Arti-Eco

Can Neural Networks and Machine Learning improve an artificial ecosystem?

Finian J. Graham

Computer Games Development

University of Central Lancashire UCLan

What Is The Problem?

Looking at games such as “Elite Dangerous” or “No Man’s Sky”, one of the major gameplay mechanics is exploration and adventure. For the most part, this is implemented well in both games, with a multitude of different environments to discover and living organisms to investigate, but there is only so much a single development studio can do. After a while, senior players will start to notice re-used entities and behaviours across multiple planets in ways that don’t quite make sense. Why would there we two creatures, light years apart, with exactly the same behaviours and design?

This project implements reinforcement learning and neural networks as a potential solution to the issue of duplicated species and exploration stagnation.

Program Overview

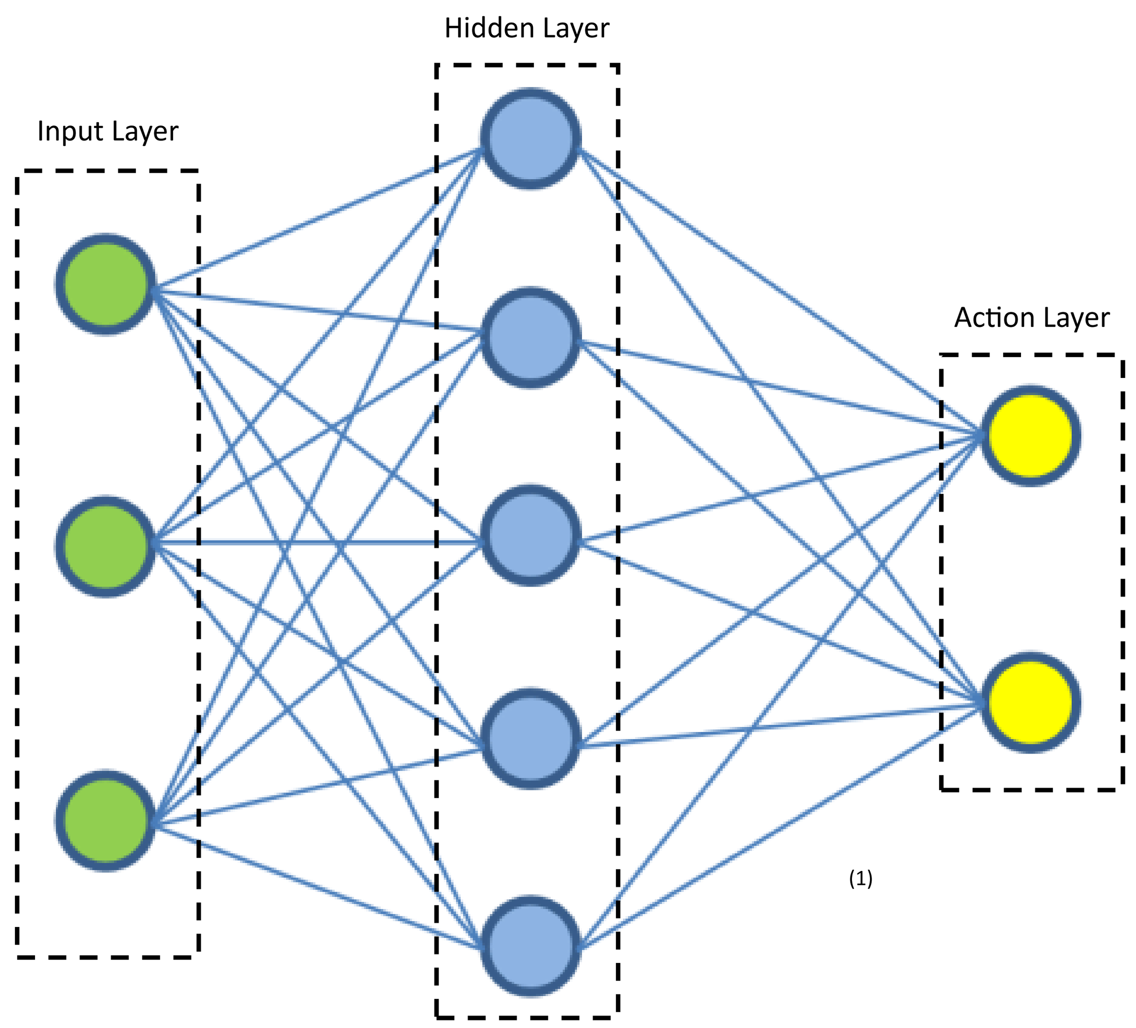
The technical aspects of this program can be split into two sections; the Network Algorithms and the Testing Application.

The application is designed to allow for easy testing of learning algorithm implementation, with a neural network architecture based entirely in C++ to allow for full flexibility and ease of implementation into other C++ based development projects. The default survival simulation is a cellular automata in which Agents, driven by neural networks, must consume nutrients in order to survive and reproduce.

The network architecture implemented is a standard Perceptron architecture in which Sensor Neurons take data from the World State, and then Action Neuron values are determined by forward propagation.

$$\nabla_\theta J(\theta) \underbrace{\approx \hat{g}}_{\text{Estimation of the gradient (given we use only one trajectory to estimate the gradient)}} = \sum_{t=0} \underbrace{\nabla_\theta \log \underbrace{\pi_\theta(a_t|s_t)}_{\text{Probability of the agent to select action at from state st given our policy}}}_{\text{Direction of the steepest increase of the (log) probability of selecting action at from state st}} \underbrace{R(\tau)}_{\text{Cumulative return}}$$

(2)



Network Components

Neuron: Neurons act as the brains of the operation. They take data in, apply any computations to the values and then pass the data along to the next layer. In this model, all neurons apart from the sensor neurons calculate the weighted sum of all input weights, skewed with a random bias.

Link: Links transfer data from one layer to the next. Each link only connects to a neuron in the next layer along, and they contain a weight that will be adjusted over time to improve the effectiveness of the network overall.

Layer: Layers simply organise the neurons into an order that the network can work with. Information always travels from left to right, in order to prevent infinite loops.

So How Does It Learn?

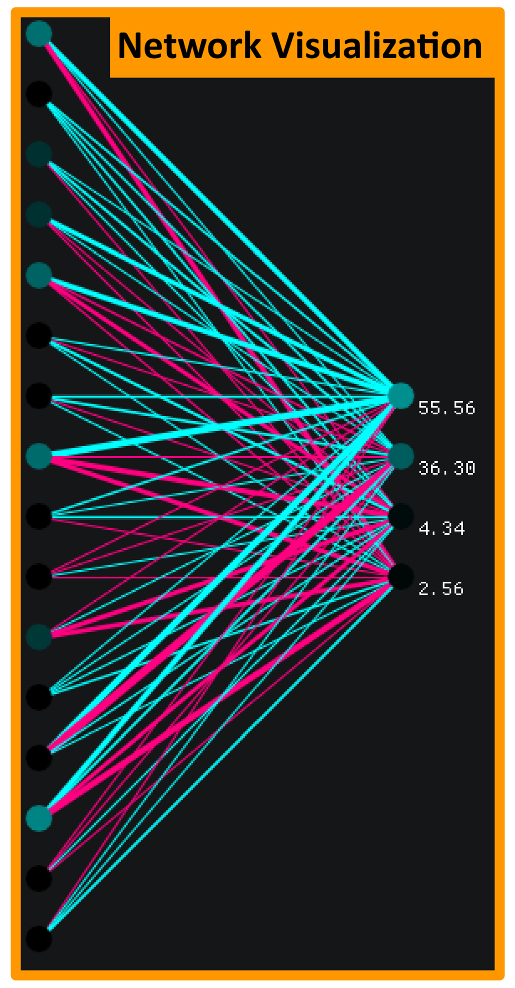
The network “learns” by a method called back-propagation. Once an action is completed, the agent determines how good that action was and then moves back through the network, increasing and decreasing the weights in order to increase the likelihood of that given action triggering during that state. The biggest challenge is that we can’t check every single action to see which would be best without a significant impact to performance, so the equation on the left represents a method called **REINFORCE** (2), which aims to estimate the best changes to the network given a limited sample size.

The Network & The World

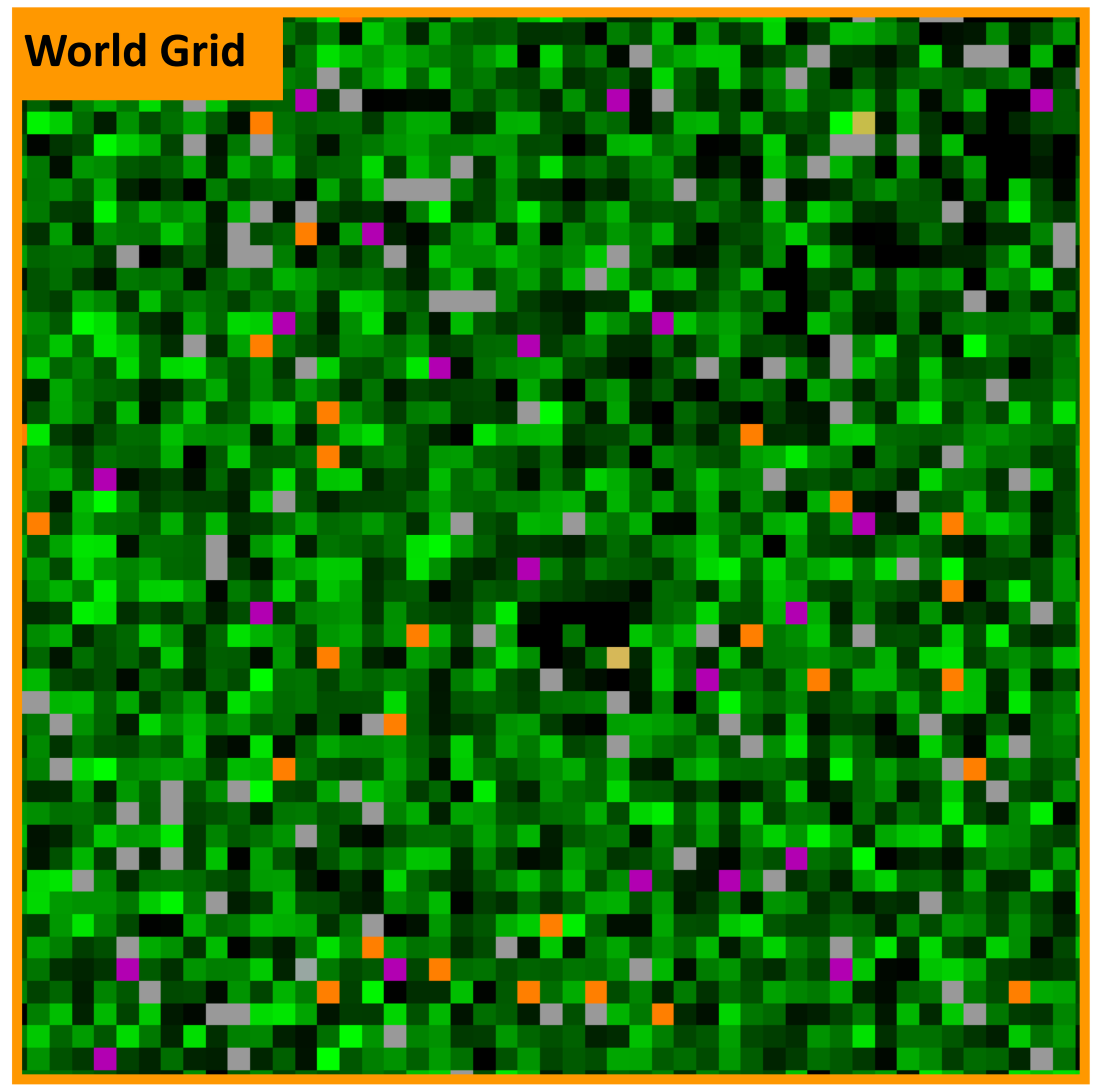
The program displays the network as follows: Cyan circles represent Neurons, with the brightness of the neuron representing neuron activation. The thickness of the links represent the link’s weight, with cyan links being positive weights and purple links being negative weights.

The world is represented by a simple grid, with each cell’s colour showing the cell’s status. Purple and Orange cells represent hazards that agents must navigate.

Network Visualization



World Grid



Ecosystem Health & Data Collection

The overall health and success of a test ecosystem can be derived from three variable: Unique Agent Species, Average Efficiency and Number of Successful Agents. A ‘Successful Agent’ is determined as an agent with an efficiency value above 1.0, meaning that the agent is gaining more energy than it is losing.

The diversity of an ecosystem is important for exploration, and this can be measured by observing the number of unique species. The more unique species, the more unique behaviours that can be observed.

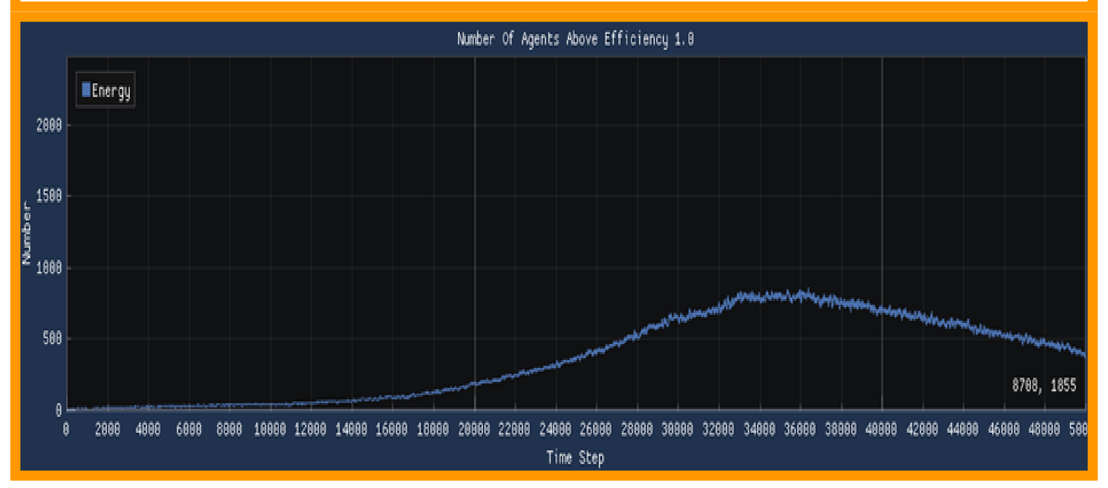
The higher the average efficiency, the more stable the current ecosystem is. If the average efficiency is low, there is a high chance of species dying out.

Conclusion

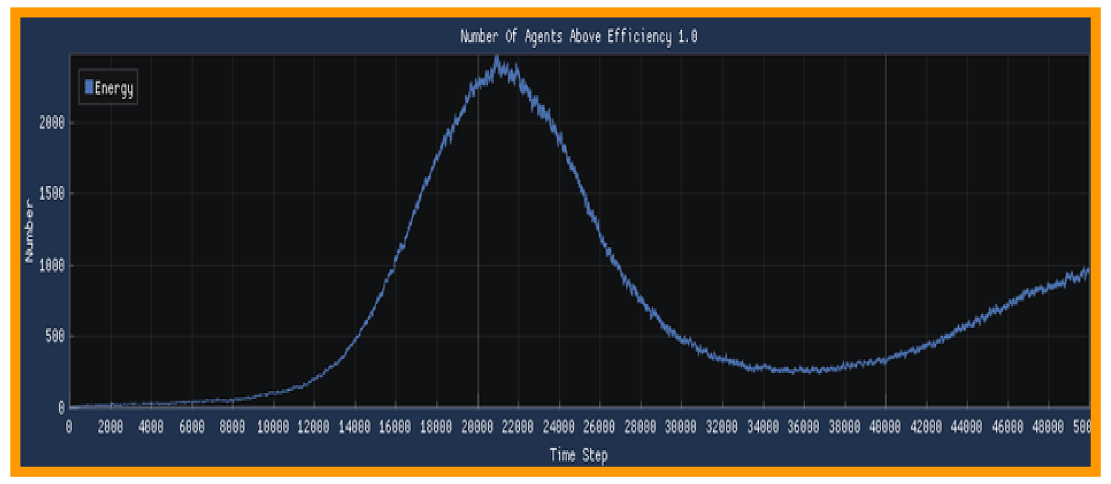
Overall, this approach to evolving and improving an ecosystem based on neural networks and survival of the fittest shows a lot of promise. The success of the ecosystem improved significantly, with the first peak of successful population occurring much earlier than in systems where agents do not learn.

One thing that could be investigated further is how to keep the system diverse, as with the learning algorithm it is very easy for one or two species to dominate and take all of the resources for themselves, leaving none for other species.

Below are graphs used to display the number of successful agents present over a time frame of 50,000 updates.



Successful agents over time with no learning algorithm.



Successful agents over time with proposed REINFORCE algorithm implemented.

References

1 Danilov, Mikhail & Karpov, Arkadi. (2018). A classification of meteor radio echoes based on artificial neural network. Open Astronomy. 27. 318-325. 10.1515/astro-2018-0037.

2 Hugging Face. (2024, April 04). The Problem of Variance in Reinforce. Retrieved from Hugging Face: https://huggingface.co/learn/deep-rl-course/en/unit6/variance-problem