Application of Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) on MNIST Dataset

Mihika Chawda, U23AI102

1 Introduction

The MNIST dataset contains handwritten digits from 0 to 9, with 60,000 training images and 10,000 test images. Each image is 28×28 pixels in grayscale. The objective of this study is to analyze and compare two dimensionality reduction techniques: **Singular Value Decomposition (SVD)** and **Principal Component Analysis (PCA)** on MNIST images.

2 Methodology

2.1 Data Loading

The MNIST images and labels are stored in IDX file format. We use Python to read the binary files into NumPy arrays. The functions used are as follows:

Listing 1: Load MNIST IDX files

```
import numpy as np

def load_images(filename):
    with open(filename, 'rb') as f:
        magic = int.from_bytes(f.read(4), 'big')
        num = int.from_bytes(f.read(4), 'big')
        rows = int.from_bytes(f.read(4), 'big')
        cols = int.from_bytes(f.read(4), 'big')
        buffer = f.read(rows * cols * num)
        data = np.frombuffer(buffer, dtype=np.uint8)
        return data.reshape(num, rows, cols)

def load_labels(filename):
    with open(filename, 'rb') as f:
        magic = int.from_bytes(f.read(4), 'big')
        num = int.from_bytes(f.read(4), 'big')
        buffer = f.read(num)
        return np.frombuffer(buffer, dtype=np.uint8)
```

2.2 Singular Value Decomposition (SVD)

Given an image matrix $X \in R^{m \times n}$, SVD decomposes it as:

$$X = U \Sigma V^T$$

where U and V are orthogonal matrices and Σ is a diagonal matrix containing singular values. The image can be reconstructed using only the top k singular values for compression.

Listing 2: SVD on a sample image

```
U, S, VT = np.linalg.svd(sample_image, full_matrices=False)
k = 50
S_reduced = np.diag(S[:k])
reconstructed_svd = U[:, :k] @ S_reduced @ VT[:k, :]
```

2.3 Principal Component Analysis (PCA)

PCA identifies principal components capturing maximum variance. Each image is flattened into a vector. PCA is fit on multiple images to reduce dimensionality:

Listing 3: PCA on MNIST images

```
from sklearn.decomposition import PCA

X_train_flat = X_train[:500].reshape(500, 28*28)
pca = PCA(n_components=k)
pca.fit(X_train_flat)
sample_flat = X_train_flat[0].reshape(1, -1)
reconstructed_pca = pca.inverse_transform(pca.transform(
    ↪ sample_flat)).reshape(28,28)
```

3 Results and Visualization

The following figure compares the original image with its reconstruction using SVD and PCA with $k = 50$ components:

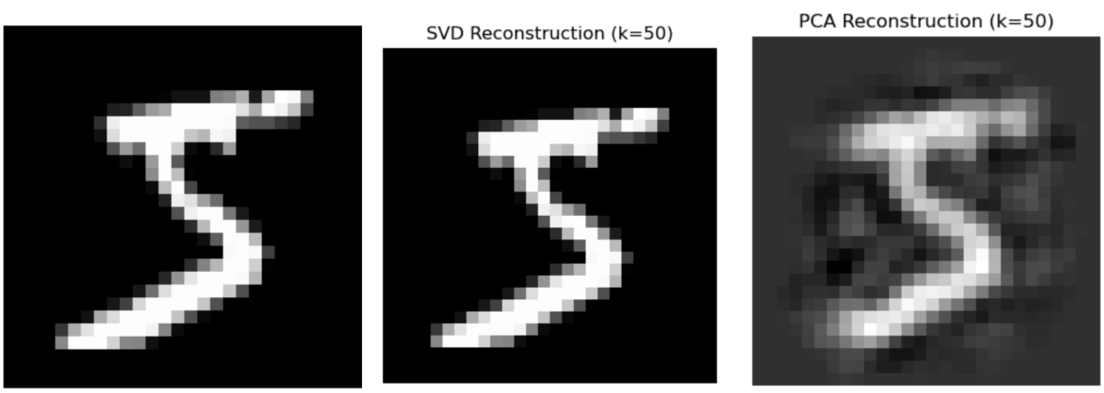

Figure 1: Original, SVD Reconstruction , PCA Reconstruction

3.1 Compression Ratio Analysis

The compression ratio for SVD is calculated as:

$$CR = \frac{m \cdot n}{k \cdot (m + n + 1)}$$

where m, n are image dimensions and k is the number of singular values used. Higher k leads to better reconstruction but lower compression.

3.2 Observations

- Both SVD and PCA can effectively reconstruct MNIST images with fewer components.
- SVD reconstructs each image individually, while PCA leverages variance across multiple images.
- For $k = 50$, visual quality is very good, and compression ratio is significantly reduced.

4 Conclusion

SVD and PCA are powerful dimensionality reduction techniques for image data. SVD is suitable for per-image decomposition, while PCA captures global variance across images. Both methods provide effective compression and reconstruction, which is valuable for storage and analysis of large image datasets.