Robot Learning

Exercise 0 – Introduction

Release date: Wednesday, 6 September 2023 - **Deadline for Homework: Monday, 11 September 2023 11:59 PM**

Blackboard:

- We will organize the exercises sheets and your submissions using the **Blackboard**-System (`https://learn.bu.edu`).
- If you have any questions regarding the exercises, please ask at the **Discussion Forum** on blackboard.
- Assignments must be worked on individually. Please upload screen captures to show successful completion.

Submitting your solutions:

- Please compress your submission using the ZIP format and *only* include necessary files.
- We will provide a template for most tasks to avoid potential confusions. Please follow them to ensure a smooth procedure.

Infrastructure:

- You are eligible to use the Shared Computing Cluster (SCC) cluster. All students in the class will be automatically enrolled in an SCC account. Please read the cluster documentation for more details (`http://www.bu.edu/tech/support/research/system-usage/scc-quickstart/`).

Exercises

0.1 Local PyTorch Installation (1 Point)

We use PyTorch for exercises. To ensure a working environment, we ask you to install PyTorch on your machine:

a) It is recommended to manage PyTorch with Anaconda. Please install Anaconda with Python version 3.5+ following instructions at https://www.anaconda.com/download/. (**Please note, for SCC:** Anaconda should be installed in /projectnb/rlvn/students/yourname directory(not in home directory). Run the command *source /projectnb/rlvn/students/yourname/anaconda3/bin/activate* to avoid any conflict with preinstalled conda versions of SCC.)

b) Create and activate your own environment by *conda create -n < your_env_name > python=3.7*. You can choose any other python version > 3.7.

c) Install PyTorch following instructions at https://pytorch.org/get-started/locally/.

d) Once installation is complete, activate your environment and attach a screenshot of the terminal by running the following three commands: "*conda env list*", "*python --version*", and "*pip list | grep torch*" (the "|" is the pipe command on linux).

e) We ask you to run the toy regression code `exercise_pytorch.py` we provided. Please save the output log in the terminal as a `.txt` file. Please pack all files in this exercise into a ZIP file and upload it to the ILIAS system. Please rename the ZIP file as your name.

0.2 Local OpenAI Gym Installation (1 Point)

We ask you to install OpenAI Gym on your machine:

a) Please install Python 3.5+ if you don't have it.

b) Unzip the `sdc_gym.zip` you downloaded together with this Exercise Sheet and enter the folder `sdc_gym`, install the Box2D enviroment by the command

```
pip install —e '.[box2d]'
```

c) After installation, please take a screen-shot of your **full** screen with the car racing game on top. Please pack the screen-shot into the ZIP file and upload it to the ILIAS system. If you are in the `sdc_gym` folder, you should be able to start the car racing game by the command

```
cd gym/envs/box2d
python car_racing.py
```

0.3 Cluster Setup (1 Point)

To get familiar with the SCC cluster, we ask you to run a small Python script:

a) We recommend using the SCC desktop version for a better interactive experience. For more details, you can refer to https://www.bu.edu/tech/support/research/system-usage/connect-scc/scc-ondemand/.

- **i)** Read the cluster documentation and log in to the cluster after you get the account.
- **ii)** Apply for a desktop in Interactive Apps with 4 cores and 1 GPU, and load python3/3.8.3, libjpeg-turbo/2.0.4, and any other modules which you think is useful.
- **iii)** Connect to your desktop and you are now having a remote computer with powerful computing resources!
- **iv)** Submit a task to the cluster to run the Python script:

```python
from pyvirtualdisplay import Display
import gym

display = Display(visible=0, size=(800,600))
display.start()
env = gym.make('CarRacing-v0')
obs, done = env.reset(), False
ep_rew = 0.0
while not done:
    obs, rew, done, _ = env.step([0.0, 1.0, 0.0])
    ep_rew += rew
print(ep_rew)
display.stop()
```

and attach a screenshot of it.

0.4 CARLA Simulator Installation (2 Points)

i) Download ubuntu version CARLA 0.9.10.1 (development) from https://github.com/carla-simulator/carla/releases. In order to avoid bugs caused by version inconsistencies, we restrict the CARLA version to 0.9.10.1.

ii) Unzip your CARLA, cd to your CARLA directory, start your carla-server by running *DISPLAY= ./CarlaUE4.sh -opengl* (there is a space after "*DISPLAY=*"). Note: It is okay to have error messages like "Disabling core dumps." or "error: XDG_RUNTIME_DIR not set in the environment." as long as it is still running.

iii) Open a new terminal, activate your conda environment and import CARLA by :

```
export CARLA\_ROOT=YOUR_CARLA_DRECTORY
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/ ↙
    ↪ carla/dist/carla-0.9.10-py3.7-linux-x86_64.egg
```

then run the toy manual control code `manual_control.py` under "YOUR_CARLA_DRECTORY/PythonAPI/examples". You are supposed to use ↑↓←→ to control the car in a simulated driving environment. Please make screenshot and compress to your submission YOUR_NAME.zip.

iv) Read documents and get to know what CARLA is and what can it do https://carla.org/.