单位代码 10006
学　　号 15231045
分 类 号 TP301

北京航空航天大學
BEIHANG UNIVERSITY

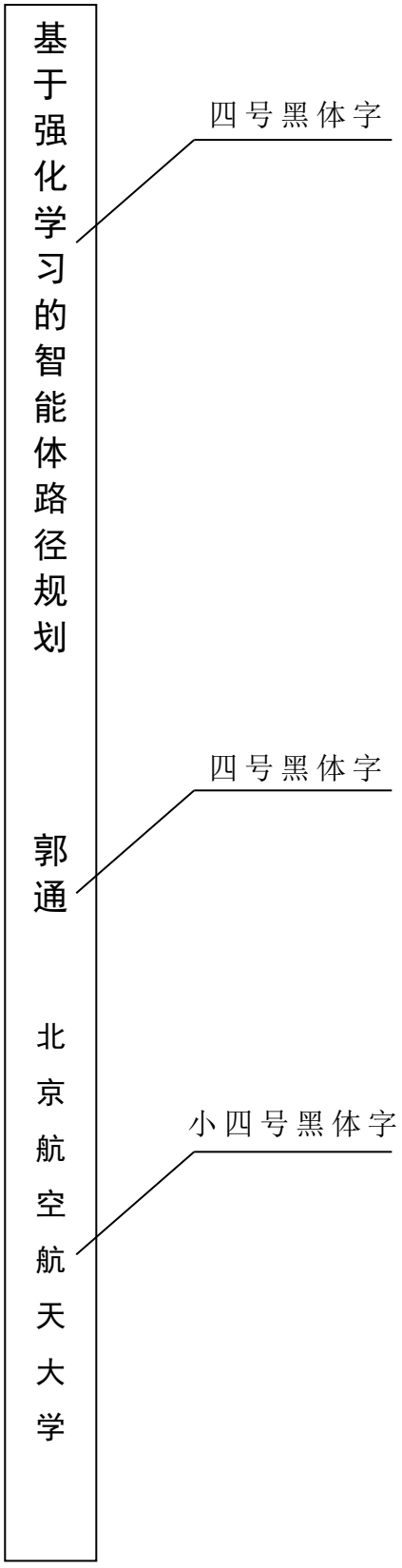
毕业设计(论文)

基于强化学习的智能体路径规划

学习中心名称　电子信息工程学院
专业名称　电子信息工程
学生姓名　郭通
指导教师　杜文博

2019 年 5 月 20 日

论文封面书脊



北京航空航天大学

本科生毕业设计（论文）任务书

Ⅰ、毕业设计（论文）题目：

基于强化学习的智能体路径规划

Ⅱ、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

此次研究主要基于 Python Pygame 可视化模块，对无人机路径规划场景进行建模，和动力学定义；模型训练所用数据，均来自所定义的环境动力学。

此次研究要求首先分析现有深度强化学习算法的不足；其次，在子模型的收敛性和鲁棒性方面进行改进；最后，通过引入分布式训练结构，提高整体模型的鲁棒性。实验结果表明，在信息不完备和场景高动态约束下，所提出的模型仍然具备一定的路径规划能力。

Ⅲ、毕业设计（论文）工作内容：

第一，对强化学习的数学原理进行了详细的介绍，为建立无人机路径规划数学模型做基础。

第二，引入深度强化学习解决经典强化学习维度灾难问题；利用双网络结构和优先经验回放技术，提高神经网络的收敛性。并进行可视化和量化实验，分析改进的效果。

第三，提出一种分布式计算模型，以分治思想将复杂问题的优化转

化为简单子问题的优化，提高整体模型的鲁棒性。并进行可视化和量化实验，分析改进的效果。

Ⅳ、主要参考资料：

[1] Ziv J, Lempel A. A universal algorithm for sequential data compression[J]. IEEE Transactions on information theory, 1977, 23(3): 337-343.

[2] Ferguson D, Stentz A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2): 79-101.

[3] Kavraki L, Svestka P, Overmars M H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[M]. Unknown Publisher, 1994.

[4] Cortes J, Siméon T, Laumond J P. A random loop generator for planning the motions of closed kinematic chains using PRM methods[C]//Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). IEEE, 2002, 2: 2141-2146.

[5] Dapper F, Prestes E, Nedel L P. Generating steering behaviors for virtual humanoids using bvp control[C]//Proc. of CGI. 2007, 1: 105-114.

[6] Urmson C, Simmons R. Approaches for heuristically biasing RRT growth[C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). IEEE, 2003, 2: 1178-1183.

[7] Kothari M, Postlethwaite I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees[J]. Journal of Intelligent & Robotic Systems, 2013, 71(2): 231-253.

[8] Yershova A, LaValle S M. Motion planning for highly constrained spaces[M]//Robot motion and control 2009. Springer, London, 2009: 297-306.

高等理工 学院（系） 电子信息工程 专业类 152312 班

学生 郭通

毕业设计（论文）时间： 2018 年 10 月 26 日至 2019 年 05 月 26 日

答辩时间： 2019 年 05 月 31 日

成　　绩：_____

指导教师：_____

兼职教师或答疑教师（并指出所负责部分）：

_____

_____

_____系（教研室） 主任（签字）：_____

注：任务书应该附在已完成的毕业设计（论文）的首页。

独创性声明

我在此郑重申明，本人所提交的毕业设计（论文），是在导师指导下由本人独立完成的研究成果，对文中所引用他人的成果，均已进行了明确标注或得到许可。毕业设计（论文）中不包含任何其他个人或集体已经发表或撰写过的研究成果，不包含他人已申请毕业证书（学位）或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示了谢意。

本人完全意识到本声明的法律结果，如有不实之处，由本人承担一切相关责任。

学生签名：

时　　间：　　年　　月　　日

基于强化学习的智能体路径规划

学　　生：郭通
指导老师：杜文博

摘　要

随着计算水平的提高，无人机被广泛应用于社会生产和人们日常生活中，为人们生活带来巨大的便利。无人机路径规划是无人机技术里最基本的问题。无人机低空空域普遍存在环境高动态、随机化和信息不完备的特点，如何在这些约束下进行较为鲁棒的无人机路径规划具有重要的现实意义。

首先，本文做了较为全面的文献综述，对比各类经典路径规划算法的优缺点和适用场景，分析经典路径规划算法的局限性，以及在信息不完备、复杂动态场景中，利用现有深度强化学习算法进行路径规划时，尚存在的收敛性慢、鲁棒性差的问题。

其次，本文在现有深度强化学习算法的基础上，采用了分治优化思想，提出一种分布式训练模型，将一个复杂深度神经网络的优化问题转化为若干个简单子神经网络的优化问题。通过引入另一个神经网络产生监督数据，改善了深度强化学习算法中利用当前神经网络参数产生监督数据而产生的训练波动问题。通过样本对目标函数更新梯度大小来分配样本权重的方法，改善了随机选取训练样本产生的训练效率低下问题。

最后，通过计算机仿真结果与对比分析，表明本工作提出的一种分布式的优化计算模型，在信息不完备、场景高动态的约束下，较现有的深度强化学习算法在收敛性和鲁棒性有一定的提升。

在附录中，本文还将给出一些深度强化学习的模型解决方案，供参考之用。

关键词：无人机路径规划，强化学习，深度神经网络

Agent Path Planning: A Reinforcement Learning Approach

Author: GUO, Tong

Tutor: DU, Wenbo

Abstract

With the improvement of computing level, UAV is widely used in social production and people's daily life, bringing great convenience to people's lives. UAV path planning is the most basic problem in UAV technology. The characteristics of high dynamic environment, randomization and incomplete information are ubiquitous in UAV low altitude airspace. How to carry out robust UAV path planning under these constraints has important practical significance. This work mainly focuses on the optimization and improvement of the method.

Firstly, this work makes a comprehensive literature review, analyses various path planning algorithms and their limitations. Secondly, on the basis of the existing deep reinforcement learning algorithm, this work proposes a distributed training model using the idea of dividing and conquering optimization. By introducing another neural network to generate supervisory data and updating the gradient of the target function to distribute the weight of the samples, the problem of training fluctuation in deep reinforcement learning algorithm is improved. Finally, through the computer simulation results and comparative analysis, it is shown that the proposed distributed optimization computing model has better convergence and robustness than the existing deep reinforcement learning algorithm under the constraints of incomplete information and high dynamic scene.

In the appendix, some model solutions of deep reinforcement learning are also given for reference.

Key words: UAV Path Planning, Reinforcement Learning, Deep Neural Network

目 录

1 绪论

无人机近年来被广泛应用于国防、民生等社会发展和人民生活的各个层次。由于无人机的灵活性、可扩展性，在诸多社会生产任务中广受关注。但同时由于空域环境的复杂与高动态，为无人机的安全运行带来了巨大挑战，如何安全、高效运行无人机系统是无人机实际工程中最为重要的环节，而无人机路径规划是问题的基础。

路径规划问题的定义是指，在考虑约束条件下，在两点间通过数学建模建立目标函数，通过优化计算得出一个优化解。这些约束条件在实际工程中包括如：路径最短、避免与障碍物发生碰撞、最大程度涵盖任务点等。而优化函数一般是根据实际的任务和场景建模建立起的两点间的距离公式。

本工作将从实际工程出发，高度关注实际无人机运行的场景约束，设计一套能够在复杂环境、多约束条件下能够高效和鲁棒地进行路径规划的算法。

1.1 课题背景及目的

随着计算力水平和人工智能领域的长足进步，无人机近年来被广泛应用于各种任务，包括空中监视、空中物流、应急救援、通信组网等。它所具备小型化、便捷化、按需制定等优点，使得无人机可以驾驭多种复杂、高难度任务，在工业界中广受青睐。

空中监视

应急救援

通信组网

空中物流

图 1.1 无人机生产任务示意图

在空中监视任务中，无人机大大减少了人类参与工作的比例，给人们带来了劳动自由。高铁运行管理中，漫长的铁路线存在诸多安全隐患，需要在不同的铁路段安置不同的检查哨，对铁路的耦合性、安全性和耐受性做评估，减少由于铁路本身的安全问题产生重大高铁运行事故的可能性。传统的检查哨方法需要花费巨大的人力物力和财力，在我国西部无人区，资源贫瘠、难以续航供应，这种传统方法的缺点体现的尤为明显。然而随着无人机技术的进步，我们获得了一种更加先进和节省开销的做法：通过无人机高精度的拍摄器件对铁路的必要元素进行拍摄作为原始数据，进而通过先进的算法以及数据处理技术，获得铁路的运行状态信息，反馈给管理中心，以便及时处理安全隐患。

在空中物流任务中，无人机节约了物流配送时间，减少了任务分配的运筹学难题。传统的物流配送，需要根据客户的配送点进行任务分配，尽少地减少由于物流配送产生的交通开销。这种方法在运筹上本身存在难度，除此，一个最大的缺点仍然是由于地面交通的高度结构化和高度秩序性会对物流配送效率产生巨大约束。通过先进的无人机技术，无人机搭载了配送模块，具备了配送物流的基本属性。空中交通的自由度大大优于地面交通，尤其在低空非禁飞区基本不存在空中交通管制，因此空中物流配送具备地面交通难以匹及的灵活性。空中物流提高了物流配送效率，提高了用户使用体验。

在应急救援任务中，无人机提高了任务执行的安全性，降低了由于人参与危险任务产生的事故风险。2008 年汶川地震后，强烈的地震造成了汶川地区大规模通信损毁，曾一度与外界失联二十个小时。在这种自然灾害的场景，通过跳伞、和地面部队参与救援存在很大的隐患，对士兵的生命存在威胁。利用无人机的高机动性和无人智能性，避免了人直接参与到事故现场进行应急救援，降低了危险任务的致命性。

在通信组网任务中，无人机的灵活性能够解决局部拥堵导致的通信堵塞，便利人们的实时通信。由于大规模集会，如：演唱会、演出或大规模会议，会产生局部地区宽带资源枯竭，难以满足大容量、高速率的实时通信需求。加载信号处理和通信平台的无人机，通过集群组网，可以形成局部通信基站，对通信资源受限地区进行通信增强，弥补由于短时拥堵时频带资源受限的缺点。

综上所述，无人机参与社会活动给人们的生活带来了便利性，增加了社会生产力的流通性，推进了社会生产的进步。

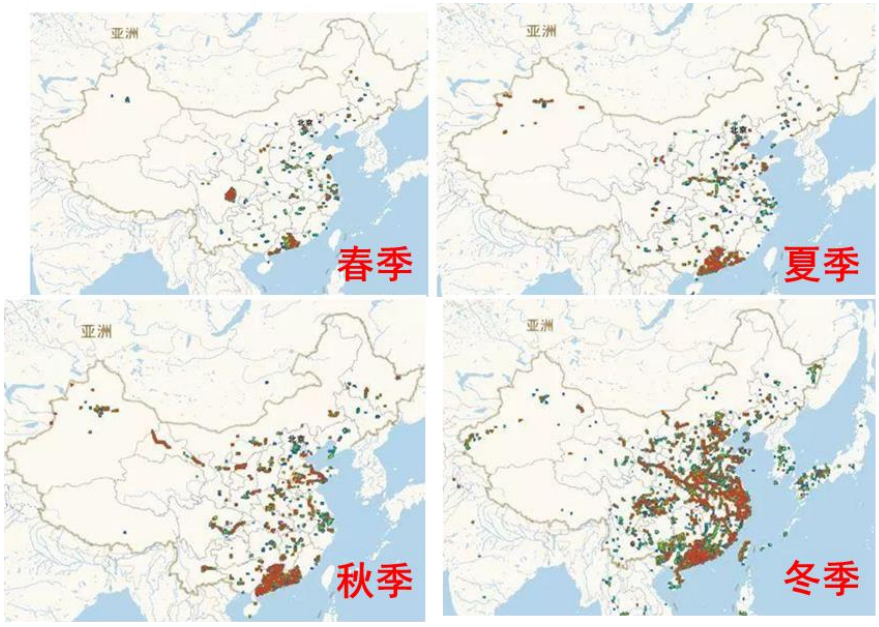


图 1.2 2017 年各季度全国无人机运行热力图

由全国无人机运行热力图可看出，无人机越来越频繁地参与到人们正常的社会生活和社会生产中，大大推动了社会生产力的发展。

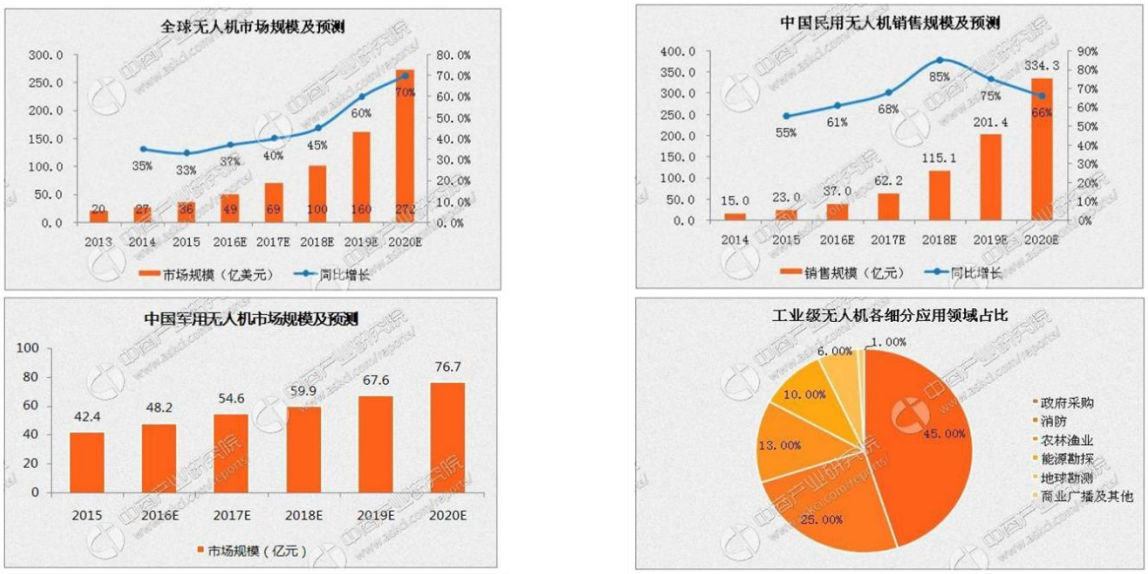


图 1.3 国内外无人机市场-年份图

由国内外无人机市场-年份图也可看出，市场对无人机的需求愈发旺盛，无人机参与社会生产各个方面的比例逐年提升。

由于无人机的计算核心是任务所驱动的，无人机在环境中执行的算法与所完成的目

标相关，因此，对不同的任务要重新进行场景构建、算法设计。由于计算力的受限和对实时性的要求，在非结构化和高度动态的环境下，无人机在以可行和安全的方式进行自主导航方面也面临着许多挑战。除此之外，多无人机协同任务中存在的“NP 问题”限制了无人机平台的扩展性，难以满足多目标规划和多任务协同问题中，人们对高效性的需求。以上原因导致到目前为止，无人机在城市环境中的部署,以及在工程中的应用十分有限。

在问题一中，场景构建是为了提取环境特征，进行地图建模，以便在计算设备上进行数字计算。而算法设计则是在考虑场景特征、环境约束以及无人机自身动力学规则的基础上，从任务目标出发，设计一套无人机与环境安全交互的控制指令，给出在约束条件下一条安全可执行的无人机路径。

在问题二中，在面临环境场景高动态、空域信息不完备、约束条件多元化的条件下，如何进一步提升无人机自主系统的安全性和可靠性，如何在问题一全局路径规划结果的基础上进一步优化目标解，优化交互控制指令，提高路径规划的效率。

在问题三中，多无人机任务协同问题的核心，是在单无人机执行任务的基础上，由任务目标驱动多个无人机在时域上进行分配路径规划结果，分时路径构成多无人机之间的关联关系，并考虑如何优化关联的鲁棒性和效率。

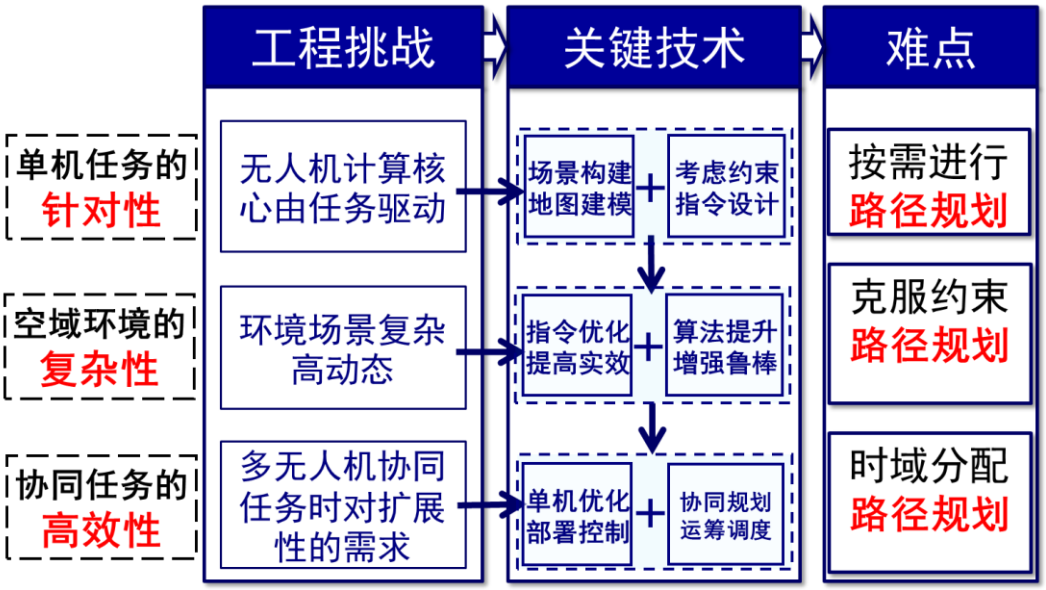


图 1.4 无人机路径规划关键技术

由三个问题难点所引申出的路径规划问题，成为了无人机技术领域最基础和困难的问题之一。为解决无人机工程应用中的关键难点而引申出的无人机路径规划问题成为科学研究的热点，如何进行更加高效、鲁棒的路径规划是机器人领域一个具有挑战性的开放性问题。

1.2 国内外研究状况

国内外学者围绕无人机路径规划、智能体路径规划展开了各层次、各方面的深度研究。目前较为成熟的方法可以分为以下几类：确定性方法、随机化方法、智能优化方法，确定性方法和随机化方法统称为传统路径规划算法。不同算法所适用的问题场景不同，在不同环境中的表现效果也存在较大差异。以下将会关注较为成熟的路径规划算法，对比分析算法的优缺点，以及算法的使用场景，进而说明我们工作的必要性。

1.2.1 确定性路径规划算法

传统的路径规划算法一般指基于图理论的一类搜索算法，以是否需要在算法执行前对环境建图分为确定性方法和随机化方法。

确定性方法中，以 A*算法[1]最具代表性。A*算法在经典 Dijkstra 算法的基础上，加入目标引导的引导函数，加速算法的收敛速度，理论证明，该算法的收敛速度和最优性趋于广度优先与深度优先算法之间。

A*算法需要对所有图节点进行重复迭代计算获得计算结果，因此其实时性仍然难以满足工程需要。由此需求引出了一类 A*算法的变体，其中最具代表性的是 D*算法[2]。D*算法在建图之后，只在向目标点移动的最短路径中对上下节点或临近节点进行检查，因此，D*算法的执行速度较高，但损失了一定的精度，做了速度与准确率之间的权衡。

A*算法及其变种，由于其在已知图模型的前提下，执行效率高、最优解质量好等特点，在路径规划问题中广泛应用。但当环境未知或复杂、场景高动态导致难以建立稳定的图的情况下，算法失效，限制了这类方法的进一步应用。

1.2.2 随机化方法

为减少对结构化环境的依赖性，进一步扩展路径规划问题所适用的场景，另一大类

路径规划方法：随机化方法，应运而生。其中最具代表的是 PRM 算法与 RRT 算法。

PRM 算法[3]（Probabilistic Roadmap）首先对场景模型随机采样，通过碰撞检测确定可执行域，进而通过在可执行域中调用最短路径算法获得最优解。有理论证明[4]，该类方法在采样点数趋于无穷时，解也趋向最优。但 PRM 算法存在多个两点边值问题[5]（BVP），BVP 问题使得在两点间构造满足约束条件的路径较为复杂，因此，一般只用于解决静态环境问题。

RRT 算法[6]（Rapidly Exploring Random Tree）利用扩展树探索空间，通过碰撞检测标记可行航迹空间，无需对环境建模，不存在 BVP 问题，不存在局部最优，算法复杂度增长平缓。同时由于其树增量式生长的特点，可以在生长过程中，将航迹规划问题中的多约束条件加入树的增长模式中，所以容易与其他方法结合，容易加入启发信息，可以用来在未知、动态环境中进行实时路径规划。因此 RRT 算法作为基算法产生了一大类优化算法，最具代表性的，有以下四类：

（1） RRT 算法与模型预测[7]（MPC）：

MPC 旨在通过搜索路径规划域内的环境信息，并预测环境信息的变化趋势，通过无人机运动模型搜索生成局部航迹，再利用环境信息修正，逐步生成全局航迹；MPC 作为在线规划框架，RRT 作为搜索算法，这种结合适用于处理动态环境下的实时路径规划；存在局部最优的可能；选取子目标未利用全局信息，容易陷入局部最优，不适合在障碍物密集环境中规划。

（2） Dynamic Domain RRT[8]：

对被障碍物阻挡而无法生长的树节点添加动态采样域，域中保留了树的空间拓展信息和障碍物的分布信息，可作为启发，引导 RRT 生长，降低对障碍物周围区域的扩展率，以减少碰撞次数；同时修正树的结构，更快在复杂细小的通道内进行生长。

（3） Anytime RRT[9]:

Anytime 框架下，先搜索初始路径，再优化路径，实时性高，先排除高代价树分支，保留新发现的比当前代价更低的分支，渐近优化 RRT；由于 RRT 的随机性，导致存在收敛性较慢的问题；由于修剪树枝，随时了一部分探测信息，存在工作冗余。

（4） RRT*[10]:

保留了树的扩展信息，利于寻找最优路径；也具有 Anytime 框架和渐进最优特性。激发代价函数计算路径两节点之间的低代价区域，在该区域内采用高采样率，提高 RRT* 优化效率。

随机化方法由于其灵活的扩展性和良好的兼容性，在很长一段时间内，一直作为机器人路径规划的主流算法。虽然在一定程度上，随机化方法减少了确定性方法中对环境的依赖，并且提高了算法的实时性，但是两者都局限于对环境进行建图，当环境约束较多、信息不完备时，两种算法都面临极大风险的发散可能。除此，两者都需要在路径规划解之后对路径解进行优化，以平滑路径，满足实际无人机动力学约束，将路径规划问题拆解为两个子问题降低了可集成性。

1.2.3 智能优化方法

为了进一步考虑多约束条件，扩展算法使用范围，一类基于仿生学的智能计算算法成为研究热点，如：遗传算法[11]、模拟退火法、鸽群算法、蚁群算法[12]等。这类算法启发于生物界，核心思想大同小异。这类算法的通过对多个约束条件进行数学方程描述，连同环境约束共同建立约束方程组，将任务目标建立为损失函数，以损失函数为优化目标，同时考虑约束方程的限制，进行组合优化。理论上，任何约束条件都可以加入优化过程中，因此，这类方法可以同时满足环境约束和无人机动力学限制。但是，算法的发散风险也随着约束条件的增加而增长。除此之外，这类方法的建模难度较高，对不同场景和任务，需要对约束条件和损失函数进行精心设计，不具有场景迁移性，在实际工程中代价较高昂。因此，这类算法在实际工程中难以扩展运用。

综上所述，无人机路径规划问题是无人机自主控制领域最基础与核心的问题，而现在缺少一种兼顾迁移性、效率与可靠性的无人机路径规划算法，能够实际运用于工程，尤其是在面对未知环境、多约束条件、以及场景复杂高动态时，仍然能够保证一定的精度和速度。

几类算法的适用性如下图所示：

1.3 课题研究方法

自谷歌 AlphaGo 问世以来，强化学习广受关注。不同于监督学习，强化学习可以利用“经验”进行特征提取和思维总结，重点聚焦智能体的决策行为，从而使它可以真正让计算设备像人一样进行思考和决定。强化学习算法引导下的模型最吸引人的便在于它的“模型兼容性”，即对不同场景都具备着迁移性，这让我们真正可能实现“一种方法解决所有问题”的美好愿望。国内外学者利用强化学习和深度学习实现了许多实际中机器人的路径规划，实际问题中的场景是复杂多变的，因此在利用深度强化学习进行智能体学习时，对神经网络的泛化能力和收敛性提出了很高的要求。

诸多学者专家围绕如何提高强化学习问题中神经网络的收敛性和鲁棒性这一话题进行了广泛和深入的研究。

实际问题中，无人机所经历的状态为连续值，为了数字计算又能保证一定的精准度，需要对连续状态离散化。但对高精度要求的任务，高粒度的离散化会产生维度灾难问题，导致无法通过存储值来进行迭代计算，因此，需要一种方法来缓解实际应用中出现的维度灾难问题。

于乃功[13]等人在 2016 年发表《基于深度自动编码器于 Q 学习的移动机器人路径规划方法》，本文用深度自动编码器代替 Q-Table，解决的多维度时的维度灾难问题。相似地，梁泉[14]等人在 2012 年发表的《位置环境中基于强化学习的移动机器人路径规划》一文中，通过模糊逻辑方法解决了连续状态空间的泛化问题。同样解决了多状态的维度灾难问题；张凤运[15]等人在《基于 RBF 网络和 Q 学习的路径搜索与移动导盲系统设计》中，利用 RBF 网络拟合 Q-Table，进而解决维度灾难问题。Duguleana[16]等人于 2016 年发表的“Neural networks based reinforcement learning for mobile robots obstacle avoidance”一文中，利用依赖神经网络的 Q-learning 算法，解决了机器人在包含静态障碍和动态障碍的环境中自主运动的问题。

可见，利用深度学习网络的强大非线性映射能力，我们可以根据历史的状态-动作集合训练一个具有前向计算功能的神经网络，存储强化学习计算中海量的交互数据，用于每一步的计算。

实际问题中，需要对无人机的状态进行描述。基于无人机的硬件设备：激光雷达、

声呐雷达，流行的状态描述方式是基于景深图像或云点图的图描述，另一类是基于多维向量的状态表征。

LeiTai[17]等人2016年发表的“Towards cognitive exploration through deep reinforcement learning for mobile robots”一文中，利用机器人所拍摄到的RAW图像作为算法的输入，当图片景深小于阈值时（处于安全区），执行DQN的基本算法。由于现实的机器人探索代价高昂，所以文章的网络训练在仿真环境中完成。

描述状态特征的多维向量维度间常存在关联，数据在高维空间往往稀疏；不同维度的放缩尺度差异较大，离散和连续属性也有不同，难以用同种准则进行归一化；多维向量中蕴藏的时序与空间信息，往往需要通过特殊的神经网络结构进行处理（如：LSTM），所以需要对神经网络的架构进行精心设计。而在深度学习中用卷积神经网络处理图像数据的技术框架十分成熟，效果卓著，应用广泛；利用图像处理中的边缘检测、图像分割与识别、景深图像信息提取等技术，能够完全表征无人机的当前状态。

实际问题中，环境的复杂、信息不完备与高动态等都有可能造成神经网络的收敛速度变慢，环境的多元多粒度又会带来对神经网络泛化能力的要求。因此，如何提高强化学习问题中神经网络的收敛速度与鲁棒性也是至关重要的。

宋勇[18]等人于2012年，在《移动机器人路径规划强化学习的初始化》一文中，通过利用人工势场法，对初始环境进行了初级建模。将势场每一点的势能值作为Q-Table的初始值，以此代替Q-learning算法中对Q-Table的随机化赋值，提高了Q-learning算法的收敛速度。

赵英男[19]在2017年《基于强化学习的路径规划问题研究》一文中，通过智能体到达目标的成功率作为指标衡量智能体对环境的掌握程度，从而动态调整Q-learning算法中的探索率。王维在2004年《一种基于强化学习的自适应变步长路径规划算法》中引进评价预测学习的自适应步长学习算法，实现了步长在线调节，加快了路径规划的计算速度。这种方法使得Q-learning算法的收敛速度平均提高了十倍以上。

可见，较为流行的方法仍然是注重于神经网络的初始化和神经网络超参数调整方面。合理进行神经网络初始化虽然有助于提高神经网络的收敛速度，但是这种方法需要先验判断经验，而在实际工程问题中，这个要求往往难以达到。而对神经网络超参数的调整

是一个经验性问题，难以有一套成熟、泛化的方法对神经网络进行优化。提高神经网络的性能仍然应该从源头出发。

高等生物在完成某个动作时不会从空白开始学习，而是存在层级式学习以及主从配合与控制。这一高等生物的行为特征引出了分层强化学习的概念。

DeepMind 团队于 2017 年 7 月发表文章”Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”提出了一种多智能体策略优化算法[20]。可以通过多个神经网络对多智能体行为进行调控和诱导，算法作用下，多智能体可以完成动态博弈、追击、围劫等智能策略行为。是第一次利用强化学习算法，指导复杂系统的行为。

Stone P[21]等人提出一种分层学习理论，对目标任务进行子任务拆解，在学习和完成子任务的基础上，结合各个子任务的模型，获得全局模型。Vezhnevets A[22]等提出可利用于现实复杂环境导航的分层强化学习模型，利用专家先验知识对强化学习问题建立动作子空间，从而对每个状态所对应的动作缩减为几个有效值，大大提高了算法的收敛性和效率。

分层强化学习在许多竞技电子游戏中（Game Soccer、Dota 等）获得了巨大成功。可以预见，如何参考分层强化学习的启发思想，将无人机路径规划问题进行分层建模，从而提高算法的性能，将会成为无人机路径规划问题的新的研究热点。

1.4 论文构成及研究内容

本工作将会由下而上，自底而上，对工作进行叙述：第一章：绪论，主要介绍本工作的问题背景、问题难点以及对于本问题的国内外研究现状，对问题框架进行梳理；第二章：强化学习的数学原理与建模，主要介绍强化学习的基本概念和数学公式，以及对无人机路径规划问题进行简单建模，以便进行强化学习算法的运行；第三章：基于深度强化学习的路径规划，主要介绍通过引入深度神经网络参与到强化学习模型的计算中来，介绍几种鲁棒和优化的深度强化学习算法，解决实际无人机路径规划问题中的维度灾难问题；第四章：基于分层强化学习的路径规划，主要介绍分布式人工智能的概念，通过对任务分层和网络匹配，实现分布式训练和并行式计算，提高算法在面临高动态、多约束的复杂场景时的鲁棒性和收敛性。

2 强化学习的数学原理

强化学习作为机器学习的方法之一，又称再励学习、增强学习，来源于生物学中的条件反射理论，其基本思想是对所希望的结果予以奖励，对不希望的结果予以惩罚，逐渐形成一种趋向于好结果的条件反射。如图 1 所示，智能体在完成某项任务时，首先通过动作 A 与周围环境进行交互，在动作 A 和环境的作用下，智能体会产生新的状态 S，同时环境会给出一个立即回报 R。如此循环下去，智能体与环境不断地交互从而产生很多数据。强化学习算法利用产生的数据修改自身的动作策略，再与环境交互，产生新的数据，进一步改善自身行为，经过数次迭代学习后，智能体能够学到完成相应任务的最优动作，也就是最优策略。

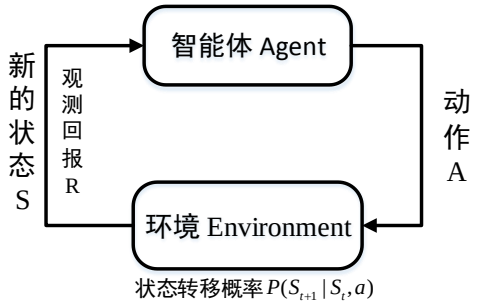


图 2.1 强化学习原理示意图

强化学习的数学基础是马尔可夫决策过程，它的特点是短记忆性、概率转移性和空间封闭性。它的主要思想是将智能体与环境的交互过程进行马尔可夫链的建模化，数学理论阐述的过程中，涉及几个重要的概念和定义，下面将用列表和示意图的方式，结合实例来进行阐述。

表 2.1 强化学习中的数学符号

概念与定义	数学符号
状态（无人机当前的位置、速度等）	$\boldsymbol{S_t}$
动作（当前状态采取的动作，直飞、左转等）	$\boldsymbol{a}$
奖励（无人机采取某动作后得到的环境反馈）	$\boldsymbol{r}$
策略（在确定状态采取确定动作的概率分布）	$\boldsymbol{\pi}$

状态转移概率（在某状态转移到下一特定状态的概率）	$P_{s'\|s}$
收获（在一个状态序列里的累积奖励）	G
价值（当前状态下的所有可能的收获的期望）	$V_{\pi}(S_t)$
动作价值（在某一状态采取某一动作的价值）	$Q(S_t，a)$

2.1 马尔可夫决策过程

马尔可夫过程是经典概率论中一类具有典型特征和特殊性质的随机过程，其核心概念是未来所发生事件的概率仅取决于当前时刻，而与过去时刻所发生的事件无关，或仅与过去有限时刻所发生的事件有关。事实上，绝大物理世界之中，随机事件都具备马尔可夫特性，又得益于马尔可夫性质，所以马尔可夫过程被广泛应用于工程问题中作为数学分析和理论建模的工具。

对于强化学习问题而言，仅依靠马尔可夫过程无法描述，因为马尔可夫过程仅涉及状态之间的转移，不涉及动作的选取。因此有必要引入马尔可夫决策过程，来对强化学习进行数学建模。马尔可夫决策过程在马尔可夫过程的状态空间基础上，引入了动作空间，完全描述了强化学习问题中智能体的行为与环境交互过程。

一步转移马尔可夫决策过程示意图如下所示：

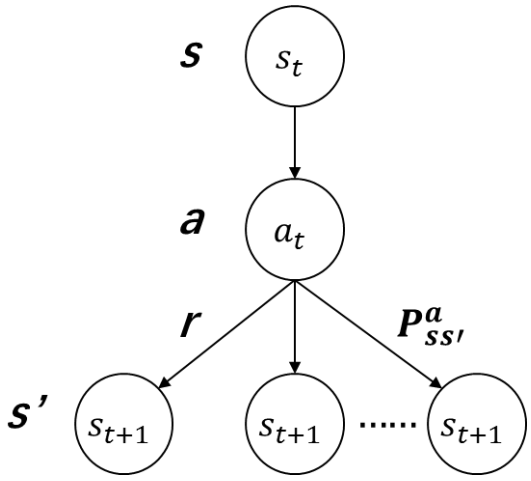


图 2.2 马尔可夫决策过程

s、s'分别代表当前时刻的状态与下一时刻所转移到的状态；a 为在当前状态 s 所采取的动作；r 代表在当前状态 s 采取动作 a 后在环境中所得到的反馈；$P_{ss'}^{a}$代表了在当前状态 s 采取动作 a 后转移到下一状态 s'的概率值。一个马尔可夫决策过程用<S,A,P,R,γ>表示。

在马尔可夫决策过程用<S,A,P,R,γ>中，有以下几个数学概念较为关键：

策略：$\pi(a|s) = P[A_t = a|S_t = s]$，表示的是基于一个确定的状态至可采取的动作空见的一个概率分布，是强化学习问题中的优化目标。

价值函数：$V_\pi(s) = E[G_t|S_t = s]$，其中$G_t$为累计收获，$G_t = R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\cdots$, R_t是 t 时刻所获得的奖励值，γ是介于 0 至 1 之间的常数，我们称为衰减因子，用于减小未来时刻状态对当前时刻状态的影响，$\gamma = 0$时，未来时刻与当前时刻独立，$\gamma = 1$时，未来时刻与当前时刻影响系数均等。累计收获反应的是在未来所可能获得的收获总和，而价值函数是对累计收获求均值，代表期望的未来累计收获。

行为价值函数：$Q_\pi(s,a) = E[G_t|S_t = s, A_t = a]$，反应的物理含义是，在当前状态采取动作 a 后未来累计收获的期望值。

$V_\pi(s)$和$Q_\pi(s)$是强化学习问题中最重要的概念，在强化学习问题中，我们期望下一时刻所产生的动作能够在未来获得最大的期望累计收获。根据上述定义我们可知：$V_\pi(s)$反映了当前状态 s 的价值，因此又称为状态价值函数；$Q_\pi(s)$反映了在当前状态 s 且采取动作 a 的明智与否。两者的关系如下图所示：

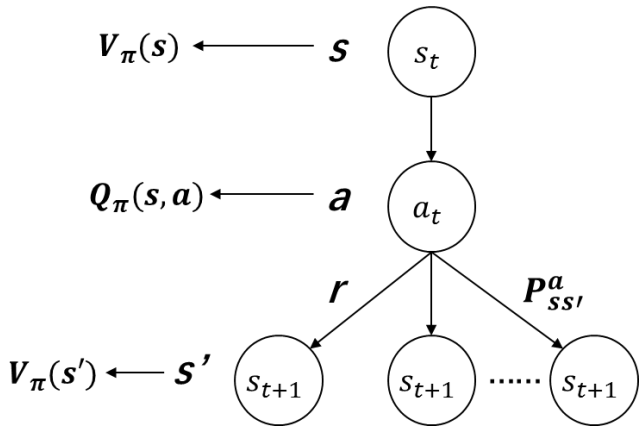


图 2.3 价值函数与行为价值函数示意图

根据关系图，分别以 S、S'为节点列写$V_\pi(s)$和$Q_\pi(s)$的节点方程可得：

$$V_\pi(s) = \sum_a \pi(a|s) Q_\pi(s,a) \tag{2.1}$$

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s') \tag{2.2}$$

$$V_\pi(s) = \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s')) \tag{2.3}$$

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \sum_a \pi(a|s) Q_\pi(s,a) \tag{2.4}$$

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s')) \tag{2.5}$$

由公式$V_\pi(s) = \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s'))$可看出，当前时刻状态价值函数可以通过下一时刻状态价值函数求解出，在经典强化学习理论中，该公式成为：贝尔曼方程，它启发：一个状态的价值可以通过该状态的奖励以及后续状态价值按照概率分布求和按照一定比例衰减联合组成。

2.2 最优策略

通过上节，我们通过建立马尔可夫决策过程对强化学习问题进行了数学建模。本节将会阐述优化目标和优化过程。

2.2.1 优化目标

上文已说明，解决强化学习问题就意味着要寻找一个最优策略，让个体在与环境交互过程中获得始终比其他策略更多的收获，最优策略我们用π^*表示。

定义最优状态价值函数$V^* = max_\pi V_\pi(s)$，最优行为价值函数$Q^* = max_\pi Q_\pi(s,a)$，由贝尔曼方程$Q_\pi(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s')$容易分析，$R_s^a$与$P_{ss'}^a$均是环境动力学所定义的客观事实，与状态函数独立，因此，当$V_\pi(s)$达到V^*时，$Q_\pi(s,a)$也达到Q^*。

策略π优于策略π'当且仅当$V_\pi(s) \geq V_{\pi'}(s)$成立。因此，最优策略下，状态价值函数等同于最优价值函数。又由于贝尔曼方程的约束，最优策略下，行为状态价值函数与最优行为价值函数也等价。

由定义可知，策略π是基于当前状态而采取某个动作的概率分布，因此最优策略的获得可以通过最大化行为价值函数来趋近：

$$\pi^* = \begin{cases} 1 & if\ a = argmax\ Q^* \\ 0 & else \end{cases} \tag{2.6}$$

因此，强化学习的优化目标可由最优贝尔曼方程表征：

$$\pi^* \to Q^* \to V^*$$

$$Q^* = R_s^a + \gamma \sum_{s'} P_{ss'}^a \sum_a \pi(a|s) Q^*$$

$$V^* = \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V^*) \tag{2.7}$$

最优贝尔曼方程不是线性方程，无法直接求解，通常采用迭代法求解。

2.2.2 优化过程

根据上文所述，策略π优于策略π'当且仅当$V_\pi(s) \geq V_{\pi'}(s)$成立。策略评估是指，在给定策略的前提下，计算所有状态的价值函数的过程。策略评估是评价给定策略优劣的手段。策略评估的具体方法可以利用同步迭代联合动态规划进行求解：从任意一个状态价值函数开始，依据所给定的策略，结合贝尔曼期望方程、状态转移概率和奖励，同步迭代更新状态价值函数，直至其收敛，更新方法由下式给出：

$$V_{k+1}(s) = \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_k(s')) \tag{2.8}$$

给定策略往往是均一概率的随机策略，在该策略下，每一个状态的价值是不一样的。对于智能体，通过不断的与环境交互，经历多次终止状态后会对每一个状态有一定的认识，智能体形成这个认识的过程就是策略评估的过程。

完成一个策略的评估过程，将会得到基于该给定策略下，每一个状态的价值，不同状态的价值反映了智能体对不同状态的认识。基于该认识，智能体可以通过一定的方式，基于状态价值来调整自己的行动策略。通常情况下，我们考虑一种贪婪策略π'来指导智能体基于状态价值调整行动策略的过程：智能体在某个状态下所选择的动作能够到达后续状态价值最大的状态，即：

$$\pi' = greedy(V_\pi) = argmax_a Q_\pi(s,a) \tag{2.9}$$

因此，基于当前状态采取贪婪策略指导智能体更新行为后，有: $Q_\pi(s, \pi'(s)) \geq Q_\pi(s,(s))$成立。

优化目标的优化方法如下：

Step 1：给定随机策略π，随机初始化所有状态价值函数$V_{k+1}(s)$；

Step 2：策略评估：迭代计算$V_{k+1}(s) = \sum_a \pi(a|s)(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_k(s'))$直到$V_{k+1}(s)$

与$V_k(s)$差异小于给定阈值；

Step 3：利用V_π计算$Q_\pi(s,a)$，并通过贪婪策略，每次选取最大动作价值函数调整智能体动作$\pi' = argmax_a Q_\pi(s,a)$；

Step 4：策略迭代：利用π'产生动作$a = \pi'(s)$作为新策略再次进行策略评估，再次返回到 Step 3；

Step 5：重复以上过程直到达到迭代次数；

2.3 不基于模型的控制

在环境转移概率完全已知的条件下，通过马尔可夫决策过程和策略优化方法，我们分别对强化学习问题进行了建模和求解。但是实际环境中环境转移概率无法定义，也无法完全获得，因此，仅仅依赖贝尔曼方程和动态规划技术无法将强化学习算法迁移至现实无人机环境中，需要在经典强化学习理论上继续扩展适用性。

2.3.1 蒙特卡洛算法和时序差分算法

策略评估由前文所述是在给定策略的情况下求解状态价值函数。考虑累进更新平均值方法：

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}[x_k + (k-1)\mu_{k-1}] \\ &= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})\end{aligned} \tag{2.10}$$

由于价值$V_\pi(S_t)$是收获 G(S_t)的均值，所以有：$\mu_k = V_\pi(S_t)$,$x_k = G(S_t)$；$k = N(S_t)$代表了计算次数；将上述公式带入累进更新平均值公式得：

$$\begin{cases} N(S_t) \leftarrow N(S_t) + 1 \\ V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G(S_t) - V(S_t)) \end{cases} \tag{2.11}$$

在一些实时任务或者无法准确统计状态访问次数的任务中，可用常数α代替$\frac{1}{N(S_t)}$，强化学习理论中，称α为学习率，由此得出蒙特卡洛算法：

Step 1：给定一预定策略π，对状态空间初始化$V(S_t)$；

Step 2：依据$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G(S_t) - V(S_t))$对状态空间中所有价值函数进行迭代更新；

Step 3：重复 Step 2 直到到达预设定更新次数；

蒙特卡洛算法需要在一个完整的状态序列的前提下，才能够进行迭代计算进行策略评估。而现实情况中，收集很多完整的状态序列不现时，尤其是在无人机路径规划任务中，获得一条完整的状态序列会消耗漫长的时间，因此在现实工程中很少使用蒙特卡洛算法进行策略评估。

为解决蒙特卡洛算法的局限性，Tesauro[23]等人提出时序差分算法，来解决策略评估必须采用完整状态序列的局限。

时序差分算法主要是从采样得到的不完整状态序列中学习，通过合理的引导，估计某状态在该状态序列序列完整后可能得到的收获，在此基础上，利用累进更新平均值法得到该状态的价值，再通过不断的采样来持续更新这个价值。

Tesauro 等人提出利用即时奖励R_{t+1}与下一状态的预估价值$V(S_{t+1})$的衰减值来预估当前状态的收获，即：$G(S_t) = R_{t+1} + \gamma V(S_{t+1})$，带入$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G(S_t) - V(S_t))$中，得：$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$，引导指的就是用$R_{t+1} + \gamma V(S_{t+1})$来估计$G(S_t)$的过程。

由此得出时序差分算法：

Step 1：给定一预定策略π，对状态空间初始化$V(S_t)$；

Step 2：依据$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$对状态空间中所有价值函数进行迭代更新；

Step 3：重复 Step 2 直到到达预设定更新次数；

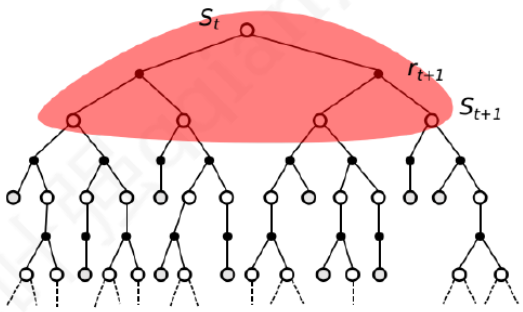
对比两种算法，蒙特卡洛算法只能在状态转移完整后才能更新每个状态的价值，而时序差分算法每经过一个状态，便可以与前一个状态一起来更新前一个状态。因此时序差分算法能够更加快速灵活地更新状态价值，进行策略估计。

蒙特卡洛算法严格利用大数定律来估计收获，是严格的无偏估计；而时序差分算法利用引导来估计，因此是有偏估计。但是由于引导是两个邻近状态价值函数之间联合决

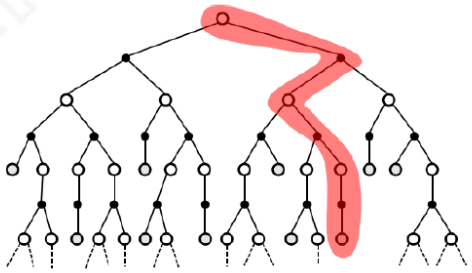
定的，因此时序差分算法方差小于蒙特卡洛算法，且初始值比较敏感，能够更加高效的学习。

除此，时序差分算法对收获的估计依赖于马尔可夫特性，即未来时刻的状态仅与当前时刻状态或之前有限个状态有关，因此在具有马尔可夫特性的模型中更加有效；蒙特卡洛学习没有这种限制，理论上可以适用于所有的环境模型。

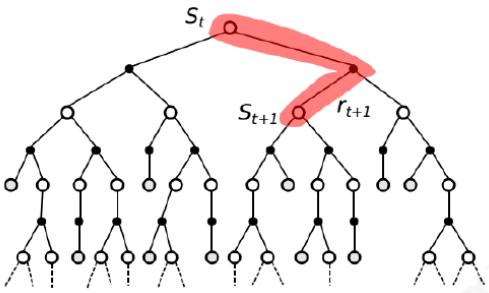
两种算法和之前所提贝尔曼优化方法对状态序列结构的依赖如下所示：



贝尔曼优化方法



蒙特卡洛算法



时序差分算法

图 2.4 三类方法对状态序列的要求

2.3.2 Sarsa 算法和 Q 学习算法

策略评估是给定策略的前提下计算状态价值的过程。策略评估之后，利用状态价值反过来可以进行动作的控制，进而优化策略。

根据产生实际动作的策略（行为策略）与更新价值所使用的策略（被评估的策略）是否是同一个策略，分为现时策略学习和借鉴策略学习。在实际中，基于时序差分算法由于其自身的优点，演化出了最为常用的两类算法：Sarsa 算法和 Q 学习算法。

由上文论述知，优化状态价值函数等价于优化行为价值函数。为了便于在策略评估之后进行策略优化，本节将会用行为状态价值函数进行数学分析。

Sarsa 算法的名称来源于下图所示的状态转移序列：

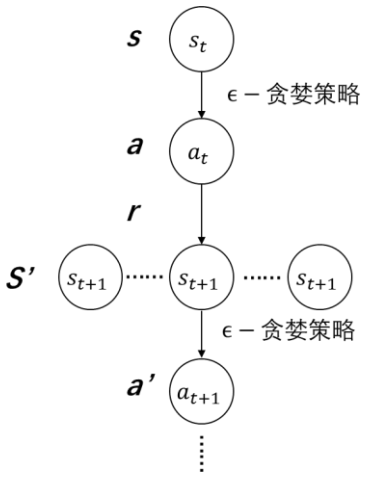


图 2.5 Sarsa 算法示意图

针对一个状态 S，个体通过行为策略产生一个动作 A，该动作会在环境中得到即时奖励 R，被环境动力学转移到 S'；智能体在状态 S'遵循当前的行为策略产生新的动作 A'，但智能体并不执行 A'，而是通过行为价值函数得到后一个状态、动作对<S',A'>的价值，利用这个价值和即时奖励 R 来更新前一个状态行为对<S,A>的价值。

在上文，行为策略一直采用贪婪策略描述，但贪婪策略存在一些局限。由于贪婪策略优化动作选择时始终考虑最大的$Q(S,A)$，可能在多次迭代后，智能体仅探索了很小一部分的状态空间，存在许多未知的状态价值，因此这种行为策略很有可能并不能得到全局的最优策略。这里采用一种优化版的贪婪策略：$\epsilon-$贪婪策略。$\epsilon-$贪婪策略在贪婪策略基础上演化而来，对其叙述如下：

$$\pi(a|s)=\begin{cases} argmax\ Q(S,A) & if\ random\ P\in[0,1]>\frac{\epsilon}{N} \\ random\ a & else \end{cases} \tag{2.12}$$

一般 N 为探索次数。行为策略描述为有一定概率选取使行为价值最大的动作；而也有一定的概率随机选取动作，以尽可能探索状态空间。随着探索次数增多，$\frac{\epsilon}{N}$趋于 0，行为策略最终趋向于贪婪策略，这是一种探索-利用上的权衡。

更新的过程借鉴时序差分算法。Sarsa 算法的叙述如下：

Step 1：初始化行为价值函数$Q(S,A)$，初始化迭代次数；

Step 2：对于状态 S 和$Q(S,A)$，依据$\epsilon-$贪婪策略产生动作（行为策略），得到即时奖励 R，转移到下一状态 S’；

Step 3：依据时序差分算法进行策略评估：$Q(S_t,a)\leftarrow Q(S_t,a)+\alpha(R_{t+1}+Q(S_{t+1},a')-Q(S_t,a))$;

Step 4：在得到$Q(S_t,a)$后，进行策略优化，若迭代次数不满足，回至 Step 2，依据$\epsilon-$贪婪策略产生动作；

Step 5：重复以上执行直到迭代次数满足；

由于 Sarsa 算法行为策略和策略评估时采用的策略是同一个策略，因此 Sara 又称为现实策略时序差分算法。

Sarsa 算法实际应用中收敛较慢，为了解决收敛性慢的问题，Watkins[24]等人提出 Q 学习算法，Q 学习算法作为最经典的强化学习算法，被广泛应用于工程中，用于机器人控制等任务。

Q 学习算法在 Sarsa 的基础上改变了策略评估过程，如下所示：

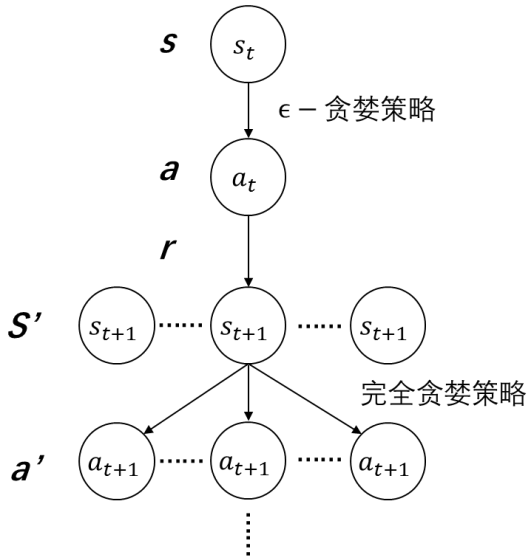


图 2.6 Q 学习算法示意图

针对一个状态 S，个体通过行为策略产生一个动作 A，该动作会在环境中得到即时

奖励 R，被环境动力学转移到 S'；智能体在状态 S'遵循完全贪婪策略产生新的动作 A'，A'能够满足在所有可能的 a'中，智能体能够获得最大的$Q(S,A')$，利用这个价值和即时奖励 R 来更新前一个状态行为对<S,A>的价值。

Q 学习算法描述如下：

Step 1：初始化行为价值函数$Q(S,A)$，初始化迭代次数；

Step 2：对于状态 S 和$Q(S,A)$，依据$\epsilon-$贪婪策略产生动作（行为策略），得到即时奖励 R，转移到下一状态 S'；

Step 3：依据时序差分算法进行策略评估：$Q(S_t,a)\leftarrow Q(S_t,a)+\alpha(R_{t+1}+\gamma maxQ(S_{t+1},a')-Q(S_t,a))$;

Step 4：在得到$Q(S_t,a)$后，进行策略优化，若迭代次数不满足，回至 Step 2，依据$\epsilon-$贪婪策略产生动作；

Step 5：重复以上执行直到迭代次数满足；

Q 学习算法行为策略和策略评估时采用的策略是不同的策略，因此，Q 学习算法也称为借鉴 Q 学习。

相比 Sarsa 算法，Q 学习每次选择最大的 Q 值进行更新，加快了算法的收敛速度，但也带来了一定的智能体风险，智能体有可能多次陷入不利的状态。在实际运用两种算法时，应当考虑到这种权衡。

2.4 实验结果与分析

由于 Q 学习算法收敛性较好，我们根据 Q 学习算法的基本思想，结合无人机实际环境，建立模型进行计算机仿真。由于 Q 学习方法的计算过程通常通过存储 Q 值表格进行，因此状态空间不能太大，否则算法很容易发散。

建立离散方格世界作为无人机的实际环境，障碍物仅考虑静态且形状规则，无人机的描述和障碍物的描述都采用方格来表示，忽略无人机的动力学方程等约束，我们建立起了最简单和基础的无人机路径规划模型。展示效果如下所示：

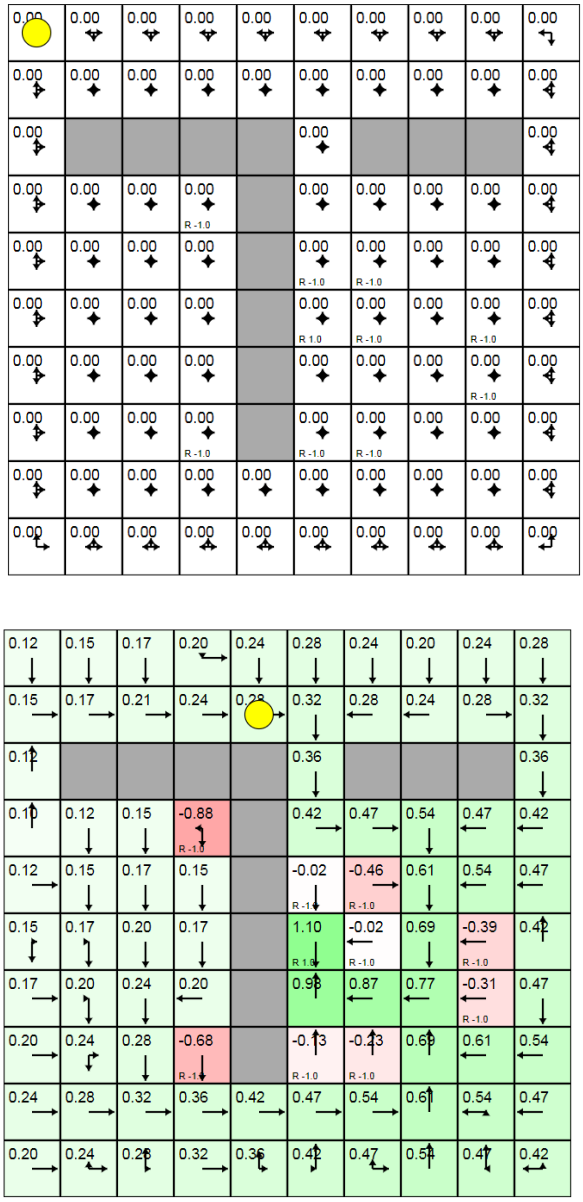


图 2.7 基于 Q 学习算法的无人机路径规划

如图所示，黄色圆圈代表无人机，黑色方块代表静态障碍物，其他颜色均为可执行区域。每一个方格的位置均不同，当我们用位置完全表征无人机状态时，在本实验中，状态空间共$10 \times 10 = 100$个，对应 100 个状态价值函数。执行 Q 学习算法，每一步进行策略评估后，进行策略迭代优化策略，循环往复，直到达到收敛要求时，每个状态的状态价值函数已经不同。如仿真结果图所示：红色方块代表代表会获得较小的未来期望奖励，是不明智的状态，无人机会尽可能避免到达这些状态。绿色方块代表状态价值较

高的状态。根据最优贝尔曼准则，选取 Q 值最大的对应的动作，标注在仿真方格之上，当无人机沿着所指示的动作运行时，能够在该环境中获得最大累积收获。如此，我们便通过算法迭代和状态价值函数实现了无人机的动作控制。

由此可见，当不考虑实际复杂约束、不考虑高动态环境、障碍物较为规则和状态数有限的条件下，Q 学习算法的执行效果较好，并且当 Q 表格计算结束后，再次进行路径规划时，理论上算法的执行时间和在 Q 表格中查找数据的时间复杂度一致，因此 Q 学习算法可以在实时性上有较高的保证。最为重要的，我们通过智能体和环境的交互，利用交互数据建立了智能体对于环境的认识，而基于这种知识，智能体能够实现一定的动作控制，这实现了在信息不完备的条件下进行路径规划的难题。这种特点是传统路径规划算法如：A*、RRT 不具备的。最后，由于环境较为简单，Q 学习算法的收敛性也不会遭受太大挑战。

2.5 小结

无论是 Q 学习算法还是 Sarsa 算法，$Q(S,A)$的计算均是通过一张大表来存储，这不太适合解决规模很大的问题。除此之外，当我们考虑环境高动态时，会极大扩张状态空间范围，导师 Q 学习算法的失效。

实际无人机的环境往往是多约束、高动态且环境未知的。工程中实现无人机自主路径规划必须解决 Q 学习算法存在的维度灾难问题。

3 基于深度强化学习的路径规划

经典强化学习在状态空间有限、环境与场景信息较为规则简单时，能够在智能体行为控制与路径选择方面发挥较优的效果。而实际工程中，无人机所处的低空态势往往属于信息高度不完备、环境动态信息多、实际约束强，且实际无人机的物理状态在量化方面均是连续值，采用连续空间离散化技术后，无人机的状态空间将会非常巨大，若继续采用经典强化学习利用 Q 表格存储 Q 值，进而进行行为控制的方法将会在实效性方面遭遇巨大挑战。

因此，考虑实际无人机工程所处的环境，亟需在经典强化学习的基础上扩展算法，解决实际环境所产生的维度灾难问题。

3.1 基础计算模型

强化学习的数学基础是马尔可夫决策过程，它的特点是短记忆性、概率转移性和空间封闭性。它的主要思想是将智能体与环境的交互过程进行马尔可夫链的建模化。

某状态价值的高低反映了该状态在该环境中是否有意义；动作价值的高低则反映了在某一状态采取某一动作的“明智与否”。利用值函数的高低来选取动作，这延伸出了经典强化学习最重要的算法之一：Q 学习方法。

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}[x_k + (k-1)\mu_{k-1}] \\ &= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})\end{aligned} \tag{3.1}$$

由于价值$V_\pi(S_t)$是收获 G(S_t)的均值，所以有：$\mu_k = V_\pi(S_t), x_k = G(S_t), k = N(S_t)$,

$$\begin{cases} N(S_t) \leftarrow N(S_t) + 1 \\ V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G(S_t) - V(S_t)) \end{cases} \tag{3.2}$$

为了提高算法的灵活性，减小方差，采用了一种对收获$G(S_t)$进行估计的方法进行迭代计算，$G(S_t) \cong R_{t+1} + \gamma V(S_{t+1})$，此时，算法的更新公式变成了：

$$\begin{cases} N(S_t) = N(S_t) + 1 \\ V(S_t) = V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) \end{cases} \tag{3.3}$$

学习率：$\alpha = \frac{1}{N(S_t)} \in (0,1]$,衰减因子$\gamma \in (0,1]$

可见，利用下一时刻收获和下一状态价值函数的衰减之和来估计一条完整的马尔可夫链的收获，算法的更新不再需要当状态转移至一个稳定状态再计算，而是在状态转移的每一步都进行一次迭代计算，大大提高了算法的速度。

为了利用值函数的高低进行动作的控制，引入了动作价值函数 Q($\boldsymbol{S_t}$，$\boldsymbol{a}$)，用动作价值函数代替上述公式中的价值函数。由于动作价值函数反映了某一动作是否“明智”，动作价值函数越高，代表该动作在智能体所处的环境动力学中更加合理，能够在未来的到较大的奖励，所以，在动作价值的更新中，为了进一步加快收敛速度，每次选取下一状态中最大的动作价值，来对当前动作价值进行更新：

$$Q(S_t) \leftarrow Q(S_t) + \alpha(R_{t+1}+\gamma Q(S_{t+1}) - Q(S_t)) \tag{3.4}$$

当$Q(S_t)$完全收敛后，对智能体动作的控制变成了简单的求极值问题：我们采取贪婪策略，即：$a = argmax_{a\in A}\boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a})$，如此，我们便利用值函数的高低间接地进行了智能体行为控制。

3.2 优化子模型

在经典强化学习问题中，每个状态的 Q 值被存储，并形成一个巨大的表格。当迭代计算结束后，就可以进行智能体的控制。但是实际问题中，状态数量往往巨大，存储难以达到要求，并且这种查表式的计算会使得实时性非常低下。利用深度神经网络的非线性性质，理论上可以实现任何的映射问题，我们使用深度学习网络对 Q 表格进行拟合[25]，称为 Deep Q Network 当神经网络收敛后，就相当于得到了一个映射函数，每当我们获得一个智能体的状态，对该状态前项通过神经网络得到一系列 softmax 形式的输出，选取最大的一类值函数所对应的动作，便是符合环境动力学要求的“智能”动作。

3.2.1 环境交互

深度强化学习模型中，可行状态非常多，如果更好探索尽可能多的状态变得至关重要。DQN 采用$\epsilon-$贪婪策略，开始 100%随机产生动作，随着训练不断进行，这个概率被不断衰减直至 10%，因此智能体又 90%的概率执行当前最优策略，仍然有 10%的概率

继续探索状态空间。这样，智能体策略从以探索为主逐渐过渡到了以利用为主。

3.2.2 模型结构

状态价值模型是$|S| \times |A|$到 R 的映射，当模型需要通过价值函数求解最优策略时，我们需要遍历$|A|$空间，在实际工程中采用这种方法效率十分低下。为了简化计算，提高计算效率，我们将$|S| \times |A|$到 R 的映射转变为$S \to [R_i]_{i=1}^{|A|}$的映射，模型的输出为长度为$|A|$的向量，采用这种模型转换结构，每一步我们仅需仅算一次，就可以获得所有动作的价值。模型转换示意图如下所示：

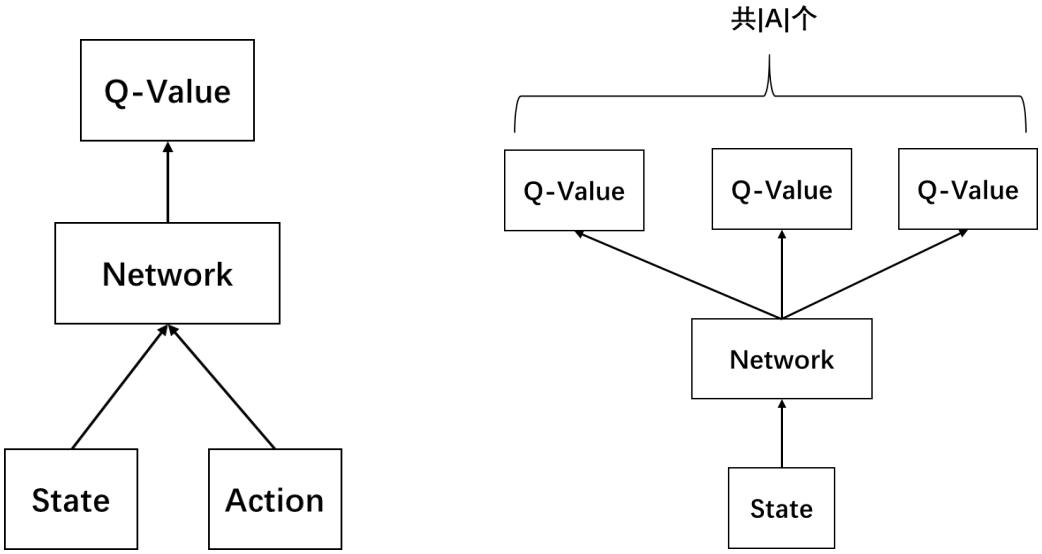


图 3.1 DQN 模型结构

3.2.3 Prioritized Replay Buffer

Q 学习方法基于当前策略进行交互和改进，每一次模型利用交互生成数据进行计算，采用梯度下降方法进行更新网路更新，网络往往需要经过多轮迭代才能够收敛。如果像 Q 学习算法中对样本的处理：使用后的样本被直接丢弃，这样样本使用效率较低。

除了样本使用效率低的缺点，马尔可夫特性决定了两个状态之间所获得数据必然存在时空上的关联，而对于基于最大似然法的机器学习模型而言，有一个前提是训练样本独立同分布，如果这个条件无法保证，训练效果将会大打折扣。

为了解决上述两个问题，Mnih[26]等人提出一种 Replay Buffer 的结构，结构示意如

下：

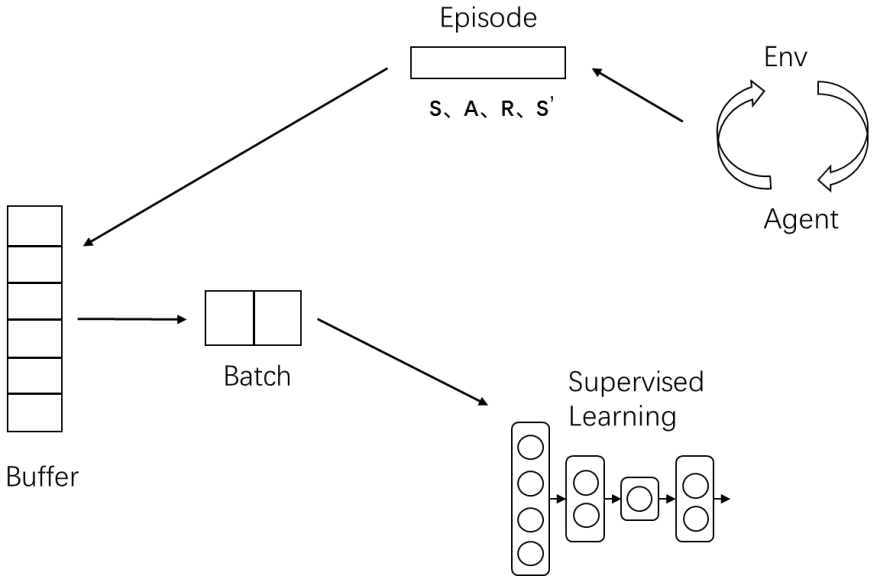


图 3.2 Replay Buffer 示意图

Replay Buffer 中保存了样本交互的信息，且其容量一般设置的比较大。Replay Buffer 的操作过程包含了收集样本和采集样本两个过程。收集样本时，按照时间顺序先后放入 Buffer 结构中，若 Replay Buffer 中存满了样本，那么新的样本将会将时间上最久远的样本覆盖。对于采样操作，会从 Replay Buffer 中均匀随机采取一批样本进行训练。

这种等概率取样仍然存在局限性。对于神经网络而言，有利于神经网络更新的仅仅只有少量的样本，那些能产生较大目标函数梯度的样本，更具有学习价值。

本工作中，提出一种为样本分配权重的方法，使得具有较大更新梯度的样本，在训练时有更高的概率被抽取到。计算方式如下：

$$p_i = \frac{1}{N}|\nabla J| \tag{3.5}$$

$$p(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}} \tag{3.6}$$

当神经网络从 Replay Buffer 中抽取样本时，按照 p(i)分布进行抽取，这样能够保证更具有学习价值的样本有较大的概率被抽取，并用于训练，提高了子模型的学习效率。

3.2.4 Target Netwok

模型另外一个不稳定的因素来自算法本身。由 Q 学习算法可看出 $Q(S_t) \leftarrow Q(S_t) + \alpha(R_{t+1}+\gamma Q(S_{t+1}) - Q(S_t))$，当前时刻状态价值函数的更新通过即时奖励和下一时刻状态价值函数进行更新，这存在巨大的安全隐患。样本数据的差异会造成波动，数据本身存在的波动导致每一轮迭代都会产生一定的波动，采用这种方法进行更新网络参数，一旦上一回合有一定的波动，那么误差立刻会传播到下一个过程，给模型带来巨大的方差。

为了稳定模型，引入另一个和原始网络结构完全一样的模型称为：Target Network 来处理，原本的模型称之为：Behaviour Network。两个模型之间的左右机理如下：

Step 1：训练开始，采用完全相同的参数初始化两个模型；

Step 2：训练过程中，Behaviour Network 不断与环境交互，获得样本；

Step 3：策略迭代和策略评估中，Q 学习得到的目标价值通过 Target Network 计算得到，然后与 Behaviour Network 的估计值进行比较得出目标值并更新 Behaviour Network；

Step 4：完成一定轮数的训练后，Behaviour Network 模型参数同步传递给 Target Network，之后进行下一阶段的学习；

有上述交互步骤容易看出，通过引入 Target Network，计算目标价值的模型在一定时间被固定，这样模型可以减轻模型的波动性。

Deep Q Network 整体的算法叙述如下：

算法 Deep Q Network

Input
Initialize： Replay Buffer D，Behaviour Network，Target Network，Parameters θ, θ^{-}
Start
for episode=1-M do:
  get state s, preprocess and get $\emptyset_1 = \emptyset(s_1)$
  for t=1,T do:
    if random probability p<ϵ *:*
      choose an action randomly
    else:
      $a_t = max\, Q(\emptyset(s_t, a, \theta))$
    perform a, and get s_{t+1} *and reward* r_{t+1}
    preprocess and get $\emptyset_{t+1} = \emptyset(s_{t+1})$, $< \emptyset_t, a_t, r_{t+1}, \emptyset_{t+1} >$ *stored in D*
    random choose a batch $< \emptyset_j, a_j, r_{j+1}, \emptyset_{j+1} >$
    if $\emptyset_{j+1}$ *is terminal:*
      compute $y_j = r_{j+1}$
    else:
      compute $y_j = r_{j+1} + \gamma max Q(\emptyset_{j+1}, a'; \theta^{-})$
    optimize $|y_j - Q(\emptyset_j, a_j; \theta)|^2$ *with GSD*
    $\theta^{-} \leftarrow \theta$ *for each C times*
  end for
end for

算法 3.1 基于 DQN 的无人机路径规划

3.3 实验结果与分析

基于 Pygame 可视化模块，我们分别仿真了离散状态空间和连续状态空间 DQN 算法的性能，在离散状态空间，我们还进一步分析了 DQN 算法应用到实际工程无人机路径规划任务中仍然面临的挑战。

3.3.1 离散空间

仍然在上一章节经典方格世界中对无人机路径规划任务进行仿真，使用 tkinter 模块可视化如下所示：

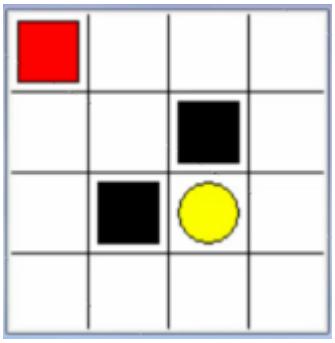


图 3.3 离散状态无人机仿真环境示意图

其中，黄色圆圈代表目标点，黑色方块代表静态规则障碍物，红色方块代表无人机。利用无人机在方格中的位置信息(x,y)完全表征其状态空间$S \in (4,4)$；无人机可采取的动作为：左、右、上、下、右上、右下、左上、左下，$A \in (8,1)$；模型具有马尔可夫特性，并定义环境动力学：当碰撞到黑色方块，则回合结束，无人机重新从初始点开始出发探索，到达黄色目标点，同样回合结束，重新回到出发点；在可行区自由移动时，获得 0 的奖励；碰撞到黑色方块，得到-1 的奖励；到达黄色目标点，得到+1 的奖励。建立强化学习模型 $MDP<S,A,R,S'>\in (16,8)$。

神经网络采用全连接网络，由于环境较为简单，仅用两层神经网络构成；每一层神经元激活函数均为 ReLu；输出层为 Sofmax 层，输出多个 Q 值，对应着多个不同的动作价值函数；模型架构如下图所示：

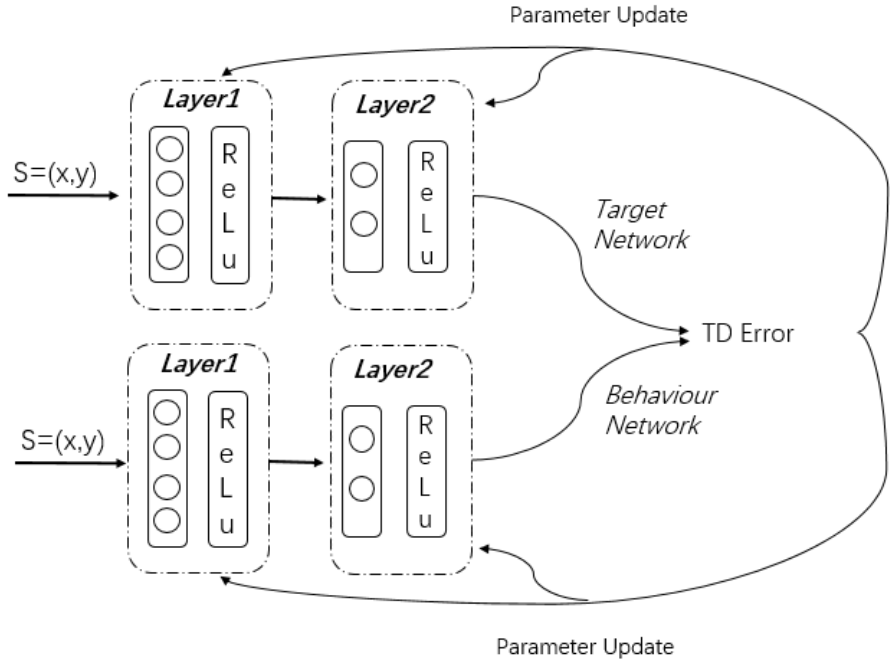
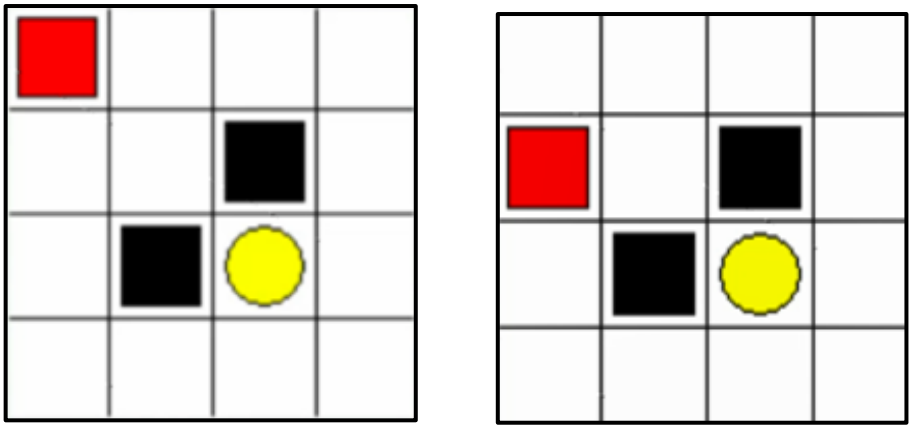


图 3.4 离散状态无人机路径规划神经网络结构示意图

训练结束后，无人机一个路径规划结果动态分解图如下所示：



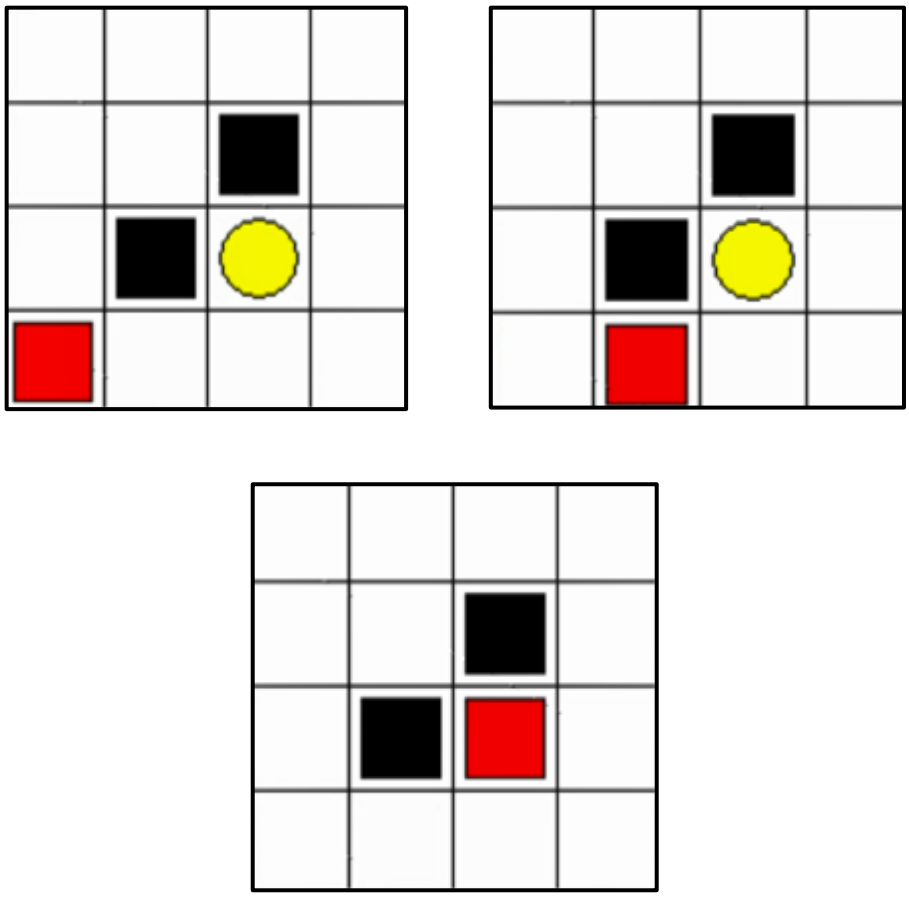


图 3.5 离散状态无人机路径规划仿真结果

3.3.2 连续空间

在 Pygame 给出的可视化环境中搭建对连续空间无人机路径规划的仿真结构，如下图所示：

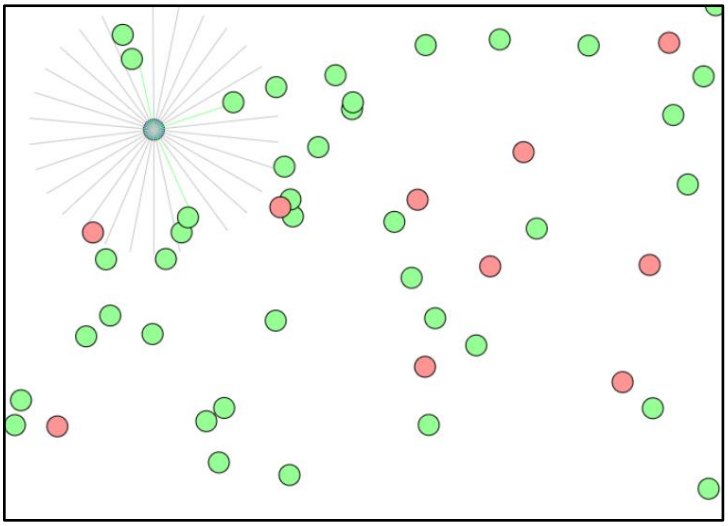


图 3.6 连续状态无人机仿真环境示意图

为了检测算法在高动态的复杂环境中是否仍具有鲁棒规划的效果，在仿真中设置了较多的动态障碍物，形状与无人机相同，模拟相同空域中其他运行的无人机，用绿色圆圈代表。红色圆圈代表在空域中随机生成的目标点，无人机需要通过避障和捕捉，在躲避碰撞的同时到达目标点。青绿色圆圈代表无人机，周围 32 条虚线模拟了无人机的声呐系统，作为对状态捕捉的手段，而有限范围的状态感知也符合实际无人机在环境中无法获得全局信息的情况。

为了便于计算机处理，无人机所在的状态空间进行细密离散化；Pygame 可视化模块采用像素描述，将横纵轴的 100 个主像素离散化，每个小像素再进行 10 个离散；仍然用横纵坐标轴来完全描述无人机的状态；无人机动作空间为 32 个探测射线方向，更符合现实工程无人机中的高机动性；无人机的动力学方程考虑基本的牛顿第二定律，不允许速度产生突变，无人机运行遵循加减速度方程。定义环境动力学：由于环境高度动态障碍物的存在，很容易使无人机发生碰撞，若每次碰撞后都重新设置状态，重新开始训练，将会经历漫长的收敛过程，并且交互序列的残缺性又会严重影响算法性能。因此在本实验中，发生碰撞或到达目标后不再重新设置无人机的状态。发生碰撞得到-1 的奖励，到达目标获得+1 的奖励，其余均为 0；发生碰撞后继续进行探索和训练，到达目标后继续寻找探测范围内最近的目标，进行路径规划。

高度动态的环境给传统神经网络的训练效果带来了巨大挑战。为了更加稳健的训练

和提取时空数据中对决策有帮助的信息，不同于离散状态仿真中采用全连接网络结构进行，而是用 LSTM 结构[27]的神经网络代替；除此，在网络深度上更深，网络参数数量上更多，算法模型和离散状态仿真中相似，下面仅给出在连续环境中所使用的神经网络架构：

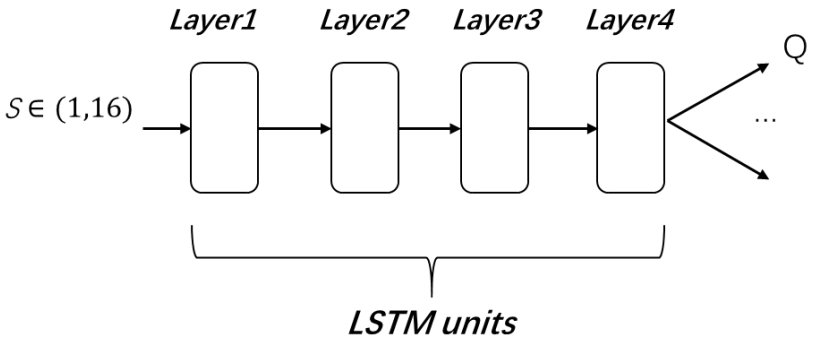
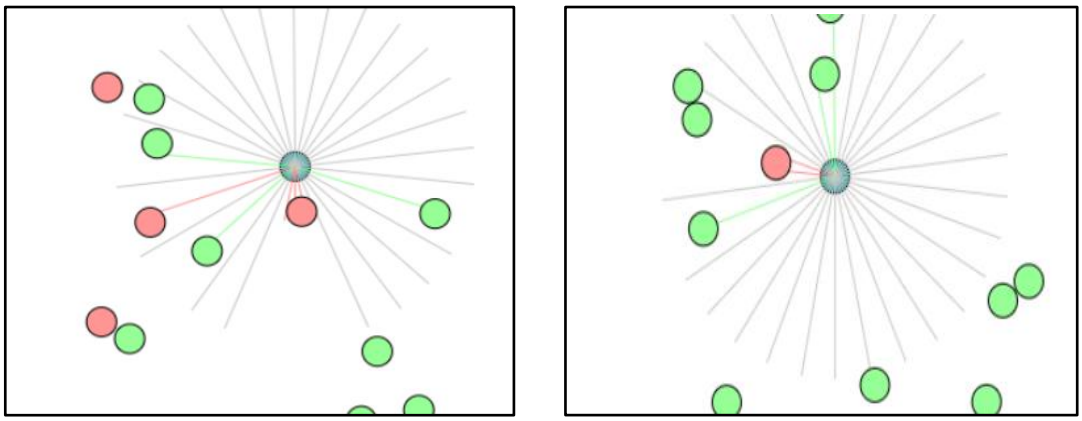


图 3.7 连续状态无人机路径规划神经网络结构示意图

仿真动态结果的部分捕捉如下图：



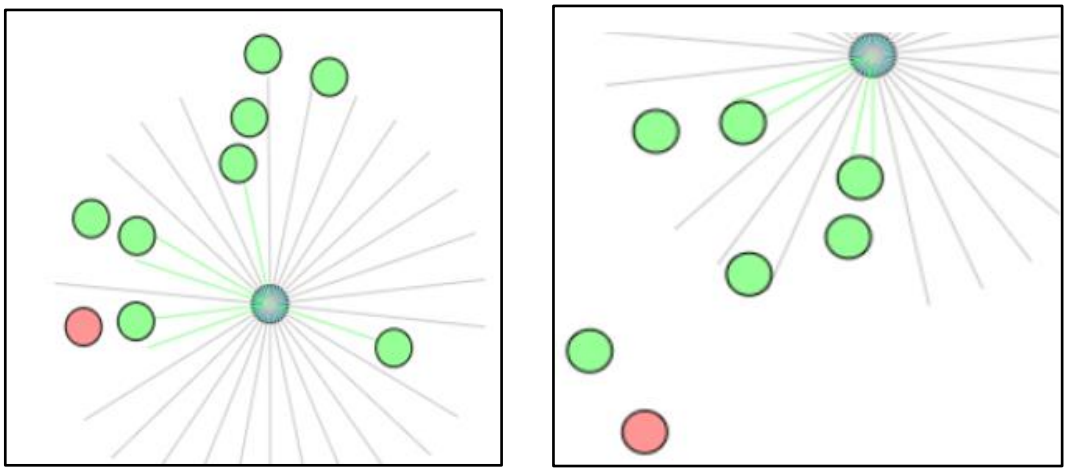


图 3.8 连续状态无人机路径规划仿真结果

通过观察无人机捕捉行为和避障行为的效果，绿色线代表无人机将在该分量的方向上产生一个减速度，红色线代表无人机将在该方向分量产生一个加速度。可以看出，引入 DQN 算法后能够在高动态、环境信息未知的情况下仍然能够进行安全的路径规划。

3.3.3 算法性能分析

仍然在连续空间中，我们分析加入 Replay Buffer 和 Target Network 之后的优化模型和原始模型在训练过程和训练效果方面进行了对比。两种模型的训练代价曲线如下所示：

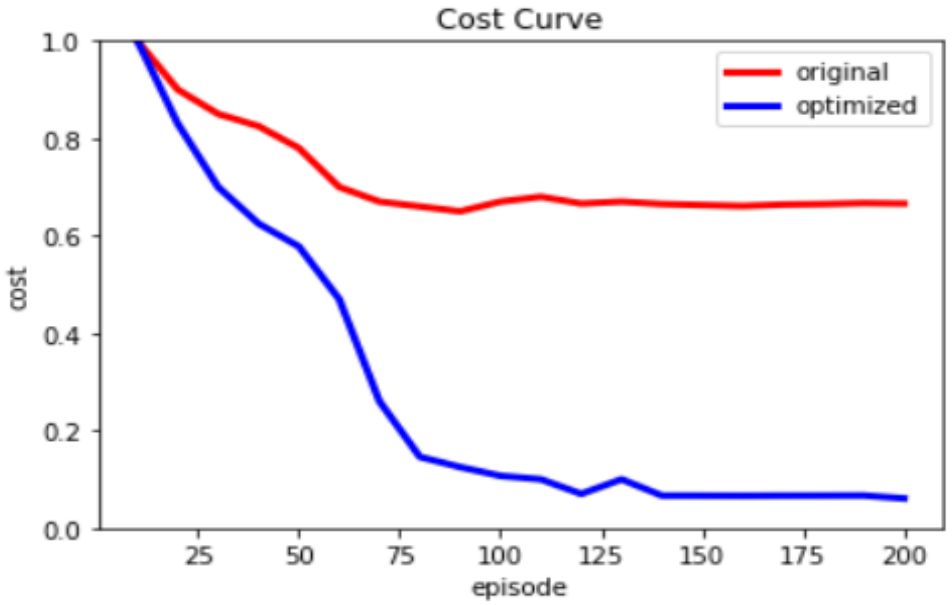


图 3.9 两类模型神经网络代价曲线对比图

对两类模型的训练代价函数求训练过程中的方差，对比结果如下所示：

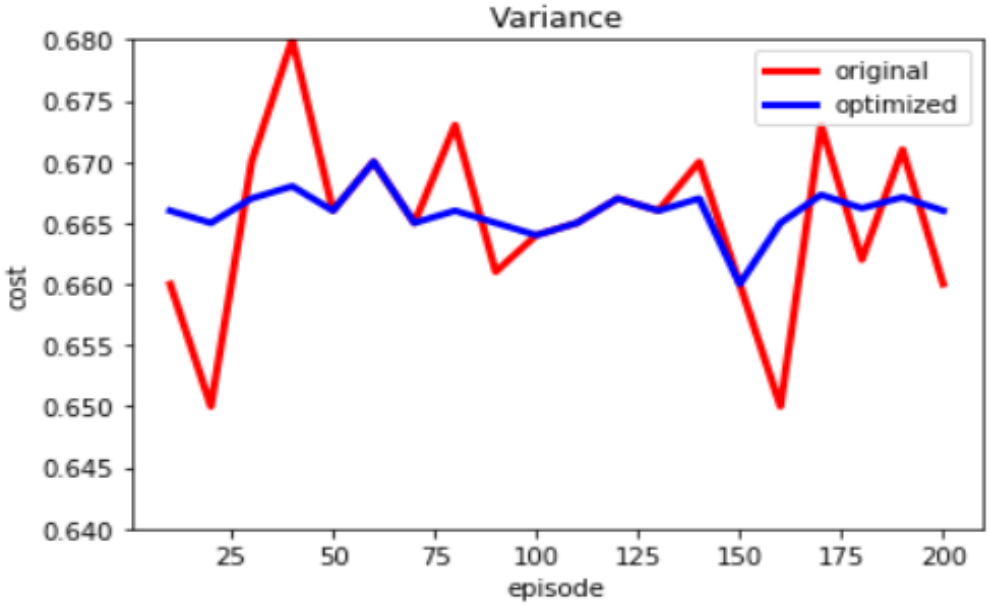


图 3.10 两类模型神经网络方差曲线对比图

综合两种对比，很明显看出，加入 Replay Buffer 和 Target Network 之后的模型在训练速度即收敛性方面更加优异；在模型训练方差方面也有减小方差的作用。

我们随机更改了无人机所处的环境，在不同时刻随机加入新的障碍物，此时的无人机环境除了在高动态和信息不完备的基础上，又考虑了环境的随机干扰，我们观察算法在这种高复杂度环境中的效果。

在训练第 1000 步时，环境中随机进入了 6 架新的无人机，作为 6 个新的障碍物，网络学习代价曲线如下图所示：

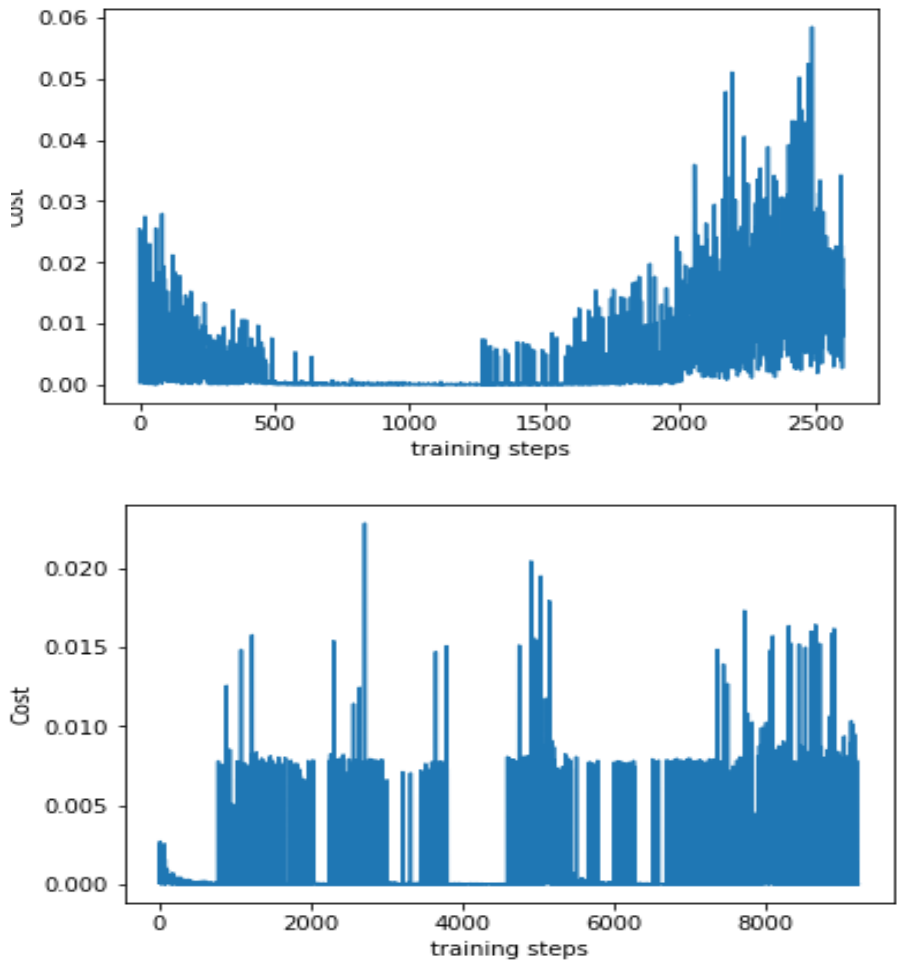


图 3.11 加入随机干扰后神经网络代价曲线

由仿真结果可以看出，环境产生随机变化后，原本已经收敛的网络再次发散，并且在后续与环境的交互中，已经基本失去的智能性，此时 DQN 算法在这种复杂的环境中已经失效。无法进行正常的路径规划。

3.4 小结

本章在第二章的基础上，为了解决经典强化学习运用于实际无人机工程时所面临的挑战，这些主要由于环境高度动态、信息高度不完备所产生的连续控制问题，经典强化学习无法通过动态规划进行贝尔曼优化，产生了维度灾难。

利用深度学习强大的非线性映射能力，将 Q 值求解转化为了神经网络的计算，解决了维度灾难的问题，应用于无人机路径规划任务时，能够在更加复杂的环境中收敛。

我们分析了加入 Replay Buffer 和 Target Network 之后的优化 DQN 和原始模型的差异，仿真结果表明，优化模型在收敛性和方差方面都更加优异。

当我们考虑随机环境时，即一种更加复杂的无人机环境，DQN 算法趋于发散，难以在高度随机化的环境中进行鲁棒的路径规划任务。

DQN 还有更多优化版本的算法，如：DDQN[28]、Dueling DQN[29]、Priority DQN[30]、Noisy Network[31]、Rainbow[32]等。但都在 DQN 原有的基础上进行小小的变动，而 Rainbow 则是考虑了所有的优化方式集合于一体。但这些方法本质上仍然都是一个深度网络在进行计算，仍然无法在更加随机化的环境中进行鲁棒规划。

4 基于分布式模型的路径规划

单一的神经网络难以满足复杂行为和复杂环境所要求的鲁棒性和强泛化能力，本章提出一种分层式的训练结构[33][34]，对无人机的每个特定的功能，设置一个网络来实现它特定的智能行为；更高层的智能行为不需要从零开始训练一个新的网络，而是在它的子功能网络的基础上进行训练。经计算分析，这种结构允许智能体展开更复杂的路径规划，而且收敛性和速度都有极大保障。神经网络的训练是典型的监督学习问题，它的训练需要有监督数据的参与，本方法中，监督数据的产生来自于智能体与环境的交互产生的序列，算法的更新步骤借鉴了深度 Q 学习方法。

状态序列的时序性质是我们想保存的具有参考价值的信息，所以，实现无人机路径规划子功能的每一个子网络都采用循环神经网络的结构。下面将结合结构图，对网络结构进行说明。

4.1 网络结构

分层网络的划分依赖于专家知识，也就是说，网络结构的设计与目标任务息息相关。分层网络的每一层结构仅实现一个基础功能，该层之上的更高层，则是在继承子功能的基础上演化出的新功能。依据此，我们以无人机路径规划为目标，考虑无人机的机动性和路径规划的效果与效率的约束，进行一个分层网络的设计。网络设计的详细结果如下：

第一层：方向控制网络；该网络实现了无人机的转向功能，不考虑转向的动作是否对无人机路径规划有益，仅仅考虑其转向动作不会产生危险，即不允许转向的碰撞。

1) 输入：单个无人机雷达探测到的空间信息向量化；

2) 输出动作：{直行，半左转、左转、半右转、右转}；

3) 训练结果：匀速运动避免碰撞障碍物；

4) 奖励函数：碰撞-1，其他为 0；

该网络与环境不断进行交互，网络逐渐收敛，网络作用于环境的结果，是对智能体进行了控制，环境中的无人机具有了自主避障能力。将训练完毕的网络和作用后的环境保存，用于下一层网络的训练和学习。

第二层：锁定目标控制网络；该网络允许无人机在探测范围内对目标点进行锁定，

在避障的同时，最安全地抵达目标区域，所以该网络仅仅考虑给出一个合理的路径规划结果（躲避空域内所有的障碍安全抵达目的地），但不考虑这个路径规划结果是否最优，也就是说不考虑路径的长短。

1) 输入：单个无人机雷达探测到的空间信息向量化；

2) 输出：{直行，半左转、左转、半右转、右转}；

3) 训练结果：迅速而准确抵达目标像素；

4) 奖励函数:抵达目标像素：1，驶离任务空域：-1，其他（偏航）：-1；

锁定目标控制网络在方向控制网络的基础上构建，这个继承关系通过环境的交互作用进行传递。在第一步网络作用的环境中，采集交互序列对第二层网络进行训练。训练结束后，智能体在具有自主避障能力的基础上，又演化出了捕捉目标点的功能。两层网络堆叠在一起使得无人机具备了最基础的路径规划能力。但这个路径规划结果还需要进一步优化。

第三层：速度控制网络；该网络是对上两步所产生的初步无人机路径规划结果进行优化的网络。它允许无人机在空旷的区域加速飞行，在高威胁区域进行减速飞行，它既考虑了现实无人机机动性的约束，又考虑的最优路径选择。它继承于前两层网络。

1) 输入：单个无人机雷达探测到的空间信息向量化；

2) 输出：不同的速度；

3) 训练结果：加速或减速以避免碰撞；

4) 奖励函数：安全空域加速：1，高威胁区域减速：0，其余为-1；

三层网络的共同作用下，当每层网络收敛完毕后，在复杂的空域中，无人机具备自主感知和自主巡航能力，能够根据目标点的位置，迅速给出一条安全且最短的路径，它的理论计算时间是张量前向通过神经网络的时间，在专用芯片上这个计算过程非常迅速。除此，在规划的路径上，无人机还具备自主的机动性，即：合理地加减速，最短时间抵达目标。

整个分层网络的结构可以用以下示意图表示：

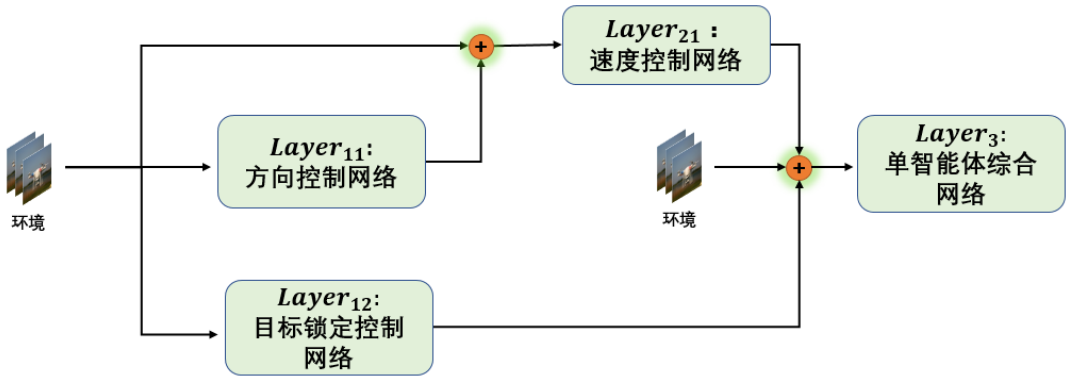


图 4.2 基础层网络结构关系连接图

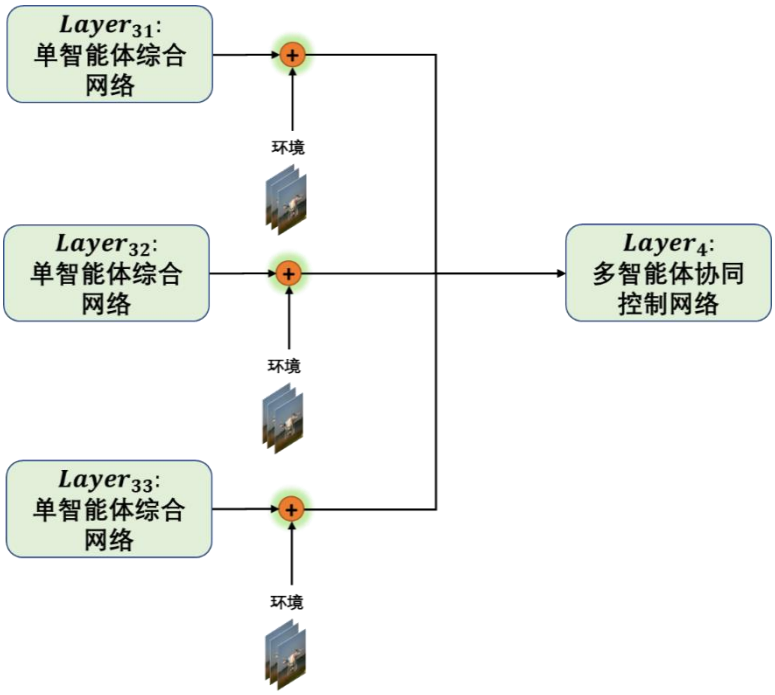


图 4.2 高层与基础层网络结构关系连接图

4.2 网络更新算法

神经网络的训练需要监督数据，监督数据来自智能体与环境的交互产生的一系列序列。为了解决“维度灾难”问题，我们使用深度学习网络对 Q 表格进行拟合：

$$Q(\boldsymbol{S},\boldsymbol{A}) \cong Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta) \tag{4.1}$$

θ代表了神经网络的参数$\theta =$（$weights$，$biaes$），是需要通过监督数据的训练才可以获得的。监督数据来自于Q学习迭代公式:

$$Q_{target}(\boldsymbol{S},\boldsymbol{A},\theta) = r_t + \gamma max Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta) \tag{4.2}$$

神经网络的代价函数设置为最小化$Q_{target}(\boldsymbol{S},\boldsymbol{A},\theta)$与$Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta)$的最小均方误差：

$$cost\ function = min\{\left|Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta) - Q_{target}(\boldsymbol{S},\boldsymbol{A},\theta)\right|^2\} \tag{4.3}$$

产生监督数据的神经网络参数和每次需要更新的神经网络参数是一样的，这会造成巨大的数据波动，为了减小方差，稳定神经网络的训练效果，我们采用时间步差更新方法，暂时冻结一个网络，只更新另一个网络：

$$Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta^-) \leftarrow Q_{eval}(\boldsymbol{S},\boldsymbol{A},\theta)$$
$$Q_{target}(\boldsymbol{S},\boldsymbol{A},\theta) \leftarrow Q_{target}(\boldsymbol{S},\boldsymbol{A},\theta)$$

其中，θ为当前时刻神经网络的参数，θ^-为上一时刻神经网络的参数。

交互数据$\{\boldsymbol{S},\boldsymbol{A}\}$通过智能体与环境交互得出，具有很高的时空关联度，会影响基于统计机器学习的神经网络的训练效果。为了减小这种相关性，采用了一个 Experience Replay Buffer 的暂时存储结构，每次计算所需要的$\{\boldsymbol{S},\boldsymbol{A}\}$，均从该结构中随机抽样得到。

另一方面，交互数据$\{\boldsymbol{S},\boldsymbol{A}\}$的时空关联对模型的认知又是有益的，为了存储这种时序特征，我们将神经网络的结构设置为具有存贮记忆的循环神经网络结构（RNN）。

当各层网络按照上述算法流程更新关闭后，便可以堆叠在一起进行无人机复杂路径规划任务。

4.3 算法

整个算法的叙述过程如下：

算法1： 基于分层强化学习的无人机路径规划

Initialization: Networks structure, Networks parameters and Q-value, $\boldsymbol{\theta}^{-}=\boldsymbol{\theta}=\mathbf{0}$, **Q=0**
Initialization: Experience Replay Buffer, maximum Buffer-Number： $\boldsymbol{N_1}$,
maximum iteration :$\boldsymbol{N_2}$, updating step: $\boldsymbol{T}$
For each layer in networks:
  While Buffer-Number< N_1 **do:**
    reacting with environment and get episode: $<\boldsymbol{S},\boldsymbol{a},\boldsymbol{r},\boldsymbol{S'}>$
    restored in Experience Replay Buffer
    Buffer-Number← Buffer-Number+1

  For each episode in episodes:
    Random choose $<\boldsymbol{S},\boldsymbol{a},\boldsymbol{r},\boldsymbol{S'}>$ from Experience Replay Buffer
    compute: forward propagate Neural Network: $Q_{eval}(\boldsymbol{S},\boldsymbol{A},\boldsymbol{\theta}^{-})$
    $Q_{target}(\boldsymbol{S},\boldsymbol{A},\boldsymbol{\theta})=r_t+\gamma maxQ_{eval}(\boldsymbol{S},\boldsymbol{A},\boldsymbol{\theta}^{-})$

    minimize: $min\{\left|Q_{eval}(\boldsymbol{S},\boldsymbol{A},\boldsymbol{\theta}^{-})-Q_{target}(\boldsymbol{S},\boldsymbol{A},\boldsymbol{\theta})\right|^2\}$
    update: $\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}+\Delta\boldsymbol{\theta}$
    for every T steps, **update** $\boldsymbol{\theta}^{-}$: $\boldsymbol{\theta}^{-}\leftarrow\boldsymbol{\theta}$

  End
End

算法 4.1 基于分层强化学习的无人机路径规划

4.4 计算结果与分析

依据本文所提出的网络结构以及所提及的网络更新策略，在计算机设备上进行了验证并对结果进行了分析，仿真结果验证了本文提出的层次网络结构较单一的深度神经网络具有更高的鲁棒性和更好的收敛性，并利用可视化模块 Pygame，输出了路径规划的结果，以便更直观地判断训练效果。

4.4.1 训练结果

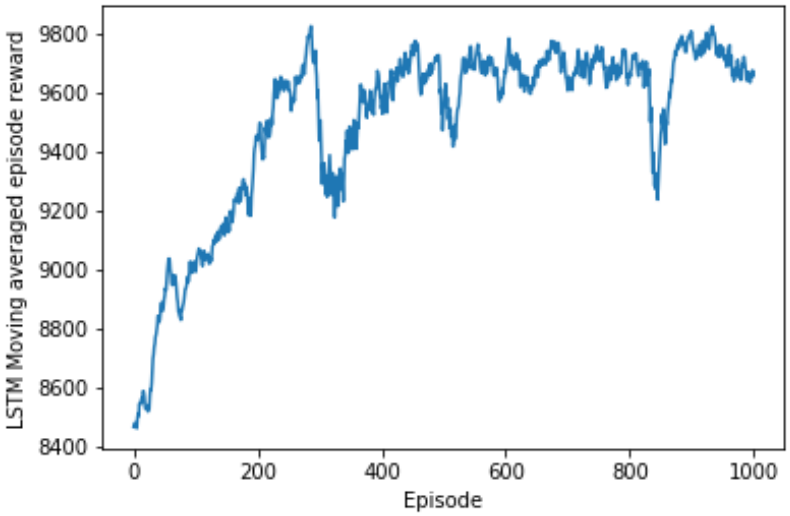
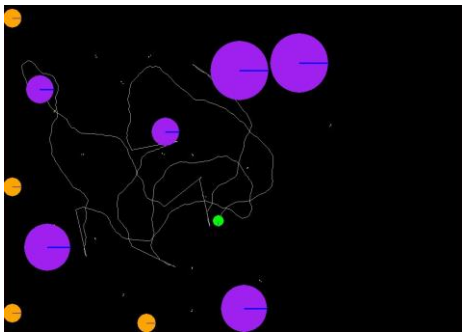


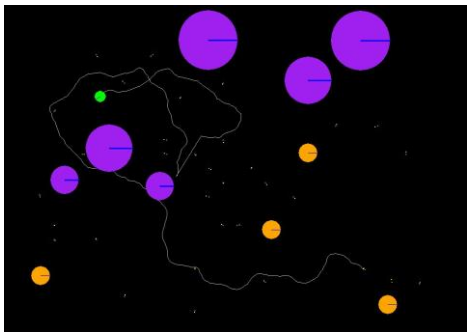
图 4.3 最高层网络训练过程

依据上述更新步骤对网络进行更新，记录每个网络所优化的策略在交互中的表现结果（累积收获值），并作出变化曲线。这里只给出最高层网络累积收获变化曲线，由变化曲线易分析，该模型下，累积收获不断增加，智能体不断学习到我们所定义的环境动力学特征。

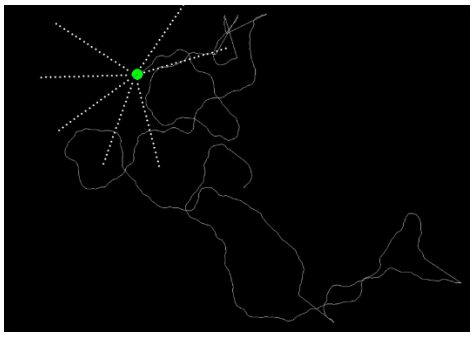
利用 Pygame 可视化模块对每层的学习结果做出可视化，一个 60 万帧训练后的训练结果如下：

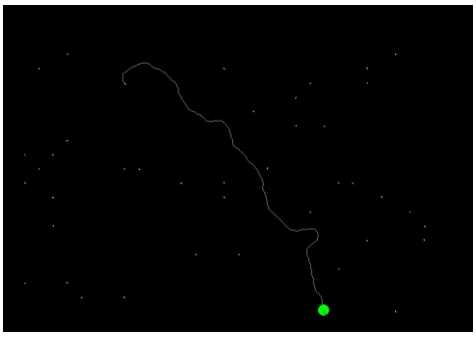


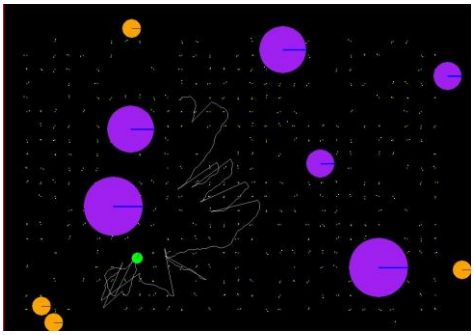
第一层网络训练前



第一层网络训练后


第二层网络训练前


第二层网络训练后


第三层网络训练前

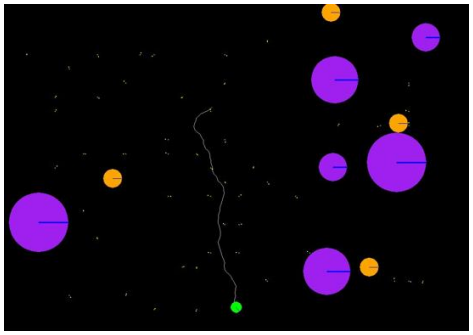

第三层网络训练后

图 4.4 分层强化学习模型仿真结果

第二层网络训练前，智能体需要经过大量的随机探索和无用搜寻才可缓慢到达目标点；当网络收敛后，智能体减小了许多随机游走，能够快速而准确抵达目标区域。

第三层网络收敛之前，由于继承了避障和捕捉目标点的功能，智能体仍然具备一定的环境感知和路径规划能力。但由于没有利用无人机本身的机动性，也没有对规划出的原始路径做优化，所以给出的路径非常曲折，效率较为低下。第三层网络训练完毕后，将安全空域加速和高威胁区域减速的约束加入智能体功能中，因此智能体可以给出一种更为平滑和高效的路径规划结果。

4.4.2 对比分析

为了更好说明本文所提出的网络结构的优点，下面将其与单一深度学习网络的学习效果以及鲁棒性进行了对比。

为了提高深度学习网络的稳定性，这里采用图像来描述智能体的状态，采用卷积神经网络提高图像特征的识别能力。卷积神经网络输出层用 softmax 分成八类，每一类代表一个动作，值得高低反应了动作的价值。深度学习网络结构如下所示：

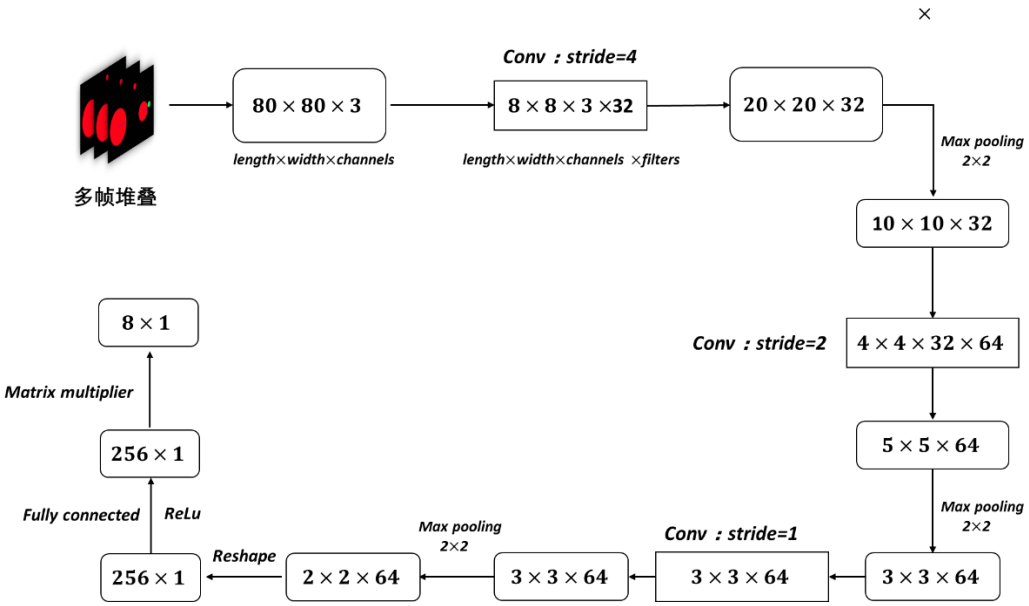


图 4.5 深层强化学习网络结构

当 Experience Replay Buffer 存储满后，开始进行训练，对比两个模型在到达一定的累积奖赏，即达到相同的表现能力时，所耗费的计算时间。除此，对环境进行更改，加入了一个新的动态障碍和一个新的静态障碍，观测两个模型的训练效果和算法收敛性。对比结果如下所示：

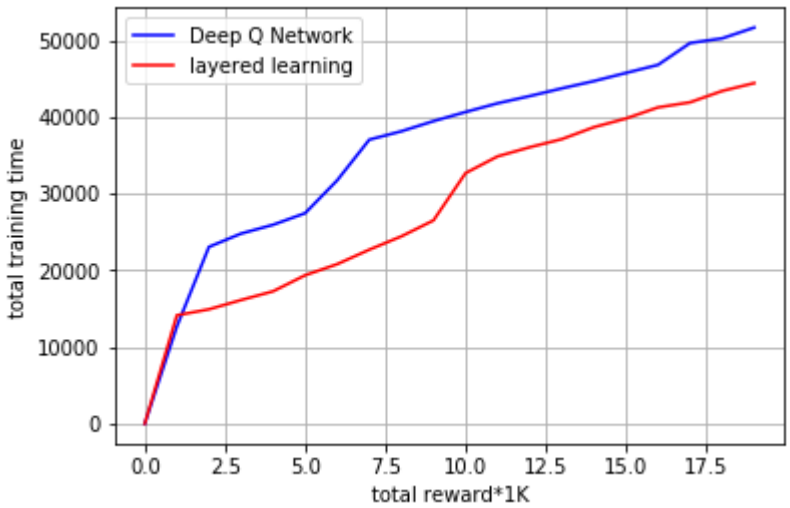


图 4.6 分层结构与单层结构收敛性对比

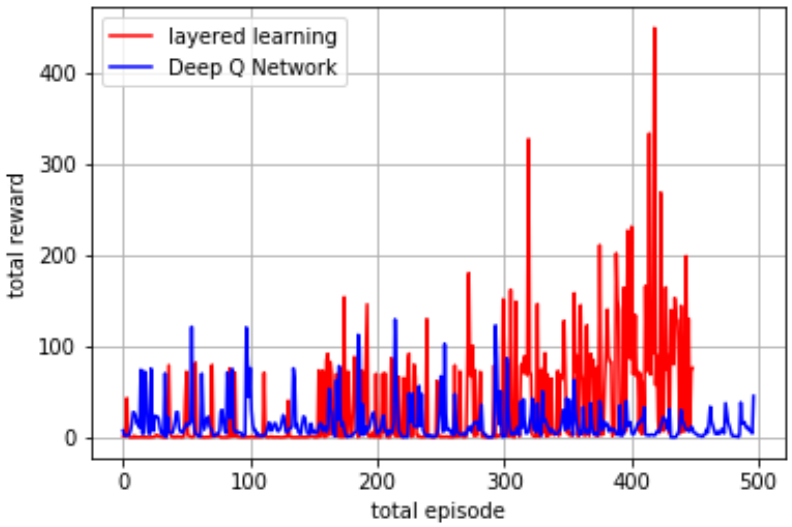


图 4.7 分层结构与单层结构鲁棒性对比

由对比计算结果可以看出，当两个模型在相同环境中，达到相同的累积收获时，单独的深层强化学习模型的计算时间时长，比分层强化学习模型长出很多。这从一个方面验证了分层模型具有更高的收敛性，因为模型具有继承性和递进性：高层所控制的高级智能行为从空白开始训练需要经过漫长的时间，低层的低级智能行为更容易从无到有通过训练获得。所以深度学习网络的收敛性在面临较为复杂的控制任务时会遭遇巨大挑战。分层强化学习模型中，高层高智能动作继承于低层低智能动作，例如：单智能体的加速驶离威胁源或在威胁度较高的空域减速的复杂行为，继承于更简单的匀速飞行网络，在匀速飞行的简单动作上，加入了检测到威胁源的控制行为，学习能力具有递进式，而非从空白开始。

从第二个测试结果可以看出，当环境轻微改变时，单一的深度学习网络的累计收获几乎不再增长，呈现出算法发散趋势；对于分层结构，当环境变化时，累计收获仍在大幅度增长。这表明，单层的深度学习网络虽然在理论上能够完成任何复杂行为的监测和调控，但是深度网络的鲁棒性非常差，当环境轻微改变时，算法便有发散的趋势，直接导致智能体不再从交互过程中学习到知识。分层网络由于每个功能由一个特定的网络结构所控制，对单一动作的控制允许环境在一定范围内变化，实质上分层网络是类似于一种分布式训练和控制的方法，所以它较单一深度网络具有更高的鲁棒性和泛化能力，更

适用于实际无人机场景中，环境复杂多变的环境。

4.4 小结

本章节在深度强化学习理论的基础上，结合无人机实际场景，提出了一种分层网络结构。该网络结构下的学习算法允许训练执行难度更复杂的规划任务，在更复杂和未知的环境中，较单一的深度学习网络也具有更高的稳定性和收敛性。

计算机仿真和对比计算，验证了算法的正确性和优越性。

结 论

本工作从无人机路径规划的背景出发，介绍了无人机路径规划的必要性；通过文献综述分析了无人机路径规划问题中的难点，对现有的算法进行了横向对比，分析了当前无人机路径规划算法的局限性。对比分析表明，现在亟需一种能够在环境高度动态、信息不完备约束条件下能够进行鲁棒规划的算法。

通过引入经典强化学习解决了信息不完备条件下无人机进行路径规划的困难。但是由于经典强化学习对状态空间的限制，导致无法在实际无人机工程问题中使用这类算法。为了解决维度灾难，引入了深度神经网络，扩展为深度强化学习算法，能够解决实际环境中的高动态和信息不完备产生的对无人机路径规划的约束。通过改变环境状态，发现对于随机干扰这类算法具有较低的鲁棒性，仍然无法满足实际环境的约束。

通过引入分布式训练和任务分层，对无人机路径规划任务进行了分层建模，由低至高的递进学习网络，在鲁棒性和收敛性方面，都较深度强化学习有更优的效果，能够在工程中实践。

强化学习在实际使用中仍然面临着奖励稀疏、依赖大量交互数据计算等缺点。如何将强化学习与监督学习、迁移学习进行结合，进而更可能批量应用于实际任务中将会成为这个领域的研究热点。

致 谢

四年大学接近尾声，经过四年的成长，我对学习的意义又有了新的认识。四年一次蜕变，发现自我，而任何一个人的成长都绝不是孤独的旅程，这一路必定有他人的参与。一路上有很多要感谢的人，他们让我的成长之路更加多姿多彩。

感谢四年来北京航空航天大学高等理工学院老师们、辅导员的悉心关照、认真负责，高工永远是我心中的港湾，永远是我的骄傲。

感谢北京航空航天大学电子信息工程学院老师们对我专业课上的辅导和教诲，感谢徐华平老师、李峭老师、孙兵老师、路辉老师、杨晨阳老师，你们的谆谆教诲春风化雨，你们对知识的深邃理解让我时刻充满对知识探索的欲望，你们的学者风范让我深受感染。

感谢杜文博教授对我研究方向的指引与耐心的教导，您对工作的态度与热情，潜移默化地影响着我，不断使我更加端正对科研的态度。您对逻辑的把握时常让我感慨自我洞察力的不足。

特别感谢支持和理解我学业和生活的父母。正是你们的辛勤工作与无私的付出，我才会有机会享受到衣食无忧的生活，有机会接受地区最优质的教育资源。你们的养育之恩我一生也难以报答。特别感谢女朋友李碧月无微不至的关怀和陪伴，你永远是我最爱的人，我爱你就像爱生命。

参考文献

[1] Ziv J, Lempel A. A universal algorithm for sequential data compression[J]. IEEE Transactions on information theory, 1977, 23(3): 337-343.

[2] Ferguson D, Stentz A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2): 79-101.

[3] Kavraki L, Svestka P, Overmars M H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[M]. Unknown Publisher, 1994.

[4] Cortes J, Siméon T, Laumond J P. A random loop generator for planning the motions of closed kinematic chains using PRM methods[C]//Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). IEEE, 2002, 2: 2141-2146.

[5] Dapper F, Prestes E, Nedel L P. Generating steering behaviors for virtual humanoids using bvp control[C]//Proc. of CGI. 2007, 1: 105-114.

[6] Urmson C, Simmons R. Approaches for heuristically biasing RRT growth[C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). IEEE, 2003, 2: 1178-1183.

[7] Kothari M, Postlethwaite I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees[J]. Journal of Intelligent & Robotic Systems, 2013, 71(2): 231-253.

[8] Yershova A, LaValle S M. Motion planning for highly constrained spaces[M]//Robot motion and control 2009. Springer, London, 2009: 297-306.

[9] Bruce J, Veloso M. Real-time randomized path planning for robot navigation[C]//IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2002, 3: 2383-2388.

[10] Hernandez J D On-line 3D path planning for close-proximity surveying

with AUVs[J]. IFAC-Papers OnLine, 2015,48(2):50-55.

[11] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE transactions on evolutionary computation, 2002, 6(2): 182-197.

[12] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on evolutionary computation, 1997, 1(1): 53-66.

[13] 于乃功, 默凡凡. 基于深度自动编码器与 Q 学习的移动机器人路径规划方法[J]. 北京工业大学学报, 2016 (2016 年 05): 668-673.

[14] 梁泉. 未知环境中基于强化学习的移动机器人路径规划[D]. , 2012.

[15] 张凤运. 基于 RBF 网络和 Q 学习的路径搜索与移动导盲系统设计[D]. 西南大学, 2017.

[16] Duguleana M, Mogan G. Neural networks based reinforcement learning for mobile robots obstacle avoidance[J]. Expert Systems with Applications, 2016, 62: 104-115.

[17] Tai L, Liu M. Towards cognitive exploration through deep reinforcement learning for mobile robots[J]. arXiv preprint arXiv:1610.01733, 2016.

[18] 宋勇, 李贻斌, 李彩虹. 移动机器人路径规划强化学习的初始化[J]. 控制理论与应用, 2012, 29(12): 1623-1628.

[19] 赵英男. 基于强化学习的路径规划问题研究[D]. 哈尔滨工业大学, 2017.

[20] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]//Advances in Neural Information Processing Systems. 2017: 6379-6390.

[21] Stone P, Veloso M. Layered learning[C]//European Conference on Machine Learning. Springer, Berlin, Heidelberg, 2000: 369-381.

[22] Vezhnevets A S, Osindero S, Schaul T, et al. Feudal networks for hierarchical reinforcement learning[J]. arXiv preprint arXiv:1703.01161, 2017.

[3] Mirowski P, Pascanu R, Viola F, et al. Learning to navigate in complex environments[J]. arXiv preprint arXiv:1611.03673, 2016.

[23] Tesauro G. Temporal difference learning and TD-Gammon[J]. Communications of the ACM, 1995, 38(3): 58-68.

[24] Watkins C J C H, Dayan P. Q-learning[J]. Machine learning, 1992, 8(3-4): 279-292.

[25] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[26] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.

[27] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. 1999.

[28] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[29] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[J]. arXiv preprint arXiv:1511.06581, 2015.

[30] Zhai J, Liu Q, Zhang Z, et al. Deep q-learning with prioritized sampling[C]//International conference on neural information processing. Springer, Cham, 2016: 13-22.

[31] Fortunato M, Azar M G, Piot B, et al. Noisy networks for exploration[J]. arXiv preprint arXiv:1706.10295, 2017.

[32] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[33] Peters J F, Henry C. Reinforcement learning with approximation

spaces[J]. Fundamenta Informaticae, 2006, 71(2, 3): 323-349.

[34] 周文吉, 俞扬. 分层强化学习综述[J]. 智能系统学报, 2017, 12(5): 590-594.

附 录

附录 A 深度学习模型相关性和解决方式

如图 A1 所示。

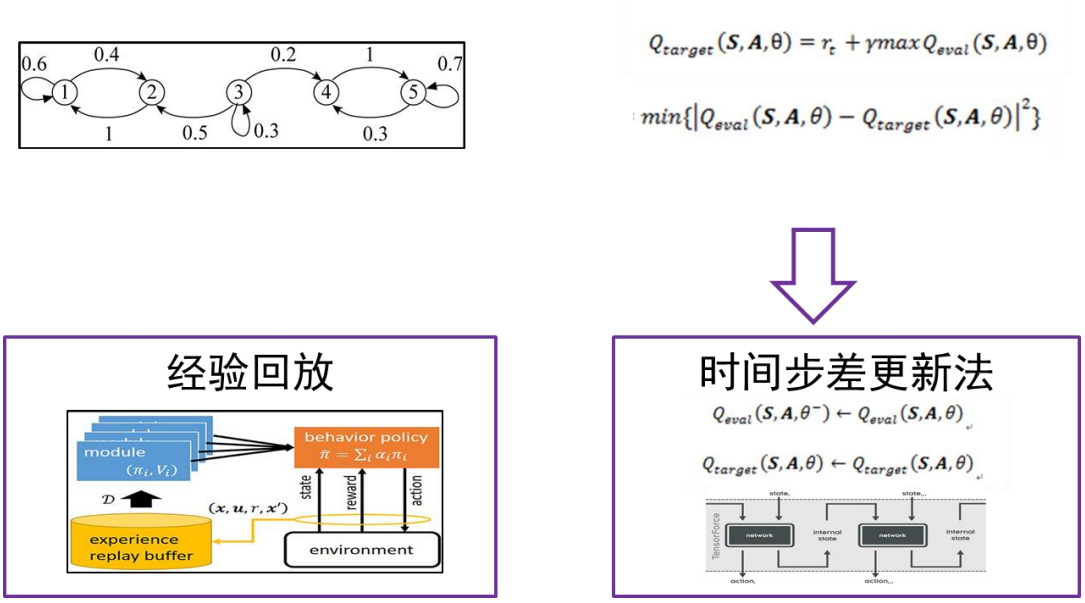


图 A1 算法示意图