name: Zhengming Zhu, Xianao Lu
ID: 19990130-2035 20021201-3338

## 1 Problem2

### 2.c RL with Linear Function approximators

For this RL problem, we use linear function approximation to estimate the Q function:$V_\omega(s) = \omega^T\phi(s)$.The training process is aimed to tuning the vector $\omega = [\omega_{a1}, \omega_{a2}, \ldots\ldots, \omega_{aA}]$.
We choose Fourier Basis to approximate the Q function because the continuous state space. The Fourier Basis with p-th order can be defined as:

$$\phi_i(s) = cos(\pi\eta_i^T s), i \in \{i, \ldots\ldots, m\}$$

The design of $\eta_i$ is very important, for this specific situation, the state spaces has two dimensions, and the order p = 2. we choose:

$$\eta_1 = [0,0]$$

$$\eta_2 = [1,0]$$

$$\eta_3 = [0,1]$$

$$\eta_4 = [1,1]$$

$$\eta_5 = [2,0]$$

$$\eta_6 = [0,2]$$

$$\eta_7 = [1,2]$$

$$\eta_8 = [2,1]$$

$$\eta_9 = [2,2]$$

Because the $\eta_i = [a,0]$ can capture the different frequency information of single $x_1$ variable, where as $\eta_i = [a,b]$ considers both $x_1$ and $x_2$ together. Furthermore, they also consider their interactions at different frequencies.
We reference the algorithm $Sarsa(\lambda)$, and the eligibility traces $\lambda \in [0,1]$ indicates a mechanism to bridge the gap between one-step TD learning (like SARSA) and Monte Carlo methods. They keep track of which states and actions have been visited and how recently and significantly they were encountered. In the lab, we use $\lambda = 0.1$.
For this part, we used the SGD with Momentum to help update the SGD process:

$$v = mv + \alpha\delta e$$

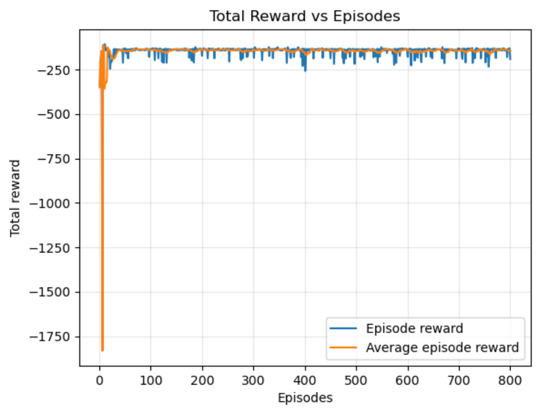
$$\omega = \omega + v$$

$$where \quad m \in [0,1)$$

we set the m = 0.2 or 0.1 and apply the np.clip() to avoid the exploding gradient problem. Then, the learning rate decay really help convergence when training. In practice, whenever total reward R > -250, the learning rate will decrease 10%, and when R > -130, the learning rate will decrease 30%.

## 1.1 2.d Do the following analysis of the policy

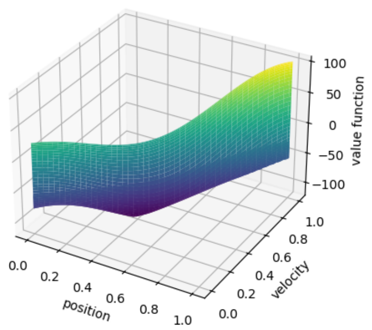
Plot a figure showing how the episodic reward changes across episodes during training:



Figur 1: $\lambda = 0.1, \varepsilon = 0.01, momentum = 0.2, \gamma = 1.0, \alpha_0 = 0.2, episodes = 800$

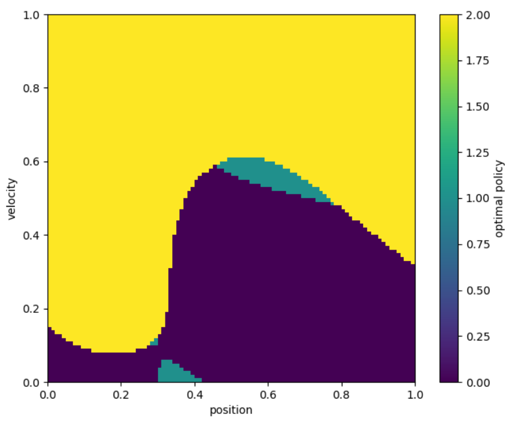
The result shows that the episodic total reward increases dramatically at first and converge.

Plot a 3D plot of the Optimal value function and policy:
Because this environment has a continuous state space, the following figures indicate the value function of the optimal policy over the state space domain. We can see that when the position close to 1.0 the value is bigger:



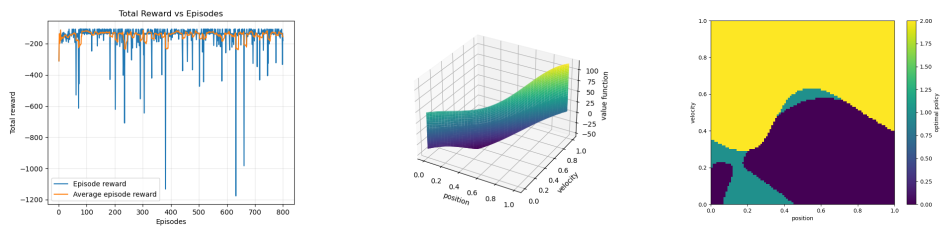
Figur 2: Optimal value function

Plot the optimal policy over the state space domain (you should get a 3D plot). As we can see according to the heat plot, we can conclude that the low velocities make the car take action to perform should be going to the left, while with high velocities the action should be going to the right.



Figur 3: Optimal policy

If we do not include $\eta = [0,0]$ in our basis, we can see the results as follows: We can observe

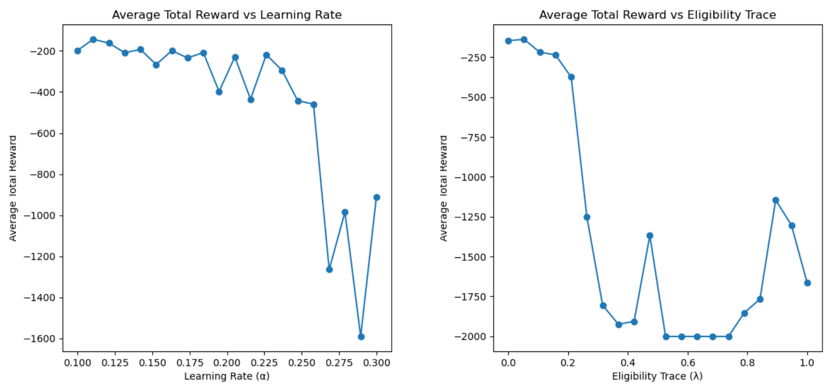


Figur 4: Results when $[0,0] \notin \eta_i$

that it might limit its ability to capture the overall level (mean) of the data, which makes the episodic total reward waves dramatically. Also, the optimal policy heat plot changes, the times of stay action decrease. However, the value function did not change much.

## 1.2 2.e Do the following analysis of the training process

Show the average total reward of the policy as a function of $\alpha$, the learning rate. We suggest to compute the average total reward out of 50 episodes (if possible, show confidence intervals). Repeat the analysis with $\lambda$, the eligibility trace. Analyse the plots.
We test different values of $\alpha_0$ and $\lambda$ from [0.1,0.3] and [0,1], As the result shows, The too high learning rate affect the average total reward much, because it may cause the train process unstable. For another aspect, the low $\lambda$ has a better performance because the model focus on more recently information.



Figur 5: Results of different $\alpha_0$ and $\lambda$

Discuss different strategies to initialize the Q-values. Which one works best in your opinion? Explain why.
In general, we initialize the Q-values as zero, there are some other strategies like random initialization, which encourages exploring at the first. For this situation, since the Q-value must be negative number, we can initial it as a negative number which is close to the target. By doing so, convergence time may shorter.

## 1.3 2.f Choose an optimal policy that solves (b).

We save the trained weight file as weights.pkl and check the solution. As a result, we achieves an average total reward of -134.2 with confidence 95%