Report for Project 2 – Continuous Control

Problem

In this project, the goal is to train 20 agents with a double-jointed arm to move to target locations. At each time step, if the agent's hand is in the goal location, a reward of +0.1 is provided. The goal of each agent is learning a strategy to keep the hand position within the target location for as many time steps as possible. The state observed at each time step is 33 dimensional, 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. The value of each entry of the vector is continuous, ranging from -1 to +1. The criteria for solving the problem is to achieve +30 scores averaged over the latest 100 episodes and all agents.

Approach

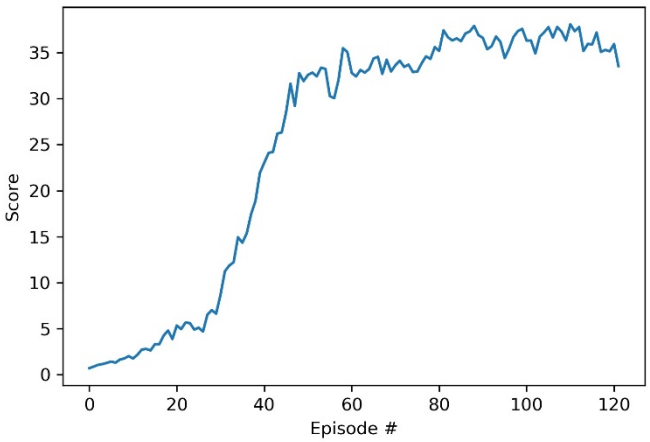
To solve the problem, I use a Deep Deterministic Policy Gradient (DDPG) method that utilizes two neural networks to learn the optimal policy and action value function, which has been successful in continuous control tasks [1]. One network is called the actor that evaluates a deterministic policy based on the current state. The other network is called the critic that takes the state as input along with the action from the actor to evaluate the action value function. Then the two networks are updated using the computed gradients. To improve learning, 20 agents are initialized that simultaneously interacts with the environment and the experience of each agent is pushed to the replay buffer which are used to update the networks. In this project, we largely adopt the above approach and tuned the hyper-parameters to achieve good performance.

The actor neural network has 2 hidden linear layers and one output layer. The numbers of hidden neurons in each layer are 512 and 256. The final output layer has 4 neurons. The output of each hidden layer is followed by Relu nonlinearity, whereas the output of the final output layer is followed by tanh function. In addition, the action value is the output value added by an Ornstein–Uhlenbeck noise, which allows for exploration. The critic neural network has two hidden layers and one output layer. The first hidden layer has 512 neurons and takes the state as input. The second hidden neuron has 256 neurons that take as input from both the actor neural network and the first hidden layer. The Relu nonlinearity is applied to each hidden layer.

To train the deep neural network, I implement the algorithm described in [1]. The hyper parameters are chosen as the following. The learning rates for both actor and critic networks are 1e-3. The reward decay factor γ=0.99. The weights of the actor and critic networks are soft-updated 10 times for every 20 time steps. The coefficient for the soft-updated is 1e-3. The replay buffer size is 1e5 and the batch size is chosen as 512.

Result

By implementing the algorithm described above, I was able to solve the problem in ~50 episodes. The scores are plotted as a function of the episodes, as shown in the figure below.



Future directions

In the future, the recent Distributed Distributional Deterministic Policy Gradients (D4PG) [2] can be tried to further improve the performance of the agent. It has been demonstrated on many continuous control tasks and achieved the state of the art performance.

Reference

1. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv:150902971. 2015.
2. Barth-Maron G, Hoffman MW, Budden D, Dabney W, Horgan D, Muldal A, et al. Distributed Distributional Deterministic Policy Gradients. arXiv preprint arXiv:180408617. 2018.