CentraleSupélec Reinforcement Learning Individual Assignment Report

Sebastian Lee[1]

[1] CentraleSupélec, Gif-sur-Yvette, 91190, France
sebastian.lee@student-cs.fr

Abstract. The report summarizes the implementation of two reinforcement learning methods in the Text Flappy Bird Gym. The source code of the implementation can be found at: https://github.com/seblee99/RL_Individual, under RL_Individual_Code.ipynb.

1 Background

The goal of this individual assignment was to implement two reinforcement learning algorithms, one Monte Carlo based algorithm and Sarsa algorithm. Two versions of Text Flappy Bird Gym were provided in class. They were 'TextFlappyBird-v0' and TextFlappyBird-screen-v0. The basic mechanics of the game were equal, other than the observation output. The major difference between the screen version and the non-screen version was that the screen version outputted text array, which was in the text format of Flappy Bird gameplay. The non-screen version outputted a horizontal distance and a vertical distance between the player and the gap between pipes. The individual assignment utilized the non-screen version as it facilitated the implementation of the reinforcement algorithms and the observation output of the environment proved to be beneficial, especially when initializing the tables for state-value functions.

2 Implementation of Reinforcement Learning Methods

2.1 Summary

The two well-known reinforcement learning algorithms, Monte Carlo Control and Sarsa, were implemented. The two algorithms were trained on the Flappy Bird environment with the height of 15 and the width of 20 with a gap of 4. Each agent was trained for 50000 episodes, with alpha of 0.1, gamma of 1.0, and epsilon of 0.1. During the training, the epsilon value is decayed by multiplying 0.9999999. Other higher values resulted in lack of exploration, resulting in no learning. After the training, the Q-tables of both agents were visualized to represent the state-value functions, and the scores per episode with moving average of 1000 episodes were plotted to determine the performance. Then, both scores per episode results were compared. Lastly, the agents

were placed in a different Flappy Bird environment configuration with the height of 10 and width of 30 with a gap of 2.

2.2 Monte Carlo Control (MCC)

According to the Section 5.3 of Introduction to Reinforcement Learning by Sutton and Barto, Monte Carlo Exploring Starts algorithm was implemented. The value of the state, which was given by the game environment in the format of "(dy, dx)", had to be converted into one dimension integer to accommodate the Q-table.

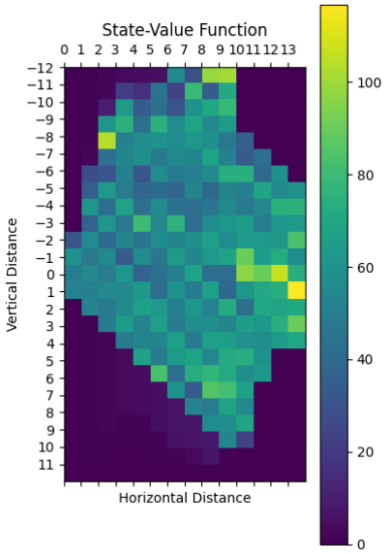
2.3 Sarsa (Originally Written Prior to Sarsa(λ))

Sarsa was implemented according to the Section 6.4 of Introduction to Reinforcement Learning by Sutton and Barto. Similar to Monte Carlo Control, the value of the state, which was given by the game environment in the format of "(dy, dx)", had to be converted into one dimension integer to accommodate the Q-table. Due to an error in the individual assignment instruction, Sarsa was originally implemented prior to Sarsa(λ), which can be found in Section 5 of the report.

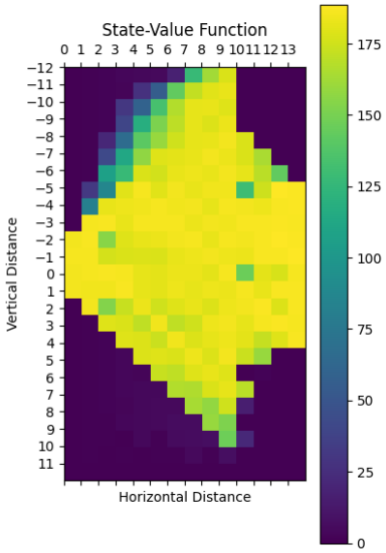
3 Results

3.1 State-Value Functions

3.1.1 Monte Carlo Control

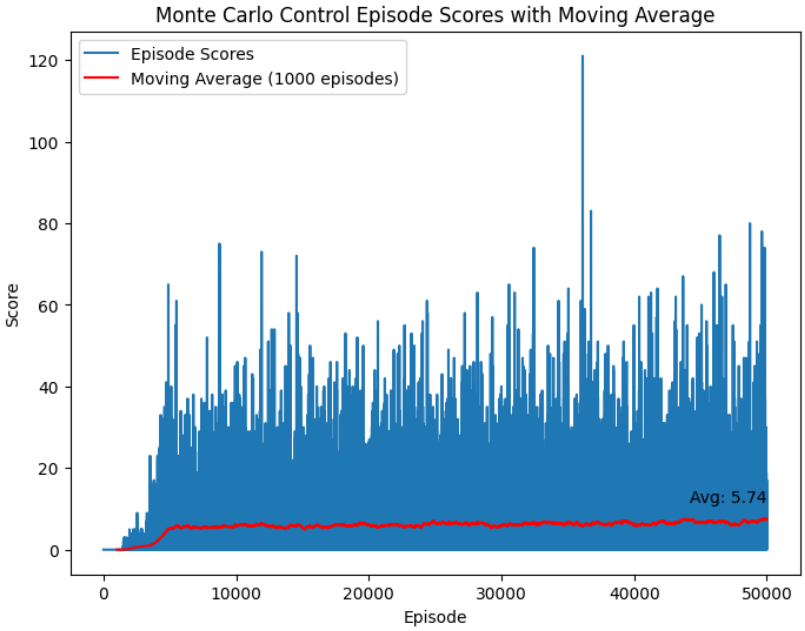


3.1.2 Sarsa

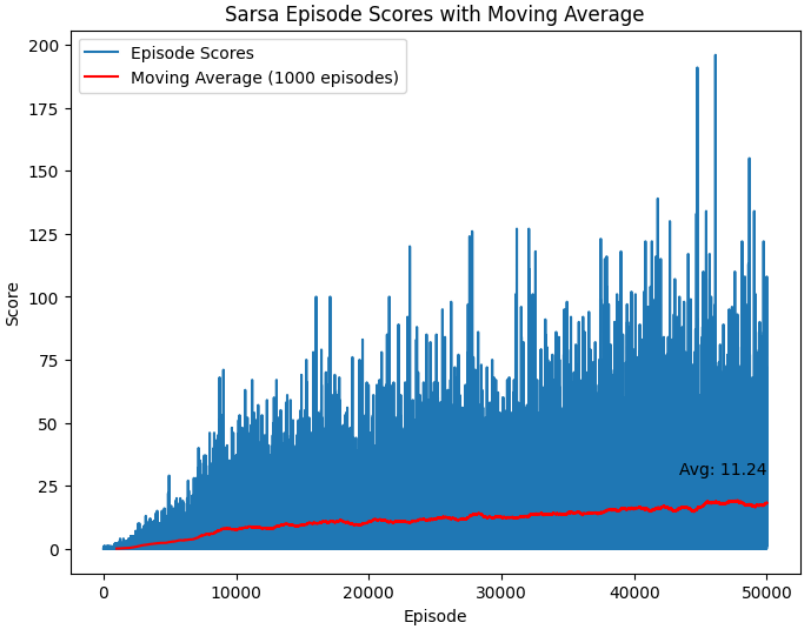


3.2 Performance Comparison

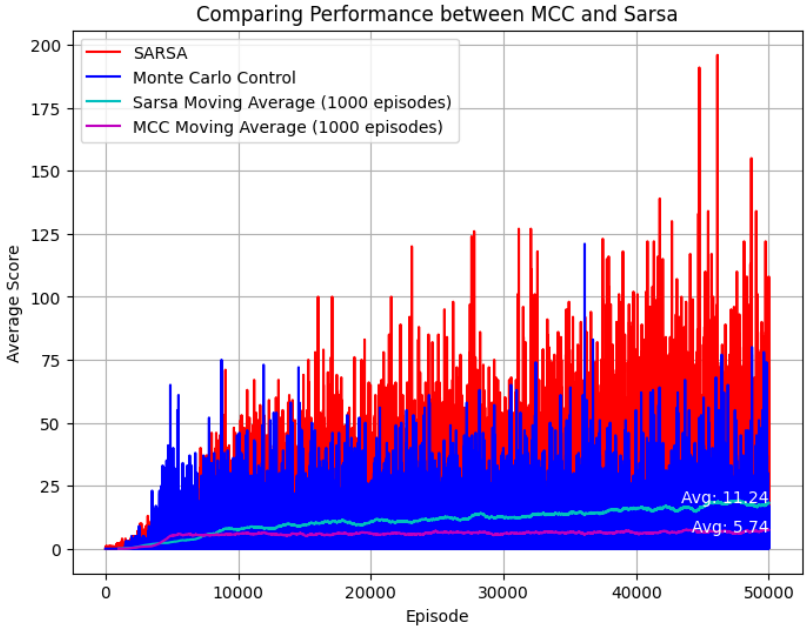
3.2.1 Monte Carlo Control



3.2.2 Sarsa



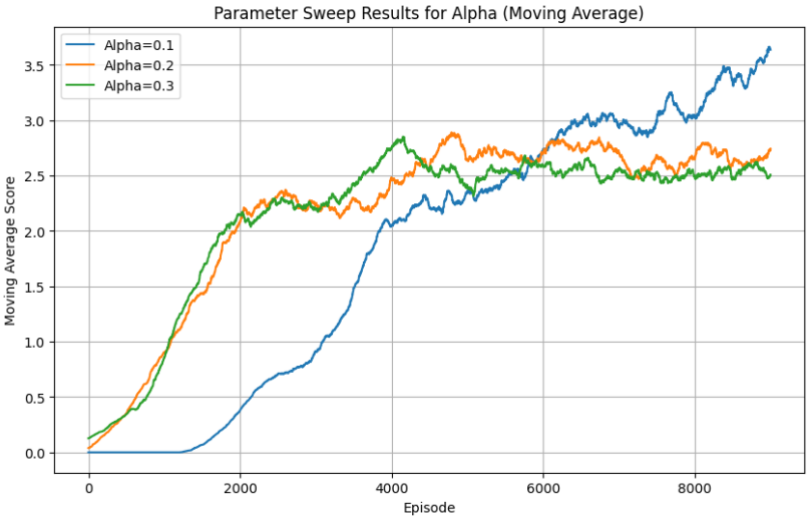
3.2.3 Combined Results of MCC and Sarsa



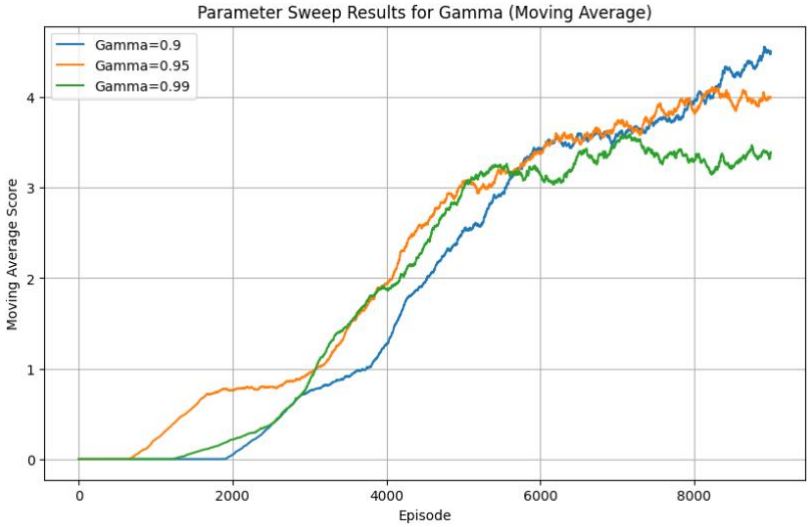
3.3 Parameter Sweeps

3.3.1 Monte Carlo Control

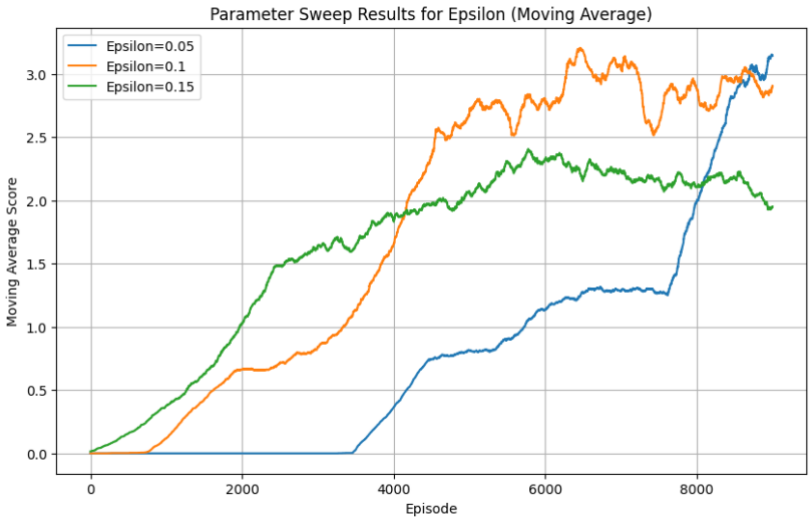
3.3.1.1 MCC Alpha



3.3.1.2 MCC Gamma

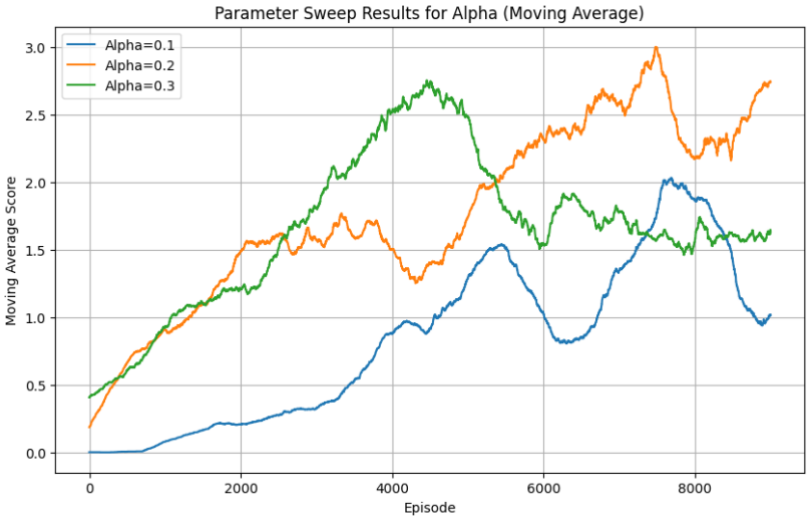


3.3.1.3 MCC Epsilon

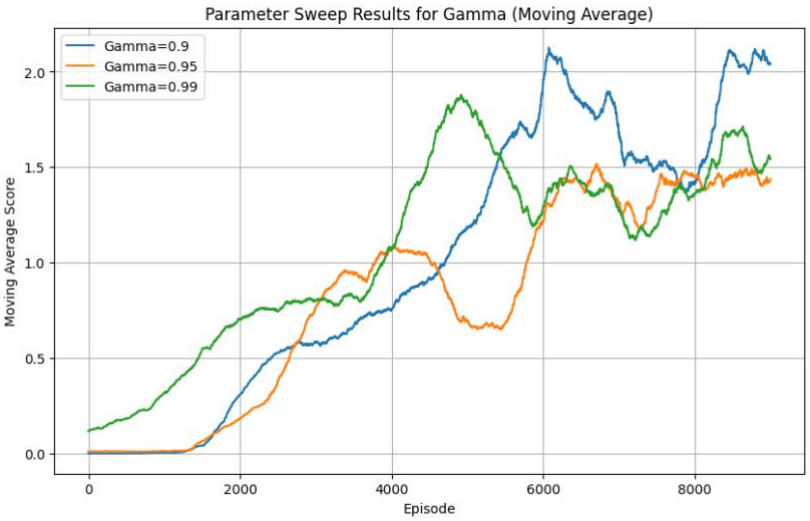


3.3.2 Sarsa

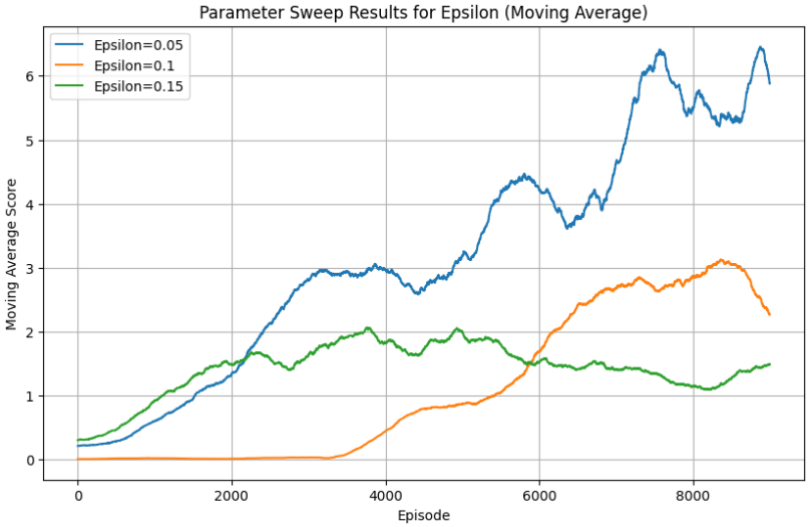
3.3.2.1 Sarsa Alpha



3.3.2.2 Sarsa Gamma



3.3.2.3 Sarsa Epsilon



3.4 Performance on Different Configuration

3.4.1 Monte Carlo Control

The pre-trained Monte Carlo Control agent was run twice in the original environment for the baseline and the new environment for 1000 episodes. The agent achieved average scores of 62.912 and 40.933, respectively.

3.4.2 Sarsa

The pre-trained Sarsa agent was run twice in the original environment for the baseline and the new environment for 1000 episodes. The agent achieved average scores of 24.13 and 23.476, respectively.

4 Discussion

Although the Monte Carlo Control agent had a better performance in the beginning, the Sarsa agent gained the ground, and eventually overtook the MC Control agent in the later episodes. While the highest score achieved by the MC agent was over 120, the Sarsa agent achieved almost 200. The adaptability to different game configuration shows that the MC agent has over 30% decrease in performance, while the Sarsa agent has less than 3% decrease in performance. This may be because of the state-value function. The heatmap of Sarsa state-value function shows a clear boundary on what action to be taken at a certain state when compared to the heatmap of MC Control state-value function. Overall, the best performing algorithm was Sarsa, with more than the double average score than the score of the Monte Carlo Control agent.

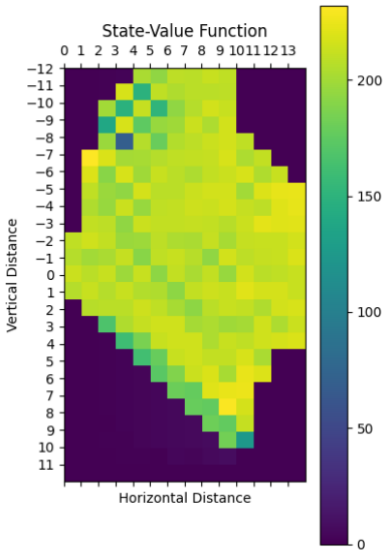
During the 1000 episodes of parameter sweeping process, it was determined that the optimal parameters were 0.1, 0.9 and 0.05 of alpha, gamma, and epsilon values for MC agent, and 0.2, 0.9 and 0.95 for Sarsa agent.

Unfortunately, the same agents cannot be used to run in the original implementation of the original flappy bird game environment due to a different dimension. It is possible to make modifications, such as increasing the scale of the Q-tables and policy table to accommodate the agents in the new environment. Alternative solution is to use the same algorithms with the different dimension as input to train the agents again.

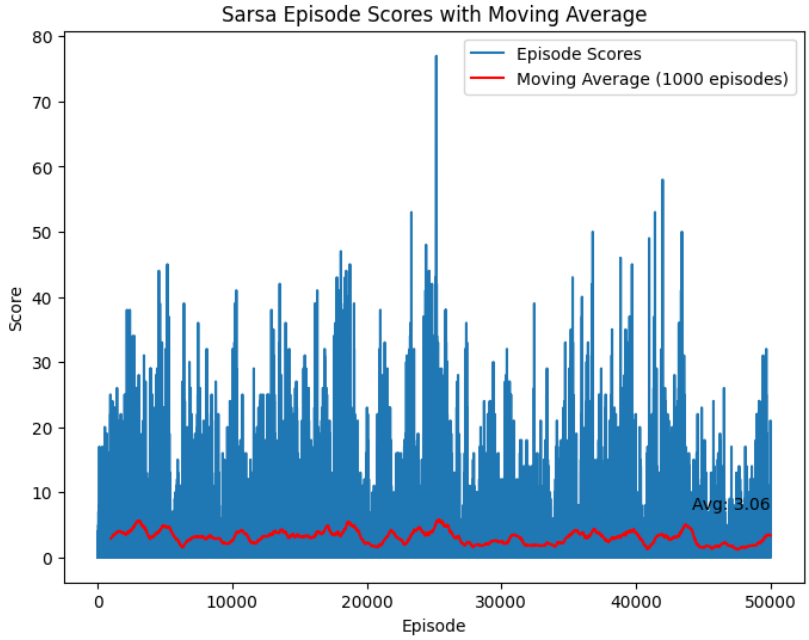
5 Sarsa(λ)

Sarsa(**λ**) algorithm was implemented according to the Section 12.7 of Introduction to Reinforcement Learning by Sutton and Barto. Trace eligibility was added from the original Sarsa model. The implementation of Monte Carlo Control, the value of the state, which was given by the game environment in the format of "(dy, dx)", had to be converted into one dimension integer to accommodate the Q-table.

5.1 State Value Function



5.2 Performance



5.3 Discussion

Like Sarsa agent, the state-value function of Sarsa(**λ**) agent appears to have a clear boundary than the MC Control agent. Even though the highest score achieved by the agent is over 75, the average of the overall performance is much worse than the MC Control agent and the Sarsa agent.