《自然语言处理基础与大模型》作业一

姓名：谷绍伟　　学号：202418020428007

倒立摆强化学习任务

选取倒立摆为实验对象，由直流电机驱动倒立摆在垂直平面内进行旋转，记录起始点为 $-\pi$，平衡位置为 0，通过强化学习的方法学习控制策略，使倒立摆能在电机的驱动下保持在平衡位置。

倒立摆的相关物理参数后控制要求见实验指导书。

1　实验思路

在倒立摆强化学习的任务中，倒立摆的物理参数后时间动力学模型已近给定，离散时间动力学模型和奖励函数也已知，同时控制电压也被离散为 $\{-3, 0, 3\}$ 三个值，因此选定学习方法即可进行实验。

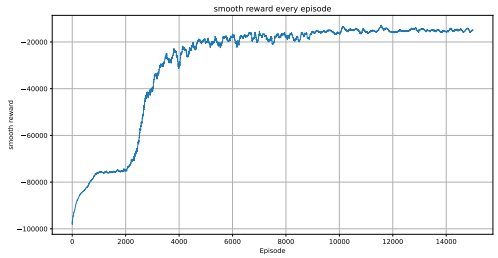
本报告中选择 Q-学习方法，智能体根据 $\epsilon-$ 贪心策略产生动作数据：$a_t \sim \epsilon - greedy(Q$ ，即按照一定的概率进行探索，其余则根据学习经验选择。Q 学习中对 Q 表的更新根据以下公式：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma max_a Q(s_{t+1}, a') - Q(s_t, a_t))$$

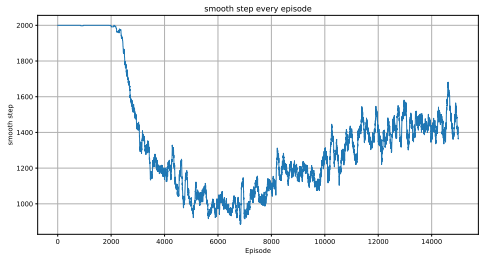
在使用中，需要对连续时间动力学模型进行离散化。而倒立摆的转角和角速度也是连续值，同样需要进行离散化，在实验中选择将角度化角速度分别离散化为 200 个值，动作选择已经被离散化为三个值，因此总的 Q 表大小为 $3 \times 200 \times 200$。由于离散会带来误差，且倒立摆难以直接停止在目标位置，实验中还设置了角度化角速度误差限制，当最终位置在限制内时，认为达到了学习目标。

2　实验及结果

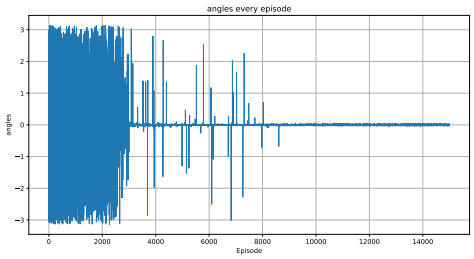
使用 python 实现倒立摆的 Q 强化学习。设置回合数为 15000，每个回合内的最大探索步数为 2000，每回合初始位置为 $-\pi$，设置角度误差上限为 $0.01rad/$，角速度误差上限为 $0.1rad/s$, 初始学习率为 0.5, $\epsilon = 0.8$, 同时为学习率后 ϵ 设置衰减因子为 0.9995，进行强化学习，记录学习中每个回合的总奖励结果后最终的误差，结果如图 1所示。每个回合的奖励函数累计值、所需步数和最终的角度、误差分别为图 1(a)、图 1(b)、图



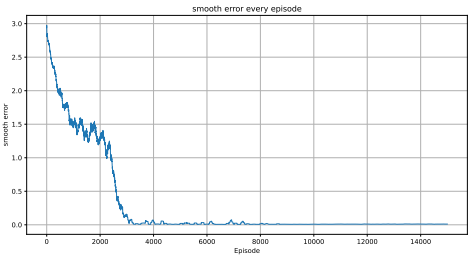
(a) 奖励累计值



(b) 每回合步数



(c) 每回合结束时角度



(d) 结束时误差

Figure 1: 倒立摆训练过程数据记录

1(c)、图 1(d)所示。为了方便观察，对结果中的奖励函数累计值、所需步数和最终误差进行了滑动平均，滑动窗口大小为 100。

从结果可以看出，随着训练回合数的增加，奖励累计值越来越大，符合强化学习最大化奖励的目标。同时每回合结束时的角度也越来越接近目标位置，误差逐渐减小。但每回合步数经历了逐渐减小再缓慢增大的过程，根据角度和误差值推断，前期步数的减小可能是由于随着训练的进行，智能体快速学习到接近平衡位置的方法，但训练次数再次增加时，模型更多的步数，能更精细地使倒立摆稳定在目标位置附近，震荡逐渐减小。

训练完成后，我们还对学习到的 Q 表进行了评估。并利用动画对倒立摆控制过程进行了可视化，可视化结果如图 2所示。

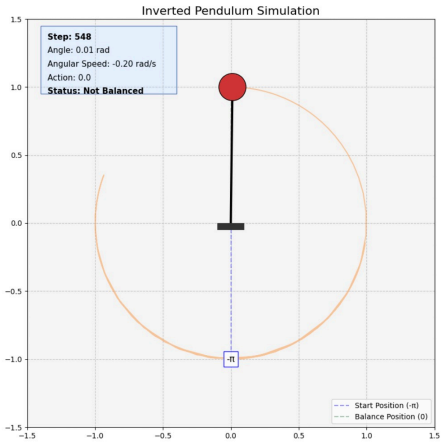


Figure 2: 倒立摆控制可视化

同时，记录了评估过程中的角度变化后误差值，结果如图 3所示。

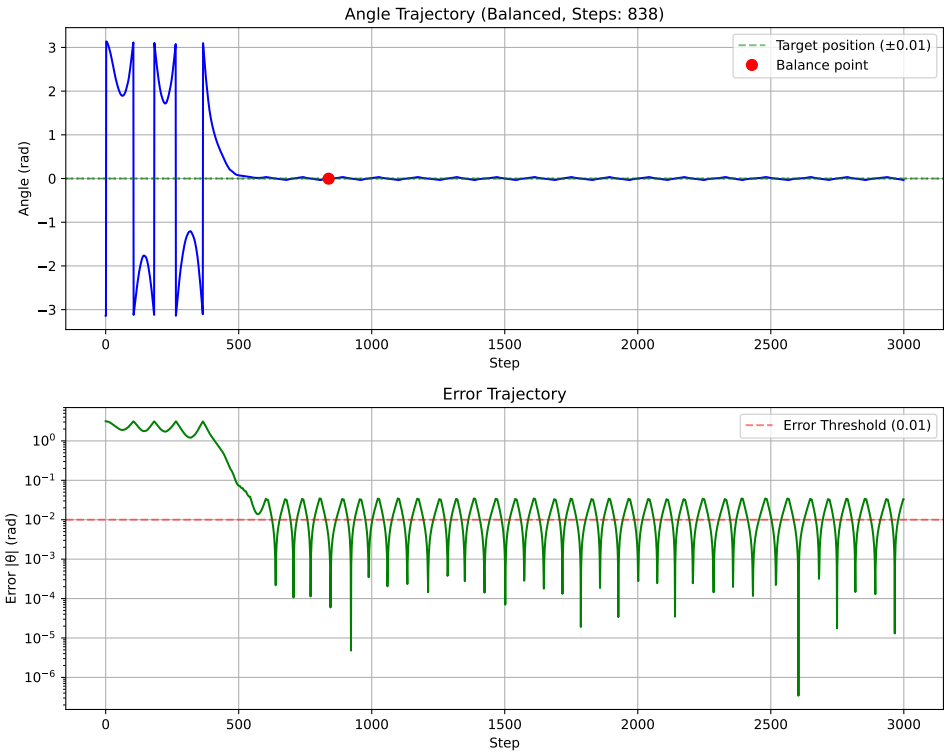


Figure 3: 模型评估结果

从评估结果中可以看出，智能体通过多次摆动累计能量后到达最高点，到达稳定点后在稳定点附近微小震荡，可视为智能体已经通过强化学习达到了控制目标。

3 讨论

为了进一步研究每回合步数限制和角度离散化间隔对倒立摆 Q 学习效果的影响，设置了两组对比实验：1）保持每回合步数限制为 2000，分别设置离散化参数为 100，400；2）保持离散化参数为 200，分别设置最大步数限制为 1000 和 500，分别训练智能体，训练过程中的收敛情况如表 1所示。

Table 1: 不同离散化参数和最大步数下训练结果

离散参数	步数限制	收敛情况	平均步数
100	2000	✓	571
400	2000	✗	-
200	1000	✗	-
200	500	✗	-

说明在进行 Q-学习时，过于细致的 Q 表可能会导致训练失败，智能体无法学习到

足够的信息，同时还需要选取合适的步数限制避免智能体在一个回合中没有足够的步数来充分学习。

4 代码及相关文件

代码文件见压缩包中 Inverted_pendulum.py 和 Q_learning.py 文件，其中 Inverted_pendulum.py 是智能体，Q_learning.py 是训练和测试文件。

optimal_policy.npy 为学习到的 Q 表。

评估结果可视化动画为 pendulum_animation_complete.gif

实验报告为 report.pdf