

implementation

Learning_Robots

implementation

Learning_Robots

1. Introduction

2. Requirements

3. Installation:

3.1 Python packages:

3.1.1 Option 1: Conda yml

3.1.2 Option 2: manual

3.1.2.1 Jupyter Notebook

3.1.2.2 Tensorflow

3.1.2.3 Matplotlib

3.1.2.4 Pandas

3.2 Extra packages:

3.2.1 OpenAi Baselines

3.2.2 MuJoCo

3.2.3 PyBullet

3.2.4 ffmpeg

4. Execution:

1. Introduction

The "visualization.ipynb" notebook runs the OpenAi baselines implementation of the PPO algorithm for 1 Million steps (using the Hyperparameters given on the PPO paper) on the Hopper-V2 enviroment (MuJoCo and PyBullet versions) and Visualizes the results comparing them to the original paper and to the baselines given by OpenAi baselines.

2. Requirements

- Linux specifically Ubuntu 18.04 LTS and Pop!_OS 20.04 ; and Mac OS Catalina were tested other versions could need different modules (OpenAi Baselines has problems under Windows and MuJoCo only runs on Linux or OS x (theoretically), windows is deprecated. I couldn't make MuJoCo run on Mac OS Catalina but PyBullet does.)
- python3 (3.6 or 3.7 , i.e. 3.5 and 3.8 don't work)
- Tensorflow 1.14 (Intel optimized version is used)
- Jupyter notebooks
- OpenAi Gym
- OpenAi baselines (<https://github.com/openai/baselines>)
- matplotlib
- pandas
- MuJoCo
- PyBullet

Download the implementation folder.

Best visualized as Jupyter Notebook.

Mujoco, i.e. mujoco-py, is currently not working on MacOS Catalina; there are several [issues](#), but PyBullet works.

3. Installation:

Assuming python3 (3.6 or 3.7) and conda are installed.

Note I used all the conda versions of the packages, it could be possible that issues arise with the pip packages.

3.1 Python packages:

3.1.1 Option 1: Conda yml

- Create a conda enviroment using the "enviromet.yml" included as follows

```
conda env create -f environment.yml
conda activate LR-tf1
```

LR-tf1: Learning Robots using tensorflow 1.14 (the intel optimized version. No GPU support.)

3.1.2 Option 2: manual

3.1.2.1 Jupyter Notebook

- **Conda**

install the notebook with:

```
conda install -c conda-forge notebook
```

- **pip**

install with:

```
pip install notebook
```

3.1.2.2 Tensorflow

The master branch of OpenAi baselines supports Tensorflow from version 1.4 to 1.14. Here version 1.14 will be used

- **Conda**

Using anaconda the intel optimized cpu version of tensorflow 1.14 was used.

- **Linux and MacOS**

```
conda install tensorflow==1.14 -c anaconda
```

- **Windows**

```
conda install tensorflow-mkl==1.14 -c anaconda
```

- **alternative pip**

Alternative the normal version of tensorflow can be installed although this was not tested.

```
pip install tensorflow-gpu==1.14 # if a CUDA-compatible gpu with proper  
drivers is available
```

or

```
pip install tensorflow==1.14
```

3.1.2.3 Matplotlib

- **Conda**

install the notebook with:

```
conda install -c conda-forge matplotlib
```

- **pip**

install with:

```
pip install -U matplotlib
```

3.1.2.4 Pandas

- **Conda**

install the notebook with:

```
conda install pandas
```

- **pip**

install with:

```
pip install pandas
```

3.2 Extra packages:

3.2.1 OpenAi Baselines

Instead of using the git version of OpenAi baselines a slightly modified one is inside the project.
(Modified defaults.py in ppo2 and other small changes)

Note: no explicit instructions for windows are given, I didn't test it on windows only on Ubuntu
18.04, 20.04 and MacOS Catalina.

Following system packages CMake, OpenMPI and zlib are needed. Those can be installed as
follows

- **Linux**

```
sudo apt-get update && sudo apt-get install cmake libopenmpi-dev python3-dev  
zlib1g-dev # All one line
```

- **Mac OS X**

Installation of system packages on Mac requires [Homebrew](#). With Homebrew installed, run the following:

```
brew install cmake openmpi
```

- **Install baselines package**

Inside the implementation folder do:

```
cd baselines  
pip install -e .  
cd ..
```

3.2.2 MuJoCo

1. Obtain a 30-day free trial on the [MuJoCo website](#) or free license if you are a student. The license key will arrive in an email with your username and password.
2. Download the MuJoCo version 2.0 binaries for [Linux](#) or [OSX](#).
3. Unzip the downloaded `mujoco200` directory into `~/mujoco/mujoco200`, (make sure that it is `mujoco200` and not e.g. `mujoco200_macos`) and place your license key (the `mjkey.txt` file from your email) at `~/mujoco/mjkey.txt`.
4. Run:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/"REPLACE WITH PATH  
TO"/mujoco/mujoco200/bin
```

5.
 - **Linux**

To install `mujoco-py` on Ubuntu, make sure you have the following libraries installed:

```
sudo apt install libosmesa6-dev libgl1-mesa-glx libglfw3 patchelf  
pip install -U 'mujoco-py<2.1,>=2.0'
```

- **Mac OS X**

Currently `mujoco-py` is not working on MacOS Catalina; there are several [issues](#), but `PyBullet` works.

If you want to specify a nonstandard location for the key and package, use the env variables `MUJOCO_PY_MJKEY_PATH` and `MUJOCO_PY_MUJOCO_PATH`.

For further details or issues see [MuJoCo GitHub](#)

Troubleshooting if training does not work see if in the Jupyter console, i.e. the console running the jupyter server, the MuJoCo path is recognized, otherwise follow the given instructions in the error.

3.2.3 PyBullet

- **pip**

install the package with:

```
pip3 install pybullet --upgrade --user  
# or  
pip install pybullet
```

For further details or issues see [PyBullet GitHub](#) or the [PyBullet Quickstart Guide](#)

3.2.4 ffmpeg

- **Mac OS X**

install the package with:

```
brew install ffmpeg
```

- **Linux**

install the package with:

```
sudo apt-get install ffmpeg
```

Extra for Ubuntu 14.04:

```
sudo apt-get install libav-tools
```

4. Execution:

Inside the implementation folder using a console run:

```
# If following the installation make sure to go out of the baseline folder and  
then  
jupyter notebook
```

and open the "visualization.ipynb" notebook.