22.03.2020

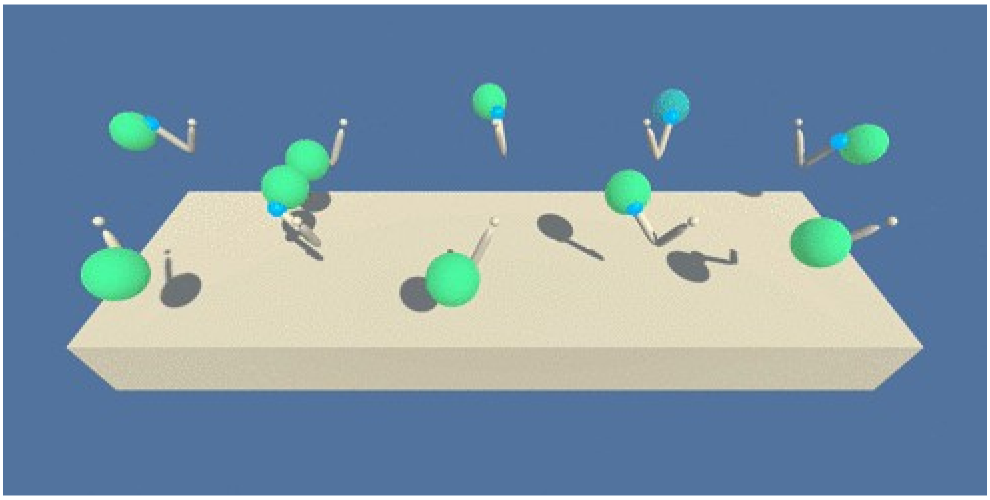
Udacity Nanodegree Deep Reinforcement Learning
Sören Klingner Report

Continuous Control

1. Introduction

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.

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 must be a number between -1 and 1.



I choose the second version, thus the goal is to train 20 agents to get an average score of +30 (over 100 consecutive episodes, and over all agents).

2. Algorithm

For this project the Deep Deterministic Policy Gradient (DDPG) algorithm was selected in order to solve the environment. This algorithm combines the actor-critic approach with the Deep Q Network algorithm using Deep Neural Networks for approximating the actor and the critic which works great in continuous action spaces environments.

3. Architecture & Hyperparameters

The Actor uses the input with no preprocessing and outputs 4 values that are the size of the action space. The Critic is using compute advantages state value.

The Actor network has an initial dimension with the same size as the state space and it uses two fully connected layers with 256 and 128 units. As an activation function relu is used and for the action space tanh is used.

The Critic network uses the same number of units per layer, but it uses a leaky_relu activation. The initial dimensions is based on the initial state size plus the action size.

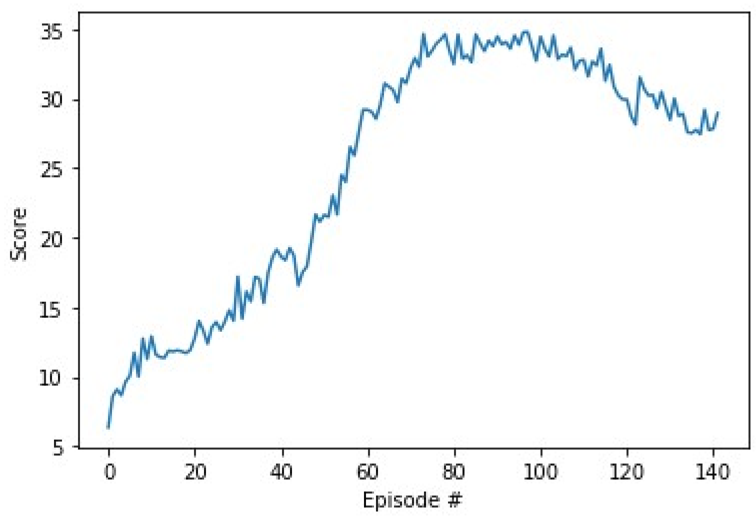
Hyperparameters

```
BUFFER_SIZE = int(1e6)  # replay buffer size
BATCH_SIZE = 128        # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR_ACTOR = 1e-4         # learning rate of the actor
LR_CRITIC = 1e-4        # learning rate of the critic
WEIGHT_DECAY = 0.0      # L2 weight decay

N_LEARN_UPDATES = 10    # number of learning updates
N_TIME_STEPS = 20       # every n time step do update

n_episodes=300          # maximum number of training episodes
max_t=1000              # maximum number of timesteps per episode

fc1_units=256           # units in first hidden layer
fc2_units=128           # units in second hidden layer
```

4. Results

Given the chosen architecture and parameters, our results are:

```
Episode: 1      Score: 6.34     Average Score: 6.34.73
Episode: 10     Score: 11.27    Average Score: 9.8115
Episode: 20     Score: 11.93    Average Score: 10.822
Episode: 30     Score: 14.03    Average Score: 11.757
Episode: 40     Score: 19.12    Average Score: 13.0026
Episode: 50     Score: 21.13    Average Score: 14.1955
Episode: 60     Score: 29.19    Average Score: 15.9232
Episode: 70     Score: 31.11    Average Score: 17.9589
Episode: 80     Score: 33.46    Average Score: 19.8930
Episode: 90     Score: 33.78    Average Score: 21.4182
Episode: 100    Score: 32.72    Average Score: 22.671
Episode: 110    Score: 32.72    Average Score: 25.032
Episode: 120    Score: 29.94    Average Score: 27.029
Episode: 130    Score: 29.43    Average Score: 28.659
Episode: 140    Score: 27.73    Average Score: 29.817
Episode: 141    Score: 27.83    Average Score: 29.900
Episode: 142    Score: 28.94    Average Score: 30.005

Environment solved in 42 episodes. Average Score: 30.00 over 100
episodes.
```



These results meets the project's requirements as the agent is able to receive an average score of at least +30 over the last 100 episodes, from episode 42 to 142.

5. Ideas for Future Work

Further improvements:

- Tune hyperparameters with automated grid search
- Prioritized experience replay

For further exploration to check for improved performance, Proximal Policy Optimization (PPO) and Distributed Distributional Deterministic Policy Gradients (D4PG) methods could be implemented.