Tutorial 4, 17.05.2018 BILHR SS 2018

Goal:
The goal of this tutorial is to implement a CMAC neural network for learning a mapping between spatial visual and motor patterns. Again, the robot should perform the same simple reaching task like in the previous tutorial. In addition, the CMAC neural net should be compared to the feedforward neural net (FFNN / MLP) of the previous tutorial.

Constraints (same as in tutorial 3):

- Depth information in the visual space is ignored.
- Keep the head and elbow of the NAO in constant positions.
- Use a colour marker attached to the NAO's hand (or finger) to collect accurate samples.

Tasks:

1) Collect the training set for the CMAC:
    - Set the stiffness for the 2 DOF shoulder joint to 0.
    - Set the stiffness for elbow and head to 0.9.
    - For each training sample, move the robot arm manually towards the object of interest. Use your code for the colour blob extraction of tutorial 2 to obtain the position of the object. This position vector is treated as input of the neural net. When the marker of the finger is close enough to the object, record the training sample (object blob position, 2 DOF shoulder position) by pressing the front tactile button on the NAO's head.
    - Collect and store 150 training samples.

2) Train the CMAC:
    - Build a CMAC with the following settings: 2 inputs, 2 outputs, resolution = 50, receptive field = 3.
    - Consider the training samples you collected with the NAO in task 1). Train the net with 75 training samples (case A), then with 150 training samples (case B). For each case, plot the results of the training phase (MSE over epochs).
    - Repeat the previous step for a receptive field = 5 (all other parameters stay the same).

3) Test the performance of your trained CMAC (trained with 150 samples, receptive field = 5). Make a video of the robot's behaviour.

4) Compare the CMAC (characteristics of training phase and execution phase) with the MLP of tutorial 3. Write down its advantages and disadvantages compared to the MLP.

Results to submit:

1) All relevant code implementing the CMAC and integrating it into the existing framework. Source code !! (no binaries, etc.)

2) The training samples of task 1).
3) The plot results of task 2).
4) Your comparison of task 4).

Don't forget to make a video of the robot's reaching behaviour.

Compress all the required results into a .zip or .tar.gz file (naming convention as in tutorial 2). Submit that file to: erhard.wieser@tum.de

Deadline: 06.06.18