Paul-Jason Mello
Professor Shim
CMPE 257
April 28th, 2022

Lab 2 Report

Abstract

The goal of this assignment is to become acquainted with many different types of machine learning methods. The assignment categorizes these into four tasks which consist of an artificial neural network, natural language processing, recommender systems, and a Random Forest implementation. Each task has a series of questions which explore the processes of each machine learning method.

Introduction

In the following sections I will describe the procedures and methodology I took in exploring the many different machine learning methods. I will then cover the results and any interesting aspects I found. As each task is vastly different, I will explain the methodology for each.

Task 1: Artificial Neural Network

I began by downloading the cifar10 dataset from keras. After some initial questions I began to visualize the data dimensions. In total there were 60,000 image entities split into 32 x 32 pixels. The dataset contains 10 unique labels for each image. After some initial visualizations and asserting that each class has an equal number of images we began building our model. Our base model uses a network of 128 -> 32 -> 10 neurons in each layer. We test multiple combinations of activation functions and optimizers. Next we tested out some regularization tactics, but these proved unimportant. As a result our final model uses a selu activation function in the 128 and 32 neuron layer, while the output layer uses a sigmoid. Then we found that Adamax works best in achieving accuracies above 50%. We plotted our f1_score and accuracy at each epoch and concluded this section.

Task 2: Natural Language Processing

In this section we began by downloading some important nltk tools for preprocessing our data for natural language processing. We then downloaded the sentiment analysis data which consists of movie reviews and their associated feelings, positive/negative. Next I preprocessed the data to remove breaks, and any characters which are not in the alphabet. From there I split the words each row and then lematized them. After this I attempted to build the model and came to office hours for help however I was unable to figure out the implementation in code.

Conceptually we use word2vec then vectorize the input. From there we can merge the words and build a random forest classifier. Overall, this was a very difficult task for me to complete.

Task 3: Recommender Systems

In the recommender systems task we started by downloading movie data, rating data, and user data. Following some brief questions we create a matrix consisting of MovieID's as a row, and UserID's as a column. In each observation the rating data is present. This created a sparse matrix filled with a significant amount of 0's. Next we performed SVD, extracted the top 50 components. Separately to SVD, I then made the appropriate covariance matrix. Now that we had eigenvalues and vectors we extract the top 50 components using PCA. At this point I was unable to figure out how to find the closest movies using the components for SVD and PCA. Despite this I believe that we follow a similar route as seen in HW 10 for recommender systems. In this regard, we use cosine similarity and then equate the rows to the appropriate movie ID in the initial data frame and display the top 10 results.

Task 4: Random Forest Implementation

In this section we were given skelton code and asked to build a model which properly acted as a random forest classifier. However, in this implementation we used decision trees to ease the process. We began by building a split mechanism to cut the data into training and test data based on a predetermined ratio. From there I implemented a subsampling algorithm which randomly selects a certain fraction of rows and one for columns. After this I built the random forest train method which uses decision trees from sklearn to create a large random forest of trees. Once I integrated these methods I built the final method which predicts outputs from an input. This was interesting to see how simple this implementation of random forest can be.

Conclusion

This lab was interesting and provided a wealth of understanding about each of these machine learning methods. I can see how different each of these methods are when they are directly compared side-by-side. Uniquely, Task 4 was relatively intuitive and I enjoyed being able to build a working implementation of random forest. Overall, my favorite task was Task 1 while Task 2 was my least favorite.