

Exercise sheet 1: Introduction to Python & Agents

Due on 3/11/2017, 2pm.

Important

1. **E-mail:** check which e-mail address is assigned to your Moodle account and make sure that you check your e-mails of this account frequently! All notifications regarding the course will be sent via Moodle.
2. **Submission:** ONE zip-file and ONE pdf-file. The names of both files should contain your surname.

General Information

All programming exercises are build in Python. You may use standard Python libraries or Anaconda (open source distribution of Python) to complete your exercises. Later, there will be some exercise sheets about Deep Learning. There are a lot of good libraries for this manner. We suggest you to use PyTorch¹, since it is easy to work with. Alternatively, you can use Keras², Caffe³, Tensorflow⁴ or Theano⁵. Keras is, besides PyTorch, a good alternative for beginners. We would not recommend, however, to work with one of the other libraries if you are not already familiar with them. There will be a small Introduction to PyTorch during the exercise session on the 8th of December.

Question 1: Introduction to Python - Chatbot (7P)

For those who are not familiar with Python at all we would recommend to go through the following tutorial: <https://docs.python.org/2.7/tutorial/>. There you find descriptions of the language syntax and basic data structures. We, however, would like to start with a little bit more advanced things.

In this exercise your are going to build your first very simple Artificial Intelligence using Python. The task is to write your own chatbot who is able to make a conversation with an user. All comments made from the chatbot should be output in the console.

¹<http://pytorch.org/>

²<https://keras.io/#keras-deep-learning-library-for-theano-and-tensorflow>

³<http://caffe.berkeleyvision.org/>

⁴<https://www.tensorflow.org/>

⁵<http://deeplearning.net/software/theano/>

During this exercise you should use the following libraries to complete your submission:

- **random**⁶: pseudo-random number generators for various distributions
- **NumPy**⁷: defines multidimensional arrays and provides efficient functions to operate with them
- **Matplotlib**⁸: library for plotting
- **SciPy**⁹: contains modules for common tasks in science and engineering

To edit this Question you need the provided file *chatbot.py* which contains the Class 'Chatbot'. The general structure and all needed functions are given.

a) Complete the functions

greetings(),
askQuestion() and
farewell()

according to their description in the commands.

The bot is supposed to react after the user inputs an answer to a question asked from the bot. For that, complete the function *reaction()* (in the main part). The reaction of the bot is based on the specific questions the bot is able to ask. More information are given in the function *askQuestion()*.

Please use the build-in function **raw_input()** to get the input of the user and for creating random numbers use the library **random**.

b) The bot should not be only able to ask questions, but also to answer questions the user asks during the conversation. For that, complete the functions

isQuestion() and
answerQuestion()

according to their description in the commands.

The rules for the user how to ask questions are as follows:

1. The user can only ask a question directly after he answered to a question asked by the bot.
2. Every answer should end with a dot and every question with a question mark.

⁶<https://docs.python.org/2/library/random.html>

⁷<http://www.numpy.org/>

⁸<http://matplotlib.org/>

⁹<https://www.scipy.org/>

(Ex: 'My name is Stan. What are your hobbies?')

After completing the functions, make sure that the chatbot always checks if the user asked a question after he reacted to answers/greeted the user and if yes that he answers them accordingly. For that call *isQuestion()* and *answerQuestion()* in *greeting()* and *askQuestion()*.

- c) Complete the function *plotGauss()* according to its description in the commands. Also change the variable 'self.shownGauss' to 0, so that the bot calls the function *plotGauss()* during the conversation.

When creating an object of the Chatbot, input at least:

- 3 different types of greetings
- 3 different questions asked from the Chatbot
- 3 different questions asked from the user and the appropriate answers
- 3 different types of farewells

Please provide the output of two conversations (+the plot with the Gauss curves) made between you and the chatbot in your submitted pdf-file.

please turn over

Question 2: Agents - Chatbot

(3P)

A chatbot, also called conversational agent or dialog system, is highly used nowadays to solve a number of business tasks across many industries. From now on assume a general chatbot, who can be more intelligent than the one implemented in Question 1.

- a) Give a PEAS description of the task environment of a chatbot.
- b) Decide and explain for a chatbot whether the environment is fully observable, deterministic, episodic, static, discrete and single-agent.
- c) Do you know any existing (used in the real world from users/customers) chatbots who are more intelligent than the chatbot from Question 1? Find 2 different types of existing chatbots, explain their task and determine the type of Agent (given the 5 basic types presented at the lecture).
- d) What do you think which type of Agent the chatbot from Question 1 is? Explain your choice.

Note: Submit exactly one ZIP file and one PDF file via Moodle before the deadline. The ZIP file should contain your executable code. Make sure that it runs on different operating systems and use relative paths. Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. The PDF file should contain your written code, all figures, explanations and answers to questions. Make sure that plots have informative axis labels, legends and captions.