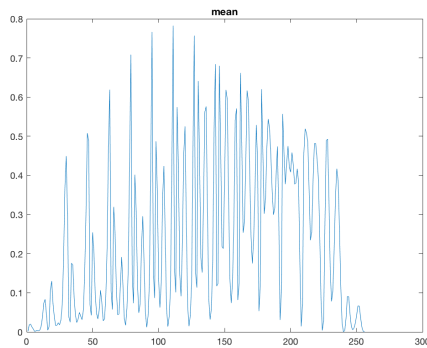


Artificial Neural Networks: Session 3

Alejandro Rodríguez Salamanca: r0650814@student.kuleuven.be

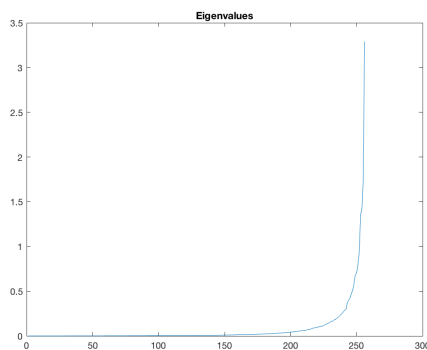
Handwritten Digits PCA and reconstruction

In this exercise we are asked to perform PCA on handwritten images of the digit 3 taken from the US Postal Service database. First of all, the mean is displayed. This plot shows the mean of each column of the input data.



It can be seen that the mean varies across the columns without showing any recognizable pattern.

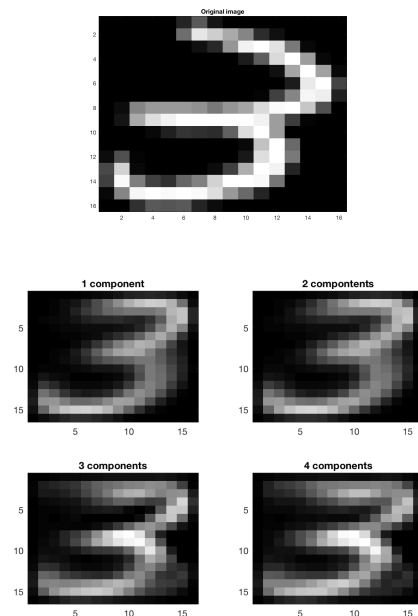
This second plot shows the eigenvalues of the covariance matrix of the input data.



Then, the Principal Component Analysis is computed. The objective is to reduce the dimensionality of our data. For this, it is decomposed into its principal components, and then, the number of components to use is decided. This plot represents the significance of each component in the original data. As we can see, few of the components capture most of the variance of the data, which means that it is possible to get rid of most

of the dimensions, reducing considerably the complexity of our dataset.

The following image shows the original digit 3 of the data set used, and below, we can see the reconstruction of this digit after projecting the original data set into one, two, three and four principal components.

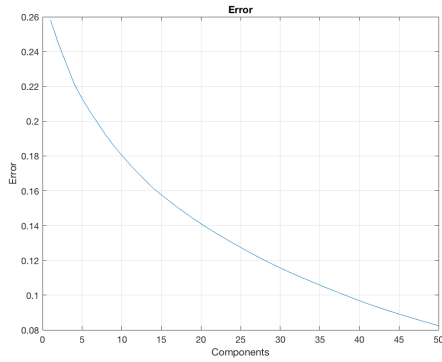


At first glance it is difficult to see how well the data has been reconstructed. Although it can be observed that the quality of the image increases with the number of components used, it is interesting to plot the error to see how it evolves.

To achieve this, we have calculated the error as $\sqrt{\text{mean}(\text{mean}((X - X_{\text{approx}})^2))}$. The inner `mean` is used to obtain the mean of each column of the matrix, and the outer `mean` to compute the mean of this new matrix, reducing it to a single value that we can plot, being `X` the original data set, and `Xapprox` the new approximated data set.

The error has been computed iteratively using one more component on each iteration, starting at 1 and ending at 50 components.

It is possible to see that the error function converges somewhere near 0.08, which means that using more than 50 components does not increase the quality of the reconstructed image, but only adds more complexity to the data adding superfluous dimensions.



If we reconstruct with 256 principal components, this is, the whole matrix, the error should be zero, as we are using the entire matrix to reconstruct the original data, which means that there is no dimensionality reduction, but only a change of basis.

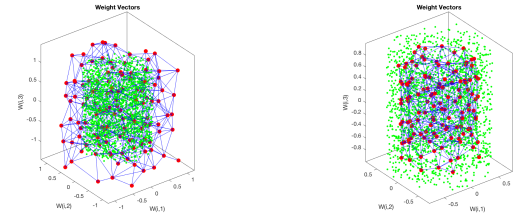
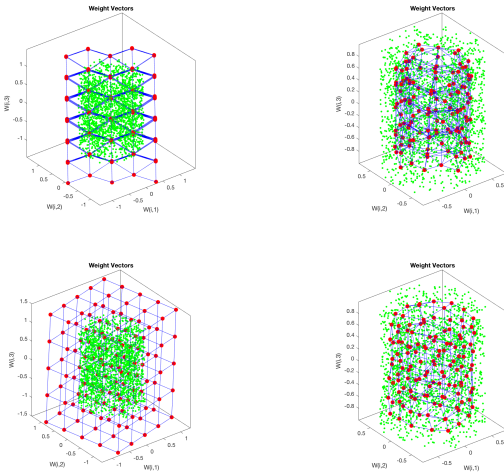
In practice, this is not true, and we can see that the error is 0.0824. This can be caused by the error carried by the operations in floating point.

SOM applied to the cylinder and Iris datasets

A self organizing map is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space of the training samples. SOMs accomplish two things, they reduce dimensions and display similarities.

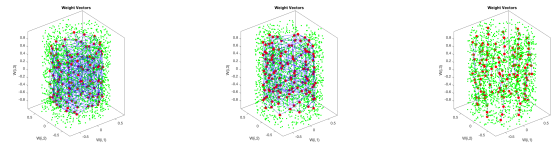
Cylinders

In this example we generate data uniformly distributed within two concentric cylinders. The following images shows the neurons in three different topologies, **hextop**, **gridtop** and **randtop**.



It is possible to see the difference in the initial disposition of the neurons depending on the selected topology, and how the neurons are arranged after training.

It is also interesting to see the influence of the distance function used and how it affects to the final result. Here, **boxdist**, **dist** and **mandist** have been used.

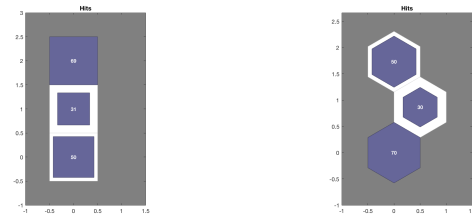


Iris

The data set consists of samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. We are going to use SOM to perform some clustering in this dataset, trying different topologies, distance functions, number of epochs and grid size.

The initial grid size is 3x1. This grid size its due to the number of species in the dataset, 3. This gives a better performance than the typical choice for the number of neurons, $s = 5\sqrt{N}$, being N the number of training data.

In the following image we can see the hits of each cluster using grid and hexagonal topologies.



If we look at the Adjusted Rand Index, the following results are obtained: **gridtop** = 0.6719, **hextop** = 0.6719, **randtop** = 0.6828. It is important to say that using **randtop** the ARI varies each time we train the network, as the shape of the clusters is stochastic.

If the size of the grid is changed, the ARI decreases as expected.