# Artificial Neural Networks: Exercise session 3

Stijn Staring (r0620003)

May 21, 2021

## 1 Principal Component Analysis

Principal component analysis is a method for dimensionality reduction. A principal component analysis uses the eigenvectors of the covariance matrix of the original data as principal components to conduct a linear transformation of the original data to a lower dimension. The eigenvectors that correspond to the largest eigenvalues are considered the most important principal components that determine data appearance the most.

$$\boldsymbol{z} = \boldsymbol{E}^T \boldsymbol{x}. \tag{1}$$

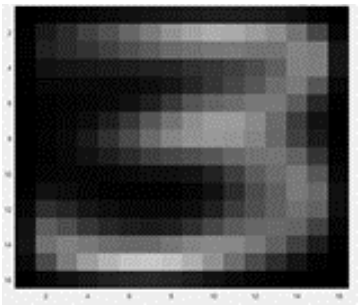
In Eq. 1, $\boldsymbol{x}$ is an original data point with dimensions $\mathbb{R}^p$ and $\boldsymbol{z}$ is the corresponding data point with reduced dimensions $\mathbb{R}^q$. $\boldsymbol{E}^T$ is the used transformation matrix with dimensions $q \times p$ and with the $q$ largest eigenvectors of the covariance matrix as rows. The new dimensions of the data point equals the amount of chosen eigenvectors that span the subspace. If all eigenvalues are distinct, $\boldsymbol{E}$ is orthogonal and the original data point can be retrieved by multiplying Eq. 1 by $\boldsymbol{E}$ because of the orthogonality property: $\boldsymbol{E} \cdot \boldsymbol{E}^T = \boldsymbol{I}$.

The first task performed is the comparison of the dimensionality reduction of random data in comparison to highly correlated data. It is found that there are less eigenvectors needed for the highly correlated data when the same RMSE is considered. The second task is to perform PCA on handwritten images of the digit 3 taken from the US Postal Service database. In this database every image consists out of $16 \times 16$ pixels which is represented by an array of 256 floating numbers. The database consists out of a total of 500 images of the digit 3. When the mean array is calculated, also the mean digit 3 is obtained which is displayed in Figure 1a. The covariance matrix is calculated for the data set and the eigenvalues derived are shown by Figure 1b. The influence of the amount of eigenvectors is demonstrated in Figure 1c. When only one eigenvector is used, the reduced images are very similar. This means that the eigenvector with the largest eigenvalue focusses on the intrinsic shape of a 3. This means in what distinguishes a 3 from another number e.g. a 5. when more eigenvectors are used and the dimensionality of the spanned subspace increases, different variations of the digit 3 becomes clear.

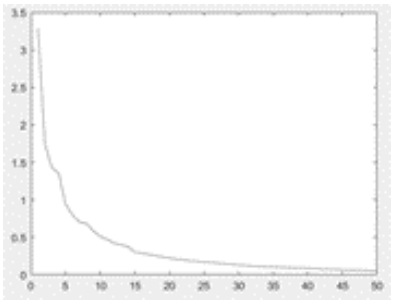
In Figure 2 the effect of the number of eigenvectors on the RMSE used in the PCA, is investigated and the distribution of the sizes of the eigenvalues is shown. From Figure 2a it is concluded that when more eigenvectors are included, the representation error decreases because the dimensionality is less reduced. When the derivative of the function is assessed, it is found that the function decreases much faster at the beginning than at the end. This means that already by the use of a small amount of eigenvectors to span the subspace with a lower dimensionality, the reconstruction error can be significantly decreased and an already good reconstruction of the original data is possible. Figure 2b shows the cumulative sum of eigenvalues used in the PCA with on the x-axis the number of excluded eigenvalues starting from the largest. It is seen that a small amount of eigenvalues have large eigenvalues after which the size decreases rapidly. The plot is similar to Figure 2a and it can be concluded that the contribution of an eigenvector in reducing the RMSE corresponds to the size of the eigenvalue. This confirms what was concluded in Figure 2a, that already good representations of the digit 3 are possible with a small amount of eigenvectors. When Eq. 1 is used without a dimensionality reduction, a very small reconstruction error equal to the default floating point precision in Matlab is found.

## 2 Stacked Autoencoders

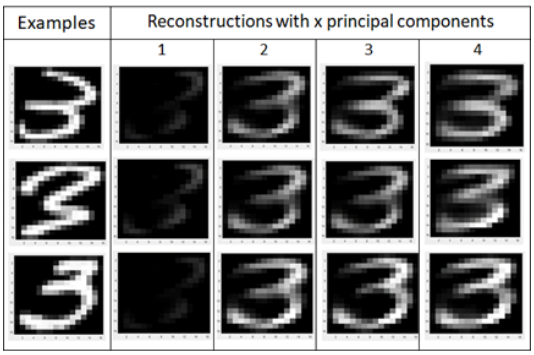
An autoencoder is an artificial network that learns efficient encodings of the original data. This is done to force the original data to be represented by only a limit amount of neurons causing a dimensionality reduction (encoding) after which the original data is reconstructed (decoding) and compared with the original input. When the outputs after decoding resemble only have a small reconstruction error, this means that the hidden layer that performed a dimensionality reduction, was able to identify the most important components that determine the data appearance, similarly as what was the goal of the principal component analysis. Stacked autoencoders refers



(a) Mean image of digit 3

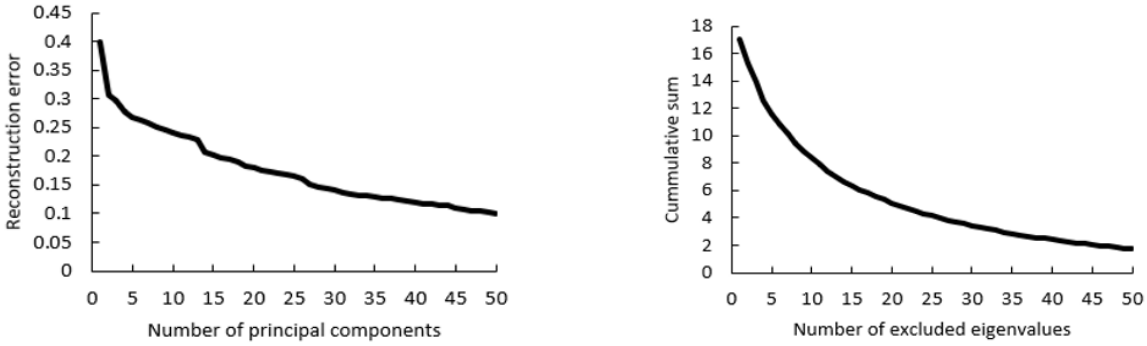


(b) Eigenvalues




(c) Increasing amount of eigenvectors

Figure 1: Results of applying PCA on the US Postal Service database.




(a) (b)

Figure 2: The influence of the amount of eigenvectors used.

to the subsequently use of autoencoders where the hidden layer of a previous encoder serves as inputs of a next one. Applications of dimensionality reduction making use of autoencoders are:

- Image compression
- Denoising
- Feature extraction

Image compression

- stacked implies to the sequential use of different autoencoders.

The use o - the use of an autoencoder and its constraints can be useful to perform a dimensionality reduction and to identify the important components that determine data appearance the most, similar as what was the goal of the principal component analysis. Applications where a dimensionality reduction is important is for example preprocessing, where noise is removed using an autoencoder or PCA.

- autoencoder can also be used when training a deep neural network layer per layer. Each layer is supposed to learn features at a different level of abstraction. By using a sparse autoencoder, the layers are forced to reduce the dimensionality and learn important feature of the inputs. The hidden layer of the autoencoder, which corresponds to one of the hidden layers of the deep neural network is feeded as input of the next autoencoder. Using

training with an autoencoder don't needs labels. A suitable solution is to make use of a validation set and early stopping.

- too many hidden units –¿ doesn't force the autoencoder to make generalizations –¿ just remembers the completer pictures. This means that the error during training can become low, but because nothing is learned –¿ the performance on the test set will be unsatisfactory. This is again to overfitting.

- the number of layers –¿ the network becomes more expressive. Each layer can learn features at a different level of abstraction. This means that more care has to be taken with overfitting. A very expressive network doesn't make generalizations but remembers the training data.