Scott Munro
Lab 5 - Neuroevolution + Game

The Game:
The game was no more than a simple pong game. The game is set up with two paddles on each side of a square “field”. A ball flies across the field unless it hits a containing wall or a player’s paddle, in which case it ricochets and flies back the otherway. Every time the ball collides with either a wall or a paddle, the balls velocity increases. Players score when the ball crosses the other player’s paddle. A player wins the game when their score reaches 10 points.

The Neural Network:
Because the game itself was fairly simple, it made sense to represent the Computer AI fairly simply. Each “brain” is composed of 3 input neurons that each connect to one output neuron. Each input neuron represents a sensor that the AI has, and the 3 sensors are the distance between the AI paddle and the ball on the x-axis, the distance between the AI paddle and the ball on the y-axis, and the current y position of the paddle. I chose these three because I feel they most accurately represent the knowledge that a human player would have during the game. The output is essentially a step function that I implemented as a sigmoidal function. If they sum of the inputs and their weights are greater than zero than I have the AI Paddle move at a fixed speed downward, otherwise they move at a fixed speed upward. Interestingly enough, I found that even though the sigmoidal function was essentially a glorified step function, when I implemented the neurons with a regular step function they performed more poorly and learned slower.

Gen:	0	fit:	133
Gen:	1	fit:	133
Gen:	2	fit:	133
Gen:	3	fit:	133
Gen:	4	fit:	814
Gen:	5	fit:	921
Gen:	6	fit:	921
Gen:	7	fit:	921
Gen:	8	fit:	921
Gen:	9	fit:	921
Gen:	10	fit:	921
Gen:	11	fit:	921
Gen:	12	fit:	921
Gen:	13	fit:	968
Gen:	14	fit:	968
Gen:	15	fit:	968
Gen:	16	fit:	968
Gen:	17	fit:	968
Gen:	18	fit:	1714
Gen:	19	fit:	1714
Gen:	20	fit:	1714
Gen:	21	fit:	1714
Gen:	22	fit:	1976
Gen:	23	fit:	1976
Gen:	24	fit:	1976
Gen:	25	fit:	1976
Gen:	26	fit:	1976
Gen:	27	fit:	1976
Gen:	28	fit:	1976
Gen:	29	fit:	1976

AI Behavior over time:
Even though over multiple trials the evolutionary algorithm would find different weights to use for neural network, the way it evolved each time was usually one of two ways. Sometimes it would begin with the paddle not moving (fit: ~150), evolve to move to one side and stay there (fit: ~500), further evolve to move to one side and try to swiftly move back to catch the ball if it got too close (fit: ~1000), and would eventually just begin to behave in a way where it would do its best to just track the ball by moving back and forth at 30 unit intervals, keeping the ball within the paddles range (fit: 1976). Othertimes, it would get lucky and be able to jump from the paddle not moving (fit: ~150) straight to its oscillatory tracking method (fit: 1976).

Interesting notes and problems:

It seems PONG may have been too simple a game to evolve a neural network for. As can be seen in the data, the neural network quickly approaches a local optima and after investigating it seems that it is because of the way I designed the game. Because the pong paddles move at a fixed rate but the balls velocity increases with every collision, there comes a point where the ball's y-velocity is too fast to be tracked by any paddle (player or AI). The simple solution would be to refrain from incrementing the ball's velocity after collisions but then the paddle quickly evolves to become unbeatable without much effort and it didn't seem as interesting a game to evolve a player for. Perhaps if the paddles were implemented to have an acceleration as well, this problem could be fixed.