Yusdel Lima Lorenzo

May 6th, 2018

Image Classification

The goal of this project was to design and implement two machine learning algorithms the first of which is the perceptron. At first the thought of creating an AI program that recognize faces and digits scared me because I did not know if i was able to do it. I recently learned python and decided on that programming language because of the ability to use lists.

My perceptron has a feature for each pixel in the face or digit and each feature has a weight attached to it. So the perceptron learns these weights on the training data and saves them for the future. When it sees a new image it multiplies each pixel with the weight at the corresponding location. Then it sums all the numbers up and if it is positive it guesses that it is a face. To my amazement this worked amazingly, I achieved around 86% accuracy on aces as opposed to my failed attempts which only got around 60%. When it came to the digits it was going to be much harder. This part of the project fooled me until i realized I could just have a seperate weight set for each different number which in the scope of the project was ten sets. Using the same methods and algorithms from the faces perceptron I was able to get the digits working.

The Naive Bayes Classifier was the next algorithm left to complete and it was harder then the perceptron at first but after looking at the lecture slides i was able to create a sudo program and slowly work towards it. My Naive Bayes Classifier only uses the number of white pixels per image to calculate the probability of a face or digit. So the Naive Bayes Classifier learns all these probabilities and then when its confronted by a new image it counts the amount of white pixels and with that information it guesses what it is. I used the same technique that the perceptron uses when it came to the digits. Having separate sets of probabilities per digit allowed the Naive Bayes Classifier to guess the digits around 90%.

When it comes to the margin of error it came in the form of the precision of data extraction and at first this was a big issue because without testing and training on the correct data it would be impossible to implement these two algorithms. The running time for these algorithms are very impressive, for example the perceptron only trains the faces for 10 seconds using 10% of the data and incrementing to 100% of the data. Testing is done between the training and the perceptron gradually gets more accurate with more data. The perceptron algorithm runs for 400 seconds to train and test both the faces and digits.

From this project I learned that AI programs are very simple to implement yet the coding has to be very precise to achieve the best results. I could imagine these algorithms being used in a search and rescue drone to find stranded hikers or boaters or even in a silly camera application of a phone.