Continuous Control Project

Troy Chevalier

November 25, 2018

The project uses the Unity Reacher environment. 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 the agent is to maintain its position at the target location for as many time steps as possible.

Learning algorithm

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.

This project uses the distributed version that contains 20 identical agents, each with its own copy of the environment. The environment is considered solved when the average (over the last 100 episodes) score across the agents is at least 30.

Model architecture

The solution uses a feedforward neural network to implement the multivariate Gaussian policy. The network has two hidden layers each with 64 parameters. The outputs correspond to the means of the multivariate Gaussian distribution. A fixed variance was used. The network architecture is depicted in Fig. 1.

Training

The proximal policy optimization (PPO) algorithm [Schulman et al., 2017] is used to optimize the neural network. The algorithm alternates between sampling data through interaction with the environment and optimizing a "surrogate" objective function. See [Schulman et al., 2015a] for details about stochastic computation graphs.

The clipped surrogate function is

$$L^{CLIP}(\theta) = min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$$

where $R_t(\theta)$ denotes the probability ratio of the current and old policy.

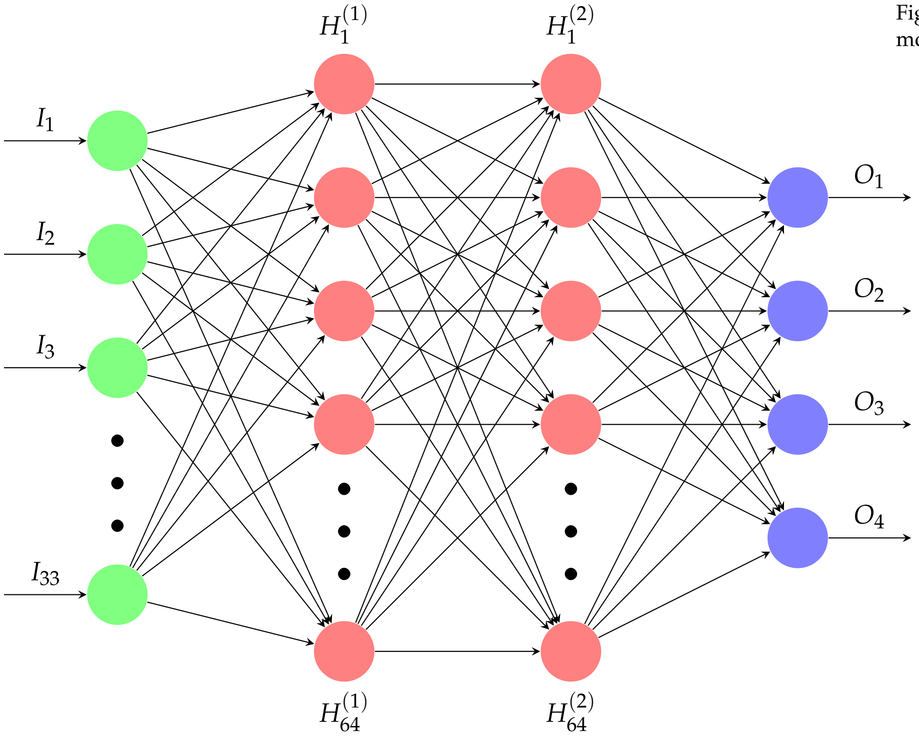

Figure 1: Feedforward network used to model the policy.

Trajectory segments of length 120 timesteps were used. Seven epochs of optimization are performed on the sampled data with minibatch sizes of 800.[1] The loss function combines the policy surrogate with an entropy bonus to ensure sufficient exploration as suggested in [Mnih et al., 2016].

The plot of rewards per episode is shown in Fig. 2.

Hyperparameters

With PPO there are many hyperparameters to tune. These include the following: (1) the clipping parameter ϵ, (2) the horizon (trajectory timesteps), (3) the number of epochs, and (4) the minibatch size.

For the clipping parameter, I tried values of 0.1 and 0.2, and I found that 0.1 worked best. The value is linearly annealed over the course of training as suggested in the PPO paper. I tried horizon lengths from 100 to 150 and the number of epochs from 3 to 8.

For the entropy coefficient, c_2, which acts as a regularizer, I used the value of 0.01 suggested in the PPO paper.

[1] The sampled data is shuffled after each epoch. The discounted future rewards are normalized (standard practice) to reduce variance.

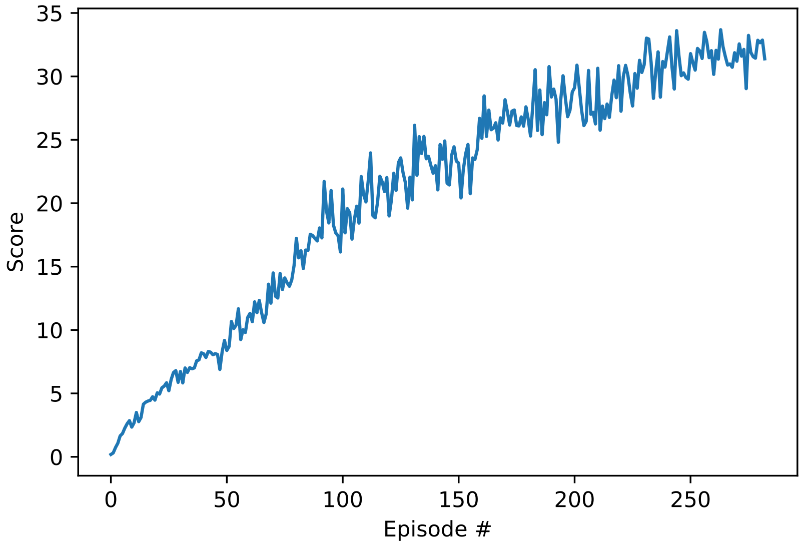

Figure 2: Rewards per episode.

Future work

Variance is a challenge when using policy gradient methods. I experimented with, but did not incorporate, the use of a "baseline" for this project. For future work I would implement a generalized advantage estimation (GAE) as described in [Schulman et al., 2015b].

References

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3528–3536. Curran Associates, Inc., 2015a. URL `http://papers.nips.cc/paper/5899-gradient-estimation-using-stochastic-computation-graphs.pdf`.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.