



**CALIFORNIA STATE SCIENCE FAIR
2011 PROJECT SUMMARY**

Name(s) M. Evan Wildenhain	Project Number J1418
Project Title Evolving Neural Networks to Play Mastermind	
<p style="text-align: center;">Abstract</p> <p>Objectives/Goals The objective is to write a computer program that trains neural networks using a genetic algorithm to play Mastermind. It is hoped that the program can train neural networks to be able to, on average, solve at least 1%, or 13 combinations, of all 1296 combinations of Mastermind. The impact of changing different settings of the neural networks and genetic algorithm will be explored.</p> <p>Methods/Materials A computer was used for programming and running the experiment. The Mastermind simulating program itself was coded by the researcher in Java using resources from both Heaton Research's open source Encog Project and Sun Microsystems. The program generates a population of neural networks, trains the networks on a random subset of Mastermind combinations using a genetic algorithm for 20,000 epochs, then exports a log summarizing the number of combinations solved by the best-evolved network as well as the score the network received from the fitness function.</p> <p>Results With four different settings, the program was able to evolve a neural network that could solve at least 13 out of 1296 combinations. The four settings that allowed the program to achieve this were with the baseline settings (off of which all of the other settings were varied), with the hidden layer size decreased, with the mutation rate increased, and with the crossover rate increased.</p> <p>Conclusions/Discussion Increasing the crossover rate allows more networks to survive, which may lead to greater diversity in the population; this diversity means a greater chance for a latent beneficial gene to survive until it is needed, producing more successful results. Also, introducing a "supermutation" function into the program greatly improves performance. Supermutation involves randomly mutating every neural network in the population when no progress is made after a certain number of epochs. This allows the neural networks to escape from local maxima and continue improving.</p>	
Summary Statement This project involves writing a Java program that uses a genetic algorithm to train a population of neural networks to play Mastermind.	
Help Received Parent helped with discussing the design of the program.	