Neuroevolution

By: Trent Berrien

Project Overview

Objective: Train an AI to play tic tac toe

Restrictions:

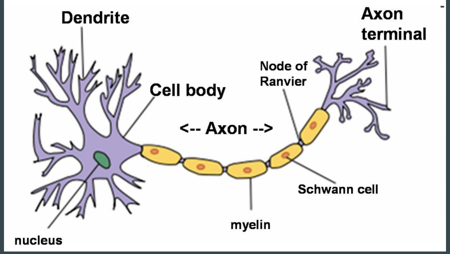
- We must never tell the AI what a good move is.
- We are only aware of the rules of the game

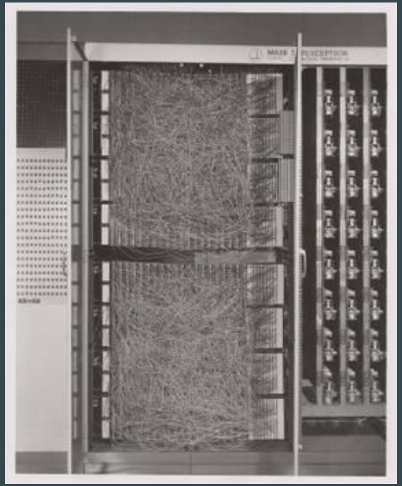
Method: Use an Artificial Neural Network trained through a Evolutionary Algorithm

Materials: Python and numpy library for running model, Google Sheets for Data Analysis

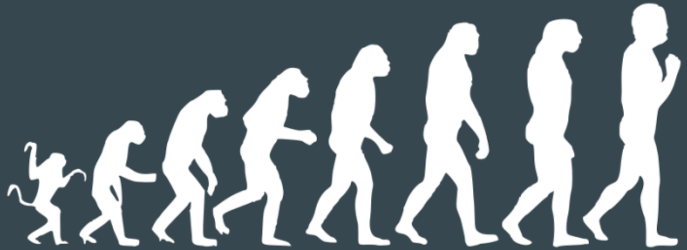
Inspiration for models used

- The Artificial Neural Network is a model which closely resembles the biological Neural Networks in our brain
- The Evolutionary Algorithm takes inspiration from the findings of Charles Darwin who theorized that species evolved through natural selection





Frank Rosenblatt’s “perceptron” (1958)



Artificial Neural Network Overview

9 input neurons, 9 first hidden layer neurons, 9 output neurons

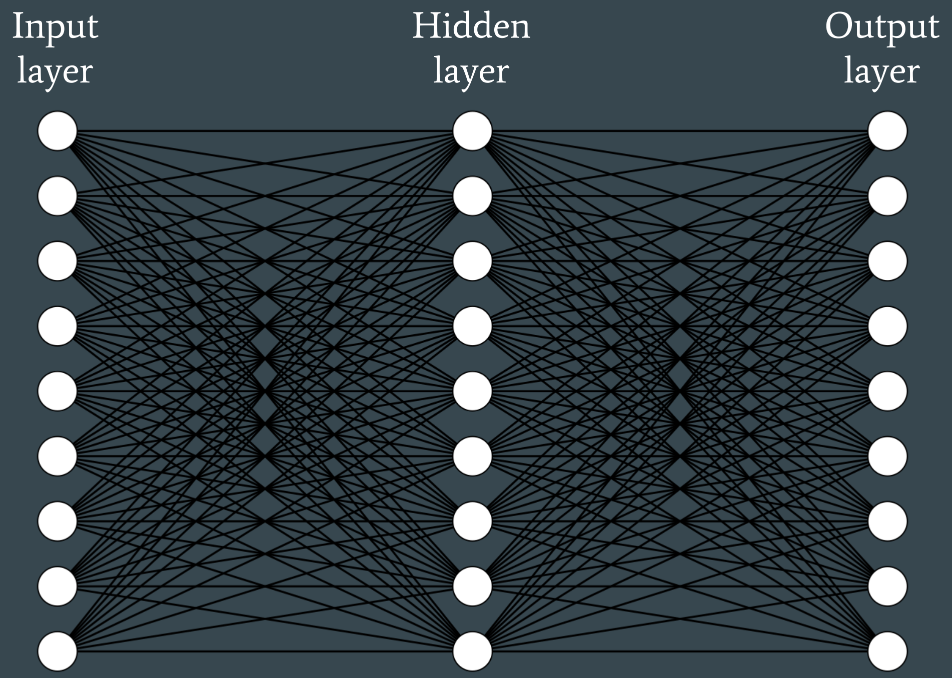
Each input and output neuron represents a space on the board

The network strictly deals with numbers

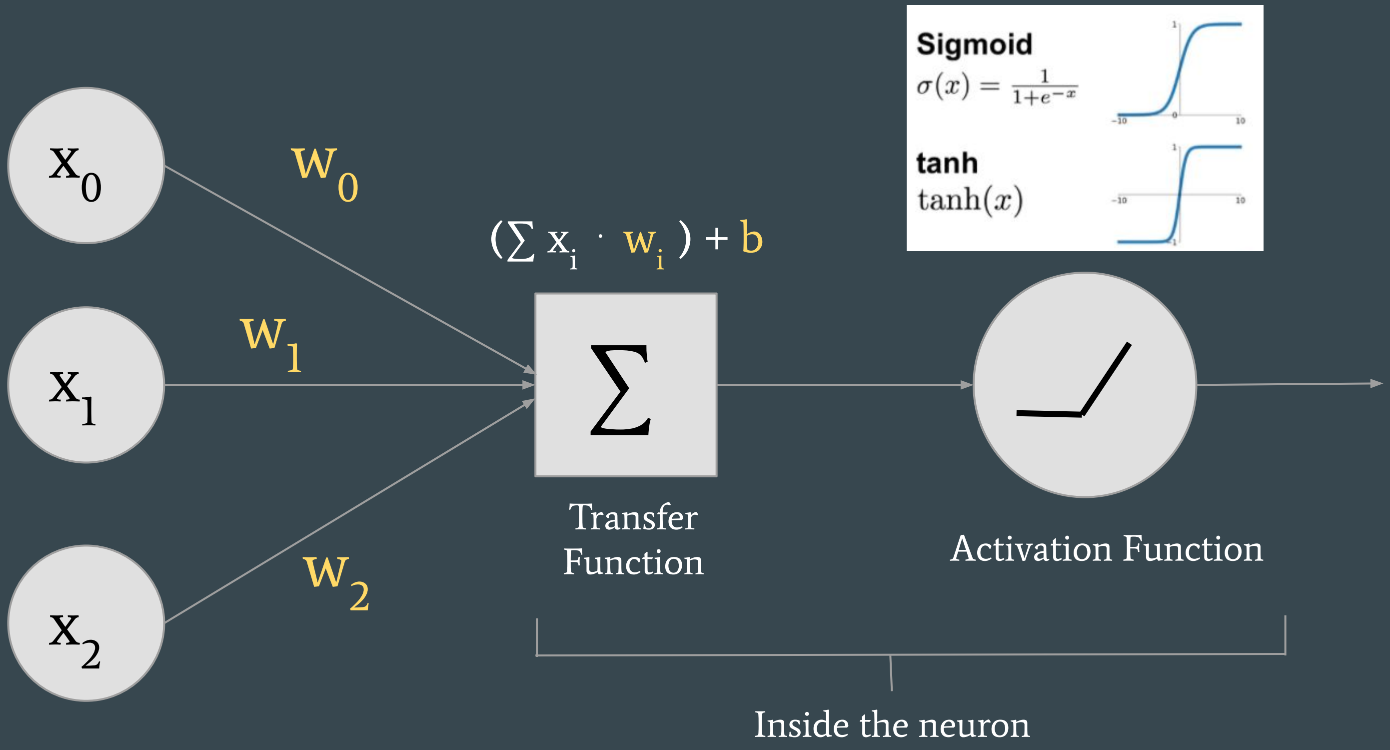
- 0 means unoccupied
- 1 is player 1's marker
- -1 is player 2's marker

The board is evaluated at a given instant by a network during its turn

The output neuron with the highest value is selected as the move the network will make



A Closer Look Into a Neural Network



Evolutionary Algorithm

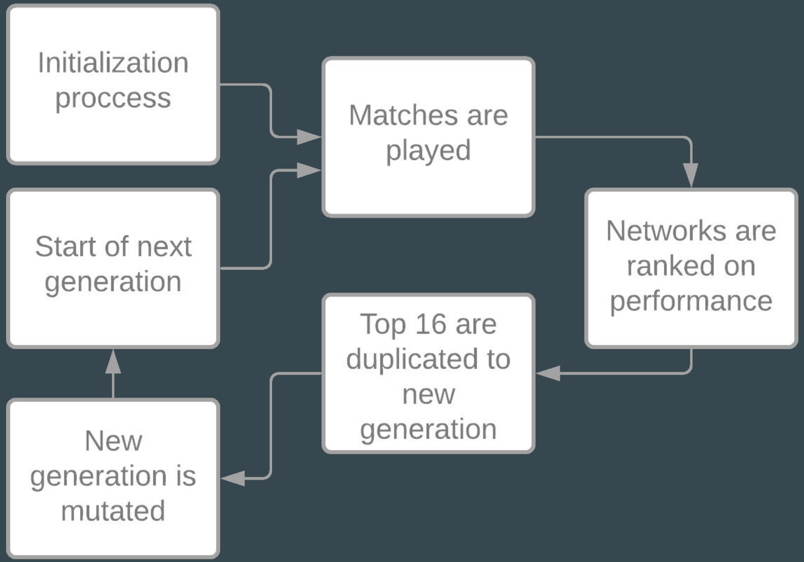
An individual is series of weights and biases corresponding to a single network (180 points of data)

Population size: 32

Intilized weights: -.5 to .5

Mutation Probability: 50%

Mutation magnitude: -.25 to .25

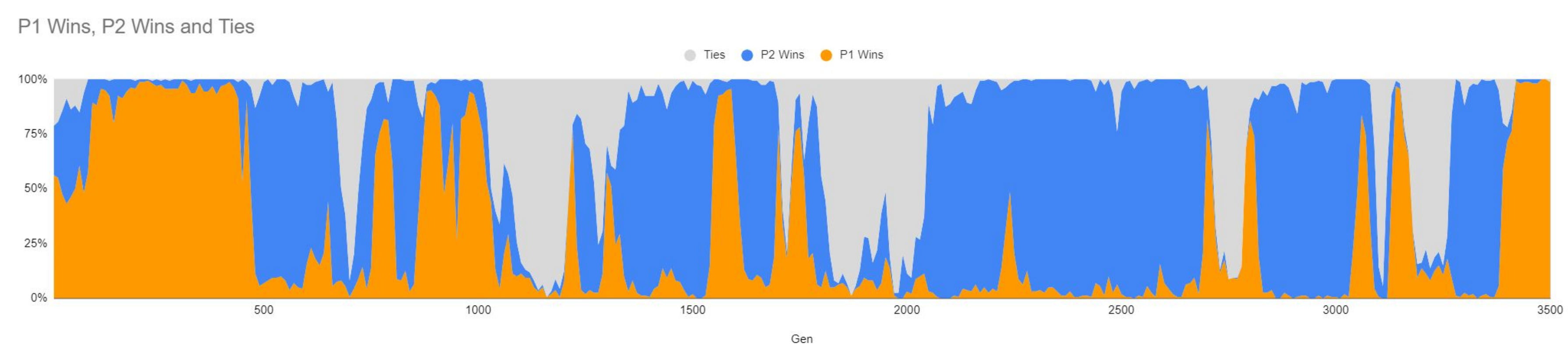


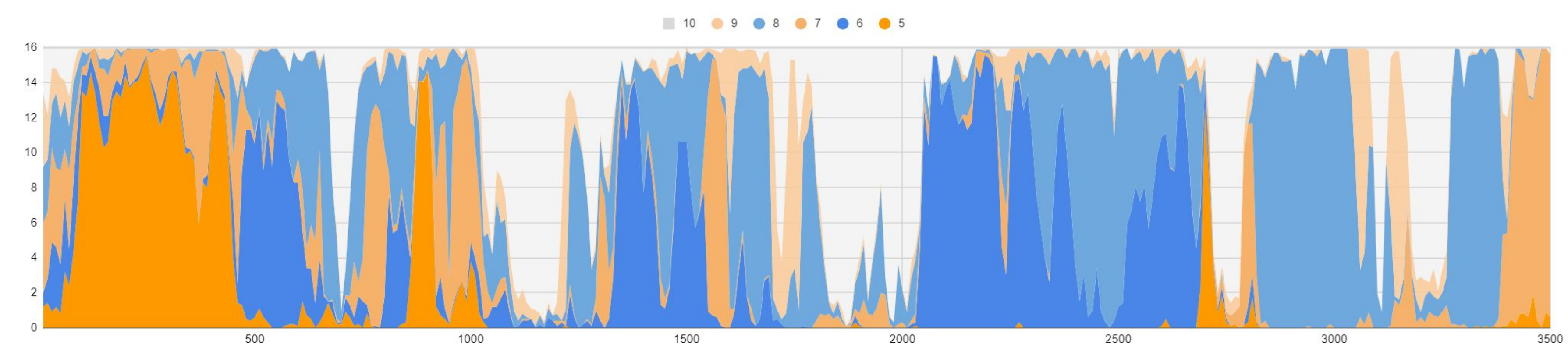
First test: Network vs Network

Process for playing matches and ranking networks:

- 32 networks are randomly paired at start of generation
- The network who starts the match is randomly chosen
- A single game is played for each pair
  - winner moves on
  - loser "dies out"
  - In the case of a tie, a winner is randomly chosen
- Winners are duplicated and mutated creating the next generation
- The next generation repeats the above steps

First test: Network vs Network Results





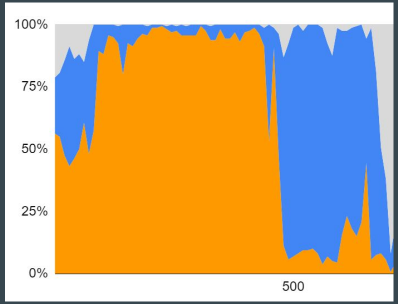
First test: Network vs Network Analysis

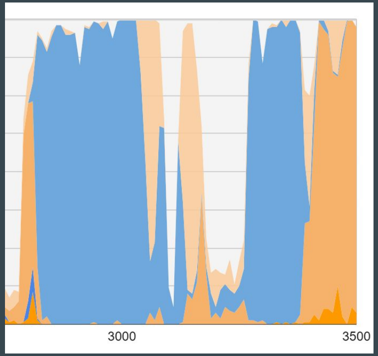
Successes:

- Strategies did develop! The moves played were not random
  - B-lining
  - Blocking
- Results showed diversity, while also showing a level of similarity

Concerns:

- No proof of objectively “better” play
- Method may not have enough stability






```
p1, p2, gen: 1 2 300
[[1. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
[[ 1.  0.  0.]
 [ 0.  0. -1.]
 [ 0.  0.  0.]]
[[ 1.  0.  1.]
 [ 0.  0. -1.]
 [ 0.  0.  0.]]
[[ 1.  0.  1.]
 [ 0.  0. -1.]
 [ 0. -1.  0.]]
[[ 1.  1.  1.]
 [ 0.  0. -1.]
 [ 0. -1.  0.]]
```

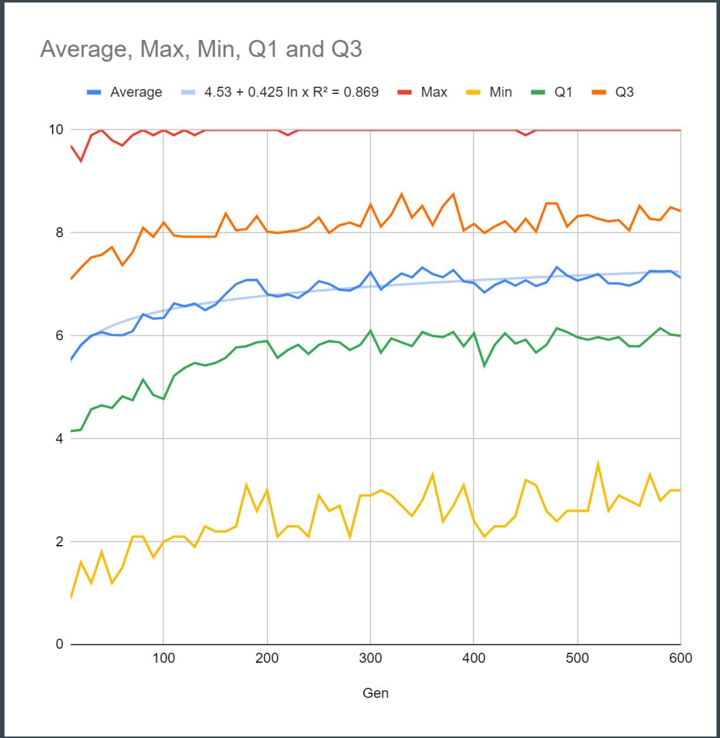
```
p1, p2, gen: 12 32 2950
[[1. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
[[ 1.  0.  0.]
 [ 0.  0.  0.]
 [ 0.  0. -1.]]
[[ 1.  0.  1.]
 [ 0.  0.  0.]
 [ 0.  0. -1.]]
[[ 1. -1.  1.]
 [ 0.  0.  0.]
 [ 0.  0. -1.]]
[[ 1. -1.  1.]
 [ 0.  0.  1.]
 [ 0.  0. -1.]]
[[ 1. -1.  1.]
 [ 0.  0.  1.]
 [ 0. -1. -1.]]
[[ 1. -1.  1.]
 [ 1.  0.  1.]
 [ 0. -1. -1.]]
[[ 1. -1.  1.]
 [ 1.  0.  1.]
 [-1. -1. -1.]]
```

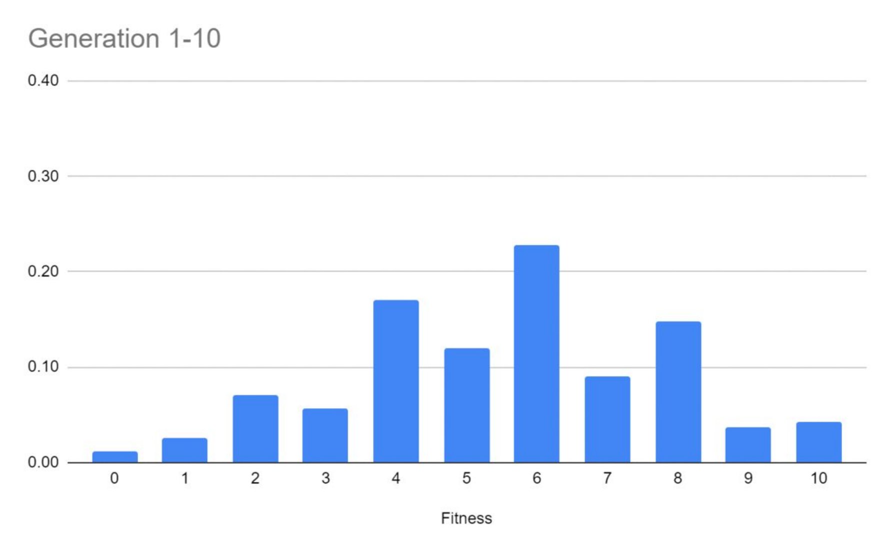
Second test: Network vs Random Moves

- Each of the 32 networks will play a series of 5 games against an opponent who makes random moves
- The player who starts the game is randomly decided for each game
- A fitness score is assigned to each network
  - Win is 2 points
  - Tie is 1 point
  - Loss in 0 points
- The networks are ranked by fitness, and the top 16 are selected to create the next generation

(This method is ten times slower per generation)

Second test: Network vs Random Moves Results





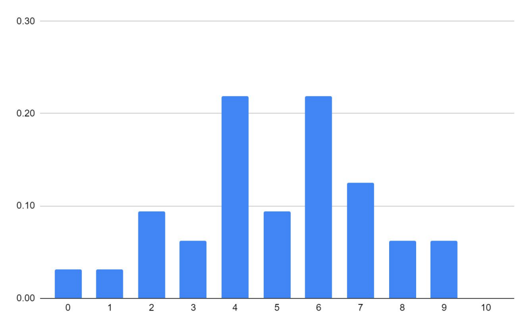
Second test: Network vs Random Moves Analysis

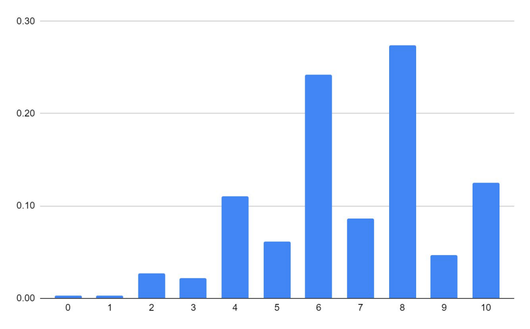
Successes:

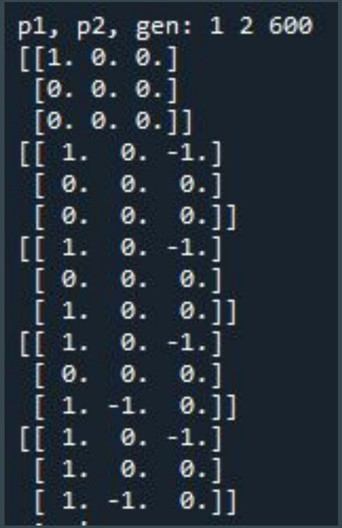
- The average did increase significantly
- Proved that the evolutionary algorithm does improve the network

Concerns:

- The strategies developed were not advanced







```
p1, p2, gen: 1 2 600
[[1. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
[[ 1.  0. -1.]
 [ 0.  0.  0.]
 [ 0.  0.  0.]]
[[ 1.  0. -1.]
 [ 0.  0.  0.]
 [ 1.  0.  0.]]
[[ 1.  0. -1.]
 [ 0.  0.  0.]
 [ 1. -1.  0.]]
[[ 1.  0. -1.]
 [ 1.  0.  0.]
 [ 1. -1.  0.]]
```

Conclusions

Successes:

- The model did develop some form of intelligence
- This method of neuroevolution has potential to solve problems we do not know the answer to

Areas to improve:

- With how many parameters there are to this model, there is plenty of opportunity for optimization
    - Adjusting mutation rates and magnitudes
    - More matches played between networks per generation
    - Implementation of NEAT algorithm, or test different typologies
- The creation of better data analytics to get better insight into the inner workings of the network would be incredibly useful

Thank you for listening. Questions?