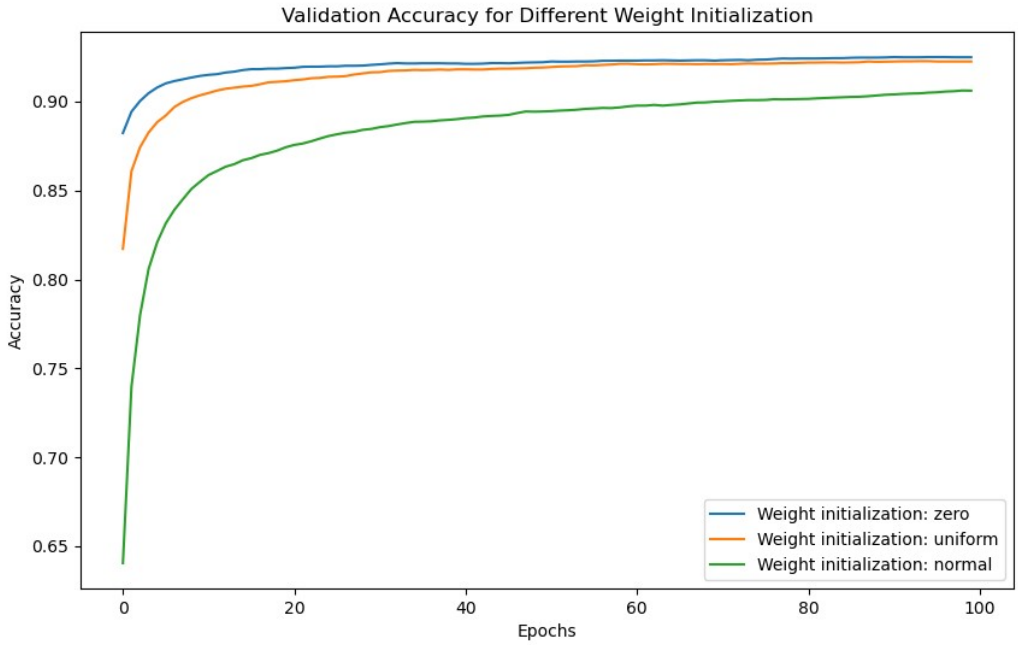


Fig. 7: Validation Accuracy for Weight Initialization Zero, Uniform, Normal Dist.

It appears when the weight initialization technique is zeros, it converges the fastest.

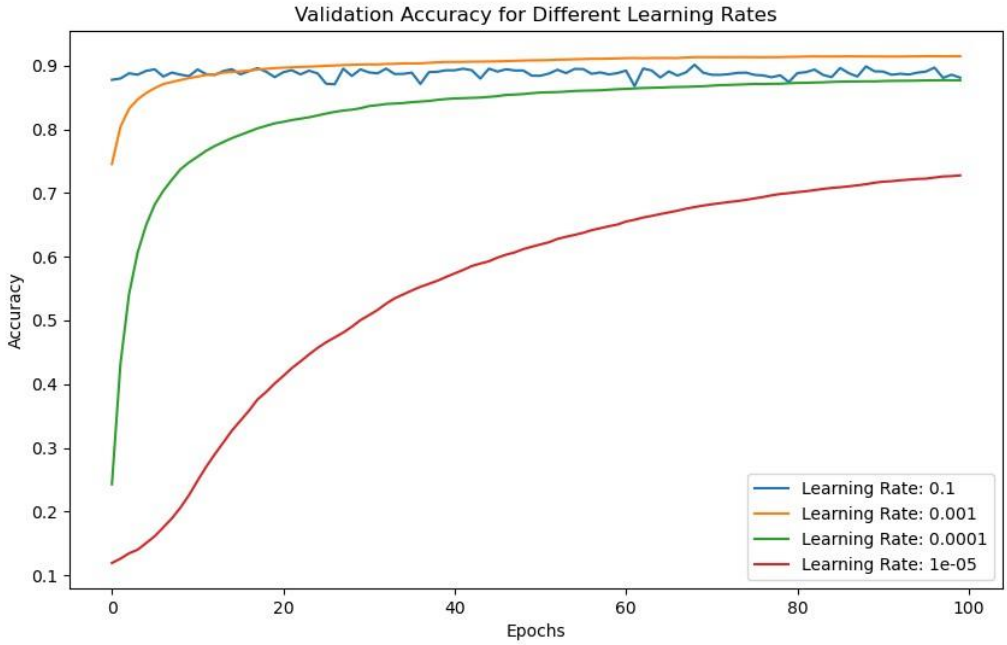


Fig. 8: Validation Accuracy for Different Learning Rates

The 0.1 learning rate is relatively large compared to other learning rates. There are some fluctuations in the model. It is because there are some overshooting which cause the model to diverge but then rapidly converge again.

The 10^{-3} learning rate is a commonly used learning rate for logistic regression. In this model, it provided a balance between convergence speed and stability, and provided better accuracy results.

The 10^{-4} learning rate can be used for a stable convergence, but it makes the convergence slower, which can be seen from the graph.

The 10^{-5} learning rate is relatively small. Hence, it led the model to a slow convergence. The model did not reach the optimal solution within 100 epochs.

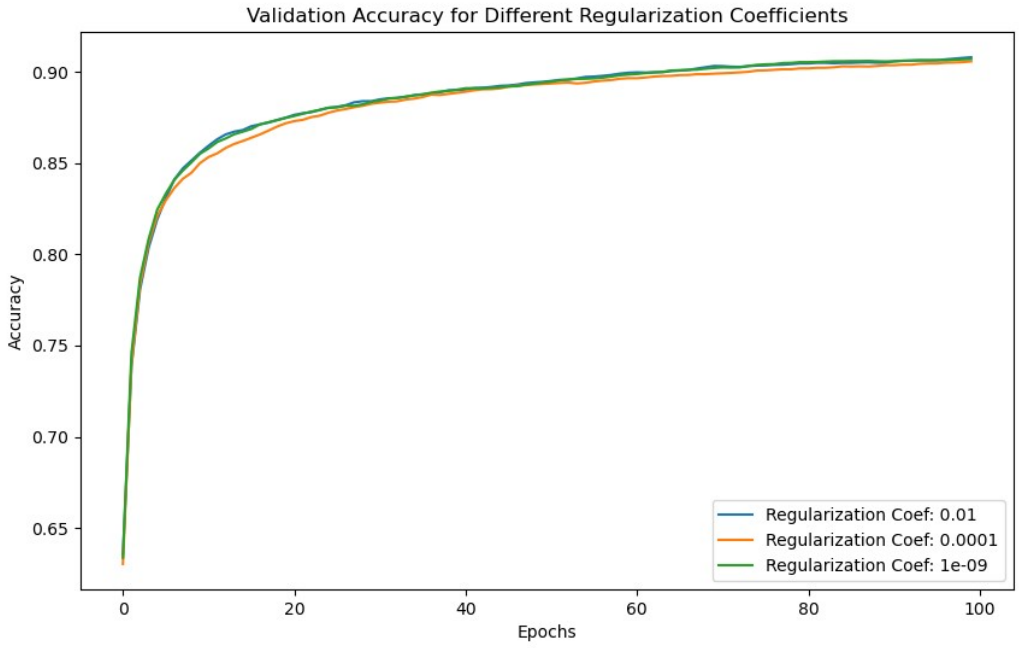


Fig. 9: Validation Accuracy for Different Regularization Coefficients

The regularization coefficient 10^{-9} is relatively weak. However, it appears that all three regularization coefficients are convenient for our model and avoided overfitting.

Question 2.3

When the model has default parameters, it performs as:

```
Test Accuracy: 0.9093
Confusion Matrix:
[[ 947    0    4    2    0   14    9    2    1    1]
 [   0 1103    5    1    0    2    3    3   17    1]
 [   7    9  906   17   10    7   20   13   38    5]
 [   5    1   26  901    0   32    3   12   22    8]
 [   1    3    7    2  893    3   13    8    9   43]
 [  11    2    7   39   11  766   12    9   23   12]
 [  11    3    8    1   10   19  899    3    4    0]
 [   1   10   25    6    3    1    0  931    3   48]
 [   8   10    9   27   14   36    9    8  843   10]
 [   9    8    2    8   32    7    1   27   11  904]]
```

Fig. 10: Confusion Matrix of the Default Model

Then, I chose the parameters as:

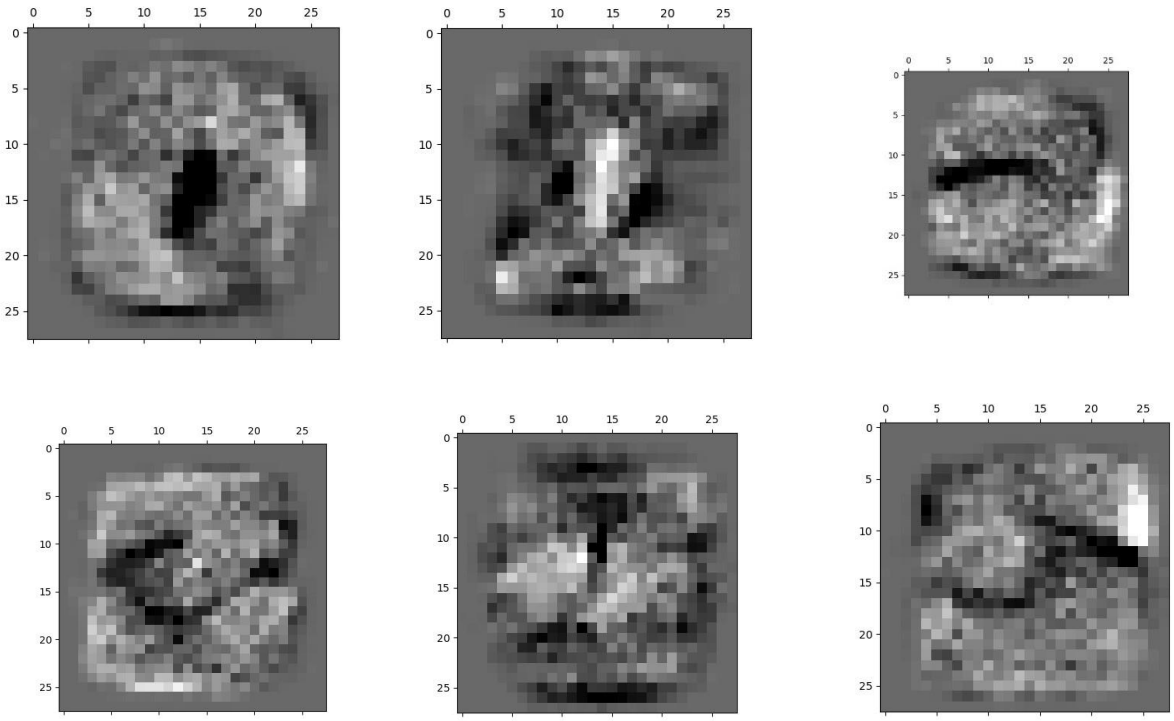
- Weights initialized with zero initialization technique
- Batch size as 64
- Learning rate as 10^{-3}
- Regularization coefficient as 10^{-2}

Then, the accuracy was increased to 0.9229

```
Test Accuracy: 0.9229
Confusion Matrix:
[[ 960    0    1    3    0    6    7    2    1    0]
 [   0 1111    2    2    0    1    4    2   13    0]
 [   9    9  917   16    7    6   15    9   35    9]
 [   5    1   16  910    0   31    4   12   18   13]
 [   1    3    4    1  904    0   11    4    7   47]
 [  10    2    2   29    8  788   15    6   25    7]
 [  11    3    4    2    8   16  908    3    3    0]
 [   1    9   20    7    6    2    0  941    2   40]
 [   8    9    6   23    8   34   12    9  853   12]
 [  11    8    1    8   19    5    0   13    7  937]]
```

Fig. 11: Confusion Matrix with New Parameters

Question 2.4



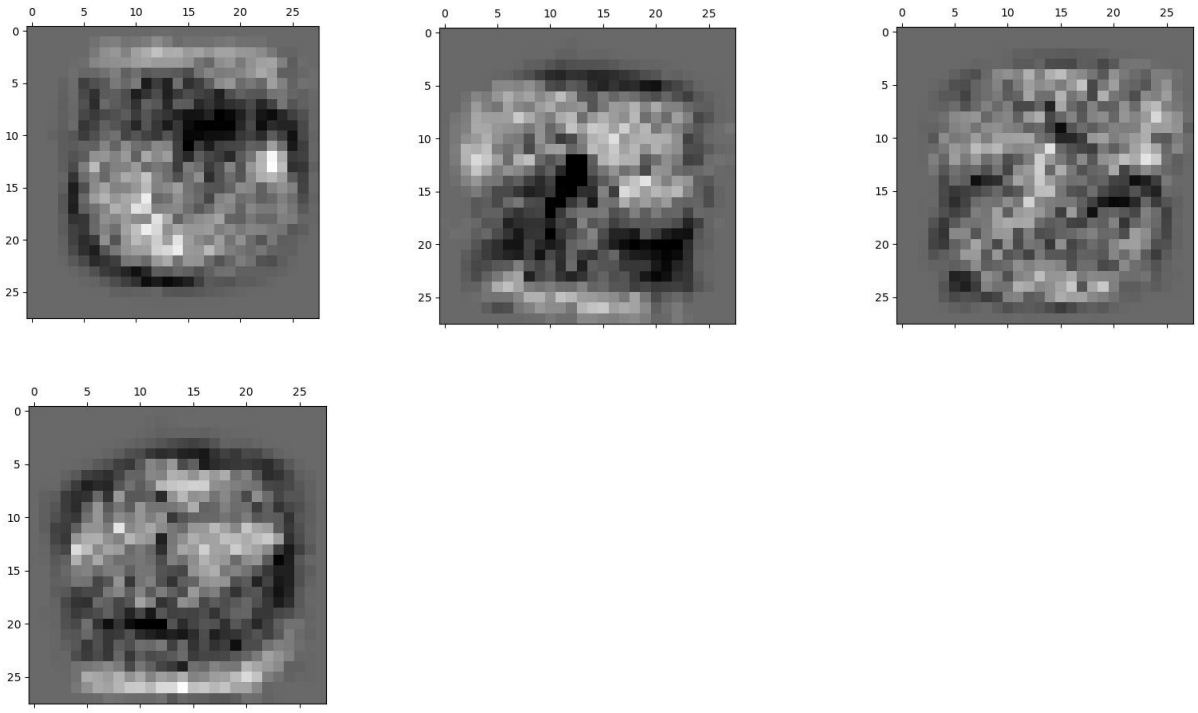


Fig. 12: Weight Images

The weight images represent the patterns that the model has learned to associate with each digit. In the above figure, there are some patterns visible to human eye. For instance, the weight image corresponding to the digit '1' captured the vertical stroke, or the weight image corresponding to digit ‘0’ captured the circular pattern. The common patterns for 2, 3 and 4’s middle part is also captured in the weights. However, the 5’s patterns are less visible to human eye, could be captured better. This can be the reason the accuracy of 5 is relatively smaller.

The blurriness in the weight images is expected since the model is assigning importance to various pixels. Also, some noise is present in the weight images since the model can possibly assign non-zero weights to pixels that don't contribute significantly to digit recognition.

In short, each image represents a digit class. Therefore, the weight images captured features that are characteristic of each digit.

Question 2.5

```
Class:  0
Precision: 0.9448818897637795
Recall: 0.9795918367346939
F1 Score: 0.9619238476953907
F2 Score: 0.972447325769854
Class:  1
Precision: 0.9619047619047619
Recall: 0.9788546255506608
F1 Score: 0.9703056768558952
F2 Score: 0.9754170324846356
Class:  2
Precision: 0.9424460431654677
Recall: 0.8885658914728682
F1 Score: 0.9147132169576061
F2 Score: 0.8988433640462654
Class:  3
Precision: 0.9090909090909091
Recall: 0.900990099009901
F1 Score: 0.905022376926902
F2 Score: 0.9025986907359649
Class:  4
Precision: 0.9416666666666667
Recall: 0.9205702647657841
F1 Score: 0.9309989701338826
F2 Score: 0.9247135842880524
```

```
Class:  5
Precision: 0.8863892013498312
Recall: 0.8834080717488789
F1 Score: 0.8848961257720381
F2 Score: 0.884002692393987
Class:  6
Precision: 0.930327868852459
Recall: 0.9478079331941545
F1 Score: 0.9389865563598758
F2 Score: 0.9442595673876871
Class:  7
Precision: 0.9400599400599401
Recall: 0.9153696498054474
F1 Score: 0.9275505174963036
F2 Score: 0.9202034030901622
Class:  8
Precision: 0.8848547717842323
Recall: 0.8757700205338809
F1 Score: 0.8802889576883385
F2 Score: 0.8775720164609054
Class:  9
Precision: 0.8798122065727699
Recall: 0.9286422200198216
F1 Score: 0.9035679845708775
F2 Score: 0.9184473632621053
```

Fig. 13: Performance Metrics

In Fig. 11, the numbers on the diagonal represent the true positives, indicating the number of instances correctly classified for each digit. Currently, the 5 and 8 have relatively small amount of correctly classified label counts. This imbalance causes a smaller overall accuracy.

Precision measures the accuracy of positive predictions. Higher precision values indicate fewer false positives. Currently, class 1 and class 6 have high precision values. Hence, the digit predictions of 1 and 6 have good accuracy in positive predictions.

Recall measures the ability of the model to capture all positive instances. Higher recall values indicate fewer false negatives. Currently, class 0 and 1 have high recall values, meaning that the model effectively captures positive instances for these two digits.

The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall. The F2 score puts more emphasis on recall. In short, high F1 and F2 scores means that the model can predict that digit accurately. Currently, class 0, 1, and 6 seems to perform the best. Other digits also perform in a considerable amount of success rate. However, classes 5 and 8 perform the lowest. Their relatively poor performance ca be seen in metric performance scores. But they can also be seen from the weight images that were discussed in question 2.4. As aforementioned, the patterns in weight images were not captured accurately compared to other images. Hence, there is a correlation between the confusion matrix and the weight images. The model's performance exhibits a correspondence between weight images' captured patterns and each class's performance metrics.