Learning Breakout using NEAT

James T. Murphy III[1] Skyler Thomas[2]

The University of Texas at Austin

December 12, 2017

[1] jamesmurphy@math.utexas.edu
[2] skylerthomas0@gmail.com

A neat application of a neat algorthim

In 2015, YouTuber SethBling trained a neural network to successfully beat the first level of Super Mario World [2].

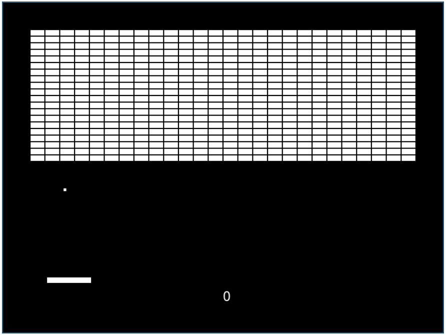
This was done using the Neural Evolution of Augmenting Technologies algorithm (NEAT) from the 2002 paper by Stanley and Mikkulainen [1].

NEAT, the algorithm

NEAT is a type of genetic algorithm that follows the biological metaheuristics of "fitness begets evolution".

1. Model neural network nodes and weights as genes.
2. Randomly mutate genomes (add/delete node, add/delete connection, change connection weight, etc.).
3. Rank individuals by fitness.
4. Allow best performing to mate using genetic crossover algorithm.
5. Repeat until fitness threshold is reached.

Breakout (AKA BrickBreaker)



We use the NEAT algorithm to train a network to play Breakout. The fitness of a genome is it's final score (out of 520) after playing a game of Breakout.

Results

Qualitatively, it appears there are three eras:

1. Pre-ball-tracking era: The network does nothing or makes small and arbitrary movements.
2. Ball-tracking era: It is apparent the paddle is tracking the ball, but it may make mistakes or infinite loop without clearing some remaining blocks.
3. Aiming era: The network actively moves to and aims the ball to clear all 520 blocks.

Results

Qualitatively, it appears there are three eras:

1. Pre-ball-tracking era: The network does nothing or makes small and arbitrary movements.
2. Ball-tracking era: It is apparent the paddle is tracking the ball, but it may make mistakes or infinite loop without clearing some remaining blocks.
3. Aiming era: The network actively moves to and aims the ball to clear all 520 blocks.

Demo time!

Results

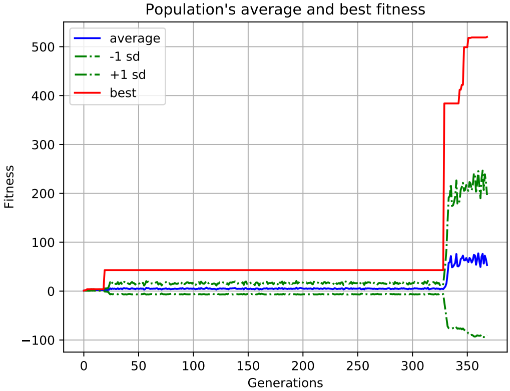
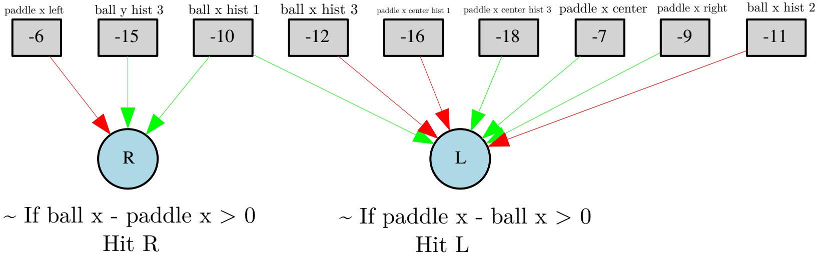


Figure: *A successful training experiment.* Generations of 200 individuals took an average time of 2.835 sec to evaluate on an Intel Core i7-4790K @ 4.00GHz in parallel on 6 cores. Total runtime ≈ 17.5 minutes.

The winning network

The most fit network is extremely simple. It has no hidden nodes and only a few connections.



This network is far too small to have memorized a sequence of inputs, it's actually tracking the ball.

Discussion

- Full board information too many inputs for success.
- Needed to give histroy of ball location for success.
- Needed to train with deterministic initial condition.
- Training prone to stagnation in a local maximum fitness.
- Success is *very* sensitive to tiny changes in hyperparameters.
- Originally attempted to do Tetris, which was too complicated for NEAT with the current setup.

Future

- Change setup:
  1. Randomly generate initial board.
  2. Make score based on 5 seconds of gameplay instead of full game.
  3. Optionally, make score weighted by the amount of time it takes to achieve it to encourage clearing blocks *quickly*.
- Compare NEAT vs. other evolutionary algorithms vs. non-evolutionary algorithms like Deep Q-learning.

[1] K. O. Stanley and R. Miikkulainen.
Evolving neural networks through augmenting topologies.
Evolutionary computation, 10(2):99–127, 2002.

[2] SethBling.
Mari/o machine learning for video games.
https://www.youtube.com/watch?v=qv6UVOQ0F44, June 2015.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller.
Playing atari with deep reinforcement learning.
arXiv preprint arXiv:1312.5602, 2013.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al.
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533, 2015.