benelot / **pybullet-gym** Public

Open-source implementations of OpenAI Gym MuJoCo environments for use with the OpenAI Gym Reinforcement Learning Research Platform.

pybullet.org/

View license

652 stars 101 forks

Star Notifications

Code Issues 28 Pull requests 5 Actions Projects Wiki Security Insights

master Go to file

benelot ... on 29 Mar

View code

PyBullet Gymperium

PyBullet Gymperium is an open-source implementation of the OpenAI Gym MuJoCo environments for use with the OpenAI Gym Reinforcement Learning Research Platform in support of open research.

OpenAI gym is currently one of the most widely used toolkit for developing and comparing reinforcement learning algorithms. Unfortunately, for several challenging continuous control environments it requires the user to install MuJoCo, a commercial physics engine which requires a license to run for longer than 30 days. Such a commercial barrier hinders open research, especially in the perspective that other appropriate physics engines exist. This repository provides alternative implementations of the original MuJoCo environments which can be used for free. The environments have been reimplemented using BulletPhysics' python wrapper pybullet, such that they seamlessly integrate into the OpenAI gym framework. In order to show the usability of the new environments, several RL agents from the Tensorforce Reinforcement Learning Library are configured to be trainable out of the box. To simplify research with the implemented environment, each environment is featured with pretrained agents which serve as unit tests for the implementations and could well serve as baselines for other purposes.

If you find our work useful in your research please consider citing as follows:

@misc {benelot2018, author = {Benjamin Ellenberger}, title = {PyBullet Gymperium}, howpublished = {\url{ https://github.com/benelot/pybullet-gym}} , year = {2018--2019} }

State of implementations

Environment Name	Implemented	Similar to Reference Implementation	Pretraine agent available
RoboSchool Envs			
InvertedPendulumPyBulletEnv-v0	Yes	Yes	No
InvertedDoublePendulumPyBulletEnv-v0	Yes	Yes	No
InvertedPendulumSwingupPyBulletEnv-v0	Yes	Yes	No
ReacherPyBulletEnv-v0	Yes	Yes	No
Walker2DPyBulletEnv-v0	Yes	No	No
HalfCheetahPyBulletEnv-v0	Yes	No	No
AntPyBulletEnv-v0	Yes	Yes	No
HopperPyBulletEnv-v0	Yes	Yes	No
HumanoidPyBulletEnv-v0	Yes	Yes	No
HumanoidFlagrunPyBulletEnv-v0	Yes	Yes	No
HumanoidFlagrunHarderPyBulletEnv-v0	Yes	Yes	No
AtlasPyBulletEnv-v0	WIP	No	No
PusherPyBulletEnv-v0	WIP	No	No
ThrowerPyBulletEnv-v0	WIP	No	No
StrikerPyBulletEnv-v0	WIP	No	No
MuJoCo Envs			
InvertedPendulumMuJoCoEnv-v0	Yes	Yes	Yes

Environment Name	Implemented	Similar to Reference Implementation	Pretraine agent available
InvertedDoublePendulumMuJoCoEnv-v0	Yes	Yes	Yes
ReacherMuJoCoEnv-v0	No	No	No
Walker2DMuJoCoEnv-v0	Yes	No	No
HalfCheetahMuJoCoEnv-v0	Yes	No	No
AntMuJoCoEnv-v0	Yes	No	No
HopperMuJoCoEnv-v0	Yes	No	No
HumanoidMuJoCoEnv-v0	Yes	No	No
PusherMuJoCoEnv-v0	No	No	No
ThrowerMuJoCoEnv-v0	No	No	No
StrikerMuJoCoEnv-v0	No	No	No

[See What's New section below](#What's New)

Basics

(taken from OpenAI gym readme)

There are two basic concepts in reinforcement learning: the environment (namely, the outside world) and the agent (namely, the algorithm you are writing). The agent sends `actions` to the environment, and the environment replies with `observations` and `rewards` (that is, a score).

The core `gym` interface is `Env <https://github.com/openai/gym/blob/master/gym/core.py>`_, which is the unified environment interface. There is no interface for agents; that part is left to you. The following are the `Env` methods you should know:

- `reset(self)` : Reset the environment's state. Returns `observation` .

README.md

- `render(self, mode='human', close=False)` : Render one frame of the environment. The default mode will do something human friendly, such as pop up a window. Passing the `close` flag signals the renderer to close any such windows.

In addition to the basic concepts of reinforcement learning, this framework extends the notion of an environment into two subconcepts being the robot (the agents directly controllable body) and the scene (all things the agents interacts with). Implementing RL environments in this way allows us to switch parts of the environment to generate new robot-scene combinations.

Installing Pybullet-Gym

First, you can perform a minimal installation of OpenAI Gym with

```
git clone https://github.com/openai/gym.git
cd gym
pip install -e .
```

Then, the easiest way to install Pybullet-Gym is to clone the repository and install locally

```
git clone https://github.com/benelot/pybullet-gym.git
cd pybullet-gym
pip install -e .
```

Important Note: *Do not* use `python setup.py install` as this will not copy the assets (you might get missing SDF file errors).

Finally, to test installation, open python and run

```
import gym  # open ai gym
import pybulletgym  # register PyBullet enviroments with open ai gym

env = gym.make('HumanoidPyBulletEnv-v0')
# env.render() # call this before env.reset, if you want a window showing the environ
env.reset()  # should return a state vector if everything worked
```

Supported systems

We currently support Linux, Windows and OS X running Python 2.7 or 3.5.

To run `pip install -e '.[all]'` , you'll need a semi-recent pip. Please make sure your pip is at least at version `1.5.0` . You can upgrade using the following: `pip install --ignore-installed pip` . Alternatively, you can open `setup.py <https://github.com/openai/gym/blob/master/setup.py>` _ and install the dependencies by hand.

Agents

As some sort of unit test for the environments, we provide pretrained agents for each environment. The agents for the roboschool envs and the mujoco were trained on the original implementations of roboschool and mujoco respectively.

Environments

The code for each environment group is housed in its own subdirectory `gym/envs <https://github.com/openai/gym/blob/master/gym/envs>`. *The specification of each task is in `gym/envs/__init__.py <https://github.com/openai/gym/blob/master/gym/envs/__init__.py>`.* It's worth browsing through both.

What's new

- 2018-01-09 Pybullet-gym is born.

Roadmap

1. [ROBOSCHOOL GYMS] The current gyms are the roboschool gyms ported to pybullet. So far, most of them work well, except for the manipulator envs striker, pusher and thrower, where the robot is not correctly loaded. This will have to be fixed with Erwin Coumans.
2. [OPENAI MUJOCO GYMS] Soon I will start to port the OpenAI gyms, which unfortunately have a slightly different observation (and probably action) vector. I can setup all the gyms quickly, but it will take a while to find out what some of observations are in mujoco and what they correspond to in pybullet. Some of the observations might not be exposed on pybullet, then we can request them, for others it is already hard to know what they are in mujoco.
3. [OPENAI ROBOTICS GYMS] Next in line would be the robotics gyms in OpenAI. These are particularly delicate simulations and might take some tuning to even be simulatable in pybullet.
4. [DEEPMIND CONTROL SUITE] Then there is Deepmind Control Suite, another set of gyms which are in mujoco and need to be freed.

Releases

No releases published

Packages

No packages published

Contributors 13

+ 2 contributors

Languages

- **Python** 100.0%