DRLND 2 Project Report

Ariel Kwiatkowski

May 2020

1 Methods and results

The approach I used is the PPO algorithm using an MLP model as the policy and value networks. Including the input and output sizes, the neural network layers are: 33, 512, 512 (followed by 4 for the policy network, and 1 for the value network). The network uses ReLU activations in all intermediate layers. It's optimized using the Adam algorithm with the learning rate $3 \cdot 10^{-4}$, performing updates in 64 minibatches across a batch of 2048 transitions each step. PPO performs 10 full sweeps through the entire batch.

The process can be summarized as follows:

1. Gather a batch of 2048 environment transitions
2. Compute discounted rewards to go with TD estimation, and advantages using GAE
3. Perform 10 runs over the that data divided in 64 minibatches, performing a gradient update each time

The graph of rewards can be seen in Figure 1, aiming for the target of 30 average reward across 100 consecutive episodes. This objective has been achieved after approximately 2000 iterations.

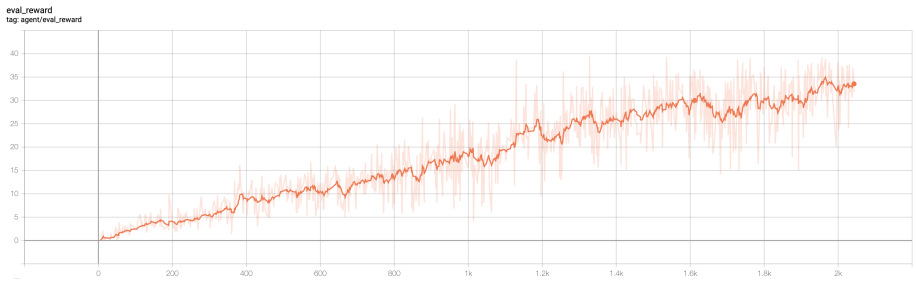

Figure 1: Rewards obtained by the agent, along with a smoothed out line (taken from TensorBoard)

2 Possible improvements

The first thing that comes to mind is a more extensive hyperparameter search – reinforcement learning in general is quite sensitive to that, so it's possible that e.g. a deeper network or a larger learning rate would perform better. For better training speed, the distributed environment could be beneficial for faster collection of experience. Also, some sort of Monte Carlo return estimation rather than TD might perform better.