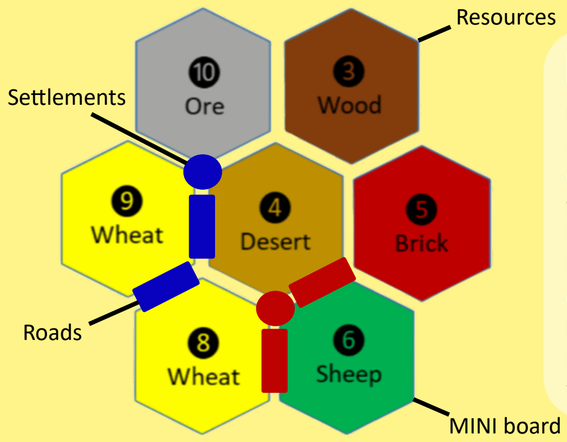
AI CATAN AGENT

WITS UNIVERSITY

Liam Brady | Supervised by Prof. Clint van Alten

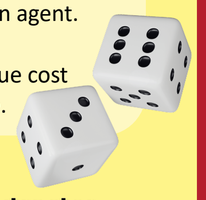
Introduction

This research report introduces a Reinforcement Learning (RL) agent which uses a combined feature extractor - Convolutional Neural Network (CNN) and Multi-Layer Perceptron (MLP) - paired with a Maskable Proximal Policy Optimisation (PPO) [1] algorithm to improve upon the PPO agent available in the Catanatron [2] environment. This improvement also demonstrates the effectiveness of dense reward shaping on agents exploring large 4-player state-action spaces and determines which feature type, spatial or non-spatial, contributes more towards the winning capability of a Catan playing agent.



Background

Network selection determines how the information of the board will be processed by an RL algorithm, and so finding the right network to represent each state of the board is crucial. A CNN was chosen because of its ability to find patterns within image data [3], and an MLP for the tabular non-spatial board features. RL trains an agent to select optimal actions in an environment by maximising cumulative rewards. This makes it suitable for teaching agents in complex games such as Catan which have various ways of rewarding an agent. For this reason, the PPO algorithm was chosen for training. Deep RL is particularly useful in large state spaces because it approximates the true cost values of different actions [4], ensuring the problem remains tractable.



Research Questions

- Can the Proximal Policy Optimisation agent in the Catanatron environment be improved with a different feature extractor?
- Can the proposed model convincingly beat three Catanatron Weighted Random bot opponents with a win rate higher than 25%?
- Can the proposed model surpass the performance of the original PPO agent in the Catanatron environment?
- Does dense reward shaping improve the performance of an agent in a large state-action space?
- Can it be determined which feature type has a greater contribution to whether the agent is more likely to win a game?

Methodology

The purpose of the mixed agent is to separate the features of the board into different feature extractors which are better suited for their respective data types. That is, a CNN would preserve the spatial data of the 3D tensors of the board, whereas an MLP would be much more efficient with non-spatial information. Making use of the mixed observations of the board, the agent separates feature extraction using a CNN for the 3D board tensors with dimensions (C=20, W=21, H=11) representing the physical features, and an MLP which accepts an observation shape of (76,) (1D vector) for the non-physical features (Figure 1).

Once the features have been processed by their respective networks, they are then concatenated and fed into the action and value networks. Unlike the vector agent, each head does not have its own extractor network, but rather one shared backbone network. The reason for this is to save computing resources. The second change made to the agent is the reward function used when training in the environment. It can be difficult for an RL agent to make successful updates to its network when it doesn't know what moves led it to success. The dense reward function aims to mend this by providing rewards for actions that steered the agent into a winning state. For training, both the sparse and dense reward functions were used.

The Numeric and Board only agents use a split version of the mixed agent network architecture - the Numeric only agent uses the MLP feature extractor, whereas the Board only agent uses the CNN feature extractor. Similarly to the mixed agent, they share one backbone extractor network. The purpose of these agents is to determine which feature makes a greater impact on the performance of a Catan playing agent and to find out whether it relies more on the spatial or non-spatial features.

Maskable PPO with COMBINED Feature Extractor

(Dual-path: CNN for board + MLP for numeric features)

Figure 1: Visual representation of the mixed agent.

Results

The mixed dense agent had the highest performance overall as well as the lowest variance (Figure 2, bottom left), surpassing the performance of the original vector sparse agent by a total of 6.32%, and substantially beating the aim of a 25% winrate against 3 Weighted Random bots (Figure 3). This is a significant improvement in performance in a stochastic 4 player environment and shows the benefits of adding both the change to the feature extractor and to the reward function when compared to the original model.

The board dense agent beat the numeric dense agent (better performing of the two numeric agents) by 7.07%, with the board sparse beating it by 8.05% (Figure 4). Clearly, the board features had a much bigger influence on the final result of the agent's performance. This could be due to the substantial disparity in number of features (20 x 21 x 11 = 4620 compared to 72), and so the numeric agent has far less information to utilise when deciding which actions are more favourable.

The vector sparse and mixed dense models had a major difference in time taken to train at 1 000 000 timesteps, with the mixed model taking 44.69 minutes longer to train than the original vector sparse model (Table 1).

Figure 3: Figure comparing all mixed and vector agents with sparse and dense rewards.

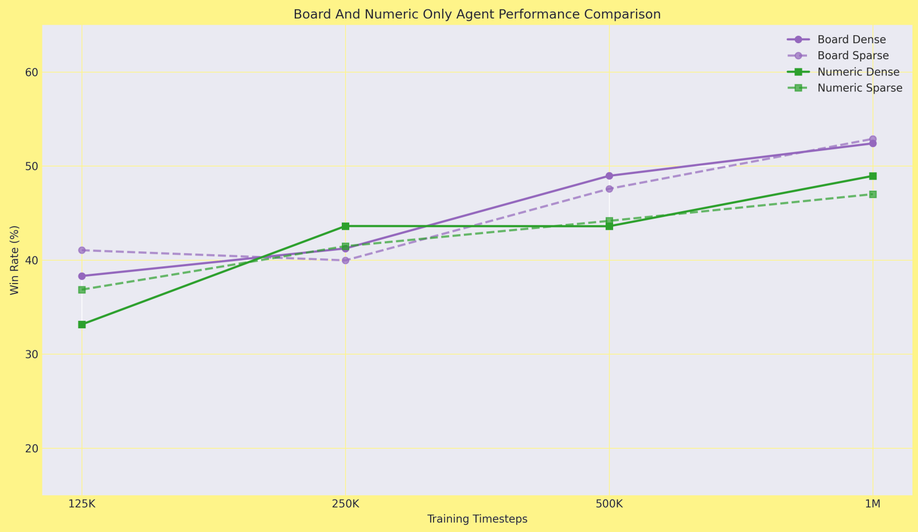


Figure 4: Full comparison of numeric and board models.

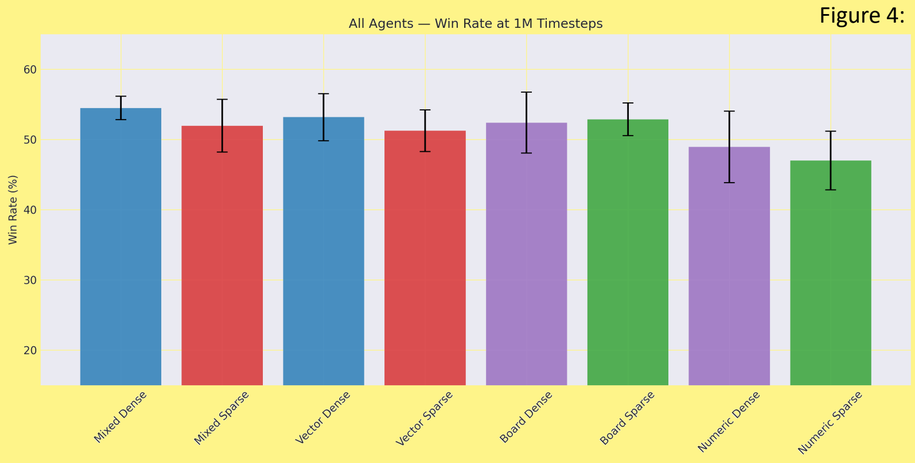


Figure 2: Bar graph of all agents with standard deviation lines.

Agent	Avg. Win Rate (1M)	Avg. Train Time (min) (1M)
Mixed Dense	54.50% ± 1.65	123.91 ± 3.28
Mixed Sparse	51.96% ± 3.75	126.34 ± 4.79
Vector Dense	53.19% ± 3.36	77.13 ± 2.26
Vector Sparse	51.26% ± 2.96	79.22 ± 5.88
Board Dense	52.41% ± 4.33	117.36 ± 5.42
Board Sparse	52.89% ± 2.31	112.51 ± 6.03
Numeric Dense	48.95% ± 5.09	94.05 ± 1.18
Numeric Sparse	47.01% ± 4.16	87.30 ± 1.24

Table 1: Average win rates and training times at 1M timesteps with standard deviation.

Conclusion

Catan's state-action space is complex, making it difficult for an RL agent to learn favourable actions effectively. The choice of feature extractor to better suit the feature data type can make a positive difference to the overall performance of a reinforcement learning agent. A dense reward function also provides an increased performance, but when combined, these two changes can make a larger improvement than what they could separately. The mixed dense agent beat the original vector sparse agent by 6.32%, showing a significant change in performance. The board sparse agent had a performance improvement of 8.05% over the better of the numeric only models, demonstrating the importance of spatial features when training a Catan-playing agent.

References

[1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[2] Bryan Collazo and Contributors. Catanatron. https://github.com/ bcollazo/catanatron, 2021.

[3] Brahim Driss and Tristan Cazenave. Deep catan. In International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pages 503–513. Springer, 2022.

[4] Konstantia Xenou, Georgios Chalkiadakis, and Stergos Afantenos. Deep reinforcement learning in strategic board game environments. In European Conference on Multi-Agent Systems, pages 233–248. Springer, 2018.

100 1922 2022