



《强化学习》课程之第二讲（2021年春季研究生）

环境的配置

苏州大学计算机科学与技术学院

主讲：刘全

目 录

2.1

PyTorch简介

2.2

PyTorch和TensorFlow

2.3

强化学习的开发环境

2.4

Gym实验环境

2.1 PyTorch简介（1）

- 对于从事深度强化学习研究的相关工作人员而言，选择一个好的工具势必会事半功倍。。
- **PyTorch**是Facebook公司在深度学习框架**Torch**基础上，使用**Python**重写的一个全新的框架。
- **PyTorch**还拥有丰富的**API**，可以快速完成深度神经网络模型的搭建和训练。

2.1 PyTorch简介 (2)

PyTorch:

- ✓ 美国互联网巨头**Facebook**公司使用**Python**开发的一个深度学习框架;
- ✓ 框架设计理念先进, 在调试、编译等方面的优势, 迅速占领**GitHub**热度榜榜首;
- ✓ 不是简单地封装**Lua**、**Torch**, 提供**Python**接口, 而是对**Tensor**之上的所有模块进行了重构。

目 录

2.1

PyTorch简介

2.2

PyTorch和TensorFlow

2.3

强化学习的开发环境

2.4

Gym实验环境

2.2 PyTorch和TensorFlow（1）

➤Tensorflow创建的是静态图

Tensorflow在定义模型的整个计算图之后才开始运行机器学习模型；

➤PyTorch创建的是动态图

PyTorch的计算图是在运行时创建的，可以随时定义、随时更改、随时执行节点；

2.2 PyTorch和TensorFlow (2)

➤ Tensorflow的学习曲线陡峭

使用Tensorflow，需要先理解Tensorflow的一些工作机制（比如会话（`session`）、占位符（`placeholder`）等）。

➤ PyTorch学习曲线简单

PyTorch更贴近Python语言，建立机器学习模型时也更直接、符合人的思维。

目 录

2.1

PyTorch简介

2.2

PyTorch和TensorFlow

2.3

强化学习的开发环境

2.4

Gym实验环境

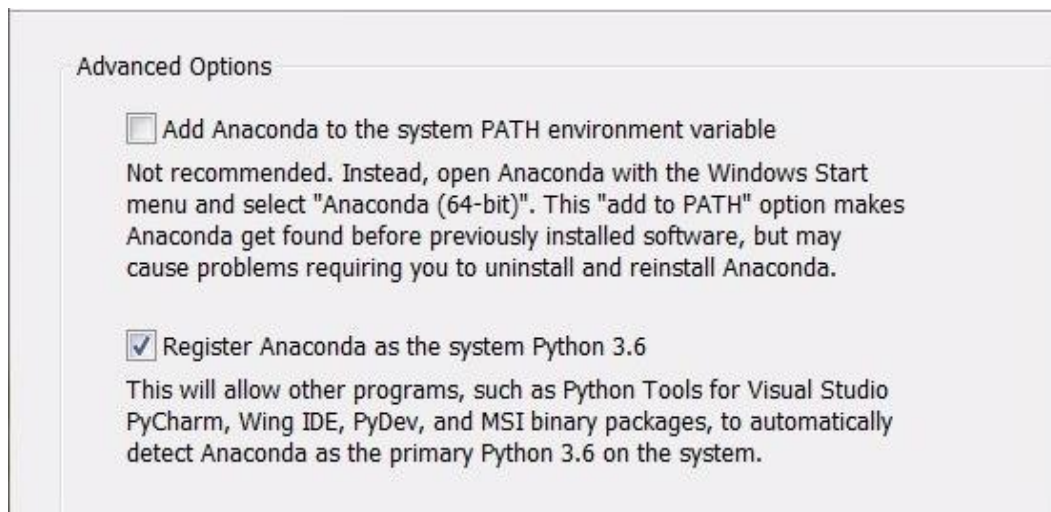
2.3 强化学习的开发环境（1） **aniconda**

➤ Windows环境Anaconda搭建

- ✓ 进入Anaconda的官网<https://www.anaconda.com>的下载界面，选择合适的Windows安装包进行下载安装即可。

注：清华镜像：

`pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple`



2.3 强化学习的开发环境 (2) **aniconda**

✓进入:

<https://microsoft-visual-studio.softonic.cn/download>

下载安装包，安装时只需勾选“使用**C++**的桌面开发”。

✓等待安装完成后重启电脑。重启后进入上述**anaconda**终端，后续的环境安装与**Linux**环境保持一致执行命令即可。

✓`conda install swig`

2.3 强化学习的开发环境（3） **aniconda**

➤Linux和Mac环境Anaconda搭建

- ✓进入**Anaconda**的官网<https://www.anaconda.com>选择所需系统的版本下载；
- ✓执行**bash Anaconda3-2019.10-Linux-x86_64.sh**（以下载的文件名为准）；
- ✓安装过程中会提示是否需要将安装路径写入到环境变量中，输入 **“yes”** 回车执行。

2.3 强化学习的开发环境（4） **aniconda**

➤ **Anaconda**搭建环境的创建、维护和管理

✓ **Anaconda**安装成功后，可以创建新环境并对新环境进行基本的环境维护和环境管理。环境管理部分都采用命令行方式。

✓ **Linux**和**Mac**用户直接打开终端即可；

Windows用户在开始菜单中打开“**Anaconda Prompt**”进入命令行模式。

2.3 强化学习的开发环境（5） **aniconda**

➤ 创建环境： **conda create --name rl python=3.7**

执行“**conda create --name rl python=3.7**”创建环境。其中，**rl**是环境名字，可以自定义；**python**版本号3.7根据自己所需版本进行调整。

➤ 查看环境： **conda env list**

“**conda env list**”命令可以查看当前用户已创建成功的环境，此时可以看到刚才创建的新环境 **rl**。

2.3 强化学习的开发环境 (6) **aniconda**

➤ 激活环境: **conda activate rl**:

“**conda activate rl**” 命令可以激活指定 **rl** 环境。

➤ 退出环境: **conda deactivate**:

“**conda deactivate**” 命令可以退出所创建的环境。

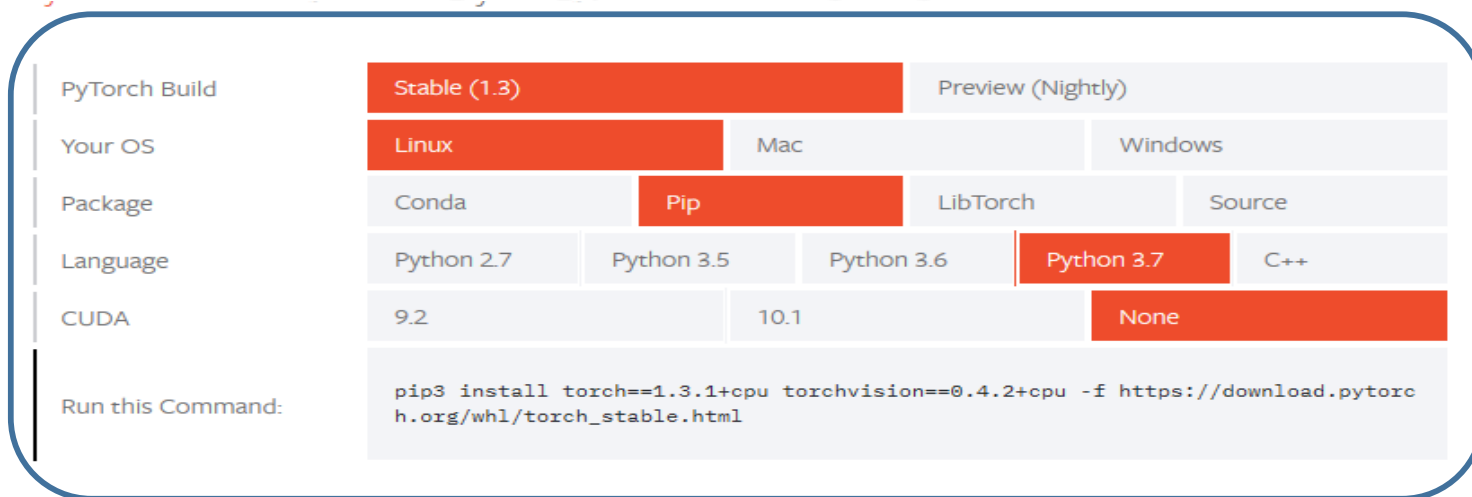
➤ 删除环境: **conda remove --name rl --all**:

“**conda remove --name rl --all**” 命令可以删除新创建的**rl** 环境。

2.3 强化学习的开发环境（7） PyTorch

➤ PyTorch环境安装

进入官网<https://pytorch.org>的get started页面



The screenshot shows the PyTorch installation guide interface. It features a sidebar with labels: PyTorch Build, Your OS, Package, Language, CUDA, and Run this Command. The main content area is a grid of buttons for selecting installation options. The selected options are highlighted in red: Stable (1.3) for the build, Linux for the OS, Pip for the package, Python 3.7 for the language, and None for CUDA. The Run this Command section displays the command: `pip3 install torch==1.3.1+cpu torchvision==0.4.2+cpu -f https://download.pytorch.org/whl/torch_stable.html`.

PyTorch Build	Stable (1.3)		Preview (Nightly)		
Your OS	Linux	Mac	Windows		
Package	Conda	Pip	LibTorch	Source	
Language	Python 2.7	Python 3.5	Python 3.6	Python 3.7	C++
CUDA	9.2	10.1	None		
Run this Command:	<code>pip3 install torch==1.3.1+cpu torchvision==0.4.2+cpu -f https://download.pytorch.org/whl/torch_stable.html</code>				

根据自己的环境配置情况进行选择。执行使用

Run this Command

中的命令即可安装。

2.3 强化学习的开发环境（7） Jupyter Notebook

➤ Jupyter Notebook的安装

执行**pip install jupyter**命令安装。

```
milaso@leo:~$ pip install jupyter
Requirement already satisfied: jupyter in ./anaconda3/lib/python3.7/site-packages (1.0.0)
Requirement already satisfied: ipywidgets in ./anaconda3/lib/python3.7/site-packages (from jupyter) (7.5.1)
Requirement already satisfied: qtconsole in ./anaconda3/lib/python3.7/site-packages (from jupyter) (4.5.5)
Requirement already satisfied: jupyter-console in ./anaconda3/lib/python3.7/site-packages (from jupyter) (6.0.0)
Requirement already satisfied: nbconvert in ./anaconda3/lib/python3.7/site-packages (from jupyter) (5.6.0)
Requirement already satisfied: notebook in ./anaconda3/lib/python3.7/site-packages (from jupyter) (6.0.1)
Requirement already satisfied: ipykernel in ./anaconda3/lib/python3.7/site-packages (from jupyter) (5.1.2)
Requirement already satisfied: widgetsnbextension~=3.5.0 in ./anaconda3/lib/python3.7/site-packages (from ipywidgets->jupyter) (3.5.1)
Requirement already satisfied: ipython>=4.0.0; python_version >= "3.3" in ./anaconda3/lib/python3.7/site-packages (from ipywidgets->jupyter) (7.8.0)
Requirement already satisfied: nbformat>=4.2.0 in ./anaconda3/lib/python3.7/site-packages (from ipywidgets->jupyter) (4.4.0)
Requirement already satisfied: traitlets>=4.3.1 in ./anaconda3/lib/python3.7/site-packages (from ipywidgets->jupyter) (4.3.3)
```


2.3 强化学习的开发环境 (8) **Gym**

➤ 强化学习中经典的实验场景:

Mountain-Car, Cart-Pole, SpaceInvaders, Go, LunarLander, MuJoCo等。

➤ 由于**DRL**的兴起, 各种新的复杂的实验场景不断涌出, 出现一系列优秀的实验平台:

◆ **OpenAI Gym, OpenAI Baselines;**

◆ **MuJoCo, rllab, TORCS, PySC2。**

2.3 强化学习的开发环境 (9) Gym

➤ OpenAI Gym的安装

OpenAI Gym的官网地址为<https://gym.openai.com/>,
源码地址为<https://github.com/openai/gym>。Gym的完整
安装需要提前安装MuJoCo。

MuJoCo官网<http://mujoco.org>。

官网的license界面注册获得权限之后再下载安装。

2.3 强化学习的开发环境 (10) **Gym**

➤ 经典OpenAI Gym的安装

安装除MuJoCo之外的Gym环境的方法。在Anaconda安装完成后，在命令行窗口输入如下命令（安装git）：

git clone <https://github.com/openai/gym.git>

2.3 强化学习的开发环境 (11) **Gym**

- ✓ 进入到克隆的**gym**目录下打开**setup.py**文件，将该文件与**MuJoCo**相关的两行代码注释掉。

```
-----
import sys, os.path

# Don't import gym module here, since deps may not be installed
sys.path.insert(0, os.path.join(os.path.dirname(__file__), 'gym'))
from version import VERSION

# Environment-specific dependencies.
extras = {
    'atari': ['atari_py~=0.2.0', 'Pillow', 'opencv-python'],
    'box2d': ['box2d-py~=2.3.5'],
    'classic_control': [],
    # 'mujoco': ['mujoco_py>=1.50, <2.0', 'imageio'],
    # 'robotics': ['mujoco_py>=1.50, <2.0', 'imageio'],
}

# Meta dependency groups.
extras['all'] = [item for group in extras.values() for item in group]

setup(name='gym',
      version=VERSION,
      description='The OpenAI Gym: A toolkit for developing and comparing your reinforcement learning agents.',
      url='https://github.com/openai/gym',
      author='OpenAI',
      author_email='gym@openai.com',
      license='')
```

2.3 强化学习的开发环境 (12) Gym

然后进入命令行，执行如下命令：

conda install swig

cd gym

pip install -e ‘.[all]’

即可等待安装完成。

```
(rl) milaso@leo:~/gym$ pip install -e ‘.[all]’
Obtaining file:///home/milaso/gym
Collecting scipy
  Downloading scipy-1.5.2-cp37-cp37m-manylinux1_x86_64.whl (25.9 MB)
    | 25.9 MB 3.4 MB/s
Requirement already satisfied: numpy>=1.10.4 in /home/milaso/anaconda3/envs/rl/lib/python3.7/site-packages (from gym==0.15.4) (1.19.1)
Requirement already satisfied: six in /home/milaso/anaconda3/envs/rl/lib/python3.7/site-packages (from gym==0.15.4) (1.15.0)
Collecting pygame<=1.3.2,>=1.2.0
  Using cached pygame-1.3.2-py2.py3-none-any.whl (1.0 MB)
Collecting cloudpickle~1.2.0
  Using cached cloudpickle-1.2.2-py2.py3-none-any.whl (25 kB)
Collecting opencv-python
  Downloading opencv_python-4.4.0.42-cp37-cp37m-manylinux2014_x86_64.whl (49.4 MB)
    | 49.4 MB 70 kB/s
Collecting atari_py==0.2.0
  Using cached atari_py-0.2.6-cp37-cp37m-manylinux1_x86_64.whl (2.8 MB)
Requirement already satisfied: Pillow in /home/milaso/anaconda3/envs/rl/lib/python3.7/site-packages (from gym==0.15.4) (7.2.0)
Collecting box2d-py==2.3.5
```

目 录

2.1

PyTorch简介

2.2

PyTorch和TensorFlow

2.3

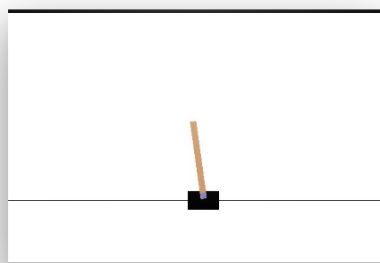
强化学习的开发环境

2.4

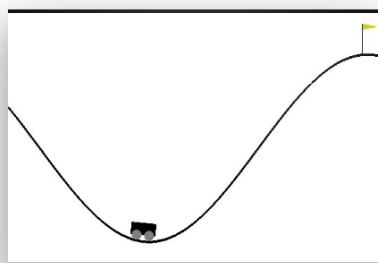
Gym实验环境

2.4 Gym 实验环境 (1)

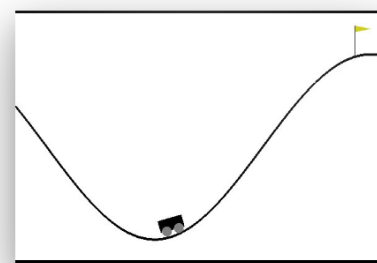
经典控制:



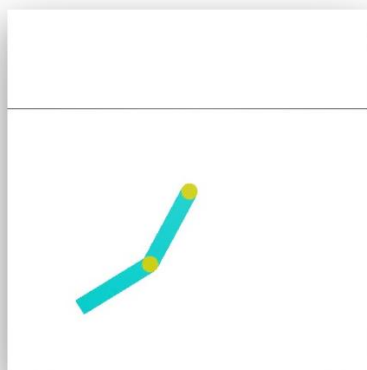
CartPole-v1
Balance a pole on a cart.



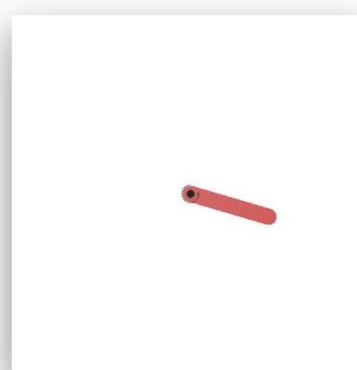
MountainCar-v0
Drive up a big hill.



MountainCarContinuous-v0
Drive up a big hill with continuous control.



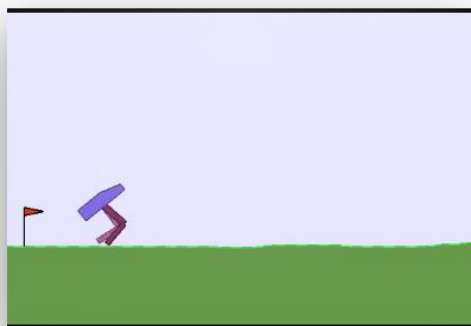
Acrobot-v1
Swing up a two-link robot.



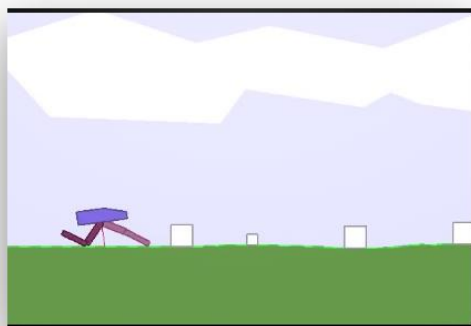
Pendulum-v0
Swing up a pendulum.

2.4 Gym 实验环境 (2)

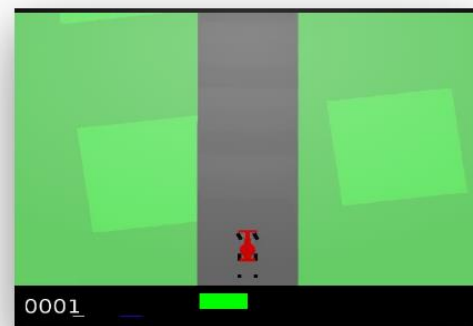
Box2D: Box2D中的连续控制任务



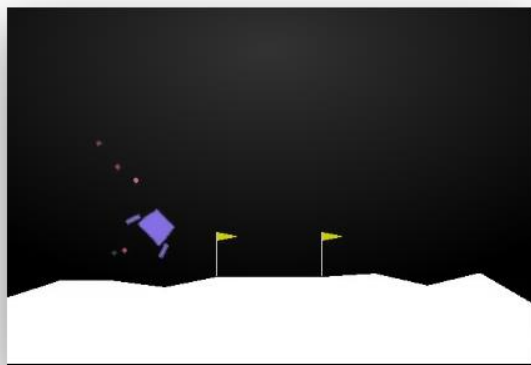
BipedalWalker-v2



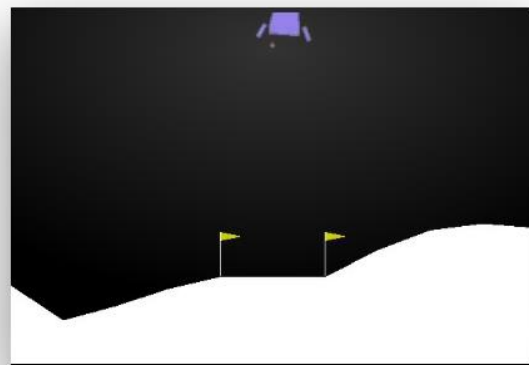
BipedalWalkerHardcore-v2



CarRacing-v0



LunarLander-v2



LunarLanderContinuous-v2

2.4 Gym实验环境（3）

➤ 经典OpenAI Gym的安装

在安装atari之前，要保证：

C:\Users\86134\anaconda3\lib\site-packages\atari_py\ale_interface\ale_c.dll

文件存在。

pip install gym[atari]

或

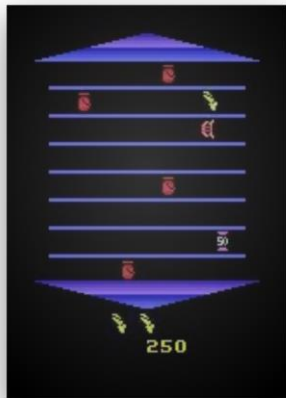
pip install atari_py

2.4 Gym实验环境 (3)

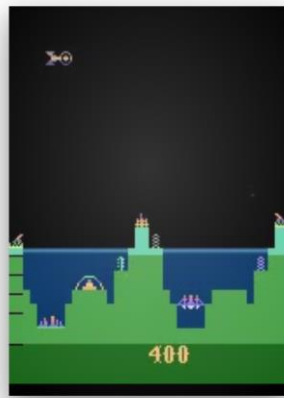
Atari: Atari 2600游戏



Alien-v0



Asterix-ram-v0



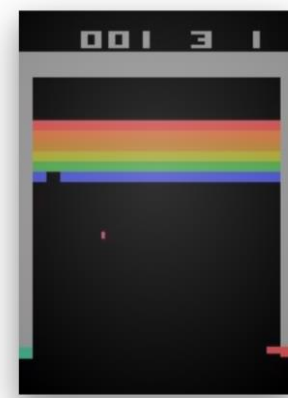
Atlantis-ram-v0



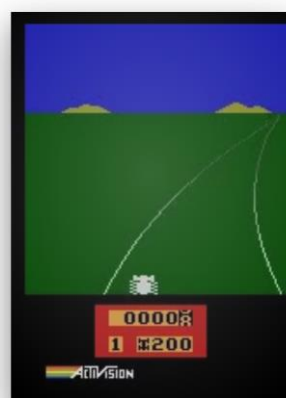
BattleZone-ram-v0



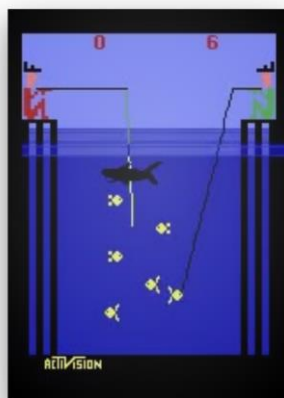
Boxing-ram-v0



Breakout-v0



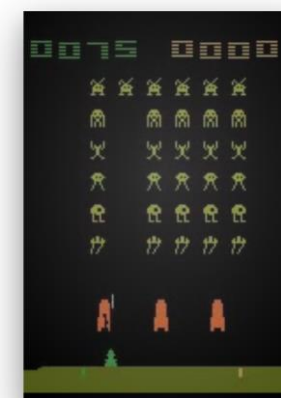
Enduro-v0



FishingDerby-v0



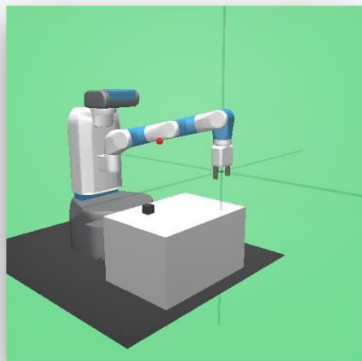
Seaquest-v0



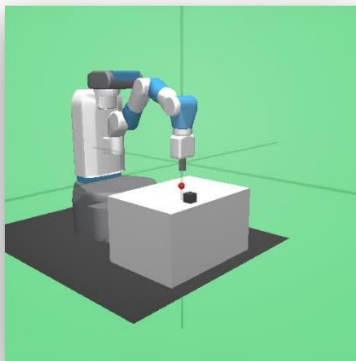
SpaceInvaders-v0

2.4 Gym 实验环境 (4)

Robotics: 抓取机器人完成基于目标的任务



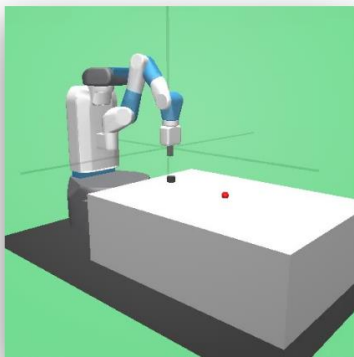
FetchPickAndPlace-v1
Lift a block into the air.



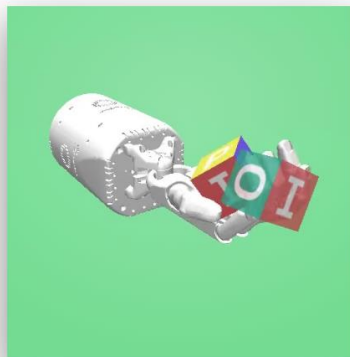
FetchPush-v1
Push a block to a goal position.



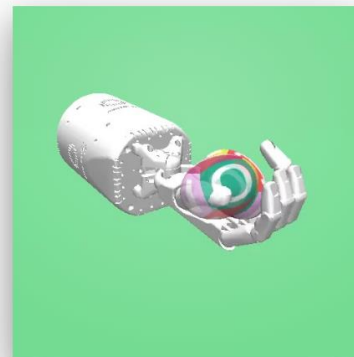
FetchReach-v1
Move Fetch to a goal position.



FetchSlide-v1
Slide a puck to a goal position.



HandManipulateBlock-v0
Orient a block using a robot hand.



HandManipulateEgg-v0
Orient an egg using a robot hand.

2.4 Gym实验环境（5）

➤ MuJoCo

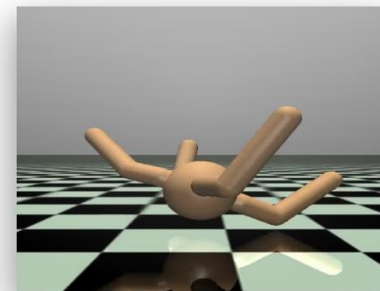
◆ MoJoCo是一个物理模拟器，可以用于机器人控制优化等研究；

◆ 官方网站为： <http://www.mujoco.org/index.html>;

◆ **注意：** MoJoCo需要license，可以在

<https://www.roboti.us/license.html>

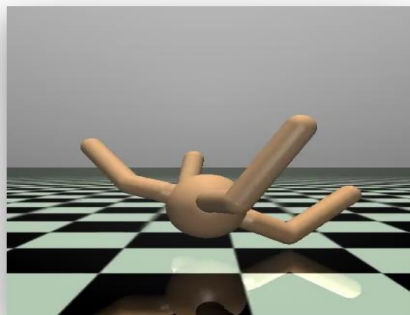
上申请试用30天的Licence。



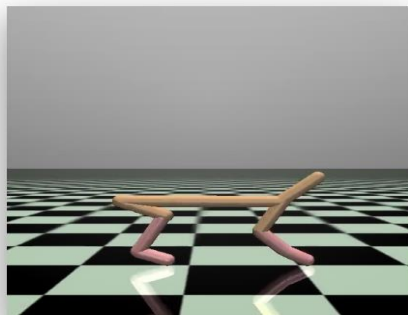
Ant-v2
Make a 3D four-legged robot walk.

2.4 Gym 实验环境 (6)

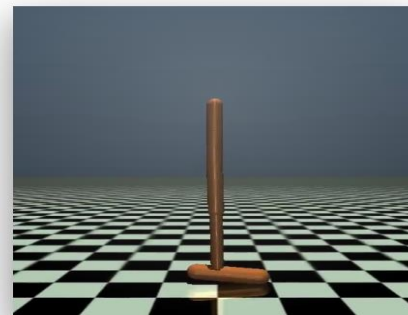
MoJoCo: 快速实体任务中的连续控制任务



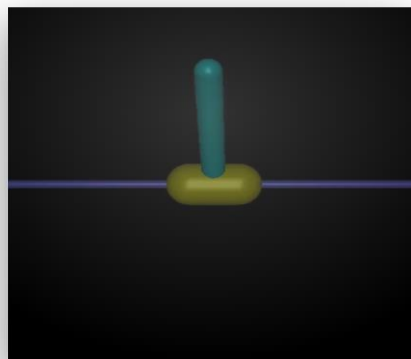
Ant-v2
Make a 3D four-legged robot walk.



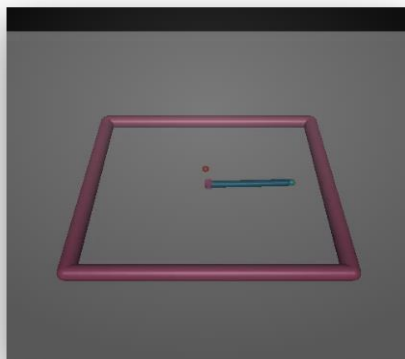
HalfCheetah-v2
Make a 2D cheetah robot run.



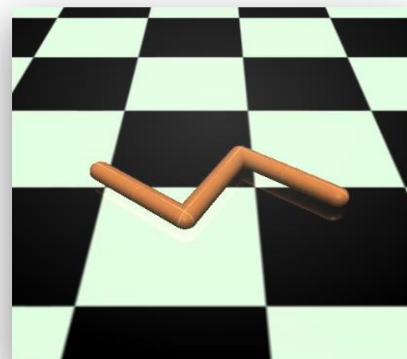
Hopper-v2
Make a 2D robot hop.



InvertedPendulum-v2
Balance a pole on a cart.



Reacher-v2
Make a 2D robot reach to a randomly located target.



Swimmer-v2
Make a 2D robot swim.

2.4 Gym实验环境（7）

➤ Gym环境体验

```
1. import gym    # 引入环境
2. env = gym.make('MountainCar-v0')    # 创建 MountainCar-v0 环境。
3. for episode in range(10):
4.     env.reset()    # 重置智能体状态
5.     print("Episode finished after {} timesteps".format(episode))
6.     for _ in range(1000):    # 进行 1000 次迭代
7.         env.render()    # 渲染
8.         env.step(env.action_space.sample())    # 执行动作。env.action_space.sample()是随机动作选择
9.     env.close()    # 关闭环境
```

2.4 Gym实验环境（8）

➤ Gym中的重要函数

- ◆ **env = gym.make('CartPole-v0')**: 创建CartPole环境;
- ◆ **env.reset()**: 每一个情节结束后, agent初始化;
- ◆ **env.render()**: 图像引擎用来显示环境中的物理图像;
- ◆ **env.step()**: 用来描述agent与环境交互的所有信息:

输入是动作 a , 输出是下一个状态 s_{t+1} , 立即奖赏 r_{t+1} 和是否终止等。

2.4 Gym实验环境（9）

- Gym的使用

使用下面代码可以简单的使用gym环境。

```
import gym
```

```
env = gym.make('MountainCar-v0')
```

```
env.reset()
```

```
env.render()
```

```
env.close()
```

如果想看看别的环境是什么样子的，可以将MountainCar-v0替换成CartPole-v0, Pendulum-v0, Acrobot-v1等，这些环境都来自Env基类。

2.4 Gym实验环境（10）

➤ 搭建扫地机器人的Gym环境

利用Gym搭建一个扫地机器人环境。

该环境描述如下：在一个的扫地机器人环境中，有一个垃圾和一个充电桩，到达[5,4]处机器人捡到垃圾，并结束游戏，同时获得+5的奖赏；左

下角[1,1]处有一个充电桩，机器人到达充电桩可以充电，且不再行走，获得+1的奖赏。环境中间[3,3]处有一个障碍物，机器人无法通过。

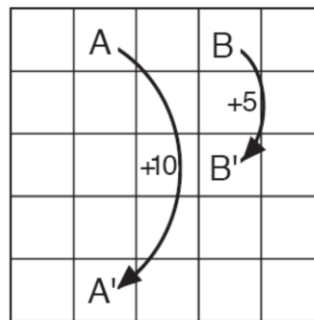
20	21	22	23	24
15	16	17	18	
10	11		13	14
5	6	7	8	9
	1	2	3	4

习题

1. 安装**Anaconda**, **PyTorch**, **Gym**, **Jupyter Notebook**等环境。
2. (编程) 使用**Gym**平台搭建**GridWorld**环境。

如右图，共有**25**个状态，在每个状态下，都可以执行**4**个动作：北、南、西、东。**奖赏设置**：agent试图离开该区域，得到**-1**奖赏，并保持原地不动；

在A点采取任何动作都转移到A'点，得到**+10**的奖赏；在B点采取任何动作都转移到B'点，得到**+5**的奖赏；其他情况奖赏为**0**。



The End