EE 658/758 Machine Learning in Engineering
Spring 2024

Assignment #3: Dimensionality Reduction
Due Date: Thursday, April 4th, 2024

PCA and Neural Network Classifier on MNIST

In this programming assignment, we will use the Principal Component Analysis (PCA) for dimensionality reduction on the MNIST dataset, followed by the implementation of a classifier. The goal of this assignment is to understand the impact of dimensionality reduction on the performance of classifiers. The bjectives are to:

- Implement PCA for dimensionality reduction on the MNIST dataset.
- Build and train a classifier to recognize handwritten digits.
- Evaluate the impact of dimensionality reduction on neural network classifier performance.

The MNISTdataset comprises 70,000 grayscale images of handwritten digits, divided into a standard split of 60,000 training images and 10,000 testing images. Each image represents a single digit from 0 to 9. Each image in the dataset is 28 pixels by 28 pixels, resulting in a total of 784 pixels per image. These images are grayscale, with pixel values ranging from 0 (black) to 255 (white), indicating the intensity of each pixel. Corresponding to each image is a label that indicates the digit it represents (0 through 9). These labels are used for supervised learning tasks, where the goal is often to predict the digit based on the pixel values.

Part I: Data Preparation

1. Load the MNIST dataset.
2. Normalize the images so that pixel values are between 0 and 1.
3. Split the dataset into training (80%) and testing (20%) sets.

Part II: Implement PCA

1. Apply PCA to reduce the dimensions of the dataset. Initially, choose 50 components for dimensionality reduction.
2. Explore the variance explained by the selected number of components and adjust if necessary to retain at least 95% of the variance.
3. Show the Scree plot.

Part III: Classification with Logistic Regression

1. Implement a Logistic Regression classifier.
2. Train the classifier on the original high-dimensional data and evaluate its performance on the test set.
3. Train the classifier on the PCA-reduced training data and validate its performance using the PCA-reduced test data.

Part IV: Neural Network Classifier

1. Design a simple neural network for digit classification. The network should have an input layer that matches the number of PCA components, at least one hidden layer, and an output layer with 10 units (corresponding to the 10-digit classes).
2. Use a softmax activation function in the output layer and a suitable loss function for classification.
3. Train the neural network on the PCA-reduced training data and validate its performance using the PCA-reduced test data.
4. Experiment with different hyperparameters for the neural network, such as the number of hidden layers, the number of neurons in each layer, the learning rate, activation function, and the type of optimizer. How do these changes affect the model's performance on both the original and reduced datasets?

Part V: Performance Evaluation

1. Evaluate the trained neural network on the test set and report metrics such as accuracy and precision.
2. Compare the performance of the neural network classifier on the PCA-reduced data to the performance of the Logistic Regression classifier.
3. Discuss the impact of dimensionality reduction on the classifier's performance, training time, and model complexity.

Submission Guidelines

Submit a Jupyter Notebook containing the code, output, and a brief discussion of your findings for each task.

Ensure your code is well-commented to explain the implementation and decision-making process.

Include visualizations where appropriate, such as a plot of the variance explained by the PCA components and training/validation loss curves for the neural network.

Downloading the MNIST Dataset

```
from sklearn.datasets import fetch_openml

# Fetch the MNIST dataset
mnist = fetch_openml('mnist_784', version=1)

# The data key contains the features, and the target key contains the labels
X, y = mnist["data"], mnist["target"]
```