20233508 DoHyoung Lee

Term project 3

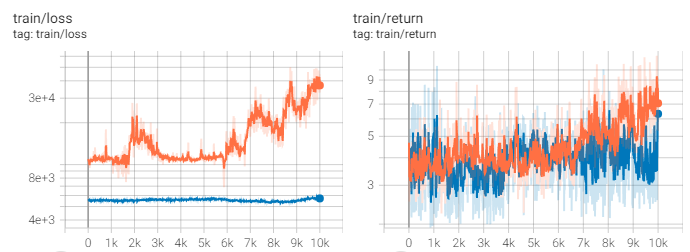
1. **Explanation of Algorithm**

Q Actor-Critic algorhtm (QAC, A. Barto, R. Sutton, 1983) evaluates the action agent did using q(s, a) function, expected return when the agent choose action a in state s. This q value role as a critic when the policy network update its parameter through gradient descent method, and q value also be evaluated and updated through TD(0) method.

In this project, there are two networks: policy network and critic network. Both have same number of layers, and same number of neurons in each layer. In one epoch, 1000 steps are operated and 24-point observation, 6-point action and reward information is saved in trajectory. Policy network receives observation information(1000 \* 24), and returns action tensor(1000 \* 6). The policy network and critic network has only varies on the initialization, but policy network weights are updated based on the gradient of absolute value of (log probability \* critic network output), where the weights of critic network are updated by mean square loss of TD error(reward + discount \* q values of next observation – q values of current observation).

1. **Plot of Return / Loss**

Figure Train Loss(Left) and Return(Right) of REINFORCE(Orange) and Q-based Actor Critic(Blue) Agent

Compare with the sample REINFORCE algorithm, QAC agent showed higher variance and lower performance during training. In contrast, the amount of loss was lower than that of REINFORCE algorithm, as was the variance of loss.

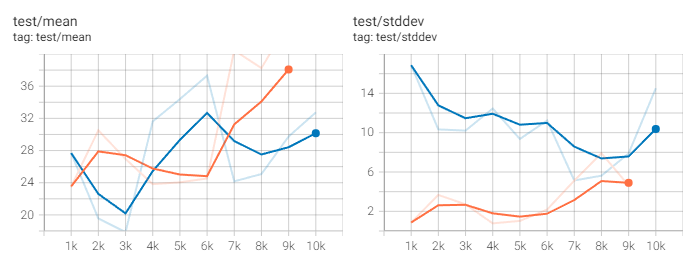
In test cases, however, the mean return of QAC showed increasing tendency. Also, the standard deviation of QAC showed decreasing tendency even though the absolute value was larger than that of REINFORCE.

Figure Test Mean(Left) and Standard Deviation(Right) of REINFORCE(Orange) and Q-based Actor Critic(Blue) Agent

1. **Analysis**

In the training sequence, the return was increased very slowly. I guess it is due to insufficient fine tuning.

1. **Hyperparameters**

asdf