Flappy Bird: Neuro-Evoln. vs NEAT

Team Members: Naud Ghebre, Ethan Jen, Matthew McBrien,
Arihan Shah, Nahom Solomon
Georgia Institute of Technology

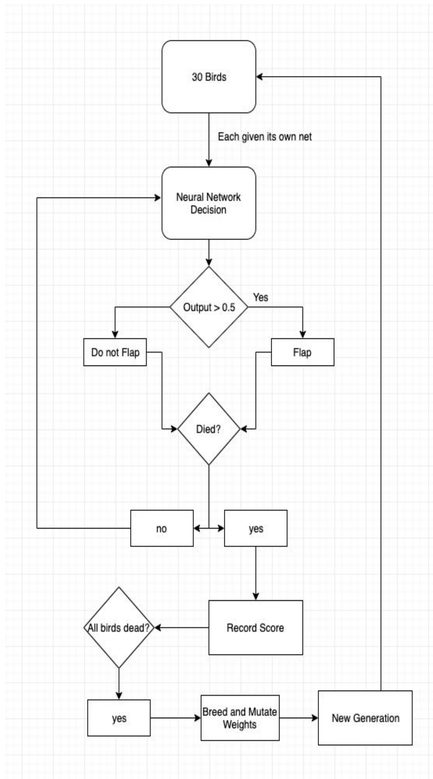
Introduction/Motivation

Video games have a predefined set of rules and a goal which make them fun and challenging. This also makes them a great playground for different reinforcement learning algorithms, machine learning algorithms which attempt to maximize some reward by training themselves as they explore the environment.

The goal of this project is to train a computer to play Flappy Bird using two different types of reinforcement learning algorithms. The types of reinforcement learning we will be using are Neuroevolution and Neuroevolution of Augmenting Topologies (NEAT). The project will compare how quickly each algorithm trains the computer to achieve certain scores.

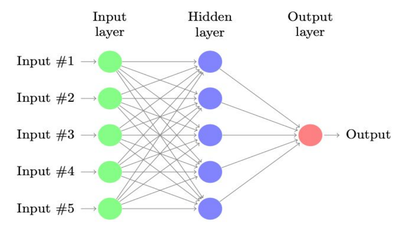
NEUROEVOLUTION

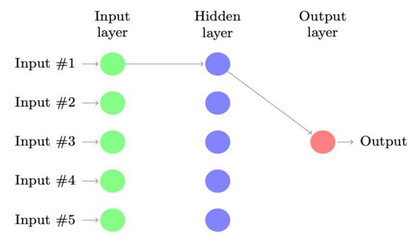
- Approach to tuning neural network weights using Genetic Algorithms.
- Human researcher creates predefined neural net topology
- Algorithm only allows for breeding and mutation of neural nets in terms of weights of connections



NEAT

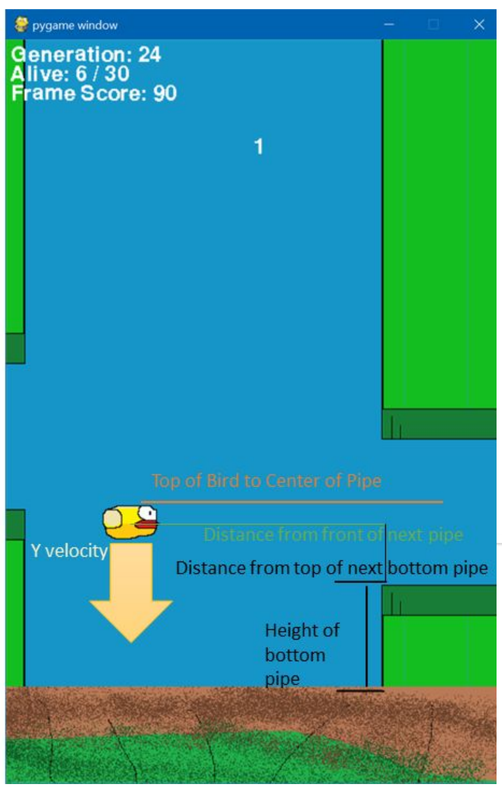
- NEAT expands upon NE by also allowing for the topology of the neural net to mutate by means of the genetic algorithm





Algorithm Parameters

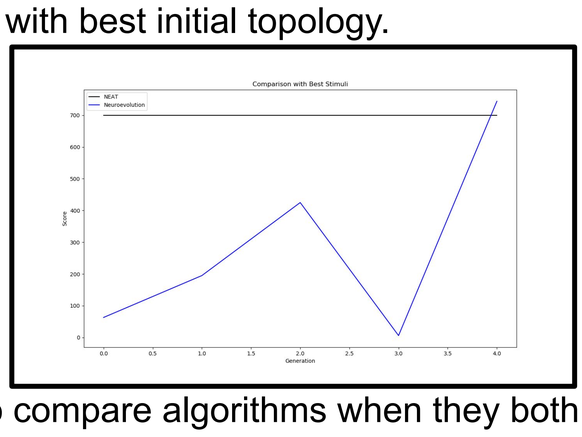
- Neural Net Inputs
  - y-velocity of bird
  - distance of top of bird to center of pipe
  - distance of bird to front of next pipe
  - bird acceleration
  - position of center of next pipe
- Neural Net Outputs
  - single output that represents whether or not to flap
- Fitness Function
  - number of frames survived
- Architecture
  - Python and Pygame



Results

Initial Topology:

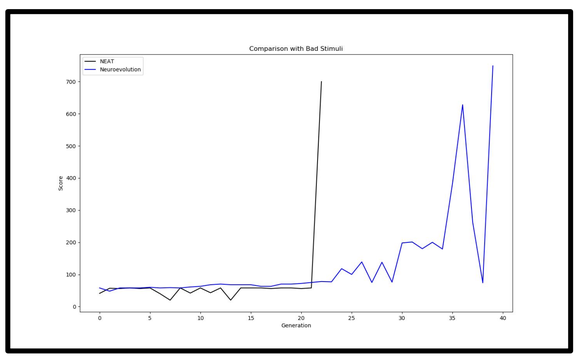
- Flappy Bird game is simple to learn with ideal neural net topology
- Algorithm: if below center: flap, else: do nothing
- Both NEAT and Neuroevolution learn algorithm very quickly with best initial topology.



- Hard to compare algorithms when they both work well

Confusing Topology:

- To better compare algorithms, give them convoluted neural network
- NEAT will be able to more quickly remove/ignore inputs and hidden nodes that aren't valuable
- Neuroevolution will have to slowly tune its weights to ignore inputs and hidden nodes



- NEAT more quickly learns flappy bird algorithm

Genetic Algorithm

1. Create a population of randomly generated members
2. Score each member of the population using a "fitness" function
3. Preserve best members of the population and randomly select members based on fitness
4. Randomly mutate genes
5. Repeat step 2 - 4 until predetermined stopping criteria (i.e. overall fitness value, number of iterations, etc)

Conclusions/Future Work

- Comparison between NEAT and neuroevolution requires complicated goal function
- Flappy Bird is not an ideal environment for comparing the algorithms
- By complicating the initial net, some conclusions can be made about the differences between the algorithms
- Comparing the algorithms against a more complicated environment (e.g. Pacman), could lead to more definite conclusions
- Time permitting, implement Deep-Q Net and compare against current findings

Member Contributions

Arihan Shah - Deep Q-Net
Naud Ghebre - Deep Q-Net
Ethan Jen - Data Preprocessing
Matthew McBrien - Neuro-Evolution, Game Design
Nahom Solomon - NEAT