学校代码：10285
学　　号：20144227023

蘇州大學
SOOCHOW UNIVERSITY

硕士学位论文

（学术学位）

基于深度Q网络算法与模型的研究

The Research of Algorithms and Architectures on Deep Q-Network

研究生姓名　翟建伟
指导教师姓名　刘全（教授）
专业名称　计算机科学与技术
研究方向　机器学习
所在院部　计算机科学与技术学院
论文提交日期　2017年5月

苏州大学学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

论文作者签名：翟建伟 日 期：2017.6.23

苏州大学学位论文使用授权声明

本人完全了解苏州大学关于收集、保存和使用学位论文的规定，即：学位论文著作权归属苏州大学。本学位论文电子文档的内容和纸质论文的内容相一致。苏州大学有权向国家图书馆、中国社科院文献信息情报中心、中国科学技术信息研究所（含万方数据电子出版社）、中国学术期刊（光盘版）电子杂志社送交本学位论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编学位论文，可以将学位论文的全部或部分内容编入有关数据库进行检索。

涉密论文□

本学位论文属　　在_____年___月解密后适用本规定。

非涉密论文□

论文作者签名：程建伟　日　期：2017.6.23

导 师 签 名：　　日　期：2017.6.23

基于深度 Q 网络的算法与模型研究

中文摘要

深度强化学习是机器学习领域中一个新的研究热点。它以一种通用的形式将深度学习的感知能力与强化学习的决策能力相结合，并通过端对端的方式学习从原始输入到动作输出的一个映射。在许多基于视觉感知的大规模决策任务中，深度强化学习方法已经取得突破性的进展。其中深度 Q 网络方法在解决一类视频游戏任务时表现出了和人类玩家相媲美的水平。然而在一些现实场景下的复杂问题中，深度 Q 网络会面临奖赏的稀疏和延迟、部分状态可观察、收敛速度慢、性能不稳定等一系列问题。本文针对上述问题，从训练算法和模型架构两方面对深度 Q 网络方法进行了改进和完善，并提出三种高效的深度强化学习算法或模型：

（1）针对深度 Q 网络训练算法不能区分不同转移序列之间重要性差异的问题，提出一种基于优先级采样深度 Q 学习算法。该算法使用一种高效的基于优先级的经验回放机制来替代随机采样，提高了有价值转移样本的利用率，并保证样本空间中每个转移序列都有一定大小的采样概率，从而提升了算法收敛的速率。

（2）针对深度 Q 网络算法不擅长解决战略性决策任务的问题，提出一种基于视觉注意力机制的深度循环 Q 网络模型。新的模型架构主要有两处创新点：一是使用由双层门限循环单元构成的循环神经网络模块来记忆较长时间步内的历史状态信息，以使得智能体能够及时响应有延迟的奖赏；二是使用视觉注意力机制自适应地将智能体的注意力集中于面积较小但更具价值的图像区域，减小了模型中可训练的权重数目，从而加快了学习最优策略的进程。

（3）针对深度确定性策略梯度算法在解决连续动作空间问题时性能不稳定的问题，提出一种基于混合目标 Q 值的深度确定性策略梯度方法。新算法通过结合使用在策略的 MC 估计和离策略的 Q 学习方法生成一种混合型的目标 Q 值，降低了目标 Q 值的评估误差，提升了算法在连续动作空间问题中的性能和稳定性。

关键词：深度学习，强化学习，深度强化学习，深度 Q 网络，深度确定性策略梯度

作　　者：翟建伟

指导教师：刘全(教授)

The Research of Algorithms and Architectures on Deep Q-Network

Abstract

Deep reinforcement learning (DRL) is a new research hotspot in the field of machine learning. By using a general-purpose form, DRL integrates the advantages of the perception of deep learning (DL) and the decision making of reinforcement learning (RL), and gains a mapping from the raw inputs to action outputs by the end-to-end learning process. DRL has made substantial breakthroughs in a variety of large-scale decision-making tasks based on the ability of visual perception. In particular, Deep Q-Network (DQN) is able to perform human-level control when handling a kind of video games. However, DQN can be faced with sparse and delayed reward, patially observed states, slow convergence speed and unstable performance problems in some complex problems approaching real scenarios. To alleviate these issues, this paper proposes three novel deep reinforcement learning methods which improve the training algorithm or model architecture on the basis of DQN. The main research is outlined as follows:

i. In order to address the problem that DQN can't dierentiate the importance of distinct transitions, this paper proposes a novel algorithm called Deep Q-Learning with Prioritized Sampling. Compared with DQN, the proposed algorithm uses a highly efficient priority-based experience replay mechanism instead of random sampling in order to increase the utilization rate of valuable samples. Furthermore, the new algorithm ensures that every transition in the sample space can be replayed with a certain probability, eventually leading to an improvement of the algorithm convergence rate.

ii. In view of the problem that DQN is not good at solving strategic decision-making tasks, this paper proposes a novel deep reinforcement learning model architecture called a deep recurrent Q-Network based on visual attention mechanism. The proposed model mainly has two innovations: (i) it uses recurrent neural networks consisting of two-layer

gated recurrent units in order to remember more historical state information of multiple time steps. This can make agents exploit delayed feedback in time to guide its next action selection online. (ii) the visual attention mechanism is used to make agents adaptively focus attention on smaller but more valuable regions of an input image. As a result, the number of weights which need to be trained is reduced, leading to an acceleration of learning near optimal policies.

iii. In order to alleviate the instability problem when using deep deterministic policy gradient algorithm to solve tasks of continuous action space, this paper proposes a novel deep reinforcement learning algorithm called deep deterministic policy gradient with mixed update targets. The new algorithm combines with the on-policy MC estimation and off-policy Q-learning in order to generate mixed target q-values. This can reduce the error when estimating target q-values and improve the performance and stability of the algorithm in continuous action space tasks.

Keywords: Deep Learning, Reinforcement Learning, Deep Reinforcement Learning, Deep Q-Network, Deep Deterministic Policy Gradient

Written by: JianWei Zhai

Supervised by: Prof. Quan Liu

目 录

第一章 引 言

深度学习（Deep Learning，DL）和强化学习（Reinforcement Learning，RL）是目前机器学习领域中较热门的两个分支。其中深度学习的基本思想是通过堆叠多层的网络结构和非线性变换，并组合低层特征以实现对输入数据的分级表达[1]。与深度学习不同的是，强化学习并没有提供直接的监督信号来指导智能体的行为。在强化学习中，智能体是通过试错的机制与环境进行不断的交互，以最大化从环境中获得的累计奖赏[2]。深度强化学习（Deep Reinforcement Learning，DRL）将具有感知能力的深度学习和具有决策能力的强化学习相结合，初步形成从输入原始数据到输出动作控制的完整智能系统[3,4]。

1.1 研究背景及意义

近年来，深度学习已经在计算机视觉[5,6]、语音识别与合成[7,8]、自然语言处理[9]、机器人[10]等领域取得广泛的应用。深度学习方法通常使用多层的神经网络和非线性变换，自动地学习对高维输入数据的认知与表达。强化学习作为机器学习领域的另一个研究热点，已经广泛应用于游戏[11]、仿真模拟[12]、工业控制[13]、机器人控制[14]和参数优化[15]等领域。在强化学习方法中，智能体通过试错的形式与环境进行不断的交互得到反馈信号，并通过最大化累计奖赏的方式学习解决问题的最优策略。随着人类社会的飞速发展，越来越多复杂的决策问题需要利用深度学习的强大感知能力来抽取出大规模输入数据的抽象特征，并以此特征为依据进行自我激励的强化学习，最终求解出问题的最优策略。因此可将具有感知能力的深度学习和具有决策能力的强化学习相结合，以形成深度强化学习方法。

受限于计算能力不足、训练数据缺失等问题，早期的一些结合深度学习与强化学习的工作在解决决策问题时存在较大的局限性。这些工作的主要思路是利用深度神经网络对高维度输入数据进行降维，以便于传统强化学习算法求解最优策略。Lange 等人[16]将深度学习中的自动编码器模型应用到传统的强化学习算法中，提出深度自动编码器（Deep Auto-Encoder，DAE）；Riedmiller 等人[17]使用多层感知器近似表示 Q 值函数，并提出神经拟合 Q 迭代（Neural Fitted Q Iteration，NFQ）算法；

Abtahi 等人[18]用深度信念网（Deep Belief Network，DBN）作为强化学习中的函数逼近器；Lange 等人[19]提出基于视觉感知的深度拟合 Q 学习算法（Deep Fitted Q-learning，DFQ），并将该算法运用到车辆控制任务中。而真正使深度强化学习成为人工智能领域研究热点的是 DeepMind 团队公开发表的相关工作。Mnih 等人[3,4]将深度学习中的卷积神经网络（Convolutional Neural Network，CNN）模型和强化学习中的 Q 学习算法（Q-Learning）相结合，提出深度 Q 网络（Deep Q-Network，DQN）方法。该模型可直接将原始的游戏视频画面作为输入信息，游戏得分作为强化学习中的奖赏信号，并通过深度 Q 学习(Deep Q-Learning)算法进行训练。最终该模型在许多 Atari 2600 游戏上的表现已经赶上甚至超过了专业人类玩家的水平；此后，DeepMind 团队又开发出一款被称为 AlphaGo 的围棋算法[20]。该算法将卷积神经网络、策略梯度（Policy Gradient）和蒙特卡洛树搜索（Monte Carlo Tree Search，MCTS）三种方法相结合，大幅度缩减了有关走子动作搜索空间的大小，提升了对棋盘形式估计的准确性。最终 AlphaGo 以悬殊的比分先后击败欧洲围棋冠军和世界围棋冠军。该事件在人工智能发展史上具有划时代的意义，并成功将深度强化学习技术推向了一个研究高峰。

深度强化学习方法结合使用深度学习的感知能力和强化学习的决策能力，因此它属于一种直接从输入感知信号到输出控制动作的端对端智能系统，并在解决大规模决策问题时具有较强的通用性。深度强化学习方法的工作机制可以大致描述为：（1）在每个时刻，智能体从环境中得到一个高维度的观察，并利用深度学习方法来感知观察，以得到抽象、低维度的状态特征表示；（2）基于智能体与环境交互得到的奖赏信号来评价各个动作的价值函数，并通过某种策略将当前状态映射为相应的动作；（3）环境对当前动作做出反应，以得到下一个观察。通过不断循环以上过程，最终智能体可获得最大的累计回报值。

近年来，深度强化学习技术已经广泛地应用于游戏[3,4,20,21,22]、机器人控制[23,24,25]、参数优化[26]、自然语言处理[27,28]、人机对话[29]、自动驾驶[30]、计算机视觉[31]等领域。另外，Guillaume 和 Wu 等人将深度强化学习方法应用到第一人称的射击类游戏中[32,33]；Schulman 等人提出一种被称为区域信赖的策略最优化（Trust Region Policy Optimization，TRPO）方法，并在一系列连续动作空间下的机器人控

制任务中取得了优异的性能[34]；Levine 等人使用引导式策略搜索（Guided Policy Search）方法完成机器手臂抓取实物的任务[10]；Zoph 等人利用深度强化学习方法来设计神经网络架构[35]；Graves 等人基于深度强化学习框架设计出可微分神经计算机（Differentiable Neural Computer，DNC）[36]。DNC 通过可读写的增强记忆功能可以完成寻找最短路径、推断缺失信息和移动拼图等一系列高层次的任务；Zhu 等人将深度残差网络（Deep Residual Network，DRN）与行动者评论家（Actor-Critic，AC）算法相结合，完成了目标驱动的视觉导航任务[37]。综上所述，深度强化学习方法是一种端对端的智能系统，并适用于现实场景下各种类型的任务，因此该方法被认为是人类迈向通用人工智能（Artificial General Intelligence，AGI）的关键途径之一。

虽然深度强化学习在很多领域已取得若干重要理论和应用成果，但由于深度强化学习问题本身所具有的复杂性，深度强化学习方法距离广度和深度应用依旧存在一定的距离。在深度强化学习的发展历程中，深度 Q 网络方法一直起着举足轻重的作用。因此本文在深度 Q 网络方法的基础上，从训练算法和模型架构两方面对深度 Q 网络进行改进，以提升智能体在解决复杂决策任务时的收敛速度和性能。这些改进的训练算法和模型架构在一定程度上解决了深度强化学习中普遍存在的一些关键性问题：

（1）有价值离线转移样本的利用率不高。在传统的深度 Q 网络训练算法中，通过经验回放机制（Experience Replay）来实时处理模型训练过程中得到的转移样本。该机制每次从样本池中等概率地抽取小批量的样本用于训练模型，然而这种方式使得智能体不能区分出不同转移样本之间的重要性差异，对有价值样本的利用率并不高，并且可能会导致一些重要样本的覆盖和重复利用等问题；

（2）延迟奖赏和部分状态可观测。在一些较为复杂的任务场景中，普遍存在的稀疏、延迟奖赏仍是深度强化学习领域中一个重要的待攻克难题。传统的深度 Q 网络方法缺乏应对有延迟奖赏和部分状态可观测问题的有效办法，因此在面对一些难度较大的战略性决策任务时表现出的性能和稳定性不够理想。

（3）连续动作空间下算法的性能和稳定性不足。在连续动作空间的决策任务中，传统的深度强化学习算法在评估目标 Q 值时仅使用离策略的 Q 学习算法，而

并没有在模型训练的初期高效地利用在线得到的一系列回报值，这使得目标 Q 值的估计不够精确，从而影响了连续动作空间中深度强化学习算法的性能和稳定性。

针对以上问题，本文对深度 Q 网络方法进行了深入研究，并从训练算法和模型架构两方面对深度 Q 网络进行改进和完善，提出了一些解决上述问题的高效算法和模型架构。

1.2 研究现状及趋势

1.2.1 研究现状

国际期刊《Nature》在 2015 年登载了一篇关于深度强化学习的文章，正式提出深度 Q 网络方法[4]。该模型在 Atari 2600 游戏任务中的表现比之前性能最好的强化学习算法更为出色，并在一些游戏中的表现赶上甚至超越了专业的人类玩家。2016 年初，《Nature》期刊又登载了一篇介绍围棋程序 AlphaGo 的文章[20]。AlphaGo 结合使用监督学习、强化学习和蒙特卡洛树搜索，并先后击败欧洲围棋冠军及世界围棋冠军。AlphaGo 的提出在通用人工智能的发展史上具有里程碑式的意义。此后，DeepMind 团队在《Nature》期刊上提出可微分神经计算机[36]，并利用该模型在一些高层次的复杂任务中取得了突破。Silver 等人在 2015 年召开了关于深度强化学习的专题讲座（RLDM 2015）。Abbeel 和 Schulman 等人在 2016 年深度学习暑期学校中讲解了有关深度强化学习的辅导课程。近两年，世界顶级的人工智能会议都举办了与深度强化学习主题相关的研讨会（ICML 2016，NIPS 2015，NIPS 2016，ICLR 2016，ICLR 2017，IJCAI 2016 等）。另外，伯克利、斯坦福等高校相继开设以深度强化学习为主题的课程。《Nature》和《Communications of the ACM》杂志也相继刊载了有关强化学习、深度强化学习方向的综述性文章[38,39]，并且重点强调深度神经网络引发了对强化学习领域研究的新一轮热情。以上相关的研究工作表明，国内外正掀起研究深度强化学习的热潮。

深度 Q 网络方法作为深度强化学习中经典的基础性算法，一直备受研究者们的关注，并且在理论和应用方面取得了一定的突破。尽管如此，受限于算法自身的局限性，深度 Q 网络在完成一些基于视觉感知的决策任务时依然会受到 Q 值估计不精确、有价值样本利用率不高、探索与利用的平衡等问题的阻碍。针对上述问题，

Hasselt 提出深度双 Q 网络（Deep Double Q-Network，DDQN）算法，该算法使用两套不同的网络权重将动作选择和策略评估分离开，降低了过高估计 Q 值的风险[40]。Bellemare 等人在贝尔曼方程中定义一种新的操作符来增大最优动作值和次优动作值之间的差异，以缓和在深度 Q 网络方法中每次选取 Q 值最大动作所带来的评估误差[41]。Schaul 等人在 DDQN 算法的基础上提出一种基于比例优先级采样的深度双 Q 网络（Double Deep Q-network with Proportional Prioritization）。该方法可基于转移样本的 TD 误差值来确定其采样时的优先级大小，从而提高了有价值样本的利用率[42]。Hasselt 等人使用动态归一化操作来替代深度 Q 网络中的区间裁剪方法。该方法可在不流失重要状态信息的前提下，统一不同任务中目标 Q 值的量级，提高了智能体在很多决策任务中的表现[43]。He 等人结合使用深度 Q 网络和一种约束性优化方法来加强有价值奖赏的传播速度，以提升算法性能和缩短模型训练时间[44]。Lakshminarayanan 等人结合使用深度 Q 网络和一种动态跳帧的方式，提出基于动态跳帧的深度 Q 网络（Dynamic Frame Skip Deep Q-Network，DFDQN）算法[45]。Vincent 等人在深度 Q 网络中使用自适应的折扣因子和学习率，提升了算法收敛的速度[46]。Osband 等人提出引导型深度 Q 网络算法（Bootstrapped DQN），该算法使用多分流网络来随机化值函数，扩展了智能体对状态空间的探索范围[47]。Stadie 等人利用一个深度预测模型来评估状态的新颖度，以确定对各个状态的探索力度[48]。Bellemare 等人使用序列密度模型生成那些不可达状态的“伪”访问次数，提升了深度强化学习模型解决复杂决策任务时的性能[49]。Munos 等人提出一种安全、高效的离策略深度强化学习算法，并在 Atari 2600 游戏中取得了更好的表现[50]。上述这些相关研究工作偏向于对深度 Q 学习训练算法的改进。

对于一些较为复杂的决策任务，仅对深度 Q 网络的训练算法进行改进是远远不够的。因此出现了不少改进深度 Q 网络模型架构的研究工作。对模型架构的改进一般是通过向原有网络中添加新的功能模块来实现的。Hausknecht 等人提出深度循环 Q 网络模型（Deep Recurrent Q-Networks，DRQN）[51]。该模型将深度 Q 网络模型中的全连接层替换为由长短期记忆单元（Long Short-Term Memory，LSTM）组成的循环网络层，以记忆时间轴上的历史状态信息。因此该模型适用于一些存在部分状态可观察问题的复杂决策任务。Foerster 等人在 DRQN 模型的基础上，提出分布式

深度循环 Q 网络（Deep Distributed Recurrent Q-Networks，DDRQN)模型，首次解决了部分状态可观察条件下的多智能体通信与合作的难题[52]。Wang 等人在深度 Q 网络模型中引入一种被称为竞争网络的新颖结构。该网络结构可将卷积神经网络提取的抽象状态特征分流为状态值函数和动作优势函数这两个支路，使得值函数的估计更加精确[53]。Junhyuk 等人在深度 Q 网络模型中加入外部的记忆网络部件，以增强智能体应对那些存在部分状态可观测和延迟奖赏问题的高层次决策任务[54]。Rusa 等人在深度强化学习模型中引入一种渐进式神经网络（Progressive Neural Networks）结构。该模型可逐层存储和提取有价值的状态特征，一定程度上解决了从仿真环境中迁移知识到真实环境的难题[55]。Li 等人结合使用监督学习中的循环神经网络和深度 Q 网络，提出了一种混合型深度强化学习模型，提升了智能体应对存在部分可观察问题的任务场景[56]。Narasimhan 等人在深度 Q 网络模型中加入长短期记忆单元，以增强模型对时序数据的记忆能力。该方法首次将深度强化学习模型成功应用于自然语言处理的任务中[27]。Kulkarni 等人提出层次化深度 Q 网络（hierarchical Deep Q-Network，h-DQN）模型。该模型可以在不同的时空尺度上设置层次化的子目标，并通过内在的激励来促进探索，从而增强智能体面对延迟奖赏、稀疏奖赏等问题时的性能[57]。

受限于深度 Q 网络方法自身的局限性，上述相关的算法和模型只适用于离散动作空间下的决策任务。因此很多相关工作都研究如何通过算法框架的完善将深度强化学习方法的应用场景拓宽到连续动作空间场景。Gu 等人基于一种被称为归一化的优势函数（Normalized Advantage Functions，NAF）方法，提出适用于连续动作空间的深度 Q 网络的扩展[25]。另外策略梯度作为一种适用于大规模或连续动作空间问题的高效方法，被广泛应用于各类基于视觉感知的深度强化学习算法中。Schulman 等人提出一种广义优势函数（Generalized Advantage Function），并将其应用于策略梯度项的构造过程中。该种基于广义优势函数的策略梯度方法可以在缩小方差的同时保证较小的估计偏差，并在一些大规模状态和连续动作空间的挑战性任务中取得了不错的表现[58]。Levine 等人提出一种引导式策略搜索方法，实现了机器人领域中从原始输入信号到动作控制的直接映射[10]。Schulman 等人基于自然策略梯度（Natural Policy Gradient）提出了区域信赖的策略最优化（Trust Region Policy

Optimization，TRPO）方法。该方法可限制同一批次数据上新旧两种策略分布的 Kullback-Leibler 差异，以避免策略参数出现过大的更新步[34]。AlphaGo 围棋程序中也使用策略梯度方法对策略参数进行精细的调整[20]。上述研究工作都通过各种版本的策略梯度方法直接优化用深度神经网络参数化表示的策略。然而在某些场景下，纯策略梯度方法会因为梯度项方差过大、训练数据不足等问题而导致出现局部最优解问题。因此可将经典的 AC 框架与策略梯度方法相结合。Lillicrap 等人模仿深度 Q 网络扩展 Q 学习的思想对确定性策略梯度（Deterministic Policy Gradient，DPG）方法进行了完善，提出一种基于 AC 框架的深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）算法。该算法不仅可以解决离散动作空间问题，也适用于一系列连续动作空间中的控制任务[23]。Heess 等人基于 AC 框架提出一种适用于连续动作空间问题的通用框架，被称为随机值梯度（Stochastic Value Gradient，SVG）方法[59]。Wang 等人基于 AC 框架提出一种具有稳定、高效经验回放机制的深度强化学习算法。该算法将基于偏移修正的截断式重要性采样、随机化的竞争网络结构、TRPO 策略优化等方法相结合[60]。Peng 等人融合多个策略网络和值网络，提出混合型行动者评论家（Mixture of Actor Critic Experts，MACE）策略优化方法[61]。Heess 等人则首次将循环神经网络用于基于 AC 框架的策略梯度方法中，提出循环确定性策略梯度（Recurrent Deterministic Policy Gradient，RDPG）和循环随机值梯度（Recurrent Stochastic Value Gradient，RSVG）方法[62]。RDPG 和 RSVG 算法可以应对一些存在部分状态可观察问题的连续动作空间任务。Hausknecht 等人将基于 AC 框架的策略梯度方法扩展到参数化的连续动作空间问题中[63]。

1.2.2 研究趋势

为了拓宽深度强化学习方法的应用场景，目前出现了不少深度强化学习领域中的前沿研究方向。其中一个热门方向是利用多任务迁移学习以提高深度强化学习算法同时处理多个复杂决策任务的能力。Parisotto 等人提出一种基于行为模拟的多任务迁移深度强化学习方法。该方法通过监督信号的指导使得单一的策略网络学会对应任务的最优策略，并将其迁移到相似的新任务场景中[64]。Rusa 等人将策略蒸馏（Policy Distillation）方法应用到深度强化学习算法中，提出一种高效的多任务迁移学习方法[65]。Rusa 等人提出基于渐进式神经网络（Progressive Neural Networks）的

迁移深度强化学习方法[55]。Schaul 等人提出一种通用的值函数逼近器（Universe Value Function Approximators，UVFAs），以同时泛化状态和目标空间，并将已学习到的知识迁移到那些环境动态性相同但目标不同的新任务中[66]。Fernando 等人提出一种被称为 PathNet 的迁移学习方法。该方法使得深度强化学习智能体可以在面对新任务时发现训练网络中可复用的部分[67]。另外在一些复杂的决策任务中，可以利用分层强化学习（Hierarchical Reinforcement Learning，HRL）将最终目标分解为多个子任务以学习层次化的策略，并通过组合多个子任务的策略形成最优的全局策略。Kulkarni 等人基于时空抽象和内在激励等技巧，提出层次化的深度 Q 网络算法（hierarchical Deep Q-Network，h-DQN）[55]。Krishnamurthy 等人结合使用时空抽象和深度神经网络，提出一种基于内部 Option 的深度 Q 学习（deep intra-option Q-learning）模型[68]。Kulkarni 等人提出深度后续强化学习（Deep Successor Reinforcement Learning，DSRL）。该方法使得智能体对突出奖赏（Distal Reward）的变化更加敏感，从而为分层深度强化学习提供更加高效的分解子目标的方法[69]。此外针对一些真实场景下的复杂决策问题，出现了一些多智能体之间可以相互合作、通信及竞争的深度强化学习模型[52,70]。

目前，随着深度学习领域中各种新颖功能网络模型的不断提出，深度强化学习模型正向结构多样化、模块复杂化的方向发展，以提高模型解决一些复杂决策任务的能力。Junhyuk 等人通过在模型中加入外部的记忆网络和由 LSTM 组成的循环神经网络部件，提出基于反馈控制机制的记忆深度循环 Q 网络（Feedback Recurrent Memory Q-Network，FRMQN）模型。该模型初步拥有了记忆和推理能力，并在一些高层次的启发式认知任务上取得了较好的表现[54]。Graves 等人将神经网络与可读写的外部存储器相结合，提出可微分神经计算机模型。该模型不仅可以通过试错或转移样本进行学习，也能像传统的计算机一样处理数据。因此可微分神经计算机是目前最接近数字计算机的神经计算系统[36]。Duan 等人创新性地提出将深度强化学习智能体编码到循环神经网络权重中的思想[71]。Blundell 等人仿照哺乳动物的学习系统设计出一种模型无关的情节式控制器（Model-Free Episode Control，MFEC）。该控制器可以快速存储和回放状态转移序列，并将回放的序列整合到结构化知识系统中，使得智能体在面对较复杂的决策任务时，能够在较短时间内达到人类玩家的

水平[72]。Tamar 等人通过在模型中嵌入一种完全可微的卷积神经网络规划模块，提出了值迭代网络（Value Iteration Network，VIN）模型。VIN 不仅可学习从状态到决策的直接映射，还可在当前的任务环境下学会长远的规划以做出更好的决策[73]。

针对连续动作空间问题，出现了一批基于异步更新步骤的深度策略梯度方法。Mnih 等人提出异步的优势行动者评论家（Asynchronous Advantage Actor-Critic，A3C）算法。该算法可以利用 CPU 多线程的功能并行执行多个智能体，并在各种连续动作空间中的控制任务上表现优异[21]。Babaeizadeh 等人提出一种混合型 A3C 算法。该算法可以结合使用 CPU 和 GPU 的计算资源[74]。Jaderberg 等人结合使用 A3C 算法和多个无监督的辅助任务，提出 UNREAL 深度强化学习算法。该算法在 Atari 2600 和 Labyrinth 游戏平台上都取得了最优的性能[22]。Mirowski 等人基于结合使用 A3C 算法和额外辅助任务的思想，将模型的应用范围扩展到复杂环境下的导航任务中[75]。Wu 等人在 A3C 算法中引入课程学习（Curriculum Learning）机制，使模型在第一人称视角的 Doom 游戏上取得了优异的表现[33]。此外，结合使用策略梯度和基于值函数的强化学习方法也是一个研究趋势。Donoghue 等人结合使用策略梯度和 Q 学习方法，提出一种被称为 PGQ 的算法。PGQ 算法具有数据利用率高和性能稳定的优点，并在一些决策任务中的性能超过了传统的深度 Q 网络和 A3C 算法[76]。类似地，Gu 等人结合无偏的在策略梯度估计方法和采样高效的离策略强化学习方法，提出一种适用于连续动作空间问题的算法，被称为 Q-Prop[77]。

1.3 研究内容

本文针对解决深度强化学习问题时存在的稀疏和延迟奖赏、部分状态可观察、收敛性能不稳定等问题，围绕深度 Q 网络方法展开研究，并从训练算法和模型架构两方面对其进行改进和完善，设计出若干高效的深度强化学习算法。文中着重分析各算法的性能和稳定性，并通过实验验证算法的性能和适用场景。具体的研究内容包含以下三个部分：

（1）提出一种基于优先级采样的深度 Q 学习算法。传统的深度 Q 学习算法使用一种经验回放机制来去除转移样本之间的相关性，以满足深度神经网络模型训练的要求。然而经验回放机制中采用的随机采样方式使得智能体区分不出不同转移样

本之间的重要性差异，并且可能会导致一些有价值样本被覆盖及重复利用等问题。针对上述问题，提出一种基于优先级采样的深度 Q 学习算法。该算法使用一种优先级的采样方式替代原先的随机采样，提高了有价值转移样本被采样的概率，从而加快智能体学习最优策略的进程。另外，新的经验回放机制还能在学习初期提高带有正奖赏转移样本的利用率，一定程度上缓解了在复杂问题中存在的稀疏奖赏问题。新算法不仅提高了收敛速度，并且在许多基于视觉感知的视频游戏任务中取得了更优的表现。

（2）提出一种基于视觉注意力机制的深度循环 Q 网络模型。在面对存在延迟奖赏和部分状态可观测问题的战略性挑战任务时，深度 Q 网络模型的性能急剧下降。针对上述问题，新的网络模型主要做了两方面的改进：一是引入由双层门限循环单元（Gated Recurrent Units，GRU）构成的循环神经网络来记忆较长时间步长内的历史状态信息，以缓解智能体在解决战略性任务时存在的延迟奖赏和部分状态可观测的问题；二是通过在模型中加入视觉注意力机制（Visual Attention Mechanism，VAM），使得智能体能够在训练过程中自适应地将注意力集中于面积较小但更具价值的图像区域，减少了网络模型可学习权重的数目，提高了智能体收敛的速度。该种基于视觉注意力机制的深度循环 Q 网络模型一定程度上解决了之前深度强化学习方法不适用于战略性决策任务的问题。

（3）提出一种基于混合目标 Q 值的深度确定性策略梯度算法。针对连续动作空间下的决策问题，该算法使用权重不同的两个深度神经网络分别表示确定性策略和值函数。其中，表示确定性策略的网络用于更新行动，对应 AC 框架中的行动者；表示值函数的网络用于逼近动作值函数，并提供梯度信息，对应 AC 框架中的评论家。该算法在估计目标 Q 值时，将在策略的蒙特卡罗（Monte Carlo，MC）估计方法和离策略的 Q 学习方法相结合，使得目标 Q 值的估计更加精确，从而提高了算法在解决连续动作空间问题时的性能和稳定性。

1.4 论文组织结构

本文共六章，具体内容安排如下：

第一章 引言。本章介绍深度强化学习的研究背景以及与本文内容相关的研究

现状和趋势，最后引出本文的主要研究内容。

第二章 背景知识。首先介绍强化学习中的一些预备知识和经典算法，包括马尔科夫决策过程、蒙特卡罗方法、Q 学习算法、AC 算法；然后介绍深度 Q 网络方法的相关背景知识。

第三章 基于优先级采样的深度 Q 学习算法。本章首先介绍深度 Q 学习算法的相关概念和流程，然后在其基础上引入优先级的采样方式，并提出基于优先级采样的深度 Q 学习算法。最后通过一些经典的 Atari 2600 游戏来验证新算法的性能和稳定性。

第四章 基于视觉注意力机制的深度循环 Q 网络模型。首先分析传统的深度 Q 网络模型在解决战略性任务时存在的不足，然后在原有模型基础上引入循环神经网络和视觉注意力机制模块，并提出基于视觉注意力机制的深度循环 Q 网络，阐明其具体模型架构的组成和训练流程。最后通过一些经典的战略性 Atari 2600 游戏来验证新模型的性能和稳定性。

第五章 基于混合目标 Q 值的深度确定性策略梯度算法。首先介绍基于 AC 框架的深度策略梯度方法的相关概念，然后介绍一种混合目标 Q 值的构造方法，并将其应用到深度确定性策略梯度算法中。最后通过一些经典的连续动作空间中的控制问题来验证新算法的有效性，证明了算法具有更好的性能和稳定性。

第六章 总结和展望。总结全文的研究工作，提出下一步可开展的研究方向，并对未来的工作进行展望。

第二章 背景知识

本章首先介绍马尔科夫决策过程的概念和一些相关术语，包括策略、状态动作值函数、最优状态动作值函数等，这些是深度强化学习方法的理论基础。随后介绍一些经典的强化学习算法，包括蒙特卡罗方法、Q 学习、AC 算法。最后介绍深度Q网络方法相关的背景知识。

2.1 马尔科夫决策过程

若一个强化学习问题满足马尔科夫性质，则可以将该问题称为一个马尔科夫决策过程(Markov Decision Process, MDP)[2]。MDP 可用来对强化学习问题进行建模。通常 MDP 模型被描述成一个四元组$\langle X,U,\rho,f\rangle$，其中：

（1） X 是所有环境状态的集合， $x_t \in X$ 表示智能体在 t 时刻所处的状态；

（2）U 是智能体可执行动作的集合，$u_t \in U$ 表示智能体在 t 时刻所采取的动作；

（3）$\rho: X\times U \to \mathbb{R}^n$ 为奖赏函数。$r_t \sim \rho(x_t,u_t)$ 表示智能体在状态 x_t 下执行动作 u_t 获得的立即奖赏值 r_t；

（4） $f: X\times U\times X \to [0,1]$ 为状态转移函数，表示智能体在 t 时刻位于状态 x_t 下执行动作 u_t 转移到下一状态 x_{t+1} 的概率，可以表示为 $x_{t+1}\sim f(x_t,u_t)$。

在强化学习中，从 t 时刻开始到 T 时刻情节结束时所获得的累计折扣奖赏定义为：

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{2.1}$$

其中，$0 \le \gamma \le 1$ 是折扣因子，用来权衡未来奖赏对累积奖赏值的影响力度。策略 $h(x,u) = \Pr(u_t = u | x_t = x)$ 表示从状态空间 X 到动作空间 U 的一个映射。状态值函数表示在智能体遵循策略 h 的情况下，从状态 x 开始到情节结束时获得的期望回报：

$$V^h(x) = \mathbb{E}_h\left[R_t \middle| x_t = x\right] \tag{2.2}$$

状态动作值函数表示在当前状态 x 下执行动作 u，并一直遵循策略 h 到情节结束这一过程中智能体所获得的累积回报值：

$$Q^h(x,u)=\mathbb{E}_h\left(R_t\left|x_t=x,u_t=u\right.\right) \tag{2.3}$$

对于所有的状态动作对，如果一个策略 h 的期望回报大于或等于其它所有策略的期望回报，那么称该策略为最优策略 h^*：

$$h^*(x,u)=\arg\max_h Q^h(x,u) \tag{2.4}$$

最优策略可能不只一个，但它们共享一个最优状态动作值函数。该函数能够最大化每个状态动作对的值函数：

$$Q^*(x,u)=\max_h Q^h(x,u),\forall x\in X,u\in U \tag{2.5}$$

最优状态动作值函数遵循贝尔曼最优方程（Bellman Optimality Equation）：

$$Q^*(x,u)=\mathbb{E}_{x'\sim X}\left[r+\gamma\max_{x'}Q(x',u')\middle|x,u\right] \tag{2.6}$$

在传统的强化学习中，可通过贝尔曼方程的迭代求解最优状态动作值函数：

$$Q_{i+1}(x,u)=\mathbb{E}_{x'\sim X}\left[r+\gamma\max_{x'}Q_i(x',u')\middle|x,u\right] \tag{2.7}$$

在上式中，当 i 趋向于正无穷时，Q_i 趋向于最优状态动作值函数 Q^*。即循环迭代上式会使得状态动作值函数最终收敛，从而得到最优策略：$\pi^*=\arg\max_{u\in U}Q^*(x,u)$。然而对于大规模状态动作空间的复杂决策问题，迭代贝尔曼方程求解最优 Q 值函数的计算代价太大。因此可利用深度神经网络等非线性函数逼近器来近似表示值函数或策略，这也是深度强化学习方法的核心思想。

2.2 强化学习经典算法

强化学习方法根据是否依赖于环境动态模型，分为基于模型的强化学习算法和模型无关的强化学习算法。基于模型的强化学习算法需要事先构造出奖赏函数和状态转移函数的具体形式，再利用贝尔曼方程迭代求解最优状态值函数或者最优状态动作值函数。相关的经典算法有值迭代和策略迭代等；模型无关的强化学习算法则不依赖于环境动态模型，它们通过与环境的不断交互得到训练样本，并根据这些样本数据来学习一个有效策略。此类算法包括蒙特卡罗、时间差分（Temporal Difference，TD）等方法。其中 Sarsa 和 Q 学习是 TD 方法中最经典的两个算法。另外强化学习中还有一类直接优化策略的方法，被统称为策略梯度方法。该种方法在

策略空间中通过优化过程搜索最优策略。AC 方法则结合使用基于值函数的 TD 方法和策略梯度方法，具有更好的性能和稳定性。

2.2.1 蒙特卡罗方法

蒙特卡罗方法的思想是以部分估计整体，并通过统计模型或者抽样获得问题的近似解。将上述思想引入到强化学习中，可得到一种无模型的强化学习方法。此类方法不需要事先知道环境的动态模型，仅通过智能体与环境不断交互获得样本数据，并利用这些序列样本计算最优值函数。MC 方法是基于完全抽样机制的，只有当智能体所处的状态到达终止状态时，估计的值函数和对应的策略才会发生改变。因此 MC 方法适用于存在终止状态的情节式任务。在智能体与环境的交互过程中，所有从环境中得到的立即奖赏值都需要被记录下来，以得到累计折扣奖赏和。具体的值函数更新公式为：

$$V^h(x)=V^h(x)+\alpha\left[R_t-V^h(x)\right] \tag{2.8}$$

其中，α 代表学习率，$R_t=r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\ldots+\gamma^{T-t-1}r_T$ 代表智能体从状态 x 出发到情节结束时所获得的累计折扣奖赏。MC 方法在计算值函数时，不依赖于其它状态的值函数。因此在学习初期各状态值函数的估计还不精确的时候，可利用 MC 方法来辅助估计值函数，以提高算法的稳定性和性能。

2.2.2 Q 学习算法

Q 学习算法是一种经典的模型无关的强化学习算法。该算法在迭代时采用的是状态动作对的奖赏和值函数 $Q(x,u)$，而并非状态值函数 $V(x)$。由于行为策略和评估策略不一致，因此 Q 学习属于一种离策略的强化学习方法。1-步的 Q 学习更新公式为：

$$Q(x_t,u_t)\leftarrow Q(x_t,u_t)+\alpha\left[r_{t+1}+\gamma\max_u Q(x_{t+1},u_{t+1})-Q(x_t,u_t)\right] \tag{2.9}$$

Q 学习算法首先初始化所有状态动作对的 Q 值；然后智能体在状态 x_t，根据 $\varepsilon-greedy$ 策略选择动作 u_t，并得到转移样本序列 $(x_t,u_t,r_{t+1},x_{t+1})$；其次根据公式 2.9 更新 Q 值。当智能体到达目标状态时，算法终止本次迭代过程。算法重新回到初始状态并开始新的迭代循环，直到所有状态动作对的 Q 值函数收敛为止。典型的单步 Q 学习算法具体流程如下：

算法 2.1 Q 学习算法

1：**初始化**：对于 $\forall(x,u)\in X\times U, Q(x,u)=0$

2：**Repeat（对每一个情节）**：

3： 初始化开始状态为 x_0

4： **Repeat（对于情节中的每一步）**：

5： 根据 $\varepsilon-greedy$ 策略选择动作 u_t，得到立即回报 r_t 和下一个状态 x_{t+1}

6： $Q(x_t,u_t)\leftarrow Q(x_t,u_t)+\alpha\left[r_{t+1}+\gamma\max_u Q(x_{t+1},u_{t+1})-Q(x_t,u_t)\right]$

7： $x_t\leftarrow x_{t+1}$

8： **Until** x_t 是终止状态

9：**Until** 所有的 $Q(x,u)$ 收敛

10：**输出**：最优策略 $h(x)=\arg\max_u Q(x,u)$

2.2.3 行动者评论家算法

行动者评论家算法结合使用值函数和策略梯度方法，是一种利用独立存储结构直接表示策略的 TD 方法。其中，行动者部分根据学习到的值函数动态地更新策略参数；评论家部分则用来估计当前状态（动作）的值函数，并评价行动者的当前策略。因此，AC 算法可同时对值函数和策略进行估计，并通过梯度信息在增大期望总回报的方向上更新策略参数。虽然 AC 算法与传统的策略迭代方法有着较大区别，但其基本的学习框架也是由策略评估和策略改进两部分组成。图 2-1 给出了 AC 算法的基本框架：

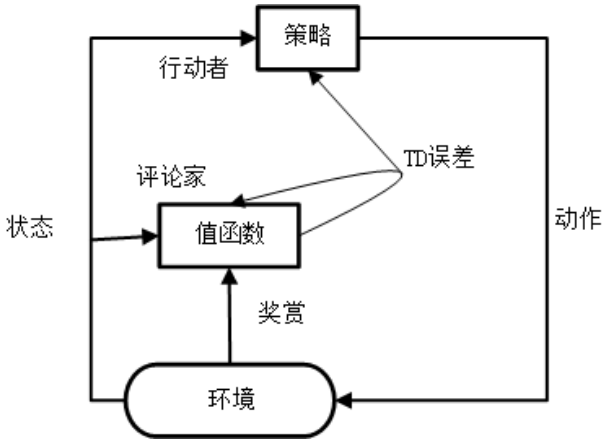


图 2-1 行动者评论家算法框架

如图2-1所示，在算法执行完一个动作之后，评论家评估新状态的值函数，并由此来判断行动者选择动作的好坏。在传统的AC算法中，一般是使用TD误差来作为衡量动作好坏的标准，并根据误差值调整策略的参数。如果TD误差为正，就更新策略参数使得该动作在以后的学习过程中被选择的概率变大。相反如果TD误差为负，则更新策略的参数以减少该动作被行动者选择的概率。

AC 框架也常被应用到深度强化学习方法中，以应对一些连续动作空间的复杂决策问题。此时可使用两个权重不同的深度神经网络分别表示策略和值函数。其中，权重为θ_h的策略网络用于选择行动，对应AC框架中的行动者；权重为θ_v的值函数网络用于评价策略网络选择动作的好坏，并提供梯度信息用于更新策略网络的权重，对应AC框架中的评论家。此时动作评价的标准与传统的AC方法有所不同，一般采用$R_t - V(x_t)$的形式，其中$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$，$V(x_t)$表示一个基线，目的是减小$R_t$项的方差。即如果$R_t - V(x_t)$项为正，就更新策略网络的权重使得对应动作在以后的学习过程中被选择的概率变大；否则更新策略网络的权重以减少该动作被选择的概率。策略网络的权重更新公式如下：

$$\theta_h \leftarrow \theta_h + \alpha\left(R_t - V(x_t)\right)\nabla_{\theta_h} \log h\left(u_t \middle| x_t\right) \tag{2.10}$$

值函数网络的权重更新公式如下：

$$\theta_v \leftarrow \theta_v - \alpha\nabla_{\theta_v}\left(R_t - V(x_t)\right)^2 \tag{2.11}$$

其中α代表学习率。

2.3 深度Q网络

深度Q网络算法属于深度强化学习领域中的开创性工作，有着极大的研究和应用价值。该算法结合使用卷积神经网络和Q学习算法，并通过几处改进一定程度上解决了用非线性的神经网络逼近器近似表示值函数时算法不收敛的问题。深度Q网络是首个在基于视觉感知的大规模状态空间任务中取得显著效果的深度强化学习方法。通过实验证明，训练完成的深度Q网络模型可在大多数Atari 2600视频游戏上取得与人类相媲美的表现，并在一些游戏上的得分超过了专业的游戏玩家。

2.3.1 训练算法

深度 Q 网络的训练算法通常被称为深度 Q 学习，该算法主要在传统的 Q 学习算法的基础上做了三处改进，分别是使用经验回放机制、固定目标 Q 值网络、缩小奖赏值范围。具体的改进为：

（1）使用经验回放机制。在每个时间步 t，将智能体与环境交互得到的转移样本 $e_t=(x_t,u_t,r_t,x_{t+1})$ 存储到回放记忆单元 D 中。训练时从 D 中随机抽取固定数量的一批转移样本，并使用随机梯度下降（Stochastic Gradient Descent, SGD）算法更新网络权重 θ。在训练深度神经网络时，通常要求各个转移样本之间是符合独立性假设的。因此上述随机采样的方式，去除了样本之间的关联性，从而提升了训练网络模型时的稳定性和性能。

（2）固定目标 Q 值网络。深度 Q 网络除了使用深度神经网络逼近状态动作值函数之外，还单独使用另一个深度神经网络来产生目标 Q 值。其中，$Q(x,u|\theta)$ 代表当前值网络的输出，用来评估当前状态动作对的值函数；$Q(x,u|\theta^-)$ 代表目标值网络的输出。在深度 Q 学习算法中，通常采用 $Y=r+\gamma\max_{u'}Q(x',u'|\theta^-)$ 项来近似表示值函数的优化目标，即目标 Q 值。当前值网络的权重 θ 保持实时更新，并且每经过 C 个时间步，将当前值网络的权重复制给目标值网络 $\theta^-\leftarrow\theta$。通过最小化值网络输出 Q 值和目标 Q 值之间的均方误差来更新网络权重。构造的误差函数为：

$$L(\theta)=\mathbb{E}_{x,u,r,x'}\left[\left(Y-Q(x,u|\theta_i)\right)^2\right] \tag{2.12}$$

上式对权重 θ 求偏导，得到以下梯度：

$$\nabla_{\theta}L(\theta)=\mathbb{E}_{x,u,r,x'}\left[\left(Y-Q(x,u|\theta)\right)\nabla_{\theta}Q(x,u|\theta)\right] \tag{2.13}$$

在算法中使用目标值网络后，一段时间内的目标 Q 值是保持恒定的，一定程度上降低了当前 Q 值和优化目标值之间的相关性，从而提高了算法的稳定性和性能。

（3）缩小误差项的范围。在不同的任务场景中，奖赏的量级可能是大不相同的。而太大的奖赏设置会导致梯度项过大，从而导致学习的不稳定。因此在深度 Q 学习算法中将误差项缩减到固定的小区间 $[-1,1]$ 内，以保证梯度项的大小处于合理的范围内。该技巧可以显著提高训练算法的稳定性。

深度 Q 学习的具体描述如算法 2-2 所示：

算法 2-2 深度 Q 学习算法

1：**初始化**：回放记忆单元 D 的容量为 N，Q 值函数的初始权重为 θ，目标 Q 值函数的初始权重为 $\theta^{-}=\theta$

2：**Repeat**（对每一个情节）：

3：　初始化状态序列为 $x_1=\{o_1\}$，并经过预处理后得到 $\phi_1=\phi(x_1)$

4：　**Repeat**（对于情节中的每一步）：

5：　以概率 ε 选择随机动作 u_t，否则选择动作 $u_t=\arg\max_u Q(\phi(x_t),u|\theta)$

6：　执行动作 u_t，并得到奖赏 r_t 和下一个观察 o_{t+1}

7：　设置 $x_{t+1}=x_t,u_t,o_{t+1}$，并经过预处理得到 $\phi_{t+1}=\phi(x_{t+1})$

8：　存储转移序列 $(\phi_t,u_t,r_t,\phi_{t+1})$ 到 D 中

9：　从 D 中随机采样固定数量的转移样本 $(\phi_j,u_j,r_j,\phi_{j+1})$

10：　如果达到终止状态，则优化目标设置为：$Y_j=r_j$

11：　否则设置为：$Y_j=r_j+\gamma\max_{u'}Q(\phi_{j+1},u'|\theta^{-})$

12：　对 $(Y_j-Q(\phi_j,u_j|\theta))^2$ 式关于权重 θ 进行梯度下降，用于更新权重 θ

13：　每隔 C 时间步重置目标值网络的权重 $\theta^{-}=\theta$

14：　**Until** 到达终止状态

15：**Until** 情节数到达上限次数 M

2.3.2 模型架构

深度 Q 网络模型架构的输入是距离当前时刻最近的四幅预处理后的图像。该输入信号经过 3 个卷积层和 2 个全连接层的非线性变换，变换成低维的、抽象的特征表达，并最终在输出层产生每个动作对应的 Q 值函数。具体地：

（1）将经过预处理之后的最近 4 幅大小为 84×84 的图像作为输入。然后对输入进行卷积操作，卷积核的大小为 $1\times8\times8$，步长为 4。并在第一隐藏层上得到 32 幅大小为 20×20 的特征图（Feature Maps）。最后使用矫正函数 $\max(0,x)$ 对第一隐藏层进行非线性变换。

（2）对第一隐藏层的输出进行卷积操作。卷积核的大小为 $32\times4\times4$，步长为 2。得到 64 幅大小为 9×9 的特征图作为第二隐藏层。最后通过矫正函数 $\max(0,x)$ 来激

活第二隐藏层。

（3）利用 64 个大小为64×3×3 的卷积核，以步长 1 对第二隐藏层的输出进行卷积操作，得到 64 幅大小为7×7 的特征图作为第三隐藏层。同理，对第三隐藏层的输出进行非线性变换，输出 64 幅大小为7×7 的特征图。最后利用两个全连接层在输出层输出动作空间中所有动作的 Q 值。具体的模型架构图 2-2 所示：

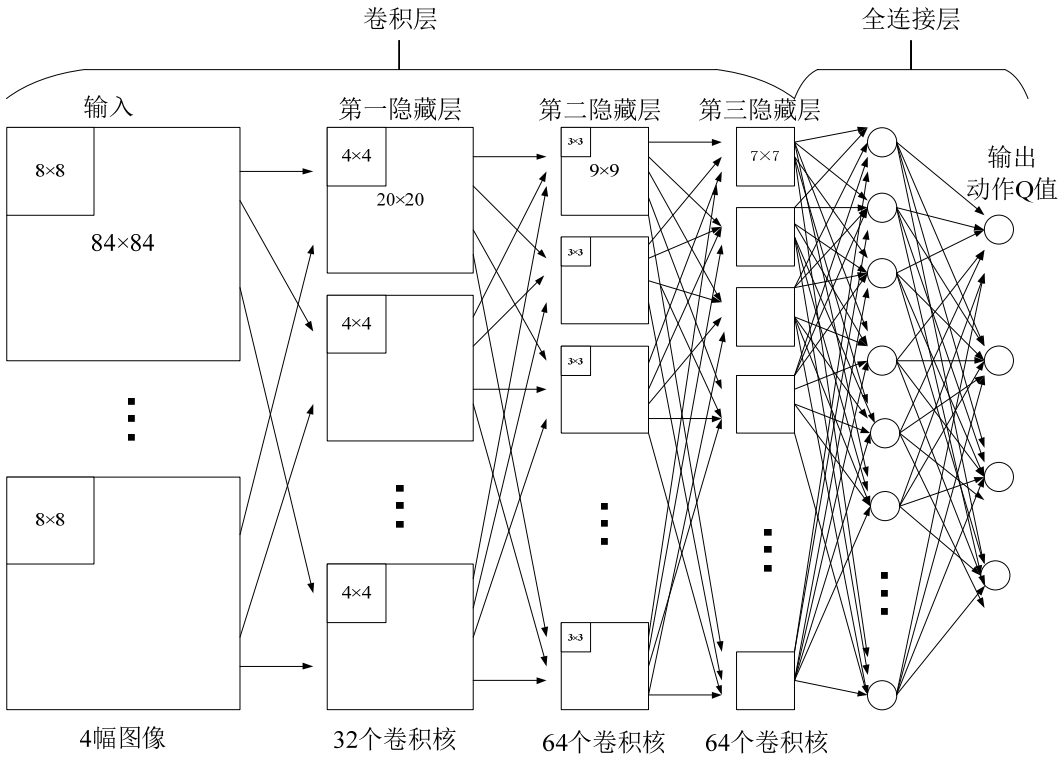


图 2-2 深度 Q 网络模型架构

2.4 本章小结

本章首先介绍马尔科夫决策过程和强化学习中的一些相关理论基础。然后重点介绍了一些经典的强化学习算法，包括蒙特卡罗方法、Q 学习和 AC 算法。然后详细阐述了深度 Q 网络的训练算法和模型架构，为后续章节做了铺垫。

第三章 基于优先级采样的深度 Q 学习算法

现代强化学习和深度学习的结合带来了众多领域的重大突破，尤其是那些需要同时感知高维度原始输入信号和做出策略选择的场景。其中，一项重大的突破是深度 Q 网络方法的提出，其基本思想是使用深度神经网络作为值函数的逼近器。深度 Q 网络擅长解决接近真实场景的复杂决策任务，比如 Atari 2600 游戏。深度 Q 网络的训练算法被称作深度 Q 学习。为了消除转移样本之间的时间相关性，深度 Q 学习使用一种被称为经验回放的采样机制。该机制在训练的每个时刻从回放记忆单元中随机抽取规定数目的转移样本。然而，该种回放机制不能区分出不同转移样本之间的重要性差异，并且可能会导致一些有价值的样本被覆盖和重复利用等问题。本章首先介绍一种基于优先级的经验回放机制，然后将该机制引入到深度 Q 学习算法中来替代原先的随机采样，并提出基于优先级采样的深度 Q 学习算法。最后通过实验表明，该种基于优先级采样的深度 Q 学习算法在一些经典的 Atari 2600 游戏上的平均得分和学习速率都有所提高。

3.1 基于优先级采样的经验回放机制

3.1.1 传统的经验回放机制

在线的强化学习智能体基于一系列高度相关的转移序列来逐步更新值函数。然而在大多数深度强化学习方法中，训练的对象通常是深度神经网络模型。此时在训练模型时对转移样本有两点基本要求：（1）转移样本之间是相互独立的；（2）转移样本在训练期间可被重复利用。因此，在深度 Q 学习算法中提出一种经验回放机制来满足上述两点要求[4]。在每个训练时间步，智能体向回放记忆单元中存放在线得到的转移序列并随机抽取数量固定的一批次转移样本用于更新模型的权重。然而随机抽取转移样本的经验回放机制存在着一些缺陷。具体地，随机抽样不能优先利用对智能体的学习有促进作用的转移序列，并且受限于有限的存储空间，一些未被及时利用的新样本可能会被覆盖掉。为了缓解上述问题，下一小节将提出一种基于优先级采样的经验回放机制来替代传统的随机采样方式。

3.1.2 优先级采样方法

通常在强化学习算法中，提高带有正奖赏或者 TD 误差值较大的转移序列利用率，可以提升智能体的学习速率。一方面，带有正奖赏的转移序列在学习初期是十分罕见的。不过这些样本对于智能体的学习却是有重要价值的。因此提高带有正奖赏转移序列的采样率，将一定程度上解决在某些复杂问题中存在的稀疏奖赏问题，从而提高智能体的性能。本章提出的优先级采样算法首先使用两个不同的回放记忆单元来区分出具有不同奖赏量级的转移序列。其中具有较高优先级的回放记忆单元 D_1 用于存放带有正奖赏的转移序列，具有较低优先级的回放记忆单元 D_2 用于存放带有负奖赏及零奖赏的转移序列。然后使用一种类似于分层抽样的方法从回放记忆单元中抽取固定数量的转移样本。具体的抽样方式是以概率 ρ 从回放记忆单元 D_1 中抽取样本，以概率 $1-\rho$ 从回放记忆单元 D_2 中抽取样本。通过上述区别奖赏值量级大小的分层采样，提高了有价值的正奖赏转移样本的利用率，从而促进了智能体的学习速度。

另一方面，为缓解一部分转移样本还未被利用就被回放记忆单元覆盖掉的问题，该优先级采样方法在转移序列中记录了每个样本在训练过程中被采样的次数，并用 v_t 表示该采样次数。此时转移样本的形式为 $e_t=(x_t,u_t,r_t,v_t,x_{t+1})$。通常那些被频繁回放的转移样本具有较小的 TD 误差值，这是由于每一次回放都使得该样本所对应的 Q 值函数更加逼近目标动作值函数。在经验回放机制中，可以优先考虑那些很久未被采样到的转移样本，因为这类样本对应的 TD 误差值较大。具体地，该抽样机制使用一种基于优先级的采样方式来确保每个转移样本都能被采样到，并且转移样本 j 的优先级被定义为：

$$p_j=\frac{1}{v_j+1} \tag{3.1}$$

由上式可知，转移样本 j 的优先级随着其采样次数的增加单调递减。

在基于离线样本的强化学习算法中，如果收集到的样本不能覆盖整体的样本空间，在某种程度上会导致学习的偏向性。而完全基于优先级的贪婪采样方式会使得转移样本缺乏多样性。具体地，基于优先级的贪婪采样方式倾向于收集那些具有较高优先级的转移样本，以避免扫描整个样本空间。不过该种抽样方式可能会使得一

些具有较低优先级的样本失去被采样到的机会，从而导致对应的 TD 误差项更新过程过于缓慢。上述问题在使用深度神经网络来表示值函数时体现的尤为明显。因此十分有必要提出一种可以同时保证样本多样性和充分利用样本优先级的经验回放机制。

3.1.3 随机化方法

上一小节提出的优先级采样会导致学习的偏向性。在本节中将介绍一种随机化的采样方法。该技巧可使得采样过程介于完全的优先级采样和随机采样之间，从而在优先利用有价值转移样本的前提下，保证回放样本的多样性。为了使得转移样本被采样的概率与对应的优先级成正比，并同时保证最低优先级对应的转移样本具有非零的采样概率，定义转移样本 j 的采样概率为：

$$P(j)=\frac{p_j^{\alpha}}{\sum_{i=1}^{i=size(D_1)} p_i^{\alpha}} \tag{3.2}$$

其中 p_j 表示转移样本 j 的优先级，参数 α 决定采样的随机化程度。当 $\alpha=0$ 时，该采样机制退化为普通的随机采样方式；当 $\alpha=1$ 时，该采样机制对应完全根据优先级的贪婪采样方式。

3.2 基于优先级采样的深度 Q 学习算法

本节基于上一节提出的随机化的采样方式，提出一种基于优先级采样的深度 Q 学习算法（Deep Q-Learning with Prioritized Sampling，PS-DQN）。该算法将一种高效的优先级采样机制和深度 Q 学习算法相结合，一定程度上缓解了传统深度 Q 学习中有价值样本利用率不高的问题，提高了智能体在面对一些基于视觉感知的决策问题时的性能和稳定性。

3.2.1 训练算法描述

在很多场景下，直接结合使用深度神经网络和 Q 学习算法会导致算法的不稳定。然而当智能体面对的是大规模状态空间下的决策任务时，深度神经网络逼近器却有着不可替代的作用。因为深度神经网络可以自动地学习到抽象、具体的低维特征表示。为了保证性能的稳定性，PS-DQN 算法有具体的两处创新点：

（1）使用一种基于优先级采样的经验回放机制来消除样本之间的相关性。该机制根据奖赏项的不同量级，将转移样本$(x_t,u_t,r_t,v_t,x_{t+1})$存储到不同的回放记忆单元中。在训练的每个时刻，智能体通过优先级采样从回放记忆单元中抽取固定大小的批次转移样本，并基于这些样本通过随机梯度下降法来更新网络模型的权重。

（2）为了进一步提升算法的稳定性，使用独立的目标值网络$Q(x,u\,|\,\theta^{-})$来产生目标 Q 值：$Y_i = r + \gamma \max_{u'} Q(x',u'\,|\,\theta_i^{-})$。与深度 Q 学习算法不同的是，新算法中使用一种“软”目标值的更新方式来替代阶段性直接拷贝当前值网络权重给目标网络权重的方式。具体地，目标值网络的权重通过缓慢追踪当前值网络的权重来更新：$\theta^{-} \leftarrow \tau\theta + (1-\tau)\theta^{-}$，其中，$\tau \le 1$。该种权重$\theta^{-}$的生成方式能够有效控制每次目标 Q 值更新的幅度，提升智能体在学习最优策略时的稳定性。

算法 3-1 给出了基于优先级采样的深度 Q 学习算法流程。

算法 3-1 基于优先级采样的深度 Q 学习算法

1：**初始化**：回放记忆单元D_1和D_2的容量为N，值网络的初始权重为θ，目标值网络的初始权重为$\theta^{-}=\theta$，$p_1=1$，minibatch 的大小设置为 32

2：**Repeat**（对每一个情节）：

3：　初始化状态序列为$x_1=\{o_1\}$，并经过预处理后得到$\phi_1=\phi(x_1)$

4：　**Repeat**（对于情节中的每一步）：

5：　以概率ε选择随机动作u_t，否则选择动作$u_t=\arg\max_u Q(\phi(x_t),u|\theta)$

6：　执行动作u_t，并得到奖赏r_t和下一个观察o_{t+1}

7：　设置$x_{t+1}=x_t,a_t,o_{t+1}$，并经过预处理得到$\phi_{t+1}=\phi(x_{t+1})$

8：　如果$r_t>0$，则将转移序列$(\phi_t,u_t,r_t,v_t,\phi_{t+1})$存储到$D_1$中

9：　否则将存储转移序列$(\phi_t,u_t,r_t,v_t,\phi_{t+1})$到$D_2$中

10：　**Repeat**（对于每个 minibatch）：

11：　如果$random()<\rho$：则

12：　以概率分布$P(j)=(p_j)^{\alpha}/\sum_i(p_i)^{\alpha}$从$D_1$中抽取$(\phi_j,u_j,r_j,v_j,\phi_{j+1})$

13：　否则：

14：　以概率分布$P(j)=(p_j)^{\alpha}/\sum_i(p_i)^{\alpha}$从$D_2$中抽取$(\phi_j,u_j,r_j,v_j,\phi_{j+1})$

15：　更新访问次数：$v_j=v_j+1$

16： 更新转移样本的优先级： $p_j = 1 / (v_j + 1)$

17： **Until** 该批次的迭代轮结束

18： 如果达到终止状态，则优化目标设置为： $\mathrm{Y}_j = r_j$

19： 否则设置为： $Y_j = r_j + \gamma \max_{u'} Q\left(\phi_{j+1}, u' \middle| \theta^-\right)$

20： 对损失函数 $L(\theta) = \left(\mathrm{Y}_j - Q\left(\phi_j, u_j \middle| \theta\right)\right)^2$ 关于权重 θ 进行梯度下降步

21： 更新目标值网络的权重 $\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-$

22： **Until** 到达终止状态

23：**Until** 情节数到达上限次数 M

3.2.2 模型架构描述

PS-DQN 算法是基于深度卷积网络模型来训练的，并且该网络模型拥有独立的输出单元以用于预测离散动作的 Q 值函数。该模型的主要优点在于它可以基于给定的抽象状态表示，计算出动作空间中所有离散动作的 Q 值函数，并且这些抽象的状态特征表示可通过一次简单的前向传播得到。PS-DQN 算法完整的模型架构如图 3-1 所示。与 DQN 训练模型不同的是，新模型在全连接层后面引入了 dropout 操作，目的是防止出现过拟合问题[78]。具体地，模型的输入是经过预处理之后得到的 84×84×4 大小的图像。第一卷积层通过 32 个大小为8×8 、步长为 4 的卷积核对输入图像进行卷积操作，并通过非线性修正单元（Rectifier Nonlinearity，RELU）来进行非线性变换。第二卷积层通过 64 个大小为4×4 、步长为 2 的卷积核对第一卷积层的输出进行卷积操作，同样连接上非线性修正单元。第三卷积层通过 64 个大小为3×3 、步长为 1 的卷积核对第二卷积层的输出进行卷积操作，然后用 RELU 进行非线性变换。卷积操作过后，在第三卷积层的后面是由 512 个神经元构成的全连接层。值得注意的是，在该层上实施了 dropout 操作。最后网络模型的输出是一个全连接层，并且每个独立的单元都对应一个动作的 Q 值函数。在 Atari 2600 游戏中，合理动作的编号是 4 到 18。

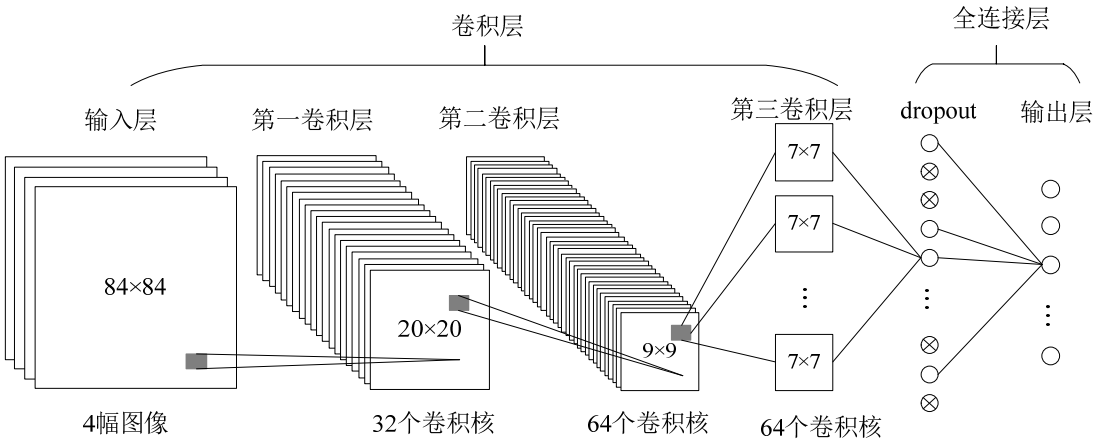


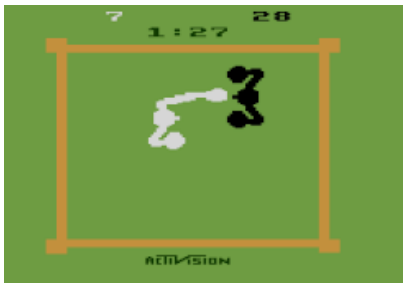
图 3-1 基于优先级采样的深度 Q 学习算法模型框架

3.3 仿真实验

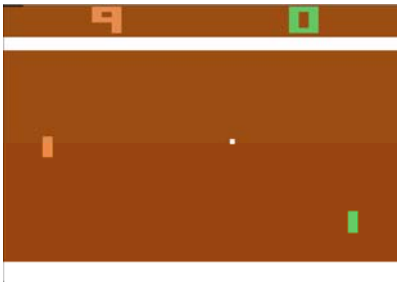
3.3.1 实验描述

本节通过 4 个经典的 Atari 2600 游戏任务来验证 PS-DQN 算法的性能。这 4 个游戏分别是打砖块（Breakout）、拳击（Boxing）、乒乓（Pong）和太空入侵者（Space_invaders）。图 3-2 展示了这 4 种视频游戏的画面。其中在 Breakout 游戏中，玩家通过控制一块能够左右移动的挡板以反弹行进中的球。当球碰到砖块时，上方被击中的砖块会消失，此时根据击中砖块的颜色，玩家得到相应的分数，并且球会反弹。当玩家未能用挡板接住反弹的球，则输掉该个回合。玩家的目的是清除所有的砖块；Boxing 游戏要求玩家与对方距离足够近时，出拳击打对手。远距离击打得+1 分，近身击打得+2 分。玩家的目的是多击打对方以得到更多的分数；在 Pong 游戏中，玩家控制球拍的上下移动以反弹乒乓球。当玩家未能反弹乒乓球时，对方得+1 分。玩家目的是尽量反弹乒乓球并夺取高分；在 Space_invaders 中，玩家使用由地堡保护的“大炮”来击打太空中的侵略者，目的是击灭侵略者的阵形，直到所有的入侵者都被消灭。上述 4 种游戏实验都是基于 Arcade 学习环境平台（Arcade Learning Environment，ALE）进行的。该平台提供了一系列具有挑战性的 Atari 2600 游戏任务编程接口，包括射击、冒险、策略、体育等类型[79]。在 ALE 平台中，智能体只能通过感知原始的高维输入图像来学习玩游戏的技巧。另外值得注意的是，本章所有实验中用到的网络架构、训练算法和超参数都保持一致。这说明 PS-DQN 算法适用于各类基于视觉感知的决策任务，具有较高的泛化性能。

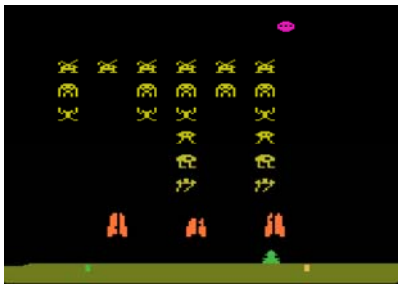
(a) Breakout



(b) Boxing



(c) Pong



(d) Space_invaders

图 3-2 四种经典的 Atari 2600 游戏截屏

3.3.2 实验设置

由于在不同游戏环境中，奖赏的量级是有区别的。因此实验中将所有的正奖赏裁剪到+1，负奖赏裁剪到-1。此外将梯度项裁剪到$[-5,5]$区间内。裁剪奖赏值和梯度项可以控制误差导数的范围，使得算法可以在不同游戏任务中设置相同的学习率，从而提高算法训练时的稳定性。另外行为策略$\varepsilon-greedy$中的参数ε设置为前 100 万时间步线性递减的形式。具体地，ε从 1.0 线性递减到 0.1，并在 100 万时间步后保持不变。“软”因子τ设置为 0.05。参数ρ在前 100 万时间步中从 0.5 衰减到 0.25，并在之后保持不变。这是由于随着模型的训练，智能体将会得到越来越多的正奖赏。在 Breakout 游戏中，设置不同大小的随机化因子α，并通过粗略的性能比较得出：当α设置为 0.6 时，算法的性能最好。所有实验中都选用 RMSProp 优化方法来训练模型，并且批量大小设置为 32。此外，算法使用一种简单的跳帧技巧。具体地，智能体每隔 4 幅视频帧作出新的动作决策，而之前的时刻重复采取上一个动作。这是由于网络模型通过前向传播处理一幅图像所耗费的计算力远远少于智能体选择新动作所耗费的计算力。实验中为了保证初始状态的多样性，额外规定智能体在作出动作决策之前先重复实施 null 操作或者不执行任何操作以跳过游戏最开始

时的若干视频帧。回放记忆单元的容量上限为 100 万个转移样本。最后在每个实验中，网络模型都通过一个图形处理器（Graphics Processing Unit，GPU）进行 5000 万时间步的训练。

总结而言，实验中仅使用输入图像、游戏得分和一套固定的超参数这些先验知识，训练出了具备强大感知和决策能力的智能体。这些训练完成的模型在一系列基于视觉感知的决策任务上取得了与人类相媲美的性能。

3.3.3 实验结果及分析

本章通过 Breakout、Boxing、Pong 和 Space_invaders 这 4 个游戏任务来验证 PS-DQN 算法的性能。由于新算法是基于传统 DQN 训练算法的改进，因此实验中主要对比 PS-DQN 和 DQN 算法的性能差异。在训练过程中，每经过 250000 时间步模型进行一次策略性能的评估。具体的评估标准是平均每情节智能体获得的奖赏之和。评估周期设置为 125000 时间步，并且在评估期间智能体一直执行 ε-*greedy* 策略，其中 ε=0.05 。

在情节式的强化学习问题中，通常将评估策略的标准设置为智能体一个情节内获得的奖赏之和。因此实验中第一个评估标准设置为：在 50 个训练周期上，DQN 和 PS-DQN 模型所获得的每情节最高平均奖赏值。图 3-3 显示了 PS-DQN 和 DQN 算法在上述 4 个游戏中的训练进程。

基于图 3-3 分析可知，在 DQN 训练算法中引入优先级采样的经验回放机制，提升了智能体在大都数游戏任务上的性能。这种性能的提升主要体现于：在学习的初期，智能体所获得的平均奖赏更高。这是由于在 PS-DQN 算法中增加了具有正奖赏的转移样本利用率，而这种样本具有丰富的学习价值和信息性。另外分析图 3-3，发现除了 Space_invaders 游戏之外，PS-DQN 算法的训练性能比起 DQN 显得更加稳定。这是由于 PS-DQN 算法使用优先级采样使得样本空间中的每个转移序列都以一定的概率被采样，而传统的 DQN 算法则容易偏向于采样那些已经被回放上百次的过时样本。从图 3-3 的训练曲线可知，每情节平均奖赏之和的评估标准存在着噪音。这是由于网络模型权重的细微改变都会导致当前策略访问到的状态空间概率分布发生很大的波动。因此实验中使用一种更加稳定的评估标准来衡量 PS-DQN 和 DQN 算法的性能差异。该评估标准被称为平均最大预测动作值函数。

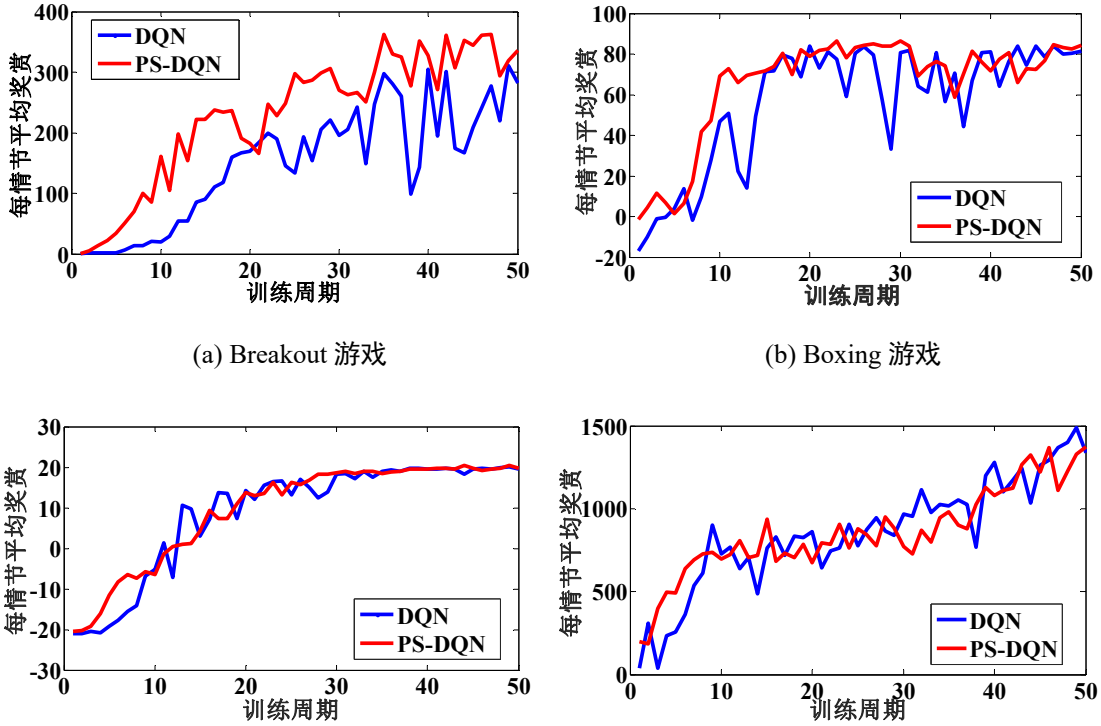
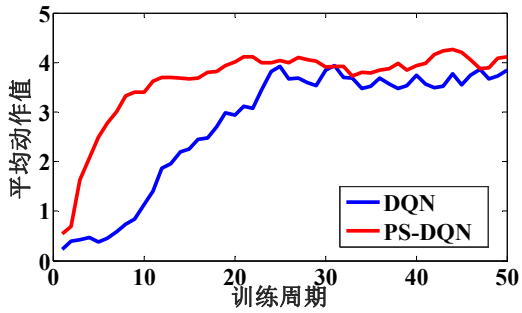
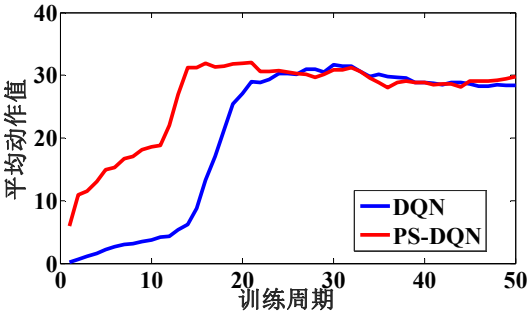


图 3-3 DQN 和 PS-DQN 在 50 次训练周期内每情节平均奖赏对比图

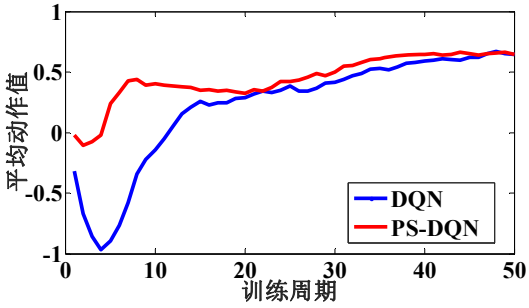
图 3-4 显示了 PS-DQN 和 DQN 算法在 50 个训练周期内平均最大预测动作值函数的对比图。分析图中的曲线，发现 PS-DQN 的收敛速度要快于 DQN。另外由于在 Breakout 和 Boxing 游戏环境中不存在负的奖赏，并且在模型训练初期智能体从环境中得到的正奖赏反馈是稀疏的。因此在使用优先级采样提高了正奖赏转移样本的利用率后，PS-DQN 算法在 Breakout 和 Boxing 游戏上的 Q 值曲线在收敛之前一直保持着平稳上升的趋势。如图 3-4(c)所示，由于在智能体训练玩游戏 Pong 的初期存在着大量的负奖赏，代表 DQN 训练进程的 Q 值曲线存在着一个低峰值。这种 Q 值的大幅度波动是不利于智能体学习的。PS-DQN 算法则通过提高正奖赏转移样本的利用率，一定程度上避免了出现平均 Q 值曲线低峰值的问题，从而提升了算法的稳定性。然而在 Space_invaders 游戏的初期，环境中并不缺少正奖赏信号。这时候 PS-DQN 算法中采用的优先级采样不可避免地导致正奖赏转移样本的过度利用。因此在图 3-4(d)中，PS-DQN 在学习的初期出现了 Q 值过度估计的问题，这一定程度上阻碍了算法性能的提升。



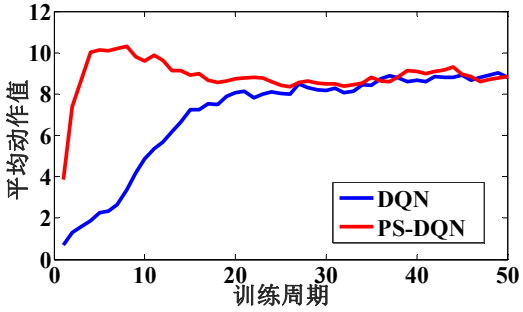
(a) Breakout 游戏



(b) Boxing 游戏



(c) Pong 游戏



(d) Space_invaders 游戏

图 3-4 DQN 和 PS-DQN 在 50 次训练周期内每情节平均最大预测动作 Q 值对比图

尽管如此，PS-DQN 算法也有一些不足之处。从图 3-3(d)中可以看出，PS-DQN 比起 DQN 算法在性能上几乎没有提升。这体现在某些训练周期内，PS-DQN 算法取得的每情节平均奖赏更低。另外 PS-DQN 由于过度使用正奖赏转移样本而导致训练进程不稳定。总结而言，PS-DQN 算法比较适用于一类基于视觉感知的决策任务，并且这类任务的共同点是在训练初期可从环境中得到大量的零奖赏和少量的正奖赏信号，比如 Breakout 和 Boxing 游戏。

3.4 本章小结

传统的深度 Q 学习训练算法不能区分不同转移样本之间的重要性差异，并且由于有限的存储空间而导致一些有价值的样本被覆盖和重复利用。针对上述问题，本章提出一个新颖的深度强化学习训练算法，即基于优先级采样的深度 Q 学习算法 PS-DQN。该算法将一种基于优先级采样的经验回放机制引入到深度 Q 学习中，一方面提高了有价值样本的利用率，另一方面使得样本空间中的每个转移样本都以一定的概率被访问到。

本章将 PS-DQN 算法应用于 4 个经典的 Atari 2600 游戏任务，并比较 DQN 和 PS-DQN 的性能差异。从实验结果可以看出使用优先级采样的 PS-DQN 在收敛速度和稳定性上都超过了传统的 DQN 算法，并且新算法在大都数游戏任务上取得了更高的平均得分。这些实验结果充分说明 PS-DQN 可适用于一些基于视觉感知的大规模决策任务。然而与 DQN 算法类似的是，PS-DQN 在应用场景上也有一些缺陷：（1）PS-DQN 算法不能解决一类难度较大的战略性挑战任务；（2）PS-DQN 只适用于离散动作空间下大规模的决策问题，无法将其应用场景扩展到更加普遍的连续动作空间。

第四章 基于视觉注意力机制的深度循环 Q 网络模型

深度强化学习方法在各种需要同时感知高维度原始输入数据和输出决策控制的任务中取得了实质性的进展。其中一种被称为深度 Q 网络的模型在解决一类趋于真实环境的复杂问题时表现出和人类玩家相媲美的水平。然而当环境中存在有延迟的奖赏而导致智能体需要长时间步的规划才能优化策略的情形中，深度 Q 网络的性能急剧下降。这说明传统的深度 Q 网络模型并不擅长解决战略性决策任务。针对上述问题，本章使用带视觉注意力机制的循环神经网络改进了传统的深度 Q 网络模型，提出一种较为完善的深度强化学习模型架构。新模型主要有两处创新点：一是使用双层门限循环单元构成的循环神经网络模块来记忆较长时间步的历史信息，从而使得智能体能够及时使用有延迟的反馈奖赏来正确地指导下一步的动作选择；二是通过视觉注意力机制自适应地将智能体的注意力集中于面积较小但更具价值的图像区域，以促进智能体更加高效地学习解决战略性挑战任务的最优策略。最后本章通过选取一些经典的 Atari 2600 战略性游戏作为实验对象来评估新模型的性能。实验结果表明，新模型在一些战略性决策任务上具有更好的性能表现和稳定性。

4.1 门限循环单元

循环神经网络（Recurrent Neural Network，RNN）是一种对时间维度上的序列数据进行显示建模的深度学习模型。RNN 通过添加跨越时间步的自连接隐藏层，使得网络中的处理单元之间既有前馈连接又有内部的反馈连接。该种新颖的连接方式使得模型具备了记忆动态时序信息的能力。即 RNN 中当前层的反馈不仅输出到下一隐藏层，还进入到下一时间步的当前隐藏层中。传统的 RNN 模型首先将输入序列$(\mathbf{x}_1,\ldots,\mathbf{x}_t)$映射为隐藏层的状态表示$(\mathbf{h}_1,\ldots,\mathbf{h}_t)$，然后再根据下面的转换得到输出$(\mathbf{z}_1,\ldots,\mathbf{z}_t)$：

$$\mathbf{h}_t = f\left(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1}\right) \tag{4.1}$$

$$\mathbf{z}_t = g\left(\mathbf{W}_{zh}\mathbf{h}_t\right) \tag{4.2}$$

其中 f 和 g 是非线性激活函数，常见的有 Sigmoid 函数和 tanh 函数，$\mathbf{W}_{ij}$ 代表连接

不同层神经单元之间的权重。

然而当开始记忆时序信息的时刻与当前时刻的间距过大时，传统的 RNN 会产生关于时间轴上历史信息的“梯度弥散”问题。即 t 时刻产生的误差梯度信号在沿着时间轴向前传播一段时间之后趋近于零，从而阻碍网络权重的进一步更新。

为了解决“梯度弥散”问题，Hochreiter 等人提出长短期记忆单元模型（Long Short-Term Memory，LSTM）[80]。LSTM 在模型内部增加多个门（Gate）结构，并通过控制它们的开关以实现对多时间步长内时序信息的记忆。通过对结构的进一步简化与改进，LSTM 衍生出不少性能优异的变体。本文提出的新模型架构中 RNN 部分运用了其中一个较流行的变体，被称为门限循环单元（GRU），其结构如图 4-1 所示。

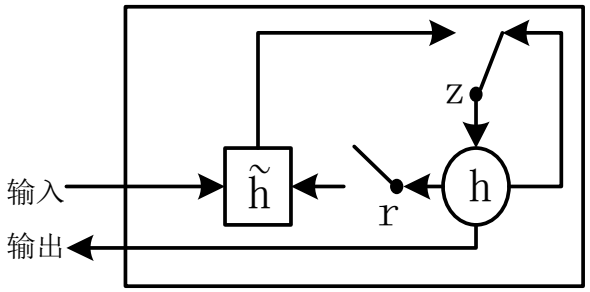


图 4-1 门限循环单元结构图

GRU 相比较于 LSTM 有一个显著的优点：GRU 可训练的权重数目更少，所以不容易出现过拟合的问题。GRU 通过不同功能门结构的组合开关，不断传播有价值的数据流，并过滤掉冗余数据，从而使得模型输出更加抽象、紧凑的多时间步长内的时序状态信息。

GRU 在 t 时刻的隐藏层激活值 $\mathbf{h}_t$ 是上一时刻激活值 $\mathbf{h}_{t-1}$ 和候选激活值 $\tilde{\mathbf{h}}_t$ 的线性组合：

$$\mathbf{h}_t = (1-\mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t \tag{4.3}$$

其中，$\mathbf{z}_t$ 表示更新门的输出。更新门采用 Sigmoid 函数 $\sigma(.)$ 来激活输入信息 $\mathbf{x}_t$ 与上一时刻的激活值 $\mathbf{h}_{t-1}$ 的线性组合值。该输出决定模型更新历史信息的程度：

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z\right) \tag{4.4}$$

候选激活值 $\tilde{\mathbf{h}}_t$ 的计算方式类似于传统的 RNN 单元：

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\left(\mathbf{r}_t * \mathbf{h}_{t-1}\right) + \mathbf{b}_h\right) \tag{4.5}$$

其中$\mathbf{r}_t$表示重置门。当$\mathbf{r}_t$接近于$\mathbf{0}$时，GRU 选择“遗忘”之前的激活值$\mathbf{h}_{t-1}$，并用当前的输入重置状态。反之当$\mathbf{r}_t$接近于$\mathbf{1}$时，表示模型选择记忆之前全部的激活信息。重置门$\mathbf{r}_t$的计算方式如下：

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r\right) \tag{4.6}$$

其中，符号*表示矩阵的内积操作符，$\mathbf{W}_{ij}$表示不同结构之间的连接权值矩阵。

4.2 视觉注意力机制

受到人类视觉机理的启发，近年来基于注意力机制（Attention Mechanism，AM）的循环神经网络被广泛应用于机器语言翻译、图像识别和主题生成等领域。Bahdanau 等人基于传统的编码器-解码器（Encoder-Decoder）框架，利用 AM 使得模型可以自动搜索与目标输出相关联的源输入语句，并在英语到法语的翻译任务上取得了很好的效果[81]。Mnih 等人提出一种基于视觉注意力机制的 RNN 模型。该模型可通过自我强化学习的 AM 得到重点关注区域或位置点的相关特征，并集中处理注意力关注的那部分像素集合，从而提升了图像的识别率[82]。Xu 等人首次将 AM 运用到视频图像数据的处理上，提出视觉注意力机制（Visual Attention Mechanism, VAM）。在每个时刻，通过 VAM 将注意力集中于对识别主题有促进作用的图像区域，提高了模型识别图片主题的正确率[83]。总结而言，通过各类任务的目标导向作用，VAM 以“高分辨率”的形式将智能体的注意力集中于图像中具有丰富信息的特定区域或像素位置，并通过训练不断调整其聚焦的区域。运用 VAM 的最大优势在于可以减小任务的复杂度，并能够以较少的训练数据、较快的训练速度取得更好的性能。本章提出的新模型架构中加入了 VAM 模块，使得智能体在每个时刻能够自适应地将注意力集中于对提升累计奖赏有促进作用的图像区域，从而大大减少了网络模型可训练的参数，提升了模型的训练速度。另外加入 VAM 模块可使得智能体在较短时间内完成对 RNN 记忆的多时间步内关键时序信息的感知，加速了智能体在面对战略性任务时学习最优策略的进程。

4.3 基于视觉注意力机制的深度循环 Q 网络模型

传统的 DQN 及一些基于 DQN 改进的深度强化学习模型在面对较为复杂的战略性挑战任务时，其表现与人类玩家的水平相差甚远。这是由于传统模型的输入状态是由离当前时刻最近的 4 幅连续视频帧（DQN）或单层 LSTM 单元记忆的连续历史信息（LSTM-DQN）构成，智能体能够感知到的历史信息相对有限。然而在战略性任务中，一个动作带来的反馈信号可能在数十步甚至上百步之后才能在值函数中体现出来，此时需要智能体跨越较长的时间步来规划策略。因此对于此类战略性挑战任务，通过有限时间步的历史画面来判断当前形势并做出决策，会出现部分有价值信息不可感知的问题。比如在国际象棋游戏中，仅根据过去几步的走子来判断当前局势并判断出下一步走棋动作是远远不够的。为了提高智能体在战略性挑战任务上的性能，可以增加输入中连续视频帧的数目以缓和部分状态不可观测的问题，但这会增加状态空间的维度，带来严重的计算负担。本节提出一种基于视觉注意力机制的深度循环 Q 网络模型（Deep Recurrent Q-Network with Visual Attention Mechanism，VAM-DRQN）。新的模型架构主要在原有的 DQN 基础上做了两方面的改进：（1）引入由双层 GRU 构成的 RNN 来记忆时间轴上的序列状态信息，一定程度上缓解了智能体在解决战略性任务时存在的部分有价值状态不可观测的问题；（2）通过在模型中引入 VAM，智能体能够在训练过程中自适应地将注意力集中于当前画面中对学习更具价值的部分区域，以促进智能体更加直观、快速地做出正确的决策。实验部分将 VAM-DRQN 模型应用到 Atari 2600 中的一些战略性游戏上。实验结果表明，VAM-DRQN 不仅能够提升智能体应对战略性挑战任务的能力，并且还能在多次实验中保持稳定的性能表现。

4.3.1 模型架构图

如图 4-2 所示，VAM-DRQN 模型主要由卷积神经网络、视觉注意力机制和循环神经网络三个模块组成。新模型架构是对高维度原始输入数据编码之后再解码成低维抽象特征的一个过程。因此下文将会基于编码器-解码器框架以 Atari 2600 游戏为例分析 VAM-DRQN 中各个模块的作用及处理输入数据时各模块之间的关联性。

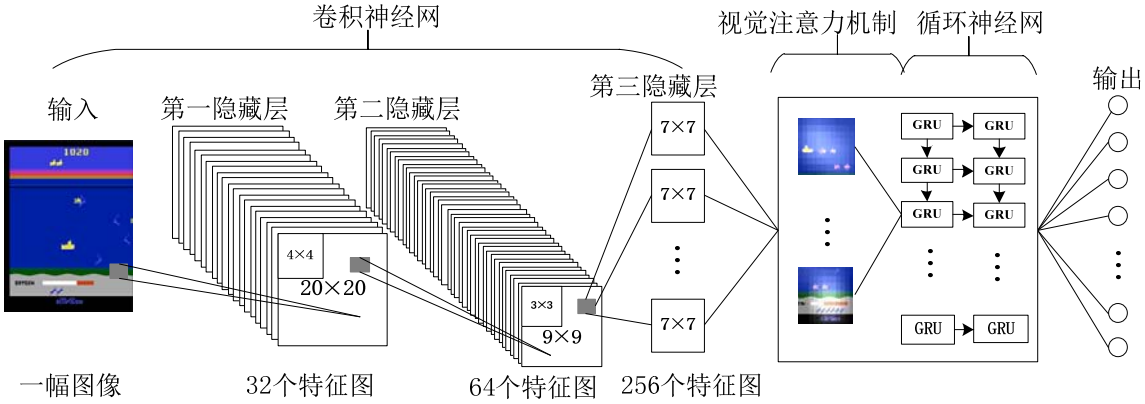


图 4-2 基于视觉注意力机制的深度循环 Q 网络模型架构

4.3.2 预处理

处理图像数据时，一般要通过预处理操作来消除图像中的无关信息以增强有价值特征的可检测性，从而最大限度地减小数据的复杂度。由于 Atari 2600 游戏画面的像素大小为210×160，直接处理原始图像所需的计算和存储代价过大。此时需要通过预处理操作对原始图像进行处理。首先将原始的三原色（RGB）图像转换成灰度图；然后通过降采样方法生成规模为110×84 的缩略图；最后去除边界一些无价值的像素点，使图像的尺寸进一步缩减为84×84 大小。此规模的图像能完整捕捉到当前的游戏画面。因此上述预处理过程并不会导致有价值特征的流失。该预处理操作仅需要对图像进行简单的灰度转换、裁剪和降采样，所耗费的计算资源远少于直接用 CNN 处理原始图像的情形。

传统的 DQN 是将离当前时刻最近的 4 幅原始游戏画面经过预处理之后输入到网络模型中，因此输入状态的规模为4×84×84。本节提出的 VAM-DRQN 模型架构，引入 RNN 来记忆游戏过程中多时间步的历史状态信息，因此只需要将该预处理操作运用于当前的一幅游戏画面中。此时输入状态的规模仅仅是1×84×84，状态空间的大小缩小至原来 DQN 的四分之一，从而减小了网络训练时所需的计算量，加快了智能体的学习速度。

4.3.3 编码器：卷积神经网络

VAM-DRQN 以 CNN 作为编码器，将高维的输入图像编码成低维、抽象的特征表达。从图 4-2 可直观地看出，CNN 模块从输入到输出一共进行了三次卷积和一次映射操作。具体步骤为：

（1）输入经过预处理之后大小为1×84×84 的当前游戏画面。通过 32 个规模为 1×8×8 的卷积核，以步长为 4 对输入进行卷积操作，得到 32 幅大小为20×20 的特征图（Feature Maps）作为第一隐藏层。然后用矫正函数 $\max(0,x)$ 对第一隐藏层进行非线性变换。

（2）通过 64 个规模为32×4×4 的卷积核，以步长为 2 对第一隐藏层的输出进行卷积操作，得到 64 幅大小为9×9 的特征图作为第二隐藏层，并通过非线性变换来激活第二隐藏层。

（3）利用 256 个规模为64×3×3 的卷积核，以步长为 1 对第二隐藏层的输出进行卷积操作，得到 256 幅大小为7×7 的特征图作为第三隐藏层。同理也对第三隐藏层进行非线性变换，输出 m 幅大小为 $n\times n$ 的特征图。其中， $m=256, n=7$ 。

在 DQN 中，将逐层卷积得到的特征图直接通过全连接层输入到包含 512 个神经单元的下一隐藏层。而 VAM-DRQN 则将第三隐藏层输出的 m 幅特征图映射为一个特征向量集合。该集合的每个向量元素代表各个特征图中不同位置上像素点的组合， t 时刻该向量集合为：

$$\mathbf{a}_t=\left\{\mathbf{a}_t^1,\ldots,\mathbf{a}_t^N\right\} \tag{4.7}$$

其中， $\mathbf{a}_t^i\in\Re^m, N=n\times n$ 。综上所述，经过三次卷积操作和一层映射变换，模型将原始的输入状态编码成不同特征图中对应位置像素点的集合，从而有利于解码器中的视觉注意力机制区分各个像素点的视觉重要性。

4.3.4 解码器：基于视觉注意力机制的循环神经网络

在解码之前，需要进一步处理编码器输出的向量集合 $\mathbf{a}_t$ ，以获得用于关联解码器和编码器的上下文向量 $\mathbf{C}_t$ 。传统形式的 $\mathbf{C}_t$ 表示 t 时刻输入图像中抽象特征的一种动态表示。然而在某些复杂的战略性决策任务中，需要智能体在短时间内完成对输入状态中关键特征的感知并依据此特征做出动作的选择。此时如果将智能体的注意力集中于整幅输入图像，会延缓网络模型的解码速度，使得智能体在短时间内无法及时感知到对正确决策有促进作用的部分有价值信息。为了缓解上述问题，本节创新性地将视觉注意力机制引入到模型架构中，以编码器输出的向量集合 $\mathbf{a}_t$ 为输入，通过 VAM 计算新的上下文向量 $\mathbf{C}_t$ ，使得智能体在每个时刻可以自适应地将注意力

集中于当前画面中面积较小但具有丰富信息的图像区域，从而加快模型解码的速度。下面具体介绍 VAM 在 VAM-DRQN 模型架构中工作流程：

（1）通过 VAM 模块中前两个全连接层计算向量集合 $\mathbf{a}_t$ 中各个像素点的视觉重要性：

$$vam\left(\mathbf{a}_t^i, \mathbf{h}_{t-1}\right) = Linear\left(Tanh\left(Linear\left(\mathbf{a}_t^i\right) + \mathbf{W}\mathbf{h}_{t-1}\right)\right) \tag{4.8}$$

其中 $Linear(x) = \mathbf{A}x + \mathbf{b}$ 是权值系数为 $\mathbf{A}$、偏移为 $\mathbf{b}$ 的线性仿射变换，$\mathbf{W}$ 为系数矩阵，$\mathbf{h}_{t-1}$ 为 RNN 模块隐藏层的输出。

（2）使用 Softmax 回归操作对第一步求得的结果进行归一化，得到每个像素点的相对视觉重要性：

$$\boldsymbol{\alpha}_t^i = \frac{\exp\left(vam\left(\mathbf{a}_t^i, \mathbf{h}_{t-1}\right)\right)}{\sum_{k=1}^{N} \exp\left(vam\left(\mathbf{a}_t^k, \mathbf{h}_{t-1}\right)\right)} \tag{4.9}$$

（3）根据每个像素点的相对视觉重要性计算出最终输入到 RNN 模块的上下文特征向量 $\mathbf{C}_t$：

$$\mathbf{C}_t = \sum_{i=1}^{N} \boldsymbol{\alpha}_t^i \mathbf{a}_t^i \tag{4.10}$$

由上述过程可知，VAM 模块得到的上下文特征向量 $\mathbf{C}_t$ 代表 $\mathbf{a_t}$ 中所有像素点关于相对视觉重要性的一个线性加权。根据 Donahue 等人关于 RNN 结构布局的研究，当使用双层记忆单元组件构成的 RNN 作为模型中的解码器来解码上下文特征时，其记忆时序状态信息的效果比用单层或四层组件的网络更明显[84]。因此在 VAM-DRQN 模型架构中，将双层 GRU 构成的 RNN 模块引入到深度 Q 网络中，用来对上下文特征向量 $\mathbf{C}_t$ 进行解码。通过上述序列化处理方法使得智能体能够感知多个时间步上的关键历史状态信息。从图 4-2 可以看出，RNN 的每一层由 256 个的 GRU 组件构成。

由于 RNN 是通过在网络中循环传递状态的方式处理时序数据，所以解码模块的输入应由 VAM 处理得到的上下文特征向量 $\mathbf{C}_t$ 和上一时刻的隐藏层输出值 $\mathbf{h}_{t-1}$ 两部分组成。RNN 模块处理数据的过程为：

（1）组合上下文特征向量和上一时刻的 RNN 隐藏层输出值 $\mathbf{I}_t=\{\mathbf{C}_t,\mathbf{h}_{t-1}\}$，并将其作为当前时刻 RNN 模块第一层的输入；

（2）将第一层的输出 $\mathbf{h}_t=\{\mathbf{h}_t^1,\ldots\mathbf{h}_t^{256}\}$ 作为第二层的输入。其中 $\mathbf{h}_t^i$ 代表 t 时刻 RNN 模块中第一层的第 i 个 GRU 组件的隐藏层输出值；

（3）最后 RNN 模块输出当前时刻 t 更新后的隐藏层输出值 $\mathbf{h}_t$。

该输出值有三方面的用途：首先作为 RNN 模块在下一时间步 $t+1$ 的输入激活值；其次作为 VAM 模块在下一时间步 $t+1$ 产生上下文向量 $\mathbf{C}_{t+1}$ 的输入；最后可以用来近似表示网络在当前输入状态下，智能体采取各种可能动作 u_t 的 Q 值。

综上所述，VAM-DRQN 处理图像数据的流程如下：首先用 CNN 编码当前的游戏画面 x_t，并使用逐层的卷积和非线性变换将输入转化成 m 幅大小为 $n\times n$ 的特征图。然后将 m 幅特征图映射到一个向量集合中 $\mathbf{a}_t=\{\mathbf{a}_t^1,\ldots\mathbf{a}_t^N\}$，其中 $\mathbf{a}_t^i\in\Re^m, N=n\times n$，$\mathbf{a}_t$ 中的每个元素分别对应卷积不同图像区域后得到的抽象特征。接下来通过基于 VAM 的 RNN 进行解码，将向量集合 $\mathbf{a}_t$ 输入到 VAM 模块中，经过一系列操作得到上下文特征向量 $\mathbf{C}_t\in\Re^m$。最后将 $\mathbf{C}_t$ 和上一时刻 RNN 模块隐藏层输出值 $\mathbf{h}_{t-1}$ 一起作为双层 GRU 模块的输入，通过处理单元内部的自反馈及前反馈产生新的隐藏层输出值 $\mathbf{h}_t$。该输出值不仅要作为下一时间步调整视觉注意力的依据输入到 VAM 中，还要作为评估当前状态 $\phi(x_t)$ 下所有可采取动作的 Q 值的依据输入到模型的输出层，其中 $\phi(.)$ 表示对原始图像的预处理操作。

4.3.5 模型架构的训练过程

VAM-DRQN 是一个各模块相互连接的端对端模型，模型中的三大部分 CNN、VAM 和 RNN 都是可微的。即新架构中每个模块可训练的参数都存在关于自身的梯度。本节使用自适应学习率的随机梯度下降算法来对模型参数进行端对端的更新。

在训练期间，使用经验回放机制从样本池中随机采样 minibatch 数量的转移样本 $\left(\phi(x_j),u_j,r_j,\phi(x_{j+1})\right)$ 作为训练数据。其中 minibatch 的大小设置为 32。另外根据当前网络的输出值和目标值之间差值的平方项来构造误差函数。由 VAM-DRQN 模型结构可知，当前网络的输出值为 $Q\left(\phi(x_j),u_j|\boldsymbol{\theta}\right)$，代表当前状态 $\phi(x_j)$ 下，采取各种可能动作 u_j 的预期累积奖赏。其中目标值设定为：

$$Y_t = \mathbb{E}_{x_{j+1}\sim X}\left[r_j + \gamma \max_{u_{j+1}} Q(\phi(x_{j+1}), u_{j+1}) \middle| \phi(x_j), u_j, \boldsymbol{\theta}_t^- \right] \tag{4.11}$$

因此误差函数的形式为：

$$L_t(\boldsymbol{\theta}_t) = \mathbb{E}_{\phi(x_j), u_j \sim h(.)}\left[\left(Y_t - Q\left(\phi(x_j), u_j \middle| \boldsymbol{\theta}_t\right)\right)^2 \right] \tag{4.12}$$

其中 r_j 表示在 $\phi(x_j)$ 状态下采取 u_j 动作得到的立即奖赏，$\gamma \in [0,1]$ 是一个折扣因子，$h\left(\phi(x_j), u_j\right)$ 表示当前状态 $\phi(x_j)$ 下可采取动作 u_j 的分布，即智能体的行为策略。这里采取的是 $\varepsilon - greedy$ 策略，表示智能体以 $1-\varepsilon$ 的概率选择 Q 值最大对应的动作，以 ε 的概率随机选取一个动作来鼓励探索。评估目标值 Y_t 时，采取对应 Q 值最大的贪心动作。由于智能体的行为策略与评估策略不同，所以该方法属于一种离策略的学习过程。

由于训练 VAM-DRQN 参数时采取的是端对端的形式，所以 $\boldsymbol{\theta}_t$ 代表当前 VAM-DRQN 中所有可训练的参数，$\boldsymbol{\theta}_t^-$ 代表阶段性固定的目标值网络的参数。更新网络参数时，需对误差函数关于参数 $\boldsymbol{\theta}_t$ 求导数以得到梯度项，再使用标准的 Q 学习更新规则：

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\left(Y_t - Q\left(\phi(x_j), u_j \middle| \boldsymbol{\theta}_t\right)\right)\nabla_{\boldsymbol{\theta}_t} Q\left(\phi(x_j), u_j \middle| \boldsymbol{\theta}_t\right) \tag{4.13}$$

以上是 VAM-DRQN 参数学习的过程，训练过程中的一些超参数设置将在下一节中具体阐述。

4.4 仿真实验

本节首先介绍了实验依据的平台和实验过程中需要设置的超参数，随后分别评估了在训练期间和训练完成后的 DQN、LSTM-DQN 和 VAM-DRQN 模型在一些 Atari 2600 战略性游戏中的表现，并结合实验结果分析说明 VAM-DRQN 的优势和适用范围。

4.4.1 实验描述

DQN 等模型运用深度 Q 学习算法来训练智能体，使其在 Atari 2600 平台中的大部分游戏得分赶上甚至超过了人类玩家。但是对于需要智能体经过长时间步的规划才能做出决策的任务而言，这些模型的表现远不能与专业的人类玩家相比。为了

提升智能体在战略性游戏上的性能，本文提出 VAM-DRQN 模型，并选取 5 个 Atari 2600 战略性游戏：深海探险（Seaquest）、星球大战（Alien）、萝卜保卫战（Gopher）、爆破彗星（Asteroids）和邪恶进攻（Gravitar）来设计实验，根据实验结果评估 VAM-DRQN 在战略性游戏上的表现，并与传统 DQN 和 LSTM-DQN 模型进行性能和稳定性上的比较。

4.4.2 实验设置

首先强调一点，不同模型在训练过程中使用相同的超参数集合。另外，由于各个模型中包含的 CNN 结构能够自动地学习良好的特征表达，因此实验中只需输入经过预处理后的游戏画面和游戏奖赏，而不需要针对特定任务去人工设计一些额外的特征。这说明此类深度网络模型在解决基于视觉感知的决策问题时是任务无关的，具有很强的泛化能力。

实验中运用一些常用的技巧以提高模型在训练过程中的稳定性。实验中采用与 DQN 类似的跳帧技巧。即每隔 4 帧智能体才根据 $\varepsilon-greedy$ 策略来选择下一步的动作，而之前时刻只是重复执行当前的动作。由于不同游戏环境下的奖赏量级有很大差异，实验时将所有正奖赏设置为 1，负奖赏设置为-1，零奖赏保持不变。另外将梯度项裁剪到 $[-5,5]$ 区间内，将误差项 $r+\gamma\max_{u'}Q(x',u'|\boldsymbol{\theta}_i^-)-Q(x,u|\boldsymbol{\theta}_i)$ 裁剪到 $[-1,1]$ 区间内。缩减 Q 值、梯度和误差项有利于在不同游戏任务中使用相同的学习率，并有效防止由于策略出现大的波动而导致结果发散或者陷入局部最优解，提高了训练时的稳定性。

模型训练时使用基于均方根的随机梯度下降法（RMSProp）来更新参数，其中动量系数设置为 0.95。每次更新用的样本集是通过经验回放机制从样本池随机抽取 minibatch 个样本得到的，其中 minibatch 的大小为 32。同时，折扣因子 γ 设置为 0.99，学习率 α 和行为策略 $\varepsilon-greedy$ 的参数 ε 都设置为从训练开始到 100 万幅视频帧区间内线性递减的形式，即学习率 α 从 0.005 递减到 0.00025，探索因子 ε 从 1.0 下降到 0.1。样本池的最大容量设置为 100 万个转移样本。在 25000 更新步之前，智能体先采取随机策略存储足够多的转移样本到样本池中，以防止在学习初期由于训练数据太少，而导致学习的偏向性。另外，LSTM-DQN 中 LSTM 单元和 VAM-DRQN 中 GRU 的隐藏层初始状态值都赋值为 $\mathbf{0}$ 向量。

4.4.3 实验结果及分析

深度神经网络模型的训练一般需要大量的训练数据和长时间的训练周期。在传统的深度学习方法中可通过分阶段来评估网络模型的训练过程。而对于强化学习，通常采用从一个情节开始到结束时获得的累积奖赏之和作为评价标准。本章提出的 VAM-DRQN 是一种深度强化学习模型，因此结合以上两种评估形式定义了一种新的度量学习过程的标准：模型训练过程中各阶段的平均每情节奖赏数。

训练 DQN 时采用了 200 个训练阶段，每个阶段包含 250000 时间步的参数更新和 125000 时间步的评估过程。借助于 GPU，DQN 需要大约两周的训练时间。为了保证参数的一致性，不同的模型都采用 100 个阶段作为训练周期。每个阶段的规模设置为 50000 时间步的参数更新过程和 25000 时间步的评估过程。这样借助 GPU 只需不到 48 小时就能训练出一个模型。训练周期缩短的原因主要有两个：首先 VAM-DRQN 模型引入双层 GRU 构成的 RNN 来记忆多时间步的历史信息，使得网络的输入状态仅是当前的一幅游戏画面，减小了状态空间的维度，降低了问题的复杂度；其次通过视觉注意力机制使得智能体有选择性地将注意力集中于一幅图像中面积较小但具有丰富信息的区域，减小了网络可训练的参数的数目，加速了网络的训练速度。在评估不同阶段的训练模型时采取的是 $\varepsilon-greedy$ 策略，其中 ε 设置为 0.05 并一直保持不变。

实验中首先比较三种模型在训练过程中面对 Seaquest 游戏任务时各阶段平均每情节获得的奖赏数。从图 4-3 可以看出，VAM-DRQN 的训练效果最优，并且与另外两种模型之间的性能差距会随着训练时间的推移而变大。LSTM-DQN 的训练性能稍优于 DQN，这是因为相比于 DQN 在每个时刻只能感知离当前时刻最近的连续 4 幅图像，LSTM-DQN 中由单层 LSTM 单元构成的 RNN 能记忆的历史状态信息相对较多，所以延迟奖赏反馈到智能体的可能性也较大。另外从图中还可以看出，VAM-DRQN 模型的训练效果远远优于其它两种模型。下面结合 Seaquest 游戏，对不同模型在训练时存在较大性能差异的原因展开具体分析。

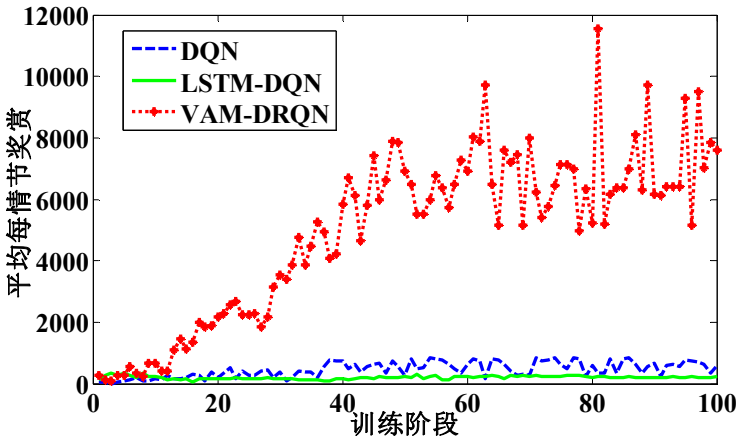


图 4-3 Seaquest 实验中采取不同模型训练时，各阶段中平均每情节奖赏对比

在 Seaquest 游戏过程中，有些动作带来的效益在很多时间步后才体现在游戏画面中并被智能体所感知。比如潜水艇在氧气不足时浮上水面储备氧气的动作。所以当输入状态信息仅仅是由离当前时刻最近的 4 幅连续游戏画面（DQN）或单层 LSTM 单元记忆的有限步长内的历史信息（LSTM-DQN）构成时，智能体可能无法及时得到有延迟的奖赏，从而阻碍了学习的进程。本章提出的 VAM-DRQN 通过双层 GRU 构成的 RNN 模块来记忆较长时间步内的历史信息，有效缓解了动作回报信号的延迟问题。另外在 VAM-DRQN 中引入视觉注意力机制，使得智能体有针对性地选择将注意力集中于对决策具有直接引导作用的一块面积较小的图像区域，减小了网络可训练的参数数目，进一步促进了智能体学习近似最优策略的过程。

从图 4-3 还可以看出，在 VAM-DRQN 的训练过程中，各阶段平均每情节得到的奖赏是有所波动的。这主要是因为模型在训练时，网络的参数细微的变动都会导致输出的动作 Q 值发生改变，从而导致下一阶段策略的分布发生很大的变化[3]。总体而言，VAM-DRQN 在训练过程中，平均每情节所获得的奖赏是会随着训练阶段持续增加的。

此外为了说明 VAM-DRQN 模型在训练过程中的稳定性，图 4-4 对比了三种模型在面对 Seaquest 游戏任务时各阶段平均每情节的最大动作 Q 值。这里的 Q 值表示在任何给定的状态下，智能体采取当前策略到情节结束能够获得的折扣累积奖赏。实验中，Q 值的量级经过裁剪之后缩小数倍，充分保证了学习过程的稳定性。

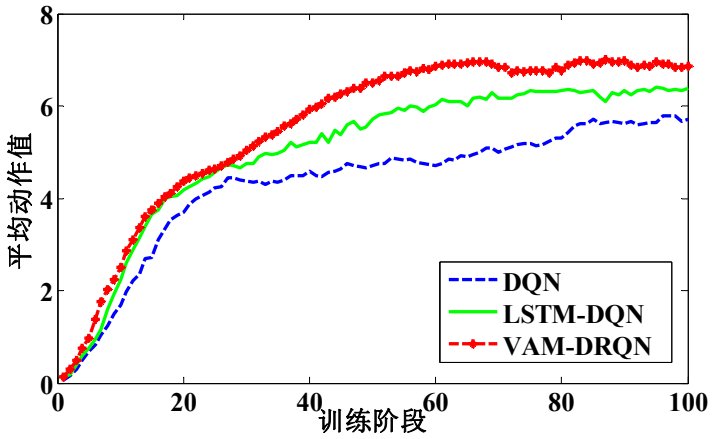


图 4-4 Seaquest 实验中采取不同模型训练时，各阶段中平均每情节最大状态动作值对比

从图 4-4 可以看出，VAM-DRQN 在训练过程中各阶段平均每情节最大 Q 值曲线是优于 DQN 和 LSTM-DQN 的，这也是因为由双层 GRU 构成的 RNN 可以记忆较长时间步内的历史状态信息，以及时将有延迟的奖赏反馈给智能体，并促进 Q 值函数的增长。而对于 DQN 和 LSTM-DQN，由于存在部分有价值信息不可感知的问题，智能体始终不能学习到能够大幅提升游戏性能的关键策略。比如在 Seaquest 游戏中当潜水艇氧气不足时，DQN 和 LSTM-DQN 无法学会浮上水面补充氧气的关键策略。同时 Q 值曲线保持平稳上升并趋向于收敛，充分说明 VAM-DRQN 模型在战略性挑战任务中的有效性和稳定性。

另外，为了直观地说明在 VAM-DRQN 模型中引入 VAM 的作用，实验中可视化分析了 Seaquest 游戏中视觉注意力机制工作的一段场景，如图 4-5 所示。

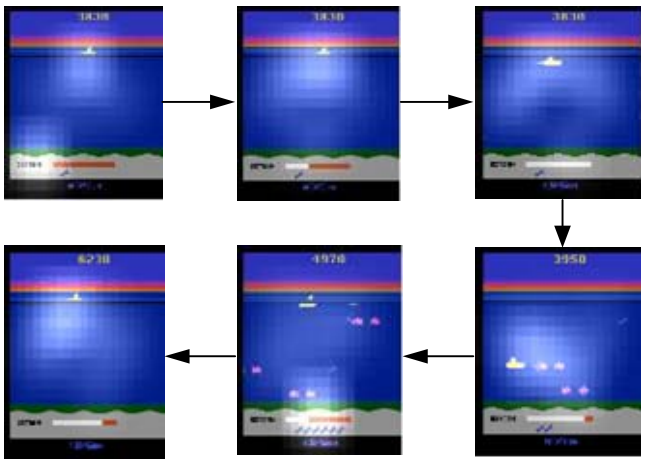


图 4-5 训练完成后 VAM-DRQN 中视觉注意力的可视化过程，其中白色阴影部分表示当前时刻智能体注意力集中的区域

首先从图 4-5 左上方一幅游戏画面可以看出，当潜艇的氧气刚耗光且已露出水面准备存储氧气时，此时的视觉注意力集中于画面下方的氧气指示器和潜艇自身。随后潜艇一直浮在水面储备氧气，该过程不需要再观测氧气探测仪，所以仅将注意力集中于潜艇上。当氧气存储完毕之后，潜艇的注意力开始向两边及下方扩散，以保证再次潜水战斗的安全性。从图 4-5 右下方画面中可以看出，当进入到战斗状态时，潜艇消耗氧气，智能体将注意力转移到离自己最近的敌方所处的那部分区域中。随着潜水时间的增加，氧气的消耗越来越多，此时注意力也更偏向于氧气指示器，当氧气不足时，智能体开始浮上水面准备补充氧气。在补充氧气的过程中，再次将注意力逐步转移到潜艇上。总体上，视觉注意力机制使得智能体在面对不同的处境时，能够将注意力转移到有利于智能体正确做出决策的区域，直观地提升了智能体在此类战略性任务上的表现。

为了进一步说明 VAM-DRQN 模型对于各类战略性任务的适用范围，本文另外评估了 VAM-DRQN 模型在面对 Alien、Gopher、Asteroids 和 Gravitar 这 4 种战略性游戏时的性能表现。图 4-6 比较了不同模型在面对上述 4 种游戏时各阶段的平均每情节奖赏。从图中可看出：在 Alien 和 Gopher 游戏中，VAM-DRQN 的学习曲线和平均每情节奖赏明显优于另外两种模型。这说明在 DQN 模型基础上引入带有 VAM 的 RNN，能够提升智能体的学习性能，从而更好地解决一些基于视觉感知的战略性 DRL 任务。然而从图 4-6 中也可以看出，在 Asteroids 和 Gravitar 游戏中，尽管模型在最后的训练阶段取得了一些进展，但是总体上性能的提升并不明显。尤其在 Gravitar 游戏上，平均每情节奖赏仍然存在着较大的波动。这是因为 Asteroids 和 Gravitar 属于操作难度比较大的游戏，游戏场景的复杂性使得智能体在学习过程中得到的正向回报相对很少，从而阻碍了基于奖惩机制的学习。另外，由于 Gravitar 游戏环境的要求，智能体在训练时需不断地重置游戏以进入新的情节。而该游戏每次重启后的场景是随机变化的，这导致在新的游戏情节中，过去 VAM-DRQN 中通过 VAM 聚焦的那部分图像区域对智能体的学习就显得意义不大。因此 VAM-DRQN 模型还无法在该类游戏上取得突破。总而言之，VAM-DRQN 模型更适用于有频繁的正向回报且游戏场景固定的一类战略性任务。

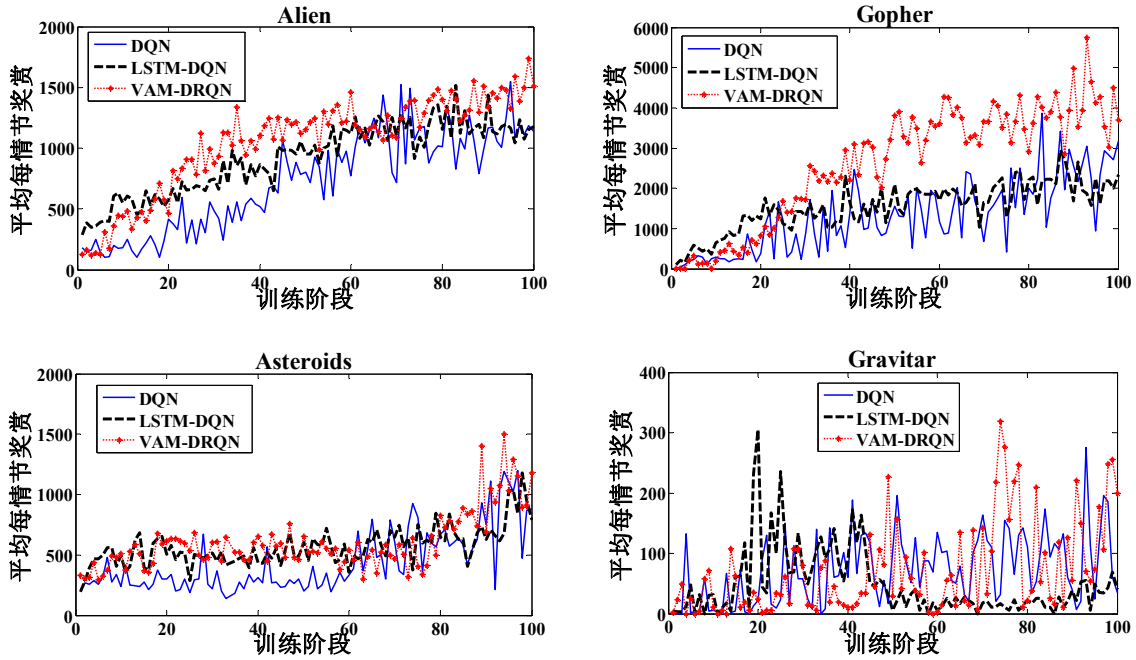


图 4-6 不同模型在 Alien、Gopher、Asteroids 和 Gravitar 游戏任务中各阶段中平均每情节奖赏比较

另一方面，一个性能优异的深度强化学习模型不仅要在训练阶段表现出良好的性能，更重要的是能够通过训练得到一个完整的、可重复利用的策略模型，以使得智能体在每一次的决策任务中，都可以依据训练得到的策略来指导智能体取得优异的表现。由不同模型在训练阶段的表现可以推断，训练完成后的 VAM-DRQN 在面对战略性游戏时的性能是优于其它模型的。

为了验证上述猜想，本节将对比经过训练之后的不同模型在面对上述五种战略性游戏时的性能差异。实验中，通过一个步长为 25000 的游戏测试过程来评估训练完成后不同模型的性能好坏。并且智能体所执行的策略是 $\varepsilon-greedy$ 策略，采取这种策略是为了最小化过度拟合的可能性。其中 $\varepsilon=0.05$。不同游戏中每个训练完成的模型都会被测试 50 次。每次游戏时的初始状态都设置为不同，以保证测试结果的多样性。每次测试获得一个得分，代表该次游戏的平均每情节所得奖赏。一方面，实验中比较了不同游戏中各个模型 50 次测试的平均得分值。另一方面，为了说明训练完成后模型的稳定性，实验中设置了 95%的置信区间来评估 50 次测试得分之间的差异性。由于采用不同方法获得的平均每情节得分在量级上有较大差距，比如 Seaquest 游戏中 VAM-DRQN 的得分均值，就远远大于 DQN 和 LSTM-DQN 的得分均值，因此可通过置信等级这个评价标准来衡量模型性能的稳定性。置信等级的定

义如下：

$$\text{置信等级} = \left(1 - \frac{\text{置信上界-置信下界}}{\text{平均值}}\right) \times 100\% \tag{4.14}$$

由上式可知，置信等级越高，模型在游戏上的表现越稳定。

评估结果如表 4-1 所示。从表中平均值一列可看出，与 DQN 和 LSTM-DQN 相比，训练完成后的 VAM-DRQN 模型在面对战略性游戏时的表现取得了一定幅度的提升。从表中最大值一列中可以看出，训练完成的 VAM-DRQN 在各游戏中的最优表现也基本优于其它模型。尤其在 Seaquest 和 Gopher 游戏中，VAM-DRQN 的游戏表现接近甚至赶上了有经验的人类玩家。不仅如此，由置信等级这一列的结果可得出结论，训练完成的 VAM-DRQN 模型还可以在多次游戏中保持较好的策略稳定性。

表 4-1 训练完成后的不同模型在战略性游戏上的测试得分评估

游戏	智能体	平均值	标准差	最大值	置信下界	置信上界	置信等级
Seaquest	DQN	1106.2	470.6	3980.0	975.8	1236.6	76.4%
	LSTM-DQN	2237.2	320..9	2640.0	2148.2	2326.2	92.1%
	VAM-DRQN	8682.2	3076.6	17550.0	7829.4	9535.0	80.4%
Alien	DQN	1461.4	379.4	2560.0	1356.2	1566.6	85.6%
	LSTM-DQN	1501.8	458.5	2740.0	1374.7	1628.9	83.1%
	VAM-DRQN	1704.2	606.0	4030.0	1536.2	1872.2	80.3%
Gopher	DQN	2750.4	1024.7	4600.0	2466.4	3034.4	79.4%
	LSTM-DQN	2932.0	1016.0	5820.0	2650.4	3213.6	80.8%
	VAM-DRQN	5347.2	2196.2	13080.0	4738.4	5956.0	77.2%
Asteroids	DQN	830.8	374.4	2250.0	727.0	934.6	75.0%
	LSTM-DQN	1009.6	392.8	2350.0	900.7	1118.5	78.4%
	VAM-DRQN	1185.2	473.2	3000.0	1054.0	1316.4	77.9%
Gravitar	DQN	55.0	82.5	250.0	-0.1	27.5	0.11%
	LSTM-DQN	101.0	155.3	500.0	57.9	144.1	14.7%
	VAM-DRQN	107.0	154.9	500.0	64.1	149.9	19.8%

4.5 本章小结

本章提出一种基于视觉注意力机制的深度循环 Q 网络模型架构。一方面，新模型通过在 DQN 基础上引入由双层 GRU 构成的 RNN 来记忆多时间步长的历史状态信息，缓解了有延迟奖赏不能及时反馈给智能体的问题。另一方面，新模型通过视觉注意力机制自适应地将注意力集中于面积较小但更具价值的图像区域中，减少了

网络可学习参数的总数，提高了学习的速率。本章通过 5 个战略性 Atari 2600 游戏，验证了 VAM-DRQN 在此类战略性任务上的有效性。实验结果表明新模型在训练过程中的平均每情节奖赏数和训练速度总体上优于 DQN 和 LSTM-DQN，尤其在 Seaquest 游戏中性能的提升最为明显。进一步地，本文还评估了各模型训练完成之后的性能，结果表明训练完成后的 VAM-DRQN 模型，不仅能提升智能体应对战略性任务的能力，并且还能保持相当稳定的性能表现。

第五章 基于混合目标 Q 值的深度确定性策略梯度算法

前面两章主要从训练算法和模型架构两方面对传统的深度 Q 网络方法进行了改进和完善。然而这些改进的深度强化学习方法只适用于离散动作空间场景，因此必须对现有方法进行算法或模型上的修改，以使其能够适应更加普遍的连续动作空间问题。其中深度确定性策略梯度（DDPG）作为一种代表性的深度强化学习算法，已经在一系列连续动作空间的决策任务中取得了不错的性能[23]。然而在模型训练前期，DDPG 算法由于目标 Q 值估计的不精确导致容易出现性能不稳定的问题。针对这一问题，本章提出一种基于混合目标 Q 值的深度确定性策略梯度算法（Deep Deterministic Policy Gradient with Mixed Update Targets，MIX-DDPG）。MIX-DDPG 结合使用在策略的 MC 更新和离策略的 Q 学习方法来估计目标 Q 值，一定程度上降低了估计目标 Q 值时的误差风险，从而提升了算法在连续动作空间问题中的性能和稳定性。

5.1 策略梯度方法

与基于值函数的强化学习方法不同，基于策略梯度的方法直接在参数化的策略空间中搜索解决问题的最优策略，并且严格保证每一步的更新都能提高当前策略的性能。此类方法的基本思想是用 θ 参数化表示策略 h，并不断计算优化目标关于参数的梯度以用于更新策略，从而使得优化目标函数达到最优或局部最优。其中目标函数可设置为关于策略的期望总奖赏：

$$J(\theta)=\mathbb{E}_h\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] \tag{5.1}$$

另外还可以通过增加总奖赏较高情节出现的概率来优化策略。假设智能体在与环境的交互中生成一个完整情节为 $\varpi=(x_0,u_0,r_0,x_1,u_1,r_1,\ldots,x_{T-1},u_{T-1},r_{T-1},x_T)$，其中 T 是情节的步数，$R^h(\varpi)=\sum_{t=0}^{T-1}\gamma^t r_t$ 表示在遵循策略 h 时该情节内所获得的折扣奖赏总和，$P(\varpi|\theta)$ 表示在当前策略参数 θ 的情况下出现情节 ϖ 的概率分布。此时目标函数可表示为：

$$J(\theta)=\mathbb{E}_h\left[R^h(\varpi)\right]=\int P(\varpi|\theta)R^h(\varpi)\,d\varpi \tag{5.2}$$

$P(\varpi|\theta)$的展开形式为：

$$D(x_0)\prod_{t=0}^{T-1} f(x_t,u_t,x_{t+1})h(u_t|x_t,\theta) \tag{5.3}$$

上式中$D(x_0)$表示初始状态的概率分布，$f(x_t,u_t,x_{t+1})$表示在状态x_t下执行u_t动作得到下一状态x_{t+1}的概率。如果只考虑确定性环境，即$f(x_t,u_t,x_{t+1})$的概率恒为1。此时目标函数可修改为如下形式：

$$J(\theta)=\int\sum_{t=0}^{T-1}\log h(u_t|x_t,\theta)R^h(\varpi)d\varpi \tag{5.4}$$

为了降低梯度项的方差，可使用如下的标准化操作：

$$J(\theta)=\int\sum_{t=0}^{T-1}\log h(u_t|x_t,\theta)\left(R^h(\varpi)-b^h(\varpi)\right)d\varpi \tag{5.5}$$

其中$b^h(\varpi)$是与当前情节ϖ相关的一个基线，通常设置为$R^h(\varpi)$的一个期望估计值，比如状态值函数$V^h(x_0)$。最后求出目标函数关于策略参数θ的导数，并在这个梯度方向上更新参数：

$$\theta_{t+1}=\theta_t+\alpha\nabla_\theta J(\theta) \tag{5.6}$$

其中α表示更新时的学习率。该步骤的目的是更新参数θ直到收敛为θ^*，以最大化期望回报$J(\theta)$。

在面对连续动作空间中的深度强化学习问题时，可采用权重为θ的深度神经网络来参数化表示策略，并利用上述的策略梯度方法以端对端的方式直接在策略空间中搜索最优策略。上述过程即为深度策略梯度方法的核心思想。与基于值函数的深度强化学习方法相比，深度策略梯度方法适用的场景更为广泛，策略优化的效果也更好。

5.2 基于行动者评论家框架的深度确定性策略梯度方法

深度策略梯度方法在每个迭代步都需要采样N个完整情节$\{\varpi_i\}_{i=1}^N$来作为训练样本，然后构造目标函数关于策略参数的梯度项以求解最优策略。然而在许多现实场景下的任务中，很难在线获得大量完整情节的样本数据。例如在真实场景下机器

人的操控任务中，在线收集并利用大量的完整情节会产生十分昂贵的代价，并且连续动作的特性使得在线抽取批量情节的方式无法覆盖整个状态特征空间。上述问题会导致算法求解最优策略时出现局部最优解。针对上述问题，可将传统强化学习中的行动者评论家框架拓展到深度策略梯度方法中。这类算法被统称为基于 AC 框架的深度策略梯度方法。其中最具代表性的是深度确定性策略梯度算法，该算法能够解决一系列连续动作空间中的控制问题。下面具体介绍 DDPG 算法的原理。

DDPG 分别使用权重为θ^{μ}和θ^{Q}的深度神经网络来表示确定性策略$u=h(x|\theta^{\mu})$和值函数$Q(x,u|\theta^{Q})$。其中策略网络用于进行选择策略，对应 AC 框架中的行动者；值网络用于评估当前状态动作对的 Q 值，评估完成后再向策略网络提供更新策略权重θ^{μ}的梯度信息。因此值网络对应 AC 框架中的评论家。在 DDPG 算法中，优化目标被定义为累积折扣奖赏：

$$J\left(\theta^{\mu}\right)=\mathbb{E}_{\theta^{\mu}}\left[r_0+\gamma r_1+\gamma^2 r_2+\ldots\right] \tag{5.7}$$

然后采用 SGD 方法对优化目标关于策略参数求偏导数。Silver 等人证明在确定性的环境下，目标函数关于权重θ^{μ}的梯度等价于 Q 值函数关于θ^{μ}梯度的期望[85]：

$$\frac{\partial J\left(\theta^{\mu}\right)}{\partial\theta^{\mu}}=\mathbb{E}_{x}\left[\frac{\partial Q\left(x,u|\theta^{Q}\right)}{\partial\theta^{\mu}}\right] \tag{5.8}$$

根据确定性策略$u=h(x|\theta^{\mu})$可得：

$$\frac{\partial J\left(\theta^{\mu}\right)}{\partial\theta^{\mu}}=\mathbb{E}_{x}\left[\frac{\partial Q\left(x,u|\theta^{Q}\right)}{\partial u}\frac{\partial h\left(x|\theta^{\mu}\right)}{\partial\theta^{\mu}}\right] \tag{5.9}$$

通过 DQN 中更新值网络的方法来更新评论家网络的权重，更新时梯度项为：

$$\frac{\partial L\left(\theta^{Q}\right)}{\partial\theta^{Q}}=\mathbb{E}_{x,u,r,x'\sim D}\left[\left(Y-Q\left(x,u|\theta^{Q}\right)\right)\frac{\partial Q\left(x,u|\theta^{Q}\right)}{\partial\theta^{Q}}\right] \tag{5.10}$$

其中$Y=r+\gamma Q(x',h(x'|\hat{\theta}^{\mu})|\hat{\theta}^{Q})$表示目标 Q 值，$\hat{\theta}^{\mu}$和$\hat{\theta}^{Q}$分别表示目标策略网络和目标值网络的权重。每次更新时，DDPG 使用经验回放机制从样本池中抽取固定数量的转移样本，并将由 Q 值函数关于动作的梯度信息从评论家网络传递到行动者网络。最后依据式 5.9 沿着提升 Q 值的方向更新策略网络的参数，以求解最优策略。

DDPG 不仅能够在一系列连续动作空间中的控制任务中表现出色，而且求得最优策略所需要的时间步也远远少于基于值函数的深度强化学习方法。然而 DDPG 算法也存在着一些问题：（1）在学习的初期，目标值网络对于下一状态动作对 Q 值函数$Q(x',h(x'|\hat{\theta}^{\mu})|\hat{\theta}^{Q})$的估计还不够精确，导致目标 Q 值的计算出现偏差，从而使得 DDPG 算法在性能上会出现一定的波动。因此 DDPG 算法不适用于那些对稳定性要求很高的实时性控制任务；（2）在有噪声干扰的复杂环境下，策略一般具有一定的随机性。由于 DDPG 算法使用确定性的策略梯度方法来优化策略，因此该方法并不适用于那些具有随机性策略特征的任务场景。本章主要针对上述问题 1，提出一种基于混合目标 Q 值的深度确定性策略梯度算法。新算法在学习过程中结合使用 MC 和 Q 学习方法来估计目标 Q 值，一定程度上减小了值网络估计目标 Q 值时的误差，提升了智能体在连续动作空间问题上的性能和稳定性。

5.3 基于混合目标 Q 值的深度确定性策略梯度算法

5.3.1 混合目标 Q 值的定义

在深度强化学习算法中，通常使用深度神经网络来表示值函数或策略。因此如果任务的状态空间中存在很多相似的状态动作对，神经网络强大的泛化作用能够大大提高算法的性能。然而重复的引导式（bootstrapped）TD 更新会在某种情况下导致值函数发散。考虑一步 TD 方法的目标 Q 值：$Y_t = r_{t+1} + \gamma \max_u Q\left(x_{t+1}, u \middle| \theta\right)$，如果$r_{t+1} > 0$并且$x_{t+1}$和$x_t$是相似状态，那么目标 Q 值将很快发散。DQN 中使用目标值网络来缓解此问题，此时目标 Q 值的形式为：$Y_t = r_{t+1} + \gamma \max_u Q\left(x_{t+1}, u \middle| \theta^{-}\right)$。其中$\theta^{-}$表示目标值网络的权重，并且每隔固定的步数将当前值网络的权重赋值给目标值网络。DDPG 算法进一步改进目标 Q 值的构造过程，其目标 Q 值的形式为：$Y_t = r_{t+1} + \gamma Q(x_{t+1}, h(x_{t+1}|\hat{\theta}^{\mu})|\hat{\theta}^{Q})$，并且目标值网络$\hat{\theta}^{Q}$和目标策略网络$\hat{\theta}^{\mu}$的权重按照$\hat{\theta} \leftarrow \tau\theta + (1-\tau)\hat{\theta}$的规则来更新。其中$\tau$一般设置为 0.001。该种目标网络权重的更新方式放缓了目标 Q 值的更新速度，极大地提升了算法的稳定性。与 DQN 不同的是，DDPG 中使用策略网络输出的确定性动作来替代下一状态动作对中 Q 值最大的动作。这是由于在连续动作空间中，直接通过计算 Q 值来确定最优的动作是行不通的。然而只有当行动者输出的确定性动作等价于值网络确定的最优动作时，目标 Q 值的估计误差才会相对较小。在模型训练的初期，目标网络对于下一状态的 Q 值函

数和策略网络对于确定性策略的估计都不够精确。此时可采用在策略的 MC 方法来辅助估计当前状态的目标 Q 值。

MC 方法可直接依据完整情节样本$\varpi=\left(x_0,u_0,r_0,x_1,u_1,r_1,\ldots,x_{T-1},u_{T-1},r_{T-1},x_T\right)$中的每个时刻奖赏值$r_t$来估计某个状态动作对的值函数，省去了构造目标值网络的繁琐过程。并且通过 MC 方法构造出的目标 Q 值不会出现发散的情况，这是由于从环境中得到的每个真实奖赏r_t都有一定的上限值。具体地，MC 目标 Q 值的计算过程如算法 5-1 所示。

算法 5-1　MC 目标 Q 值的计算过程

1：**初始条件**：完整情节样本$\varpi=\left(x_0,u_0,r_0,x_1,u_1,r_1,\ldots,x_{T-1},u_{T-1},r_{T-1},x_T\right)$，回放记忆单元$D_{MC}$的容量为$N$，$R\leftarrow 0$

2：**For** $t=T,T-1,\ldots,0$ **do**

3：　　$R\leftarrow r_t+\gamma R$

4：　　$y_t^{MC}\leftarrow R$

5：　　将转移序列$\left(x_t,u_t,y_t^{MC},r_t,x_{t+1}\right)$存储到$D_{MC}$中

6：**输出**：目标 Q 值的估计y_t^{MC}

其中y_t^{MC}表示在策略的 MC 目标 Q 值的估计。由算法 5-1 可知，y_t^{MC}被当做一项独立的元素存储于转移序列中。因此在需要利用 MC 方法估计某个状态动作对的值函数时可直接从转移序列中获取，而不需要任何额外的计算。然而在仅使用 MC 方法来估计值函数时，动作的探索（例如ε-*greedy*）会导致 Q 值估计出现一定的偏差。并且随着目标网络权重的不断更新，下一状态动作对的 Q 值函数和确定性策略的估计也逐渐精确，此时应该侧重于利用离策略的 Q 学习方法来估计目标 Q 值。因此本节提出一种混合型的目标 Q 值构成方式。新的目标 Q 值由两部分组成：在策略的 MC 估计值和离策略的一步 Q 学习估计值。具体的形式为：

$$Y^{MIX}=\beta y^{MC}+\left(1-\beta\right)y^{Q-learning} \tag{5.11}$$

其中$\beta\in[0,1]$，用于动态调整构成整体目标 Q 值时各组成部分所占的比重。当$\beta=0$时，混合目标 Q 值Y^{MIX}退化为传统的 1 步 Q 学习估计值；当$\beta=1$时，Y^{MIX}为传统的 MC 估计值。在模型训练初期，目标值网络对于下一状态的 Q 值和目标策略网络

对于确定性策略的估计都不够精确，此时使用一步 Q 学习会导致目标 Q 值出现一定的偏差，从而影响算法的稳定性。因此在训练的初期应该侧重于使用 MC 方法来估计目标 Q 值。而随着模型的不断训练，对于 Q 值和确定性策略的估计会越来越精确，并且动作的探索也不会导致 Q 值的估计出现太大的波动。此时应该逐渐增加 Q 学习更新部分在构成整体目标 Q 值时所占的比例。因此本节使用一种 β 线性衰减的机制来动态调整在不同学习时刻整体目标 Q 值中各组成部分所占的比重。具体设置为在前 100000 时间步 β 从 1 线性递减到 0。

5.3.2 训练算法描述

在连续动作空间场景下，直接结合使用深度神经网络和策略梯度方法会导致算法性能的不稳定。因此本章提出的 MIX-DDPG 算法运用以下技巧来提高算法训练时的稳定性：

（1）使用本文第三章提出的基于优先级采样的经验回放机制来替代随机采样。该采样机制首先根据奖赏值是否大于零，将转移样本 $(x_t,u_t,r_t,v_t,x_{t+1})$ 存储到不同的回放记忆单元。然后根据转移样本优先级确定的采样概率分布从回放记忆单元中抽取固定大小的训练样本，并基于这些样本的随机梯度项来更新值网络和策略网络的权重。该种抽样方式提高了有价值转移样本的利用率，并使得样本空间中的每个转移样本都以一定的概率被访问到，从而有助于提升算法的稳定性和收敛速度；

（2）使用目标值网络和目标策略网络。具体地，目标网络的权重通过缓慢“追踪”当前网络权重的方式来更新：$\hat{\theta}^{Q} \leftarrow \tau\theta^{Q}+(1-\tau)\hat{\theta}^{Q}$ 和 $\hat{\theta}^{\mu} \leftarrow \tau\theta^{\mu}+(1-\tau)\hat{\theta}^{\mu}$，其中，$\tau$=0.001。通过该技巧限制了目标值更新的速度，提升了算法的稳定性；

（3）使用一种随机化过程来保证智能体对未知状态动作空间的探索。算法通过在当前动作中添加随机化过程产生的噪音来促进探索：$u_t = h\left(x_t \middle| \theta^{\mu}\right)+N_t$。

（4）使用上节中定义的混合目标 Q 值来构造损失函数。在算法训练的初期，目标网络对下一状态的 Q 值和确定性策略的估计都不够精确。此时仅通过单步的 Q 学习来确定目标 Q 值会导致算法性能的不稳定。MIX-DDPG 算法结合使用在策略的 MC 方法和离策略的单步 Q 学习方法，构造出一种新颖的混合目标 Q 值，减小了目标 Q 值的估计误差，提升了算法的性能和稳定性。

算法 5-2 给出了基于混合目标 Q 值的深度确定性策略梯度算法的执行流程。

算法 5-2 基于混合目标 Q 值的深度确定性策略梯度算法

1：**初始化**：回放记忆单元 D_1 和 D_2 的容量为 N，值网络和策略网络的初始权重为 θ^Q 和 θ^μ，目标值网络和目标策略网络的初始权重为 $\hat{\theta}^Q \leftarrow \theta^Q$ 和 $\hat{\theta}^\mu \leftarrow \theta^\mu$

2：**Repeat**（对每一个情节）：

3：　初始状态设置为 x_0

4：　初始化一个随机过程 N 用于动作的探索

5：　**Repeat**（对于情节中的每一步）：

6：　根据当前的策略网络和探索噪声来选择动作 $u_t = h\left(x_t \middle| \theta^\mu\right) + N_t$

7：　执行动作 u_t，并得到奖赏 r_t 和下一个状态 x_{t+1}

8：　如果 $r_t > 0$，则将转移序列 $\left(x_t, u_t, r_t, v_t, x_{t+1}\right)$ 存储到 D_1 中

9：　否则将存储转移序列 $\left(x_t, u_t, r_t, v_t, x_{t+1}\right)$ 到 D_2 中

10：　**Repeat**（对于每个 minibatch）：

11：　如果 $random() < \rho$：则

12：　以概率分布 $P(j) = (p_j)^\alpha \Big/ \sum_i (p_i)^\alpha$ 从 D_1 中抽取 $\left(x_j, u_j, r_j, v_j, x_{j+1}\right)$

13：　否则：

14：　以概率分布 $P(j) = (p_j)^\alpha \Big/ \sum_i (p_i)^\alpha$ 从 D_2 中抽取 $\left(x_j, u_j, r_j, v_j, x_{j+1}\right)$

15：　更新访问次数：$v_j = v_j + 1$

16：　更新转移样本的优先级：$p_j = 1 \Big/ (v_j + 1)$

17：　**Until** 该批次的迭代轮结束

18：　利用 1 步的 Q 学习估计目标 Q 值：$y_j^{Q-learning} = r_{j+1} + \gamma Q(x_{j+1}, h(x_{j+1} | \hat{\theta}^\mu) | \hat{\theta}^Q)$

19：　从回放记忆单元 D_{MC} 中搜索当前状态 x_j 的 MC 估计值 y_j^{MC}

20：　设置混合型目标 Q 值：$Y_j^{\mathrm{MIX}} = \beta y_j^{MC} + (1-\beta) y_j^{Q-learning}$

21：　通过最小化损失函数 $L\left(\theta^Q\right) = \left(Y_j^{MIX} - Q\left(x_j, u_j \middle| \theta^Q\right)\right)^2$ 来更新值网络

22：　更新策略网络：$\nabla_{\theta^\mu} J\left(\theta^\mu \middle| x_j\right) \approx \frac{1}{N} \sum_j \nabla_u Q\left(x_j, h\left(x_j \middle| \theta^\mu\right) \middle| \theta^Q\right) \nabla_{\theta^\mu} h\left(x_j \middle| \theta^\mu\right)$

23：　更新目标网络的权重 $\hat{\theta}^Q \leftarrow \tau\theta^Q + (1-\tau)\hat{\theta}^Q$，$\hat{\theta}^\mu \leftarrow \tau\theta^\mu + (1-\tau)\hat{\theta}^\mu$

24：　**Until** 到达终止状态

25：**Until** 情节数到达上限次数 M

5.4 仿真实验

5.4.1 实验描述

本章首先通过平衡杆（Cart-Pole）、二级倒立摆（Double Inverted_Pendulum）这两个经典的连续动作空间任务来验证 MIX-DDPG 算法的性能。图 5-1 展示了这 2 种任务的环境示意图。

图 5-1 平衡杆和二级倒立摆环境示意图

平衡杆任务的目标是通过移动杆下方的小车使连接在上方的杆保持垂直不倒。该任务的观察状态包括小车位置 p、小车速度 $\hat{p}$、杆偏离垂直方向的角度 θ、杆摆动的速度 $\hat{\theta}$。动作集合设置为水平方向上施加给小车的力，范围为 $[-5N,5N]$。为了保证实验的稳定性，小车和杆的最大速度限制为 $4\,m/s$。当 $|x|>2.4m$ 或 $|\theta|>0.2\pi$ 时，情节结束。奖赏函数设置为：

$$r(x,u)=10-(1-\cos(\theta))-0.00001\|u\|_2^2 \tag{5.12}$$

二级倒立摆的目标是通过移动杆下方的小车使连接在上方的两个杆保持垂直不倒。该任务的观察状态包括小车位置 p、杆 l_1 和 l_2 偏离垂直方向的角度 θ_1 和 θ_2 和杆 l_1 和 l_2 摆动的速度 $\hat{\theta}_1$ 和 $\hat{\theta}_2$。动作集合设置为水平方向上施加给小车的力，范围为 $[-10N,10N]$。奖赏函数设置为：

$$r(x,u)=10-0.01x_{tip}^2-(y_{tip}-2)^2-0.001\hat{\theta}_1^2-0.005\hat{\theta}_2^2 \tag{5.13}$$

其中 x_{tip} 和 y_{tip} 代表杆顶端的坐标值。上式中的 $0.01x_{tip}^2+(y_{tip}-2)^2$ 代表距离惩罚项，$0.001\hat{\theta}_1^2+0.005\hat{\theta}_2^2$ 表示速度惩罚项。当 $y_{tip}<=1\,m$ 时，情节结束。

另外本章还通过 MuJoCo 平台中的 Half-Cheetah 任务（如图 5-2）来进一步验证 MIX-DDPG 算法的性能。其中 MuJoCo 是一个模拟机器人运动和控制的程序接口。Half-Cheetah 任务的目标是通过控制平面上的四足机器人的关节点摆动幅度，

加快机器人向前移动的速度。此类控制任务动作的自由度很大，并且需要大量的动作探索步来防止算法陷入局部最优解。具体地，该任务有 9 个关节点，包括 2 处腿关节、1 处躯干关节和 6 处驱动关节。因此比起平衡杆和二级倒立摆，Half-Cheetah 更具挑战性。该任务的观察状态有 20 维度，包括 9 个关节点的角度、9 个关节点的速度和机器人质心点的坐标。奖赏函数设置为：

$$r(x,u)=v_x-0.05\|u\|_2^2 \tag{5.14}$$

其中 v_x 表示机器人前进的速度。

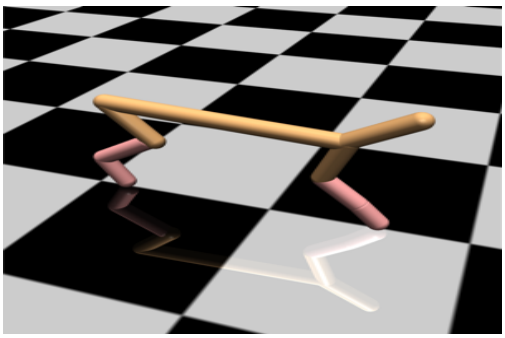


图 5-2 Half-Cheetah 环境示意图

5.4.2 实验设置

首先需要考虑 MIX-DDPG 算法中有关网络模型的一些设置。由于实验中采用的是低维度的状态特征表示（位置、速度等），因此只需使用较浅的神经网络模型来分别表示值网络和策略网络。其中值网络和策略网络都设置为具有三层隐藏单元的全连接网络，每层的神经元数量分别为 500、400 和 300，并且每个隐藏层之后都使用 RELU 函数进行非线性变换。为了保证初始时值网络和策略网络的输出值接近于零，两个网络的输出层权重初始化为$[-3\times10^{-3},3\times10^{3}]$区间内的均匀分布。输入层和隐藏层的权重则被初始化为$[-(\sqrt{f_{in}})^{-1},(\sqrt{f_{in}})^{-1}]$，其中 f_{in} 表示当前层的输入值。

另外由于在不同类型的控制任务中，状态特征向量中每一维度的量级大小是有差别的。因此在面对不同环境时，设置一套固定的网络模型和超参数来高效地学习策略是不切实际的。为了提高泛化性，MIX-DDPG 使用了批量归一化（Batch Normalization）的技巧[86]。该技巧可使得不同环境下每批次样本中的各维度数据拥有统一的均值和方差，缓解了训练时由于协方差变换而导致性能不稳定的问题。除了值网络和策略网络的输出层之外，其余每层都使用批量归一化操作来保证输入的

向量拥有统一的均值和方差。

最后考虑 MIX-DDPG 训练算法中需要设置的一些超参数。“软”因子 τ 设置为 0.001，参数 ρ 在训练期间从 0.5 线性衰减到 0.25，随机化因子设置为 0.6。实验中采用的是 Adam 优化方法[87]来学习模型的参数，值网络和策略网络模型的学习率分别设置为 0.0001 和 0.001。训练时批量大小设置为 128，回放记忆单元 D_1、D_2 和 D_{MC} 的容量上限设置为 100 万个转移样本。针对动作的探索，实验中采用 Ornstein-Uhlenbeck 过程来产生随机化的噪声。其中，$\theta = 0.15, \sigma = 0.3$。另外为了保证模型在训练的不同时刻，目标 Q 值的估计都保持一定的精确度。实验中将参数 β 设置为线性递减的形式。具体设置为：在训练的前 100000 时间步从 1 线性衰减到 0。因此在构造混合目标 Q 值时，智能体能够在训练的前期重视在策略的 MC 估计值，而在训练的中后期则偏向于离策略的 Q 学习估计值。最后实验中采用的神经网络模型需要 500 个周期共 25 万时间步进行训练。

5.4.3 实验结果及分析

实验中，将 MIX-DDPG 算法应用到平衡杆、二级倒立摆和 Half-Cheetah 任务中，目的是验证算法在连续动作空间问题中的性能。另外为了充分说明 MIX-DDPG 算法相比于传统策略优化方法的优点，实验中选取了一些经典的解决连续动作空间问题的优化算法作为对比实验，包括基于加权奖赏值的回归[88]（Reward-Weighted Regression，RWR）、截断式自然策略梯度[34]（Truncated Natural Policy Gradient，TNPG）和 DDPG 算法。实验中重点对比各算法在训练周期中各阶段平均每情节的奖赏数。为了保证参数的一致性，不同算法都采用 500 个阶段作为训练周期。每个阶段包含 500 时间步的参数更新和 250 时间步的评估过程。

首先分析 MIX-DDPG 算法在操作难度较低的平衡杆和二级倒立摆问题中的性能。由于 MIX-DDPG 是基于 DDPG 算法的改进，因此主要分析这两个算法之间的性能差异。由图 5-3 和图 5-4 可以看出，MIX-DDPG 算法的收敛速度远优于 DDPG 算法。具体体现在：MIX-DDPG 算法在平衡杆任务中大概需要 15 个训练阶段就可以达到收敛状态，而 DDPG 算法需要大约 90 个训练阶段才勉强收敛。在操作难度稍大的二级倒立摆问题中，MIX-DDPG 算法在大约 30 个训练阶段之后达到收敛状态，而 DDPG 算法则需要大约 150 个训练阶段才收敛。收敛速度加快的主要原因是：

MIX-DDPG 算法使用基于优先级的回放机制，在训练前期增加了有价值转移样本的利用率，从而促进了智能体的学习。另外从图 5-3 和图 5-4 可以看出，MIX-DDPG 算法性能的稳定性很好。这体现在 MIX-DDPG 算法的性能曲线在收敛之前可以保持平缓的上升，在收敛之后也一直保持稳定，而 DDPG 算法在训练期间的性能曲线一直有波动。MIX-DDPG 算法稳定性提升的原因在于：一方面，该算法使用基于优先级的回放机制，使得样本空间中的每个转移样本都以非 0 的概率被访问到，一定程度上降低了算法陷入局部最优解的风险；另一方面，该算法结合使用 MC 估计和单步 Q 学习的方法，构造出一种新颖的混合目标 Q 值，减小了估计目标 Q 值的误差。此外 TNPG 算法的性能与 MIX-DDPG 算法最为接近。这是由于 TNPG 每次更新时在梯度方向上限制策略分布的改变幅度，从而保证智能体的不断学习。不过与 MIX-DDPG 算法相比，TNPG 算法在训练过程中的稳定性还是相对较差。

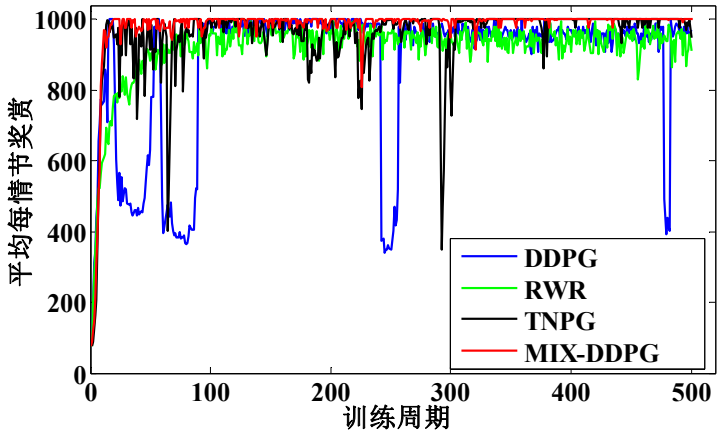


图 5-3 Cart-Pole 中不同优化算法在训练时，各阶段中平均每情节奖赏对比

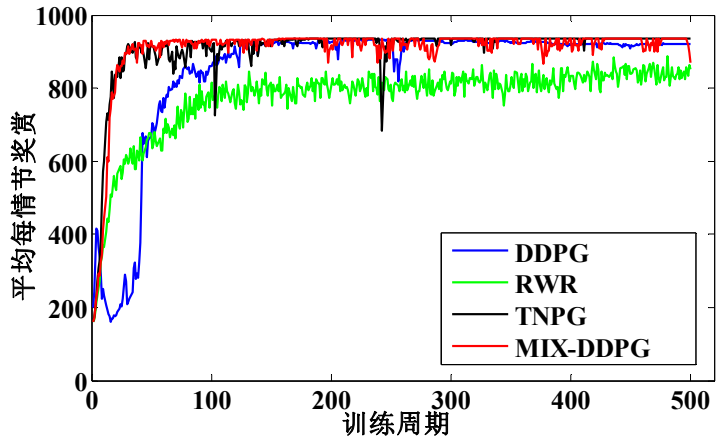


图 5-4 Double Inverted_Pendulum 中不同优化算法在训练时，各阶段中平均每情节奖赏对比

接着分析 MIX-DDPG 算法在操作难度较大的 Half-Cheetah 任务中的性能表现。从图 5-5 可以看出，MIX-DDPG 算法在状态特征空间较大的复杂任务中仍然能够取得较快的收敛速度和较稳定的性能表现。不过由于任务的复杂性，MIX-DDPG 算法需要大约 400 个训练阶段才大致收敛。另外由于 Half-Cheetah 任务中的状态特征空间较为复杂，需要表征能力较强的逼近器来表示策略和值函数。传统策略优化方法中的线性逼近器包含的可训练权重数目不够，在面对大规模状态空间的决策任务时容易出现欠拟合和局部最优解的问题。而在 MIX-DDPG 算法中使用权重数目较多的策略网络和值网络，泛化大规模状态空间下策略和值函数的效果更好。因此 MIX-DDPG 算法在面对复杂决策任务时，收敛速度和稳定性上的优势更为明显。

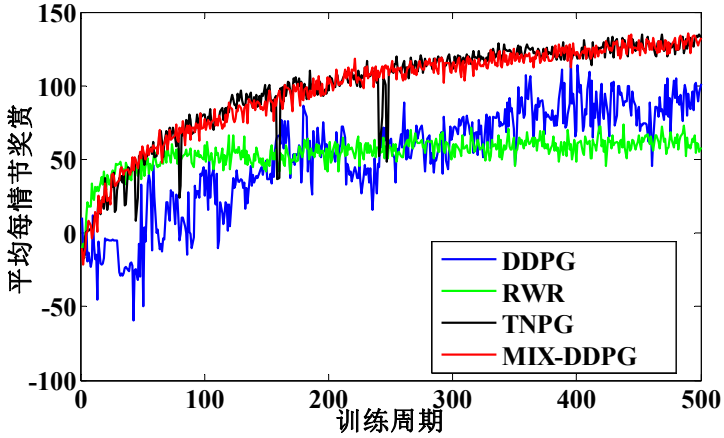


图 5-5　Half-Cheetah 中不同优化算法在训练时，各阶段中平均每情节奖赏对比

综上所述，MIX-DDPG 算法在简单和复杂的连续动作空间任务中都具有收敛速度快、算法性能好和稳定性高的优势。

5.5 本章小结

本章针对传统的深度确定性策略梯度方法估计目标 Q 值时存在一定误差的问题，结合在策略的 MC 估计和离策略的 Q 学习方法构造出一种混合型的目标 Q 值，并提出一种基于混合目标 Q 值的深度确定性策略梯度方法。然后本章将 MIX-DDPG 算法应用于 3 个经典的连续动作空间任务问题。从实验结果可得出结论：在解决连续动作空间下的控制任务时，MIX-DDPG 算法具有收敛速度快和性能稳定的优点。

第六章 总结与展望

6.1 总结

深度强化学习作为机器学习领域的一个新颖分支，具有广阔的应用场景。但是目前大多数深度强化学习算法在面对现实场景下的复杂问题时，都面临着稀疏和延迟奖赏、部分状态可观察、收敛速度慢、稳定性较差等问题。针对上述问题，本文从训练算法和模型架构两方面对传统的深度 Q 网络方法进行了改进和完善。主要的研究内容总结如下：

（1）传统的深度 Q 网络方法区分不出转移样本之间的重要性差异，因此不能充分利用有价值的样本。另外有限容量的经验回放池会导致出现样本的覆盖和重复利用。针对上述问题，提出一种基于优先级采样深度 Q 学习算法。该算法主要的创新点在于提出一种高效的基于优先级的回放机制来替代随机采样。该种优先级采样机制可以提高有价值转移样本的利用率，并保证样本空间中的每个转移序列都以一定大小的概率被采样，从而促进了算法收敛的速度。另外该种抽样方式在学习初期可充分利用带有正奖赏的转移样本，一定程度上缓解了复杂问题中存在的稀疏奖赏问题。实验结果表明 PS-DQN 算法可适用于一些基于视觉感知的大规模决策任务，并且在收敛速度、稳定性和得分性能上都超过了传统的 DQN 算法。

（2）针对传统的深度 Q 网络不擅长解决战略性深度强化学习任务的问题，将循环神经网络和视觉注意力机制引入到深度 Q 网络模型架构中，提出一种基于视觉注意力机制的深度循环 Q 网络模型。新模型主要有两处创新点：一是使用由双层门限循环单元构成的循环神经网络模块来记忆较长时间步内的历史状态信息，从而使智能体能够及时响应有延迟的奖赏；二是使用视觉注意力机制自适应地将智能体的注意力集中于面积较小但更具价值的图像区域，减小了模型架构可训练的权重数目，从而加快学习最优策略的进程。实验结果表明 VAM-DRQN 模型可以提升智能体应对一些操作难度较大的战略性任务的能力，并且还能保持较快的收敛速度和较好的稳定性。

（3）针对传统的深度确定性策略梯度方法在训练前期估计目标 Q 值时存在一

定误差的问题，提出一种基于混合目标 Q 值的深度确定性策略梯度方法。该算法较之于传统的 DDPG 主要有两处不同：一是结合使用在策略的 MC 估计和离策略的 Q 学习方法来生成一种混合型的目标 Q 值，一定程度上降低了估计目标 Q 值时的误差；二是使用第三章提出的优先级采样来替代原先的随机采样。实验结果表明，MIX-DDPG 算法可适用于连续动作空间下的控制任务，并且加快了收敛速度，增强了性能的稳定性。

6.2 展望

本文基于深度 Q 网络方法设计出一系列深度强化学习算法和模型，提高了智能体在面对离散和连续动作空间任务时的表现，并且取得了更快的收敛速度和更好的稳定性。但仍存在着一些问题值得在后续的研究工作中进行进一步的优化：

（1）本文提出一系列基于深度 Q 网络的改进算法，尽管改善了稀疏和延迟奖赏、部分状态可观察、收敛速度慢、稳定性不高等问题，但是这些改进算法的训练还很依赖于强大的硬件支撑，网络模型的训练还很耗时。而异步的强化学习方法一方面可减少模型的训练时间，另一方面也能降低训练时对硬件的要求。因此可利用异步的强化学习框架来进一步扩展本文提出的算法。

（2）本文对传统深度强化学习算法中的经验回放机制进行了改进，提高了一些有价值样本的利用率。不过这些算法的训练还是要依赖于大量的样本数据。在很多现实场景下的复杂决策任务中，真实样本数据是十分缺乏的。因此可通过无监督的生成模型（比如对抗生成网络[89]）来生成大量仿真的训练数据，从而降低模型训练对样本数量的要求。

（3）本文提出的基于视觉注意力机制的深度循环 Q 网络模型具备了很强的感知和决策能力，因此提升了智能体应对战略性挑战任务的能力。然而在面对一些高层次的认知启发式任务时，智能体不仅需要具备感知和决策能力，而且得拥有一定的记忆、规划与推理能力。因此将一些外部的记忆[36,90,91]与规划网络部件[73]引入到现有模型中以提高智能体解决高层次复杂任务的性能，是下一步主要的研究方向。

参考文献

[1] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[2] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 1998.

[3] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

[4] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[5] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. International Conference on Neural Information Processing Systems, 2012, 25(2): 1097-1105.

[6] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[A]. Computer Vision and Pattern Recognition[C]. Las Vegas: IEEE, 2016: 770-778.

[7] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[J]. Acoustics, Speech and Signal Processing, 2013, 38(2003): 6645-6649.

[8] Oord A V D, Dieleman S, Zen H, et al. WaveNet: A generative model for raw audio[J]. CoRR abs/1609.03499, 2016.

[9] Cho K, Merrienboer B V, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[A]. Empirical Methods in Natural Language Processing[C]. Doha: ACL, 2014: 1724-1734.

[10]Levine S, Finn C, Darrell T, et al. End-to-End training of deep visuomotor policies[J]. Journal of Machine Learning Research, 2016, 17(39): 1-40.

[11]Tesauro G. TD-Gammon, a self-teaching backgammon program, achieves master-level play[J]. Neural Computation, 1994, 6(2): 215-219.

[12]高阳，周如益，王皓，曹志新. 平均奖赏强化学习算法研究[J]. 计算机学报，2007, 30(8): 1372-1378.

[13]傅启明，刘全，王辉，肖飞，于俊，李娇. 一种基于线性函数逼近的离策略 q(λ)

算法[J]. 计算机学报, 2014, 37(3): 677-686.

[14]Kober J, Peters J. Reinforcement learning in robotics: a survey[J]. International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[15]Baker B, Gupta O, Naik N, et al. Designing neural network architectures using reinforcement learning[J]. arXiv preprint arXiv:1611.02167, 2016.

[16]Lange S, Riedmiller M. Deep auto-encoder neural networks in reinforcement learning[A]. International Joint Conference on Neural Networks[C]. Barcelona: IEEE, 2010: 1-8.

[17]Riedmiller M. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method[J]. European Conference on Machine Learning, 2005, 3720: 317-328.

[18]Abtahi F, Fasel I. Deep belief nets as function approximators for reinforcement learning[J]. AAAI Conference on Lifelong Learning, 2011, 5(1): 2-7.

[19]Lange S, Riedmiller M, Voigtlander A. Autonomous reinforcement learning on raw visual input data in a real world application[J]. International Joint Conference on Neural Networks, 2012, 20: 1-8.

[20]Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[21]Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[A]. International Conference on Machine Learning[C]. New York: ACM, 2016: 1928-1937.

[22]Jaderberg M, Mnih V, Czarnecki W M, et al. Reinforcement learning with unsupervised auxiliary tasks[J]. arXiv preprint arXiv:1611.05397, 2016.

[23]Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

[24]Duan Y, Chen X, Houthooft R, et al. Benchmarking deep reinforcement learning for continuous control[A]. International Conference on Machine Learning[C]. New York: ACM, 2016: 1329-1338.

[25]Gu S, Lillicrap T, Sutskever I, et al. Continuous deep q-learning with model-based acceleration[A]. International Conference on Machine Learning[C]. New York: ACM, 2016: 2829-2838.

[26]Hansen S. Using deep q-learning to control optimization hyperparameters[J]. arXiv preprint arXiv:1602.04062, 2016.

[27]Narasimhan K, Kulkarni T, Barzilay R. Language understanding for text-based games using deep reinforcement learning[A]. Empirical Methods on Natural Language Processing[C]. Austin: ACL, 2015: 1-11.

[28]Guo H. Generating text with deep reinforcement learning[J]. arXiv preprint arXiv:1510.09202, 2015.

[29]Li J, Monroe W, Ritter A, et al. Deep reinforcement learning for dialogue generation[J]. arXiv preprint arXiv:1606.01541, 2016.

[30]El Sallab A, Abdou M, Perot E, et al. Deep reinforcement learning framework for autonomous driving[J]. Autonomous Vehicles and Machines, Electronic Imaging, 2017.

[31]Caicedo J C, Lazebnik S. Active object localization with deep reinforcement learning[A]. International Conference on Computer Vision[C]. Santiago: IEEE, 2015: 2488-2496.

[32]Lample G, Chaplot D S. Playing FPS games with deep reinforcement learning[A]. AAAI Conference on Artificial Intelligence[C]. Phoenix: AAAI, 2016: 2140-2146.

[33]Wu Y X, Tian Y D. Training agent for first-person shooter game with actor-critic curriculum learning[J]. International Conference on Learning Representations, 2017.

[34]Schulman J, Levine S, Moritz P, et al. Trust region policy optimization[A]. International Conference on Machine Learning[C]. Lille: ACM, 2015: 1889-1897.

[35]Zoph B, Le Q V. Neural architecture search with reinforcement learning[J]. arXiv preprint arXiv:1611.01578, 2016.

[36]Graves A, Wayne G, Reynolds M, et al. Hybrid computing using a neural network with dynamic external memory[J]. Nature, 2016, 538(7626): 471-476.

[37]Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning[J]. arXiv preprint arXiv:1609.05143, 2016.

[38]Littman M L. Reinforcement learning improves behaviour from evaluative feedback[J]. Nature, 2015, 521(7553): 445-451.

[39]Krakovsky M. Reinforcement renaissance[J]. Communications of the ACM, 2016, 59(8): 12-14.

[40]Van H V, Guez A, Silver D. Deep reinforcement learning with double q-learning[A]. AAAI Conference on Artificial Intelligence[C]. Phoenix: AAAI, 2016: 2094-2100.

[41]Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay[J]. International Conference on Learning Representations, 2016.

[42]Bellemare M G, Ostrovski G, Guez A, et al. Increasing the action gap: New operators for reinforcement learning[A]. AAAI Conference on Artificial Intelligence[C]. Phoenix: AAAI, 2016: 1476-1483.

[43]Hasselt H V, Guez A, Hessel M, et al. Learning functions across many orders of magnitudes[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016.

[44]He F S, Liu Y, Schwing A G, et al. Learning to play in a day: Faster deep reinforcement learning by optimality tightening[J]. arXiv preprint arXiv:1611.01606, 2016.

[45]Lakshminarayanan A S, Sharma S, Ravindran B. Dynamic frame skip deep Q network[J]. arXiv preprint arXiv:1605.05365, 2016.

[46]François-Lavet V, Fonteneau R, Ernst D. How to discount deep reinforcement learning: Towards new dynamic strategies[J]. arXiv preprint arXiv:1512.02011, 2015.

[47]Osband I, Blundell C, Pritzel A, et al. Deep exploration via bootstrapped DQN[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016: 4026-4034.

[48]Stadie B C, Levine S, Abbeel P. Incentivizing exploration in reinforcement learning

with deep predictive models[J]. arXiv preprint arXiv:1507.00814, 2015.

[49]Bellemare M G, Srinivasan S, Ostrovski G, et al. Unifying count-based exploration and intrinsic motivation[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016: 1471-1479.

[50]Munos R, Stepleton T, Harutyunyan A, et al. Safe and efficient off-policy reinforcement learning[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016: 1046-1054.

[51]Hausknecht M, Stone P. Deep recurrent q-learning for partially observable MDPs[J]. arXiv preprint arXiv:1507.06527, 2015.

[52]Foerster J N, Assael Y M, Freitas N D, et al. Learning to communicate to solve riddles with deep distributed recurrent q-networks[J]. arXiv preprint arXiv:1602.02672, 2016.

[53]Wang Z, Freitas N D, Lanctot M. Dueling network architectures for deep reinforcement learning[A]. International Conference on Machine Learning[C]. New York: ACM, 2016: 1995-2003.

[54]Oh J, Chockalingam V, Singh S, et al. Control of memory, active perception, and action in Minecraft[A]. International Conference on Machine Learning[C]. New York: ACM, 2016: 2790-2799.

[55]Rusu A A, Rabinowitz N C, Desjardins G, et al. Progressive neural networks[J]. arXiv preprint arXiv:1606.04671, 2016.

[56]Li X, Li L, Gao J, et al. Recurrent reinforcement learning: a hybrid approach[J]. arXiv preprint arXiv:1509.03044, 2015.

[57]Kulkarni T D, Narasimhan K, Saeedi A, et al. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016: 3675-3683.

[58]Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation[J]. arXiv preprint arXiv:1506.02438, 2015.

[59]Heess N, Wayne G, Silver D, et al. Learning continuous control policies by stochastic

value gradients[A]. Advances in Neural Information Processing Systems[C]. Montreal: MIT Press, 2015: 2944-2952.

[60]Wang Z, Bapst V, Heess N, et al. Sample efficient actor-critic with experience replay[J]. arXiv preprint arXiv:1611.01224, 2016.

[61]Peng X B, Berseth G, van de Panne M. Terrain-adaptive locomotion skills using deep reinforcement learning[J]. ACM Transactions on Graphics, 2016, 35(4): 1-12.

[62]Heess N, Hunt J J, Lillicrap T P, et al. Memory-based control with recurrent neural networks[J]. arXiv preprint arXiv:1512.04455, 2015.

[63]Hausknecht M, Stone P. Deep reinforcement learning in parameterized action space[J]. arXiv preprint arXiv:1511.04143, 2015.

[64]Parisotto E, Ba J L, Salakhutdinov R. Actor-mimic: deep multitask and transfer reinforcement learning[J]. International Conference on Learning Representations, 2016.

[65]Rusu A A, Colmenarejo S G, Gulcehre C, et al. Policy distillation[J]. arXiv preprint arXiv:1511.06295, 2015.

[66]Schaul T, Horgan D, Gregor K, et al. Universal value function approximators[A]. International Conference on Machine Learning[C]. Lille: ACM, 2015: 1312-1320.

[67]Fernando C, Banarse D, Blundell C, et al. PathNet: evolution channels gradient descent in super neural networks[J]. arXiv preprint arXiv:1701.08734, 2017.

[68]Krishnamurthy R, Lakshminarayanan A S, Kumar P, et al. Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks[J]. arXiv preprint arXiv:1605.05359, 2016.

[69]Kulkarni T D, Saeedi A, Gautam S, et al. Deep successor reinforcement learning[J]. arXiv preprint arXiv:1606.02396, 2016.

[70]Tampuu A, Matiisen T, Kodelja D, et al. Multi-Agent cooperation and competition with deep reinforcement learning[J]. arXiv preprint arXiv:1511.08779, 2015.

[71]Duan Y, Schulman J, Chen X, et al. RL $^ 2$: Fast reinforcement learning via slow reinforcement learning[J]. arXiv preprint arXiv:1611.02779, 2016.

[72]Blundell C, Uria B, Pritzel A, et al. Model-free episodic control[J]. arXiv preprint arXiv:1606.04460, 2016.

[73]Tamar A, Levine S, Abbeel P, et al. Value iteration networks[A]. Advances in Neural Information Processing Systems[C]. Barcelona: MIT Press, 2016: 2146-2154.

[74]Babaeizadeh M, Frosio I, Tyree S, et al. GA3C: GPU-based A3C for deep reinforcement learning[J]. arXiv preprint arXiv:1611.06256, 2016.

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

[76]O'Donoghue B, Munos R, Kavukcuoglu K, et al. PGQ: Combining policy gradient and Q-learning[J]. arXiv preprint arXiv:1611.01626, 2016.

[77]Gu S, Lillicrap T, Ghahramani Z, et al. Q-Prop: Sample-efficient policy gradient with an off-policy critic[J]. arXiv preprint arXiv:1611.02247, 2016.

[78]Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4): 212-223.

[79]Bellemare M G, Naddaf Y, Veness J, et al. The Arcade learning environment: An evaluation platform for general agents[J]. J. Artif. Intell. Res.(JAIR), 2013, 47: 253-279.

[80]Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735.

[81]Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[82]Mnih V, Heess N, Graves A. Recurrent models of visual attention[A]. Advances in Neural Information Processing Systems[C]. Montreal: MIT Press, 2014: 2204-2212.

[83]Xu K, Ba J, Kiros R, Courville A, Salakhutdinov R, Zemel R, Bengio Y. Show, attend and tell: Neural image caption generation with visual attention[A]. International Conference on Machine Learning[C]. Lille: ACM, 2015: 2048-2057.

[84]Donahue J, Anne H L, Guadarrama S, et al. Long-term recurrent convolutional

networks for visual recognition and description[A]. Computer Vision and Pattern Recognition[C]. Boston: IEEE, 2015: 2625-2634.

[85]Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[A]. International Conference on Machine Learning[C]. Beijing: ACM, 2014: 387-395.

[86]Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[A]. International Conference on Machine Learning[C]. Lille: ACM, 2015: 448-456.

[87]Kingma D, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[88]Kober J, Peters J R. Policy search for motor primitives in robotics[A]. Advances in Neural Information Processing Systems[C]. Vancouver: MIT Press, 2009: 171-203.

[89]Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[A]. Advances in Neural Information Processing Systems[C]. Montreal: MIT Press, 2014: 2672-2680.

[90]Graves A, Wayne G, Danihelka I. Neural turing machines[J]. arXiv preprint arXiv:1410.5401, 2014.

[91]Weston J, Chopra S, Bordes A. Memory networks[J]. arXiv preprint arXiv:1410.3916, 2014.

攻读硕士学位期间公开发表(录用)的论文及参与的项目

一、公开发表(录用)的学术论文

[1] Zhai J, Liu Q, Zhang Z, et al. Deep q-learning with prioritized sampling[A]. International Conference on Neural Information Processing[C]. Kyoto: Springer International Publishing, 2016.（CCF 推荐 C 类会议，已发表）

[2] 翟建伟, 刘全, 章宗长, 钟珊, 周倩, 章鹏. 一种基于视觉注意力机制的深度循环 Q 网络模型[J]. 计算机学报, 2016: 1-15.（已录用）

[3] 刘全, 翟建伟, 章宗长, 周倩, 章鹏, 徐进. 深度强化学习综述[J]. 计算机学报, 2017: 1-28.（已录用）

[4] Zhu H, Zhu F, Fu Y, et al. A Kernel-based sarsa (λ) algorithm with clustering-based sample sparsification[A]. International Conference on Neural Information Processing[C]. Kyoto: Springer International Publishing, 2016.（CCF 推荐 C 类会议，已发表）

[5] Sun C, Ling X, Fu Y, et al. Sparse Kernel-Based Least Squares Temporal Difference with Prioritized Sweeping[A]. International Conference on Neural Information Processing[C]. Kyoto: Springer International Publishing, 2016.（CCF 推荐 C 类会议，已发表）

[6] 章鹏, 刘全, 钟珊, 钱炜晟, 翟建伟. 连续空间中的一种动作加权行动者评论家算法[J]. 计算机学报, 2016: 1-14.（已录用）

[7] 章鹏, 刘全, 钟珊, 翟建伟, 钱炜晟. 增量式双自然策略梯度的行动者评论家算[J]. 通信学报, 2017.（已录用）

二、参加的科研项目

[1] 国家自然科学基金项目“基于模糊逻辑的大规模强化学习理论及方法”，项目编号：61472262.

[2] 国家自然科学基金青年项目“基于覆盖数的部分可观察不确定性规划理论及方

法”，项目编号：61502323.

[3] 国家自然科学基金项目“面向 tableau 模型的逻辑强化学习理论及方法研究”，项目编号：61070223.

[4] 省自然科学基金项目“基于覆盖数的合作多智能体规划方法研究”，项目编号：16KJB520041.

致谢

三年前,我通过考研进入苏大继续自己的学习生涯。时光飞逝，一转眼硕士三年已经接近尾声。回顾这三年时光，感悟良多，收获颇丰。在此，我要感谢这三年中给予我指导、帮助、鼓励与关怀的老师、同学以及家人。

首先，我要感谢我的导师刘全教授，谢谢您这三年对我学习和生活上的悉心指导。在校期间，您多次询问我的学术研究进展，并指导相关的研究内容。在我遇到科研上的难题时，努力帮助我开拓研究思路，指明解决问题的方向。另外，刘老师定期开展的强化学习讨论班，营造了一种良好的学术氛围，为课题组成员的科研之路打下了夯实的基础，并加强了整个课题组老师和同学之间的凝聚力。刘老师严谨的治学态度，一丝不苟的工作作风，深厚的学术造诣一直使我受益匪浅。谢谢您平时对我们的严格要求，才使得我们在这三年里有所收获、有所提高。在此，特别向刘老师表以最诚挚的感激。

其次，我要感谢学科组的各位老师，谢谢你们平时在学习和生活上对我的指导和帮助。特别地，我要感谢章宗长老师在我论文撰写与投稿期间对我的悉心指导。

接着，我要感谢与我同届的章鹏、钱炜晟、朱海军和孙慈嘉同学陪我度过了人生中最充实的三年时光，谢谢你们平时对我的关心和帮助。我还要感谢钟珊、尤树华、施梦宇，许志鹏，周谊成，许丹，陈仕超等师兄师姐们，以及徐进、梁斌、周倩等师弟师妹们，是你们在科研和生活中给予了我很大的帮助。

然后，我要感谢我的家人和朋友，特别是我的父母。在我求学的三年里，你们竭尽所能为我提供物质和精神上的支持，并一直无私地鼓励和陪伴我。你们对我无私的爱和奉献，伴随着我成长的每一阶段，使我时刻保持着进取和感恩的心。

最后，感谢所有在百忙之中抽出宝贵时间审阅论文的各位老师；感谢所有在百忙之中参加我论文答辩的老师们，在此致以我崇高的敬意和衷心的感谢。

苏州大学
硕士学位论文
（学术学位）

苏州大学研究生院统一印制