Artificial Intelligence and Data Analytics for Engineers

# Exercise 8

## Introduction

Solving exercises is not mandatory, and exercises will not be graded or corrected by the lecture team. However, we strongly advise you to do as many exercises as possible to prepare for the final exam. Some of the tasks will be discussed with a presentation of the solutions during the exercise session on Thursday. If you want to do exercises at home, we suggest installing Anaconda (https://www.anaconda.com/distribution/ - you will want to download the Python 3.7 version).

In case you have any questions, feel free to send us an e-Mail to: aidae@ima-ifu.rwth-aachen.de.

## Task 0) Setting up your Python environment

Since you created a Python environment in the first week, we only need to enable the environment from now on. This needs to be done **every time** you open a new terminal, for instance after a system restart or if you accidentally closed the Terminal. Open a terminal and enter:

```
source activate aidae
```

Now you are ready to start programming in Python. Remember to ensure that spyder is installed in your environment (`conda install spyder`) and launched from the terminal where you activated the environment.

## Task 1) Get familiar with open ai gym interface

a) For the following tasks we will use a simulated environment provided by OpenAI Gym. For this you need to install two additional libraries. Ensure that you have activated the environment and enter the following commands into your terminal:

```
pip install gym
pip install piglet
```

b) A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright. The episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center.A nice animation of that process can be found here: https://gym.openai.com/envs/CartPole-v0/

See task 2 on the next page

**Task 2) to improve our hill-climbing agent learning**

Run the task_2.py file. In `agent.train(train_steps=100)` adjust train_steps parameter. It doesn't look that this agent learns, does it?

a) Follow the instructions in the file

The current train process is missing logic for parameter update. We need to fix it to make our agent learn!

Implementation tip:

1) implement parameters update in train function, following logic:

new_parameters = parameters + random_noise * noise_scaling

2) check if new parameters are better than the old one

3) if step 2 is True, take these parameters as the parameter set used In the agent

4) repeat

b) Think about other ways to improve learning. Tip, how about noise_scaling decay during training? Why would it make sense? Try to implement it!