PCA and Autoencoder

Wavelix

1 Tasks

1. Implement PCA on the Wine dataset, with only the first 2 components considered. Then, use the 2 components to reconstruct the data, and visualize the output.
2. Implement both linear and non-linear autoencoder on the Wine dataset. Then reconstruct the data using the trained autoencoder, and visualize the output.
3. Consider the reconstruct error of the PCA, linear autoencoder, and non-linear autoencoder. Analyze the result and discuss which method provides the best reconstruction accuracy and why.

2 Solutions

2.1 PCA

The core idea of PCA is to project high-dimensional data into a lower-dimensional data. Firstly we normalize the data by

$$X_{\text{normalized}} = \frac{X - \mu}{\sigma}$$

which ensures all features are on the same scale, and avoids bias from varying feature ranges.

Then, we compute the covariance matrix of the normalized data by

$$\mathbf{Cov}(X_{\text{normalized}}) = \frac{1}{n-1} X_{\text{normalized}}^T X_{\text{normalized}}$$

and obtain eigenvalues and eigenvectors. Sort the eigenvalues in descending order and select the top 2 eigenvectors to form the projection matrix `self.reduced_data`.

After that, project the data into the 2-dimensional space using

$$X_{\text{reduced}} = X_{\text{normalized}} \cdot V$$

where V is the matrix of the top 2 principal eigenvectors. The result is a reduced representation of the original data in the principal component space.

The `reconstruct(self)` method demonstrated the performance of PCA's reconstruction. The reconstruction is calculated as

$$X_{\text{reconstructed}} = X_{\text{reduced}} \cdot V^T$$

The reconstruction error is computed by

$$\text{Reconstruction Error} = \frac{1}{m} \sum_{i=1}^{m} \left|\left| X_{\text{normalized}}^i - X_{\text{reconstructed}}^i \right|\right|^2$$

2.2 Autoencoder

In this section, we construct a class named Autoencoder, which can implement both linear-autoencoder and non-linear autoencoder. The autoencoder has both an encoder and a decoder. The encoder maps the input data X into a lower-dimensional space, while the decoder reconstruct X from the compressed data. This process minimizes reconstruction loss, which is represented as

$$\text{Loss} = \frac{1}{m} \sum_{i=1}^{m} \left|\left| X^i - \hat{X}^i \right|\right|^2$$

Similar to the process of PCA, we firstly normalize the data. Here we can choose the activation function from **tanh**, **sigmoid** and **ReLU** for the hidden layers of non-linear autoencoder.

In forward propagation, each layer apply the following transformation:

$$Z = XW + b$$

$$A = f(Z)$$

where f is the activation function. For a linear autoencoder, the activation function is bypassed.

In backward propagation, we calculate the gradient for the weights and bias by

$$\delta = \frac{\partial L}{\partial Z} = (A - Z) \cdot f'(Z)$$

$$\frac{\partial L}{\partial W} = \frac{1}{m} A^T \delta$$

$$\frac{\partial L}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} \delta^{(i)}$$

We implement BGD update, and the parameters is updated by

$$W = W - \lambda \frac{\partial L}{\partial W}$$

$$b = b - \lambda \frac{\partial L}{\partial b}$$

3 Results

Figure 1 shows the visualization of principal components in the two dimensional space, which is generated by PCA. The reconstruction loss is 0.445937.

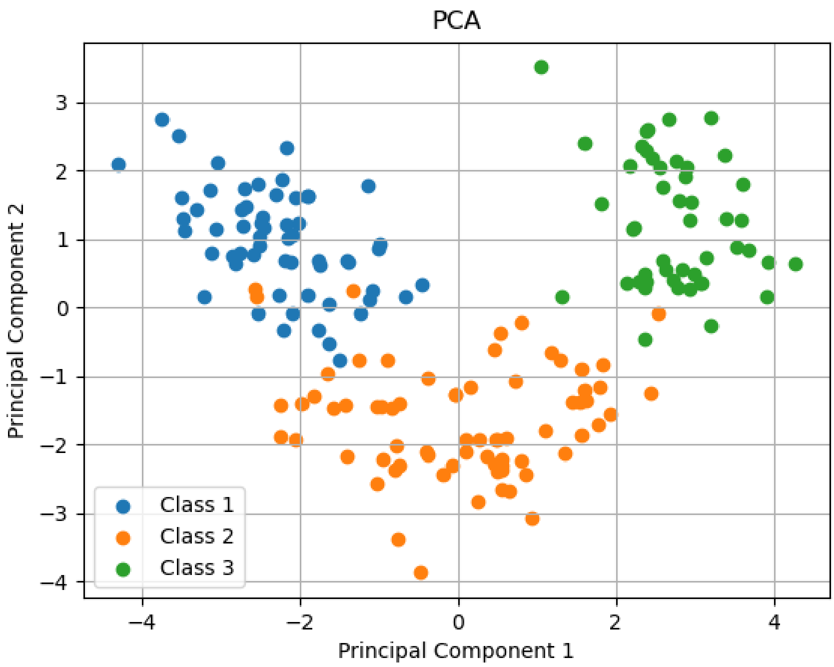
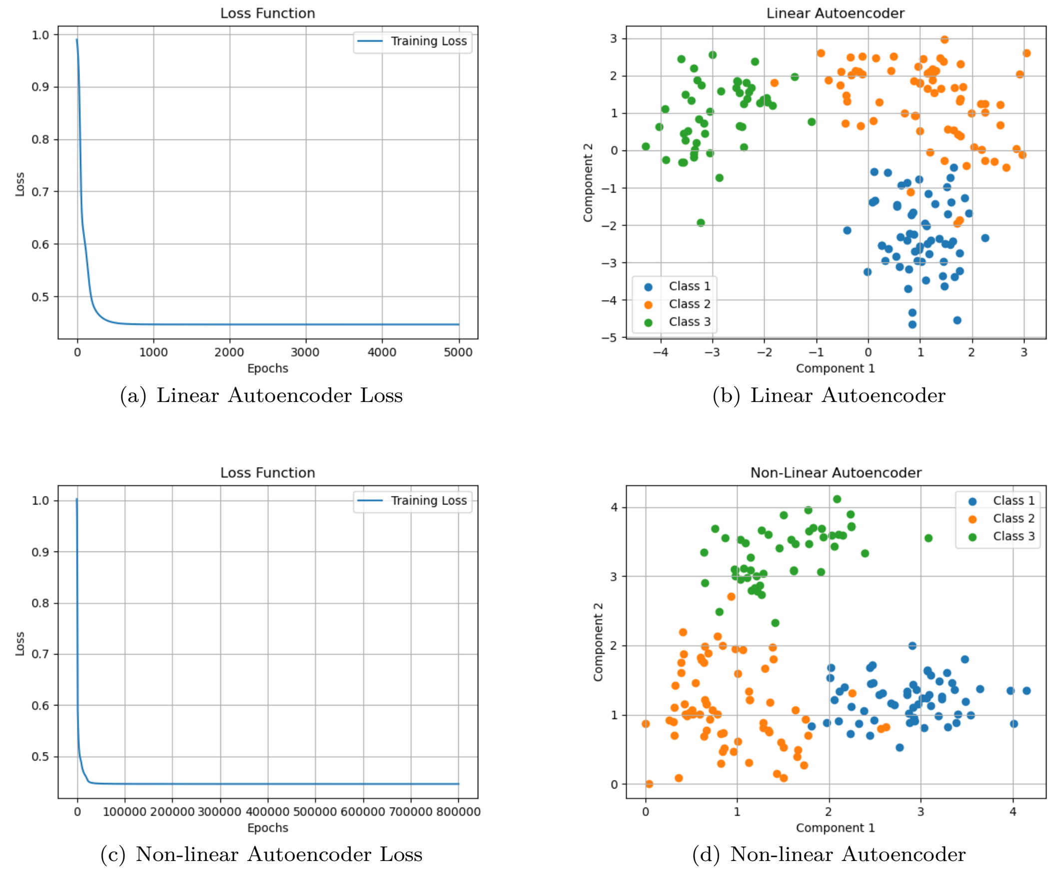


Figure 1: PCA

Figure 2 shows the visualizations of principal components in the two dimensional space, which are generated by linear autoencoder (lr=0.01, epochs=5000) and non-linear autoencoder (**ReLU**, lr=0.001, epochs=800000), with only one hidden layer with 2 neurons. The reconstruction error for them are all 0.445937, the same with the performance of PCA.



(a) Linear Autoencoder Loss

(b) Linear Autoencoder

(c) Non-linear Autoencoder Loss

(d) Non-linear Autoencoder

Figure 2: Autoencoder

4 Analysis

During the training, we found the smallest reconstruction error of linear autoencoder is 0.445937, the same with PCA, because the goal of linear autoencoder is to find

$$\min_{W,V} \frac{1}{2N} \sum_{n=1}^{N} ||x^n - VWx^n||^2$$

In other words, the optimal solution is PCA for the case when the mean of the data is 0.

Theoretically, the nonlinear autoencoder can provide the best reconstruction accuracy, since it can learn nonlinear patterns from the dataset, while the other two method can only learn linear patterns. However, with varies hyperparameters and layer size tried, during the training, we found the lowest reconstruction error is also the same with that of PCA. The reason for that may be the training data can already be well represented by linear model, and the linear encoder can already capture the main features of data. In that case, although non-linear autoencoders are more complex, they do not have additional non-linear patterns to learn, thus failing to demonstrate advantages.