文章编号: 1007-1423(2022)02-0067-06 DOI: 10.3969/j.issn.1007-1423.2022.02.010

基于改进Q-Learning的路径规划算法

张小月，韩尚君，陶青川，余艳梅

（四川大学电子信息学院，成都 610065）

摘要： Q-Learning是一种经典的强化学习算法。然而，它存在着收敛速度慢的缺点，而且由于存在着一定概率的探索，该算法可能会浪费很多时间。为解决这些问题，在Q-learning基础上引入初始化Q表格，同时提出“探索引导”方法。仿真实验结果表明，该改进可以减少训练次数，加快收敛速度，例如在Gym库中的悬崖寻路场景中，改进的方法能缩短30%的训练次数。

关键词： 强化学习；Q-Learning；路径规划

0 引言

路径规划一直是机器人领域的重点问题，也是未来研究的热点。常见的路径规划算法有Dijskra、RRT[1]、A*、蚁群算法[2]，可以在连续或是离散的空间中实现寻路。近年来机器学习兴起，由Watkins提出的Q-Learning算法[3]又重新回归人们的视野，该方法在数字动画[4]、游戏、个性化推荐、无人驾驶[5-7]等众多领域有着广泛的应用。而RRT、A*等方法有着计算量大、实时性差的缺点，Q-Learning通过训练能快速寻找到最短路径，它在路径规划上有着天然的优势[8-9]。

强化学习的灵感来源于心理学，智能体从与环境的交互中学习来获取经验，这个经验会指导智能体根据环境的状态执行动作，并根据环境的反馈增加新的经验。本文对经典的强化学习算法Q-Learning算法进行改进，优化Q表格初始值，使用“探索引导”，解决了Q-Learning收敛速度慢的问题。

1 Q-Learning算法理论

1.1 强化学习组成结构

强化学习主要是由智能体和环境构成，其通信渠道有奖励、状态和动作。强化学习的框架如图1所示，S_t是环境在t时刻的状态，A_t是智能体在环境中t时刻执行的动作，A_t使得环境的状态变为S_{t+1}，在新状态下环境产生了新的反馈R_{t+1}，智能体根据S_{t+1}和R_{t+1}执行新的动作A_{t+1}，如此循环往复直到迭代结束。

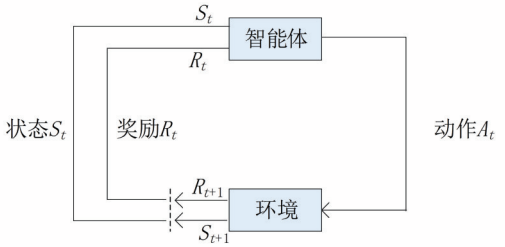


图 1 强化学习的框架

1.2 马尔科夫决策

假设强化学习的求解过程满足马尔科夫属性即无后效性，系统的下一个状态只与当前的状态有关，与之前的更早的状态无关。马尔科夫决策过程(MDP)[10]的四元组是<S,A,P,R>，S是状态集合，S_t表示t时刻的状态，A是动作集，A_t表示t时刻的动作，R是奖励函数，R_t=$R(S_t, A_t)$表示在状态S_t下执行A_t后智能体获得的奖励，P是状态转移概率，记作$P(S_{t+1}, R_t \mid S_t, A_t)$，表示$t$时刻状态为$S_t$执行动作$A_t$后，获得奖励$R_t$且下一个状态为$S_{t+1}$的概率分布。完整的马尔科夫决策模型如图2所示。

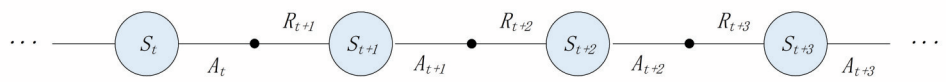


图2 马尔科夫决策链

因为现实生活中，奖励往往是延迟的，不能只考虑当前的单步收益，并且还需要考虑未来的奖励。想要使未来收益之和更加合理，距离当前越远的收益，对现在的影响越小，引入折扣因子γ，是一个介于[0,1]的常数。使用G_t来表示未来累积奖励，表达式如下：

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

1.3 探索与利用

对于免模型的环境，探索和利用是相辅相成的，想要获取更多的环境信息就需要探索，想要提高奖励、制定最优策略需要进行利用，两者同样重要。强化学习算法训练时的轮数是有限的，探索的占比增加会导致利用的次数减少，所以需要权衡探索与利用的使用比例。

ε-贪婪算法用于表示探索与利用的行为，以ε概率进行“探索”，即智能体随机选择一个动作，以1-ε概率进行“利用”，选择奖励最大的动作作为下一个要执行的动作，这时会利用已知的环境和奖励信息。数学表达式如下：

$$A = \begin{cases} Random\ Action, & P = \varepsilon \\ \arg\max_a Q(a), & P = 1 - \varepsilon \end{cases} \quad (2)$$

ε参数的选择会影响收敛速度，当ε的值较大时，探索的机会更多，模型的收敛速度快；当ε的值较小时，利用的机会更多，模型会更加稳定，但收敛速度比较慢。若动作对应的奖励不确定性较小、概率分布较宽时，建议使用较大的ε值；若动作对应的奖励不确定性较小、概率分布较集中时，少量尝试就能接近真实的奖励，可以使用较小的ε值。ε的值通常取一个常数，比如0.1或0.01。

1.4 Q-Learning算法

时序差分算法Q-Learning使用Q表格记录动作价值，下一个时刻的Q_{t+1}值会影响当前时刻的Q_t值，使得Q_t逼近未来总收益G_t。Q-Learning在选择动作时会默认选择最优策略，即最大Q值对应的动作，但这可能与下一时刻的实际执行动作不一致，所以说Q-Learning是一种偏向最优Q的策略。动作值函数$Q(S_t, A_t)$[1]如下：

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max Q(S_{t+1}, a) - Q(S_t, A_t)\right] \quad (3)$$

根据ε-贪婪算法来决定使用哪一个动作，1-ε的概率执行最大Q值对应的动作，ε的概率随机执行一个动作。执行完动作A_t后，智能体与环境交互，获得下一个状态S_{t+1}和奖励R_{t+1}。在更新Q值时，时序差分目标使用动作值函数的最大值对应的动作a，在策略上想要获得最大收益，但实际上会执行动作A_t。

2 基于改进Q-Learning的路径规划算法

2.1 经典Q-Learning实例

本文借用百度飞桨(PaddlePaddle)的开源Q-Learning项目，对该项目中的Q-Learning算法进行改进。如图3的栅格模型[11-12]所示，智能体是乌龟，起点为左下角乌龟所在位置，终点为右下角格子。OpenAI的gym库提供了常用的强化学习环境，这里解决的是Gym库中的悬崖寻路问题，黑色区域是悬崖，每次乌龟掉落悬崖后会重新回到上一步的位置。Q表格是一个大小为12×4且初始值为0的矩阵，随后使用ε-贪婪算法执行动作，更新状态和奖励，使用动作值函数(公式3)更新Q表格。训练次数设置为1500次，每次乌龟到达终点就会重回起点开始新一轮的训练，测试结果如图3的路线所示，智能体会根据Q值选择最优路径。

对于每轮的奖励，初始值是0，乌龟每走一个白格子，奖励减1，每走一个黑格子，奖励减100。如果走最短的路径，需要走13步，得到的奖励是-13，也是所有路径中收益最大的情况。

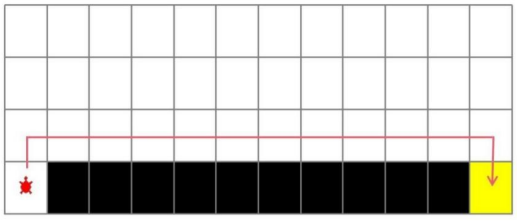


图3　悬崖环境(CliffWalking-v0)

图4是训练迭代次数-步数曲线图，纵坐标表示步数，横坐标表示为训练迭代次数。随着训练次数的增加，步数会逐渐减少到40以下。曲线一直都没有收敛，因为根据ε-贪婪算法有ε概率随机选择动作的情况，这里ε设置了0.1，所以智能体有90%的情况进行利用，10%的情况进行探索。

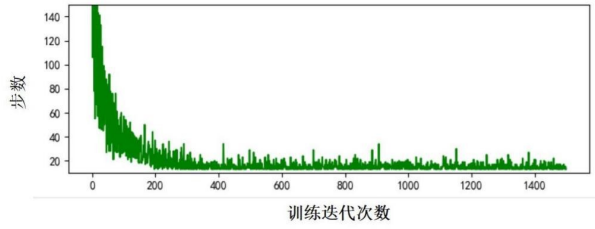


图4　原始的训练次数-步数曲线图

为了方便观察，图5还绘制了训练次数-奖励(相反数)曲线图，Q-Learning算法训练1500次，由于奖励是负数，所以奖励的相反数越小说明实际的奖励越大。图5作为图4的补充，因为智能体无论是走黑格子还是白格子，步数都是增加1，而白格子的奖励是-1，黑格子的奖励是-100，所以使用奖励的相反数作为纵坐标能区分智能体走不同格子的情况。

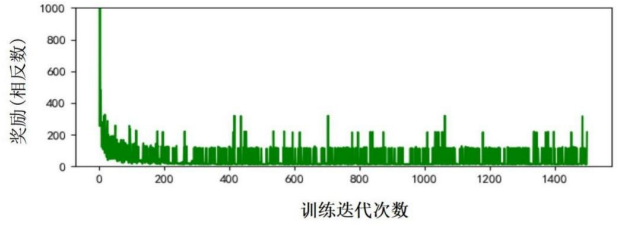


图5　原始的训练迭代次数-奖励(相反数)曲线

2.2　改进Q-Learning的方法

本文分别从Q表格的初始值和探索训练过程两个不同的角度来改进Q-Learning，以此来减少训练次数。对Q-Learning的具体改进如下：一是引入初始化Q表格的方法并使用欧氏距离或曼哈顿距离修改Q表格的初始值，二是提出使用“探索引导”来避开障碍物。

2.2.1　初始化Q表格

传统的Q-Learning会将Q值初始化为0，想要智能体选择最短路径，可以初始化Q值为当前位置到目标位置的距离的倒数或相反数[13]。离目标越近Q值就越大，智能体在训练初期更容易朝着目标位置的方向前进。

假设两个点的坐标分别为(x_1, y_1)和(x_2, y_2)，那么两点之间的欧式距离$d_o = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$，两点之间的曼哈顿距离为$d_m = |x_1 - x_2| + |y_1 - y_2|$。设乌龟执行动作之后会获得新的坐标位置$A$，计算$A$点与终点的欧式距离$d_o$和曼哈顿距离$d_m$。为了证明如何初始化$Q$表格能使得乌龟更快找到最优路径，设计了三组实验：(a)使用$1/d_o$即欧氏距离的倒数来初始化Q表格。(b)使用$1/d_m$即曼哈顿距离的倒数来初始化Q表格。(c)使用$-d_o$即欧氏距离的相反数来初始化Q表格。

2.2.2　探索引导

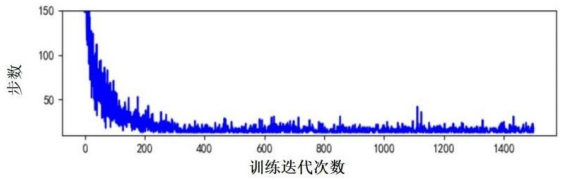
针对10%的探索的情况，智能体有可能会多次选择掉入悬崖(黑格子)的动作，这种情况应当避免。本文提出的“探索引导”目的是在探索的时候引导智能体尽量选择无障碍的路线，方法具体内容如下：当智能体在前几次掉入悬崖后，当前位置对应的Q值会远小于Q表格中的其他Q值。在10%随机动作之前加一个判断，排除Q值较小即容易掉入悬崖的动作，这样就可以加快奖励收敛的速度。

2.3　实验结果

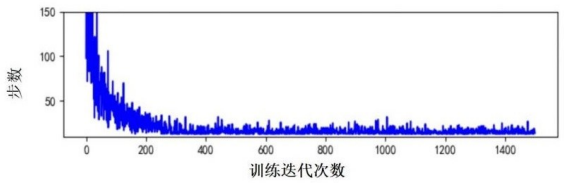
2.3.1　初始化Q表格的实验结果

2.2.1设计的三组实验的训练迭代次数-步数

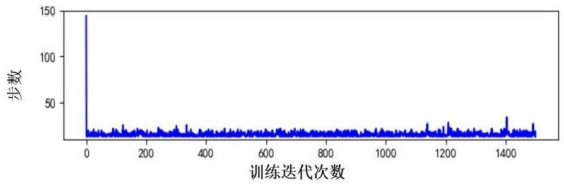
曲线图如图6所示。可以直观地看出图6(c)即Q表格初始值为$-d_o$的情况表现最好，基本上训练几次，步数就减少到25以下了。另外实验两组都是在训练次数快到400的时候步数才减到40以下，(a)比(b)稍微好一点，但不明显。



(a) $Q=1/d_o$



(b) $Q=1/d_m$



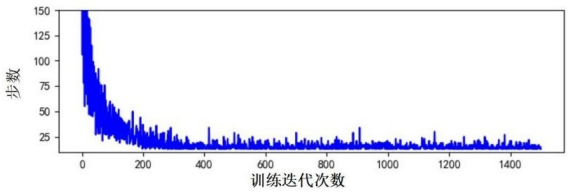
(c) $Q=1/-d_o$

图6 Q表格不同初始中对应的训练迭代次数-步数曲线

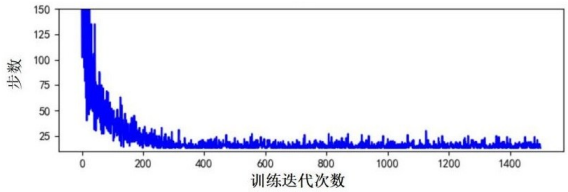
手动设置训练次数，在减少训练次数后，可以得到获取最优路径的最少训练次数，Q表格初始化为0时，至少需要训练280次；Q表格初始化为$1/d_o$时至少需要训练260次；Q表格初始化为$1/d_m$时至少需要训练310次；Q表格初始化为$-d_o$时至少需要训练20次。由于存在10%的概率探索，最少训练次数具有偶然性，但无论如何，使用$-d_o$来初始化Q表格可以减少训练次数是毋庸置疑的。由于Q值初始化为曼哈顿距离的情况表现不佳，后面的实验只会用到欧式距离。

2.3.2 探索引导的实验结果

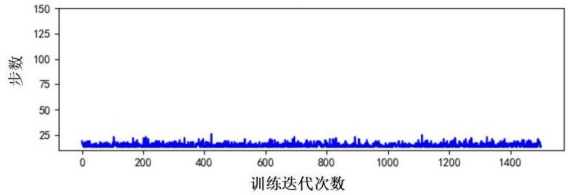
本文设置“探索引导”的阈值为-50，即智能体不会探索Q值小于-50的格子。图8记录了使用“探索引导”后，Q表格不同初始值对应的训练迭代次数-步数曲线图。这一次Q值初始为$-d_o$时，步数在刚开始就收敛到了一个很低的数。只看图7的(a)(b)即Q值初始值分别为0和$1/d_o$时，“探索引导”好像并没有改善的作用，但是从奖励的角度来看，“探索引导”可以减少走黑格子即奖励值为-100的情况。



(a) $Q=0$



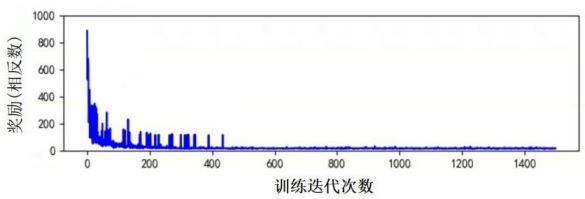
(b) $Q=d_o$



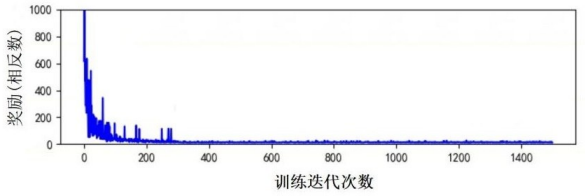
(c) $Q=-d_o$

图7 使用“探索引导”后，Q表格不同初始值对应的训练迭代次数-步数曲线

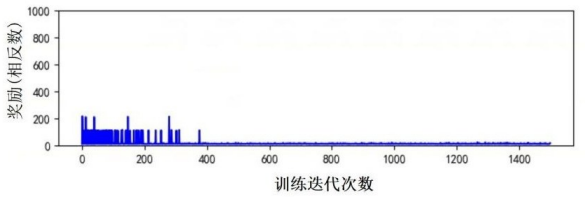
如图8所示，加入了探索引导后，随着训练次数的增加，奖励能够快速收敛。(a)图是Q值初始化为0的时候，“探索引导”使得奖励在训练了450次左右就收敛了。这里收敛的训练次数跟2.3.1中最少训练次数是不一样的，奖励收敛后，进行测试的时候，智能体一定会走最优路径，而最少训练次数是一个不稳定的值，运气好的情况下，在训练次数等于最少训练次数的时候智能体会选择最优路径，但并不是每次都能成功。图8分别是Q初始化为0、$1/d_o$和$-d_o$的情况下，使用了“探索引导”的训练次数-奖励(相反数)曲线图，(b)中奖励收敛速度更快，大概训练300次左右就收敛了，比Q值初始为0的情况训练次数少了30%，(c)图中，奖励的初始值更小。



（a）Q=0



（b）$Q=d_o$



（c）$Q=-d_o$

图8　使用“探索引导”后，Q表格不同初始值对应的训练迭代次数-奖励(相反数)曲线

3　结语

实验结果显示，可以从两个方面来改善Q-Learning：一是Q表格的初始值，使用$-d_o$可以减少训练次数，利用目标点与当前位置的距离作为先验知识，智能体会选择执行离目标点更近的动作；二是使用了“探索引导”，让智能体在多次“碰壁”后能够学习避开障碍的经验，下次探索的时候排除掉落悬崖的动作，从剩下的动作中随机选择。这两个方法能够均能减少训练次数，不仅减少了时间和计算的成本，还提高Q-Learning的效率。

参考文献：

[1] LAVALLE S M. Rapidly-exploring random trees: a new tool for path planning [J]. Research Report, 1999.

[2] 张松灿，普杰信，司彦娜，等. 蚁群算法在移动机器人路径规划中的应用综述[J]. 计算机工程与应用，2020，56(08)：10-19.

[3] WATKINS C, DAYAN P. Technical note: Q-learning [J]. Machine Learning, 1992, 8 (3/4): 279-292.

[4] KUFFNER J J, LATOMBE. Autonomous agents for real-time animation [D]. Phd Thesis Stanford University, 1999.

[5] YOON J, CRANCE C D . Path planning for Unmanned Ground Vehicle in urban parking area [J]. IEEE, 2011：887-892.

[6] 任子玉. 智能车自主避障路径规划研究综述[J]. 软件导刊，2017，16(10)：209-212.

[7] 卫玉梁，靳伍银. 基于神经网络Q-learning算法的智能车路径规划[J]. 火力与指挥控制，2019，44(02)：46-49.

[8] 郭新兰. Q-learning算法下的机械臂轨迹规划与避障行为研究[J]. 机床与液压，2021，49(09)：57-61，66.

[9] 高乐，马天录，刘凯，等. 改进Q-learning算法在路径规划中的应用[J]. 吉林大学学报(信息科学版)，2018，36(04)：439-443.

[10] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: an introduction [J]. IEEE Transactions on Neural Networks, 2005, 16 (1): 285-286.

[11] 王勇. 智能仓库系统多移动机器人路径规划研究[D]. 哈尔滨：哈尔滨工业大学，2010.

[12] 于红斌，李孝安. 基于栅格法的机器人快速路径规划[J]. 微电子学与计算机，2005(06)：98-100.

[13] 于盛. 基于强化学习多无人机路径规划算法研究及实现[D]. 哈尔滨：哈尔滨工业大学，2020.

作者简介：

张小月(1995—)，女，安徽滁州人，硕士，研究方向为计算机应用与路径规划

韩尚君(1998—)，男，四川南充人，硕士，研究方向为计算机应用与路径规划

陶青川(1972—)，男，四川南充人，硕士生导师，副教授，研究方向为模式识别与智能系统

通信作者：余艳梅(1975—)，女，四川广安人，硕士生导师，副教授，研究方向为图像处理，E-mail：yuyanmei@scu.edu.cn

收稿日期：2021-10-11　　**修稿日期：**2021-12-09

Path Planning Algorithm Based on Improved Q-Learning

Zhang Xiaoyue, Han Shangjun, Tao Qingchuan, Yu Yanmei

(College of Electronics and Information Engineering, Sichuan University, Chengdu 610065)

Abstract: Q-Learning is a classic reinforcement learning algorithm. However, it has the disadvantage of slow convergence speed, and due to the existence of a certain probability of exploration, the algorithm may waste a lot of time. To solve these problems, on the basis of Q-learning, the initial Q form is introduced and the "exploration guide" method is proposed. Simulation experiment results show that the improvement can reduce the number of training sessions and speed up the convergence speed. For example, in the cliff pathfinding scene in the Gym library, the improved method can shorten the training episodes by 30%.

Keywords: reinforcement learning; Q-Learning; path planning

(上接第66页)

Research on AGV Path Planning of "Goods-to-Person" System Based on Q-learning

Zhang Xianglai, Jiang Shangrong, Luo Qin

(College of Management, Harbin University of Commerce, Harbin 150000)

Abstract: With the development of intelligent logistics, AGV plays an increasingly important role in modern intelligent warehouse and gives birth to the picking mode of "goods-to-person". With kiva warehouse as the research background, this paper studies the local path planning of single AGV and improves the dilemma of traditional Q-learning "exploration and utilization". By introducing the arcsine function to dynamically adjust the greedy factor epsilon, the randomness of AGV in the early exploration can be avoided. The later reduction ensures the purpose of AGV path finding. The simulation model is established by using the grid method. The simulation results show that the improved Q-learning can find the optimal path in the warehouse, and the running time is reduced by about 28% compared with the traditional Q-learning, which effectively improves the efficiency of the algorithm.

Keywords: Q-learning; AGV; path learning; goods-to-person system