Reinforcement Learning

目录

[1. RL Course by David Silver 1](#_Toc152795774)

[1.1. Introduction to Reinforcement Learning 强化学习简介 1](#_Toc152795775)

[1.1.1. 强化学习与其他学科 1](#_Toc152795776)

[1.1.2. 强化学习的特点 2](#_Toc152795777)

[1.1.3. 强化学习的基本结构 2](#_Toc152795778)

[1.2. Markov Decision Process (MDP) 马尔科夫决策过程 3](#_Toc152795779)

[1.2.1. 马尔科夫链 3](#_Toc152795780)

[2. MuJoCo Simulation 4](#_Toc152795781)

[2.1. 配置 4](#_Toc152795782)

[2.2. 兼容性报错相关解决方案 4](#_Toc152795783)

[2.3. 基本逻辑 5](#_Toc152795784)

[3. Gymnasium 5](#_Toc152795785)

# RL Course by David Silver

## Introduction to Reinforcement Learning 强化学习简介

### 强化学习与其他学科

强化学习是一个涉及多个学科的交叉领域，它主要与六门学科存在交叉，分别是工程学Engineering、计算机科学Computer Science、神经科学Neuroscience、心理学Psychology、经济学Economics和数学Mathematics

·**工程学Engineering**：

工程学与RL的交集在于最优控制理论Optimal Control Theory，作为控制理论的一个分支，最优控制旨在找到解决一个具体问题的最佳工程方案，这也是强化学习的目标。

损失函数Loss Function的概念来源于最优控制理论。

·**计算机科学Computer Science：**

计算机科学与强化学习相关的话题是机器学习Machine Learning，强化学习的智能体策略经常会用到深度神经网络Deep Network。

事实上，强化学习常被视为机器学习的一种。

·**神经科学Neuroscience 与 心理学Psychology：**

强化学习设置奖励的思想借鉴了动物脑神经中的奖赏系统Reward System，即利用多巴胺Dopamine等神经递质提供的愉悦感指导个体做出有利决策。

心理学方面，条件反射Conditioning等动物行为也为强化学习所借鉴。

**·经济学Economics：**

经济学中有有限理性Bounded Rationality的概念，指的是个体因受认知限制、算力等原因限制，无法作出效用最大化的决策。在强化学习中，智能体Agent也会因为对环境的认知局限而难以做出最优的策略。

·**数学Mathematics：**

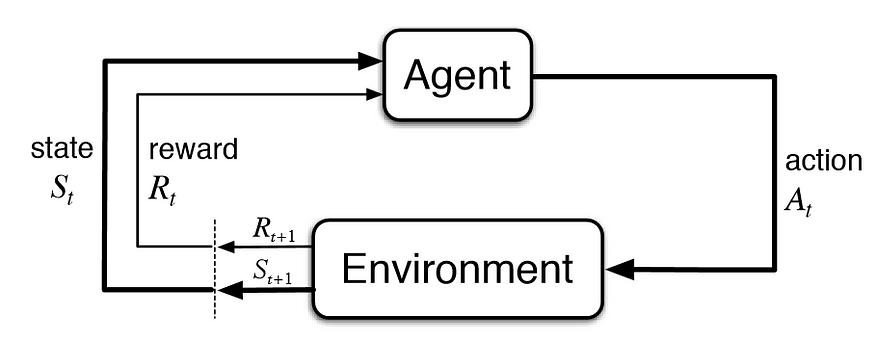
数学领域中与强化学习相关的子领域是运筹学Operational Research，这也是一门有关最优决策的学科。

### 强化学习的特点

强化学习也是机器学习的一个分支，但与传统的机器学习方法（如深度学习Deep Learning、确定性算法）有很大不同：

1. **无监督Unsupervised：**训练过程中没有人类老师或者数据标签Tag指导智能体每个状态下的决策，决策完全由奖励Reward来决定，常具有一定的随机性
2. **反馈滞后Delayed Feedback：**Agent不一定每一个Step都能接受到Reward反馈，可能很多个Step以后才有奖励。若触发奖励的事件非常少，称为稀疏奖励Sparse Reward
3. **时间的重要性Importance of Time：**一般的机器学习中，数据集的数据之间没有相关性，也没有时间上的先后关系，在概率上属于独立同分布i.i.d。但RL的Agent每一个Action都影响下一个State，因此前后不同State收集的数据具有时间上的关联性。
4. **动作影响数据采集Data acquisition affected by Action：**Agent做出的Action影响下一Step的状态，也影响这一状态下能采集到的数据

### 强化学习的基本结构



强化学习的两大主体是智能体Agent和环境Environment (Env)，Agent是训练的对象，环境是Agent探索的场景，也是互动的对象。

在一步Step中，根据当前状态，智能体会从环境接收到一个观测Observation和奖励Reward，根据内部策略Policy产生一个动作Action作用与环境，并进入下一个状态。

奖励Reward是环境产生的标量信号，用于衡量智能体当前Step的好坏，而智能体的目标就是在与环境的交互中取得最大的累积奖励Cumulative Reward。一切强化学习问题都可以归结成对累计奖励最大化问题（Reward Hypothesis）

智能体在过去的所有观测、奖励、动作的集合称为历史History，这是智能体决定动作、环境决定奖励、观测的依据。但一般而言，完整的历史数据过于冗长，不便于处理，因此这里提出状态State的概念。

在RL领域，状态State特指智能体用于决定下一步动作，环境用于决定下一步奖励的信息，它是历史的一个函数。

环境状态被环境用于决定下一步输出的观测和奖励，它对于智能体来说不一定是已知的，即使已知，也可能包含很多对决策无用的信息。

智能体状态是智能体用于动作决策的信息，也是强化学习算法可以利用的信息。它可以是历史的任意函数。

若强化学习过程中任意一个状态都包含了所有的历史信息，就意味着此时的决策并不需要特意访问过去的状态，这种状态称为信息状态Information State或马尔科夫状态Markov State，这样的系统就具有了马尔科夫性质Markov Property。一般的强化学习算法都假定了系统的马尔科夫性质，这避免了智能体决策过程中冗长的历史状态访问，确保智能体仅依靠当前状态即可找到最优策略。

以强化学习控制的无人机为例，某一Step无人机的位置、角度、速度、角速度四个值就组成一个马尔科夫状态，但若不知道速度或角速度，仅通过位置信息不能判断下一个状态，必须还要考虑上一个状态的位置，这种状态就不是马尔科夫状态。

## Markov Decision Process (MDP) 马尔科夫决策过程

大多数强化学习问题的环境都基于马尔可夫性质，可以用马尔科夫决策过程概括之。最简单的马尔科夫过程仅考虑状态和状态之间的转移概率，称为马尔科夫链Markov Chain。

无论是环境完全可测Observable还是部分可测的强化学习问题都可用马尔可夫过程描述，包括只有一个状态的多臂赌博机问题Multi-Armed Bandit。

一个马尔科夫状态：

假设存在个马尔科夫状态，它们两两之间可以定义状态转移概率State Transition Probability。

从状态转移到状态的概率是：

状态转移概率矩阵State Transition Probability Matrix则描述任意两个状态之间的转移概率，每一行描述同一个出发点转移到其他状态的概率，因此同一行的概率之和为1：

### 马尔科夫链

马尔科夫链由状态和转移概率两部分组成，其中是有限个状态的集合，是描述中所有状态的转移概率矩阵。若画成流程图，马尔科夫链很像是带有转移概率的有限状态机Finite State Machine。

# MuJoCo Simulation

MuJoCo是Roboti LLC公司开发的一款基于C++的机械仿真软件，是目前最流行的强化学习仿真软件之一。2022年被Google Deepmind收购以后，MuJoCo成为开源免费软件，但新旧版之间出现了不少兼容性问题。

·Roboti LLC公司旧官网（2.0.0版本以前）：<https://www.roboti.us/index.html>

·新版本Github下载（2.0.0版本以后）：<https://github.com/google-deepmind/mujoco/releases>

·新MuJoCo官网：<https://mujoco.org/>

原版的MuJoCo基于C++，而大多数的机器学习开发项目使用Python，因此一般在Python上编写控制代码，但需要mojoco\_py库作为中继才能在Python中控制MuJoCo。

最新版本的mujoco\_py已经停止了对Windows系统的支持，因此在Windows系统上能用的mujoco\_py对应MuJoCo版本停留在了2.1.0。

## 配置 – Windows 10

配置过程主要的难点在于两点：兼容性和版本问题、文件读取问题。Python环境相关的配置略过。

1. 创建.mujoco文件夹：在C盘的User/Omen（Omen是用户名）文件夹下创建一个.mujoco文件夹，将解压后的整个mujoco210文件夹放入其中；
2. 添加环境变量：在系统变量中添加名为MUJOCO\_PY\_MUJOCO\_PATH的变量（注意！变量名只能是这个），指向mujoco文件所在的路径C:\Users\OMEN\.mujoco\mujoco210
3. 安装mujoco\_py库：pip install mujoco\_py

## 兼容性报错相关解决方案 – Windows 10

1. 降低cython库的版本：mujoco\_py与3.0及以上版本的cython不兼容，需要pip install "cython<3"（Cython是一个基于Python的C语言生成器，实现代码执行的高效性）
2. 在每个脚本前加入以下代码：

|  |
| --- |
| import os  os.add\_dll\_directory("C://Users//OMEN//.mujoco//mujoco210//bin") |

否则会报错：“DLL load failed while importing cymj”

（以上内容编写日期2023/11/30）

## 配置 - Ubuntu

## 基本逻辑

# Gymnasium

Gymnasium是一个针对单智能体场景的强化学习测试环境库，这些环境包括雅达利Atari的经典游戏场景、基于Mujoco的机械模型以及一些常见的控制模型（倒立摆）。

这些环境在库中统一封装为Env类，有相似的step()、reset()、close()方法作为通用接口，且定义了基本的状态State、动作Action以及奖励Reward，对于强化学习领域的入门非常方便。

·Gymnasium文档：<https://gymnasium.farama.org/>

·PyTorch文档：<https://pytorch.org/docs/stable/torch.html>

·Numpy文档：<https://numpy.org/doc/stable/reference/arrays.ndarray.html>

## Env – 环境封装类

Env类是Gymnasium内置的强化学习环境的统一封装格式，