硕士学位论文

基于深度强化学习的非完备信息机器博弈研究

RESEARCH ON IMPERFECT INFORMATION MACHINE GAME BASED ON DEEP REINFORCEMENT LEARNING

王鹏程

哈尔滨工业大学

2016年12月

国内图书分类号：TP391.4　　学校代码：10213
国际图书分类号：621.3　　密级：公开

工学硕士学位论文

基于深度强化学习的非完备信息机器博弈研究

硕士研究生：王鹏程

导　　师：王轩教授

申请学位：工学硕士

学　　科：计算机科学与技术

所在单位：深圳研究生院

答辩日期：2016年12月

授予学位单位：哈尔滨工业大学

Classified Index: TP391.4
U.D.C: 621.3

Dissertation for the Master Degree in Engineering

RESEARCH ON IMPERFECT INFORMATION MACHINE GAME BASED ON DEEP REINFORCEMENT LEARNING

Candidate: Pengcheng Wang
Supervisor: Prof. Xuan Wang
Academic Degree Applied for: Master Degree in Engineering
Speciality: Computer Science and Technology
Affiliation: Shenzhen Graduate School
Date of Defence: December, 2016
Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

从人工智能这一概念被提出至今，机器博弈一直是其最具挑战性的研究方向之一，机器博弈又分为完备信息机器博弈和非完备信息机器博弈。非完备信息机器博弈的特点是智能体在博弈过程中无法获得全部的局面信息。真实世界的很多决策问题都可以抽象为非完备信息博弈问题，例如机场规划、网络安全、金融能源等问题。因此，对非完备信息机器博弈的研究具有重要的现实意义。

解决非完备信息机器博弈问题的传统方法是部分可观测马尔科夫决策过程模型以及强化学习算法。然而强化学习在信息不完备以及高维的状态空间下无法收敛，仅通过有限的数据和反复测试无法遍历到所有状态。本论文采用深度强化学习算法解决非完备信息机器博弈问题，用深度学习网络替换强化学习中的状态-动作值函数。同时针对深度强化学习算法决策时无法考虑历史信息的问题，提出在深度强化学习网络中加入长短期记忆模型。

本论文提出了基于蒙特卡洛博弈树搜索的回报函数计算方法，通过比较每局得到的收益与蒙特卡洛博弈树搜索得到的期望收益，判断对智能体的奖惩情况。

传统方法需要手动提取特征，很难发现特征间的内在联系，同时训练时需要大量的领域知识，可扩展性差。本论文提出了适合于深度强化学习等模式匹配算法的扑克建模方法，这种编码方式只需要很少的领域知识便可以将相同的网络结构应用于不同的扑克游戏，实现了很好的可扩展性。

最后本论文将改进的深度强化学习算法应用到非完备信息机器博弈中，实现了德州扑克机器博弈系统。从感知到动作端对端地学习策略，避免了复杂的人工提取特征的过程，与传统的学习算法相比，达到了更高的博弈水平。改进的深度强化学习为大规模机器博弈系统的实现提供了一个可行的方法，同时为扩展到现实生活中提供了可能。

关键词：深度强化学习；非完备信息机器博弈；长短期记忆模型；扑克建模

Abstract

Since the concept of artificial intelligence has been proposed, the machine game has been one of its most challenging research directions . Machine game can be divided into perfect information machine game and imperfect information machine game. The characteristic of imperfect information machine game is that agent can't get all information in the game process. Many real-world decision-making problems can be abstracted as imperfect information game problems, such as airport planning, network security, financial energy and other problems. Therefore, it is great practical significance to study the imperfect information machine game.

The traditional method of solving the imperfect game machine game problem is partially observed Markov decision process model and reinforcement learning algorithm. However, the reinforcement learning algorithm can't guarantee convergence in imperfect state and high latitude state space. Only through limited data and repeated testing can't traverse all the state. In this paper, deep reinforcement learning algorithm is used to solve the game of imperfect information machine, and the state-action value function in reinforcement learning is replaced by a deep learning network. Aiming at the problem that historical information can't be considered in the decision-making process of deep reinforcement learning algorithm, we propose to add the long-short term memory model to the deep reinforcement learning algorithm.

In this paper, a reward function based on Monte Carlo tree search is proposed. By comparing the return of the game and expected reward of the Monte Carlo game tree search, we can judge whether the agent should be rewarded or be punished.

Traditional methods need to extract features manually. It is difficult to find the internal relations between features. Besides, training requires a lot of domain knowledge, which makes poorly scalability. This paper proposes a poker modeling method, which is suitable for pattern matching algorithms such as deep reinforcement learning. This coding method can apply the same network structure to different poker games with very little domain knowledge.

Finally, this paper applies the improved deep reinforcement learning algorithm to the Texas poker game system. Learning from end-to-end avoids the complex process of extracting features manually. Comparing with the traditional reinforcement learning algorithm, it can achieve a higher level of intelligent. Improved deep reinforcement learning can provide a feasible method for the realization of large-scale machine game system and provide the possibility for

extending to real life.

Keywords: deep reinforcement learning, imperfect information machine game, long-short-term memory model, poker modeling

目　录

第1章 绪　论

1.1 课题研究背景和意义

1.1.1 课题研究背景

人类成长的过程是一个不断学习和思考的过程，大脑作为其核心汇集了各种生活中的智慧，指导着人类在生活中做出合理的决策。假如机器也拥有了这样的学习能力，拥有了将看到、听到、学到的智慧分析存储，并能够使用正确的策略解决面临的问题，称机器具备了人工智能（Artificial Intelligence，AI）[1]。人工智能是一门极具挑战的学科，涉及的领域也十分广泛，它主要研究如何使计算机像人类一样思考、学习、行动，像人一样与世界进行合理且有意义的交互。虽然现在出现了各种各样的机器人，一些领域也开始将人工智能融入到他们的产品中，但是人工智能的总体水平还不高，而且往往只是瞄准其中的一部分内容，不具备通用性。因此，研究人工智能仍是一项富有挑战性的任务。

从人工智能这一概念被提出至今，机器博弈作为其重要方向之一，一直是研究者最想要攻破的难关，同时机器博弈也是检验人工智能研究水平的重要标准[2]。自从计算机诞生以来，有无数的研究者投身到这个富有挑战性的领域，包括，著名的计算机之父冯•诺依曼、算法和程序设计的先驱者唐纳德•克努特、人工智能之父艾伦•图灵等，这些计算机领域的先驱者都对机器博弈有过一定的研究。早在1952年，图灵就曾写过一个国际象棋机器博弈程序，可惜当时计算机没有足够的计算能力执行，他就通过纸笔模仿计算机与人类对战，虽然最后没有战胜人类，但是却是机器博弈的一个伟大的开端。

机器博弈分为完备信息机器博弈和非完备信息机器博弈，完备信息机器博弈是指智能体可以看到全部的游戏状态，不存在信息隐藏，例如围棋、象棋、五子棋等的机器博弈。非完备信息机器博弈是指智能体只能够看到自身的游戏状态以及一些公共的信息，无法掌握全部的局面信息，例如扑克机器博弈中，对智能体而言对手的手牌是不可见，因此获得的局面信息是不完备的。根据游戏的下一时刻状态是否完全由当前状态和博弈玩家的行为所决定，又可把机器博弈分为确定性机器博弈和非确定性机器博弈。像德州扑克机器博弈中发公共牌是一个随机的过程，不受博弈双方的影响和控制，因此德州扑克机器博弈是一种非确定性机器博弈。而围棋机器博弈就是一个典型的确定性机器博弈，下一时刻的博弈局面完全由玩家的行为以及当前的棋子分布决定。

棋牌类游戏是人们生活中休闲娱乐的方式之一。扑克游戏包括很多变种，

例如德州扑克、Kuhn扑克、斗地主等。在扑克游戏中，玩家无法观测到全部的局面信息，同时发牌也是一个不受玩家影响的随机过程，因此扑克游戏既具有非完备性也具有非确定性，这给游戏带来了更多的挑战性，使得玩家不断的在游戏的过程中学习新的策略，例如在德州扑克中玩家可以通过经验的积累，分析当前局面获胜的可能性而选择加注或者弃牌，同时玩家还可以分析对手的弱点，选择欺诈的策略来赢得更多的筹码。世界扑克大赛（World Series of Poker, WSOP）自1970年开始举办已有40多年的历史，每年春夏在拉斯维加斯的各大赌场举办，吸引了世界上成千上万的扑克爱好者参与其中。

1.1.2 研究的目的和意义

博弈问题不仅存在于生活中的各个方面，甚至军事、经济等国家层面也有着广泛的应用。博弈问题往往可以转化为游戏中的模型，历史上，游戏一直推动着人工智能和机器博弈的进步（Samuel，1959；Tesauro，1995[3]；Riedmiller等人，2009[4]；Gelly 等人，2012[5]；Bowling 等人，2015[6] ；Silver等人，2016[7]）。游戏理论将游戏定义为一个冲突区域或者多方的合作，而只要双方存在某些方面的冲突，博弈问题也就随之而来。之所以学习比较简单的博弈游戏，其中一个目的是通过简单的模型研究新的算法，可以扩展到更加复杂的真实世界，例如机场和网络安全、金融、交通管制和疏导等（Lambert III 等人，2005[8]；Nevmyvaka 等人，2006[9]；Bazzan，2009[10]；Tambe，2011[11]；Urieli & Stone，2014[12]； Durkota 等人， 2015[13]）。大部分这些真实世界问题都是基于非完备的信息以及高维度的信息状态空间，无法获知对手的情况，也无法对全局信息进行精准的把握和预知，然而，许多已经应用到经典问题中的机器学习方法，在信息不完备的游戏中缺少收敛的保证。另一方面，许多传统理论方法缺少抽取相关模式、并从数据中概况的能力。这让传统算法的可扩展性有限，除非使用人类专家知识、启发式方法和建模来将该领域抽象化至可控的规模。然而，获取人类专业知识经常需要昂贵的资源和时间。此外，人类很容易出现非理性的决策或者假设（Selten，1990[14]；Ariely，2008[15]）。这让我们希望开发新的算法，端到端地学习有用的策略。深度强化学习算法（Deep Reinforcement Learning，DRL）[16]是将深度学习与强化学习结合起来从而实现从感知到动作的端对端学习的一种全新的算法，简单的说，就是和人类一样，输入感知信息比如视觉，然后通过改进的深度神经网络，直接输出动作，中间没有人工处理的过程。深度强化学习利用深度学习网络克服了高维度状态空间的问题，避免了复杂的人工提取特征的过程，同时利用强化学习的特性无需像监督学习一样大量的训练数据，实现了自我学习，具备了使智能体实现完全自主的学习一种

甚至多种技能的潜力。因此如何将深度强化学习应用到非完备信息博弈中，对于机器博弈水平的提升以及人工智能的发展有着深远的意义。

世界年度计算机扑克大赛（Annual Computer Porker Competition， ACPC）为扑克类非完备信息机器博弈的研究者提供了一个权威的评测平台，每年年初由人工智能国际联合会议（International Joint Conference on Artificial Intelligence，IJCAI）和美国人工智能协会（American Association For Artificial Intelligence，AAAI）主办的会议（AAAI Conference on Artificial Intelligence）联合举办。比赛吸引了包括阿尔伯特大学大学、卡内基梅隆大学、伦敦大学、麻省理工大学等国际知名高校以及一些独立研究机构的研究者参与。因此，本论文选择扑克游戏作为非完备信息机器博弈的研究对象。

1.2 国内外相关研究及综述

国际象棋[17]和西洋双陆棋[18]是机器博弈中最早开始研究的。20世纪80年代，Berliner研发的西洋双陆棋智能体打败了当时的世界冠军Villia[18]。1997年，IBM经过多年国际象棋的研究推出超级计算机“深蓝”，击败了人类国际象棋世界冠军卡斯帕罗夫[19]，打破了人类在国际象棋项目上不可战胜的神话，也标志了在完备信息博弈中取得重大突破。非完备信息机器博弈方面，研究最多的是德州扑克[20,21]。1999年，Billings等人开发了第一个德州扑克机器博弈程序[22]，虽然是基于规则的，但为后来的研究奠定了基础。随后几年越来越多的研究者投入到对德州扑克等扑克游戏的研究中，并在在对手建模、纳什均衡、后悔值最小化等方面取得了优秀的成果。2006年Billings成功把对手建模和博弈树搜索应用到德州扑克中[23]；2007年，Zinkevich等人提出了虚拟遗憾最小化算法（Counterfactual Regret Minimization，CFR）[24]，在两人零和博弈游戏中达到了近似纳什均衡； 2015年，Bowling等人在《Science》上发表文章，证明其改进后的CFR+算法已经完全解决两人限制性德州扑克[6]，但是CFR算法无法解决规模超过10^{18}的博弈问题，且为了达到纳什均衡需要大量的领域知识。

强化学习模拟人类在与环境交互中不断学习的过程，它是一种自学习算法，也是解决机器博弈问题常用的算法之一。Minsky在20世纪50年代第一次提出了强化学习的概念，源于对人类学习的研究。1957年，Bellman提出了经典的马尔科夫决策模型[25]以及动态规划方法。20世纪80年代，Watkins提出了Q学习算法[26]，强化学习迎来了新的一轮研究热潮。1992年，Tesauro利用强化学习算法中的$TD(\lambda)$算法开发的西洋跳棋程序TD-Gammon[27]具有打败人类顶尖大师的智能体水平。21世纪初，强化学习已经广泛的应用于游戏博弈、机器人控制、调度管理等方面。2001年，Dahl首次将Q学习应用在二人德州扑克游戏中，其智

能体程序通过学习可以应对特定类型的对手[28]。2009年Van den Broeck等人在无限注德州扑克玩法上引入了蒙特卡罗方法[29,30]，自此，蒙特卡罗博弈树搜索算法开始普遍应用。2012年Passos等人再次在德州扑克中应用强化学习算法，并结合特定的对手模型，取得了不错的效果[31]。小规模的机器博弈问题已经得到了很好的解决，但是围棋、扑克等拥有高维状态空间的博弈问题依然难以得到解决。2006年，Hinton等人将一篇关于深度学习的文章发表在《Science》上[32]，掀起了研究者对深度学习的研究热潮；2012年，在ImageNet评测上采用深度学习训练的智能体将图片识别的错误率从26%降低到了惊人的15%；2013年，谷歌DeepMind团队第一次提出了深度强化学习算法（Deep Reinforcement Learning，DRL）[16]，将强化学习与深度学习相结合，从纯图像输入，智能体完全通过自我学习来玩Atari游戏；2015年，谷歌DeepMind团队在《Nature》上发表文章，使用深度强化学习算法在Atari游戏上实现了人类水平的控制[33]。深度强化学习从此成为强化学习和深度学习两个领域的前沿研究方向。2016年，谷歌DeepMind团队研发的基于蒙特卡洛博弈树搜索和深度强化学习算法的围棋机器博弈智能体Alpha Go[7]在与世界围棋冠军韩国职业九段棋手李世石（Lee Sedol）的对弈中以4:1胜出，引起社会的广泛关注；Hinton，Bengio及Lecun在《Nature》上发表的深度学习综述一文最后也将深度强化学习作为未来深度学习未来的发展方向[34]。但是传统的深度强化学习算法本身无法处理需要考虑长时记忆的问题。

扑克建模方面，传统的算法都是人工提取游戏中的特征，需要大量的领域知识，将算法应用到相似的游戏中需要重新对算法进行设计，可扩展性差，同时人的某些固定思维有时候也会使结果有所偏向，无法达到一个均衡的状态，容易被对手找到弱点。2011年Lu ś Filipe Te ófilo首次在对德州扑克的研究上应用模式匹配算法[35]，给扑克游戏研究注入了新鲜的血液。2016年，Yakovenko等人将卷积神经网络应用到扑克中[36]，提出了监督学习的方式处理扑克问题，但是训练时需要大量的数据且无法在与不同的对手博弈过程中实时调整自己的策略。

在国内，机器博弈也有了一定的进展，2006年后，为了吸引更多研究者投入到机器博弈的研究中，中国人工智能学会每年都会举办一次计算机博弈大赛。中国象棋机器博弈最早开展研究的学校是东北大学，其开发的“棋天大圣”战胜了许银川等中国象棋顶尖高手；对于围棋的研究工作以北京邮电大学的研究成果最为显著，通过改进的UCT算法以及触发静态搜索等算法实现了自适应的围棋机器博弈[37]。

然而，国内研究者对非完备信息博弈的研究非常少，更多的集中在对完备

信息博弈的研究。哈尔滨工业大学深圳研究生院计算机应用研究中心从2003年开始对四国军棋机器博弈进行研究。从2012年起，增加了对德州扑克机器博弈的研究，并从2013年开始参加AAAI举办的年度计算机扑克大赛，获得2013年多人德州扑克项目第三名，2014年三人Kuhn 扑克项目第四名、多人德州扑克项目第四名、2016年二人非限制性德州扑克Uncapped Total Bankroll Results 榜第四名。主要研究成果有非完备信息机器博弈系统的信息表示[38]、启发函数优化算法[39]、大规模博弈树搜索[40] 、对手建模[41]、风险模型分析[42]等。其中的一些研究方法中需要用到状态-动作值函数，而受到非完备信息机器博弈高维状态空间的影响限制了算法的性能。因此需要辅以一些新的方法使得这些算法能够更好的应用到非完备信息机器博弈中。

1.3 主要研究内容及组织结构

1.3.1 主要研究内容

本论文以非完备信息机器博弈为研究对象，采用改进的深度强化学习算法克服传统算法在解决非完备信息机器博弈面临的瓶颈，本论文主要研究内容包括以下几方面：

（1）充分调研国内外相关领域研究现状，分析传统的解决非完备信息机器博弈算法存在的缺陷。

（2）针对传统强化学习算法无法处理高维状态空间的问题，提出在非完备信息机器博弈中应用深度强化学习算法，用深度学习网络替代强化学习中的状态-动作值函数。

（3）分析深度强化学习应用于非完备信息机器博弈的瓶颈：做决策时无法考虑过去的历史动作及状态。而在非完备信息机器博弈中对手之前的动作意图也是决策时必须加以考虑研究的重要因素。针对上述问题提出在深度学习部分加入长短期记忆模型。

（4）非完备信息机器博弈中，有限的时间和资源使得无法扩展整棵博弈树，所以最优收益未知的情况下不能仅仅将游戏结果作为回报值。针对上述问题提出基于蒙特卡洛树搜索的回报函数。

（5）提出适合于深度学习等模式匹配算法的扑克建模方法，使得仅需要很少的领域知识便可以将相同的网络结构应用于不同的扑克游戏，达到很好的可扩展性。

（6）将基于改进的深度强化学习算法应用在非完备信息机器博弈上，实现能够自我学习的德州扑克博弈智能体，并且效果相对传统算法有所提升。

1.3.2 论文的组织结构

论文的结构主要分四章，每章的具体内容如下：

第1章阐述了人工智能和机器博弈的相关背景、研究目的和意义。具体分析了国内外解决非完备信息机器博弈问题的传统算法以及深度强化学习的研究状况。最后概括了本论文主要的研究内容及论文结构。

第2章对非完备信息博弈中面临的问题进行了分析，介绍了博弈树搜索以及机器博弈中的估值函数，最后介绍了部分可观测马尔科夫决策过程以及如何把非完备信息博弈转换为POMDP模型。

第3章深入探究了深度强化学习算法，并针对在非完备信息机器博弈应用深度强化学习算法时无法考虑过去历史的问题加以解决，最后提出了基于蒙特卡洛博弈树搜索的回报函数以及深度强化学习中扑克游戏的模式匹配建模方法。

第4章是实验与结果分析，详细介绍了基于改进的深度强化学习算法实现的德州扑克博弈系统，主要包括系统架构、训练方式以及网络结构三个方面，最后对实验结果进行了分析。

第2章 非完备信息机器博弈

2.1 非完备信息游戏中的博弈问题

非完备信息博弈是指游戏中存在信息隐藏，玩家只能观测到游戏中部分状态的一类博弈游戏，例如，德州扑克、四国军棋、麻将等。非完备信息博弈中面临的困难主要有以下几点：

（1）非完备信息性 不同于象棋、围棋这种完备信息博弈可以看到全部的游戏状态，非完备信息博弈中玩家只知道自身的状态以及一些公共信息，故游戏中每一个玩家看到的游戏局面都是不同的，所以不能像完备信息博弈一样根据当前的局面推测对面玩家的最优策略而进行应对，这给游戏中多了许多的未知性。同时这也给游戏双方提供了施展更多策略的可能，例如，玩家在德州扑克中可以采用欺诈的策略，在牌力很小的情况下选择加注而致使对手弃牌。不同的策略体现了游戏双方的博弈水平。

（2）非确定性 非确定性是指博弈的进程不完全由博弈玩家决定，外部的随机因素会影响博弈的进程。像德州扑克、麻将这类游戏中发牌是一个随机的过程，它在很大程度上影响了博弈的最终结果。因此对于这种非确定性的博弈，玩家的目标便是获胜或止损，也就是在随机事件对自己不利、有可能导致输掉游戏的情况下尽可能令自己的损失最小化，而当随机事件对自己有利的情况之下尽可能最大化自己的收益。

（3）变化性 在非完备信息博弈游戏中，玩家获得的信息并不是一成不变的而会随着游戏的进行发生变化，例如德州扑克的不同阶段，牌桌上可见的公共牌的数量逐渐增多，玩家能够获知的信息也相应增多，因此需要不断的对掌握的信息进行调整，重新对当前局面做出判断。

（4）多人游戏 两人博弈游戏是机器博弈早期的研究重点，例如国际象棋，围棋等，随着计算机计算能力的提升以及一些创新算法的提出，研究人员逐渐将研究重点转移到对多人游戏上，但是随着人数的增加，游戏的状态空间也呈指数型增长。

2.2 博弈树搜索

博弈游戏根据博弈玩家采取动作的时序性不同又分为范式博弈（Normal Form Game）和扩展式博弈（Extensive Form Game）。剪刀包袱锤游戏是典型的范式博弈，游戏玩家同时做出动作，而即使玩家的动作有先后，后手也无法在

游戏开始前获知先手的动作。扑克游戏是典型的扩展式博弈，游戏玩家轮流做动作，且后手玩家能够知道先手玩家所做的动作。

大多数非完备信息博弈也都属于扩展式博弈，本论文主要对扩展式博弈进行研究。扩展式博弈常用博弈树来表示，以德州扑克为例，某一游戏状态下的博弈树如图2-1所示，其中博弈树中的每个节点都表示当前游戏中的可能局面状态，博弈树中的边表示游戏的合法动作，根节点下的子节点表示从根节点状态采取策略动作后到达的新的局面节点，叶节点表示游戏结束状态，这里假设玩家A的牌值大于玩家B的牌值，因此叶子节点中的值分别是玩家A的收益值与玩家B的收益值。

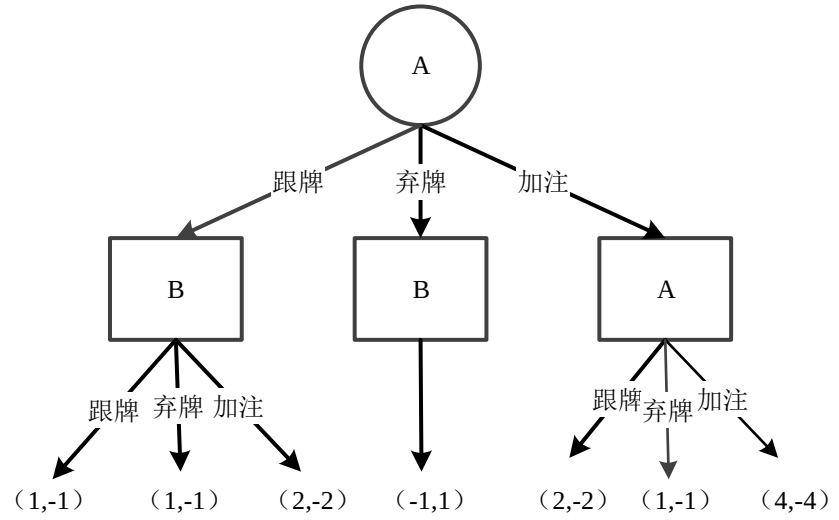

图2-1 德州扑克中某一状态下的博弈树

博弈树搜索是将玩家对弈的整个过程通过博弈树的形式展开来表示。极小极大搜索算法[43]是博弈树搜索的基本算法，在两人游戏中，假设博弈双方分别是A和B，该算法是一个零和算法，即如果一个策略能使A获得更大的收益，则会使B失去更多的筹码，反之亦然，因此玩家A和B都会尽力选择使自己获得最大收益的动作，在博弈树中体现为玩家A从可选择的动作中选择能够获得最大收益的动作，玩家B选择能使A获得最少收益的动作，即一个最大最小的选择过程。

游戏开始建立一个根节点，随着游戏的进行博弈树不断的扩展，最后会建立一棵很深的博弈树。如果把整个博弈树都展开，不仅扩展的时间复杂度很高，而且这棵博弈树也会非常庞大。因此常见的做法而并不是全部展开，而是仅仅扩展一定的层数，同时由于扩展不到代表游戏结束状态的叶子节点，引入估值函数来做为扩展到最后的回报值。图2-2显示了极小极大搜索两层的博弈树，扩展的叶子节点是估值函数给出的一个回报值，代表了玩家A能获得的收益，自底向上，首先玩家B会选择会使自己获得最大收益的动作，即使A获得的回报值

最小的行为，从图2-2中可以看到玩家B从9、-6、0，这三个回报值选择了最小的回报值-6。而玩家A会选择能获得最大收益的动作，从图2-2中可以看到玩家A从-6、4、-3，中选择了最大的回报值4。这就是极小极大搜索的基本思想。

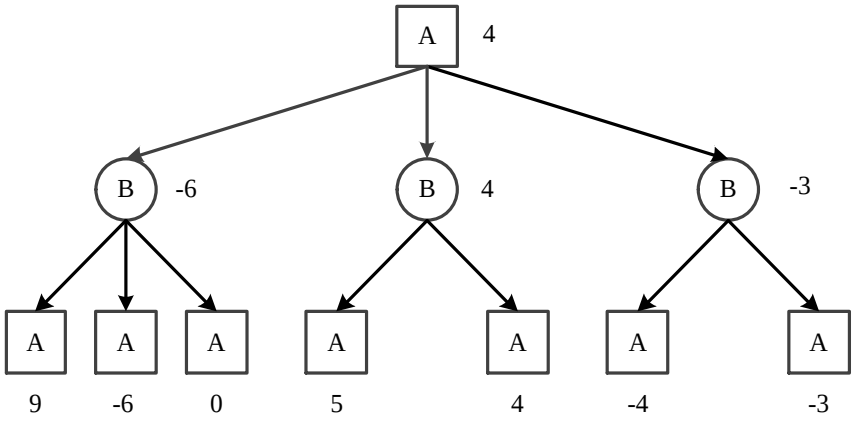

图2-2 极小极大搜索博弈树

2.3 估值函数

估值函数是对游戏局面进行估值，根据估值的结果判断应该选择的策略。估值函数的好坏直接决定了博弈水平的高低，因此在机器博弈中估值函数是研究者重点研究的内容。

2.3.1 静态估值函数

静态估值函数是指仅仅根据当前局面进行估值，不考虑时间序列以及对手策略等问题。静态估值是传统的机器博弈中最常用的方法。例如，在德州扑克中，Bill提出了在Preflop阶段的静态估值函数[44]：

（1）首先赋予每张牌一个分值。例如，A~J的分值依次为10~6分，其他的牌的分值为牌值的一半。手牌估值开始为手牌中牌值较大的牌的分值。

（2）如果两张手牌的牌值相同，手牌估值变为之前的两倍；

（3）如果两张手牌的花色相同，手牌估值加上2分；

（4）考虑手牌是否连续，手牌估值减去相应的差值；

那么，静态估值函数的计算公式见式（2-1）：

$$V = \sum f_i(M) \tag{2-1}$$

式中的 M 表示当前的局面， f_i 为上面每一个条件的计算函数。

2.3.2 动态估值函数

静态估值函数仅适用于简单的博弈游戏，在复杂的博弈游戏中很难达到很

好的效果。例如四国军棋中，对手前几步移动的一个棋子可能对当前局面有很大影响，决策时要综合当前局面与过去的历史信息进行考虑，因此传统的静态估值函数仅考虑当前局面存在较大误差，同时，针对不同的对手往往需要采用不同的估值函数。

人类对事物的评价也不是一成不变的，随着环境的改变以及时间的推移不断的变化着。与人类相似，智能体也需要在与环境交互中，不断地积累，不断地更新估值函数。动态估值函数的更新是一个学习的过程，它的设计主要从以下几个方面进行考虑：

（1）机器博弈游戏里的状态空间数量级大部分都在10^{30}以上，智能体需要在通过有限的数据对所有状态进行正确的评估。

（2）机器博弈游戏每一局都是一个时间序列，而且每一个时刻的状态都存在着一定的联系，智能体决策时需要根据当前局面以及之前的时间节点综合判断。在时间序列上学习新的估值函数。

（3）机器博弈游戏中，对手不一定采用最优的策略，所以针对不同的对手策略，智能体需要在博弈过程中不断地调整估值函数。

2.4 部分可观测马尔科夫决策过程

部分可观测马尔可夫过程（Partially Observable Markov Decision Processes，POMDP）是马尔可夫决策过程（Markov Decision Processes，MDP）的扩展。MDP是智能体根据当前实际环境状态做出自己的策略，但是大部分情况下，环境的实际状态难以获知，POMDP的提出就是为了解决这个问题。POMDP涉及的领域非常广泛，例如科技中的机器人控制、军事中的目标追踪、医疗中的诊断等。

POMDP可以用一个七元组（S，A，Ω，T，P，R，γ）来表示。

（1）$S=(s_1,s_2,...,s_n)$是所有状态的集合；

（2）$A=(a_1,a_2,...,a_n)$是智能体可采取的所有动作集合；

（3）$\Omega=(o_1,o_2,...,o_n)$表示智能体能观测到的所有观测状态；

（4）T是状态空间的转移函数；

（5）状态转移概率集合$P_{ss'}^{a}$表示学习智能体采取动作行为a从状态s转移到状态s'的概率；

（6）回报集合$R_{ss'}^{a}$表示以概率$P_{ss'}^{a}$从状态 s转变到状态 s' 后得到的即时回报。

（7）γ是折扣因子。

非完备信息博弈实际上可以当作在一棵博弈树上进行状态转移。例如在德州扑克中分为玩家结点、对手结点、随机结点和叶子结点。其中，玩家结点和对手结点代表的玩家双方的博弈，他们都可以采取弃牌、跟注、加注三个动作行为。随机结点指博弈过程中出现的随机过程，它不受双方玩家的影响。例如德州扑克游戏中的公共牌发牌过程，公共牌是由发牌器随机生成的，不受玩家的控制。叶子结点表示本局游戏结束。对于每局比赛，AI智能体都可以通过上述四个结点描述在博弈树间进行扩展。例如，通过随机节点发牌到达玩家节点，玩家节点通过采取动作到达对手节点，对手节点采取动作到达叶子节点、随机节点或者玩家节点，如此往复直到扩展到叶子节点。图2-3显示了二人德州扑克的部分博弈树。

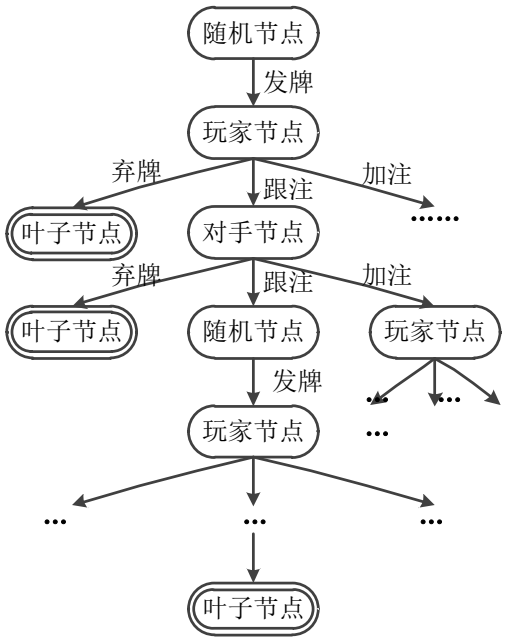

图2-3 二人德州扑克的部分博弈树

如果把每一局转移合在一起可以抽象成POMDP模型的转移。二人德州扑克的博弈树的POMDP模型如图2-4所示。

POMDP和非完备信息博弈都属于在时间序列上做决策的模型，在该模型中，环境的状态是不能完全识别，动作的回报也不能立即获得。

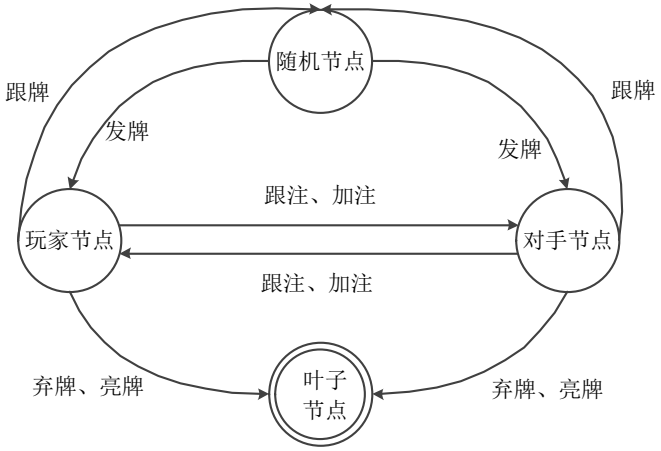

图2-4 二人德州扑克下的POMDP决策过程

2.5 本章小结

本章首先对非完备信息博弈中面临的问题进行了分析，接着介绍了博弈树搜索以及博弈中的估值函数，最后介绍了部分可观测马尔科夫决策过程以及如何把非完备信息博弈转换为POMDP模型。

第3章 基于深度强化学习的非完备信息机器博弈

3.1 深度强化学习

3.1.1 强化学习

人类成长的过程是一个不断探索，不断学习的过程，人类学会的生活中的技巧，更多的是通过自学获得的。例如：人类小时候学习投篮时，并不是一开始就掌握了如何控制投篮的力度、角度以及正确的投篮姿势，而是在不断投篮的过程中摸索其中的技巧。当投中时会想继续保持这样的姿势力度，当没有投中时会去调整自己的发力、投篮的角度等。通过这样一个不断试错，不断学习的过程，慢慢就学会了如何投篮。这是生活中的强化学习的例子，本质是智能体在与环境的交互中不断的探索，不断的学习。

强化学习模拟人类在与环境交互中不断学习的过程，它是一种自学习算法，也是解决机器博弈问题常用的算法之一。Minsky在20世纪50年代第一次提出了强化学习的概念，源于对人类学习方式的研究。强化学习中，智能体与外部环境进行交互，其行为引起环境状态改变，根据一定的评价标准获得奖励或惩罚（也称强化信号），继而智能体根据强化信号调整自己的策略，不断重复这样的学习过程。强化学习的目标是优化智能体的行为策略，使之在当前状态下找到最佳的行为映射，获得指标最好的强化信号[45]。

强化学习模型过程属于马尔可夫决策过程，包括：

（1）行为集合 A：学习智能体可以采取的所有行为集合；

（1）状态集合 S：智能体可以到达的所有状态集合；

（3）状态转移概率集合 $P_{ss'}^{a}$：表示学习智能体采取动作行为 a 从状态 s 转移到状态 s' 的概率；

（3）回报集合 $R_{ss'}^{a}$：表示以概率 $P_{ss'}^{a}$ 从状态 s转变到状态 s' 后得到的即时回报。

智能体在当前状态 s 下采取 A 中的某个策略，引起环境的改变，获得相应的强化信号。例如图3-1中，学习智能体在 t 时刻采用动作 a_t，环境状态从 s 变成 s'，同时智能体得到一个环境反馈的强化信号 r。

强化学习的目标是优化智能体的行为策略，使之在当前状态下找到最佳的行为映射（即 S 和 A 之间的最好映射），这就是状态-动作值函数 [46]。

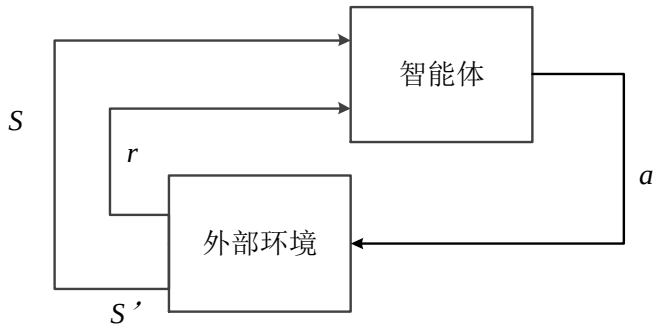

图3-1 强化学习模型图

$V^{\pi}(s)$ 为在策略 π 下的状态值函数，表示状态s下的期望回报，计算公式见式（3-1）：

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}\{r_{t+1} + \gamma r_{r+2} + \gamma^{2} r_{t+3} + \cdots \mid s_t = s\} \\ &= E_{\pi}\{r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\} \\ &= \sum_{a} \pi(s,a) \sum_{s'} P_{ss'}^{a}[R_{ss'}^{a} + \gamma V^{\pi}(s')] \end{aligned} \tag{3-1}$$

式中 E_{π} 表示学习智能体采取策略 π 时的期望回报，$\gamma \in [0,1]$ 是折扣因子，表示未来动作对期望回报的影响，γ 是一个大于0且小于1的数，表示智能体更关注最近时刻的回报，未来时刻的影响则相对较小。$\pi(s,a)$ 表示在策略 π 下状态s采取动作a的概率。

同样，可以定义在策略 π 下，处于状态s时，采用动作a的状态-动作值函数 $Q^{\pi}(s,a)$，其形式化定义如公式（3-2）所示：

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}\{R_t \mid s_t = s, a_t = a\} \\ &= \mathrm{E}_{\pi}\{r_{t+1} + \gamma r_{t+2} + \gamma r_{t+3} + \cdots \mid s,a\} \\ &= \mathrm{E}_{\pi}\{r_{t+1} + \gamma Q^{\pi}(s',a') \mid s,a\} \end{aligned} \tag{3-2}$$

强化学习的算法主要包括：动态规划算法，时序差分算法，蒙特卡洛算法，和Q学习算法。每个算法都有自己的局限性，例如，动态规划算法有科学理论和数学研究方面的支持，但是它使用的条件非常严格[47]；时序差分方法虽然不需要全部环境信息，但是收敛速度比较慢；蒙特卡洛算法需要对外界进行抽样，不能进行时序上的计算。Watkins于1989年提出来的Q学习算法是强化学习中一个重要的里程碑[48]。

Q学习算法是智能体在当前的状态下分析采取每一个动作所获得的回报来采取策略的一种算法。Q值的计算公式（3-3）如下：

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_{a} Q(s_{t+1},a) - Q(s_t,a_t)] \tag{3-3}$$

上述公式中，$Q(s_t,a_t)$ 表示智能体在状态为 s_t 时采取动作 a_t 时的状态-动作Q

值。r_{t+1}表示智能体到达状态s_{t+1}所获得的回报。α是学习速率，α越小收敛越慢，α越大收敛越快，但是会容易产生振荡。$r_{t+1}+\gamma \max Q(s_{t+1},a)$整体是Q学习的学习目标。图3-2是Q学习的回溯图。

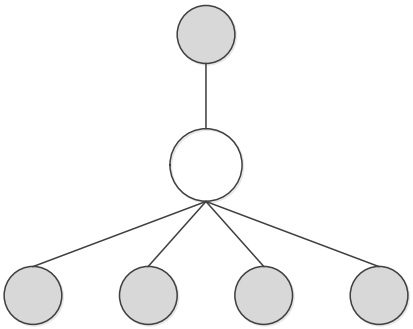

图3-2 Q学习的回溯图

Q学习算法在下面给出：

算法 3-1 Q 学习算法

初始化：所有状态s，行为a，Q值$Q(s,a)$

循环：执行下面操作，直到s_t是结束状态

（1）分析当前状态s_t，选择当前状态下的最佳动作策略a_t，执行策略a_t后到达下一个状态s_{t+1}；

（2）分析状态s_{t+1}，计算回报r_{t+1}；

（3）通过公式（3-3）更新Q值：

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha[r_{t+1}+\gamma \max Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \quad (0 \le \gamma < 1)$$

（4）更新状态$s_t \leftarrow s_{t+1}$；

Q学习是一种流行的偏离策略强化学习方法。在每一个状态下采取最高预估值的行动，因此Q学习实际训练维护了一个状态-动作矩阵，如图3-3所示：

$$Q=\begin{array}{c} \text{状态}\backslash\text{动作} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array}\begin{array}{c} \begin{array}{cccc} a_0 & a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & w_{0,3} \\ w_{1,0} & w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,0} & w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,0} & w_{3,1} & w_{3,2} & w_{3,3} \\ w_{4,0} & w_{4,1} & w_{4,2} & w_{4,3} \\ w_{5,0} & w_{5,1} & w_{5,2} & w_{5,3} \end{bmatrix} \end{array}$$

图3-3状态-动作值函数矩阵

表中的每一项w就是状态-动作对应的$Q(s,a)$值，处于不同的状态需要在这个表中选取使Q值最大的动作执行。

3.1.2 深度学习

人工神经网络是对人脑神经网络的计算模型的抽象，模拟人类的学习和思考方式，网络自身通常是对自然界某种函数或者算法的逼近，神经网络通过大量相互连接的神经元节点进行计算。

深度学习的概念源于Hinton等人提出的深度信念网（Deep Belief Networks，DBNs）[32]，是包含了多个隐藏层的人工神经网络，如图3-4所示，它通过组合低层特征形成更加抽象的高层表示，以发现数据特征间的内在联系，多层非线性结构使其具备了强大的特征表达能力。深度学习避免了复杂的人工提取特征的过程，而是隐式地从训练数据中进行学习，更能够刻画数据的丰富内在信息。同时大数据时代的到来以及硬件计算速度的提升使得越来越多的研究者开始将深度学习应用到不同的领域，如图像识别、自然语言处理、语音识别翻译等。深度学习常用的模型有自动编码器，卷积神经网络，递归神经网络等。

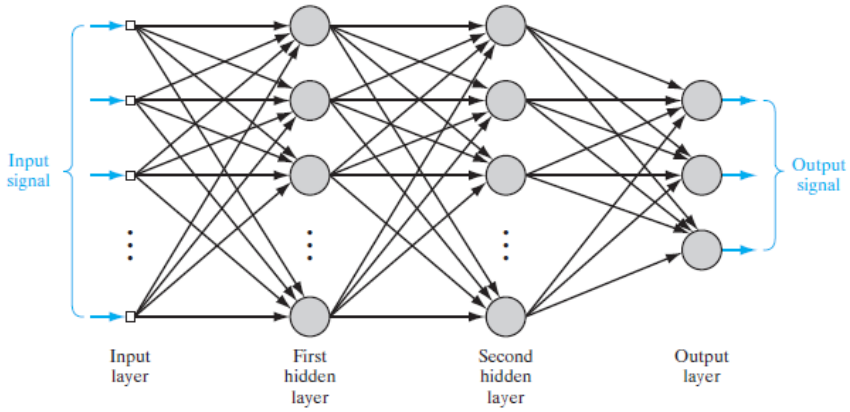

图3-4 深度学习网络

卷积神经网络（CNN）[49]源于对猫视觉皮层的研究，它是深层次网络模型的一种特殊形式。像大多数神经网络一样包含了几个隐藏层，其中每一层对输入向量应用仿射变换，随后是用激活函数进行非线性变换，而不同的地方是相邻两层的神经元之间并不是全连接，采用的是局部连接的方式，因为人类的认知是从局部到全局的，我们在对一个事物进行认知的时候开始的时候是看到了许多局部特征，将这些局部特征聚合后才有一个全局的认识。图像的空间联系也是局部的像素联系较为紧密，可以将局部特性当作一个特征来学习，而距离较远相关性则较弱，因此无需将它们直接联系在一起，所以只需局部连接然后在更高层将局部的信息综合起来得到全局信息，例如图3-5，如果采用全连接的方式，第m和第m-1层连接所需的权重数为$3\times5=15$，而采用局部连接的方式m层的每个神经元与m-1层的三个神经元相连，所需的权重数量为$3\times3=9$，这种结构将学习到的特征限定在局部空间里，可以使网络中权重数量大大降低，降低模

型整体的复杂度。同时因为图像的一部分的统计特性与其他部分是一样的，重复单元能够对特征进行识别，而无需考虑它在可视域中的位置，所以可以通过共享同一层的部分神经元的权值来进一步减少网络中的权重数量。例如图3-5，m-1到m层的相同颜色的连接线代表相同的权值，通过这样一个权值共享的方式使m与m-1层连接所需的权重数下降为3，进一步降低了卷积神经网络中的权重数量，同时权值共享使得卷积神经网络具有平移不变形，即对于输入中的平移变化，会以同样的距离和方向出现在特征图中。

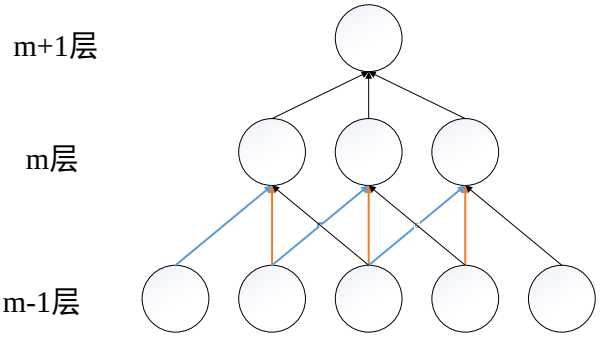

图3-5 局部连接示意图

卷积神经网络采用局部连接以及权值共享本质上是模拟动物的初级视皮层中的简单细胞，以卷积的方式在输入的每个位置提取局部特征。图像可以直接作为卷积层的输入，无需像传统算法一样要将大量的特征提取出来，避免了提取特征时的个人偏向，同时能够发现图像特征间的内在联系，提高了算法的准确度。因此卷积神经网络在计算机视觉方面达到了很好的效果，如图像分类、目标检测、人脸识别等，并且一些其他的领域也在将卷积神经网络引入到他们的研究中，例如自然语言处理、语音识别等。

3.1.3 深度强化学习算法

非完备信息博弈问题可以通过部分可观测马尔科夫过程模型来建模，然后利用强化学习算法来进行学习训练。但是，非完备信息博弈拥有高维的状态空间（例如Johanson 2013年的论文指出非限制德州扑克的状态空间达到7.16×10^{75}[50]）。如果状态空间或者动作空间特别大，那么状态-动作值函数$Q^{\pi}(s,a)$对应的矩阵就会非常大，一方面是计算机无法存储如此巨大的矩阵，另一方面没有足够的训练数据或者通过反复测试无法遍历到所有的情况，因此当智能体遇到新的情况时无法选出正确的动作。传统方法只适用于在较小规模博弈问题上。

针对上述存在的问题，需要对传统算法进行改进，联系到人类的行为特点，人类面对新的环境和事物，更多的是将新的情况与记忆进行比对，然后采用与过去相似的做法。因此，如果用一个线性或者非线性函数来表示估值函数，输

入任意的状态都能输出结果，就可以把状态-动作值函数矩阵的更新问题变成一个函数拟合问题，相近的状态也就可以得到相近的输出动作。这就是估值函数拟合：$Q(s,a\boldsymbol{w})\approx \hat{Q}(s,a,\boldsymbol{w})$，通过不断更新调整参数$\boldsymbol{w}$来逼近状态-动作值函数$Q^{\pi}(s,a)$。结合深度强化学习刻画高维状态的能力，我们使用深度神经网络作为强化学习的状态-动作估值函数估计，使用梯度下降算法代替强化学习的迭代更新，这就是深度强化学习的基本思想。其中强化学习最常用的是Q学习算法，深度学习网络的一般使用卷积神经网络。

深度Q学习算法（Deep Q-learning，DQN）就是将深度学习中的卷积神经网络和强化学习中的Q学习结合在一起。卷积神经网络可以直接将原始图像作为输入，避免了复杂的人工提取特征的过程。同时卷积神经网络能够从原始数据中提取到更加抽象、更加有代表性的特征。Q学习算法是强化学习中的一个重要里程碑，Q学习是对状态-动作值函数进行训练更新，不需要像其他强化学习算法一样知道环境的全部信息。

深度Q学习首先用一个卷积神经网络CNN替换状态-动作值函数，随机初始化网络的权重$\boldsymbol{w}$，如公式（3-4）所示：

$$Q(s,a,\boldsymbol{w})\approx Q^{\pi}(s,a) \tag{3-4}$$

将待训练的数据输入到深度Q学习网络中，经过卷积层、下采样层的操作以及激活函数的处理，提取到更加抽象的特征，得到每个动作对应的Q值，选择输出对应的Q值最大的动作$a=\max_a Q(s,a;\boldsymbol{w})$。因为初始阶段网络连接的权重是随机初始化的，所以开始时计算出的Q值最高的动作$a=\max_a Q(s,a;\boldsymbol{w})$也是完全随机的，智能体表现出的是随机的探索，当深度Q学习网络收敛时，随机探索的情况减少。

深度Q学习网络权重的更新使用均方差（mean-square error，MSE）来定义目标函数（objective function）也就是损失函数（loss function），如公式（3-5）所示：

$$L(\boldsymbol{w})=E[\underbrace{(r+\gamma\max_{a'}Q(s',a',\boldsymbol{w})}_{Target}-Q(s,a,\boldsymbol{w}))^2] \tag{3-5}$$

公式中s'是一时刻的状态，r是当前动作得到的回报值。可以看到，$r+\gamma\max_{a'}Q(s',a',\boldsymbol{w})-Q(s,a,\boldsymbol{w})$就是Q学习状态-动作值函数更新中计算出的$Q$值与旧矩阵中的$Q$值的差值，在Q学习中以一定的学习率来学习这个差值，而深度Q学习中则通过梯度下降来处理网络权重的更新。参数$\boldsymbol{w}$关于损失函数的梯度如公式（3-6）所示：

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = E[(r + \gamma \max_{a'} Q(s', a', \boldsymbol{w}) - Q(s, a, \boldsymbol{w})) \frac{\partial Q(s, a, \boldsymbol{w})}{\partial \boldsymbol{w}}] \tag{3-6}$$

梯度计算出来便可以沿着梯度下降的方向进行迭代，例如用随机梯度下降算法（Stochastic gradient descent，SGD)来最小化 $L(\boldsymbol{w})$。下面给出了在梯度下降过程中，深度强化学习网络中参数的更新公式如公式（3-7）所示：

$$\boldsymbol{w} = \boldsymbol{w} - \alpha \nabla_{w} L(\boldsymbol{w}) \tag{3-7}$$

其中 α 为学习率，代表梯度下降中的步长。图3-6显示了深度强化学习网络的网络结构。

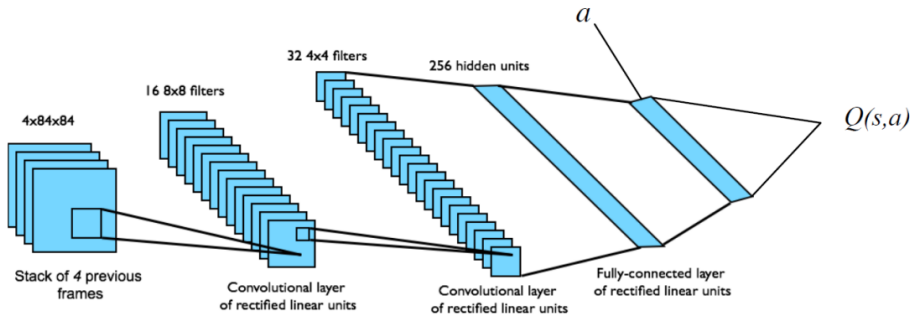

图3-6 深度强化学习网络

下面给出深度Q学习的算法流程：

算法 3-2 深度强化学习算法

初始化：初始化回放容器D，深度Q神经网络中神经元的权重随机赋值
循环：每一节片段episode
　　初始化观测序列 $s_1 = \{x_1\}$ 和要进行处理的序列 $\varphi_1 = \varphi(s_1)$
　　循环：这个片段中的时刻从1 ~ T
　　每次使用一个概率 ε 选择一个随机动作 a_t
　　否则从当前训练的Q神经网络中选择Q值最大的动作

$$a_t = \max_a Q^*(\varphi(s_t), a; \boldsymbol{w})$$

　　在模拟器中执行动作 a_t，观察得到的回报以及新的观测 x_{t+1}
　　使 $s_{t+1} = s_t, a_t, x_{t+1}$ 和处理序列 $\varphi_{t+1} = \varphi(s_{t+1})$
　　将新得到的一次转换 $(\varphi_t, a_t, r_t, \varphi_{t+1})$ 存到D中
　　从D中随机抽取样本 $(\varphi_j, a_j, r_j, \varphi_{j+1})$

$$\text{得到 } y_j = \begin{cases} r_j & \text{叶子节点 } \varphi_{j+1} \\ r_j + \gamma \max_{a'} Q(\varphi_{j+1}, a'; \boldsymbol{w}) & \text{非叶子节点 } \varphi_{j+1} \end{cases}$$

　　沿着 $(y_i - Q(\varphi_j, a_j; \boldsymbol{w}))^2$ 梯度下降，调整神经元的权重

算法中使用了一个回放容器（replay memory），深度学习算法要求数据分布的独立性假设，如果数据之间有强相关性，直接从这些连续样本学习是低效的，同时计算出来的模型也会有偏向。而强化学习所处理的模型一般是马尔科夫决

策过程，训练数据是前后相关的序列。例如，下一次动作的选择不仅受到当前状态的影响，同时也会影响到下一时刻的状态。这意味着后面产生数据的分布也在变化，训练出来的策略可能非常不稳定，结果也会产生动荡。因此为了打破这样一个强相关性，需要进行随机采样以打破这种依赖关系。另一方面是历史局限性问题，因为当前局面只能看到游戏的一部分，无法有一个全局的认识，智能体在选择策略时会偏向某一情况，而多次经历这种情况又会使智能体在这个偏向上持续增强。例如在一个局面下向右走会得到最大的回报，智能体选择向右走，所以右边的情况会被更多的遍历到，训练数据也会大部分是右边的情况，从而影响到进一步的学习，并且更新参数容易迅速陷入一个较差的局部最小值。因此为了解决历史局限性的问题，需要将更多的历史样本加入到训练中，从而解决这样一个偏向性。Long-Ji Lin 93年通过使用回放容器来实现了机器人运动控制[51]，参考它的思想，使用回放容器存储过去的历史。将智能体每次与环境交互时得到的“状态-动作-新状态-回报”序列回放容器中，每次训练时从这个回放容器中随机抽取大小为batch的样本，采用mini-batch 梯度下降法进行参数的更新。

强化学习中经常面临的一个问题是“探索”或“利用”问题（Exploration and Exploitation Problem），例如在蒙特卡洛博弈树搜索时会存在这样一个问题，即每次选择探索尚未充分的陌生节点还是选择已经扩展过许多次且估值最大的节点。深度Q学习算法使用ε-贪心探索解决这样一个困境：在训练深度Q学习网络时，我们以ε的概率选取随机的动作作为下一步动作，1-ε的概率选取分数最高的动作。ε随着时间从1减少到0.1。这意味着开始时，系统完全随机地探索状态空间，最后以固定的概率探索。

深度强化学习克服了传统强化学习算法无法处理高维状态空间的问题，结合强化学习与深度学习，实现了从感知（Perception）到动作（Action）的端对端学习的一种全新的算法，避免了传统识别算法中复杂的特征提取过程。同时智能体在自我学习的过程中可以发现特征之间的内在联系。同时深度强化学习能够为大规模机器博弈，非完备信息机器博弈系统的实现提供了一个可行的方法，同时为将算法扩展到现实生活中提供了可能。

深度强化学习可扩展性强，比如同样的扑克游戏仅通过略微改变扑克的建模方式与最后动作的种类而不需要关心如何通过已有的领域知识建立学习方法等。因此深度增强学习具备了使智能体实现完全自主的学习一种甚至多种技能的潜力。

3.2 改进的深度强化学习网络

深度强化学习算法输入的是当前局面，所以深度强化学习算法只能根据当前状态计算最后输出动作对应的*Q*值，而无法考虑过去的历史信息。例如四国军棋这种，需要根据前面的对手、队友的动作来判断他们的可能的棋子分布，又或是对手在当前状态看似没有作用的一个动作会在十几步后发挥至关重要的作用，这些都是博弈时需要进行考虑的问题。而深度强化学习在处理这种后面的动作的选择与前面的动作之间的相关性较强的问题时效果较差。针对上述问题，我们尝试为深度强化学习网络增加记忆，使智能体做决策不仅能够考虑当前观察到的局面而且综合了记忆中存储的历史信息综合判断。

递归神经网络（RNN）又分为时间递归神经网络（Recurrent Neural Network），和结构递归神经网络（Recursive Neural Network），本文主要使用的是时间递归网络，递归神经网络的核心思想是网络隐含层存在一条指向自己的回路，并且这条回路也有一定的权重，可以随着梯度下降进行权值更新，通过这种连接，隐藏层的输入不仅来自输入层或者前一层隐藏层，还有一部分来自上一时刻隐藏层的输出，隐含层能够一定程度上将过去的信息保存下来，然后当新的数据输入时综合过去信息进行考虑，如图 3-7是一个简单的典型的递归神经网络，可以看到每个隐藏层都有一条指向自己的回路。

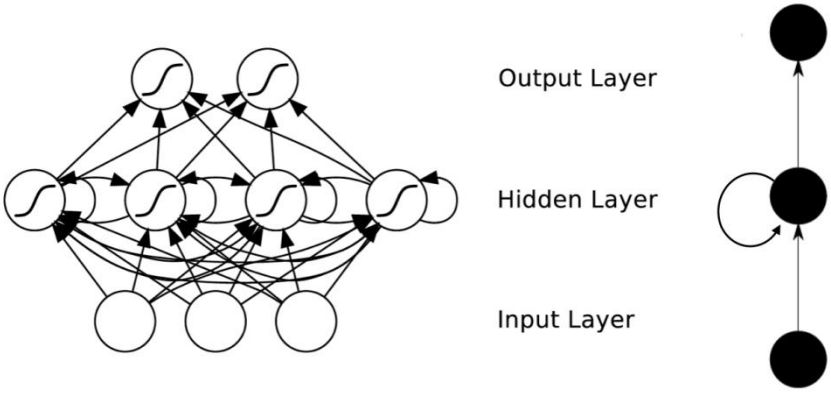

图3-7 传统RNN结构

传统RNN的层数表示存储了过去记忆的多少，层数越多，网络能够记住的历史信息越多，但层数过多时就会带来梯度消失或者梯度爆炸的现象。RNN沿着时间进行反向传播BPTT（Back Propagation Through Time），其中梯度消失是RNN最常见的问题，究其原因是梯度下降时导数的链式法则导致了连乘的形式，而连乘中的其中一项是激活函数的导数，激活函数的导数的最大值一般是小于1的，例如激活函数Sigmoid的导数最大值为1/4，很多小于1的项连乘就很快就会

逼近于零，使得梯度还没到达最前面的时刻便已经为零了。因此传统 RNN 实际能够利用的历史信息非常有限。图3-8直观的解释了梯度消失。

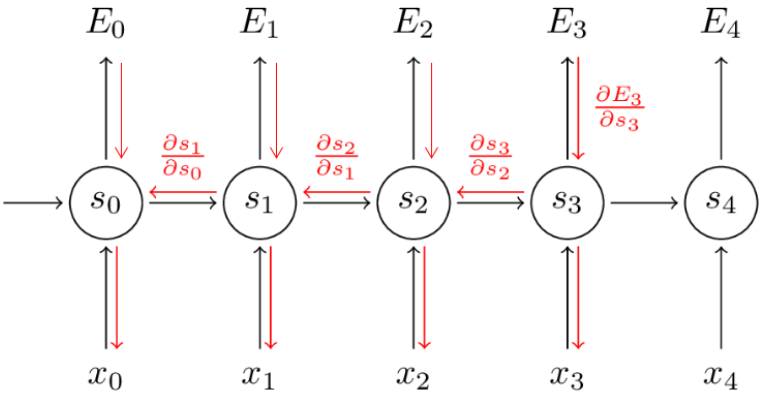

图3-8 梯度消失示意图

图中红色的箭头代表了梯度传递的方向，可以看到梯度不只是输出层到隐藏层和隐藏层到输入层，还沿着时间序列从当前时刻的隐藏层传到前一时刻的隐藏层，又因为梯度消失的原因，可能梯度到达 s_1 已经逼近与零，无法有效的根据梯度更新前面时刻对应的权重。

长短期记忆模型（Long Short Term Memory，LSTM）由Hochreiter & Schmidhuber 等人提出[52]，他们用LSTM单元替换了RNN隐含层的神经元，从而解决了 RNN 的梯度消失问题。后被Alex Graves进行了改良和推广，现在使用最广泛的 LSTM单元如图 3-9所示。

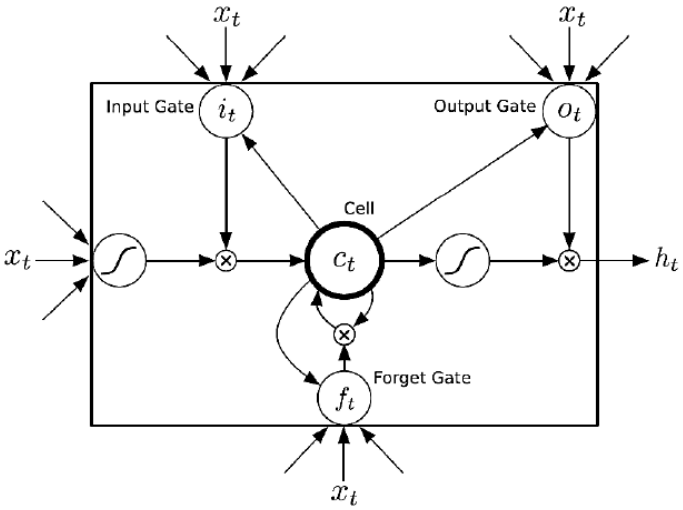

图3-9 LSTM 单元示意图

LSTM 单元将历史信息保存在记忆单元（Memory Cell）中，它能够记住长期的信息，同时通过两个专门设计的“门”结构来删除或者增加记忆单元中

的历史信息，这两个“门”分别是遗忘门（Forget Gate）和输入门（Input Gate），另外还有一个输出门（Output Gate），用于控制记忆单元输出到下一层神经元的状态的多少。

LSTM单元的更新分为以下几个步骤：

（1）更新LSTM单元，第一步是决定需要从记忆单元中忘掉什么信息，这个决定通过遗忘门来实现，遗忘门的值为 f_t，它通过输入的信息 x_t 以及前一时刻LSTM单元的输出值 h_{t-1} 经过Sigmoid变换输出一个介于0到1之间的数，$f_t = 0$ 表示完全丢掉记忆单元的中的记忆，$f_t = 1$ 表示完全保留，f_t 的计算公式为公式（3-9）：

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \tag{3-9}$$

其中 W_{xf} 是对应输入数据，W_{hf} 对应上一时刻LSTM单元输出的权值

（2）下一步是决定有什么样的新信息需要加入到记忆单元中，创建一个新的候选值向量 $\tilde{c}_t$，$\tilde{c}_t$ 的计算公式为公式（3-10）：

$$\tilde{c}_t = \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \tag{3-10}$$

这里使用Tanh进行处理，得到一个介于-1到1的值。

（3）输入门是决定候选信息中有多少信息量需要加入到记忆单元中，输入门的值为 i_t，$i_t = 0$ 表示记忆单元不再受新的输入的影响，仅通过输出门进行更新，$i_t = 1$ 表示将候选值向量全部加入到记忆单元中，i_t 的计算公式为公式（3-11）：

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i) \tag{3-11}$$

（4）更新记忆单元，计算当前时刻记忆单元状态值 c_t，我们把前一时刻的记忆单元 c_{t-1} 与 f_t 相乘，丢弃需要遗忘的信息。接着加上 $i_t \times \tilde{c}_t$，将新的信息加入到记忆中，c_t 的计算公式为公式（3-12）：

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{3-12}$$

由公式可见，记忆单元状态更新取决于通过输入门调节的当前的候选记忆单元值 $\tilde{c}_t$ 和通过遗忘门调节的自身状态 c_{t-1}。

（5）输出门用于控制记忆单元状态值的输出，输出门的值为 o_t，o_t 的计算公式为公式（3-13）：

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \tag{3-13}$$

（6）最后 LSTM 单元的输出 h_t 由公式(3-14)给出。

$$h_t = o_t \times \tanh(c_t) \tag{3-14}$$

LSTM这样的结构使其具有保存和更新长距离历史信息的能力，例如，如

果 $f_t = 1$，记忆单元中的信息就能长时间的一直保持下去；如果 $i_t = 0$ 即输入门保持关闭，记忆单元便不再受新的输入的影响。

LSTM的记忆单元能够保存过去的记忆，那么将LSTM与深度强化学习网络相结合便可以达到每次判断都能够综合当前局面和前几步的记忆选择得分最高的动作执行。

具体的实现方式是用一个全连接的LSTM层替换深度强化中的全连接层，使用擅长提取特征和发现特征之间的内在联系的卷积神经CNN处理输入的状态，然后将提取出的当前局面的特征传给长短期记忆模型LSTM，接着是一个用于分类的全连接层。最后一层是输出层，输出的是动作对应的*Q*值。我们把这样的网络结构称为LSTM_DQN，它的网络结构如图3-10所示，用全连接的LSTM层替换了第一层全连接层。

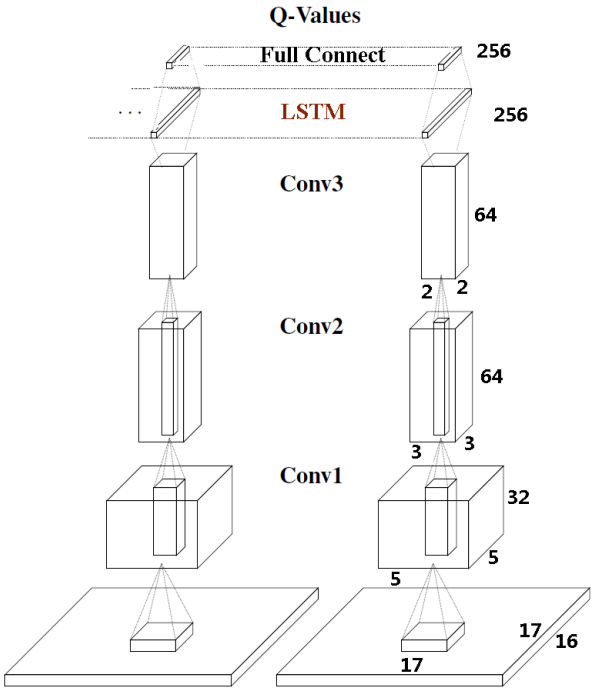

图3-10 改进的深度强化学习网络

与普通的深度强化学习不同的是，最后的动作选择由 $a_t = \max_a Q^*(s_t, a; \boldsymbol{w})$ 变为公式（3-15）：

$$a_t = \max_a Q^*(s_t, h_{t-1}, a; \boldsymbol{w}) \tag{3-15}$$

h_{t-1} 是前一个时间节点的LSTM记忆单元保存的内容。

回放容器中保存的状态-动作-新状态-回报变为整个一局游戏的状态-动作-回报序列，即 $(s_t, a_t, r_t, s_{t+1}) \rightarrow (s_1 a_1 r_1 s_2 a_2 r_2, \ldots s_T a_T r_T)$，因为每次*Q*值的计算都

需要考虑过去的历史，所以不能单纯的存储当前的状态。

参数的更新使用的BPTT（Back Propagation Through Time），梯度沿着时间序列进行反向传播。

下面给出改进的深度强化学习的算法流程：

算法 3-3 改进的深度强化学习算法

初始化：初始化回放容器D，深度Q神经网络中神经元的权重随机赋值

循环：每一节片段episode

初始化网络中LSTM层的记忆细胞cell，初始状态为 s_1

循环：这个片段中的时刻从1 ~ T

以概率 ε 选择一个随机动作 a_t

否则从当前训练的改进的深度Q学习网络中选择Q值最大的作：

$$a_t = \max_a Q^*(s_t, h_{t-1}, a; \boldsymbol{w})$$

在模拟器中执行动作 a_t，得到新的观测 x_{t+1}，回报 r_t

如果达到片段终点跳出循环

将新得到的历史序列 $(s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_T, a_T, r_T)$ 存到D中

从D中随机抽取大小为batch的样本 $(s^i_1, a^i_1, r^i_1, s^i_2, a^i_2, r^i_2, \ldots, s^i_T, a^i_T, r^i_T)_{i=1,\ldots batch}$

得到 $y^j = \begin{cases} r_t^i & \text{叶子节点} \\ r_t^i + \gamma \max\limits_{a'} Q(s_{t+1}^i, h_{t+1}^i, a'; \boldsymbol{w}) & \text{非叶子节点} \end{cases}$

沿着 $(y_i - Q(s_j, h_t, a_j; \boldsymbol{w}))^2$ 采用 BPTT 进行反向传播，调整神经元的权重

3.3 基于蒙特卡洛博弈树搜索的回报函数

扑克属于非完备信息博弈，发牌过程是随机非确定性的，同时对手的底牌也是未知信息，这种不确定性带来的巨大状态空间导致无法扩展完整信息的博弈树，因此无法得知最优收益值，同样也无法仅仅将游戏结果作为回报值，例如德州扑克机器博弈中，如果一局的结果为负又会分为两种情况，第一种是智能体根据当前局面以及过去的历史推测出了错误的选择而导致的失败，此时应该惩罚系统，第二种是智能体虽然选择了弃牌，导致输掉了这局，但是综合当前局面和过去历史防止失去更多的筹码而做出的合理的决定，此时也应该奖励系统，因此需要针对不同的情况对游戏结果进行区分，需要设计新的回报函数。

3.3.1 蒙特卡洛博弈树搜索

蒙特卡洛方法是一种人工智能问题中做出最优决策的方法，该算法将所求解的问题同一定的概率模型相联系，通过多次的模拟或抽样得到一个近似的结果。蒙特卡罗方法常用于数值计算问题以及求解优化问题，对于求解过程过于

复杂而导致无法在有限的时间或者没有足够的资源去获得准确解的问题，蒙特卡洛方法通过大量的随机抽样获得这个问题的近似解。随着计算机性能的提升，并行计算的技术越来越成熟，蒙特卡洛方法也逐渐被研究者关注和利用。

蒙特卡洛博弈树搜索（MCTS）就是将蒙特卡洛方法应用在博弈树搜索中，通过蒙特卡洛抽样方法逐步建立和扩展博弈树，它结合了随机模拟的一般性和博弈树搜索的准确性，可以使那些有可能成为最优走步的分支获得更多被探索的机会，在有限的时间内使用有限的资源提高搜索的准确率。MCTS将当前待评估局面作为根节点开始不断构建搜索树，树中的每个中间节点包含了对当前局面的估值信息，具体的搜索过程如图3-11所示：

（1）扩展节点选择：从根节点开始，递归的选择其子节点中评分最高的节点进行扩展。博弈树的叶子节点代表游戏的结束局面，可选择的扩展的节点是非叶子节点，且至少还有一个子孩子节点没有被扩展过。

（2）扩展过程：选中节点一个或者多个子节点会被展开并加入博弈树，同时选中成为它们的父节点。

（3）模拟估值：从新加入博弈树的节点开始通过蒙特卡洛方法随机生成博弈双方的合理动作，直到游戏结束，得到一个确定的结果。

（4）反向传播：将模拟估值得到的结果从叶节点开始层层回溯给各自的父节点，根据结果调整这些父节点的估值。

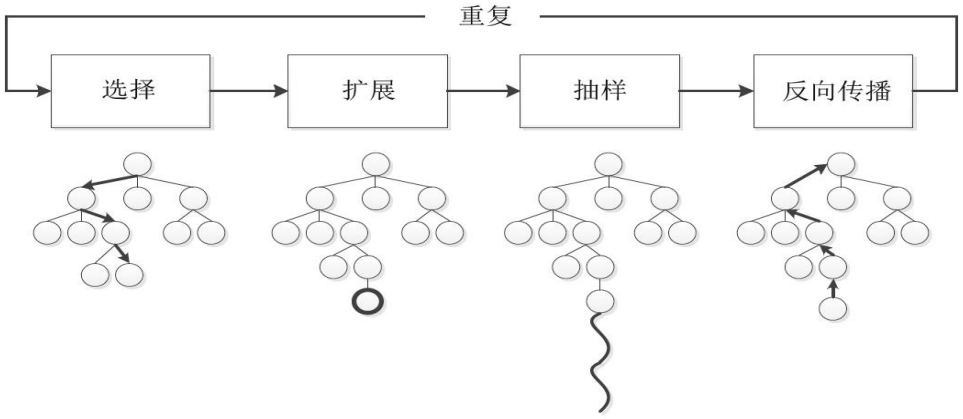

图3-11 蒙特卡洛博弈树搜索算法流程

在规定的时间或者搜索次数内重复以上过程，最后根据根节点的估值判断搜索的结果。

将UCB（Upper Confident Bound Algorithm）算法最早由Auerf提出，用于求解节点选择时的利用-探索矛盾，即每次扩展节点选择时选择当前估值最高但是扩展过很多次的节点，还是估值次优但是扩展次数很少的节点。在UCB算法中，每次都会选择具有最大的UCB值的节点，UCB值的计算公式如公式（3-16）：

$$UCB_i = \overline{X}_i + c\sqrt{\frac{\ln N}{n_i}} \tag{3-16}$$

$\overline{X}_i$ 表示节点的估值，c 表示调节因子，n_i 表示节点访问次数，N 表示其父节点已经被访问的总次数，可见算法在寻找“利用”和“探索”中的平衡点。图3-12为基于UCB策略的蒙特卡洛博弈树。

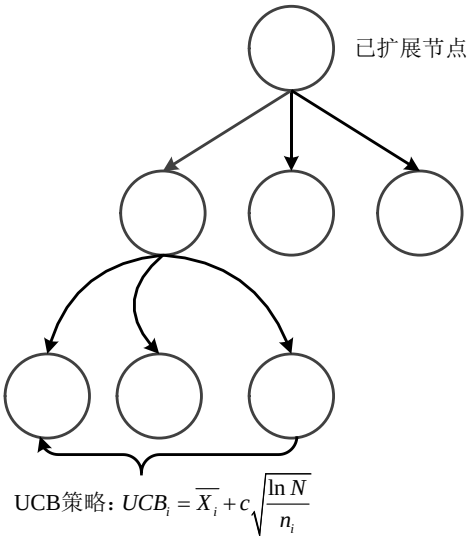

图3-12 基于UCB策略的蒙特卡洛博弈树

3.3.2 蒙特卡洛博弈树搜索应用于深度强化学习中

在非完备信息博弈树中应用MCTS算法，博弈树由四种节点构成：玩家节点，对手节点，随机节点以及叶子节点。其中玩家节点是指需要玩家做出决策的节点。因为非完备信息机器博弈中部分信息不可见，针对扑克游戏中对手节点的状态，或者根据一定的概率进行预测，或者通过随机抽样进行确定。随机节点是指游戏中的随机过程，不受博弈双方控制和影响，例如德州扑克中的公共牌是随机发出的，不受博弈双方的影响，同时在河牌阶段之前公共牌的信息也是不全的，在一些状态空间很小的游戏可以完全扩展，但是像非完备信息机器博弈这种复杂的游戏同样需要进行随机抽样。叶子节点代表了一局游戏的结束，在MCTS中需要将游戏结果层层回溯给各自的父节点。

图3-13是MCTS应用于非完备信息博弈中扩展了一部分的博弈树，可以看出对手节点以及随机节点都无法知道确定的信息而需要进行随机抽样。每局游戏结束，将游戏结果 r 作为返回值回溯给父节点，回溯直到根节点。

定义 $n(s)$ 为通过蒙特卡洛树搜索访问状态 s 的次数，$V(s)$ 是状态 s 的评估函数，计算公式见式（3-17）：

$$V(s)=\frac{1}{n(s)}\sum_{i=0}^{n(s)} r(i) \tag{3-17}$$

上述公式中的$r(i)$为第i次回报值。

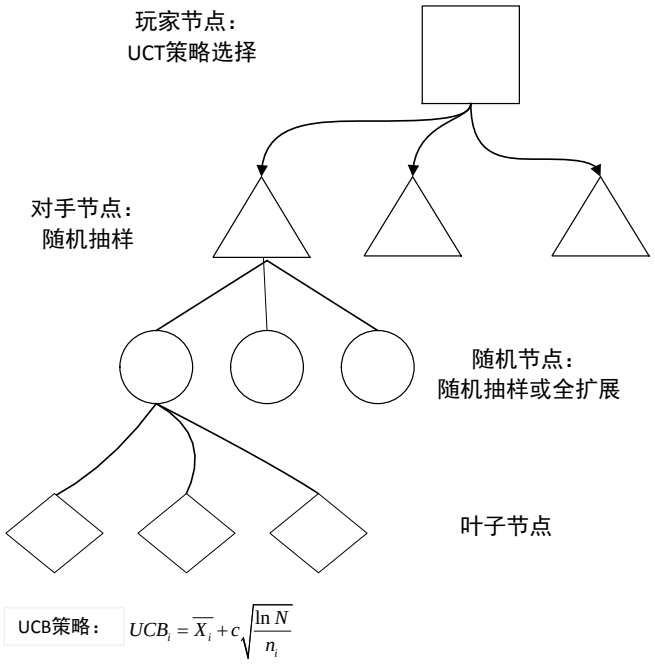

图3-13 MCTS应用于非完备信息博弈

蒙特卡洛树搜索是一个循环计算的过程，每次需要选择UCB最大的节点，状态s下UCB的计算公式见式（3-18）：

$$UCB=V(s)+c\sqrt{\frac{\log N(s)}{n(s)}} \tag{3-18}$$

其中$N(s)$为s的父节点的访问次数，c是调解开发和利用平衡的参数。

通过蒙特卡洛博弈树搜索探索一定次数，设根节点为S，即博弈树搜索的开始状态，则最后的结果的计算公式为公式（3-19）：

$$V(S)=\frac{1}{n(S)}\sum_{i=0}^{n(S)} r(i) \tag{3-19}$$

计算出的结果值为在博弈树搜索开始状态S下玩家的期望收益$V(S)$，假设深度强化学习网络最后得到的回报值为$R(S)$，如果$R(S)$为正，同时$V(S)<R(S)$则说明智能体当前采取的策略导致智能体比期望中获得了更好的收益，则需要奖励系统；但如果$R(S)$为正，同时$V(S)\geq R(S)$，虽然赢得了筹码，但是当前牌力下智能体采用了不合理的策略导致获得的筹码数比期望的少，此时仍然应该惩罚系统。如果$R(S)$为负，但$V(S)\geq R(S)$则说明智能体当前采取的策略防止了输掉更多的筹码，则说明当前的网络有效，则需要奖励系统；如果$R(S)$为负，

同时$V(S)<R(S)$则说明智能体当前采取的策略导致智能体比期望的收益输掉了更多，则需要惩罚系统。如图3-14显示了基于蒙特卡洛博弈树搜索的回报函数流程图。

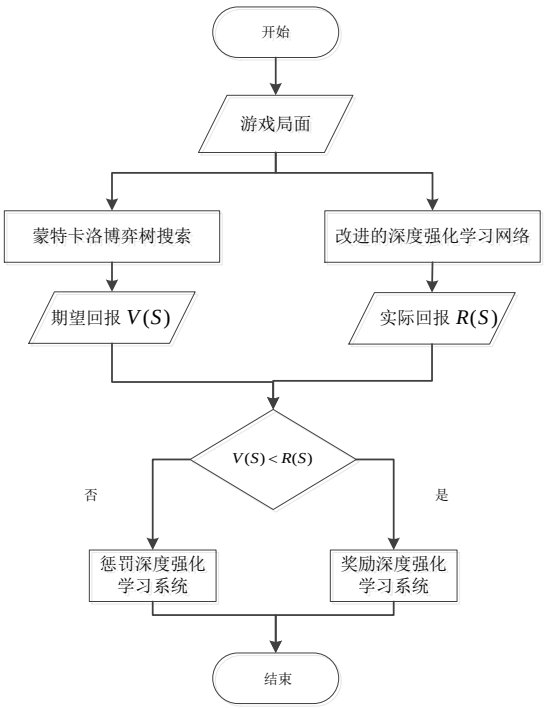

图3-14 基于蒙特卡洛博弈树搜索的回报函数流程图

3.4 扑克游戏的模式匹配建模方法

扑克游戏是人工智能体领域很大的一个挑战，主要的原因是扑克游戏巨大的状态空间无法很好的处理。同时扑克游戏包括很多变种，不同的扑克游戏都有着不同的规则，每种扑克游戏都需要各自大量的领域知识进行支撑。传统应用在扑克游戏的算法需要手动选取特征，而人类根据自身经验选择的特征往往并不能完全代表整个博弈局面，容易陷入主观偏见，同时可能有些不被关注的特征会决定整个博弈过程的发展，特征与特征之间的联系也是需要考虑的很重要的一个方面。另一方面，传统算法可扩展性差，有些博弈游戏例如德州扑克、2-7三换牌、Kuhn扑克等虽然形式大致相同，但是如果用传统方法实现需要设计不同的数据结构，针对不同的游戏设计不同的策略，需要大量的领域知识，且必须有丰富的游戏方面的经验。

如果可以像图像识别一样将整个局面交给模式匹配算法去识别，并直接输出需要执行的动作便可以很好的解决上面的问题。深度强化学习所使用的深度学习网络是卷积神经网络，能够很好的发现图像特征与特征之间的联系，同时图像可以直接作为输入，避免了复杂的人工提取特征的过程。

扑克游戏可以作为深度强化学习等模式匹配算法的输入面临的两个挑战：

（1）如何表示：一个学习模型所输入的数据需要有良好的表现形式，其中需要包含丰富的局面信息，如玩家的手牌，任何已知的公共信息，以及之前所采取的行动，游戏回合等等。

（2）无偏性：从当前局面提取出的信息必须不具有偏向性，一个好的学习算法可以自己从输入数据中提取中相关的特征信息，不需要人为导向，而人的某些固定思维有时候也会使结果有所偏向，无法达到一个均衡的状态，容易被对手找到弱点。

为了解决上面的问题，我们提出了一个新的扑克类游戏表示方式，它能够被用在相似的不同的扑克游戏，作为深度强化学习算法等模式匹配算法的输入能够很好的进行学习训练，仅需要很少的领域知识便可以训练出一个高水平智能体。

本章展示了基于三维矩阵的表示如何表示三个扑克游戏：

（1）二人德州扑克：二人德州扑克拥有四个下注轮，每个玩家拥有两张手牌，手牌只对玩家自己可见。五张公共牌分三个轮次发完：首先三张手牌（Flop阶段），然后是一张单牌（Turn阶段），最后一张单牌（River阶段），每个玩家的手牌加上公共牌中选出最好的五张牌比大小，最大的获胜。

具体的建模方式如下：

一副扑克去掉大小王有52张牌其中有2-A共13种牌值，且每种牌值都有4种花色，所以我们可以用一个 4×13 的稀疏的0-1矩阵代表一张牌，如图3-15。因此在这个矩阵中只有一个元素是非0的，而实际工程中，我们往往将这个 4×13 的矩阵用0填充成 17×17 大小的矩阵，这样做并不会增加额外的信息，但是能够方便卷积层以及pooling层的处理。

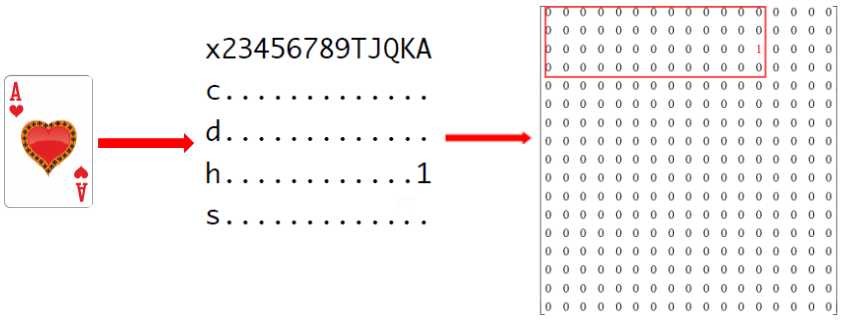

图3-15 扑克牌的矩阵表示

对于德州扑克我们可以用 $2\times17\times17$ 的三维矩阵表示自己的手牌，$5\times17\times17$ 的三维矩阵表示五张公共牌，同时额外增加两个满牌矩阵（17×17），一个包括了所有的公共牌，表示公共牌组合起来的牌力，另一个包括我的手牌与所有的公共牌，表示我的牌组合起来的能够达到的牌力。这样做主要有两个优点：

a、大量输入使得能够很好的方便卷积层提取特征，发现特征之间的内在联系。

b、满牌矩阵使得整个编码过程不仅能够对每一张牌建模，同时能够发现牌组合起来的模式，如图3-16所示，一对（两张牌拥有相同的牌值，但是花色不同，那么这两张牌会在同一列出现），同花（五张牌拥有相同的花色，那么这五张牌会出现在同一行），而不需要进行专门的分类判断和花色同构判断（例如，例如AsKd和KhAc本质上是一样的）。

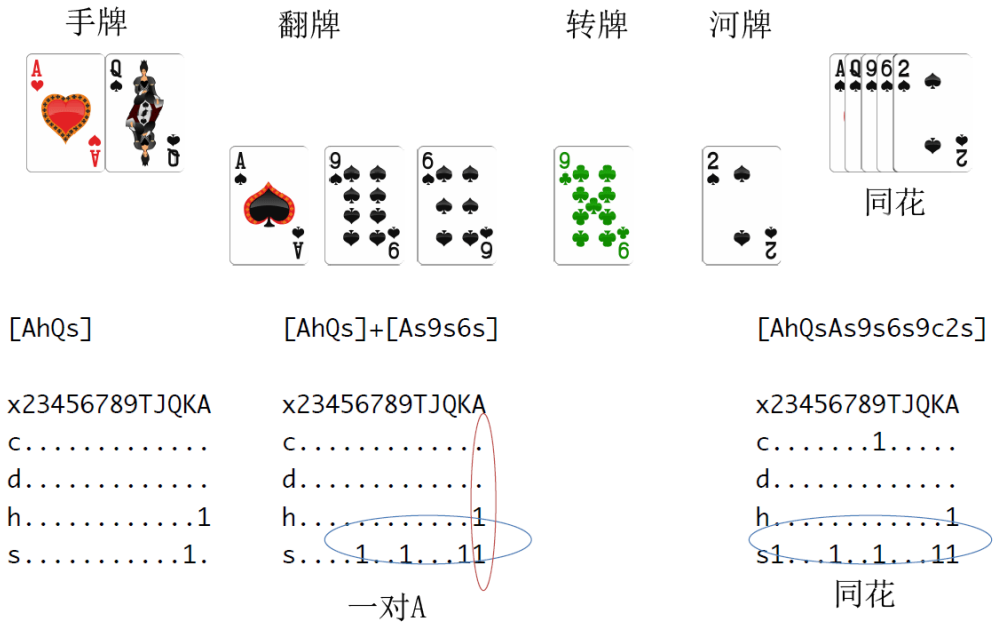

图3-16 满牌矩阵

同样整个游戏状态，上下文关系也是我们需要考虑的很重要的因素。

a、游戏状态：

对于德州扑克这种多回合的游戏，我们需要知道当前处于哪个阶段，德州扑克共有四个阶段（Perflop、Flop、Turn、River），因此我们增加四个 17×17 的矩阵表示这四个阶段，例如当处于其中一个阶段时，这个阶段对应的矩阵元素都为1，其他阶段对应的矩阵的元素都为0，如图3-17所示。

底池中的筹码量即每一个牌局里众人已押上的筹码总额，也即该局的奖金数目，可以用一个 17×17 的矩阵表示，假设下注的最小数目是5，那么可以使矩阵中2c位置为1其他位置为0，如果下注的数目是20，那么矩阵中2c&2d&2h&2s位置的值为1其他位置为0，这样可以用矩阵中1的数目的多少直观的表示底池的筹码数，方便卷积层提取特征，图3-18表示的是当前的底池筹码数量是150。

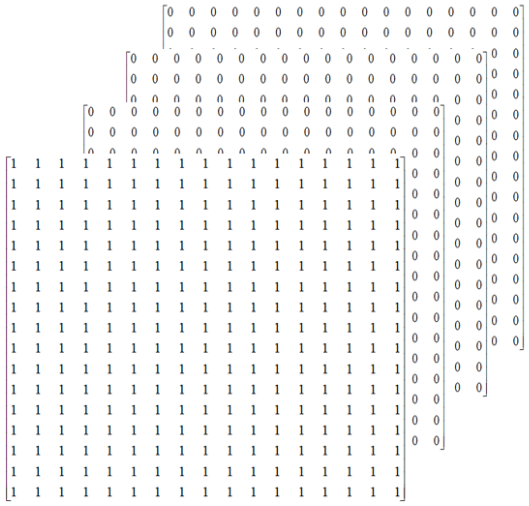

图3-17 扑克游戏中所处阶段的表示方法

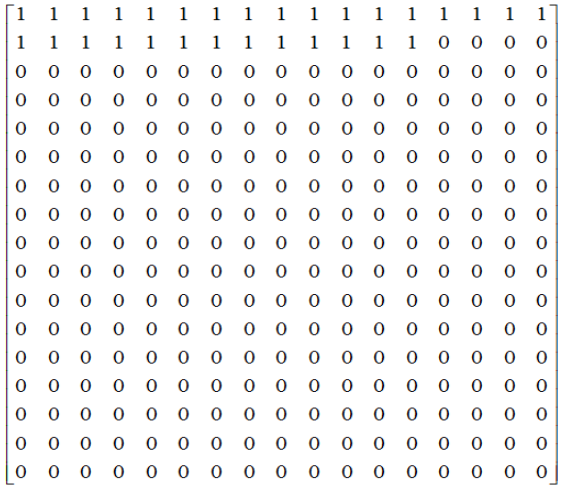

图3-18 扑克游戏中底池筹码数量表示方法

b、上下文关系：

用两个17×17 的矩阵表示对手在我做动作之前的动作，第一个矩阵表示跟注或者加注，因为对手前一个如果选择弃牌则本轮比赛结束，所以只有两种状态，跟注矩阵元素全为0，加注矩阵元素全为1。第二个矩阵是当第一个矩阵为1时即当对手加注时，第二个矩阵需要表示对手加注的数量。

因此每次训练都需要将当前博弈局面编码成大小为$16 \times 17 \times 17$ 的三维矩阵，然后作为输入传入卷积层，输出为每个动作对应的Q值。表3-1显示了二人德州扑克建模所需提取的局面信息。

（2）Kuhn扑克：在Kuhn扑克游戏中，牌组是有J、Q、K和A四张牌组成的。开始的时候每人发一张牌，然后剩下的一张牌是隐藏的。Kuhn 扑克只有一轮且要在翻牌前进行押注，押注筹码是固定的，游戏允许加注、跟注或者弃牌。

表3-1 二人德州扑克建模提取的局面信息

特征	矩阵个数	描述
手牌	2	自己的手牌
公共牌	5	公共牌
所有公共牌	1	所有公共牌一个矩阵
所有牌	1	所有牌合到一个矩阵
阶段数	4	德州扑克的四个阶段
底池筹码数	1	底池筹码数
本轮对手的动作	2	本轮对手的动作

Kuhn扑克只有四张牌，所以建模时0-1稀疏矩阵的大小变为4×4，而其他方面与德州扑克类似，手牌与公共牌数量以及轮次数不同，因此只需改变所需的特征数量，表3-2显示了Kuhn扑克建模所需提取的局面信息。

表3-2 Kuhn扑克建模提取的局面信息

特征	矩阵个数	描述
手牌	1	自己的手牌
公共牌	1	公共牌
所有牌	1	所有牌合到一个矩阵
阶段数	1	德州扑克的四个阶段
底池筹码数	1	底池筹码数
本轮对手的动作	2	本轮对手的动作

因此每次训练都需要将当前博弈局面编码成大小为$7\times4\times4$的三维矩阵，然后作为输入传入卷积层，输出为三个动作对应的Q值。

（3）2-7三次换扑克：2-7三次换游戏中，获胜的则是最佳低牌牌组。这项游戏称为2-7三次换是因为其最佳牌组是没有同花的7-5-4-3-2。游戏开始，每位玩家会拿到五张面朝下的底牌，接着开始第一轮下注，玩家可选择跟注、加注或弃牌。第一轮下注结束后，没有弃牌的玩家可以进行换牌，也就是舍弃不要的牌，由新发的牌取代，以组成更佳的一手牌。玩家可以选择“全部保留”，即一张牌都不换，也可以把五张牌全部换掉。换牌结束后，就开始第二轮下注，第三轮下注结束后，如果没有玩家弃牌，则进行摊牌，握有最佳牌组的玩家赢得底池。

2-7三次换扑克建模时0-1稀疏矩阵的大小为17×17，没有公共牌，阶段数

为3，而本轮对手的动作主要是考虑对手换了几张牌，可以是一张也可以是5张，所以需要5个17×17 的矩阵表示。表3-3显示了2-7三次换扑克建模所需提取的局面信息。

表3-3 2-7三次换扑克提取的局面信息

特征	矩阵个数	描述
手牌	5	自己的手牌
所有牌	1	所有牌合到一个矩阵
阶段数	3	德州扑克的四个阶段
底池筹码数	1	底池筹码数
本轮对手的动作	5	本轮对手的动作

因此每次训练都需要将当前博弈局面编码成大小为$15 \times 17 \times 17$ 的三维矩阵，然后作为输入传入卷积层，输出层的大小为32，即总共会面临的所有选择的总数，不换总共有一种选择，换一张共有5种选择，换两张共有10种选择，换三张共有10种选择，换4张1种选择，换5张一种选择，因此共有32个输出，输出的值为每种选择对应的Q值。

综上大多数扑克游戏都可以采用这样的表示方式，建模方式是无偏的，因此这样的建模方式加上改进的深度强化学习LSTM_DQN便可以无需先验知识的情况下进行训练学习。

3.5 本章小结

本章首先深入探究了深度强化学习算法的原理，如何将深度学习与强化学习结合在一起。分析了深度强化学习应用到非完备信息机器博弈中存在的短时记忆问题并通过长短期记忆模型LSTM的引入加以解决。之后介绍了基于蒙特卡洛博弈树搜索的回报函数，解决了这类扑克游戏中游戏结果表示时面临的问题。最后介绍了可以作为深度强化学习等模式匹配算法输入的扑克建模方法，并通过具体的三种扑克游戏说明这种建模方式的可扩展性。

第4章 非完备信息机器博弈系统的实现和实验分析

本论文以非完备信息机器博弈为研究对象，实现了基于改进的深度强化学习算法的二人德州扑克系统。本章将对系统框架、系统的训练方式、具体的网络结构以及一些参数的选择进行了说明。然后对两种网络结构（深度强化学习网络和改进的深度网络）以及不同的梯度下降算法进行了实验对比。最后实验智能体与其他一些智能体进行了对弈，并根据比赛结果进行了分析。

4.1 德州扑克机器博弈系统

4.1.1 德州扑克机器博弈系统框架

本文的主要开发平台是：NVIDIA Tesla K20 GPU运算卡，显存容量5GB ，CUDA并行处理核心2496个，峰值双精度浮点性能1.17Tflops，峰值单精度浮点性能3.52Tflops。操作系统为：Ubuntu 14.06 LTS，x64架构。开发语言主要有C++、C和Python。

图4-1显示了德州扑克机器博弈系统的通信交互框架，德州扑克机器博弈系统通过socket进行通信，服务器的角色相当于德州扑克牌桌，负责发牌、确保游戏按规则进行，判别比赛胜负等等，客户端对应的便是每一个玩家。

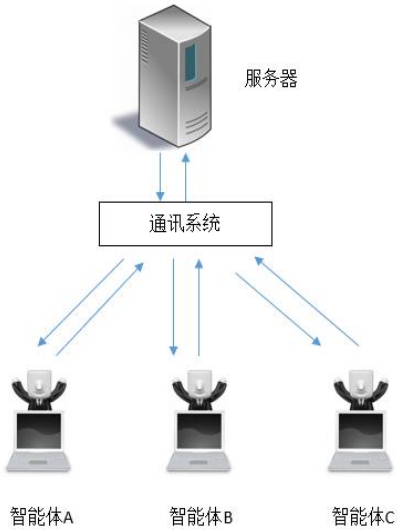

图4-1 德州扑克机器博弈系统的通信交互框架

游戏初始化阶段，服务器端为每个玩家分配一个固定的端口，并将局面信息（包括手牌，公共牌，对手动作以及结束信息等）分派给每一个玩家，玩家

接收到服务器端传来的信息则根据自己训练好的策略产生相应的动作，然后将动作信息传给服务器端，服务器端对局面进行解析，将新的公有信息分派给每一个玩家。如此往复便可以实现机器博弈，图4-2显示了德州扑克机器博弈系统数据通信流程图。

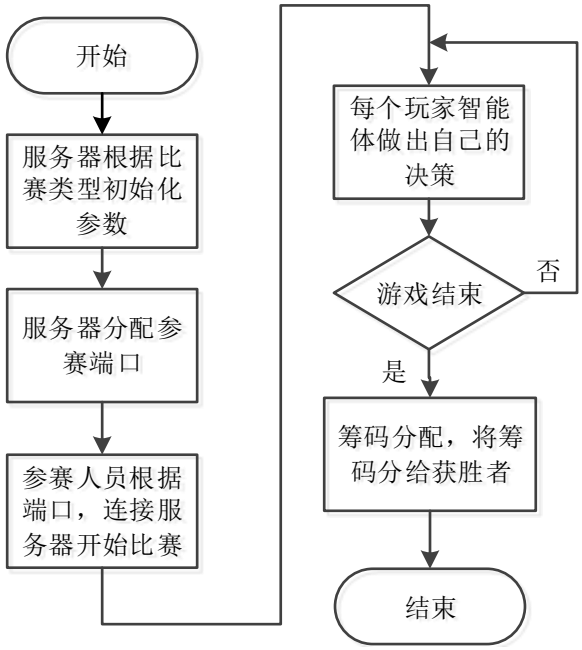

图4-2 德州扑克机器博弈系统数据通信流程图

图4-3为德州扑克系统比赛界面，可以看到我们改进的深度强化学习算法智能体hitsz_DQN在第10776局时赢得了60筹码，根据双方的手牌信息和公共牌信息可知，hitsz_DQN最终获胜的牌型为对8，本轮结束hitsz_DQN的总筹码数变为130465而ACPC官方智能体acpc_player的筹码总数为-130465。在比赛过程中还可以清晰的看到双方交替加注或跟注的过程，便于博弈智能体程序进行改进。

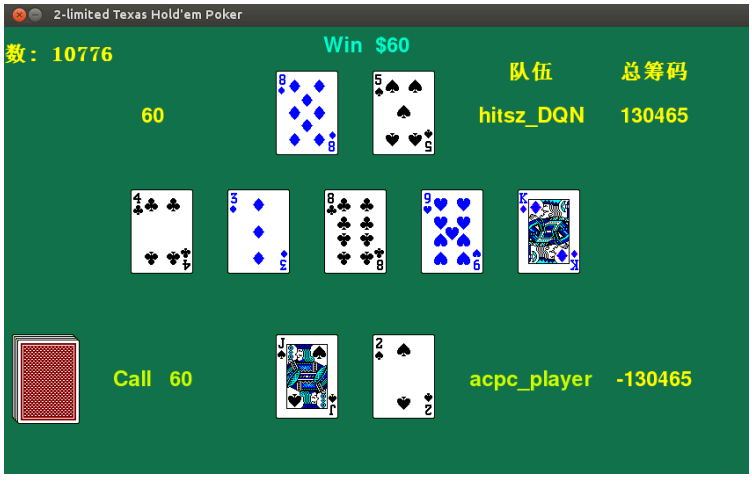

图4-3 系统界面

4.1.2 德州扑克智能体训练方式

采用改进的深度强化学习网络的智能体在游戏起始阶段不知道任何德州扑克的取胜策略，深度学习网络中的参数采用服从高斯分布的随机初始化，因此，开始的时候智能体选取的动作也是随机的。在与ACPC中的历年参赛的部分智能体或者自身对弈的过程中不断强化自己，根据基于蒙特卡洛博弈树搜索的回报函数得到的回报值，选择奖励或者惩罚智能体，反复训练后智能体便学会了德州扑克中的取胜策略。

实验智能体的训练过程：

（1）实验智能体与二人德州扑克服务器端通过socket通信。

（2）当服务器端将当前局面传给智能体，利用第三章中的德州扑克的建模方法，将服务器端传来的状态转化为 $17\times17\times16$ 的三维矩阵作为深度强化学习网络的输入，经过三层卷积层以及一层LSTM后得到三个动作对应的Q值，将训练出的Q值最高的动作作为智能体所采用的动作传给服务器端。

（3）对手做动作。

（4）重复（2）和（3）的过程直到本局游戏结束，根据基于蒙特卡洛博弈树搜索的回报函数得到的结果判断智能体最后的动作是否合理，如果合理则奖励智能体，如果不合理则惩罚智能体。

（5）重复（2）、（3）、（4）直到整个游戏结束。

整个系统的训练过程不需要人为参与，不需要手工提取特征，完全让智能体自主学习如何玩德州扑克，仅需要很少的领域知识。图4-4为深度强化学习网络与德州扑克系统的交互框图。

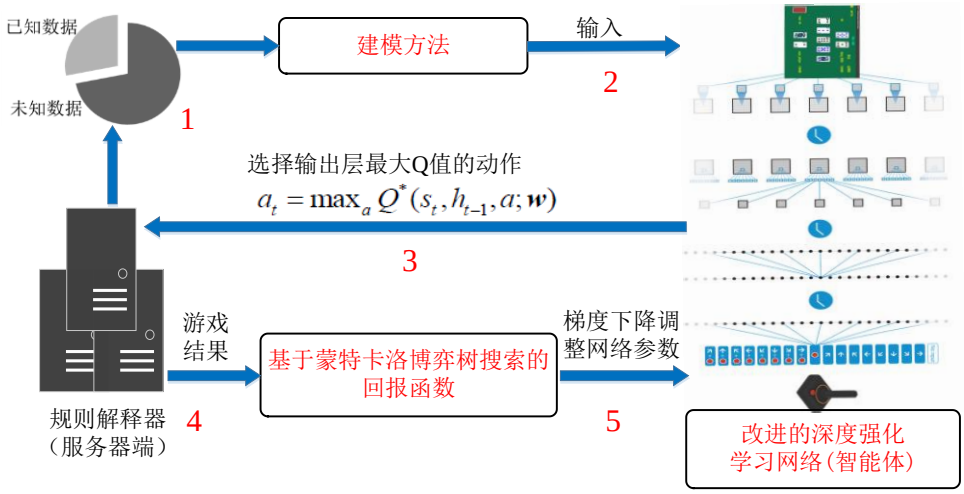

图4-4 深度强化学习网络与德州扑克系统的交互框图

其中的（3）中的对手又可以分为ACPC大赛历年参赛的部分智能体、简单策略对手，以及自己的不同参数的版本：

① ACPC大赛历年参赛的部分智能体：主要的对战对象包括ACPC官方提供的智体，过去几年哈工大参赛智能体，以及采用卷积神经网络监督学习训练的智能体[31]。本实验智能体通过与上面的优秀智能体对弈的过程中不断提升自己的游戏水平。

② 简单策略对手：我们设置了一个拥有很简单的策略的对手，通过牌力计算如果当前能够观察到的牌牌力很大，那么会以50%的概率选择raise，以50%的概率选择call，保证不会弃掉很好的牌，而其他情况40%的概率选择raise，40%的概率选择call，20%的概率选择fold，这个偏随机的对手的设置的原因主要是为了让智能体能够应对那些经常采用欺诈策略的对手，即手里的牌力很小，但是却选择加注或者跟注的策略。

③ 自我对弈：自我对弈的目的一方面是为了调整超参数，优化方法的选择，以达到最优的效果，实验智能体与对战智能体采用相同的网络结构，相同的初始化网络权值，不同的是超参数的值或者不同的优化方法，主要包括学习率，dropout的参数，激活函数的选择，梯度下降算法。然后进行自我对弈，根据对弈的结果决定采用怎样的参数和优化方法。另一方面是通过自我对弈，自我发现存在的问题，不断调整策略达到一个比较平衡的状态。同时我们在训练的过程中会保留一些阶段性的不同版本，当我们训练好一个智能体，可以与过去的版本进行对弈，通过分析对弈结果，比较不同版本的改变会带来哪些影响。

4.1.3 改进的深度强化学习网络结构

图4-5为改进的深度强化学习网络结构，输入层为一张$17\times17\times16$的三维矩阵。网络中包括三个卷积层，第一个隐藏层包含32个5×5的卷积核，步长为2，第二个隐藏层包含64个3×3的卷积核，步长为2，第三个隐藏层包含64个2×2的卷积核，步长为1。网络中没有pooling层，因为pooling最直接的作用是引入了不变性，更关注是否存在某些特征而不是特征具体的位置。pooling层可以看作加了一个很强的先验，让学到的特征要能容忍一些的变化，但我们的网络的输入层中牌值大小与网络中的位置有直接的联系，因此不能忽略位置信息，所以不能使用pooling。紧接着卷积层的是LSTM层，大小为256，最后是大小为256的全连接层和一个softmax分类器，最后输出三个动作对应的Q值。

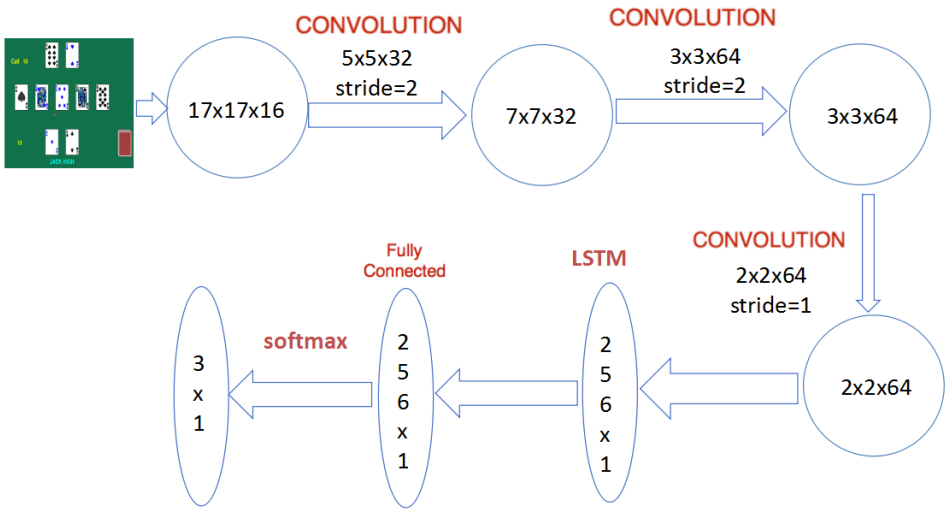

图4-5 改进的深度强化学习网络结构

（1）本文使用 Rectified Linear Unit （ReLU）[53]做为卷积层的激活函数，ReLU是一种近似生物神经激活函数，它的具体计算公式如公式（4-1）：

$$y_i = \begin{cases} x_i & if\ x_i \geq 0 \\ 0 & if\ x_i < 0 \end{cases} \tag{4-1}$$

图4-6为ReLU的坐标系表示，ReLU更容易学习优化，因为其分段线性性质，导致其前传、后传和求导都是分段线性，而传统的Sigmoid函数，由于两端饱和，在传播过程中容易丢弃信息。同时因为ReLU的在反向传播求误差梯度时不涉及除法，计算量相对较小，因此收敛速度更快。由ReLU的计算公式（4-1）可知ReLU能够使隐藏层的一部分神经元输出为零，这种神经元权值的稀疏性一定程度上能够防止过拟合的出现，所以实验效果也较好。

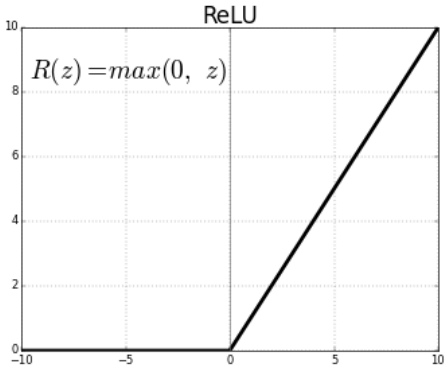

图4-6 ReLU激活函数

（2）本文使用的梯度下降算法是Adagrad。Adagrad的思路是基于：针对于每一个参数都有他自己的学习速率，对于不常出现的参数，每次更新大一点；对于经常出现得参数，每次更新小一点。这种优化算法，在稀疏矩阵中应用比

较多。

（3）深度学习训练时最常见的一个问题是过拟合，即训练出的网络在训练集上有很好的效果，但是在验证集上表现的结果却差强人意。为了防止过拟合，常用的方法主要有扩充数据集、正则化、Dropout[55]等。其中 Dropout是现在研究人员最常用的方法之一。Dropout主要思想是每次迭代不是更新网络中的所有参数，而是随机的丢掉部分参数。本文将Dropout加在了全连接层之后，设置的Dropout率为0.5，即每次更新时会随机的将50%的参数设置为零。Dropout随机的丢掉部分参数，随机性的引入增强了网络的泛化能力，防止了过拟合的发生。

（4）其他一些参数设置如表4-1所示。

表4-1 改进的深度强化学习网络参数设置

参数	大小	描述
minibatch	32	每次从回放容器中抽取的样本数量
回放容器	500000	回放容器的容量
折扣因子 γ	0.99	Q值更新时的折扣因子
初始的 ε	0.9	ε -贪心探索的初始化 ε 大小
终止的 ε	0.0	ε -贪心探索的终止的 ε 大小
学习率	0.000001	参数更新的学习率

4.2 实验结果分析

首先对两种网络结构（深度强化学习网络和改进的深度网络）进行了实验对比。然后比较了不同的梯度下降算法对结果的影响，并对可能的原因进行了分析。最后实验智能体与其他一些智能体进行了对弈，并根据比赛结果进行了分析。

4.2.1 两种网络效果对比

为了对比加入LSTM的网络LSTM_DQN与只是深度强化学习的网络DQN的效果差异，两种网络智能体进行相互对弈，同时保证除了网络结构不同，其他参数都是相同的。图4-7是DQN的网络结构。

通过获得筹码的多少来验证效果的好坏，获得的筹码越多，智能体的水平越高，效果越好。德州扑克的比赛中，在下注顺序上分为大盲位和小盲位。为了比赛的公平起见，会设置一个比赛种子，比赛种子的作用是随机发牌的种子，因此，相同种子情况下每一局所发的牌是不变的，交换参赛者的次序，使参赛

的选手能够避免顺序上的优势，比赛的胜负完全取决于策略的优劣。

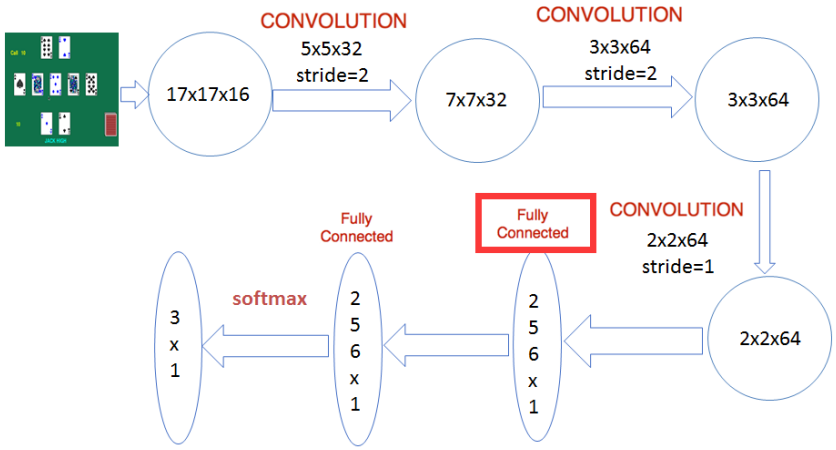

图4-7 DQN智能体的网络结构

比赛使用不同的种子交换顺序进行3000局，比赛结果如表4-2，可以看出，无论设置什么样的种子，本论文实现的智能体在3000局比赛中都能获得胜利。

表4-2 LSTM_DQN与DQN的对比实验

随机种子	DQN	LSTM_DQN
1（小盲位）	-11420	11420
1（大盲位）	-9040	9040
234（小盲位）	-15234	15234
234（大盲位）	-13515	13515
6666（小盲位）	-19720	19720
6666（大盲位）	-15731	15731

不设置随机种子时，每次都是随机发牌进行比赛。 LSTM_DQN智能体与DQN智能体分别进行1000局，3000局，5000局，10000局比赛结果如图4-8所示。

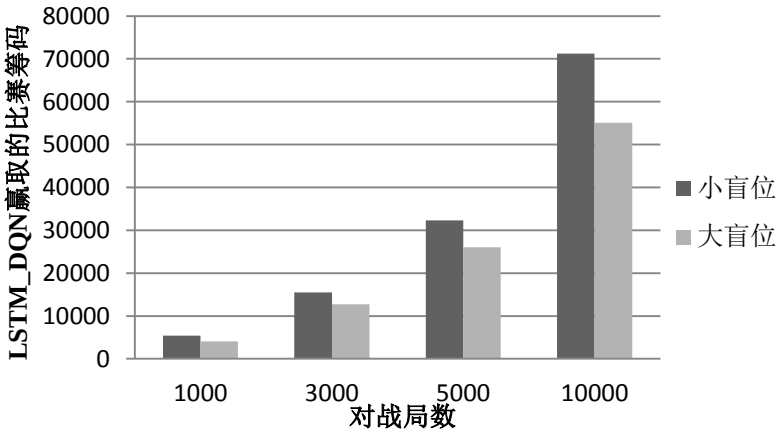

图4-8 LSTM_DQN智能体与DQN智能体

从结果中可以看出，无论进行比赛对局多还是少，大盲位还是小盲位，加入LSTM的智能体LSTM_DQN都可以战胜只是DQN的智能体，因此加入LSTM层适合于非完备信息机器博弈中。

4.2.2 不同梯度下降算法实验效果对比

为了选择最适合于当前网络的梯度下降算法，需要对比不同的梯度下降算法对实验结果的影响。

不同的梯度下降算法实现的智能体与ACPC官方智能体进行对弈，通过每局获得的平均收益（平均收益=当前赢得的总筹码÷当前完成的局数）。来确定不同参数对结果的影响。网络中的其他参数设置都是相同的。

对比不同的梯度下降算法，包括随机梯度下降算法SGD、RMSprop、Adagrad和Adam。从图4-9可以看出网络中采用Adagrad梯度下降算法能够获得更高的每局平均收益。Dean等人在中指出，Adagrad非常大的提升了梯度下降的鲁棒性，并用于训练大规模神经网络[54]，所以实验效果也表现的更好。

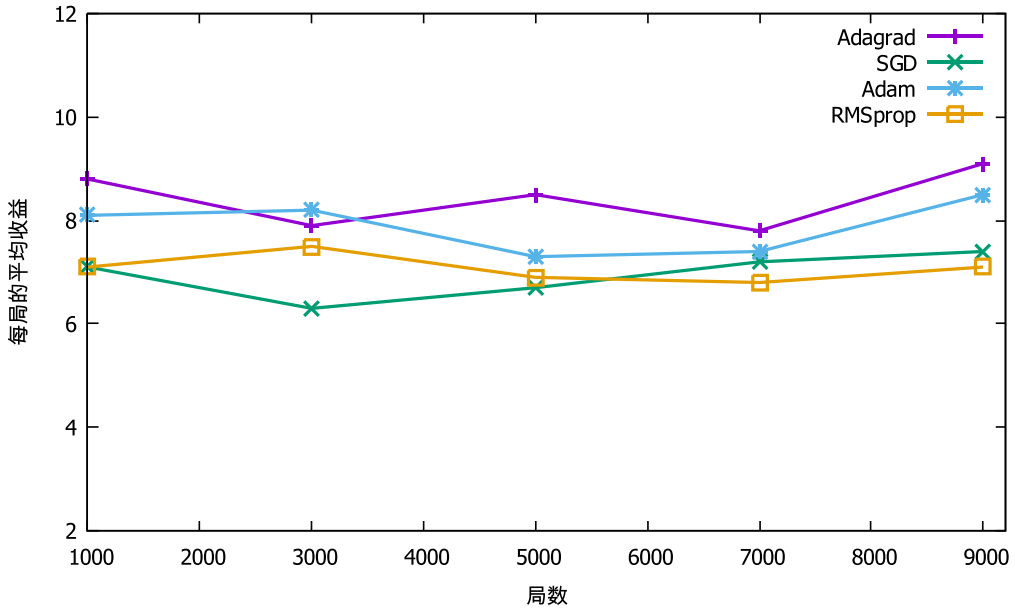

图4-9 采用不同梯度下降算法的效果对比

4.2.3 改进的深度强化学习智能体与其他智能体对比

进行对比测试的其他智能体包括：ACPC官方提供的智能体，基于对手建模算法的智能体[56]（获得了2013年ACPC二人限制性德州扑克第四名）；以人工神经网络作为估值函数的智能体[57]（获得2014年ACPC二人限制性德州扑克第四名）；基于CFR算法和对手建模的智能体（获得2016年ACPC二人非限制性德州

扑克第四名）；基于Q学习算法的智能体；以及采用卷积神经网络监督学习进行训练的智能体[31]。

为减少实验误差，所有比赛都采用相同的种子，相同的种子玩家获得的牌也是相同的，即输赢完全取决于玩家的策略。

图4-10显示了本论文的智能体与其他6个不同的对手进行博弈时，每局的获得的平均收益（平均收益=当前赢得的总筹码÷当前完成的局数）。

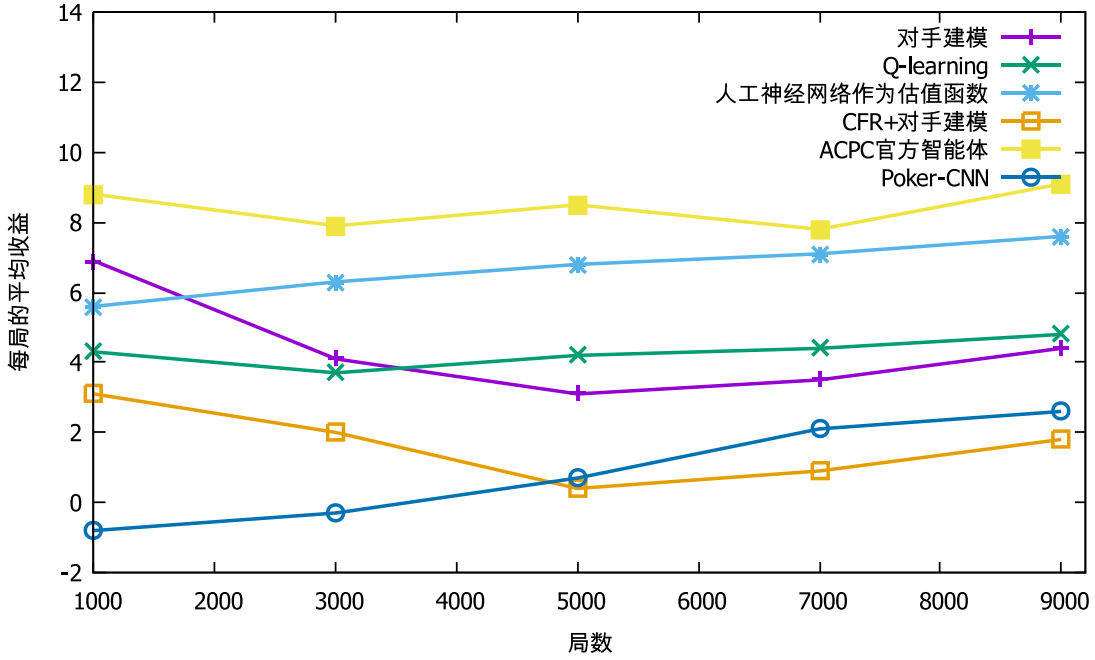

图4-10 实验智能体与其他智能体对弈每局获得的平均收益

根据图4-10可知，改进的深度强化学习算法实现的智能体与其他智能体相比整体上是比较优的。ACPC官方智能体的策略没有偏向性，同时策略里面带有部分的随机策略，所以本论文智能体获得的平均收益相对平稳；与Q学习算法实现的智能体对弈时，结果也相对平稳，但是因为Q学习在一些没有遇到过的局面时无法选择正确的动作，因此在这些局面出现时，改进的深度强化学习智能体能够更大的概率赢得本局的筹码；与对手建模相关的算法实现的智能体对弈，开始的时候本论文智能体赢得的筹码数比较多，因为起初的时候基于对手建模的智能体由于不清楚对手的策略所以采用相对保守的策略，所以许多时候得不到很好的结果。而随着局数的增加对对手的策略认识加深，输掉的筹码也相对减少。但是局数再增加，基于对手建模的智能体输掉的筹码相对又增加了，可以看出深度强化学习智能体在面对对手策略的调整时有一定的适应调整能力；以人工神经网络作为估值函数或者用卷积神经网络来做监督学习的两个智能体随着局数的增加输掉的筹码越来越多，主要是因为监督学习训练出来的智能体

无法动态调整自己的策略。而深度强化学习实现的智能体可以根据比赛的结果实时训练，实时调整，因此一定程度上深度强化学习实现的智能体有根据不同的对手进行策略调整的能力。

4.3 本章小结

本章主要以二人德州扑克为代表，介绍了如何将改进的深度强化学习算法应用到非完备信息机器博弈系统中并分析了实验结果。首先介绍了德州扑克机器博弈系统框架、系统的训练方式以及LSTM_DQN具体的网络结构，最后通过不同结构网络和不同的梯度下降算法的效果对比，验证网络结构的合理性。与其他ACPC大赛历年参赛的部分智能体对比，验证了LSTM_DQN算法的有效性。

结　论

机器博弈中的非完备信息机器博弈是人工智能领域的研究热点，也是检验人工智能发展水平的重要依据。对非完备信息机器博弈进行研究不仅可以带来巨大的商业价值，还可以促进人工智能的发展。因此，越来越多的研究者投入到非完备信息机器博弈的研究中。强化学习等解决非完备信息机器博弈的传统算法面临的主要问题是高维度的信息状态空间，仅通过有限的数据和反复测试无法遍历到所有的情况。另一方面，传统方法需要手动提取特征，很难发现特征间的内在联系。同时训练时需要大量的领域知识，可扩展性差。本论文针对以上问题，采用改进的深度强化学习算法解决非完备信息机器博弈问题，实现了从感知到动作的端对端学习，无需像传统算法一样要将大量的特征提取出来。同时仅需要很少的领域知识，与传统的强化学习算法相比，达到了更高的博弈水平。概括来说，本论文的主要贡献有如下4个方面：

（1）本论文将深度强化学习应用到非完备信息机器博弈中，用深度学习网络替换了强化学习中的状态-动作值函数。同时针对深度强化学习算法无法考虑历史信息的问题，用全连接的LSTM层替换深度强化中的全连接层，综合卷积层提取出的特征与LSTM中保存的记忆计算出各个动作对应的Q值。

（2）本论文提出了基于蒙特卡洛树搜索的回报函数，通过比较每局得到的收益与蒙特卡洛博弈树搜索得到的期望收益，判断应该奖励智能体，还是应该惩罚智能体。

（3）本论文提出了适合于深度强化学习等模式匹配算法的扑克建模方法，主要的实现方式是将手牌，公共牌，阶段信息，上下文信息编码成16张大小为17×17的0-1矩阵，然后将这个$17\times17\times16$的三维矩阵作为深度强化学习网络的输入。通过二人德州扑克、2-7三次换扑克和Kuhn扑克说明了这种建模方式的可扩展性。同时，这种编码方式只需要很少的领域知识便可以将相同的深度强化学习网络应用于不同的扑克游戏，实现从零开始的训练学习。

（4）本论文将改进的深度强化学习算法应用到非完备信息机器博弈中，实现了德州扑克机器博弈系统。系统的训练过程不需要人为参与，不需要手工提取特征，完全让智能体自己去学习如何玩德州扑克，仅需要很少的领域知识。同时相较过去参加ACPC大赛取得优异成绩的智能体，博弈水平有了较大提升。深度强化学习能够为大规模机器博弈系统的实现提供了一个可行的方法，同时为将算法扩展到现实生活中提供了可能。

虽然基于改进深度强化学习在非完备信息博弈取得了一定的进步，但是因

为加入了LSTM层，所以训练速度相对来说较慢，隐藏层层数较少一定程度上影响了实验的效果，如何优化网络，使得训练速度加快以及准确性提升还需要进一步研究。同时德州扑克机器博弈系统中学习了德州扑克中的三个动作：加注、跟注和弃牌。但是对加注时的筹码数量没有进行学习，现在仍是基于规则的，后面可以考虑结合actor-critic算法[58]与深度强化学习算法，实现在连续空间的动作选择。

参考文献

[1] Pomerol J C. Artificial Intelligence and Human Decision Making[J]. European Journal of Operational Research, 1997, 99(1): 3-25.

[2] Rhalibi A, Wong K W. Artificial Intelligence for Computer Games: An Introduction[J]. International Journal of Computer Games Technology. 2009, 12(3): 351-369.

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

[4] Riedmiller M, Gabel T, Hafner R, et al. Reinforcement learning for robot soccer[J]. Autonomous Robots, 2009, 27(1): 55-73.

[5] Gelly S, Kocsis L, Schoenauer M, et al. The grand challenge of computer Go: Monte Carlo tree search and extensions[J]. Communications of the ACM, 2012, 55(3): 106-113.

[6] Bowling M, Burch N, Johanson M, et al. Heads-up limit hold'em poker is solved[J]. Science, 2015, 347(6218): 145-149.

[7] 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.

[8] Lambert Iii T J, Epelman M A, Smith R L. A fictitious play approach to large-scale optimization[J]. Operations Research, 2005, 53(3): 477-489.

[9] Nevmyvaka Y, Feng Y, Kearns M. Reinforcement learning for optimized trade execution[C]//Proceedings of international conference on Machine learning, 2006: 673-680.

[10] Bazzan A L C. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control[J]. Autonomous Agents and Multi-Agent Systems, 2009, 18(3): 342-375.

[11] Tambe M. Security and game theory: algorithms, deployed systems, lessons learned[M]. Cambridge University Press, 2011.

[12] Urieli D, Stone P. TacTex'13: a champion adaptive power trading agent[C]//Proceedings of international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, 2014: 1447-1448.

[13] Durkota K, Lisý V, Bošanský B, et al. Approximate solutions for attack graph games with imperfect information[C]// Proceedings of Decision and Game Theory for Security, Springer International Publishing, 2015: 228-249.

[14] Selten R. Bounded rationality[J]. Journal of Institutional and Theoretical

Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft, 1990, 146(4): 649-658.

[15] Ariely D. Predictably irrational[M]. New York: HarperCollins, 2008.

[16] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013, 154(4): 67-69.

[17] Bernstein A, Roberts M. Computer v Chess-player[J]. Scientific American, 1958, 198(6): 96-105.

[18] Berliner H. Backgammon Computer Program Beats World Champion[J]. Artificial Intelligence. 1980, 14(2): 205-220.

[19] Campbell M, Joseph H, Feng H. Deep Blue[J]. Artificial Intelligence, 2002, 134(1): 57-83.

[20] Billings D, Burch N, Davidson A, et al. Approximating Game-Theoretic Optimal Strategies for Full-scale Poker[C]//Proceedings of International Joint Conference on Artificial Intelligence, 2003: 661-668.

[21] Rubin J, Watson I. Computer poker: A review[J]. Artificial Intelligence, 2011, 175(5): 958-987.

[22] Papp D. Dealing with Imperfect Information in Poker[M]. University of Alberta, 1999, 12(24): 212-230.

[23] Billings D, Davidson A, Schauenberg T, et al. Game-tree Search with Adaptation in Stochastic Imperfect-Information Games[M]. Springer Berlin Heidelberg, Computers and Games, 2006: 21-34.

[24] Zinkevich M, Johanson M, Bowling M, et al. Regret Minimization in Games with Incomplete Information[C]//Proceedings of Advances in Neural Information Processing Systems, 2007: 1729-1736.

[25] Bellman R. A Markovian decision process[R]. RAND CORP SANTA MONICA CA, 1957. [30] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[26] Watkins C. Learning from Delayed Rewards[D]. University of Cambridge, Doctoral dissertation, 1989: 210-215.

[27] Tesauro G. Temporal Difference Learning if Backgammon Strategy[C]// Proceedings of the Ninth International Workshop on Machine Learning, 2014: 451-457.

[28] Dahl F A. A Reinforcement Learning Algorithm Applied to Simplified Two-Player Texas Hold'em Poker [C]// Proceedings of European Conference on Machine Learning 2001, 2001: 85-96.

[29] Broeck V d, Driessens K, and Ramon J. Monte Carlo Tree Search in Poker Using Expected Reward Distributions[M]. Advances in Machine Learning,

Springer Berlin Heidelberg, 2009: 367-381.

[30] Iolis B, Bontempi G. Comparison of Selection Strategies in Monte Carlo Tree Search for Computer Poker[C]// Proceeding of the Annual Machine Learning Conference of Belgium and The Netherlands, BeNeLearn, 2010: 1-4.

[31] Luís Filipe Teófilo, Nuno Passos, Luís Paulo Reis, Henrique Lopes Cardso. Adapting Strategies to Opponent Models in Incomplete Information Games: A Reinforcement Learning Approach for Poker[M]. Autonomous and Intelligent Systems, Springer Berlin Heidelberg, 2012: 220-227.

[32] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

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

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

[35] Luís Filipe Teófilo, Luís Paulo Reis. Building a No Limit Texas Hold'em Poker Playing Agent based on Game Logs Using Supervised Learning[M]. Autonomous and Intelligent Systems, Springer Berlin Heidelberg, 2011: 73-82.

[36] Yakovenko N, Cao L, Raffel C, et al. Poker-cnn: A pattern learning strategy for making draws and bets in poker games using convolutional networks[C]// Proceedings of AAAI Conference on Artificial Intelligence, 2016: 144-152.

[37] He S, Wang Y, Xie F, et al. Game Player Strategy Pattern Recognition and How UCT Algorithms Apply Pre-knowledge of Player's Strategy to Improve Opponent AI[C]//Proceedings of Computational Intelligence for Modeling Control & Automation, 2008: 1177-1181.

[38] 马骁, 王轩, 王晓龙. 一类非完备信息博弈的信息模型[J]. 计算机研究与发展, 2011, 47(12): 2100-2109.

[39] 王轩, 许朝阳. 时序差分在非完备信息博弈中的应用[C]. 中国机器博弈学术研讨会, 2007: 16-22.

[40] Zhang J, Wang X, Yang L, et al. Analysis of UCT Algorithm Policies in Imperfect Information Game[C]//Proceedings of Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on IEEE, 2012: 132-137.

[41] Zhang J. Building Opponent Model in Imperfect Information Board Games[J]. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2014, 12(3): 1975-1986.

[42] Zhang J, Wang X. Using Modified UCT Algorithm Basing on Risk Estimation Methods in Imperfect Information Games[J]. International Journal of

Multimedia and Ubiquitous Engineering, 2014, 9(10): 23-32.
[43] Rabinowitz P H. Minimax Methods in Critical Point Theory with Applications to Differential Equations[M]. American Mathematical Society, 1986: 85-100.
[44] Chen B, Ankenman J. The mathematics of poker[M]. ConJelCo LLC, 2006，8(12): 21-25.
[45] Heinrich J, Silver D. Smooth UCT Search in Computer poker[C]//Proceedings of International Joint Conference on Artifical Intelligence, 2015: 554-560.
[46] Bowling M, Veloso M. Multiagent Learning Using a Variable Learning Rate[J]. Artificial Intelligence, 2002, 136(2): 215-250.
[47] 郭茂祖, 刘扬, 黄梯云. 加强学习主要算法的比较研究[J]. 计算机工程与应用, 2004, 37(21): 16-18.
[48] Watkins C J, Dayan P. Q-learning[J]. Machine Learning, 1992, 8(3): 279-292.
[49] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series[J]. The handbook of brain theory and neural networks, 1995, 3361(10): 1995.
[50] Johanson M. Measuring the size of large no-limit poker games[J]. Computer Science, 2013, 4(5): 23-25.
[51] Lin L J. Reinforcement learning for robots using neural networks[M]. Carnegie Mellon University, 1992: 21-34.
[52] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[53] Lennie P. The cost of cortical computation[J]. Current biology, 2003, 13(6): 493-497.
[54] Dean J, Corrado G, Monga R, et al. Large scale distributed deep networks[C]//Advances in neural information processing systems, 2012: 1223-1231.
[55] 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.
[56] 吴松. 德州扑克中对手模型的研究[D]. 哈尔滨: 哈尔滨工业大学, 2013:19-23.
[57] 李景鹏. 非完备信息博弈估值算法的研究[D]. 哈尔滨: 哈尔滨工业大学, 2014:6-14.
[58] Konda V R, Tsitsiklis J N. Actor-Critic Algorithms[C]//Proceedings of Conference on Neural Information Processing Systems. 1999, 13: 1008-1014.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于深度强化学习的非完备信息机器博弈研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：王鹏程　　日期：2017年1月4日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：王鹏程　　日期：2017年1月4日

导师签名：王东　　日期：2017年1月4日

致　谢

时光匆匆，研究生生活不知不觉中已经到了结尾的阶段，同时也意味着自己的学生时代进入了告别阶段。此时的我，除了有对学校的不舍，更多的还是想对关怀和帮助过我的人表达最衷心的感谢！

首先，衷心感谢我的导师王轩教授。在读研两年半期间，王老师给予我许多锻炼机会，从中我学到了很多宝贵的知识和经验。在科研方面，王老师悉心指导我课题研究，带领我进入非完备信息机器博弈的科学世界。并且每周都会组织课题组讲解一些机器博弈方面的前沿文章，帮助我把握课题的方向。在研究遇到瓶颈时，王老师的鼓励，让我有充足的信心和勇气去克服课题中的困难。在日常生活中，王老师尽可能多的给我机会去参加社会活动，锻炼与人沟通的能力，从中学到了许多做人做事的道理，令我受益匪浅。王老师还时常带领我们爬山打球，不仅使我的研究生生活丰富多彩，更强健了体魄。

同时，我也要衷心感谢蒋琳老师，课题的研究和论文的撰写都离不开蒋琳老师的耐心指导。在论文撰写的过程中，蒋老师细心修改我的论文，提醒我需要注意的问题，给了我很大的帮助。

此外，我要感谢实验室的所有师兄师姐们，特别要感谢张加佳师兄在论文的选题和研究方案的制定上给予的帮助。张加佳师兄还经常抽出宝贵的时间与我探讨课题研究中出现的问题，让我少走弯路。这里我还要感谢代佳宁、朱航宇、林云川、胡开亮同学在项目中予以我的帮助和鼓励。感谢14级实验室的同学，在研究生期间给予我的帮助。

最后，我要感谢我的家人，感谢他们在物质和精神上的支持。是他们在背后默默的支持，使我能够一心一意投入学习，顺利完成学业。