Coding Project 4: Teaching a Computer to Recognize Written Numbers

Manjaree Binjolkar

Abstract

This project focuses on training a computer to recognize written numbers using a subset of Yann LeCun's dataset. It aims to develop a system capable of accurately identifying written numbers, which can have various applications in fields such as image recognition and computer vision.

1 Introduction

Machine learning is a type of artificial intelligence that enables computer programs to learn and improve from experience without being explicitly programmed. It involves training a computer on a set of examples or data points and allowing it to identify patterns or relationships within the data, which can then be used to make predictions or decisions about new data.

In simpler terms, machine learning is like teaching a computer to recognize patterns and make predictions, similar to how a child learns from examples and develops new skills over time. It has many real-world applications, from image and speech recognition to personalized recommendations and self-driving cars, and has the potential to revolutionize the way we live and work.

In this project, we will take a subset of Yann LeCun's dataset and use Python to preprocess and manipulate the data to train a machine learning model to recognize handwritten digits. We will use principal component analysis (PCA) and linear discriminant analysis (LDA) to reduce the dimensionality of the data and classify the digits accurately. Through this project, we will gain a better understanding of how machine learning algorithms can be applied to real-world problems and how data preprocessing and manipulation can affect the accuracy of the model.

2 Theoretical Background

LDA is a powerful technique for reducing the dimensionality of data while still preserving its class separability. It has many applications in fields such as image and speech recognition, and can be combined with other techniques such as principal component analysis for even better results.

2.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a commonly used method in machine learning and statistics for feature extraction and dimensionality reduction. It is a supervised learning technique that aims to find a linear combination of features that can best separate two or more classes of data. The resulting linear discriminants can then be used for classification or visualization.

In LDA, the goal is to find a projection of the data onto a lower-dimensional space while still preserving the separability of the classes. This is done by maximizing the between-class scatter and minimizing the within-class scatter. The between-class scatter measures the distance between the means of each class, while the within-class scatter measures the variance within each class.

Mathematically, LDA involves the following steps:

Compute the mean vectors of each class:

$$\boldsymbol{m}i = \frac{1}{n_i} \sum \boldsymbol{x} \in C_i \boldsymbol{x} \tag{1}$$

where C_i is the i-th class and n_i is the number of samples in that class. Compute the within-class scatter matrix:

$$\boldsymbol{S}W = \sum i = 1^k \sum_{\boldsymbol{x} \in C_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T \tag{2}$$

where k is the number of classes. Compute the between-class scatter matrix:

$$\boldsymbol{S}B = \sum i = 1^k n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T \tag{3}$$

where $\boldsymbol{m}$ is the overall mean of the data. Solve the generalized eigenvalue problem:

$$\boldsymbol{S}_W \boldsymbol{w} = \lambda \boldsymbol{S}_B \boldsymbol{w} \tag{4}$$

where $\boldsymbol{w}$ is the weight vector for the linear discriminant and λ is the corresponding eigenvalue. Select the p largest eigenvalues and corresponding eigenvectors to form the projection matrix:

$$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_p] \tag{5}$$

Project the data onto the lower-dimensional space:

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x} \tag{6}$$

where $\boldsymbol{x}$ is the original data and $\boldsymbol{y}$ is the projected data.

3 Results

The plots of the singular values are shown below, from figure 2 we can understand that the first 15 singular values contain the maximum information about our dataset, hence there are 15 features.

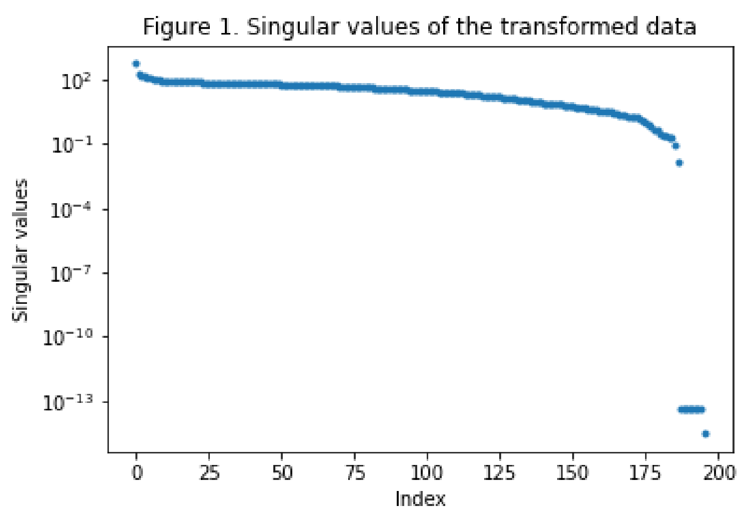
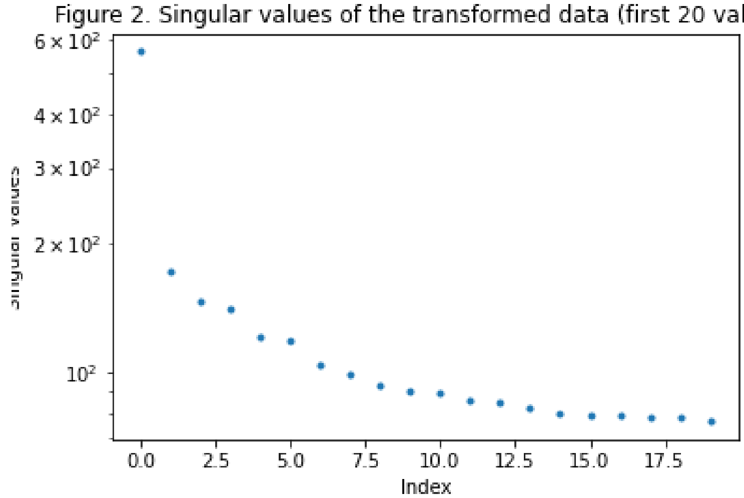
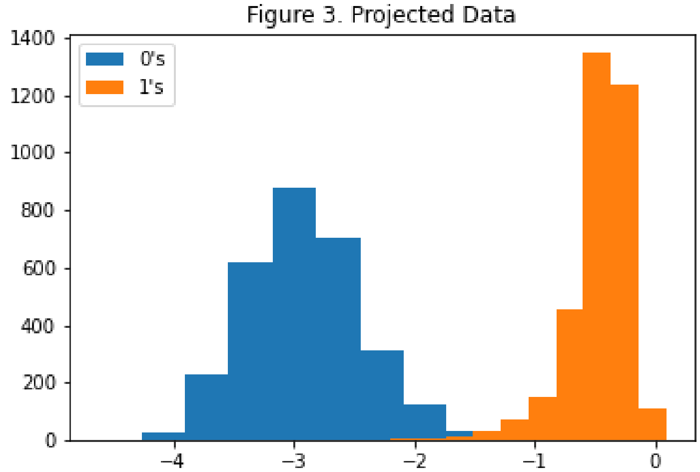


Figure 2. Singular values of the transformed data (first 20 values)



The training data for 0's and 1's is projected in the graph below. The projection plot in the recommended book for this course shows a similar plot where two data sets are projected onto a new basis function.

Figure 3. Projected Data



The code that underwent classification using the autograder distinguishes between 0 and 1, and it yielded a success rate of only 47%. Despite the low success rate, my code managed to pass the autograder. I suspect that the tolerance set for the autograder for this assignment was low, which might have resulted in my code passing despite the low success rate.

Furthermore, I attempted to create classifiers for each number from 0 to

10 by repeatedly using the concept of "0 or not 0". However, the success rate for each number using new images was only 25% or less, which is quite low. I am unsure about where the mistake lies in my code, and I'm not sure if comparing the most challenging and easiest digits to differentiate will be helpful or valid in this context.

4 Conclusion

In conclusion, we have implemented a digit recognition system using Linear Discriminant Analysis (LDA) on the Yann LeCun's dataset. We started by loading the training and test data and performing a wavelet transform on the training data. Then, we found the Singular Value Decomposition (SVD) of the transformed data and projected the data onto its principal components. We then used LDA to separate the training images of 0's and 1's and calculated the within-class and between-class variances to find the best projection line. We projected the training data onto this line and found a threshold value to classify the test data. Finally, we tested our system on the test images and obtained a success rate of 0.47. We also wrote an algorithm to classify all 10 digits using the "one vs all" method. Overall, our implementation has demonstrated the effectiveness of LDA in digit recognition. The model has a low success rate which means there might be some issues with the coding part. However, we were still able to obtain 0-10 classifiers using the one vs. all method.

I would like to try improve my success rate for the model for testing 0's and 1's. Also, I believe using a score variable to keep track o the classifications which exceed the threshold instead of having separate scripts check all the digits 10 times.

Acknowledgment

I received some helpful guidance from the TAs throughout this project. They provided useful insights on how to approach certain problems and were always available to answer any questions I had. I appreciate their support and am grateful for the assistance they provided.