Coding Project 4: Teaching a Computer to Recognize Written Numbers

Tyler Shakibai

Abstract

Digit classification has become a ubiquitous benchmark task in machine learning and computational data analysis. Using a subset of Yann LeCun's MNIST dataset, we will train a classifier that can discern values from images of handwritten digits. In this analysis, we will predominantly be using linear discriminant analysis, taking advantage of the desirable properties of the modes of the principal components. To train the classifier, we will be iteratively training smaller classifiers that can only make distinctions on a set of two numbers, which in unison, are capable of making multi-class decisions for the ten handwritten digits.

1 Introduction

Machine learning is a field at the intersection of mathematics, statistics, and computer science that involves building computational models that are capable of "learning." For the purposes of this project, we will only consider supervised learning tasks such as digit classification. In supervised learning, models are provided with information to train on, and the associated labels that the input data is mapped two. Machine learning architectures are designed to recognize patterns between input and output data, with the goal of being able to predict future outputs given new and unseen inputs. For example, in this project, we want to give a machine learning model information in the form of handwritten images with the corresponding labels so that model can learn the relation between input and output. Afterward, the model will be given new handwritten digits it has not yet seen and hopefully, it will generalize well and make an accurate classification prediction.

In this report, we will use linear discriminant analysis (LDA) to perform a number of small binary classification tasks to distinguish between digits. Each handwritten image is represented as a 28×28 greyscale image. The LDA classifier is used to determine the most and least distinguishable digits on the dataset we are working with. When pieced together, these binary classifiers can make multi-class decisions in the aggregate.

2 Theoretical Background

In this section, we will delve into the theoretical underpinnings of linear discriminant analysis to understand how it can be applied to our task. This discussion will be focused primarily on what makes LDA useful as both a linear classifier and a tool for dimensionality reduction.

2.1 Linear Discriminant Analysis

Linear discriminant analysis is a technique used to find a linear combination of variables, or features, that aptly characterizes or distinguishes multiple classes of objects. Objects can be defined as deterministic events, outcomes, or state representations. The goal of LDA is to find a fitting projection that minimizes the distance between data points within the same class while maximizing the distance between data points that are in different classes.

To illustrate let us consider the classes X_1 and X_2 which are matrices composed of different sets of data. We make the claim that each has a vector of means of each row μ_1 and μ_2. We can definite the intra-class scatter matrix as

$$S_\omega = \sum_{i=1}^{2} (x_i - \mu_i)(x_i - \mu_i)^T$$

and the inter-class scatter as

$$S_b = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

where ω is the solution to the eigenvalue problem

$$S_b \omega = \lambda_{\max} S_\omega \omega$$

However, since we are interested in making multi-class decisions for digit classification, we need to generalize to N classes as such

$$S_\omega = \sum_{i=1}^{N} (x_i - \mu_i)(x_i - \mu_i)^T$$

$$S_b = \sum_{i=1}^{N} (x_i - \mu_i)(x_i - \mu_i)^T$$

3 Results

First, look at a snapshot of our data to understand what we are looking for from linear discriminant analysis and our principal components.

Figure 1: Subset of the handwritten digit training data.

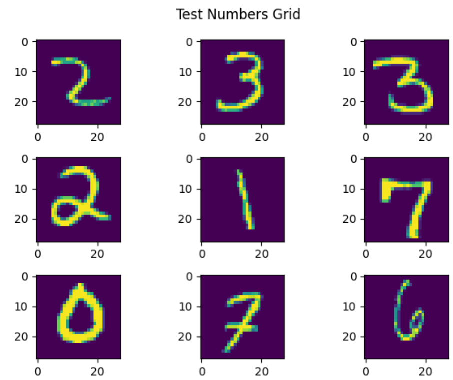

Figure 2: Subset of the handwritten digit test data.

We then examine the singular values we received from taking the discrete wavelet transform (DWT) of our training data. The capture of a significant portion of the energy, or information, of this data, we will utilize the first 15 singular values.

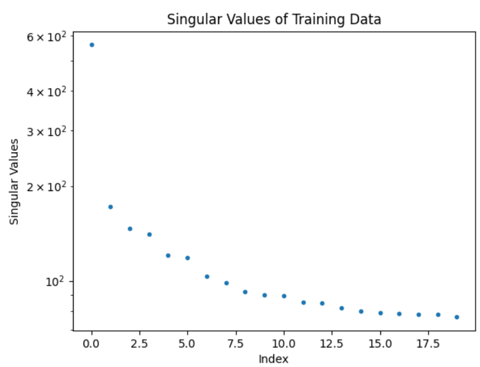

Figure 3: Value and indices of singular values extracted from training data.

Using the dimensionality reduction techniques we mentioned above, we then extracted the first four principal components of the processed training data.

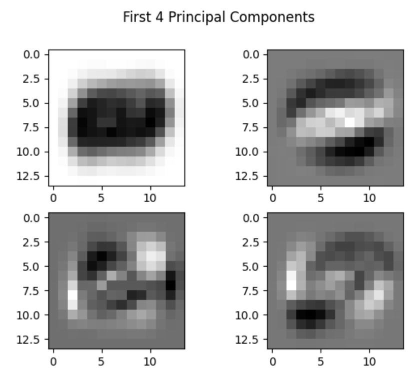

Figure 4: First four principal components of our training data.

Using available python libraries, we were able to plot an insightful projection of handwritten digits onto principle components two, three, and five.

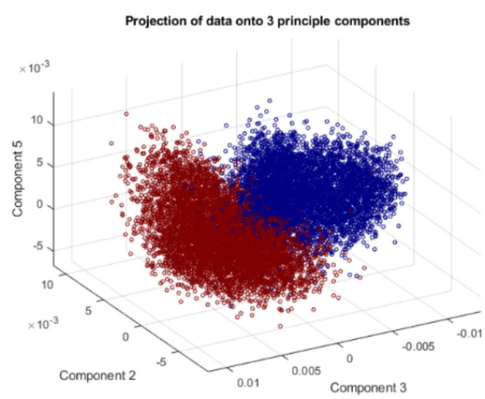

Figure 5: Three-dimensional projection of handwritten 2's and 3's on three distinct principle components.

4 Conclusion

We developed a reasonably effective digit classifier using linear discriminant analysis to distinguish between handwritten digits from Yann LeCun's MNIST dataset. We used PCA and LDA to conduct dimensionality reduction to alleviate feature redundancy. As is the goal of machine learning, we were able to provide our model with new handwritten digits to see how the classifier performs. The error on the test set was 0.75 or an accuracy of 25% which leaves plenty of room for improvement. We were also able to project our data onto a projected space composed of principle components two, three, and five.

Acknowledgment

I attended office hours with the course's TAs, Katie and Michael, and heavily utilized the Numpy and Scipy Documentation.