**Question 1**

**What we did:**

In this question, we loaded the mnist\_784 dataset, which has 70k rows and 784 columns. We then used PCA to retrieve the first and second principal components and plot them onto a 1D hyperplane. After that, we used Incremental PCA to reduce the number of dimensions from 784 to 154 and saw how the original images compare to the compressed images.

**Images:**

A number in a square

Description automatically generated with medium confidence

The image above is the plot of each digit from 0 to 9. They’re the first occurrences of each digit in the dataset.

A graph of a graph with a red line

Description automatically generated

The image above is the projections of the first and second principal components onto a 1D hyperplane.

A collage of numbers

Description automatically generated

A number with numbers on a black background

Description automatically generated with medium confidence

The images above show the difference between the dataset with the original number of dimensions (784) in comparison to the dataset with the compressed number of dimensions (154).

We can see that the compressed version is a bit blurrier than the original. The color of the background also seems to be more grey-ish compared to the pitch black of the original.

**Lesson learned:**

* Incremental PCA uses mini batches, so it is more ideal for large datasets because we don’t have to load the whole dataset into the memory.
* Even though the dimension is reduced by about 80%, we can still clearly tell the digits using the compressed images.
* Higher Explained Variance Ratio means that compression will be better as it will capture more variance from the original.
* Incremental PCA with 154-D was able to capture about 95% of the original mnist\_784. I found this by summing up the “explained variance ratio” of the 154 principal components.

**Question 2**

**What we did:**

In this question, we generated a Swiss Roll dataset. We then used Kernel PCA with linear, rbf, and sigmoid kernels to reduce dimensions from 3 to 2. We then applied Logistic Regression to the Swiss Roll dataset. Because the label was continuous, I simply converted them to integers so we can work with Logistic Regression. After that, we used GridSearchCV to find the best kernel along with the gamma value for Kernel PCA.

**Images:**

A graph of a number of dots

Description automatically generated with medium confidence

The image above is the plot of the Swiss Roll dataset we generated. It has 1500 samples, and the random state for the generation is set to 0.

A diagram of a brain

Description automatically generated

The image above is the plots of the 3 kernels for Kernel PCA.

The plot on the left is the result of using the linear kernel. As we can see, it is shaped like that because it’s basically just projecting the 3d Swiss roll to 2d instead. Kind of like looking at the cross section. This doesn’t unroll the Swiss roll effectively as we can still see a spiral shape.

The plot in the middle is the result of using the rbf kernel with gamma 0.04. As we can see, it separates the dataset quite well. There is only a little spiral shape left, and some regions are still overlapping.

The plot on the right is the result of using the sigmoid kernel with gamma 0.04. As we can see, it still has the spiral/circular shape from the original Swiss roll, and it has quite a lot of overlaps.

Overall, the plot of rbf result seems to be the one that unrolls the Swiss roll best because it has the least overlaps, and it also removes a lot of the spiral shape from the original Swiss roll. The worst one seems to be linear as it is not made to separate non-linear datasets. We simply cannot separate non-linear datasets with just a straight line.

A graph with a line

Description automatically generated

The image above shows the affect of gamma on the cross-validation accuracy on each kernel type. We see that the gamma value of 0.03 is the best for both kernels.

**Lesson learned:**

* Learned more about the differences between linear, rbf, and sigmoid kernel on Swiss roll dataset.
* Learned that the higher the gamma value, the lower the influence of each data point has on other points. In other words, if the gamma value is high, the radius of influence is low. On the other hand, the lower the gamma value, the higher the radius of influence.
* Higher gamma can potentially lead to overfitting, lower gamma can potentially lead to underfitting.
* Learned that even though the sigmoid kernel is not the best at separating Swiss roll dataset visually, it can still perform better in Logistic Regression.