Ben Wright

Matrix Methods in Machine Learning

Project Proposal

October 21, 2020

The dataset that will be used to practice knowledge developed in CS 532 will be the MNIST database of handwritten digits, found at <http://yann.lecun.com/exdb/mnist/>. The database, according to the page found via the link, is comprised of a training set of 60,000 and a test set of 10,000 images of centered, fixed-sized images.

Further, the website reveals that the training set is made of 30,000 handwritten digits from Census Bureau employees and 30,000 handwritten digits collected from high school students. The original pixelized binary scans have been normalized, meaning some grayscale pixels now appear due to anti-aliasing. Images are stored in a special file format called IDX, described on the site. At least 250 different writers are represented in the training dataset.

Also, the dataset is labeled (with the appropriate digit numbers) making it prime for analysis by supervised learning techniques. Namely, we propose to study the following question: For a fixed digit, can the computer be taught to tell whether a given handwritten image represents that digit? There are four main techniques which might be interesting to apply to this data: 1. Linear regression/ridge regression, 2. Support vectors, 3. Neural networks with several layers, and 4. Naïve Bayes.

Roughly, the idea of each method is as follows: Linear regression encodes the labels into a vector y, and the training data into a matrix X, and solves the associated least-squared-error problem. Various modifications can be made, like instead of a matrix X, using a low-rank approximation to X, or also weighting by the norm of the solution vector. The solution vector w can then be used to define a classifier which answers our question. A support vector machine constructs a classifier by choosing a hyperplane that separates the classes of data with a maximal margin (complications may arise where such a hyperplane does not exist). In this case, a hyperplane separates data into “represents 4” and “does not represent 4,” for example. A neural network is an algorithm that attempts to imitate the process of learning as done in the animal brain, the author does not know much more about it beyond that but intends to learn in and outside class by the deadline noted below. Finally, the naïve Bayes approach designs classifiers based on the assumption that entries of a feature vector are independent, and then assigns a class based on Bayes’ formula of probability theory. This method is especially useful when the data has more than two classes, which in our case, the data has 10 classes. It will be interesting to try this last method with various types of pre-processing, i.e., looking at different sets of features for the data.

The project will be implemented and maintained at the following link: <https://github.com/users/bwright8/projects/1>. The project will be written in Python and maybe some parts redone in R. The timeline to completion is subject to change but will be loosely as follows: By November 5, have data from IDX files loaded and pre-processed. By November 17, implement linear regression and naïve Bayes, have one-two pages of the final report written. By December 1, have all learning algorithms implemented and tested, have 3 pages of the final report and analysis written. This leaves 2 pages of conclusions and analysis to write before the final report is due on December 12.