CS747: Programming Assignment 3

Report by: Atharva Mete | 190010012

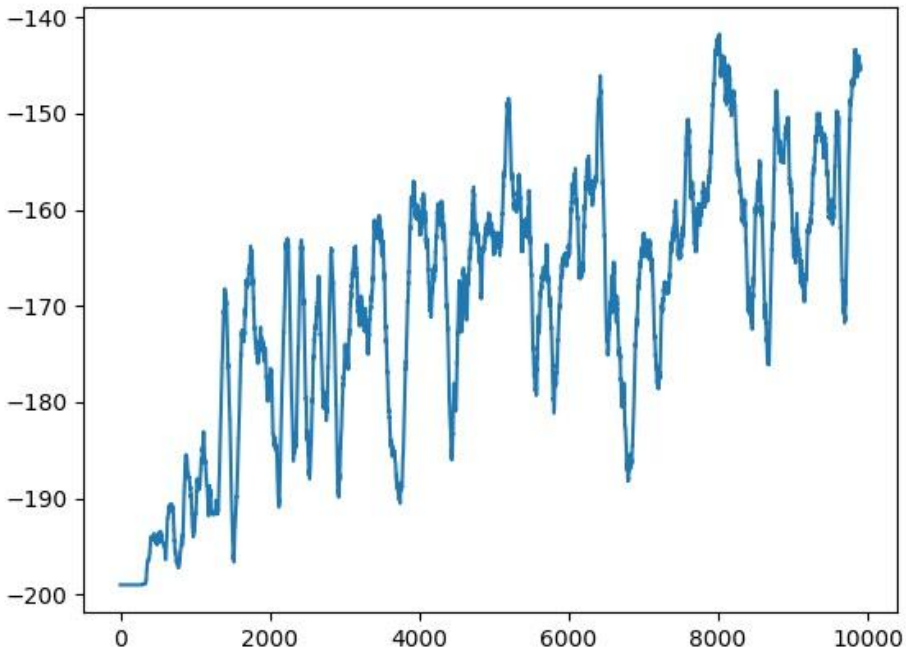
Introduction

In this assignment, I implemented SARSA(0) algorithm for the reinforcement learning problem of an agent which can learn to escape the valley. The algorithm is implemented using weights, and the weights are updated with linear TD(0).

Task 1: Tabular Sarsa

In this, we represent state by a 2D matrix of size (n_pos*n_vel,3), where n_pos and n_vel are corresponding discretized intervals. Thus we have weights corresponding to every pair of position, velocity, action.
get_table_features function: This function returns this state matrix with one element corresponding to a particular position and velocity as '1', and others as '0'.
choose_action function: I have used the epsilon greedy algorithm for choosing action. When there is a tie the action is chosen at random. epsilon = 0.1
sarsa_update function: This function is used to update the corresponding weights. Linear TD(0) is used to perform the weight update with learning_rate = 0.1

Following is the plot I got after training for 10000 episodes:

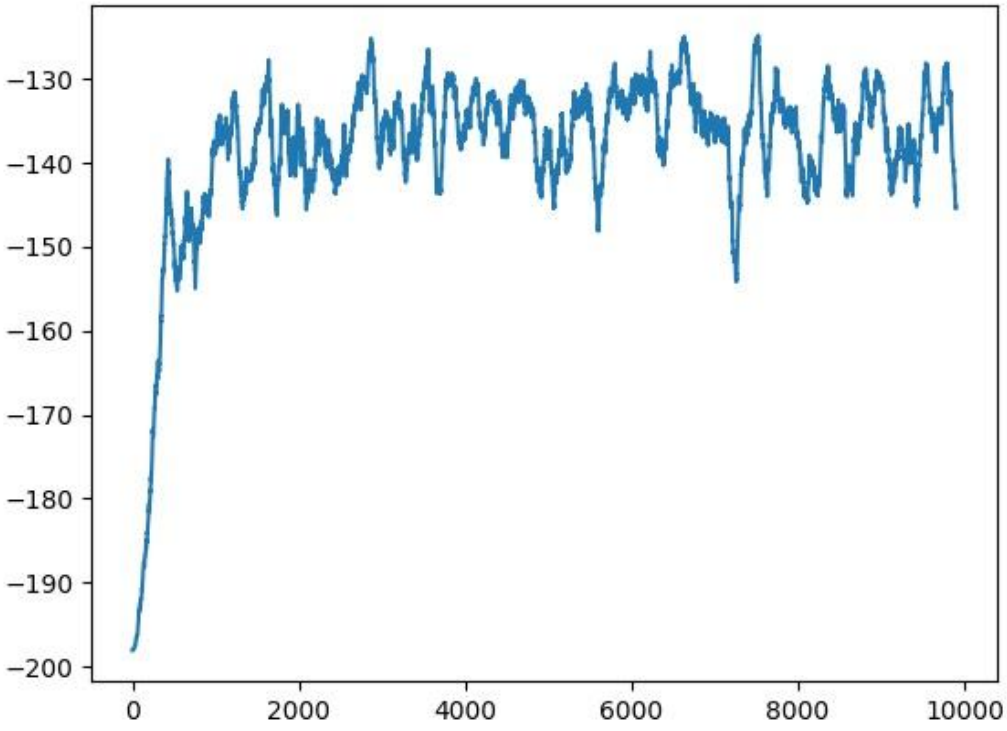


Task 2: Sarsa with Linear Function Approximation

In this task, I have used tile coding so as to come up with a better feature vector.
The choose_action and sarsa_update functions are the same as task 1.

get_better_features function: In this, we represent state by a 2D matrix of size (n_tiles,n_pos*n_vel,3), where n_tiles is the number of tilings and n_pos and n_vel are corresponding discretized intervals. Thus we have weights corresponding to every pair of position, velocity, action for each layer of tiling.
This function returns the state matrix with a total of 'n_tiles' entries as '1' (one element from each layer) and others as '0'.

Following is the plot I got after training for 10000 episodes:



Observation:

Task 1:

- I got a test reward of **-144.58** for 100 episodes.
- Increasing epsilon isn't much help as when weights are zero enough amount of exploration is already being done by the algorithm.
- A large learning rate causes the value of Q to diverge and decreases the reward.

Task 2:

- I got a test reward of **-118.94** for 100 episodes.
- Increasing epsilon isn't much help as when weights are zero enough amount of exploration is already being done by the algorithm.
- The number of tilings is set to 5, increasing the number of tiling to 10 increases the reward