DSAI homework 4 Report
F74031158 CSIE107 王韻華

Idea

In this homework, I used Deep Q Network algorithm to finish the Mountain-Car mission. I trained the NN by giving the input of current state (current position of the car) and the output of action-value (probability of going left , stand, going right). During the training process, I updated action-value with high reward as car going more away from the origin position, I also give more weight of the reward as the car going more right away from origin point.

Reward += 1 + abs(observation[0]) + 3 * observation[0] (observation[0] = position of the car)

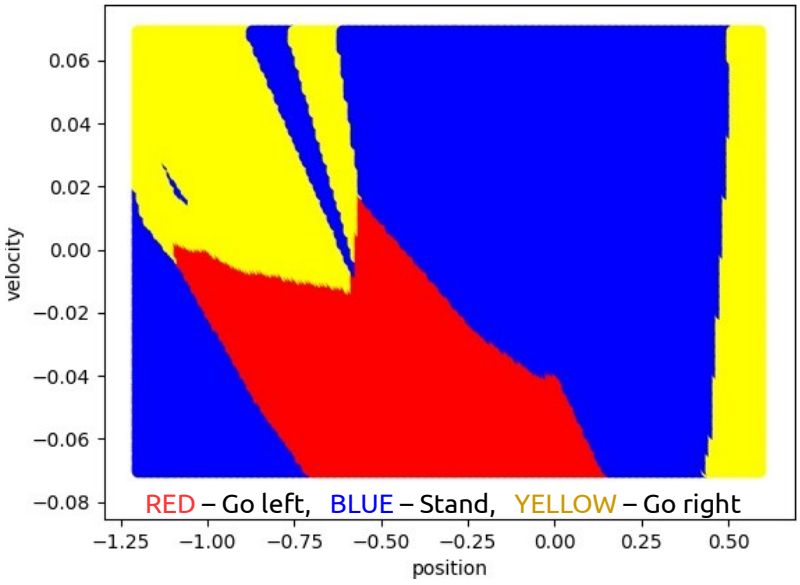
Questions

Q1 : *What kind of RL algorithms did you use?*
A1 : I used Deep Q Network algorithm to finish the Mountain-Car mission.
It's a value-based algorithm because it rely on the neuron network to learn the action-value so that it can further choose a proper action to perform.
Instead of directly update the long-term reward as Policy gradient, DQN arrive at the optimal policy indirectly by learning the optimal action-value.

Q2 : *This algorithms is off-policy or on-policy? why?*
A2 : DQN is off-policy algorithm, since its process of updating Q-value is not always decided by the current policy. It has a chance to make a random action and further update Q-value based on the reward of the random choice.

Q3 : *How does your algorithm solve the correlation problem in the same MDP?*
A3 : As long as the car climb to the mountain using actions that generated by DQN agent, I will give great reward to all the previous actions. My idea is to regard all the actions that helps to reach the mountain as good actions, further train my agent better.

Analysis



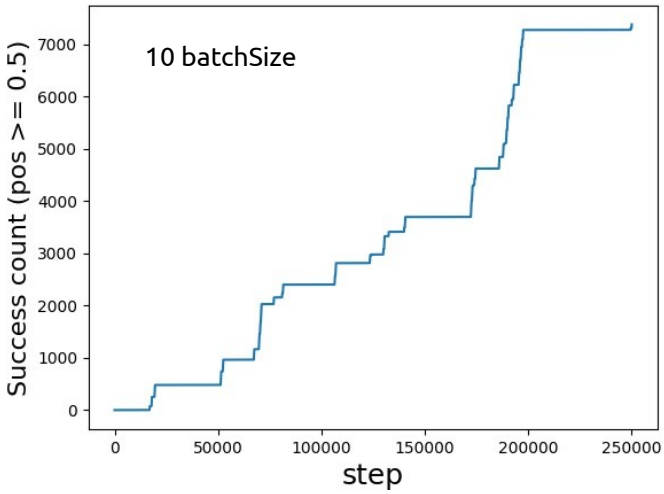
The slice unit of position : 0.01 , velocity : 0.001

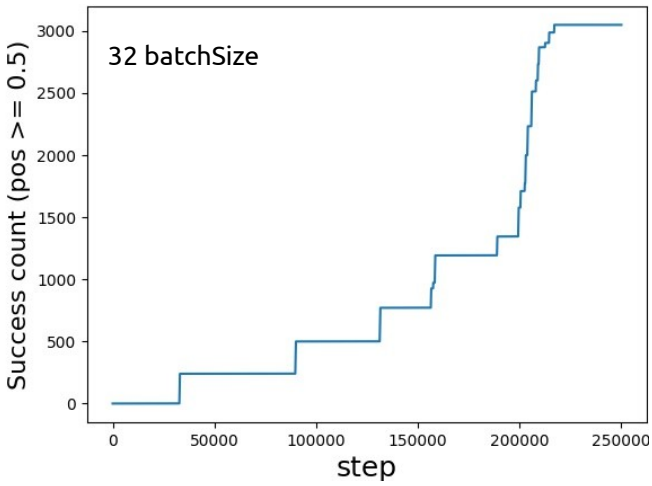
Each point in the figure is the action-value output by the model given position and velocity.

Over all, in order to go up to reach the flag, the agent will first go left until the car has some speed.

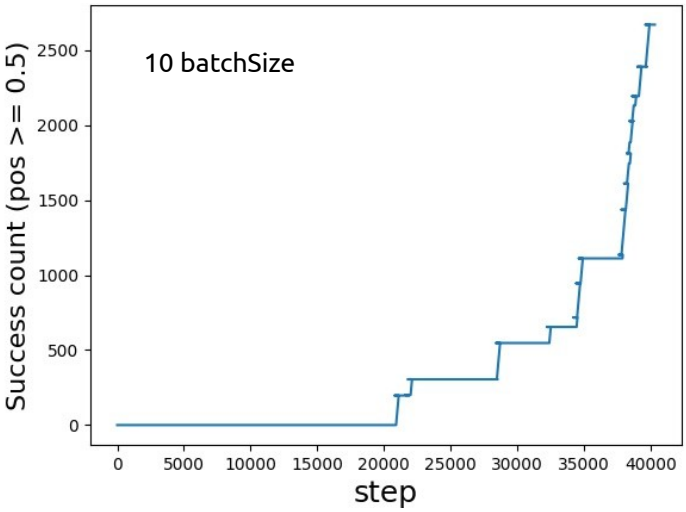
At last, as the car going right and climbing the mountain, the agent will tend to be stand by to low the speed.

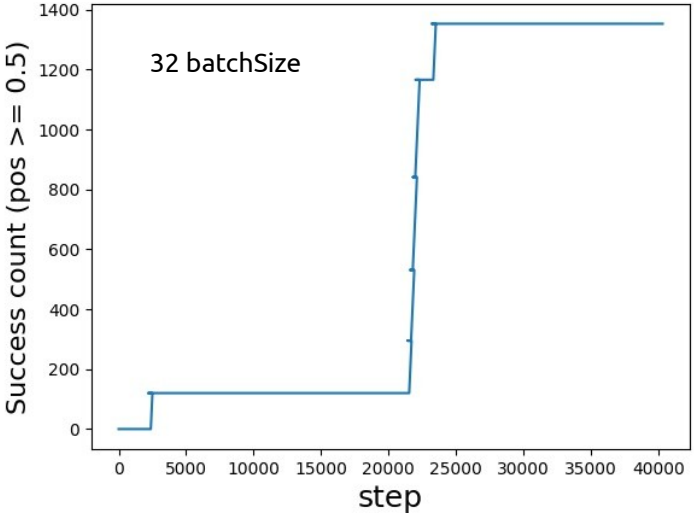
learning-curve with agent trained by 500 episodes



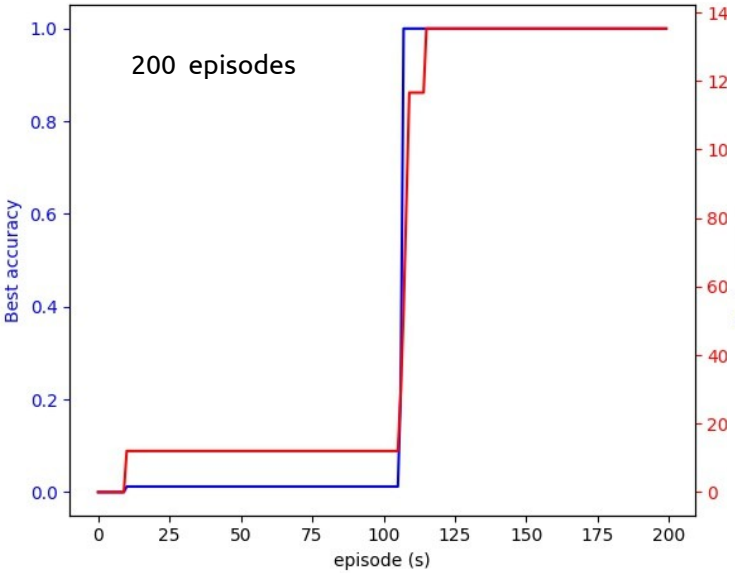


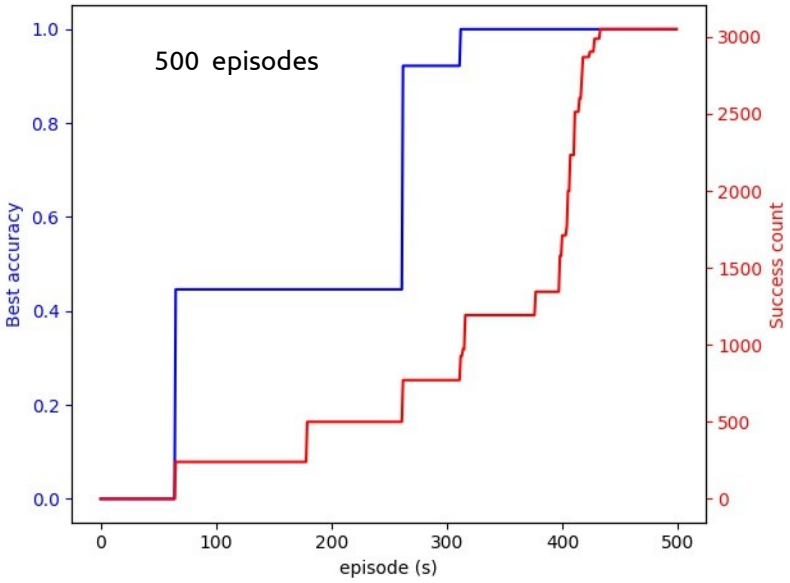
learning-curve with agent trained by 200 episodes





Accuracy and Success count with 32 batchSize





We can see that smaller batch size brings better training process(more success).

Furthermore, because the agent learn from random good event, so the train-episodes has weaker influence.