

# Learning Breakout using NEAT

James T. Murphy III<sup>1</sup> Skyler Thomas<sup>2</sup>

The University of Texas at Austin

December 10, 2017

---

<sup>1</sup>jamesmurphy@math.utexas.edu

<sup>2</sup>skylerthomas0@gmail.com

# A neat application of a neat algorithm

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



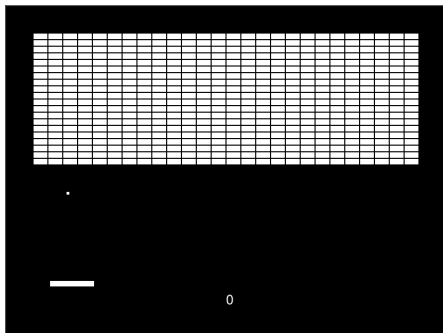
This was done using the Neural Evolution of Augmenting Technologies AlgOrithm (NEAT-O) from the 2002 paper by Stanley and Mikkulainen [1].

# 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)



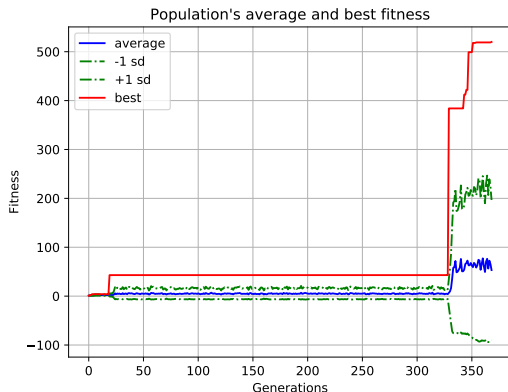
We use the NEAT algorithm to train a network to play Breakout. The fitness of a genome is it's final score (out of 520) after playing a game of Breakout.

# Results

Qualitatively, it appears there are three eras

1. Pre-ball-tracking arc: The network does nothing or makes small and arbitrary movements
2. Ball-tracking arc: It is apparent the ball is tracking the ball, but it may make mistakes or not infinite loop without clearing some remaining blocks
3. Aiming-arc: The network actively moves to and aims the ball to clear all 520 blocks.

# 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  $\approx$  17.5 minutes.

# Discussion

- ▶ It is difficult to predict what the network will and will not learn with respect to the inputs to the network and more so with respect to the hyperparameters.
- ▶ The network is incredibly sensitive to parameters and hyperparameters
- ▶ The number of generations required to successfully complete the task has a critical point.

# Future

- ▶ Explore the relationship between fitness and solution
- ▶ Compare with DQN





[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=qv6UV0Q0F44>, 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.