Kiel University
Christian-Albrechts-Universität zu Kiel

Exercise Sheet 2
Autonomous Learning
May 22, 2019

Task 1 - SARSA

A. Extend the agent of the previous worksheet with a SARSA learning component.
   (a) The agent is supposed to walk as far as possible per step.
   (b) For modelling the state space, only the x- and y-distance from marine to beacon shall be used. Apart from that, divide the state space into sections:

$$x_1 < s_1 < x_2$$

$$y_1 < s_1 < y_2$$

   This helps to reduce the state space.

B. Research another non-ϵ-greedy exploration strategy.
   (a) Which one did you choose?
   (b) How does it work?
   (c) Implement the exploration strategy into the framework.

C. Visualise the learning progress analogously to Q-learning.

D. The trained SARSA agent shall be started with the python file (`RunSARSA.py`). The agent is supposed to read the trained Q-table and use it without further learning.

E. You should be able to start the SARSA agent training via the python file (`TrainSARSA.py`). By this, the agent should generate a new Q-table.

Task 2 - Deep Q-Network

A. Extend the agent with a DQN learning component.

B. For modelling the state space, no further research is needed.

C. Implement a Neural Network with *keras*:
   (a) 1 input layer
   (b) 1 hidden layer (16 nodes + ReLu)
   (c) 1 hidden layer (32 nodes + ReLu)

   (d) 1 output layer (8 nodes + linear)

D. Save the trained weights as `hdf5`-format.

E. Apply replay memory under the following conditions:

   (a) Total number: 100.000

   (b) The replay memory has to be filled with at least 6000 entries before learning starts.

F. Use all ϵ-greedy exploration strategies under the following conditions:

   (a) $\epsilon_{start} = 1$

   (b) $\epsilon_{min} = 0.05$

   (c) linearly decreasing until episode 1.000

G. Visualise the learning process as usual.

H. The trained DQN-agent shall be started with the python file (`RunDQN.py`). The agent is supposed to read the trained weights and use them without further learning.

I. You should be able to start the training of the DQN-agent via python fiel (`TrainDQN.py`). By this, the agent should learn new weights.