Continuous_Control

May 25, 2020

1 Continuous Control

In this notebook, we use the Unity ML-Agents environment for the second project of the Deep Reinforcement Learning Nanodegree program.

1.0.1 1. Start the Environment

We begin by importing the necessary packages. If the code cell below returns an error, please revisit the project instructions to double-check that you have installed Unity ML-Agents and NumPy.

```
In [1]: # when running in Udacity workspace
        !pip -q install ./python
```

```
tensorflow 1.7.1 has requirement numpy>=1.13.3, but you'll have numpy 1.12.1 which is incompatib
ipython 6.5.0 has requirement prompt-toolkit<2.0.0,>=1.0.15, but you'll have prompt-toolkit 3.0.
```

```
In [2]: from ddpg_agent import Agent
        from collections import deque
        import matplotlib.pyplot as plt
        import numpy as np
        import random
        import torch
        from unityagents import UnityEnvironment

        %matplotlib inline
```

Next, we will start the environment! ***Before running the code cell below,*** change the `file_name` parameter to match the location of the Unity environment that you downloaded.

- **Mac**: `"path/to/Reacher.app"`
- **Windows** (x86): `"path/to/Reacher_Windows_x86/Reacher.exe"`
- **Windows** (x86_64): `"path/to/Reacher_Windows_x86_64/Reacher.exe"`
- **Linux** (x86): `"path/to/Reacher_Linux/Reacher.x86"`
- **Linux** (x86_64): `"path/to/Reacher_Linux/Reacher.x86_64"`

- **Linux** (x86, headless): `"path/to/Reacher_Linux_NoVis/Reacher.x86"`
- **Linux** (x86_64, headless): `"path/to/Reacher_Linux_NoVis/Reacher.x86_64"`

For instance, if you are using a Mac, then you downloaded `Reacher.app`. If this file is in the same folder as the notebook, then the line below should appear as follows:

```
env = UnityEnvironment(file_name="Reacher.app")
```

```
In [3]: # select this option to load version 1 (with a single agent) of the environment
        env = UnityEnvironment(file_name='/data/Reacher_One_Linux_NoVis/Reacher_One_Linux_NoVis.

        # select this option to load version 2 (with 20 agents) of the environment
        # env = UnityEnvironment(file_name='/data/Reacher_Linux_NoVis/Reacher.x86_64')
```

```
INFO:unityagents:
'Academy' started successfully!
Unity Academy name: Academy
        Number of Brains: 1
        Number of External Brains : 1
        Lesson number : 0
        Reset Parameters :
                goal_speed -> 1.0
                goal_size -> 5.0
Unity brain name: ReacherBrain
        Number of Visual Observations (per agent): 0
        Vector Observation space type: continuous
        Vector Observation space size (per agent): 33
        Number of stacked Vector Observation: 1
        Vector Action space type: continuous
        Vector Action space size (per agent): 4
        Vector Action descriptions: , , ,
```

Environments contain ***brains*** which are responsible for deciding the actions of their associated agents. Here we check for the first brain available, and set it as the default brain we will be controlling from Python.

```
In [4]: # get the default brain
        brain_name = env.brain_names[0]
        brain = env.brains[brain_name]
```

1.0.2 2. The State and Action Spaces

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`.

1.0.3 3. DDPG Model

When training the environment, set `train_mode=True`, so that the line for resetting the environment looks like the following:

```
env_info = env.reset(train_mode=True)[brain_name]
```

```
In [6]: def ddpg(n_episodes=200, max_t=1000, solved_score=30.0, consec_episodes=100, print_every
                 actor_path='actor_ckpt.pth', critic_path='critic_ckpt.pth'):
            """Deep Deterministic Policy Gradient (DDPG)

            Params
            ======
                n_episodes (int)      : maximum number of training episodes
                max_t (int)           : maximum number of timesteps per episode
                train_mode (bool)     : if 'True' set environment to training mode
                solved_score (float)  : min avg score over consecutive episodes
                consec_episodes (int) : number of consecutive episodes used to calculate score
                print_every (int)     : interval to display results
                actor_path (str)      : directory to store actor network weights
                critic_path (str)     : directory to store critic network weights

            """
            instant_scores = []
            scores_window = deque(maxlen=consec_episodes)  # mean scores from most recent episod
            moving_avgs = []                               # list of moving averages

            for i_episode in range(1, n_episodes+1):
                env_info = env.reset(train_mode=train_mode)[brain_name] # reset environment
                states = env_info.vector_observations                   # get current state for
                scores = np.zeros(num_agents)                           # initialize score for e
                agent.reset()
                for t in range(max_t):
                    actions = agent.act(states, add_noise=True)         # select an action
                    env_info = env.step(actions)[brain_name]            # send actions to enviro
                    next_states = env_info.vector_observations          # get next state
                    rewards = env_info.rewards                          # get reward
                    dones = env_info.local_done                         # see if episode has fin
                    for state, action, reward, next_state, done in zip(states, actions, rewards,
                        agent.step(state, action, reward, next_state, done, t)
                    states = next_states
                    scores += rewards
                    if np.any(dones):                                   # exit loop when episode
                        break

                instant_scores.append(np.mean(scores))            # save mean score for the episo
                scores_window.append(instant_scores[-1])          # save mean score to window
                moving_avgs.append(np.mean(scores_window))    # save moving average
```

```
            if i_episode % print_every == 0:
                print('\rEpisode {} \tInst. Reward: {:.1f}\tAvg. Reward: {:.1f}'.format(\
                      i_episode, instant_scores[-1], moving_avgs[-1]))

            if moving_avgs[-1] >= solved_score and i_episode >= consec_episodes:
                print('\nEnvironment SOLVED in {} episodes!\tMoving Average ={:.1f} over las
                                        i_episode-consec_episodes, moving_avgs[-1], consec_e
                if train_mode:
                    torch.save(agent.actor_local.state_dict(), actor_path)
                    torch.save(agent.critic_local.state_dict(), critic_path)
                break

        return instant_scores, moving_avgs
```

In [7]:
```
# run the training loop
agent = Agent(state_size=state_size, action_size=action_size, random_seed=1)
scores, avgs = ddpg()
```

```
Episode 1         Inst. Reward: 1.1        Avg. Reward: 1.1
Episode 2         Inst. Reward: 0.9        Avg. Reward: 1.0
Episode 3         Inst. Reward: 1.0        Avg. Reward: 1.0
Episode 4         Inst. Reward: 1.1        Avg. Reward: 1.0
Episode 5         Inst. Reward: 1.6        Avg. Reward: 1.1
Episode 6         Inst. Reward: 0.2        Avg. Reward: 1.0
Episode 7         Inst. Reward: 1.7        Avg. Reward: 1.1
Episode 8         Inst. Reward: 0.3        Avg. Reward: 1.0
Episode 9         Inst. Reward: 0.9        Avg. Reward: 1.0
Episode 10         Inst. Reward: 0.1        Avg. Reward: 0.9
Episode 11         Inst. Reward: 0.9        Avg. Reward: 0.9
Episode 12         Inst. Reward: 0.2        Avg. Reward: 0.9
Episode 13         Inst. Reward: 0.5        Avg. Reward: 0.8
Episode 14         Inst. Reward: 0.2        Avg. Reward: 0.8
Episode 15         Inst. Reward: 1.3        Avg. Reward: 0.8
Episode 16         Inst. Reward: 1.2        Avg. Reward: 0.8
Episode 17         Inst. Reward: 1.7        Avg. Reward: 0.9
Episode 18         Inst. Reward: 1.4        Avg. Reward: 0.9
Episode 19         Inst. Reward: 2.2        Avg. Reward: 1.0
Episode 20         Inst. Reward: 0.9        Avg. Reward: 1.0
Episode 21         Inst. Reward: 2.1        Avg. Reward: 1.0
Episode 22         Inst. Reward: 1.5        Avg. Reward: 1.1
Episode 23         Inst. Reward: 1.8        Avg. Reward: 1.1
Episode 24         Inst. Reward: 1.0        Avg. Reward: 1.1
Episode 25         Inst. Reward: 5.7        Avg. Reward: 1.3
Episode 26         Inst. Reward: 3.1        Avg. Reward: 1.3
Episode 27         Inst. Reward: 2.1        Avg. Reward: 1.4
Episode 28         Inst. Reward: 3.1        Avg. Reward: 1.4
Episode 29         Inst. Reward: 3.8        Avg. Reward: 1.5
Episode 30         Inst. Reward: 1.1        Avg. Reward: 1.5
```

```
Episode 31        Inst. Reward: 5.2        Avg. Reward: 1.6
Episode 32        Inst. Reward: 4.1        Avg. Reward: 1.7
Episode 33        Inst. Reward: 2.3        Avg. Reward: 1.7
Episode 34        Inst. Reward: 3.1        Avg. Reward: 1.8
Episode 35        Inst. Reward: 3.7        Avg. Reward: 1.8
Episode 36        Inst. Reward: 3.2        Avg. Reward: 1.8
Episode 37        Inst. Reward: 2.7        Avg. Reward: 1.9
Episode 38        Inst. Reward: 3.1        Avg. Reward: 1.9
Episode 39        Inst. Reward: 2.0        Avg. Reward: 1.9
Episode 40        Inst. Reward: 3.8        Avg. Reward: 2.0
Episode 41        Inst. Reward: 3.7        Avg. Reward: 2.0
Episode 42        Inst. Reward: 3.2        Avg. Reward: 2.0
Episode 43        Inst. Reward: 8.1        Avg. Reward: 2.2
Episode 44        Inst. Reward: 5.0        Avg. Reward: 2.2
Episode 45        Inst. Reward: 2.7        Avg. Reward: 2.2
Episode 46        Inst. Reward: 1.8        Avg. Reward: 2.2
Episode 47        Inst. Reward: 3.8        Avg. Reward: 2.3
Episode 48        Inst. Reward: 11.9        Avg. Reward: 2.5
Episode 49        Inst. Reward: 13.1        Avg. Reward: 2.7
Episode 50        Inst. Reward: 5.2        Avg. Reward: 2.7
Episode 51        Inst. Reward: 12.1        Avg. Reward: 2.9
Episode 52        Inst. Reward: 17.5        Avg. Reward: 3.2
Episode 53        Inst. Reward: 20.4        Avg. Reward: 3.5
Episode 54        Inst. Reward: 7.3        Avg. Reward: 3.6
Episode 55        Inst. Reward: 15.0        Avg. Reward: 3.8
Episode 56        Inst. Reward: 18.2        Avg. Reward: 4.1
Episode 57        Inst. Reward: 20.6        Avg. Reward: 4.3
Episode 58        Inst. Reward: 18.4        Avg. Reward: 4.6
Episode 59        Inst. Reward: 5.1        Avg. Reward: 4.6
Episode 60        Inst. Reward: 20.6        Avg. Reward: 4.9
Episode 61        Inst. Reward: 23.8        Avg. Reward: 5.2
Episode 62        Inst. Reward: 11.7        Avg. Reward: 5.3
Episode 63        Inst. Reward: 27.8        Avg. Reward: 5.6
Episode 64        Inst. Reward: 18.7        Avg. Reward: 5.8
Episode 65        Inst. Reward: 24.8        Avg. Reward: 6.1
Episode 66        Inst. Reward: 13.2        Avg. Reward: 6.2
Episode 67        Inst. Reward: 5.1        Avg. Reward: 6.2
Episode 68        Inst. Reward: 22.7        Avg. Reward: 6.5
Episode 69        Inst. Reward: 35.7        Avg. Reward: 6.9
Episode 70        Inst. Reward: 23.4        Avg. Reward: 7.1
Episode 71        Inst. Reward: 23.2        Avg. Reward: 7.3
Episode 72        Inst. Reward: 27.3        Avg. Reward: 7.6
Episode 73        Inst. Reward: 34.2        Avg. Reward: 8.0
Episode 74        Inst. Reward: 23.1        Avg. Reward: 8.2
Episode 75        Inst. Reward: 7.8        Avg. Reward: 8.2
Episode 76        Inst. Reward: 27.3        Avg. Reward: 8.4
Episode 77        Inst. Reward: 26.4        Avg. Reward: 8.7
Episode 78        Inst. Reward: 26.4        Avg. Reward: 8.9
```

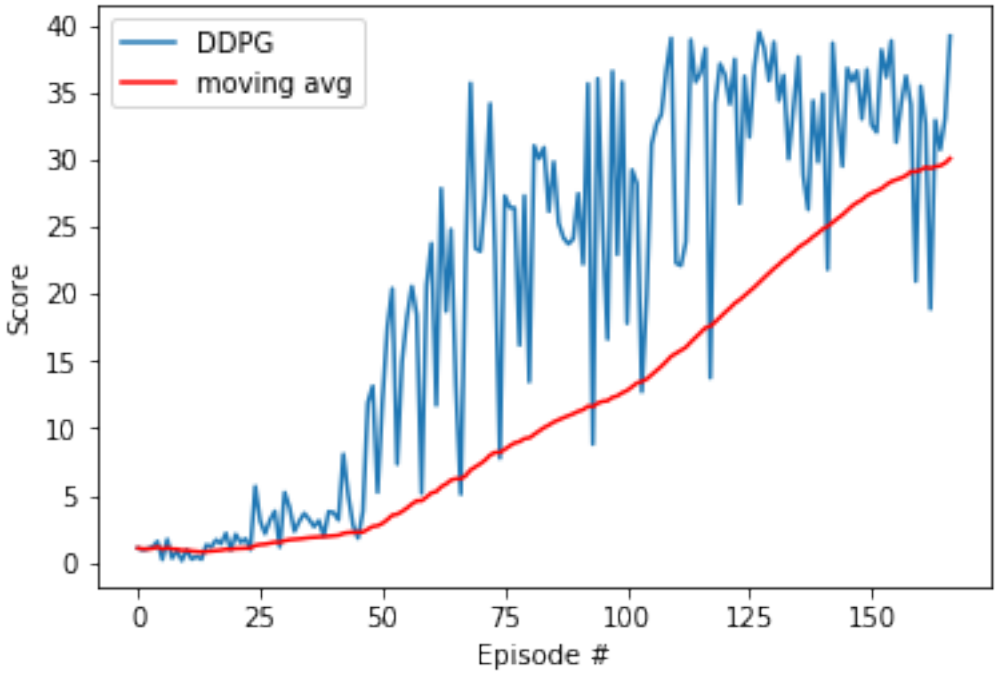
```
Episode 79        Inst. Reward: 16.2        Avg. Reward: 9.0
Episode 80        Inst. Reward: 27.3        Avg. Reward: 9.2
Episode 81        Inst. Reward: 13.4        Avg. Reward: 9.3
Episode 82        Inst. Reward: 31.0        Avg. Reward: 9.5
Episode 83        Inst. Reward: 30.1        Avg. Reward: 9.8
Episode 84        Inst. Reward: 30.9        Avg. Reward: 10.0
Episode 85        Inst. Reward: 26.1        Avg. Reward: 10.2
Episode 86        Inst. Reward: 29.8        Avg. Reward: 10.5
Episode 87        Inst. Reward: 25.3        Avg. Reward: 10.6
Episode 88        Inst. Reward: 24.2        Avg. Reward: 10.8
Episode 89        Inst. Reward: 23.7        Avg. Reward: 10.9
Episode 90        Inst. Reward: 24.1        Avg. Reward: 11.1
Episode 91        Inst. Reward: 27.5        Avg. Reward: 11.3
Episode 92        Inst. Reward: 22.2        Avg. Reward: 11.4
Episode 93        Inst. Reward: 35.6        Avg. Reward: 11.6
Episode 94        Inst. Reward: 8.8        Avg. Reward: 11.6
Episode 95        Inst. Reward: 36.0        Avg. Reward: 11.9
Episode 96        Inst. Reward: 24.9        Avg. Reward: 12.0
Episode 97        Inst. Reward: 16.6        Avg. Reward: 12.0
Episode 98        Inst. Reward: 36.6        Avg. Reward: 12.3
Episode 99        Inst. Reward: 22.9        Avg. Reward: 12.4
Episode 100        Inst. Reward: 35.8        Avg. Reward: 12.6
Episode 101        Inst. Reward: 17.8        Avg. Reward: 12.8
Episode 102        Inst. Reward: 29.2        Avg. Reward: 13.1
Episode 103        Inst. Reward: 28.2        Avg. Reward: 13.4
Episode 104        Inst. Reward: 12.7        Avg. Reward: 13.5
Episode 105        Inst. Reward: 19.9        Avg. Reward: 13.7
Episode 106        Inst. Reward: 31.1        Avg. Reward: 14.0
Episode 107        Inst. Reward: 32.6        Avg. Reward: 14.3
Episode 108        Inst. Reward: 33.3        Avg. Reward: 14.6
Episode 109        Inst. Reward: 36.6        Avg. Reward: 15.0
Episode 110        Inst. Reward: 39.0        Avg. Reward: 15.3
Episode 111        Inst. Reward: 22.3        Avg. Reward: 15.6
Episode 112        Inst. Reward: 22.1        Avg. Reward: 15.8
Episode 113        Inst. Reward: 23.9        Avg. Reward: 16.0
Episode 114        Inst. Reward: 38.9        Avg. Reward: 16.4
Episode 115        Inst. Reward: 35.8        Avg. Reward: 16.7
Episode 116        Inst. Reward: 36.4        Avg. Reward: 17.1
Episode 117        Inst. Reward: 38.3        Avg. Reward: 17.5
Episode 118        Inst. Reward: 13.7        Avg. Reward: 17.6
Episode 119        Inst. Reward: 34.1        Avg. Reward: 17.9
Episode 120        Inst. Reward: 37.1        Avg. Reward: 18.3
Episode 121        Inst. Reward: 36.3        Avg. Reward: 18.6
Episode 122        Inst. Reward: 34.1        Avg. Reward: 18.9
Episode 123        Inst. Reward: 37.5        Avg. Reward: 19.3
Episode 124        Inst. Reward: 26.7        Avg. Reward: 19.6
Episode 125        Inst. Reward: 36.2        Avg. Reward: 19.9
Episode 126        Inst. Reward: 31.7        Avg. Reward: 20.1
```

```
Episode 127         Inst. Reward: 36.8        Avg. Reward: 20.5
Episode 128         Inst. Reward: 39.5        Avg. Reward: 20.9
Episode 129         Inst. Reward: 38.3        Avg. Reward: 21.2
Episode 130         Inst. Reward: 35.9        Avg. Reward: 21.5
Episode 131         Inst. Reward: 38.7        Avg. Reward: 21.9
Episode 132         Inst. Reward: 34.4        Avg. Reward: 22.2
Episode 133         Inst. Reward: 36.3        Avg. Reward: 22.5
Episode 134         Inst. Reward: 30.0        Avg. Reward: 22.8
Episode 135         Inst. Reward: 33.8        Avg. Reward: 23.1
Episode 136         Inst. Reward: 37.6        Avg. Reward: 23.4
Episode 137         Inst. Reward: 28.8        Avg. Reward: 23.7
Episode 138         Inst. Reward: 26.3        Avg. Reward: 23.9
Episode 139         Inst. Reward: 34.4        Avg. Reward: 24.3
Episode 140         Inst. Reward: 29.8        Avg. Reward: 24.5
Episode 141         Inst. Reward: 34.9        Avg. Reward: 24.8
Episode 142         Inst. Reward: 21.8        Avg. Reward: 25.0
Episode 143         Inst. Reward: 38.7        Avg. Reward: 25.3
Episode 144         Inst. Reward: 33.6        Avg. Reward: 25.6
Episode 145         Inst. Reward: 29.5        Avg. Reward: 25.9
Episode 146         Inst. Reward: 36.8        Avg. Reward: 26.2
Episode 147         Inst. Reward: 35.8        Avg. Reward: 26.5
Episode 148         Inst. Reward: 36.6        Avg. Reward: 26.8
Episode 149         Inst. Reward: 33.0        Avg. Reward: 27.0
Episode 150         Inst. Reward: 36.7        Avg. Reward: 27.3
Episode 151         Inst. Reward: 32.6        Avg. Reward: 27.5
Episode 152         Inst. Reward: 32.0        Avg. Reward: 27.7
Episode 153         Inst. Reward: 38.2        Avg. Reward: 27.8
Episode 154         Inst. Reward: 36.1        Avg. Reward: 28.1
Episode 155         Inst. Reward: 38.8        Avg. Reward: 28.4
Episode 156         Inst. Reward: 31.3        Avg. Reward: 28.5
Episode 157         Inst. Reward: 34.1        Avg. Reward: 28.6
Episode 158         Inst. Reward: 36.2        Avg. Reward: 28.8
Episode 159         Inst. Reward: 34.1        Avg. Reward: 29.1
Episode 160         Inst. Reward: 20.9        Avg. Reward: 29.1
Episode 161         Inst. Reward: 35.5        Avg. Reward: 29.2
Episode 162         Inst. Reward: 32.6        Avg. Reward: 29.4
Episode 163         Inst. Reward: 18.9        Avg. Reward: 29.3
Episode 164         Inst. Reward: 32.9        Avg. Reward: 29.5
Episode 165         Inst. Reward: 30.7        Avg. Reward: 29.5
Episode 166         Inst. Reward: 33.1        Avg. Reward: 29.7
Episode 167         Inst. Reward: 39.2        Avg. Reward: 30.1

Environment SOLVED in 67 episodes!        Moving Average =30.1 over last 100 episodes
```

In [8]:
```
# plot the scores
fig = plt.figure()
ax = fig.add_subplot(111)
```

```
plt.plot(np.arange(len(scores)), scores, label='DDPG')
plt.plot(np.arange(len(scores)), avgs, c='r', label='moving avg')
plt.ylabel('Score')
plt.xlabel('Episode #')
plt.legend(loc='upper left');
plt.show()
```



In [9]:
```
env.close()
```