Machine Learning Course Autumn semester

By Dr. Mahdi Eftekhari

Noise Reduction Using PCA and Autoencoders

Objective

The aim of this practice is to implement two methods for noise reduction:

1. **PCA (Principal Component Analysis):** A classical dimensionality reduction technique that removes noise by reconstructing data using its principal components.
2. **Autoencoders:** A neural network-based, non-linear dimensionality reduction technique that reconstructs clean data from noisy inputs.

By the end of this practice:

- **Understand** how PCA and autoencoders work for dimensionality reduction and noise removal.
- **Implement** these methods to reconstruct noisy images from the Fashion MNIST dataset.
- **Compare and analyze** the reconstructed results from both techniques.

Applications of PCA in Noise Reduction

- PCA reduces data dimensions by retaining only the most significant components, which represent the most critical information in the dataset.

- In the context of noise reduction, PCA reconstructs the data using these important components while discarding less significant components, which often contain noise. This helps in reducing the overall noise in the images.

Applications of Autoencoders in Noise Reduction

- Autoencoders are neural networks trained to reconstruct input data. They consist of an encoder that compresses the input into a smaller latent representation, and a decoder that reconstructs the original data from this compressed representation.

- The **latent space** learned by the encoder filters out irrelevant noise and retains only the essential patterns, making autoencoders effective for noise reduction.

- This **non-linear approach** often outperforms PCA in handling complex datasets and noise structures.

Step-by-Step Guide

Step 1: Load the Fashion MNIST Dataset and Add Noise

1. **Load the Fashion MNIST dataset**: The dataset contains 28x28 grayscale images of clothing items. Normalize the images to a range of [0, 1] to prepare for further processing.

2. **Add Gaussian noise**: To simulate noisy images, add Gaussian noise to the images. This will introduce random noise into the images to test the effectiveness of noise reduction methods.

Step 2: PCA Implementation for Noise Reduction

1. **Flatten the images into vectors**: Convert the 28x28 images into 1D vectors (784 dimensions) to apply PCA.

2. **Apply PCA**: Use PCA to compute the top k principal components of the noisy data. These components will capture the most important features of the images.

3. **Reconstruct the images**: Reconstruct the images using these top components, effectively reducing the noise in the images by discarding the less important components.

Step 3: Autoencoder Implementation for Noise Reduction

1. **Define the autoencoder architecture**: Create an encoder that compresses the input image and a decoder that reconstructs it back to its original size.

2. **Train the autoencoder**: Use noisy images as input and clean images as target outputs for training. The model will learn to reconstruct the clean images from the noisy ones.

3. **Use the trained model**: Once trained, use the autoencoder to reconstruct noisy test images and reduce the noise.

Step 4: Compare Results

1. **Visualize the images:**
   - Show the original images.
   - Show the **noisy images** (the ones with added Gaussian noise).
   - Show the **PCA-reconstructed images** (denoised by PCA).
   - Show the **autoencoder-reconstructed images** (denoised by the trained autoencoder).

2. **Analyze the results:** Compare how well PCA and autoencoders have reduced the noise and preserved the important features of the images.

Detailed Explanation of Step 3

What is an Autoencoder?

An autoencoder is a type of neural network that learns to compress data (reduce dimensions) and then reconstruct it back to its original form. It consists of two main parts:

1. **Encoder**: Compresses the input data into a smaller latent representation (a bottleneck layer).
2. **Decoder**: Reconstructs the original data from the latent representation.

For noise reduction:

- The encoder focuses on extracting meaningful features while ignoring noise.
- The decoder reconstructs the clean image from these features, effectively reducing noise in the process.

What You Need to Do:

1. **Define the Autoencoder Architecture**:
    - The input layer matches the size of the flattened image (28x28 = 784).
    - The encoder reduces the input size to a smaller latent dimension (e.g., 100).
    - The decoder reconstructs the original input size (784).

2. **Train the Autoencoder**:
    - Use noisy images as input and clean images as the target during training.
    - The model learns to map noisy images to their denoised versions by minimizing the reconstruction error (using Mean Squared Error (MSE) as the loss function).

3. **Use the Trained Model**:
    - After training, use the trained autoencoder to denoise the test images by passing the noisy images through the encoder and decoder.

Good luck ☺