Christian Carreras

CSC 548

05/03/2017

Final (Question 1: a1B)

The results I found using K-nearest-neighbor have been interesting as they have been informative. I discovered that the algorithm performs fairly well with most of my tests averaging around ninety-six percent accuracy. However, the success of this algorithm turns out to be double-edged sword. The more training data the algorithm has access to, the more accurate it will become. The more training data there is, the longer it will take to make a prediction. With a training dataset of sixty-thousand it takes around thirty-five seconds to make a prediction on my current system. This is due to the fact that the Euclidean Distance has to be mapped from every single training instance to the current test instance in order to compile a list of nearest neighbors. It turns out that each prediction is around O(n*m) where n is the size of the training dataset and m is the size of the dataset image (28x28). With my dataset, that comes out to around O(47,040,000) which is quite substantial for one test. I cannot begin to imagine how much bigger the training set has to grow to see an improvement to a near-perfect level. I hypothesis that the training set and the time to complete the prediction would have to grow exponentially to see even the smallest improvements of a couple percent. That being said, I can also theorize that the graph of dataset size to accuracy is non-linear. The closer the algorithm's accuracy gets to one-hundred percent, the closer the dataset gets to exponential growth. These results have led me to believe that K-nearest neighbor is an excellent algorithm to get up and running quickly while having a favorable degree of accuracy. Be that as it may, if accuracy needs to be as close to one-hundred percent as possible, one may be waiting longer than it takes to implement the algorithm itself.

Sample Output