CS4795 Final Project Paper

Gavin Long

December 2024

1 Abstract

This paper explores an AI system for navigating a spacecraft from Earth's orbit to the Moon's vicinity, implemented inside a space simulation environment. The approach employs an evolutionary algorithm to train a feed=forward neural network, where multiple agents are initialized and attempt to complete the given navigation task. The training process involves evaluating agent performance, selecting the top performers, and evolving a new set of agents through random mutation. Despite achieving a rapid learning curve in within the first 50 generations and accumulating roughly 140 hours of simulated training, the AI was able to achieve a 51% success rate in completing the task due to limitations in the scoring system and sensitivity of spaceflight to trajectory changes.

The finings highlight the need for improvements in both network complexity and the scoring mechanism to improve performance. Future work focuses on integrating more advanced training techniques, refining the scoring criteria for better alignment with desired behaviors, and implementing an adaptive learning rate.

2 Introduction

The core idea of this project was to create an AI that could control the trajectory of a spaceship from Earth's orbit to the Moon's proximity. My reason for choosing this subject is that I find space interesting and space navigation provides a well balanced scenario, where movement is complex but inputs are relatively straightforward. this project is built on top of a previously developed space simulation that i have made. this simulation is used as the foundation for the environment that the AI will train in.

3 Problem Statement

There is a spaceship orbiting Earth that needs to get to the Moon. The AI must perform the necessary actions to navigate the spacecraft to the proximity of the Moon, accounting for gravitational influences and orbital mechanics involved in space travel.

4 Methodology

4.1 Network design

The AI implementation employs an evolutionary approach. initially, 10 different agents attempt to complete the task. The three that perform the best are then selected and evolved into 10 new agents. this cycle repeats until an agent is able to achieve a satisfactory result.

The neural network of the AI uses a basic feed-forward neural network design based on the architecture described in "Neural Network Design" by Hagan et al [1]. Each layer of the network is made up of some number of neurons, where the output of each neuron can be described as

$$a = f(Wp + b)$$

The input p is a matrix of the outputs of the previous layer of neurons, W is a matrix of the the weight of each input and b is the bias of this neuron. All of that is then run through the activation function *fx* to receive the output of this neuron. The activation function *f(x)* is expressed as

$$f(x) = x/(1 + x^2)$$

The activation function ensures the output will be a value between minus one and one.

4.2 Training Process

Each agent is initialized as a network of a predefined size and will maintain that size throughout training. The training process can be broken into 6 basic steps

1. Initialize 10 randomized networks
2. All 10 agents attempt the navigation task and are scored based on performance
3. The three best performers are selected as the winners
4. Winners are randomly selected to create 10 new agents

5. Weights of each of the new agents are randomly adjusted for variation

6. Steps 2. through 5. are repeated until acceptable fitness is achieved

4.3 Scoring System

Agents are scored based on their ability to follow a predefined course, ability to reach the Moon and a small bonus for pointing towards the Moon and approaching the Moon. Agents will also receive penalties for straying too far from the path, crashing, increasing their distance to the moon and also a small penalty for the longer they take. Straying off the path, crashing or getting further from the moon are considered blocking penalties, so if an agent receives that penalty once then they will not be able to receive points from performing positive actions in that category till the end of the round. For example, if an agent was to stray too far from the path, they would not be able to return to the path to receive points this round.

5 Results

5.1 Successes

During training the agents was able to demonstrate a rapid initial learning curve, showing the ability to perform generally desirable actions in less than 50 generations. Additionally, implementing the neural network in C# resulted in a reasonably high performance system, capable of running at 15 to 20 times real-time speed despite inefficiencies in the code implementation. This performance allowed the agent to accrue approximately 140 hours of training time over the course of less than a day.

5.2 Challenges

Bonuses for pointing towards the moon and penalties for time taken seem to have had little effect on the training performance and could have possibly decreased performance to a degree. The scoring system is also quite lacking as after approximately 140 hours of training, the AI only achieved a 51% success rate of reaching the Moon. The extreme sensitivity of trajectories to small changes in movement was likely a major factor in the poor success rate, but the largest deficiencies are likely related to the scoring system being unable to effectively reward the desired behavior.

6 Conclusion

6.1 Future Work

The current implementation was most limited by the simplicity of the network design and the unrefined scoring system. Potential ways these could be improved include:

1. Incorporating back-propagation or other training methods to improve learning efficiency.
2. Improving the complexity of the network architecture and allowing network size and layout to change during training.
3. More accurately defining desired behavior and modifying the scoring system to be more effective.
4. Providing the AI with some active feedback on the score its receiving and how well its performing.
5. Implementing a system to adjust the learning and mutation rates during training to balance exploring new strategies and refining existing ones.

7 Github

https://github.com/LongGavin/CS4795-Spaceship-Game-AI.git

References

[1] M. T. Hagan, H. B. Demuth, Mark Hudson Beale, Orlando De Jesús, and E. Al, Neural network design. S. L.: S. N, 2016.