Tutorial 4, 09.06.2022 BILHR SS 2022

Setup:
Please install Jupyter Notebook for python 2 on your system in order to load the tutorial files. VS Code has a related plugin (https://code.visualstudio.com/docs/datascience/jupyter-notebooks).

Goal:
The goal of this tutorial is to implement a feed forward neural network (multi-layer perceptron) 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 color marker attached to NAO's hand to collect accurate samples.

Tasks:
1) Collect the training data for the MLP (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.
    - **Note**: you can reuse the samples from the CMAC tutorial.

2) Implement a three/four-layer neural network with no libraries (**only numpy, matplotlib is allowed!**) to classify MNIST dataset (classification problem, use this python library https://pypi.org/project/python-mnist/).
    - Implement your FFNN in numpy and test it on the MNIST dataset
    - Adapt and use your implemented neural network to perform a regression problem by mapping visual and motor patterns enabling the robot to perform a simple reaching task. 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.

3) Compare the CMAC (characteristics of training phase and execution phase) of last tutorial with the

MLP of this tutorial. Write down its advantages and disadvantages compared to the MLP.

Results to submit:

1) All relevant code implementing the FFNN (**MNIST** version and **NAO** version) and integrating it into the existing framework. Source code !! (no binaries, etc.)

2) Your training data of task 1) (*normalized* values) as a txt-file.

3) A video of the NAO reaching, using the MLP implementation

4) Your comparison of CMAC and MLP as .txt file.

Compress all the required results into a .zip or .tar.gz file (naming convention as in tutorial 2). Submit that file to: **bilhr-lecture.ics@ei.tum.de**

Deadline: 29.06.2022, 23:59