Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Matthias Schubert
Sebastian Schmoll

Munich, July 4th, 2018

Managing Massive Multiplayer Online Games

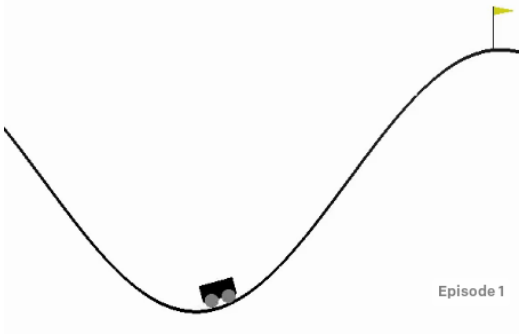
SoSe 2018

Exercise Sheet 13: Reinfocement Learning

Discussion: July 11th, 2018

Exercise 13-1 *Mountain Car* (Homework)

Mountain Car is a well known benchmark for reinforcement learning algorithms.



As you can see in the above picture, this benchmark has an environment which contains a car located in a valley between two mountains. The goal is to drive the car to the top of the right mountain. However, the car's engine is not strong enough to climb the mountain. The OpenAi Gym framework provides for the programming language Python an convenient API for this (and other) benchmarks. The source code below shows how to implement a random agent acting in this environment.

```
import gym
env = gym.make("MountainCar-v0")
observation = env.reset()
for _ in range(1000):
        env.render()
        action = env.action_space.sample() # TODO agent here (random actions)
        state, reward, done, info = env.step(action)
```

You can use the source code from the lecture side as starting point.

(a) Implement an ϵ-Greedy policy (AbstractQLearn.choose_action() in q.py) with help of the abstract get_q method.

(b) Implement the table-based q-learning algorithm that tries to optimize the cumulated rewards given by the environment (QLearn in discrete_q.py). Note that the state space is a two dimensional continuous vector. Use an discretization interval such that each dimension has 10 intervals. Train the agent for 20000 episodes and see the result.

(c) Implement the q-learning algorithm with a Gaussian radial basis function as function approximation (LinQLearn in linear_q.py). Use 15 basis functions per dimension. It is important to set proper variance values for the Gaussian kernels. A good starting values is the distance to the next Gaussian kernel. Train the agent for 500 episodes.