CALIFORNIA SCIENCE & ENGINEERING FAIR
2019 PROJECT SUMMARY

Name(s) **Maya Basu**	**Project Number** **S1402**

Project Title

Creating a Neural Network to Play the Game of Connect Four Using a Genetic Algorithm

Abstract

Objectives

My project explores a genetic algorithm as a method for producing a resultant digital neural network competent at a strategy game, namely Connect Four. I quantify competency as a being able to beat or tie the general population at Connect Four more than 90% of the time. Unsupervised methods have produced successful AI such as AlphaGo Zero. A genetic algorithm is an alternate type of unsupervised learning which is unproven and relatively unstudied.

Methods

I wrote code in the Rust programing language using CLion as my IDE to implement a digital neural network and my genetic algorithm, and validated it using TensorFlow. My code is hosted open source on GitHub for version management. I ran my program several times with different hyperparameters on both a Macintosh and a Linux workstation. I created a logic based Connect Four player to act as a benchmark for measuring progress. I ran my genetic algorithm for 100,000 generations and graphed the ability of the seed of each 500th generation against the benchmark.

Results

During the genetic algorithm, the mutation magnitude used the create each generation diminishes from 1 to a set value. I graphed four separate algorithms with different end mutation magnitudes using performance against the benchmark. Interestingly, the graphs are roughly step shaped, and "step" in order from least to greatest end mutation magnitude. I wondered what would happen if I selected the seed of each generation by using their performance directly against my benchmark, and recorded distinct step shaped graphs which also "step" in order from least to greatest end mutation magnitude.

Conclusions

I played the resultant neural network from every genetic algorithm and won, causing me to deem a study against the general population unnecessary. However, one of the neural networks did exhibit blocking strategies. Given the computing times used to train AlphaGo Zero, I may simply need to run my genetic algorithm for more than 300 times as long to achieve comparable results. However, my data suggests that sharply decreasing the mutation magnitude speeds up the genetic algorithm s performance. This is an important result for genetic algorithms in general, and I will probe how far this trend continues. I also plan to build concurrency into my program to make it faster, and look for gains in performance by using convolutional neural networks.

Summary Statement

My project explores a genetic algorithm as an alternate unsupervised method for producing a digital neural network competent at a strategy game.

Help Received

I wrote all of my code by myself in Rust and receive occasional help with debugging from my father. I obtained a student licence from WebStorm and used CLion as my IDE. I graphed my results using FreeMat. During my project, I discussed my plans with Drs. Basu, Skatter, and Quadri.