Programming Assignment1

Exec Edu - Advanced Topics in Machine Learning and Game Theory

Gyuseok Lee

1. Introduction

This assignment is designed for how to apply the Proximal Policy Optimization well to Hanab i. Hanabi is one of the famous games with multi-players, so it would be trained and evaluate d through the multi-Agent Reinforcement Learning. Therefore, our task is how to well apply t he MARL to Hananbi. Note that is it the cooperative problems based on actor-critic and share d network, and each actor is trained by local observation, but the critic is trained by global st ate which is all gathered information from each actor. In my method, I used two different los s function: PPO-Clip and KL-penalty coefficient. In the rest part, I would like to explain the method, experiment, and conclusion in the order.

2. Method

As I mentioned above, there are two different methods for our task. One is the PPO-Clip and the other thing is KL-penalty coefficient. Each formula is like below. By using these each fo rmula, I can update the policy through different methods.

$$\frac{1}{|M_i|} \sum_{(s_t,a_t,r_t)\in M_i} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t,a_t), \mathrm{clip}(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}, 1-\epsilon, 1+\epsilon) A^{\pi_{\theta_k}}(s_t,a_t) \right)$$

Formula1 Loss for PPO-Clip

$$\frac{1}{|M_i|} \sum_{(s_t,a_t,r_t)\in M_i} (\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t,a_t) - \beta \ \mathrm{KL}\ [\pi_{\theta_k}(a_t|s_t), \pi_\theta(a_t|s_t)]$$

Formula2 Loss for KL-penalty

3. Experiment

In this experiment, I would like to apply each method to Hanabi. Specifically, I take the hyper parameter tuning. For PPO-clip, I set the the number of env steps **1,000,000** and **2,000,000.** As you can see the below figure, the more the number of env steps, the better performance.

So, I conclude that the more number of env steps usually guarantees the better performance.

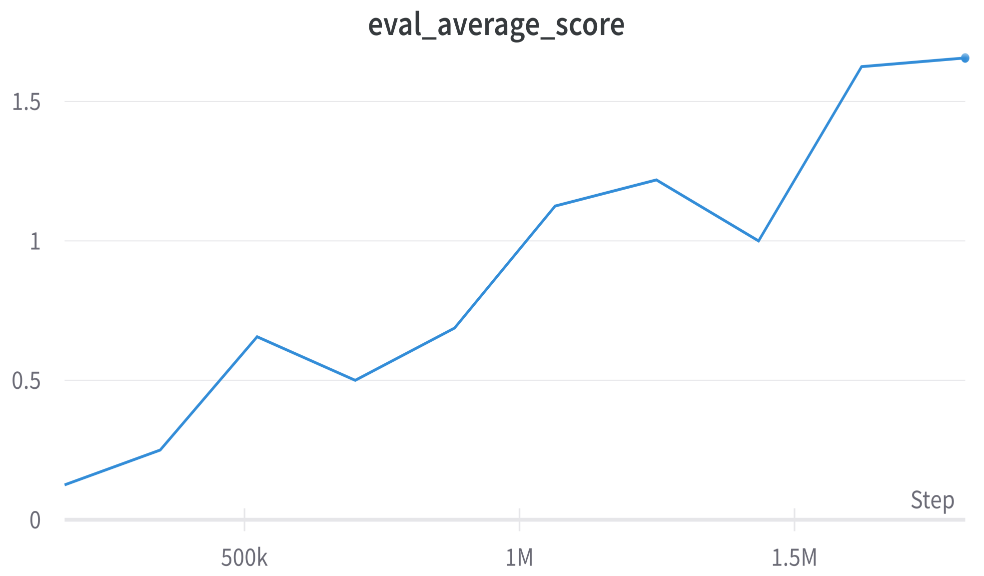

Figure 1 Evaluation for PPO-Clip (1,000,000 steps)

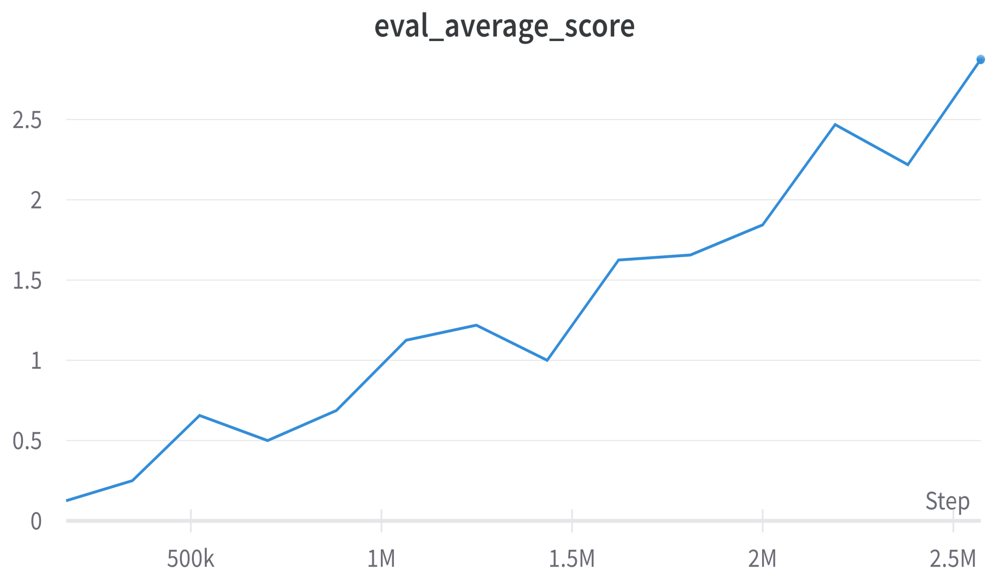

Figure 2 Evaluation for PPO-Clip (2,000,000 steps)

For KL-penalty coefficient, I set the Beta as **0.5 and 0.2** As you can see, I think that KL-divergence has unstable performance. Plus, when I used 0.5 and 0.2 setting, there are little different results. I think that hyperparameter setting for beta is very important to use KL-divergence coefficient setting, but its performance is lower than PPO-clip.

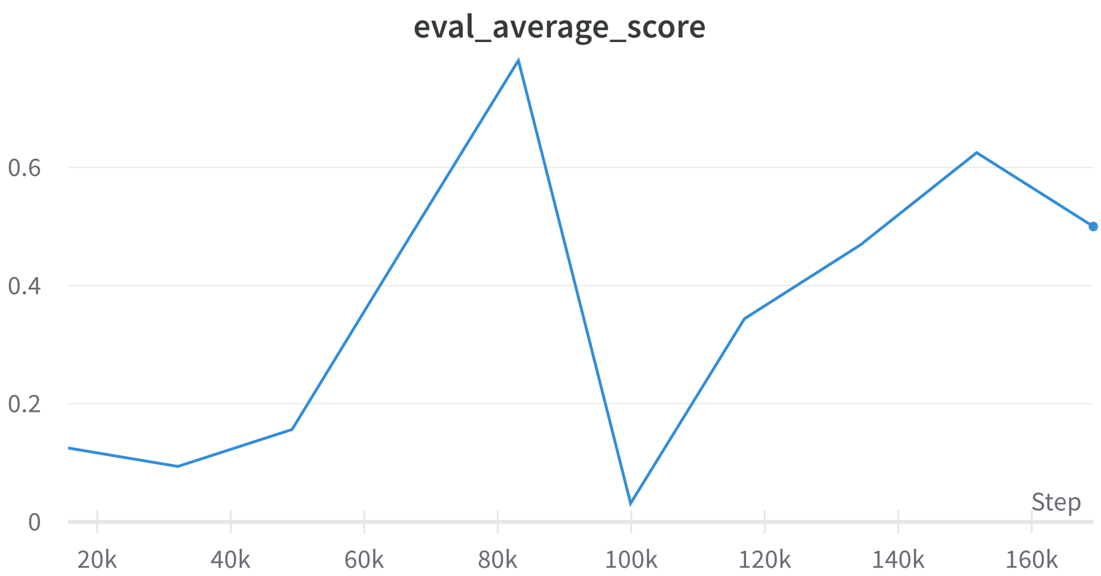

Figure 3 Evaluation for KL-penalty (beta = 0.5)

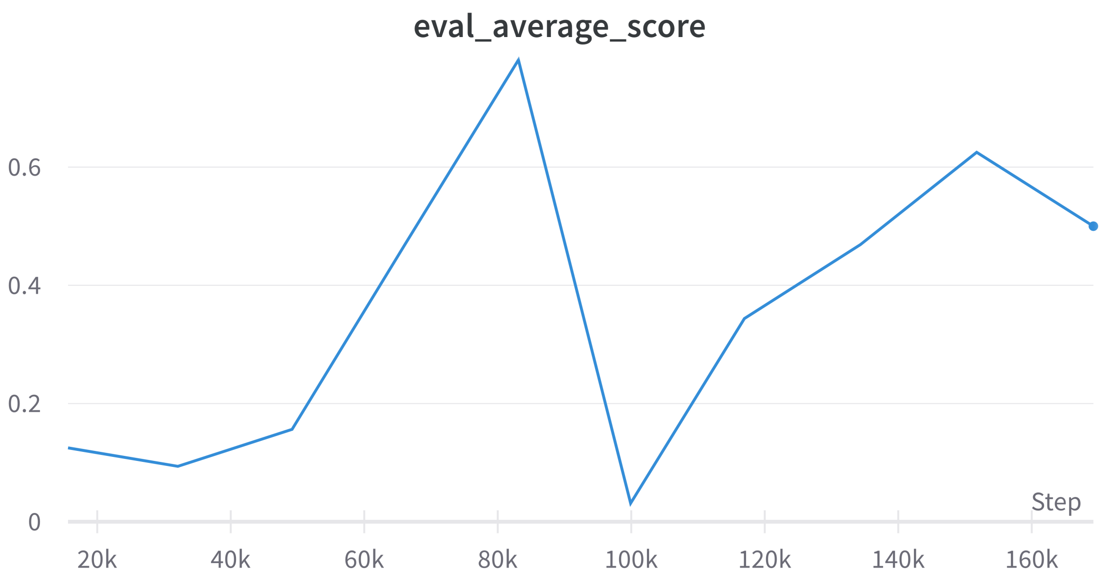

Figure 4 Evaluation for KL-penalty (beta = 0.2)

4. Conclusion

In conclusion, I can train the two different methods PPO-Clip and KL divergence penalty coefficient for Hanabi, which can be represented as Multi-Agent Reinforcement Learning based on shared Actor-Critic structure. As a result, PPO-Clip has the better performance than KL-divergence coefficient. In the future work, I would like to apply various algorithm like COMA and Q-Mix, which is one of the best credit assignment algorithms in RL.