Cody Herberholz

CS445

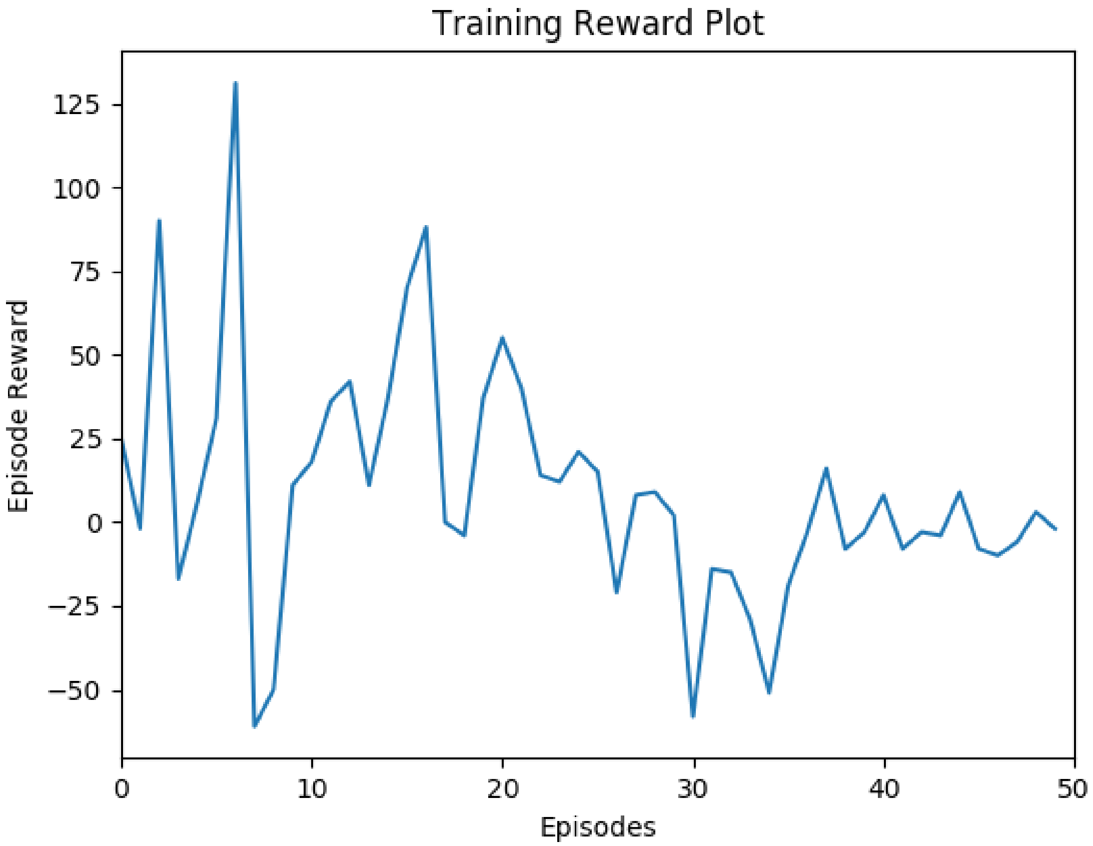
3/19/2017

Reinforcement Learning

Part 1

The task for part one was to write a Q-learning method for Robby using a Q-matrix. Robby will learn over a series of 5,000 episodes, during each of which he will perform 200 actions. The cans are placed randomly and Robby is placed randomly. The discount factor is set to 0.9 and learning rate to 0.2.

Test Reward Average -	7.697
Test Standard Deviation-	12.031



Discussion

When observing the results of the first experiment, the program responds positively toward Q-learning but by the time the 20th episode interval is hit the reward total per record starts to decrease which could be a byproduct of overfitting.

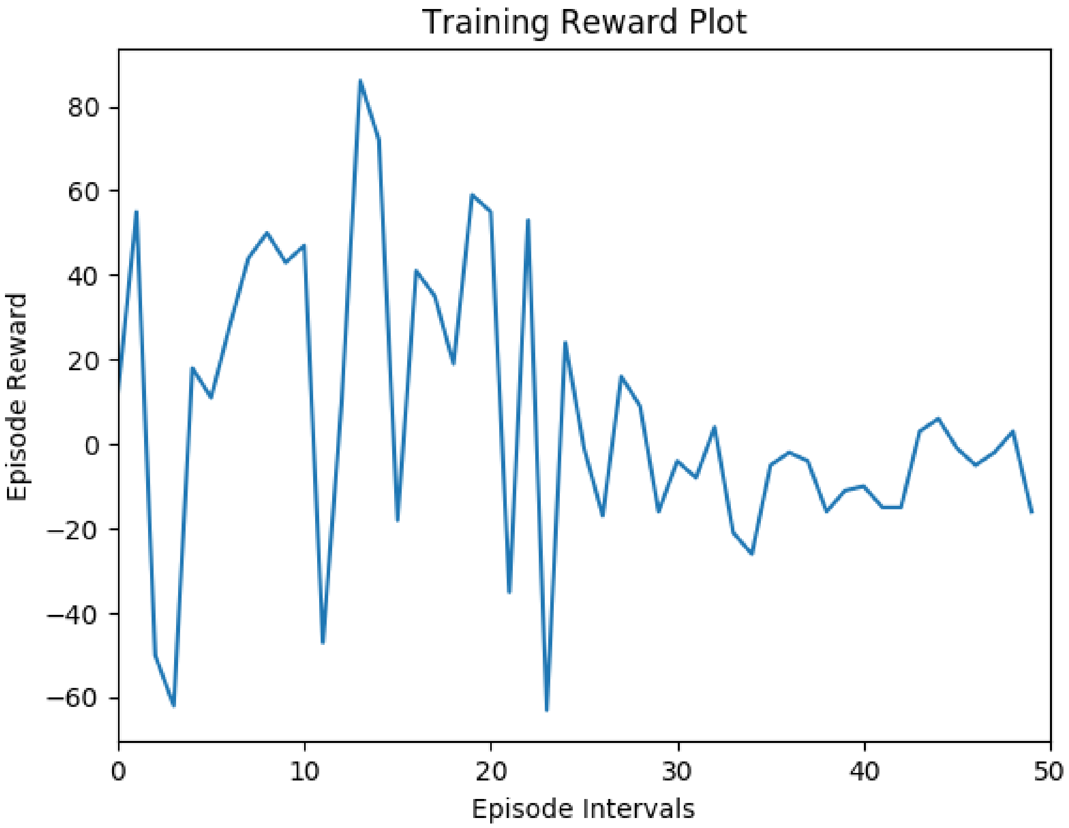
Part 2

The task of the second experiment was to follow the same procedure but to experiment with the learning rate. The 4 different learning rates that will be chosen are 0.25, 0.50, 0.75, and 1.00.

Learning Rate = 0.25

Test Reward Average - -35.981

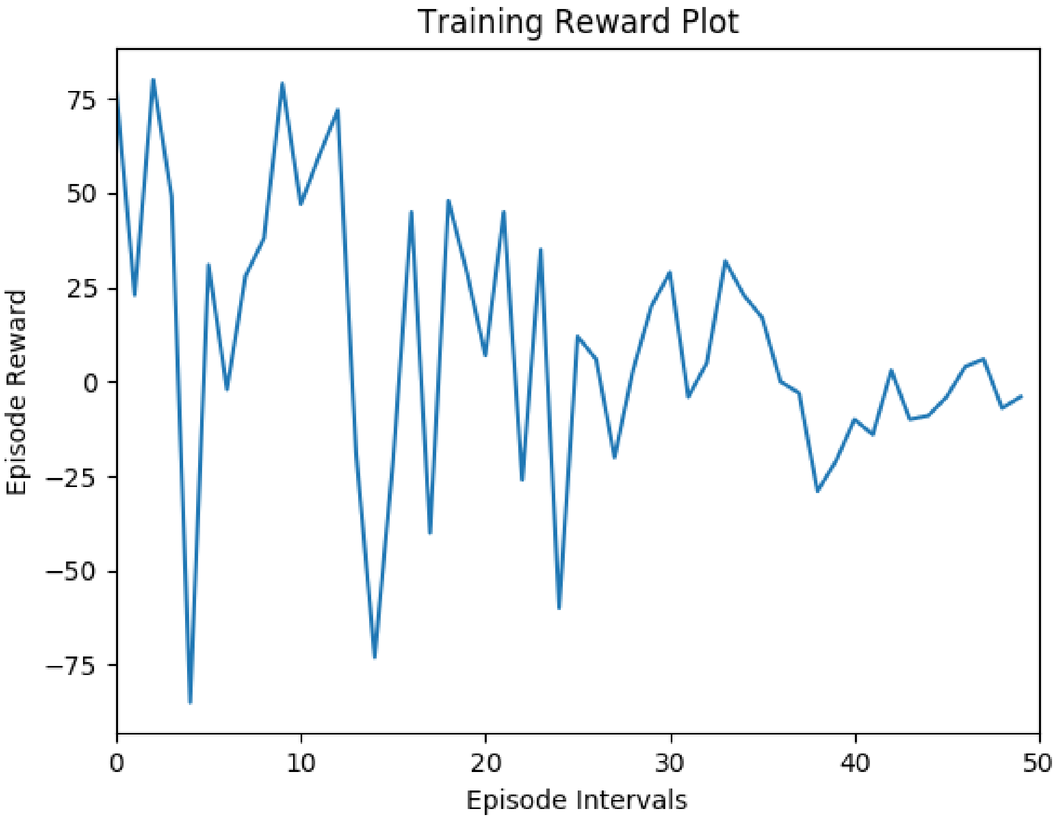
Test Standard Deviation- 43.519



Learning Rate = 0.50

Test Reward Average - 2.501

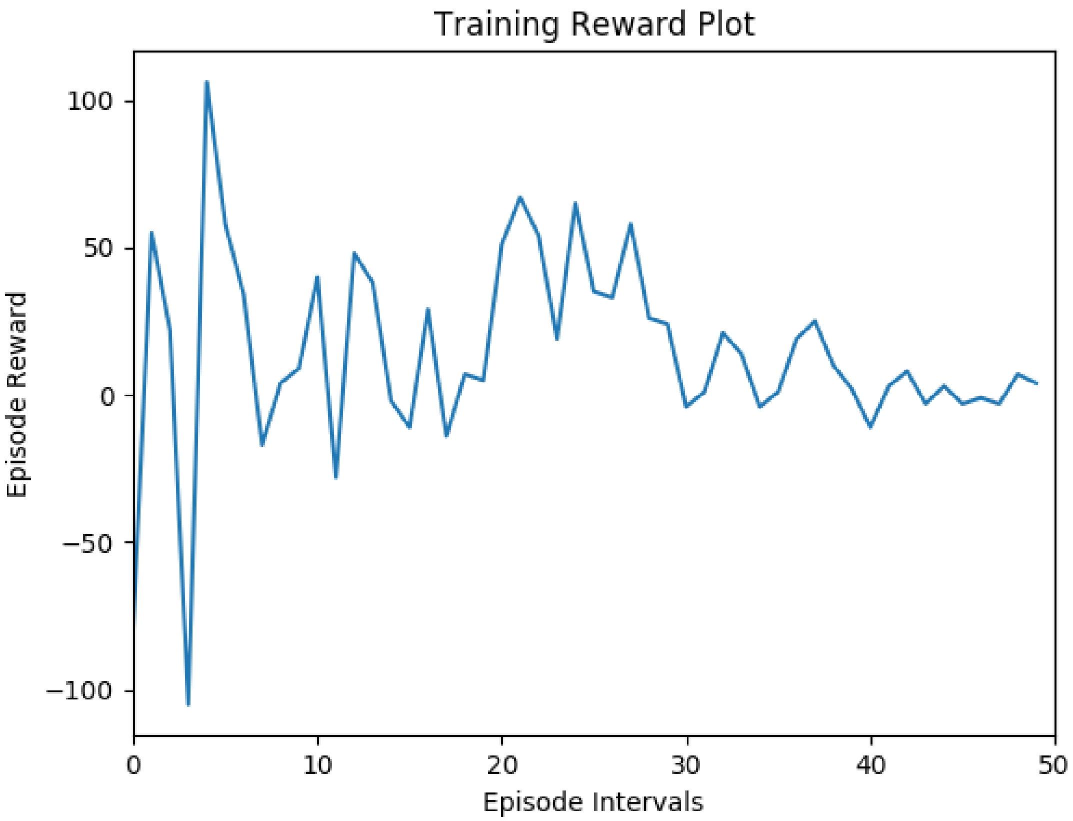
Test Standard Deviation- 15.557



Learning Rate = 0.75

Test Reward Average - -16.876

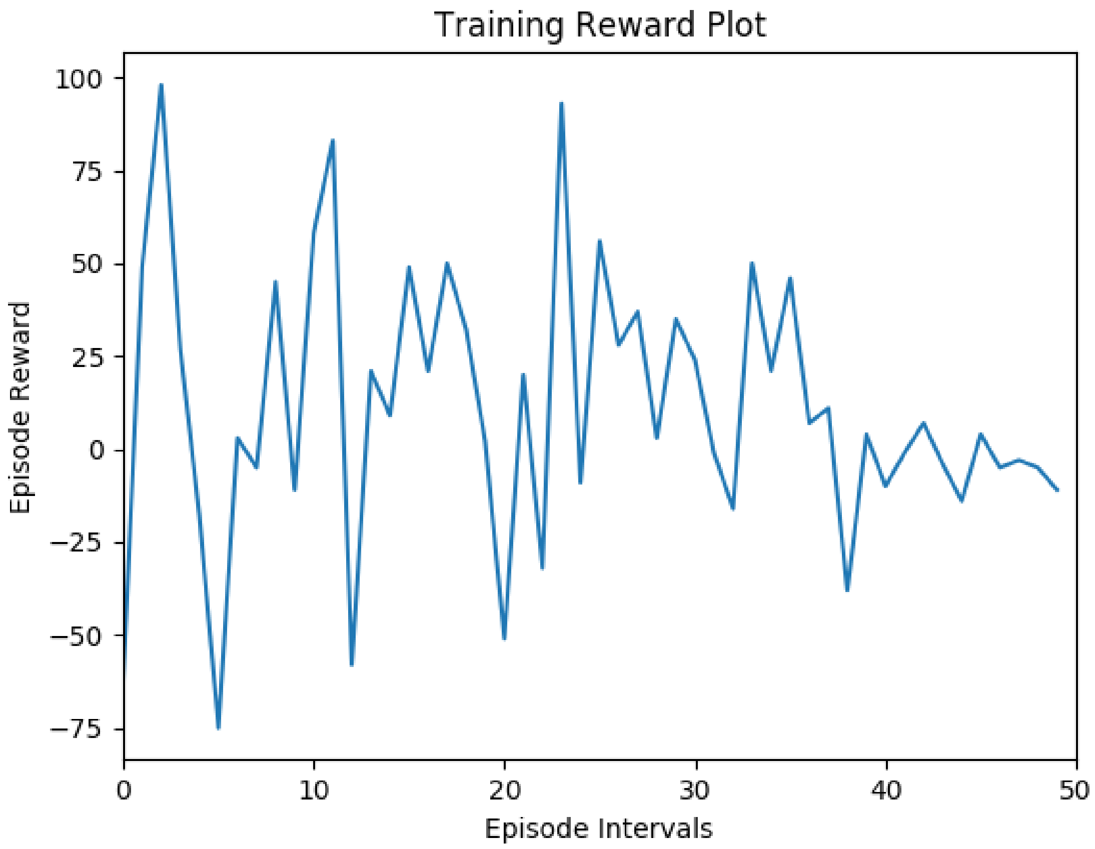
Test Standard Deviation- 40.777



Learning Rate = 1.00

Test Reward Average - 4.843

Test Standard Deviation- 13.610



Discussion

When observing the data, as the learning rate increased, the chance for increased rewards per episode occurs later on in the data. This can be seen from the occurrence of a high reward in the learning rate = 1.0 graph. Also seen in the learning rate = 0.75 graph is the decrease of the effect of overfitting.

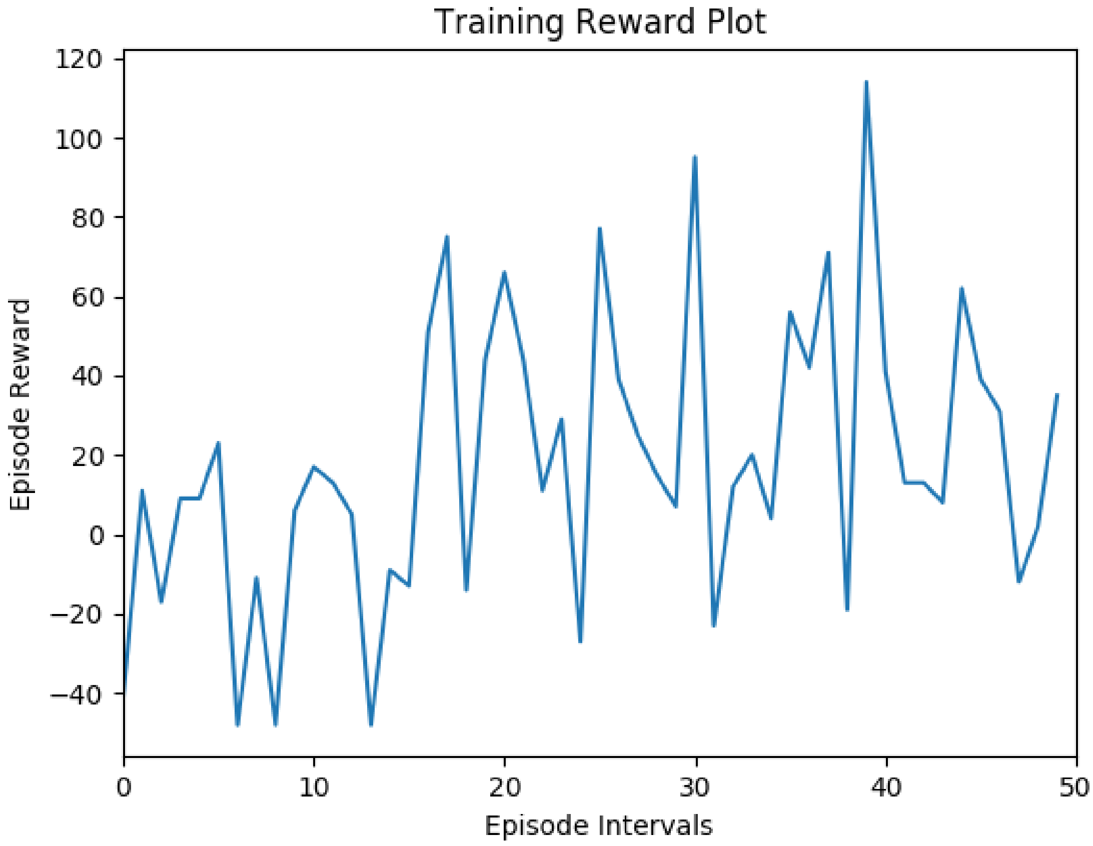
Part 3

The task of the third experiment was to follow the same procedure but to experiment with a constant value of epsilon.

Epsilon = 0.60

Test Reward Average - 6.973

Test Standard Deviation- 37.320



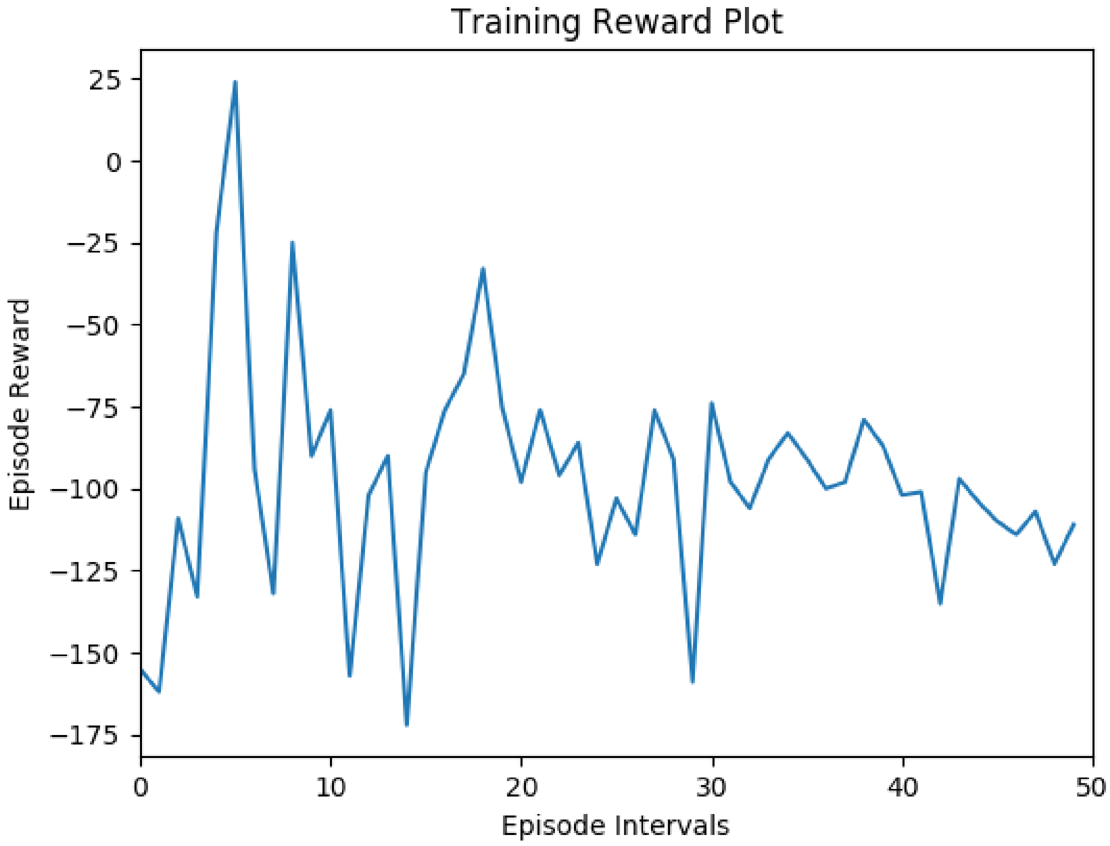
Discussion

When observing the data with the new constant epsilon, the program has an improved chance of doing better in the later changes. Instead of choosing the action with the best Q-table value throughout the second half of the episode intervals, the program can now choose to be random and have more of an opportunity to not get stuck doing something that might not be the most optimal path for rewards.

Part 4

The task of the fourth experiment was to follow the same procedure but to provide an action tax, which subtracts 0.5 for each action taken.

Test Reward Average -	-96.903
Test Standard Deviation-	13.156



Discussion

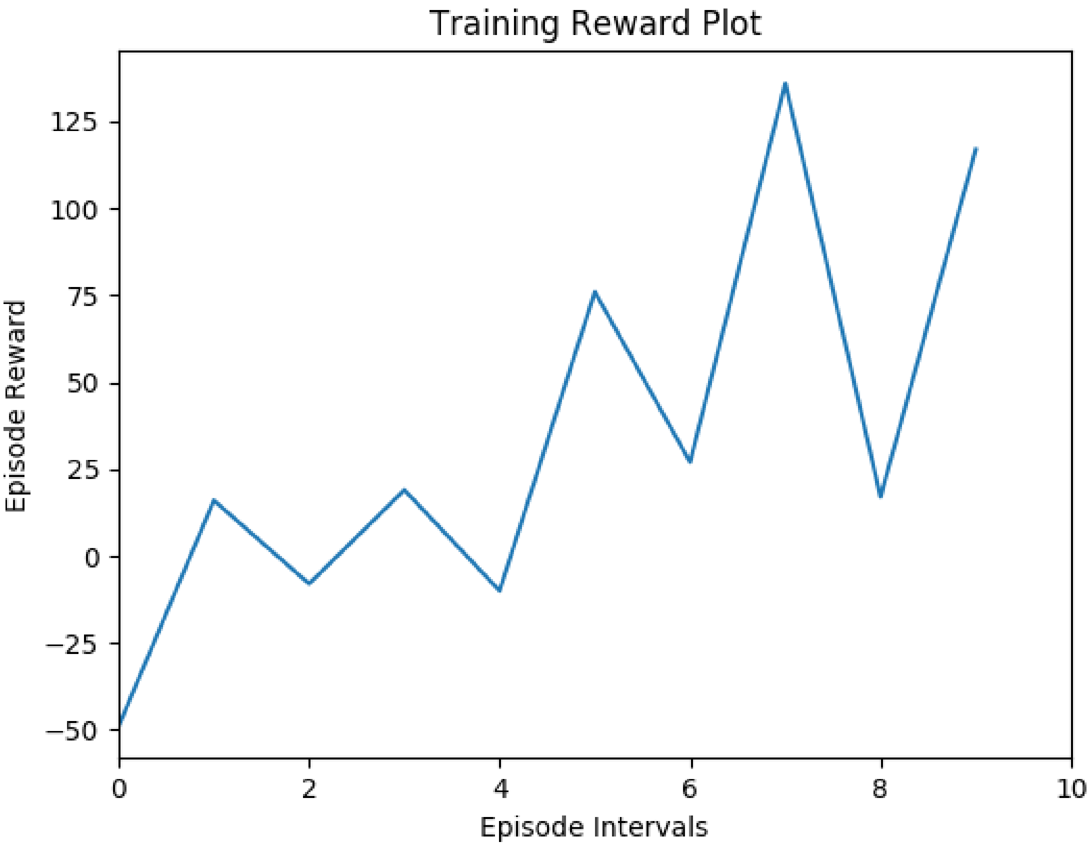
When observing the data, it can be seen that the data is much more prone to be providing negative rewards. Initially it starts out low and increases up to a positive amount but then drops and never reaches that pointed again.

Part 4

The task of the fifth experiment I shall follow the same procedures but us varying lengths of episodes and actions to see how it affects the training plot and testing. The new episodes shall be 1000, 2000, and 3000. Actions will be increased to a constant 500.

N = 1000, M = 500

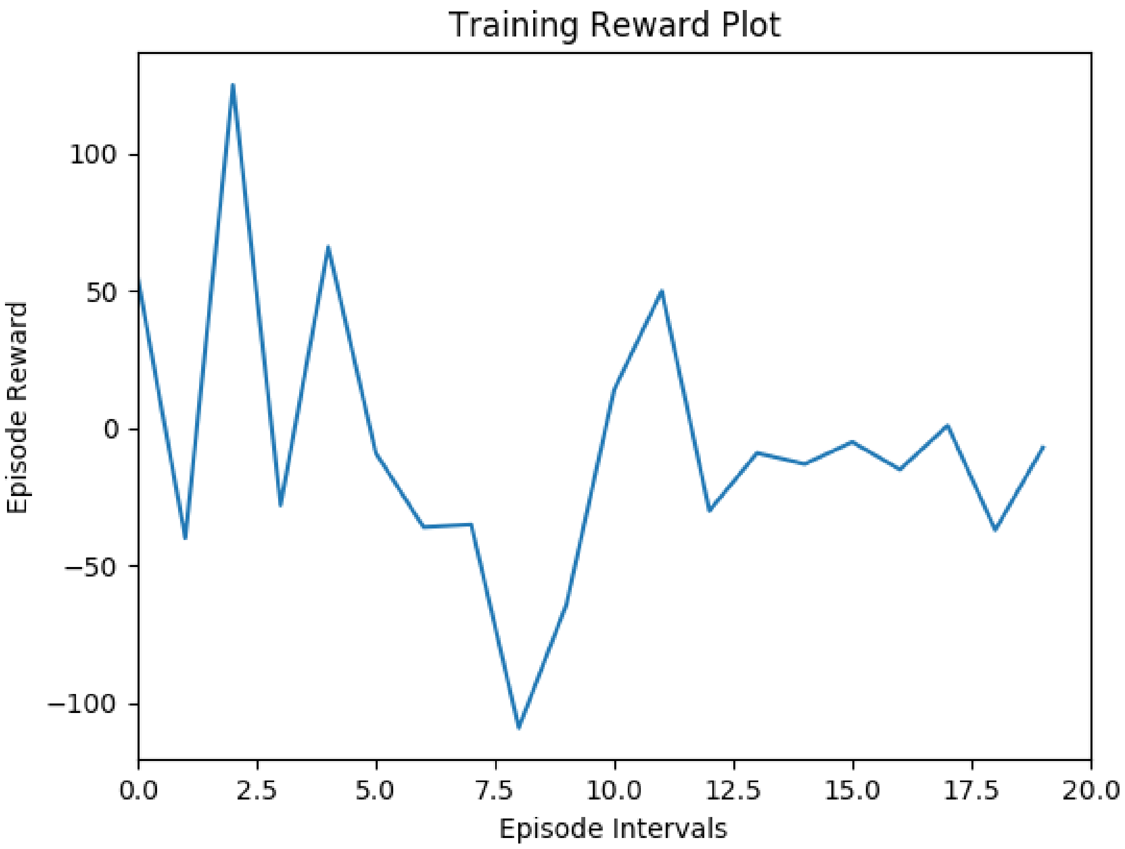
Test Reward Average -	-25.302
Test Standard Deviation-	75.834



N = 2000, M = 500

Test Reward Average - -84.852

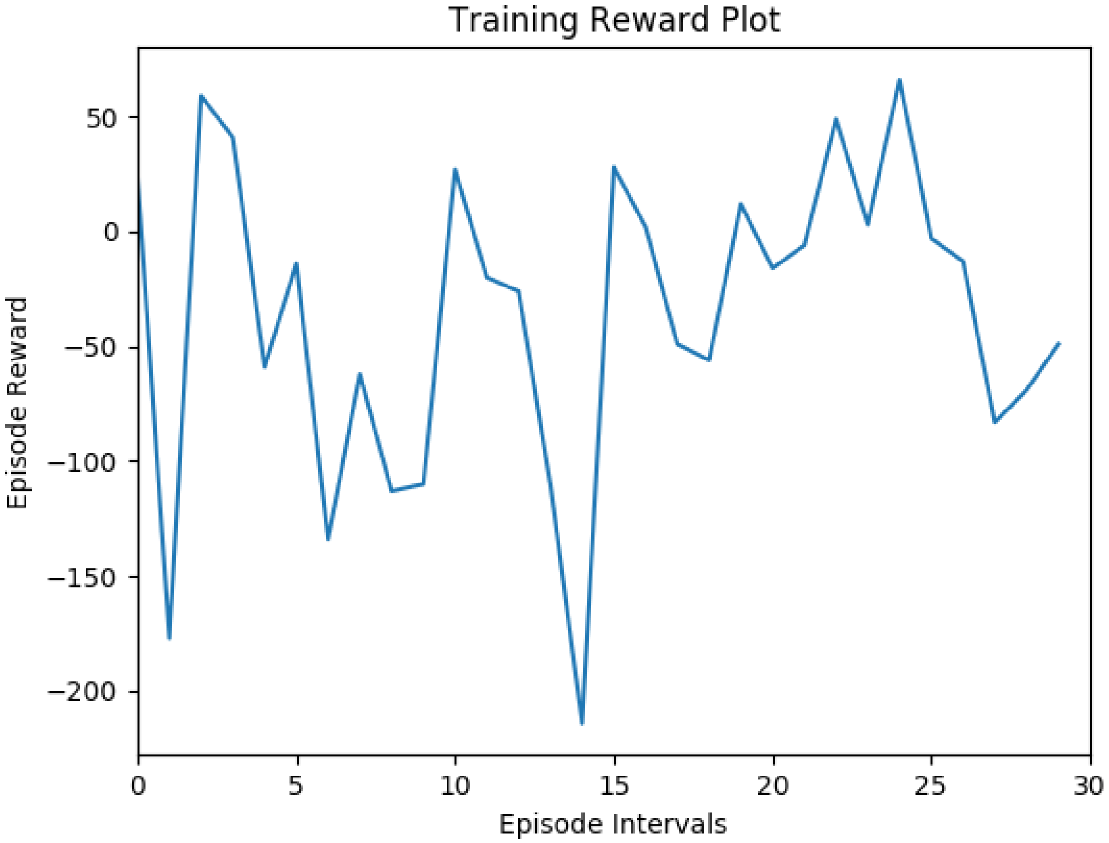
Test Standard Deviation- 62.907



N = 3000, M = 500

Test Reward Average - -55.888

Test Standard Deviation- 54.609



Discussion

When observing the data, as the number of episodes increased with 500 actions per episode the standard deviation decreased for the test set. As expected, the smallest episode provided gave the best data with the highest amount of rewards per episode.