Set up your own RL environment

Linux OS based on the Intel chip (x86-64)

You can apply for a single account with the access to iHPC. The service is provided by UTS.

For more information, please check https://ihpc.research.uts.edu.au/pages/home

Step 1.

Download and install Anaconda Python distributed package for Linux OS.

Anaconda | Individual Edition

Anaconda's open-source Individual Edition is the easiest way to perform Python/R data science and machine learning on a single machine.

https://www.anaconda.com/products/individual

Btw, if you have no idea on how to install Anaconda, please check the iHPC document page.

Step 2.

Create a **virtual** Python environment and activate it.

```
conda create --name gym python=3.6 --no-default-packages # creat
<<'COMMENT'
Then you will get:

The following NEW packages will be INSTALLED:

  _libgcc_mutex      pkgs/main/linux-64::_libgcc_mutex-0.1-main
  _openmp_mutex      pkgs/main/linux-64::_openmp_mutex-4.5-1_gnu
  ca-certificates    pkgs/main/linux-64::ca-certificates-2022.2.1-h06a4308_0
  certifi            pkgs/main/noarch::certifi-2020.6.20-pyhd3eb1b0_3
  ld_impl_linux-64   pkgs/main/linux-64::ld_impl_linux-64-2.35.1-h7274673_9
  libffi             pkgs/main/linux-64::libffi-3.3-he6710b0_2
  libgcc-ng          pkgs/main/linux-64::libgcc-ng-9.3.0-h5101ec6_17
  libgomp            pkgs/main/linux-64::libgomp-9.3.0-h5101ec6_17
```

```
  libstdcxx-ng       pkgs/main/linux-64::libstdcxx-ng-9.3.0-hd4cf53a_17
  ncurses            pkgs/main/linux-64::ncurses-6.3-h7f8727e_2
  openssl            pkgs/main/linux-64::openssl-1.1.1m-h7f8727e_0
  pip                pkgs/main/linux-64::pip-21.2.2-py36h06a4308_0
  python             pkgs/main/linux-64::python-3.6.13-h12debd9_1
  readline           pkgs/main/linux-64::readline-8.1.2-h7f8727e_1
  setuptools         pkgs/main/linux-64::setuptools-58.0.4-py36h06a4308_0
  sqlite             pkgs/main/linux-64::sqlite-3.37.2-hc218d9a_0
  tk                 pkgs/main/linux-64::tk-8.6.11-h1ccaba5_0
  wheel              pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
  xz                 pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
  zlib               pkgs/main/linux-64::zlib-1.2.11-h7f8727e_4


Proceed ([y]/n)? y

COMMENT
source activate gym # now, activate it
```

Step 3.

Install MoJoCo simulation environment

- Download the MuJoCo version 2.0.0 binaries from https://roboti.us/download.html for Linux.
- Unzip the downloaded mujoco200 file into `~/.mujoco/mujoco200_linux`, and put your license key (the **mjkey.txt** file) at `~/.mujoco/mjkey.txt`. MoJoCo is an open sourced simulation platform for robot controlling. The free license can be found on https://roboti.us/license.html
- Set up the MuJoCo's environment variables in `~/.bash_profile`

```
export LD_LIBRARY_PATH=/home/your_user_name/.mujoco/mujoco200_linux/bin:$LD_LIBRARY_PATH
export MUJOCO_PY_MUJOCO_PATH=/home/your_user_name/.mujoco/mujoco200_linux/
export MUJOCO_PY_MJKEY_PATH=/home/your_user_name/.mujoco/mjkey.txt
```

- And then execute the following order in the Terminal

```
source ~/.bash_profile # make the environment variables active
```

Step 4.

Install and use `mujoco-py` that allows using MuJoCo simulator from Python 3.x

https://github.com/openai/mujoco-py

```
pip install 'mujoco-py<2.1,>=2.0'
```

Let’s test it

```
python
import mujoco_py
import os
mj_path, _ = mujoco_py.utils.discover_mujoco()
xml_path = os.path.join(mj_path, 'model', 'humanoid.xml')
model = mujoco_py.load_model_from_path(xml_path)
sim = mujoco_py.MjSim(model)
print(sim.data.qpos)
sim.step()
print(sim.data.qpos)
<<'COMMENT'
If everything works, You will get:

Python 3.6.13 |Anaconda, Inc.| (default, Jun  4 2021, 14:25:59)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import mujoco_py
>>> import os
>>> mj_path, _ = mujoco_py.utils.discover_mujoco()
>>> xml_path = os.path.join(mj_path, 'model', 'humanoid.xml')
>>> model = mujoco_py.load_model_from_path(xml_path)
>>> sim = mujoco_py.MjSim(model)
>>> print(sim.data.qpos)
[0.  0.  1.4 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ]
>>> sim.step()
>>> print(sim.data.qpos)
[-1.12164337e-05  7.29847036e-22  1.39975300e+00  9.99999999e-01
  1.80085466e-21  4.45933954e-05 -2.70143345e-20  1.30126513e-19
 -4.63561234e-05 -1.88020744e-20 -2.24492958e-06  4.79357124e-05
 -6.38208396e-04 -1.61130312e-03 -1.37554006e-03  5.54173825e-05
 -2.24492958e-06  4.79357124e-05 -6.38208396e-04 -1.61130312e-03
 -1.37554006e-03 -5.54173825e-05 -5.73572648e-05  7.63833991e-05
 -2.12765194e-05  5.73572648e-05 -7.63833991e-05 -2.12765194e-05]

COMMENT
```

Step 5.

Install OpenAI's Gym and test it. https://gym.openai.com/docs/

```
pip install 'gym[all]'==0.18.3
```

Note that the installation will replace your `mucojo-py` version with the version 1.5.0 (yes, it's a terrible bug). You need to re-install mujoco-py by executing `pip install 'mujoco-py<2.1,>=2.0'` in the Terminal.

Now, let's test the Gym environment.

```
import gym
env = gym.make('Hopper-v2') # create a hopper robot
init_state = env.reset()
print(init_state)
'''
you will get a list like this:

[ 1.24926322e+00  2.84251878e-03  2.08722202e-03 -3.67633332e-03
  4.79440147e-03 -2.13144403e-03 -3.03359369e-03 -4.20995353e-03
  5.28215601e-04 -4.77199605e-03  3.44786460e-03]

'''
```

For more details about interaction and visualization, please check the official documentation. And there are lots of RL environments you can select.