Report of Continuous Control Project

1 Problems

In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. Thus, the goal of your agent is to maintain its position at the target location for as many time steps as possible. There exists 20 identical agents, each with its own copy of the environment.

The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

The environment is considered solved, when the average (over 100 episodes) of accumulated scores across the 20 agents is at least +30.

2 Learning Algorithm

2.1 Proximal Policy Optimization

Proximal Policy Optimization (Schulman *et al.*, 2017), also known as PPO, is employed to solve this problem. PPO is a policy gradient algorithm, and aims to update the parameterized policies by taking the largest step possible to improve performance while keep old and new policies close enough. Specifically, the objective function to maximize during each iteration is:

$$L(\theta_i) = \hat{\mathbb{E}}_t \left[\min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t, \text{clip} \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right]$$

where $\hat{A}_t$ is the estimated advantage function, which is calculated using Generalized Advantage Estimation (Schulman *et al.*, 2015), also known as GAE.

Thus PPO can be regarded as a actor-critic type algorithm. To enable better exploration, the policy entropy is added as a regularization term in the objective function.

2.2 neural Network Architecture

A three-layer fully connected neural network is employed to estimate the parameterized policy and state-value function at the same time. The first two hidden layers are with 128 and 64 neurons respectively. For policy (actor), the last layer is a fully connected layer with 4 neurons. For value-function (critic), the last layer is a fully connected layer with 1 neuron.

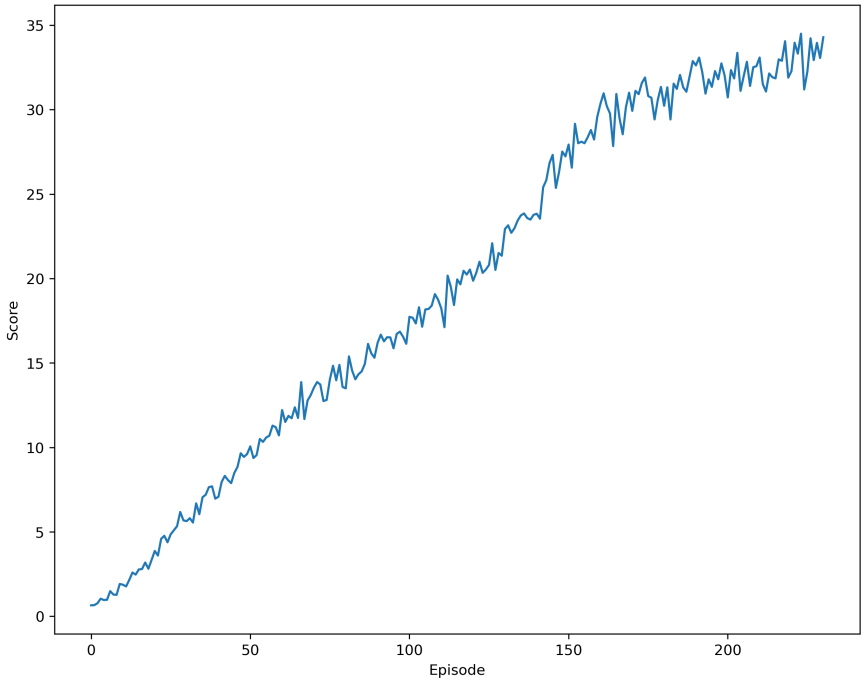

Figure 1: Reward plot of PPO

2.3 Hyperparameters

- Discount factor: 0.99
- Learning rate: 1e-4
- Batch size: 64
- Use collected trajectories 5 times for parameter updates
- Clip parameter: 0.2
- Lambda parameter for GAE: 0.95
- Coefficient for entropy term in objective function: 0.001

3 Results

The plot of average (across all 20 agents) accumulated rewards per episode is shown in Figure 1. In total, 300 episodes are run.

3.1 Summary

- PPO need 230 episodes to solve the problem.
- PPO is relatively computational efficient, and the optimization procedure takes about 20 minutes in CPU setting.

4 Future Work

- Try Deep Deterministic Policy Gradient algorithm (Lillicrap *et al.*, 2015).
- Try other more recent proposed methods.

References

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.