EE 551 Final Project Gabriella Tantillo 12/17/2020

My idea for my final project was inspired by a homework assignment from EE 695 Applied Machine learning. Given a dataset, my goal was to find the polynomial to fit the data. This is a pretty simple task, but I think it is effective way to understand machine learning. There are of course many ways to accomplish polynomial fitting in Python and python has many built in functions to use.

I generated a dataset by creating a random set of 50 numbers to be my set of x values. I then created a base equation to plug the x values in with some noise and generate the y values. I then was able to generate an polynomial to fit the dataset. I started by using the polyfit function from the NumPy package. This is quite straightforward because it takes in the x values, y values, the estimated degree of the polynomial and returns the coefficients in degree descending order. The coefficients, using NumPy’s poly1d allows for the coefficients to be considered a 1-dimensional polynomial, allowing the x values to be “plugged” into the equation and generate the y values. I checked my results empirically by graphing the generated polynomial and the dataset on the same graph.

In order to get more practice with Python, I also implemented a Genetic Algorithm using both Crossover and Mutation methods of generating offspring. I used this <https://towardsdatascience.com/a-simple-genetic-algorithm-from-scratch-in-python-4e8c66ac3121> article to get me started. Genetic algorithms mimic how natural selection works, choosing the best features to move through the new generations. I started by creating a “population” aka my starting point of guesses that would lead me to what numbers provide the best result. After the one time generation of the population is created the Genetic Algorithm follows the pattern of selecting the best traits by using a fitness score, and performing crossover and mutation to get the best traits even better.

I was able to practice for loops, and a lot of array manipulations by implementing a genetic algorithm. The algorithm could use some future fine tuning of parameters, but empirically it looks quite good. Some runs of the genetic algorithm produce garbage results, but it eventually generates a polynomial that fits quite well.