Training an RL agent to play Starship Meteor Game

Alok Malik

December 2020

1 Introduction

In this task we're going train an RL agent to play the game of starship meteor where a starship has to avoid vertically falling meteors. The RL algorithm we're using in this task is Actor-Critic with eligibility traces described in Reinforcement Learning book [1] by Sutton and Barto.

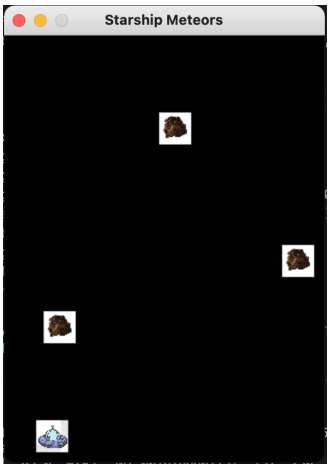


Figure 1: Screenshot of game in in play

Figure 1 shows the game in play. The spaceship is in the lower left corner of screen and there are three asteroids falling vertically from top. Spaceship can only move left or right to avoid asteroids.

2 Task

The task of the RL agent is to survive for 1000 timesteps without hitting any of the falling asteroids. If the starship is hit by an asteroid the game is over. The game will always have 3 fixed number of asteroids at a time on screen. The asteroids fall vertically in straight lines with a randomly chosen speed from 0 to 5 coordinates per time step. The action space of

space ship consists of 3 different actions on the x axis: move left, stay there or move right. The feature space consists of two features from each asteroid, X length which is the distance on x axis between asteroid and spaceship and the Euclidean distance between spaceship and asteroid. These two features are provided for each asteroid to the RL agent.

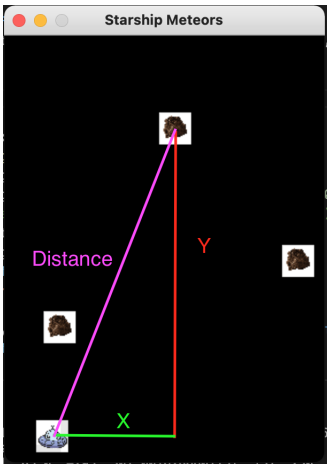


Figure 2: The input features of algorithm are X and Distance

The reward function is a hyperbolic function which takes distance as input. The reward is very small until the asteroid comes within a certain distance of the spaceship, after which it increases exponentially. The reward function is below:

$$reward = -e^{10\times(100-distance)/500}$$

We sum the reward for all the meteors present on screen. With this reward function we've tried to emulate fight and flight response [2] of human beings which activates sudden flow of adrenaline when human's face a life-threatening situation. Similarly it receives a sudden huge amount of negative reward when asteroids reach with certain distance of spaceship.

The approach to model this task uses actor critic architecture for RL [3]. Specifically Actor-Critic with eligibility traces [4] algorithm with Fourier basis [5] function of order 3.

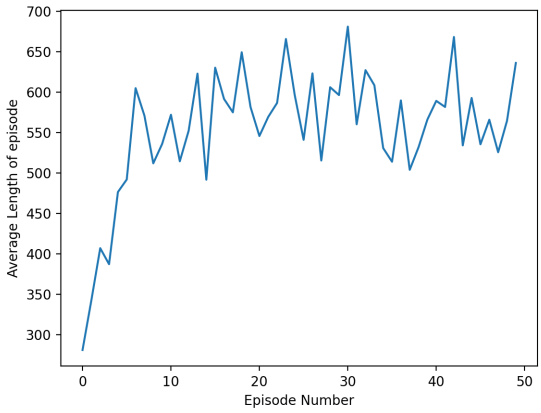
The parameters used for training task as below:

$$\lambda^\theta = .7, \lambda^w = .7, \alpha^\theta = .007, \alpha^w = .007$$

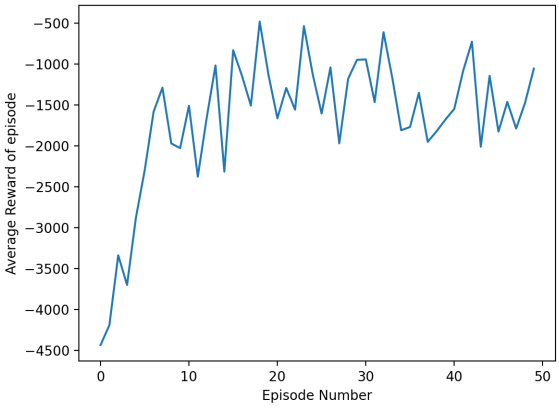
The task is treated as a non-discounted task with $\gamma = 1$.

3 Results

The task clearly shows that RL agent is learning to avoid obstacles. In the Fig. 3a below, the average length of episode from first episode is around 400 and increases to 700 near the end. At the length of 1000 the agent wins the game. Similarly in Fig. 3b we can see reward is increasing as the number of episodes go on starting from negative 3000 reward to positive 1000.



(a) Average length of episode vs Episode number



(b) Average reward of episode vs Episode number

Figure 3: Results above are averaged over 50 runs

4 Conclusion and Limitations

We can see from the results in above section that our RL agent is certainly able to learn the task and show learning behavior by avoiding asteroids. But the asteroid gets stuck in places where reward is locally maximum, that's one of the major limitations which we can try to overcome by including potential reward shaping or encouraging more exploration in the algorithm. Also we could use different reward function, as the current reward function is a exponential function Fourier basis function might have difficulty estimating rewards from exponential function accurately.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] R. McCarty, "The fight-or-flight response: A cornerstone of stress research," in *Stress: Concepts, cognition, emotion, and behavior*. Elsevier, 2016, pp. 33–37.

[3] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 834–846, 1983.

[4] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.

[5] G. Konidaris, "Value function approximation in reinforcement learning using the fourier basis," *Computer Science Department Faculty Publication Series*, p. 101, 2008.