Programming Assignment 1 Report

Hojeong Lee

1. Task 1

1) Task 1.3

For the given hyperparameters, the ‘average_step_rewards’ graph is like Figure 1.

Figure 1.

2) Task 1.4

To improve over the Task 1.3, I run the experiment for 100k steps like Figure 2. The other metrics such as ‘average_score’ or ‘eval_average_score’ seem to be converged compared to the shorter experiment.

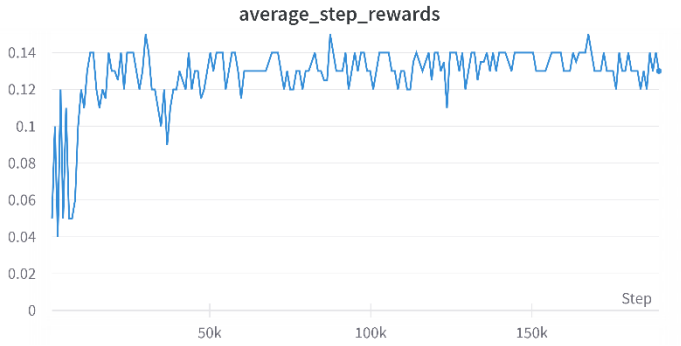

Figure 2.

I set the network more complex by increasing hidden_size as 1024 and execute for 30k steps. At some points in 'average_step_rewards' metric, it achieves higher value than the above two experiments. However, as the network became more complex, the other metrics seems to be overfitted.

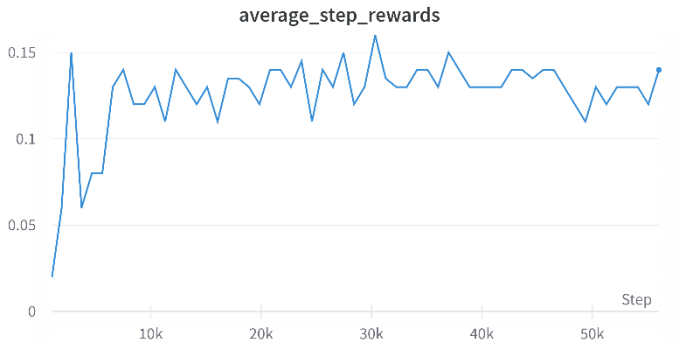

Figure 3.

So, I conversely decrease the size of the network to 256. Then, I got the much better results in Figure 4. The policy loss increases step by step, and the rapid changes of it do not appear thanks to clipping.

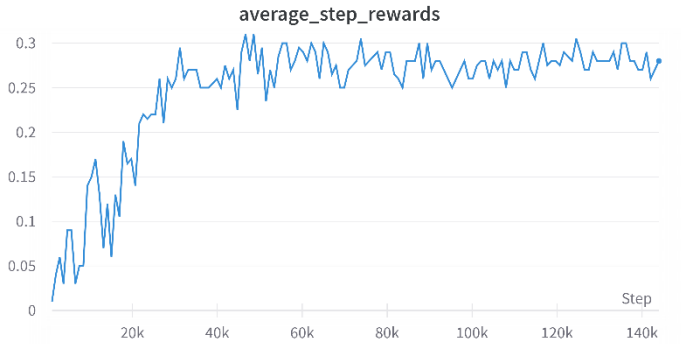

Figure 4.

2. Task 2

1) Task 2.2

For task 2, I set the self.use_KL_pen = True, and initially set beta = 0.5. With the given hyperparameters, I get the Figure 5. Some points of experiment, it appears higher values of average score than the experiment with the given values at Task 1.

Figure 5.

2) Task 2.3

Even though I do the longer experiments, the results in Figure 6 seems to be not converged (unstable). I think that this is because of the effect of value beta, so I do with smaller value to deal with unstable rewards.

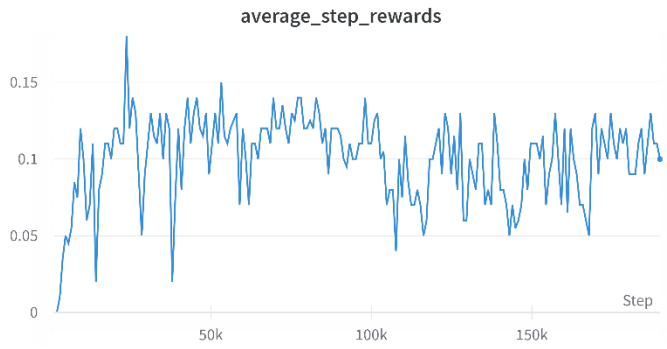

Figure 6.

Although I reduced the value of beta to 0.01, in Figure 7, it seems still unstable.

Figure 7.

So, I tried the deeper network with hidden_size as 1024 for 75k steps, but its result looks unstable, and the rewards decreases after some steps in Figure 8. To figure out this the problem with changing the size of the network, I set the hidden_size as 256 with beta = 0.5 and do the experiment again. Then, I got much better result of rewards in Figure 9. However, it cannot solve the problem of large change of policy loss. In the training, the policy loss sometimes decreases so rapidly to -30.

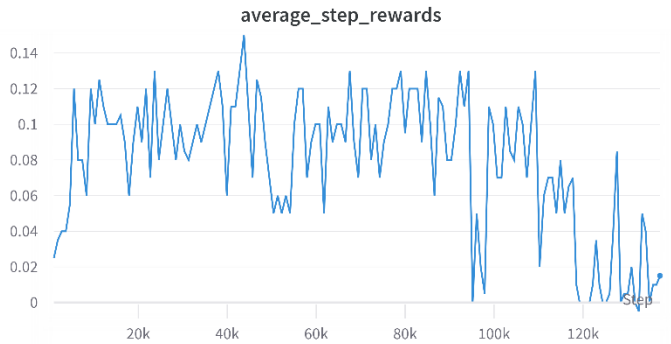

Figure 8.

Figure 9.

3) Task 2.4

For the results of my experiment, I think that the using the clipped version is better than the KL-penalty version. First, the clipped version controls the large policy change if it is outside the range as I learned in the lecture. Also, as there is one more additional hyperparameter called beta in the KL-penalty version which makes me harder to optimize parameters in the training. As a result, KL-penalty version can update the policy large which can make the training unstable at some points, and it can be resolved with the clipping version.