Prof. Frank Hutter, Prof. Joschka Bödecker

Dr. Marius Lindauer
Gabriel Kalweit

Reinforcement Learning
Exercise 6
Submit until **Thursday, January 11 at 2:00pm**

1 Policy Gradient Methods (20p)

Implement the missing parts of the `Policy` class and the REINFORCE algorithm from the lecture in

`YOUR_REPO/exercise-06/scripts/reinforce.py`.

We again apply the algorithm on the discrete Mountain Car Environment, so use a softmax output for your policy network. However, we modified the environment so as to apply the same action four times in a row (i.e. to skip three frames) and we slightly changed the reward to shorten training times.

An exemplary parameter setting is given in the script. You can use the neural network implementation provided or choose to replace it by your own.

You can again play around with the parameters and also introduce baselines (e.g. constant or an approximated value function). However, including baselines is optional and you do not have to do it to get full points. Submit a short report about your experiments and your experiences and compare with Q-learning. Save and submit your last policy and your best learned policy by

`tf.train.Saver().save(sess, "./policies.ckpt")`.

`best_policy.update(sess)` copies the weights of the current policy to the weights of the `PolicyCopy` object and can be used to store the best policy found so far.

2 Bonus: Experiences (1p)

Submit an `experiences.txt`, where you provide a brief summary of your experience with this exercise, the corresponding lecture and the last meeting. As a minimum, say how much time you invested and if you had major problems – and if yes, where.

Please push your solutions to subdirectory `exercise-06` in your assigned git-repository by **Thursday, January 11 at 2:00pm**. **Solutions after that or via email will not be accepted.**