



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

人工智能实验指导书

实验 3-强化学习

2024 春

目录

| | |
|-------------------------|---|
| 人工智能实验指导书 | 1 |
| 1、 实验目的 | 3 |
| 2、 实验内容 | 3 |
| 3、 实验简介 | 3 |
| 3.1 Gym 库 | 3 |
| 3.2 CartPole | 4 |
| 3.3 深度 Q 网络 (DQN) | 5 |
| 4、 实验步骤 | 5 |
| 5、 实验结果提交 | 5 |
| 6、 参考资料 | 7 |

1、 实验目的

本实验旨在实现以下目标：

1. 理解强化学习的基本原理，通过实践探索强化学习的训练过程，从而加深对该领域核心概念的理解。
2. 掌握 DQN 算法，理解其原理和实现方式。
3. 尝试使用不同的优化策略来提高模型的学习效率和性能，通过对比评估不同优化策略对模型训练和性能的影响，进一步优化强化学习模型在实际问题中的表现。


2、 实验内容

使用 DQN (Deep Q-Network) 训练 Agent，以实现 CartPole 环境的强化学习。尝试使用不同的优化策略，以提高 Agent 的学习效率和性能。

3、 实验简介

3.1 Gym 库

Gym 是由 OpenAI 开发的一款用于强化学习的 Python 库，提供了 Atari、MuJoCo、Box2D 等环境，包括了经典的控制问题、连续控制问题和各种强化学习任务。Gym 提供了一致的 API 接口，使得开发者可以轻松地通过 API 与环境交互，并且在不同的环境进行强化学习，对应环境等信息可参考网址 [Gymnasium Documentation \(farama.org\)](https://gymnasium.farama.org/)


 Gymnasium Documentation

Q Search

INTRODUCTION

Basic Usage

Compatibility with Gym
v21 to v26 Migration Guide

API

Env

Register and Make

Spaces

Wrappers

Vector

Utils

Experimental

ENVIRONMENTS

Classic Control

Box2D

Toy Text

MuJoCo

Atari

Third-Party Environments

TUTORIALS

Initializing Environments

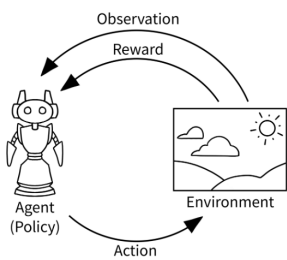
Initializing environments is very easy in Gymnasium and can be done via the `make` function:

```
import gymnasium as gym
env = gym.make('CartPole-v1')
```

This will return an `Env` for users to interact with. To see all environments you can create, use `gymnasium.envs.registry.keys()`. `make` includes a number of additional parameters to adding wrappers, specifying keywords to the environment and more.

Interacting with the Environment

The classic "agent-environment loop" pictured below is simplified representation of reinforcement learning that Gymnasium implements.



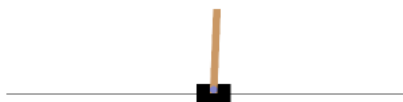
This loop is implemented using the following gymnasium code

```
import gymnasium as gym
```

v0.29.0 (latest)

3.2 CartPole

CartPole 环境是 Gym 库中的一个经典强化学习环境。在 CartPole 环境中，一个小车通过左右移动来平衡一个倒立的杆子。



环境的状态由四个连续值组成，分别是小车的**位置**、**速度**、**杆子的角度**和**角速度**。

| Num | Observation | Min | Max |
|-----|-----------------------|-----------------------------------|---------------------------------|
| 0 | Cart Position | -4.8 | 4.8 |
| 1 | Cart Velocity | -Inf | Inf |
| 2 | Pole Angle | ~ -0.418 rad (-24°) | ~ 0.418 rad (24°) |
| 3 | Pole Angular Velocity | -Inf | Inf |

Agent 的动作空间是离散的，可以选择**向左**或**向右**推动小车。

- 0: Push cart to the left
- 1: Push cart to the right

Agent 需要学会通过观察环境状态并选择合适的动作来保持杆子的平衡，以

获得尽可能长的平衡时间。Agent 可以通过观察环境状态和奖励信号来调整自己的策略，以最大化长期累积奖励。这个简单而直观的环境使得研究者可以快速验证和比较不同强化学习算法的效果，很适合初学者。

详细内容请阅读：[CartPole](#)

3.3 深度 Q 网络 (DQN)

DQN 是一种基于深度学习的强化学习算法，用于解决离散动作空间下的 Q 学习问题。DQN 通过神经网络逼近 Q 值函数，实现 Agent 的决策过程。

详见理论课课件。

4、 实验步骤

0、可申请集群上带 GPU 的环境，运行速度更快。

1、 配置运行环境

(1) 安装 Gymnasium :

```
pip install gymnasium
```

(2) 安装 Matplotlib:

```
pip install matplotlib
```

(3) 安装 Pytorch:

访问：<https://pytorch.org/get-started/locally/> 根据自己机器的配置安装相应的 pytorch 版本。

2、 运行 reinforcement_q_learning.ipynb 并查看结果。

3、 优化代码，可从神经网络结构的优化、超参数调优、优化经验回放区、奖励函数的设计、探索策略的设计等方面着手。要求至少进行 4 项优化。

4、 分析比较各种优化方法带来的结果。

5、 实验结果提交

➤ 提交物:

- 代码
 - 实验报告（一定要用模板）
 - 打成 zip 包提交
- 作业提交平台: <http://grader.tery.top:8000/#/courses>
- 截止时间: 见 PPT

6、 参考资料

- 1、 [Reinforcement Learning \(DQN\) Tutorial — PyTorch Tutorials 2.3.0+cu121 documentation](#)
- 2、 [DQN 算法 \(boyuai.com\)](#)