CS747 Assignment 3

Pradipta Parag Bora

190050089

1 Task 1

The following plot was observed while training the agent using simple tabular SARSA:

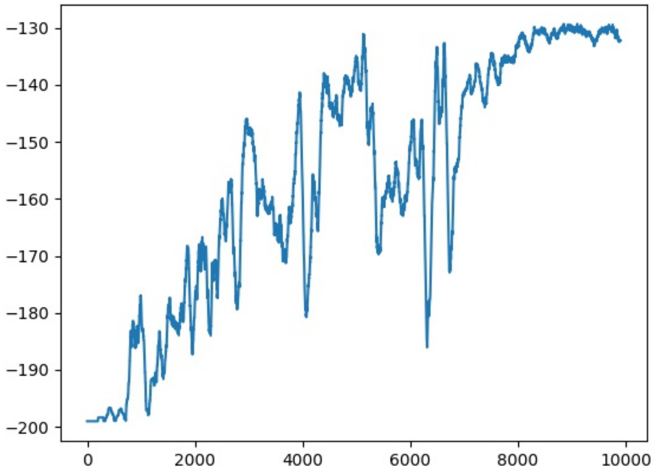

Figure 1: Plot of reward vs episodes

We have set $\epsilon = 0.001$, learning rate $= 0.1$ and initial weight vector to be 0 for this agent. Setting initial weight vector to be 0 implies that we have started with initial Q value estimate to be 0 which we have done to enable optimistic start so that the agent explores more. We have also discretised the x space into 20 bins and the v space is also broken into 20 bins.
Therefore since the initial Q value estimates are far higher than their actual values (as rewards are either 0 or -1) this agent will be optimistic towards non explored states and will therefore explore them which means that we don't need a high value of ϵ here.
Observing the plot the agent steadily learns in the beginning but as there are many trajectories sometimes there are ups and downs in the agents behaviour in the learning phase. However after 7000 episodes we seem to have mostly arrived at a policy that takes around 130 steps to finish the mountain car game. This is through traditional tabular SARSA since our feature vector only has one entry set to one. Since there needs be one entry for each value, the total size of the feature vector is $20 \times 20 \times 3$, since we need different values for each action.
On a test run this agent is able to finish the game in 131 steps (averaged over 100 runs) to give a score of -131.

2 Task 2

Here we have used 8 tilings, so we have essentially 8 features for each state (where each feature is computed similar to the last part but with an offset corresponding to each tiling). We expect this agent to generalise and learn quickly than the standard SARSA implementation. We have used the sample value of ϵ and number of tiles and learning rate as that of the previous part.
We get the following plot:

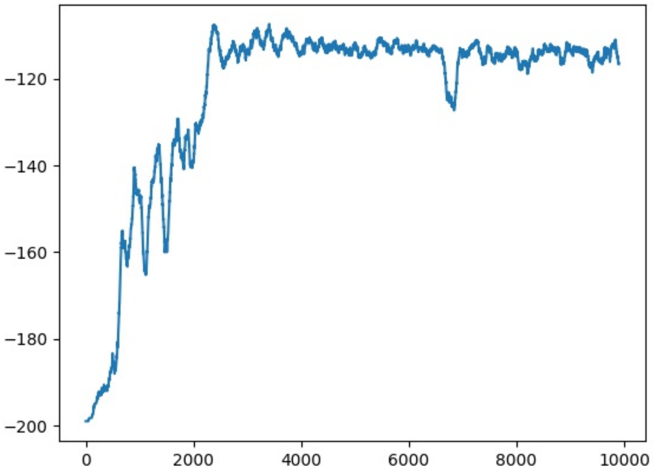

Figure 2: Plot of reward vs episodes

We can see that this agent converges quickly to a better policy than that obtained in the previous part. We can attribute this to faster learning due to generalisation of tile coding. Once it reaches -110 the value seems to be more or less fixed but with some fluctuations corresponding to random sampling of actions to take and exploration.
Further increasing the number of tilings improves the scores but it also becomes slower to train so we are fixing it to 8 for now (since we are meeting our goal). Running this learnt policy on 100 test episodes gives an average score of -113.