Tutorial 4, 23/05/2019

Goal:

The goal of this tutorial is to implement a mixed density network (MDN) for learning a mapping between spatial visual and motor patterns. The mapping between visual and motor patterns enables the robot to perform a simple reaching task.

Constraints:

- Depth information in the visual space is ignored, i.e. only 2D coordinates in image plane.
- Keep the head and elbow of the NAO in constant positions.
- Use a colour marker attached to NAO's hand to collect accurate samples.

Task 1:

1. Compute by hand (use word or latex for that) the update equations for a 2-layer MDN (input layer, output layer, hidden layer) using backpropagation and submit the solutions in a PDF file. Please consider two input neurons for your model, one output, one kernel (Gaussian function) and one hidden layer with a random number of neurons.

Task 2:

2. Collect the training data for the MDN (visual blob position as input, shoulder position as output)
    - Set the stiffness for the 2 DOF shoulder joint to 0.0.
    - Activate the stiffness for elbow and head (0.9 or greater).
    - For each training sample, move the robot arm manually towards the object of interest.
    - Use your code for the colour blob extraction form the last tutorial to obtain the position of the object.
    - This position vector is treated as the input of the neural net. When the marker of the finger is close enough to the object, record the training sample by touching the front tactile button on the NAO's head.
    - Collect and record ca. 50-60 different training samples.
    - Normalize the input data (2D blob position) and output data (shoulder position) to values between 0.0 and 1.0, suitable for the FFNN.

3. Implement A **MDN network** (using keras or pure tensorflow) to perform a regression problem by mapping visual and motor patterns enabling the robot to perform a simple reaching task.

4. **Test the performance** of your trained neural network on the robot. Each time a new joint position sample arrives, your feed forward model should take the latest visual sample (blob position) as input and compute the corresponding joint positions to be sent to the robot. Note that the output delivered by the neural network has to be de-normalized to the actual robot joint angles before sending them to the robot. **Make a video of the robot's reaching behaviour.**

Delivery:

Please submit the following documents by **05-06-2019 midnight** at the very latest:

- All relevant code implementing the MDN neural network for the regression part.
- The PDF file for the backpropagation part.
- Your training data of task 2) (normalized values) as a txt-file.

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: zied.tayeb@tum.de