GA la GA: Playing Arcade Games Using NEAT Algorithm

Genetic Algorithms
ECEN - 474/674

Tyrell Allen, Undergraduate, Senior ECE Student
Dedrian Webb,Undergraduate, Senior ECE Student
Mateo Smith, Undergraduate, Senior ECE Student
Deriech Cummings II, Graduate EE Student

Dr. Abdollah Homaifar
Professor, Department of Electrical and Computer Engineering
North Carolina A&T State University

Abstract

With an increased interest in Artificial Intelligence (AI) and Machine Learning (ML), delving into possible applications and scenarios in which these fields are applicable is commonplace. Our project uses an algorithm (Neuroevolution of Augmenting Topologies or NEAT) to investigate its ability to learn and play retro games (Galaga on the GameBoy). An emulator (PyBoy) is used to emulate the game of choice and a model is constructed through Python. The result is a model capable of playing the game on its own. This result shouldn't differ from other compatible games: the only thing that needs changing is the fitness function of the game.

Background Knowledge

Genetic Algorithms are search algorithms based on how nature evolves. It creates a set population with random features and bases its survival on how these features perform in a black-boxed fitness function. The better a species does, then the more it populates the population. On the contrary, the worse a species does, the more it dies out. The randomness of the system is what makes this work. Species mutate, crossover, and repopulate solely on how well it does and the probability for these random changes to occur.

But what if we take this same concept and use it to determine the performance of neural networks? This is the idea of the Neuroevolution of Augmenting Topologies, or NEAT for short. The main concept is the same: We take a population and test its performance on a black box. The difference is what the features are. Instead of a set of characters used to determine an individual's fitness, we use a neural network instead. The performance of the neural network, still based on a blackboxed fitness function, determines the survival of a species.

Problem Formulation

A way to show the capabilities of AI is through the use of video games. Games in themselves have a complexity that makes them fun. Knowing what to do at the right time takes quick calculation of the scenario in front of the player, translating it to button on the controller. We can use this complexity to showcase how well NEAT can overcome these challenges and perform the calculations and actions needed to play these games well.

For this project, we use two well-known retro games:

- Galaga: Galaga is an arcade shooter made by Namco and Midway you control a spaceship tasked with shooting down a squad of aliens. The aliens have a setup

phase and an attack phase. In the setup phase, the aliens will fly onto the screen where they will idle. In the attack phase, the aliens will fly down to attack you, and then return to their idle area. You are tasked with shooting down enemy aliens, moving out of the way of incoming aliens, and progressing through as many levels as possible.

- Super Mario Land: Super Mario Land is a 2D-platformer on the Game Boy. The goal of the game is to progress through the different levels by defeating enemies and gaining power-ups. The game gets harder as it progresses, with more enemies to defeat or more obstacles to go across.

Methodology

- Input Layers: 32x32 pixel game area
    - Contains information about the game area
    - Condensed for easier AI application
- Output Layer: 1*6 control scheme
    - Possible inputs: left, right, shoot
- Random Operators for NEAT Algorithm:
    - Crossover - children keep similar model qualities from their parents. Add different qualities from better parent
    - Things to mutate
        - Weight/Neuron parameter
        - Adding a hidden neuron/weight
        - Deleting a hidden neuron/weight
- Fitness Function - Accessed through the pyboy registers
    - Time - How long does it take for the model to complete a round?
    - Accuracy (Shots hit/Shots fired) - How well does the model shoot
    - Score - How many points does the model create
- Tools being used
    - Python - High Level programming language (used for ML applications and Data analysis)
    - Pyboy - Emulates game and create game area for NEAT algorithm to train on
    - Tkinter - Visualizes the game area
    - Neat-python - Python implementation of NEAT algorithm

Milestones and Deliverables

Initial Galaga Example	4/3/24
Initial Mario Example	4/22/24
Expected Completion	5/2/24

References

- **https://bgb.bircd.org/manual.html**
- *Stanley, Kenneth O, & Miikkulainen, R. (2002). Efficient evolution of neural network topologies. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600). https://doi.org/10.1109/cec.2002.1004508*
- *Goldberg, D. E. (2013). Genetic algorithms in search, optimization, and machine learning. Pearson.*
- *SethBling. (2015, June 13). Mari/O - Machine Learning for Video games. YouTube. https://www.youtube.com/watch?v=qv6UVOQ0F44*
- *PyBoy Github: Baekalfen/PyBoy: Game Boy emulator written in Python (github.com)*
- *McIntyre, A., Kallada, M., Miguel, C. G., Feher de Silva, C., & Netto, M. L. neat-python [Computer software] CodeReclaimers/neat-python: Python implementation of the NEAT neuroevolution algorithm (github.com)*

Figures

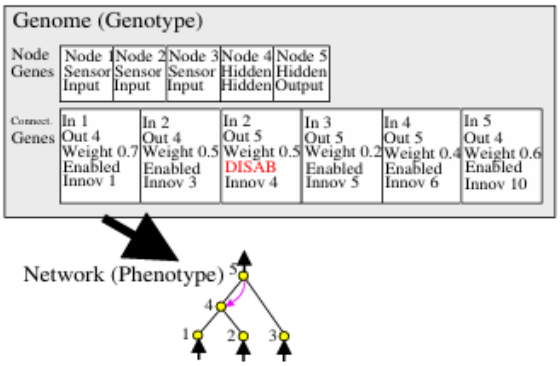

Fig. 1. Example of genome in population

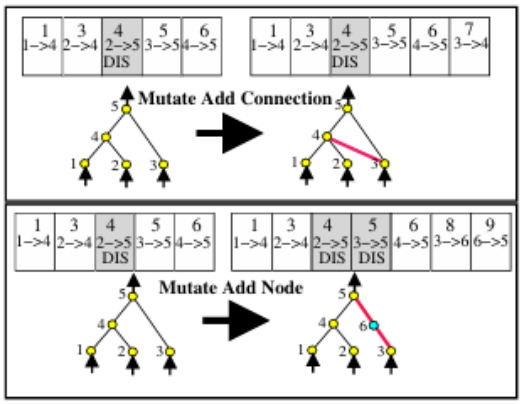

Fig. 2. Instance of mutation with a genome. Genes are added to a list of genes as they mutate

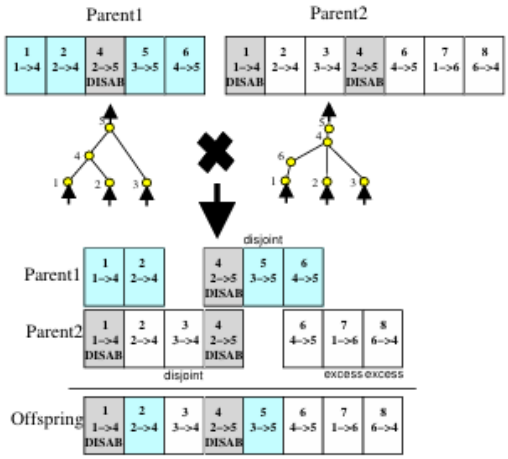

Fig. 3. Crossover between parents. Shows use of historical markings to make crossover easier