1. For this problem, you are asked to use the data in the `Pizza.csv` file (the same file used for HW #4). The feature values are contained in columns 3 through 9. Construct an autoencoder with 3 layers (i.e., one hidden layer for the code). Let $\mathbf{X}$ be the matrix that contains the samples as its rows. Similarly, the rows of $\hat{\mathbf{X}}$ are the reconstructed inputs from the autoencoder. You will evaluate the performance of autoencoders using the mean squared error (MSE) given below as you vary the dimension of the code:

$$MSE(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n}\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2, \tag{1}$$

where n is the number of samples, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

   (a) Design an optimal autoencoder with the linear encoder and decoder without nonlinear activation functions, which minimizes the MSE. Compute the MSE as a function of the dimension of the code $h \in \{1, 2, \ldots, 6\}$.

   Ans: The weights for optimal encoder and decoder can be obtained from the singular value decomposition of the matrix $\mathbf{X}$, which is also related to the Principal Component Analysis (PCA), as discussed during the lecture. The MSE is plotted in Figure 1.

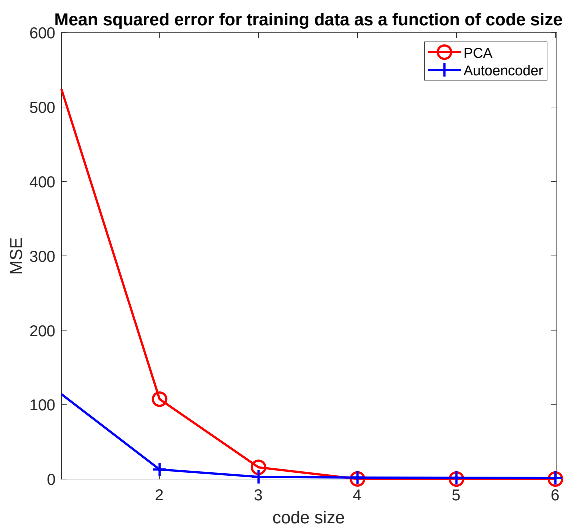

Figure 1: Plot of MSE for problem 1.

   (b) Train an autoencoder with the ReLU activation function using the data with varying size of the code h. Plot the MSE as a function of h.

   Ans: The plot of MSE achieved by the autoencoder trained for 2000 epochs is shown in Figure 1. Although it is hard to see in the figure, the linear encoder/decoder achieves smaller MSE when the code size is 3 or larger (but the MSE is already very small).

2. Consider the optimal autoencoder with linear encoder and decoder in problem 1(a) above. For a fixed code dimension h, explain how the MSE in (1) of the optimal autoencoder can be computed from the singular values of $\mathbf{X}$.

Ans: Recall that the singular value decomposition of $\mathbf{v}X$ is given by $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ contain the left singular vectors and the right singular vectors, respectively, as columns, and $\mathbf{\Sigma}$ is a diagonal matrix with singular values as diagonal elements. This can be rewritten as follows.

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{7} \sigma_i \mathbf{U}_{\cdot i} \mathbf{V}_{\cdot i}^T,$$

where $\mathbf{U}_{\cdot i}$ and $\mathbf{V}_{\cdot i}$ are the i-th column of $\mathbf{U}$ and $\mathbf{V}$, respectively.

For fixed code dimension h, the reconstructed input matrix $\hat{\mathbf{X}} = \sum_{i=1}^{h} \sigma_i \mathbf{U}_{\cdot i} \mathbf{V}_{\cdot i}^T$, and $MSE(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{2} \sum_{i=h+1}^{7} \sigma_i^2$.