Reinforcement Learning
Exercise 10

Jim Mainprice, Philipp Kratzer, Yoojin Oh, Janik Hager

Machine Learning & Robotics lab, U Stuttgart

Universitätsstraße 38, 70569 Stuttgart, Germany

June 29, 2021

1 DQN on the Cart-Pole (7P)

In this exercise you will implement the deep Q-learning (DQN) algorithm. The code template can be found on github (`https://github.com/humans-to-robots-motion/rl-course`) in *ex10-dqn/ex10-dqn.py*. Please update the repository using "git pull". You will need tensorflow2 *python3 -m pip install tensorflow* in order to run the code. For this exercise we will again use the Cart-Pole environment from gym: `https://gym.openai.com/envs/CartPole-v1/`

a) What are the main issues of Q-Learning when approximating the Q-function with a neural network? How does DQN tackle them? (1P)

b) The code template does naive Q-Learning with a neural net. It always uses the latest (s, a, r, s') pairs to compute the loss and update the weights. Your task is to implement DQN:

- Implement a replay buffer. Store the observed pairs and randomly sample a batch for updating.
- Implement fixed target weights. Keep another network with a separate set of weights for computing the target. Only update the target weights delayed. *Hint: you can use the model.set_weights and get_weights functions*

Compare the learning performance of the naive implementation with DQN. Do this by running it for 1000 iterations and plot the episode rewards. (4P)

c) The code template uses Adam as optimizer. How does the performance change with SGD? Repeat your comparison from task b). (1P)

d) Consider the "Breakout-v0" environment: `https://gym.openai.com/envs/Breakout-v0/`. What kind of neural network architecture would you use for this environment? (1P)

2 DDPG vs TRPO (3P)

In this exercise you will compare the performance of DDPG and TRPO using standard implementations. We recommend to use the *stable-baselines* package (`https://stable-baselines.readthedocs.io/en/master/guide/quickstart.html`). It can be installed using pip: *python3 -m pip install 'stable-baselines3[mpi]'*. Care, it needs an older tensorflow version as was used in task 1 (*python3 -m pip install tensorflow==1.15.0*), you can, for example, use a python virtual environment to have both installed.

a) Run the TRPO and DDPG algorithms on the continuous mountain car environment "MountainCarContinuous-v0" (`https://gym.openai.com/envs/MountainCarContinuous-v0/`). You can try different action noise types for DDPG (e.g. Ornstein Uhlenbeck). (2P)

b) Compare the performances of the algorithms. Which algorithm performs better on this environment, why? (1P)