Deep Reinforcement Learning Nanodegree

Project 2: Continuous Control

Learning Algorithm

1. Algorithm

   I use DDPG to solve this project. In DDPG, we use two deep neural networks. We can call one the actor and the other the critic. The actor is used to approximate the optimal policy deterministically. The critic learns to evaluate the optimal action value function by using the actors best believed action. Also in DDPG, we use a replay buffer and soft updates. you have two copies of your network weight, the regular and the target network. A soft update strategy consists of slowly blending your regular network weights with your target network weights.

   I add mean-zero Gaussian noise to actions at training time and I reduce the scale of the noise over the course of training.

2. Hyperparameters

   I used the following hyperparameters in my code:

   - BUFFER_SIZE = int(1e6) # replay buffer size
   - GAMMA = 0.99 # discount factor
   - TAU = 1e-3 # for soft update of target
   - LR_ACTOR = 1e-4 # learning rate of the actor
   - LR_CRITIC = 1e-3 # learning rate of the critic
   - WEIGHT_DECAY = 0 # L2 weight decay
   - UPDATE_EVERY = 20 # how often to update the network
   - BATCH_SIZE = 640 # minibatch size
   - NUM_OF_UPDATES = 1 # how many times to update

3. Model architectures

The architecture of Actor-Critic and the number of nodes in each layer is as follows:

Actor

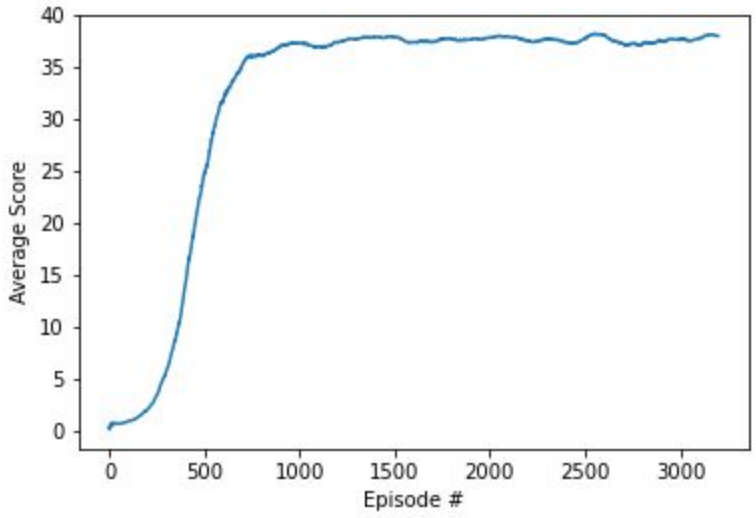
- Input layer = 33 (state_size)
- Hidden layer 1 (relu) = 128
- Hidden layer 2 (relu) = 64
- Output layer (tanh) = 4 (action_size)

Critic

- Input layer = 33 (state_size)
- Hidden layer 1 (relu) = 128
- Concatenate = 132 (layer 1 size + action size)
- Hidden layer 2 (relu) = 64
- Output layer = 1

Plot of Rewards

The reward graph is shown below. My agent got an average score (over 100 episodes) of +30 in about 600 episodes.



Ideas for Future work

1. Trust Region Policy Optimization (TRPO)
2. Truncated Natural Policy Gradient (TNPG)
3. Proximal Policy Optimization (PPO)
4. Distributed Distributional Deterministic Policy Gradients (D4PG)