# 绪论

## 1.1 课题背景及研究的目的和意义

无人机被越来越广泛的使用于军事、农业、生产、物流和娱乐。执行任务时，无人机的真实飞行环境和预设飞行环境常有不同。除了已知的地形和静态障碍，还有由于信息更新不及时产生的未知静态障碍和动态威胁源。能根据实际情况临时修改飞行状态，自主规避出现在预定航迹上的静态/动态障碍的无人机，可以应对更丰富的应用场景和更复杂的飞行环境如：可以实现人和无人机在同一环境中工作；可以实现相互没有通信的无人机、机器人、机械臂等在同一环境中工作；可以在执行任务途中避免突然出现的飞鸟，行人，其他飞行器和地形改变带来的损耗，提高任务的成功率。

（此处应用场景可扩写）

## 1.2 国内外研究现状和发展趋势

基于飞行安全的需要，无人机的路径设定需要综合考虑威胁场约束和无人机自身性能约束等条件，所规划出的航迹既要尽可能减少无人机在飞行中与障碍物发生碰撞的概率，又要使得综合性能指标最小，并根据实际需要进行飞行中的局部调整。作为组合优化问题研究的热点之一的，其精度及速度往往直接影响到任务的效果，国内外专家也展开了深入的研究，并取得了相应的显著成果。

### 1.2.1 智能避障算法研究现状

对于航迹规划中地形和威胁的规避,S.J.Asseo提出了对地形要求不高且收敛速度快的最速下降法 ；任敏提出效率较高的Hopfield神经网络法 ；AminJN等基于遗传算法展开研究；Szczerba等选择优化经典的最优启发搜索算法A\*算法；白俊强等选择优化蚁群算法。以上研究算法相对成熟，但主要针对静态航迹规划，在真实的环境情况下存在状态信息不断变化的威胁源，因此动态航迹规划问题更符合实际情况，能更好地体现航迹规划的自主性。目前，不确定环境下无人机的动态航迹规划问题仍是研究的难点，该问题要求依据威胁状态信息的变化情况，在允许的时间内找到代价小的可行航迹。

针对实时航迹规划问题，国外Iris Hong Yangy和James Doebble获得了可自动躲避地形障碍物的航迹，但未考虑除地形之外的其它威胁；Stentz提出D\*算法进行局部路径修正，耗时短，但威胁模型同样不能反映真实环境。

国内，龙涛针对二维航迹规划问题，将威胁代价表示为距离的函数，可同时达到顺利规避威胁和航程时间短的战术要求，但并不太适用于求解考虑地形因素在内的三维空间航迹规划。

严平提出了分阶段思想的多任务动态航迹规划算法，但对威胁的具体类型未做相关描述，不能获得准确的威胁代价。

（可改进，要注意引用文献）

### 1.2.2 强化学习算法研究现状

强化学习是智能体学习如何将环境状态映射到行为动作才能获取最大回报的过程，是一种奖惩式的学习方式，智能体寻求能够产生较大奖赏的动作，同时避免低奖赏的动作，是一种适应性比较强的学习方法。近年来，强化学习方法成为智能系统设计的核心技术之一，被广泛应用于人工智能和机器人学等领域。由于众多专家学者不断深入研究强化学习算法，并作出许多杰出贡献，使得该方法在智能机器人领域得到非常广泛的应用且取得突飞猛进的发展，该方法实现移动机器人的路径规划和提高对环境的自适应性成为机器人学领域研究和应用的热点之一[6]。

（可改进，要注意引用文献）

强化学习在心理学中指的是有机体怎样在环境给出的奖励或着惩罚刺激时渐渐养成对刺激的预期，从而发生可以得到最优利益的习惯性行为。此方法有普适性，所以在大量别的领域也会研究，如博弈论、多主体系统学习、运筹学、遗传算法、信息论、模拟优化方法、群体智能、统计学以及控制论[9]。强化学习概念由来已久，1954年由Minsky提出。最开始叫做随机神经模拟强化计算器。1957年贝尔曼提出贝尔曼方程，运用动态规划来解决马尔科夫决策问题带来的最优控制问题。在1989年，Watkins提出了Q学习方法，极大地提高了强化学习应用的广度与范围。

强化学习是一个序列决策问题，它要做的是不断地选择某些行为，然后从这些行为完成后取得的最优结果来作为最后的成果[10]。它在无标签反馈给算法如何做的情景下，通过试着改变一些行为，而后取到一个结果，由判断的结果是对是错，从而对之前的行为进行反馈。通过反馈调整前面的行为，通过持续的调整算法可以学习到在特定的情况下选择特定的行为从而得到最优的结果。

强化学习就是智能体（Agent）从环境到行为映射的学习，最终使得奖励信号函数值最大[11]。如果智能体的一个行为策略引起环境出现正面的奖赏信号，那么智能体以后产生这个行为策略的趋势就会变得更强，反之则趋势降低。

随着最近令人兴奋的深度学习成果，受益于大数据，强大的计算和新的算法技术，目睹了强化学习的复兴，特别是强化学习和深度神经网络的组合，即深度强化学习（Deep Reinforcement Learning）。基于深度强化学习的深度 Q 学习优先经验重放算法在智能体决策中的运用逐步增多。

深度强化学习算法是将深度学习与强化学习的特点结合起来形成了新的高效人工智能学习算法[30]。它可以运用到诸如机器人行为控制，汽车自动驾驶，图形游戏控制等诸多领域研究之中。

## 1.3 主要研究内容

在实际的环境中，无人机可能会遇到包括静态的和运动的在内的各种威胁源，无法提前获得规划区域内的全部信息。为了使无人机具备自主飞行规划能力来绕过这些威胁，需要根据当前周围的环境信息来自动选择一条较优路径，也就是进行航迹的实时调整，从而实现不确定环境下的动态航迹规划。

本文研究的不确定干扰下无人机航迹动态规划问题主要包括无人机与干扰无通信下的规避动作决策。

我们的目的是为了解决用目标（吸引子）和障碍物（驱避剂）可以描述的机器人的任务，这些机器人通过具有未知动力学加速度控制。我们要求强化学习的解决方案兼具高效和适应性。高效意味着偏好估计增强学习可以在完全连续和无限的物理空间内实时控制机器人;适应，意味着单次学习可以应用于许多任务。

我们将无人机当做一个外力引发运动的系统，将其运动建模成一种特殊的非线性系统。假设一个机器人有*m*个自由度，如果在时间*n*，将一个加速度施加于机器人的质心，新的位置速度向量（状态）如式1-1

D: s(n+1)=f(s(n))+g(s(n))a(n) (1-1)

一个马尔科夫决策过程（一个(S, A, D, R)组合，状态空间,动作空间），如1-1式构成了一个任务，即时为状态空间S中的每一个状态的分配奖励R:S→R[2]。一个马尔科夫决策过程的解是一个从状态空间S到行动空间A的映射π。在机器人更新周期中获得的累加奖励最大化。将奖励设为：

(1-2)

0≤γ≤1是折扣系数。近似值迭代法通过把状态和奖励值的方程近似成一个线性映射的方法求得连续状态马尔科夫过程的解，如式1-3。

(1-3)

近似值迭代法选用特征向量F(s)，然后通过在状态空间采样并观察奖励这种方法学习得到不同特征间的权重。它以期望最大化的方式迭代地更新θ。

参数学习结束以后，强化学习进入计划阶段。规划者已知式1-3的近似奖励函数和初始状态，并使用关于状态奖励函数的贪心策略的闭环控制来生成轨迹，其中状态s'是将动作a应用于状态s后的结果。遵照式1-3贪婪策略的连续空间中的动作选择，是对未知函数的多变量优化。存在几种基于抽样的有效近似策略，例如分层最优化树（Hierarchical Optimistic Optimization applied to Trees 简称HOOT）。 HOOT使用分层离散法逐渐缩小对输 入空间最有希望的区域的搜索，从而确保错误尽可能小[19]。在实践中，HOOT适用于具有许多小规模最大值的奖励函数的单一机器人规划。

### 1.3.1研究方案

偏好评估增强学习解决偏好平衡问题需要两个阶段，学习和行动(如图1-1）。

##### 图1-1 偏好评估增强学习框架

为了进行学习，用户提供有关该问题的基本信息：机器人的自由度，最大加速度等，以及一组目标（偏好）。 基本系统信息被编码为马尔科夫决策过程，如第1.3.2节所示。同时，考虑到由任务目标和障碍组成的偏好，偏好评估增强学习使用第1.3.3节中描述的方法生成特征。

马尔科夫决策过程和特征设置好后，学习阶段在简化的问题空间中使用基于近似值迭代的增强学习算法之一来发现偏好之间的相对权重（偏好评估）。与（3）近似的值函数是基于偏好的特征的线性映射。一旦学习到了偏好权重，行动阶段可以开始了。

行动过程是一个闭环反馈系统，可以在线决策或在模拟中离线计划轨迹。 因为特征捕捉到了任务中非常关键的元素，所以这些特征在比训练时更大的状态空间和行动空间里依然有效，这允许用户解决更大数量级的问题。这些特征可以同时实现效率和适应，前者通过在小数量级的训练样本上训练，后者通过把规则使用到更大数量级的问题上。是否使用自动生成的特征是偏好评估增强学习和普通增强学习的重要区别，自动生成特征使偏好评估增强学习成为一种几乎无需调试的学习方法。

### 1.3.2马尔科夫决策过程的建立

对于一般的偏好评估增强学习公式，我们假设机器人在连续状态和动作空间中工作。通过施加到质心的加速度来控制。令s,,∈分别作为机器人的位置，马尔科夫决策过程状态空间是S= ,=2,是机器人的自由度。状态空间s∈S是s=[s,的联合矢量 ,动作a∈A=是联合加速度矢量,a=.假设为未知的状态转移方程是一个离散时间仿射系统[18]。

机器人达到目标时，奖励R设为1，否则为0。(S,A,D,R)的组合定义了本机器人问题的马尔科夫决策过程。

在动态避障任务中，马尔科夫决策过程是机器人位置和速度的联合矢量，S =。动作空间A =是二维空间上各方向的加速度。

### 1.3.3特征的选择

对于一般的偏好估计增强学习方程，我们定义偏好平衡任务有个目标:,偏好与目标相关联。目标有自己的位置和速度，存在于位置和速度空间∈,i=1,···，，每个目标要么吸引要么排斥这个机器人。吸引机器人的，我们称其为减小距离偏好，排斥机器人的，我们称其为减小密度偏好。

为了学习有个目标的偏好平衡任务，我们给每个目标设置一个特征，假设低维任务空间和高维马尔科夫决策过程空间, 我们令任务偏向的特征为

（1-4）

特征们将状态空间S和状态空间维度一起映射到偏好空间，并根据偏好类型，测量与目标间的直线距离或单位面积密度，让（s）做位将机器人的状态投射到包含的最小子空间上的投射。例如，如果目标 是位置空间中的点（s）即是机器人的位置，同样的，如果 是速度空间上的点（s）即是机器人的速度。

定义减小距离偏好的特征为：

(1-5）

定义减小密度偏好的特征为：

（1-6）

算法1总结了特征选择的过程。

|  |
| --- |
| 算法1 偏好估计增强学习特征选择 |
| 输入 偏好类型目标 |
| 输入 马尔科夫决策过程(S, A, D, R), |
| 输出  1. for i = 1, . . . , do  2. if is intensity then  3. *{intensity preference}*  4.\else  *5.* *{distance preference}*  6.end if  7,end for  8 return |

##### 表1 偏好学习算法流程

在动态避障任务中，有两种自然偏好，1）减少与目标间的距离，2）增大和障碍间的距离。结合两种偏好来构建特征向量，  
 （1-7）

根据算法1，提供目标坐标和障碍物坐标，特征定为：

, （1-8）

G是目标位置，是机器人位置，d是与最近障碍的最小距离，β是根据经验选择的常量0.01m。

## 1.4 论文的主要工作

本文将深度强化学习算法用于智能体决策问题。首先对深度强化学习算法进行详细的分析介绍，然后会将智能体决策问题改变为经典的马尔科夫决策过程（Markov Decision Process）问题。本文采用改进优先经验重放机制用来解决样本的相关性分布问题。最后，通过设计实验来验证新的改进后的深度强化学习算法与以往经典算法相比是否有对智能体决策效果有所提升。

本文的主要贡献如下：

1.由实验的具体情形搭建了一个深度强化学习的网络架构，该架构能够在繁杂的强化学习环境中通过输入原始的环境参数有效的学习到游戏的控制策略。

2.实现深度强化学习随机经验重放算法与经典的优先经验重放算法的效果验证，同时也为新算法的效果提供了对比案例。

3.此外，提出了一种改进的优先经验重放算法，用于改进经典的深度Q学习（Deep Q-learning Network,DQN）优先经验重放算法,从海量的历史经验池中选出更加有效的训练样本。

4.通过设计实验来进行验证本文所提出的新的改进后的优先经验重放算法对于提升智能体学习操控游戏的能力是否可行。同时通过对比之前的经典算法来对新方法进行评估。

## 1.5 论文组织结构

本文共分为六章，各章主要内容如下：

第一章介绍当前深度强化学习的应用领域与背景，然后简要说明深度强化学习的发展概况与简单原理。随后介绍在智能体决策中如何应用深度强化学习算法。对国内外研究现状进行了简单回顾，最后阐明本篇论文的主要工作与贡献。

第二章开始会对深度强化学习与之前已有的机器学习方法的联系与区别进行分析，然后交代深度强化学习的基本思想与训练过程。然后将引出本文所用到的神经网络结构，对实验中所用到的神经网络的原理和训练方法进行分析说明。接下来将会介绍强化学习算法在智能体决策中应用的现状以及强化学习与监督学习和非监督学习之间的关系。接下来会详细分析强化学习原理还有马尔科夫决策过程，还有就是分析了值迭代与策略迭代的不同算法。

第三章开始将介绍了智能体决策发展现状与未来方向，然后说明强化学习运用到智能体决策的相关情况并将会分析强化学习Q学习算法。接下来分析验证用于智能体决策的深度Q学习与随机经验重放算法还有深度Q学习与优先经验重放算法。

第四章开始将会先介绍深度强化学习背景并分析了深度强化学习中的梯度下降算法与学习率。然后重点分析研究深度Q学习与经验重放，并在之前传统算法的基础上对深度Q学习优先经验重放算法进行改进分析研究，最后会形成改进的深度Q学习优先经验重放算法。

第五章开始将会先介绍实验软件环境，然后描述实验软件程序架构与实验方案设计。随后进行实验结果分析，包括通过一些仿真截图说明改进的算法可以学习到环境信息并做出合理的决策方案，还有各算法网络架构选择，各个算法的代价函数分析，各算法效果分析还有各算法效果对比等，并通过这些对实验结果的分析最后得到结论。

第六章对论文的研究内容与实验结果进行总结，然后说明研究工作的不足之处以及在未来的改进方向。

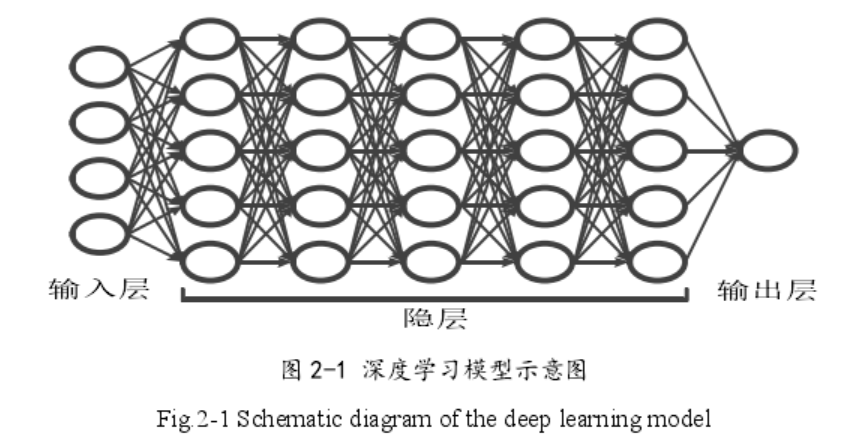
# 第2章 深度强化学习算法研究

强化学习是机器学习的一种。智能体运用强化学习寻找最优策略来解决序列决策问题。这种策略可以决定智能体在碰到每种不同的状态时怎样做出最优的决策选择。用一个数值表示奖励信号或者是惩罚信号，用来逐步改变智能体的决策动作，从而获得最优的回报值。结合深度学习算法形成并运用深度强化学习算法对游戏的操作控制，从而获得更高的得分同时花费更少的游戏时间。因此，很有必要了解强化学习的基本理论。

## 2.1 深度学习算法分析研究

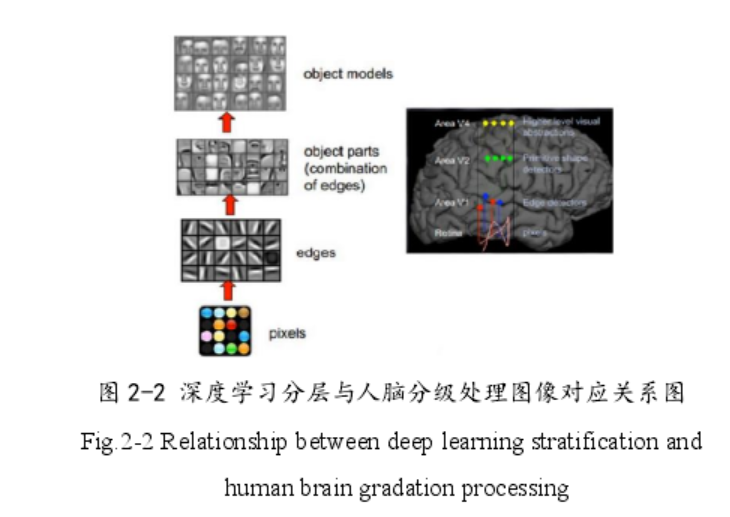
### 2.1.1 深度学习的基本思想与训练过程

深度学习相对于浅层学习只有输入输出层和一层隐含层来说，它含有输入输出层与多个隐含层，如图2-1，每一层隐含层从更低层接收信息并进行训练后将训练之后的信息向更高层传送[12]。与传统的神经网络相对比，深度学习的层数更多，所以会具有更强的表达能力，对复杂目标进行描述时也会更加详尽。



##### 图2-1 深度学习模型示意图

深度学习的这种特点与人脑处理图像时的方式很像[12]。原理上大脑对信息的处理是分级的，可以把大脑皮层划分为不同的区域。Retina区域处于较低层级，对图像的像素特征来进行提取，而比它高一级的层级被称为V1 区域，它负责对目标的边缘特征进行识别并进行提取，比V1区域高一级的层级被称为V2区域，它负责对目标的形状进行探索、提取，然后在比V2区域更高的层级内对图像所包含的局部特征、整体特征进行探测等。因此可以说目标的高层特征是由它的低层特征的抽象组合而得到的，低层特征的抽象程度越高，获得的高层特征就越能够更清晰的表示目标[13]，存在的不确定也就会越少，更有利于对目标进行描述和分类。



##### 图2-2 深度学习分层与人脑分级处理图像对应关系图

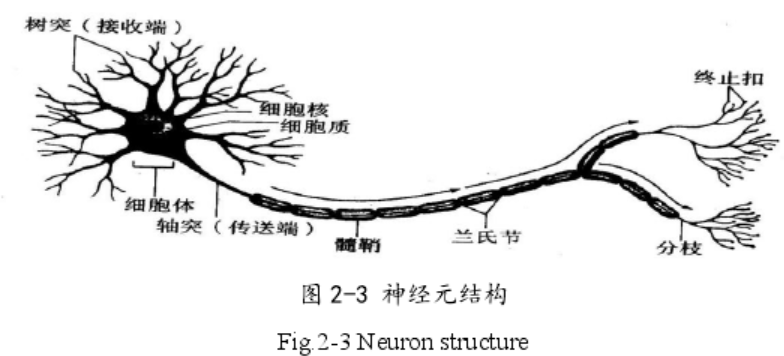
深度学习的概念是对多个层进行分层处理，意思是前一层的输出作为后一层的输入。运用此种办法就能够实现对输入的信号实行分层次传递的效果。

### 2.1.2 神经网络

神经网络（Neural Networks, NN）或者说是人工神经网络，它是由具有适应性的简单单元形成的形成网状互连的网络，它的结构可以仿照生命神经系统对真实物体做出的交叉反应。能够大量服务于图像识别，语音识别以及对自然语言的处理中去。

#### 2.1.2.1 人工神经网络单元

人类大脑中的神经网络单元是神经元，它由细胞体、树突、轴突和突触等组成。如图2-3所示。



##### 图2-3 神经元结构

人工神经网络神经元其实比较简单，各个神经元与其它神经元之间的连接是具有权重的，它能够和前面神经网络层中的全部相关的神经元都有连接关系。每一组神经元之间的连接都有其各自的权重，通常的情况下是一些随机值[14]。这个权重可以是实数负值，实数正值，可以非常小，也可以非常大，可以是零。和这个神经元相互连接的所有神经元的值都可以乘上各自所对应的权重。然后把这些值用以求和。在这个基础之上，还会额外加上一个bias，可以理解为偏置，它的功能是可以用来避免输出为零的情况发生，并且能够加快一些操作，这让解决某些个问题所需的神经元的数量会有所减少。这个bias也是一个数字，有些时候是一个实数常量，经常是-1或者1，有些时候也会有些变化。这个总和最终被输入到一个激活函数中去，这个激活函数的输出结果最终会成为这个神经元最后的输出。

神经元j的输入向量表示为：

 （2-1）

式2-1中,(i =0, 1, 2,…, n)代表的是第i个神经元的输入，n代表的是输入的神经元个数。

加权向量表示为：

 (2-2)

式2-2中，i, j =0, 1, 2,…, n)代表的是第i个神经元到第j个神经元的权重。

神经元j的阈值为，如果用表示阈值，则其和神经元j的权重为。因此神经元j的n多个输入神经元加权和可以表示为：

(2-3)

则神经元j的输出可表示为：

(2-4)

式2-4中，f(∙)是神经元的激励函数，代表了它的输入和输出之间的关系。

#### 2.1.2.2 激活函数

函数f:R→R,是一个实数的值到值的一种映射关系，一般称之为激活函数。

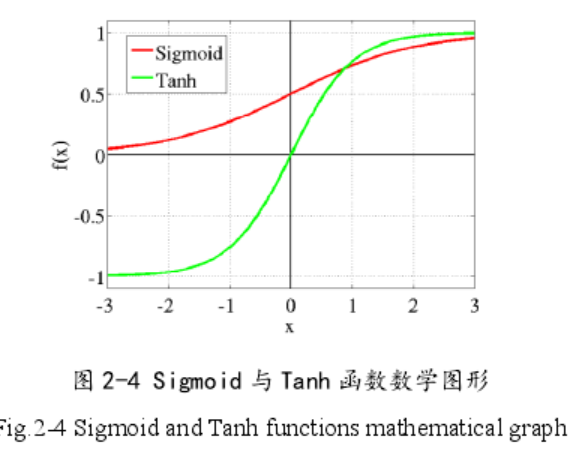
由于神经网络在数学上是可微的[15]，从而所应用的激活函数要能够确保数据输入与输出时也是可微的，运算特征就是不间断地循环计算，从而在每个循环过程中的每个神经元的值也是一直改变的。

在当前大多数的卷积神经网络中，都是运用ReLu函数。另外采用ReLu函数可以有效避免梯度消失问题。

|  |  |
| --- | --- |
| 函数 | 函数关系 |
| sigmoid函数 |  |
| tanh函数 |  |
| ReLU函数 |  |

##### 表2-1 常用的激励函数

如表2-1中所示为常用的几个激活函数模型公式。其中sigmoid函数与tanh函数的数学图形可表示为图2-4，ReLU函数图形为图2-5所示。

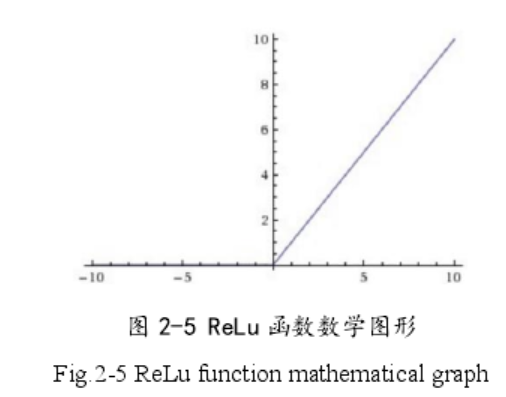


##### 图2-4 sigmoid 与tanh函数数学图形

激活函数不是用来去激活什么，它的作用是如何把激活的神经元的特征通过函数把特征保留并映射出来[16]，保存特征，去掉数据冗余，这是神经网络能解决非线性问题的关键。

由于线性模型的表达力不够，激活函数可以加入非线性因素。已知对于神经网络中的图像，可以采取卷积的方法处理，就是说对于图像中的每个像素点赋予一个权值，这是一个线性操作。而样本不一定是线性可分的，为解决这个问题，能够进行线性变化，或是加入非线性因素，从而处理好线性模型所不能解决的问题。

要注意到网络中的各个层可以使用不同的激活函数，激活函数也有很多形式，其中有几个重要的激活函数需要重点掌握。所使用的激活函数都必须符合一定的条件，即非线性、可微性、单调性、f(x)≈x等条件。

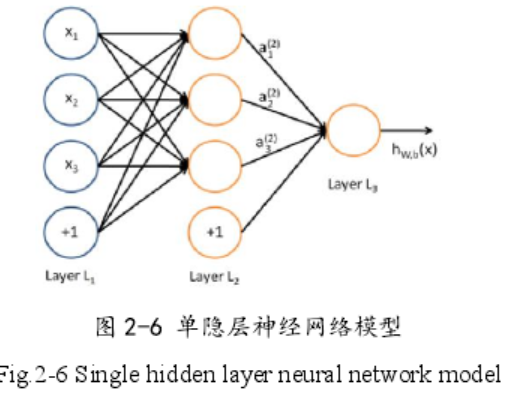


##### 图2-5 ReLu函数数学图形

### 2.1.2.3 神经网络结构

模拟生物的神经网络时，当神经元传递电信号时会向它相连接的神经元发送一些化学成分并改变靠近神经元的电位，并且当电位超过阈值时，神经元就会被激活[17]。人工神经网络可以说是将很多个单独的神经元结合在一块形成的网状结构。通过不同的权重把各个神经元重新连接，多个神经元组合的输出就可以成为另外一个神经元的输入存在。这种的关系能够在各层之间随意组合并传递下去。

如图 2-6 就是一个关于神经网络的实际例子模型：



##### 图2-6 单隐层神经网络模型

图中是输入单元，将原始数据输入给它们，Layer L1表示的是输入层。是中间单元，用来处理数据，然后再传送给下一层，Layer L2表示隐含层。最后是输出单元，它负责计算。Layer L3表示的是输出层。然后，每层网络中标上的“+1”的节点表示的是网络的偏置节点。这种的节点表示的是网络的起始激活程度。第L层激活值为,那么第L+1层的激活值可以按照步骤得到：

(2-6)

(2-7)

深度学习的目标是可以令含有众多隐层的多隐层神经网络能够训练网络。可以更好的协同工作所诞生出来的全新的架构方式。按照定义深度学习的网络架构符合多层神经网络[18]。一般来说多层神经网络只包含输入输出与隐含层。而隐含层到底究竟含有多少层要根据需求来说，到现在为止无法确定含有多少层的神经网络是最好的。而深度学习的代表，卷积神经网络就是在之前神经网络中新放入特征学习的需求，这种是仿照生命体大脑在信息信号分析处理中的分级而来的。

## 2.2 强化学习算法分析研究

### 2.2.1 强化学习原理

#### 2.2.1.1 强化学习与监督学习和非监督学习

监督学习的概念表示标明部分数据为正确，另外的部分是错误的，随后令程序判别之前的数据对或者是错。所以有监督学习，必须是有标签的。

无监督学习按照字面意思理解与监督学习相反，是不对数据进行标明对或者是错误的，而是令机器自主的来判别某些数据在一起比较像同一类的从而归划到一类中去。

强化学习或者叫做增强学习。在监督学习时机器的每个预测都有相对应的答案，或对或错，而强化学习却不会在预测之后收到明确的回复，收到的只有不知道是否有关系的反馈。这就要求机器持续的去学习试错，并在之后改变决策程序。

强化学习应用的是没有被标记的数据并且可以由一些方式得到你想要了解的正向的接近结果是否会到来，这就是奖惩函数。一般的“冷热游戏”比较好的诠释过这个想法。

可以说强化学习不依靠数据的标签来学习，它是依靠自己不断地积累反馈。

强化学习非常适应学习一些交互过程，如下围棋等，Alpha Go的成功强化学习起就到了至关重要的作用[24]。

#### 2.2.1.2 强化学习原理分析

强化学习是一个非常必要的学习工具，在通常情况下，作用在智能体上以实现自适应控制，如机器人控制与系统优化等等。一般来说，人们不知道智能体是怎样实现决策的，所以不会有用于监督学习的必要样例。但如果能够设计智能体最终目的以及不同结果的成本与奖励，那样智能体就可以通过强化学习自主的寻找最优化的策略。如果能提供强化学习的最终目的，智能体就能够自主的学会如何去最大限度地调整自己的行为策略以达到最终目的。

强化学习可以说是机器学习的重要组成，它由多个学科领域交叉而成，强化学习的本质是实现自主的进行决策[25]，而且能够做出连续决策过程。强化学习是智能体智能系统（智能体）利用和外在环境的相互交互，然后智能体利用试错机制重复不断的来学习新知识，从而改变自己的方案策略来配合适应多种多样的环境或者说是完成某项任务使命。

如果能对智能体位于环境中的所有情形进行合理的评价[26]，设计一个适合的回报函数，那只要能找到一条总的回报值最大的策略，这就是智能体的最优决策方案。

为了能够更加深入的熟悉强化学习，接下来要分析马尔科夫决策过程（Markov Decision Processes，MDP），一般来说强化学习的相关问题可以转换为有限状态下的马尔科夫决策过程来最终完成求解。

### 2.2.2 马尔科夫决策过程

MDP是一种在环境中建模决策的数学方法。一个环境用一套规则来定义一个真实的或虚拟的世界。本文专注于虚拟环境，特别是具有相应游戏机制限制的游戏。

MDP的核心问题是为决策者（在此称为代理人）找到最优的政策功能。

马尔可夫决策过程是一个离散时间状态下的随机过程，假设你为智能体，处于一种环境中。而周围的环境置于一些状态下的时候。这时候智能体能够从环境中选择一种动作来完成行动。这些行为有时可以得到某些奖励（例如得分增加）。

而这些行动通过改变环境同时生成了新的状态，紧接着智能体能够完成下一行动，等等。有关如何选择这些操作的规则称为策略。环境大多数情况下不会是设定好了的，说明接下来的状态很大概率会有一些随机性。

完整的马尔科夫决策过程通常会由五元组组成，即，每个元素的意义这样描述：

·S是状态States的集合，如在无人机控制问题中，S可用来描述无人机的位置、方向等状态；

·A是动作Actions的集合，如用来描述无人机所有可行的运行方向；

·是转移概率矩阵，表示在状态s的情况下，如果进行动作a，转移到下一状态的概率；

·γ是一个大于等于0，小于1的值，称为折扣因子Discount Factor；

·R是一个由状态和动作到实数的映射，称为奖励方程Reward Function,也能叫回报函数。

状态和动作集合以及从一个状态到另一个状态转换的规则构成了马尔可夫决策过程[27]。而过程的一个片段（episode）形成状态，动作和奖励的有限序列为：

这里表示状态，是动作，是执行动作后的奖励。片段以最终状态结束（例如，“游戏结束”画面）。马尔可夫决策过程依赖马尔可夫假设，下一个状态的概率仅可以由当前状态和动作确定，而不是由先前的状态或动作来确定。

如果实现长时间的优越表现，就不可以单纯思考当前的回报奖励，之后的长期状态的回报奖励也要充分的考虑到。比较容易算出只一次的马尔科夫决策在一个片段中总的回报：

(2-8)

那么从时间t开始的总体的未来奖励可以表示为：

(2-9)

因为所有的环境都不是可以安排好了的，因此不能够确定当下次运用一样的动作的时候是否能够反馈到相同的回报奖励，所以当面对将来越多的时候分歧也就会越来越多的产生[28]。所以，会给未来的奖励打个折扣：

(2-10)

公式2-10中提到的γ为介于0与1之间的一个数，称之为折扣因子，距离现在时间点越近的奖励会优先考虑而相反的距离现在越远的考虑得越少。从公式看出在时间t时刻对未来回报折扣能够使用t+1时的回报奖励来表达：

(2-11)

当设置折扣因子时γ=0的时候，策略将会比较短视，就只可能依靠即时的回报奖励。

当现在与将来的回报都比较重要的话，就会将回报因子调整成γ=0.9。当然如果的环境是确定的，那么可以设置γ=1。

智能体想要的优秀的策略是可以将选择的行动最大化（折扣后）未来的回报奖励。

运用强化学习的目标永远都是使得在整个马尔科夫决策过程中全局回报函数的均值最大，观测到伴随着次数的增加，慢慢接近于0[29]。

策略π是把状态映射到动作的函数，即，所以当位于状态π的时候，选择的动作是,就能够定义政策π的价值函数是：

(2-12)

公式2-12中的函数可以表达出在最初状态是的状态时，运用政策π能够得到的回报函数的期望，把上式化简后就成为了贝尔曼等式（Bellman Equation）：

(2-13)

上式中的中右侧第一项R(s)可以表达当下立即回报（Immediate Reward），代表的是当下选择动作a能够马上就得到的回报奖励，而第二项则是求和项，要表达的是在做出这样的选择时，未来的回报奖励（Future Reward）。

贝尔曼方程处理马尔科夫决策过程问题时得心应手，尤其是在完成有限状态（|S|<∞）马尔科夫决策的时候，能够把每个价值函数写成贝尔曼等式的方式。这会把每个状态的函数转化为线性方程，非常有利于问题的顺利完成。

定义最优价格函数Optimal Value Function为：

(2-14)

贝尔曼等式也有最优函数的形式：

(2-15)

同样可以定义最优政策函数为：

(2-16)

由此可以得出结论：

(2-17)

根据最优策略执行得到的值函数要优于比其他采用任何策略得到的值函数。

### 2.2.3 值迭代与策略迭代

之前的小节知道了贝尔曼方程与马尔科夫决策过程问题的优化目标,这一节分析两种求解有限状态马尔科夫决策过程参数模型的算法，状态集S和动作集A满足|S|<∞,|A|<∞。

算法2-1是对值迭代法的描述：

|  |
| --- |
| 算法2-1：值迭代法 |
| 初始化状态s的值函数V(s)为0  重复过程：{  对每一个状态s,更新值函数  }  直到值收敛 |

由上面所描述的算法可知要对第二步进行更新会有两个办法，第一是同步方法，就是说同时把全部状态都更新它们的价值函数；而第二种方法则成为异步方法，就是说渐渐的遍历状态中已有的状态，同时一步一步的将每个状态完成更新，状态完成更新的前后顺序会对当时的结果发生影响，而在最后结果收敛的时候，最后结果必然是相同的。

算法2-2是对策略迭代算法的描述：

|  |
| --- |
| 算法 2-2：策略迭代法 |
| 初始化策略=random action  重复过程：{  令  对于每一个状态s，  }  直到值收敛 |

由上面提到的算法能够看出，内循环始终运算当前政策π下的值函数，而且应用运算过后的值函数来重新更新政策π，像这样的称之为值函数V的贪心算法。

要知道优化内循环的步骤a时要对状态集s中的全部状态进行求解过程，所以需要运算|s|次。

所以当面对拥有比较少的状态的马尔科夫决策过程的时候，政策迭代速率会比值迭代的速率高出不少，而状态集合变成了上万或者更多的时候，政策迭代速率会变得比较慢，而这个时候会采用值迭代方式。这也是论文中采用的迭代形式。

完成有限次的迭代后V一定会收敛于,则会收敛于.

值迭代与策略迭代适用于不同场景。事件规模比较小的时候适宜运用策略迭代实现快速收敛；而当问题较为复杂时，运用值迭代会加快运算速度提升效率，有利于大范围的方程组的求解。

## 2.3 本章小结

本章开始分析了深度强化学习与之前已有的机器学习方法的联系与区别，然后交代了深度强化学习的基本思想与训练过程。更深层次的网络模型通过适合的训练导致深度神经网络比简单的传统的神经网络有更好的表现。

接下来分析了深度学习是一种包含多个隐层结构的神经网络，所以从单个人工神经元和激励函数到神经网络结构做了分析说明。

随后介绍了强化学习算法在智能体决策中应用的现状以及强化学习与监督学习和非监督学习之间的关系。然后来详细分析了强化学习原理还有马尔科夫决策过程，最后分析了值迭代与策略迭代的不同算法，为后续强化学习与深度学习的结合打好基础。

# 第3章 智能体决策算法

在本章文中，将研究如何更好地教智能体作出决策并取得最好的成绩。强化学习应用到智能体决策领域已经是很普遍的了，本章将分析对比不同算法，即传统强化学习算法，深度Q学习随机经验重放算法还有深度Q学习优先经验重放算发应用于智能体决策中的情况。

## 3.1 智能体决策问题分析介绍

### 3.1.1 智能体决策问题发展现状

用于决策的强化学习已经是一个比较完备成熟的研究，但是只依靠强化学习的应用在现今看来已不足以满足需要。这篇论文结合深度学习与强化学习并运用到智能体领域中来，教会智能体做决策。

深度强化学习算法是将深度学习与强化学习的特点结合起来形成了新的高效人工智能学习算法[30]。它可以运用到诸如机器人行为控制，汽车自动驾驶，图形游戏控制等诸多领域研究之中。本文主要研究将深度强化学习算法运用于操控智能体使得智能体能够在任务中取得更好的成绩。

可以先想想一下人类是如何完成任务的。首先，人类眼睛与大脑发现环境的变化并开始对收到的图像信号进行理解并处理。然后，人类的大脑将环境信息转化为语义信息，包括游戏中物体类型，具体位置等。然后人类玩家根据以往的经验选择一个操作。最后，环境收到人类的反应发生变化到下一个状态，人类得到一定的回报和反馈，比如到达目标，或是任务失败。循环往复直到结束游戏。

和人类类似，使机器智能体（Agent）完成任务也要经历相似的过程。人类可以通过反复完成任务积累经验并在以后将任务完成得更好，获得更高的成绩。也希望智能体可以和人类一样。虽然人脑在对环境处理的优秀能力无与伦比，但是通过将深度强化学习算法运用到智能体对自身的操控中来，也可以得到和人类玩家一样的能力甚至超过人类。将深度学习最得意的特征提取性能与强化学习的对周围环境的学习能力相结合并更加高效运用到任务中是本文的研究重点。

### 3.1.2 强化学习应用于智能体决策

#### 3.1.2.1 应用强化学习的决策方法发展

运用强化学习进行智能体决策是一个非常成熟的研究领域，常用于衡量算法在环境中的表现如何。本节介绍强化学习中的一些最重要的成就。

TD-Gammon是一种能够在棋盘游戏西洋双陆棋中达到专家水平的算法。该算法由Gerald Tesauro于1992年在IBM的Thomas J Watson研究中心开发。TD-Gammon由三层神经网络组成，并使用称为TD-Lambda的RL技术进行训练。TD-Lambda是Richard S.Sutton发明的一种时间差异学习算法。玩家可以执行的所有可能动作并估计该特定动作的奖励。然后选择产生最高奖励的动作。TD-Gammon是利用自游戏方法来改善人工神经网络参数的第一种算法之一。

在2015年末，Alpha GO成为第一个赢得人类专业围棋玩家的算法。Alpha GO是一个RL框架，它使用蒙特卡洛树搜索与两个深度神经网络[31]进行价值和策略评估。价值是指假设智能体的一个预期未来回报。策略网络试图了解哪个动作最好。

Alpha GO的最早版本使用了来自人类专业人士所玩游戏的训练数据。在最近的版本中，Alpha GO Zero只使用游戏来训练人工智能。在最近的更新中，Alpha GO被推广到Chess和Shogi（日本国际象棋）中，只使用24小时就达到超人类游戏水平。

DOTA2是玩家控制英雄比赛的策略类玩家游戏。游戏目标是击败敌方英雄并摧毁他们的基地。2017年8月，Open AI发明了一种基于RL的AI，在一场比赛中击败职业玩家。训练仅使用自训练完成，算法学习如何利用游戏机制表现良好。

Deep Stack是一种算法，可以在Texas Hold扑克中进行专家级别的比赛。该算法结合神经网络使用树搜索在游戏中执行合理的动作。Deep Stack是一种通用算法，旨在解决信息不完善的问题。

人工智能方面还有其他一些重要的成就，而且强化学习算法会越来越多的运用到游戏操作控制方面。

#### 3.1.2.2 强化学习 Q-Learning 算法

Q-Learning是强化学习的一种无模型算法。这意味着马尔科夫决策过程在整个学习过程中保持隐藏状态。目标是通过估计动作价值函数来学习最优策略，从而在状态s在环境中执行动作a时产生最大期望回报。

在Q学习中可以定义一个函数代表当状态s选择a行动后未来（折扣）奖励的最大值，并从该时间点继续优化。

(3-1)

介绍Q(s,a)的办法可以为位于状态s选择动作a时可以当游戏完结时得到最棒的奖赏分数，这被叫做Q函数，这是为了展示它能够表达已有状态下一个动作行为的质量。

当在一个状态下的时候接下来要确定是向左走还是向右走，这个时候就要像选择可以在游戏最终结果得到最高奖励的一步。一旦拥有Q函数，就变得很容易决则，利用Q值最高的动作来做就好了。即遵循公式 3-2：

(3-2)

字符π表示的是一种策略，表示的是在不同状态下的时候选择动作的规则。

然后，获得这个Q函数，只需要关注一个转换<s,a,r,s’>。这就像之前提及带有折扣的未来奖励，可以把现在状态的Q值描述为包含下个状态的Q值：

(3-3)

这被称为贝尔曼方程。当认真思考时，就知道这是比较合乎逻辑的，当下的状态和行动最大的未来奖励就是取决于目前的直接奖励再加下个状态的最大未来奖励[32]。

Q-Learning的表达的主旨思想是可以运用贝尔曼方程迭代来慢慢的逼近Q函数。在初级状态下，Q函数可以简化实现为一个表，其中的状态是行接着执行的动作是列。

Q学习算法是：

|  |
| --- |
| 算法 3-1：Q 学习算法 |
| 初始化策略Q值  在状态，采用动作（贪心算法），得到  根据)计算更改Q值  达到目标状态，终止一次迭代，在执行1到 4，至学习结束。 |

上面表中的α是代表的是学习率，它的作用是表示之前的Q值与新提出的Q值的差异大小并决定有多少被考虑进去。尤其要注意的是当α=1时，将两个取消，所能获取的更新方法与贝尔曼方程一样。么用来更新Q(s,a)的最大Q(s’,a’)只是一个近似值，位于学习早期阶段有时会是是完全不对的，然而，随着算法的一次次迭代更新，近似值就会变得越来越精准，而且如果已经展现出当前值得到充分的更新，那么Q函数就会变得收敛同时得到真实的Q值。

Q-Learning的更新规则基于值迭代：

(3-4)

公式显示了传播环境中每个离散时间步长的估计Q值的迭代过程。α是算法的学习率，奖励函数，并且通常在-1<x <1之间以增加学习稳定性。γ是折扣因子，折扣未来状态的重要性。

## 3.2 用于智能体决策的深度Q学习与经验重放算法

### 3.2.1 深度Q学习结合随机经验重放算法

#### 3.2.1.1 深度Q学习结合随机经验重放算法分析

经验重放，又叫做经验回放，可以利用Q学习算法来评估每一个状态的未来奖励与利用卷积神经网络逐渐逼近Q函数。在实际操作中必须利用很多大量技巧才可以令其收敛。同时利用较长处理时间用来处理神经网络，在单个GPU上需要花费一周的时间。

最重要的技巧是经验回放。在决策过程中，所有经验<s,a,r,s’>都被存储在回放存储器中。学习如何直接从视觉和言语等高维度感官输入控制智能体是强化学习（RL）的长期挑战之一。在这些领域进行操作的大多数成功的RL应用程序都依赖于手工制作的功能以及线性值功能或策略表示。显然，这种系统的性能主要是依赖特征表示的质量。

最近在深度学习方面取得的进展使得从原始感官数据中提取高级特征成为

可能，这导致了计算机视觉和语音识别的突破。这些方法利用一系列神经网络架构，包括卷积网络，多层感知器，受限玻尔兹曼机器和递归神经网络，并且已经利用了监督学习和无监督学习。似乎很自然地要问，类似的技术是否也能对强化学习感官数据有益。

然而，强化学习与深度学习相结合时提出了几个挑战 [33]。首先，迄今为止最成功的深度学习应用程序需要大量手写标记的训练数据。另一方面，RL算法必须能够从标量奖励信号中学习，这种信号经常是稀疏，噪声和延迟的。事件与事件之间的延迟可能是数千步长，与输入与目标之间直接相关的监督学习相比，似乎尤其令人望而生畏。另一位学者认为，几乎所有的学习算法都是独立的，而不是强化学习者通常会遇到高度相关的状态序列。此外，在RL中，数据分布随着算法学习新行为而改变，这对假定固定的基础分布的深度学习方法可能是有问题的。

本节分析了神经网络能够在复杂的RL环境中从原始数据中恢复成功的控制策略。网络训练有Q学习算法的变体，随机梯度下降更新权值。为了减轻相关数据和非平稳分布的问题，使用了一种经验重播机制，它随机抽样以前的转换，从而平滑了许多过去的行为的训练分布。

将方法应用到决策中。实验结果将在第五章详细说明。网络只是从环境输入，奖励和终端信号，以及一系列可能的行为中学到东西，就像人类玩家一样。此外，网络架构与用在培训的全部参数在整个过程中都会维持不变。

也许最有名的强化学习的成功故事是TD-Gammon，一种通过重新强化学习和自训练，并达到了超人类游戏水平的双陆棋游戏程序。TD-gammon运用与Q-learning相似的无模型强化学习算法，并利用具有一个隐含层的多层感知器近似值函数[34]。

但是，尽早尝试使用TD-Gammon，包括应用棋类游戏，Go和棋子都不太成功。这导致了一个普遍的观点，即TD-Gammon方法是一种仅在双陆棋中起作用的特殊情况，这可能是因为骰子卷中的随机性有助于探索状态空间并且使得值函数特别平滑。此外，下降趋势与非线性函数逼近器或实际上与关闭策略学习结合使用无模型重新强化学习算法可能会导致Q网络发散。随后，强化学习的大部分工作集中在具有更好收敛保证的线性函数逼近器。

最近，人们对将深度学习与强化学习相结合的兴趣有所恢复。深层神经网络用于密切环境ε;有限玻尔兹曼机已被用于估计值函数或策略函数。此外，与Q学习有关的分歧问题已通过梯度时间差分方法得到了部分解决。这些方法在用非线性函数逼近器评估固定策略时被证明是收敛的。或者当使用Q-learning的受限变体学习具有线性函数近似的控制策略时。但是，这些方法还没有扩展到非线性控制。

也许与自己的方法最相似的先前工作是神经网络适配Q学习（NFQ）。NFQ使用RPROP算法更新Q网络的参数。但是，它利用的是批量更新，在每次迭代时的计算成本和数据集的大小成比例，当要考虑随机梯度更新时，每次迭代具有较低的恒定成本，并且可以扩展到大型数据集。NFQ也已经成功应用于简单的真实世界的控制任务，使用纯粹的视觉输入，首先使用深度自动编码器来学习任务的低维表示，然后将NFQ应用于这种表示。相比之下，的方法直接从游戏图像输入端到端应用强化学习；因此它可能会学习与歧视行为值直接相关的特征。

Q-learning之前也与经验重放和简单的神经网络相结合，但是再次从低维度状态开始，而不是从原始视觉输入开始。

#### 3.2.1.2 深度Q学习结合随机经验重放算法实现

最近在计算机视觉和语音识别方面的突破依赖于有效地训练深度神经网络和大量训练集。几乎所有的成功实例都是从原始输入中直接获取，使用基于随机梯度下降的轻量级更新。通过向深层神经网络提供足够的数据，通常可以学习比手工特征更好的表示。这些成功激发了强化学习的方法。的目标是将强化学习算法连接到直接在RGB图像上操作的深层神经网络，并通过使用随机梯度更新有效地处理训练数据。

该体系结构更新了网络的参数，该网络直接根据算法与环境的交互作用从经验上的在线样本估计值函数的参数。由于这种方法能够超越20年前最好的人类步步高玩家，很自然地想知道，二十年的硬件的不断发展与现代深度神经网络架构和可扩展强化学习算法是否会产生重大进展。

与TD-Gammon和类似的在线方法相比，利用了一种称为经验回放的技术，将代理人的经验存储在数据中的每个时间步，设置,将许多轮次汇集到一个重放记忆中。在内部循环算法中，应用Q学习更新或最小化更新，从存储样本池中随机抽取经验样本e〜D。在执行重放经验之后，代理根据贪婪策略选择并执行操作。由于使用任意长距离输入音频网络的历史很困难，Q函数改为修正由函数φ产生的历史的固定长度表示。完整的算法，可以称之为深度Q学习，在算法3-2中给出。

|  |
| --- |
| 算法3-2：深度Q学习结合随机经验重放算法 |
| 初始化重放记忆D到内存N中  使用随机权重初始化动作价值函数Q  for episode=1 to M do  初始化序列并且预处理此序列  for t=1,…,T do  根据概率选择一个随机动作  其他情况下令  在模拟器中执行动作并观察即时回报与图像  令并且预处理  保存经验单元组到历史经验队列D中  从D中随机采样mini-batch大小的样本  当达到最终的时，令否则令  在执行梯度下降操作  end for  end for |

这种方法比标准的在线Q学习有几个优点。首先，每一步的经验都可能用于许多权重的更新，这可以提高数据效率。

其次，由于样本之间的强相关性，直接从连续样本中学习是不够的；对样本进行随机化打破了这些相关性，因此降低了更新的方差。第三，当在线学习时，当前参数决定了参数的下一个数据采样。比如当最大化动作为左移的时候，那么训练样本就会由来自左侧的样本支配；而如果最大化动作随后变化到右侧，那么训练分布也会切换。很容易看出不希望的反馈环路可能会出现并且参数可能会陷入局部最小值，甚至发生灾难性的分歧。通过使用经验回放，行为分布在其许多以前的状态中被平均，平滑学习并避免参数中的振荡或分歧。请注意，在通过体验重播进行学习时，需要学习关闭策略（因为当前参数不同于用于生成样本的那些参数），这就激发了Q学习的选择。

实践中，在执行更新时，首先计算存储内存中最后N个经验元素，并从D中随机抽取样本。这种方法在某些方面受到限制，因为内存缓冲区不会区分重要的转换，并且由于有限的内存大小N而始终会覆盖最近的转换。同样，均匀采样对于重放内存中的所有转换同等重要。更复杂的抽样策略可能强调可以从中学到最多的过渡，类似于优先扫描。

### 3.2.2 深度Q学习结合传统优先经验重放算法

#### 3.2.2.1 深度Q学习结合传统优先经验重放算法分析

之前分析的经验重放算法的效果比较好，但目前的做法还是对过往经验均匀采样的[35]。如果能根据经验的重要程度进行有侧重的重放效果可以更好。

在线强化学习（RL）智能体[36]在观察一系列经验的同时逐步更新其参数（策略，价值函数或模型）的参数。以最简单的形式，他们在一次更新后立即丢弃传入的数据。与此相关的两个问题，第一是强相关的更新打破许多流行的基于随机梯度的算法的独立同分布假设，第二是快速遗忘可能稍后有用的罕见经验。

经验重放解决了这两个问题：通过将经验存储在重放记忆中，通过混合更新和近期更新的体验，打破时间相关性成为可能，而且罕见的体验将不仅仅用于一个更新。这在Deep Q-Network（DQN）算法中得到了证实，该算法通过经验重放来稳定训练以神经网络为代表的神经网络功能。具体而言，DQN使用了一个大的滑动窗口重放记忆，随机统一对其进行采样，并平均重新访问每个过渡点8次。一般来说，经验重放可以减少学习所需的经验，并用更多的计算和更多的内存来代替它，这通常比RL代理与环境的交互更便宜。

在本小节中，研究如何确定重播哪些转换的优先次序可以使体验重复性的有效性和有效性以及完全重播的效果。RL智能体可以从某些转换中比其他人更有效地学习的关键字。转换可能或多或少是令人惊讶的，多余的或与任务相关的。有些转换可能不会立即用于转换，但是可能会导致其功能增加。经验重放可以按照他们所经历的确切顺序，从处理转换中解放在线学习智能体。优先重放进一步解放了智能体以与他们经历的频率相同的频率考虑转换。

特别是，通常以重要的时序差分（Temporal Difference,TD）误差为衡量标准，重新审视学习进展。这种优先级可能会导致多样性的丧失，通过随机优先级来缓解这种多样性，并引入偏置，用重要性抽样来纠正。

许多神经科学研究已经确定了啮齿类动物海马体验重现的证据，表明先前体验的序列可以在清醒休息或睡眠期间重播。与奖励相关的序列似乎更频繁地重播。具有高度TD误差的经历似乎也更频繁地重播。

众所周知，通过按照适当的顺序对更新进行优先级排序，可以使值迭代等规划算法更加高效。优先广泛的选择下一次更新哪个状态，如果更新已执行，则根据值的变化优先化。TD误差提供了衡量这些优先级的一种方法。的方法使用了类似的优先方法，但对于无模型RL而不是基于模型的规划。此外，使用从样本中学习函数逼近器时更加鲁棒的随机优先级。TD误差也被用作确定资源集中位置的优先级机制，例如，在选择何处探索或选择哪些特征时等。

在监督学习中，当类别识别已知时，有许多技术可以处理不平衡数据集，包括重采样，欠采样和过采样技术，可能与集合方法相结合。最近的一篇论文引入了一种形式化的样式，用于解释有经验重放算法的深度强化学习；该方法将经验分为两个桶，一个用于正面，一个用于负面奖励，然后从每个桶中挑选一个固定分数来重放。这只适用于具有“正面/负面”经验的自然概念的领域。

使用重播内存可以在两个层面上进行设计选择：完全的体验，以及重播的体验（以及如何实现）。本节只涉及后者，假设其内容不在的控制范围内，最有效地使用重放记忆进行学习。

#### 3.2.2.2 深度 Q 学习结合传统优先经验重放算法实现

优先经验重放的核心组成部分是衡量每个状态转移序列（transition）重要性的标准。一个理想化的标准是RL代理可以从当前状态的转变（预期的学习进度）中学到的数量。虽然这个衡量标准不是直接可以获得的，但合理的替代指标是过渡期TD误差δ的大小，它表明过渡的“令人惊讶”或意料之外的情况：具体而言，该值与其下一步自举估计有多远。这特别适用于增量式在线RL算法，如SARSA或Q-learning，它们已经计算出TD误差并按比例δ更新参数。TD误差可能会影响到某些情况。

贪婪的TD误差优先级有几个问题。首先，为了避免在整个重播内存中进行昂贵的扫描，TD误差仅针对重放的转换进行更新。其中一个结果是，首次访问时具有较低TD误差的转换可能无法长时间重放（这意味着永远不会使用滑动窗口重播存储器）。此外，它对噪声尖峰敏感，这可以通过引导来增强，其中近似误差作为另一个噪声源出现。最后，贪婪的优先级重点集中在小部分经验上：错误收缩缓慢，特别是在使用函数逼近时，意味着最初的高错误转换频繁重放。这种多样性的缺乏使系统容易受到冲击。

近似误差表现为噪音的另一个来源。最后，贪婪的优先级重点集中在小部分经验上：错误收缩缓慢，特别是在使用函数逼近[37]时，意味着最初的高错误转换频繁重播。这种多样性的缺乏使系统容易受到冲击。

为了克服这些问题，引入了随机抽样方法，该方法介于在纯贪婪优先和均匀随机抽样之间。确保在过渡的优先级中被采样的概率是单调的，同时保证即使对于最低优先级的过渡也是非零概率[38]。

状态转移序列为i的优先级与TD-error的差值有关，每个优先级出现的概率是：

(3-5)

其中pi>0是转换i的优先级。指数α确定使用多少优先级，α=0对应于统一的情况。

考虑的第一个变体是直接比例优先化，其中,其中是一个小的正常数，可以防止一旦错误为零时不会重新访问转换的边缘情况。第二个变体是间接的，基于等级的优先级，其中,其中rank(i)是当重播内存按照|排序时的过渡i的等级。在这种情况下，P变成指数为α的幂律分布。这两种分布在都是单调的，但后者可能更稳健，因为它对异常值不敏感。

为了从公式3-5中有效地采样，复杂度不能取决于N。对于基于秩的变体，可以用具有k个相等概率的分段线性函数来近似累积密度函数。段边界可以预先计算（它们只在N或α改变时才改变）。在运行时，对一个段进行采样，然后在初始化时对其进行均匀采样。这个工作特别适用于以minibatch为基础的学习算法：选择k作为最小批量的大小，并从每个片段中精确地采样一个转换，这是一种分层采样形式，具有平衡小批量的附加优势。

根据这个概率值在经验重放池中来进行transition优先级的选定，从而影响最后游戏的结果。

但是在优先经验重放算法中对优先级出现的概率的设定没有经过数学推导与对比实验就拿来直接用了，导致可能会出现更好的实验结果而没有被发现并被运用到训练游戏当中去。

## 3.3 本章小结

本章首先开始介绍了智能体决策发展现状与未来方向，然后说明了强化学习运用到智能体决策的相关情况并分析了强化学习Q学习算法。接下来分析验证用于智能体决策的深度Q学习与随机经验重放算法还有深度Q学习与优先经验重放算法。

# 第四章 改进深度Q学习优先经验重放算法

随着最近令人兴奋的深度学习成果，受益于大数据，强大的计算和新的算法技术，目睹了强化学习的复兴，特别是强化学习和深度神经网络的组合，即深度强化学习（Deep Reinforcement Learning）。基于深度强化学习的深度Q学习优先经验重放算法在智能体决策中的运用逐步增多，本章将专注于改进深度Q学习优先经验重放算法在处理游戏图像中的运用。

## 4.1 深度强化学习算法分析

### 4.1.1 深度强化学习背景分析

强化学习是一个近年来蓬勃发展的研究领域，并且在计算机游戏中以人工智能为基础的对手显示了巨大的潜力[39]。这一成功主要归功于卷积神经网络的广泛功能，使算法能够从嘈杂的环境中提取有用的信息。

强化学习通常是关于顺序决策制定，解决科学，工程和艺术领域广泛领域的问题。

一直在见证突破，如深度Q网络，Alpha Go和可微分神经计算机;以及新颖的体系结构和应用，如异步方法，对决网络架构，价值迭代网络，机器翻译的双重学习，语音对话系统，信息提取，指导性政策搜索，生成对抗模仿学习，无监督强化和辅助学习以及神经构架设计等。

学习如何直接从视觉和言语等高维度感官输入控制智能体是强化学习的长期挑战之一。在这些领域进行操作的大多数成功的RL应用程序都依赖于手工制作的功能以及线性值功能或策略表示。显而易见的是这种系统的性能是十分依赖于特征表示质量的[40]。

最近在深度学习方面取得的发现可以从原始感官数据中提取到高级特征不再是难事，这导致了计算机视觉和语音识别等领域的突破。这些方法利用一系列神经网络架构，包括卷积网络，多层感知器，受限玻尔兹曼机器和递归神经网络，并且已经利用了监督学习和无监督学习。似乎很自然地要问，类似的技术是否也能对RL感官数据有益。

然而，强化学习从深度学习[41]的视角看问题时会有几个挑战。首先，迄今为止最成功的深度学习应用程序需要大量手写标记的训练数据。另一方面，RL算法必须能够从标量奖励信号中学习，这种信号经常是稀疏，噪声和延迟的。事件与事件之间的延迟可能是数千步长，与输入与目标之间直接相关的监督学习相比，似乎尤其令人望而生畏。另外的一些学者会认为，几乎所有的学习算法都是独立的，而不是强化学习者通常会遇到高度相关的状态序列。另外在强化学习中，数据分布会随着算法学习新行为而改变，这对假定固定的基础分布的深度学习方法能会产生一些问题[42]。

本文将会论证卷积神经网络能够在较为复杂的 RL 环境中将有效信息从原始游戏图像数据中恢复成功的控制策略。网络训练有Q学习算法的变体，随机梯度下降更新权值。为了减轻相关数据和非平稳分布的问题，使用一种经验重播机制，它随机抽样以前的转换，从而平滑许多过去行为的训练分布。的目的是创建一个单一的网络，它可以成功地阻止游戏的可能性。网络没有提供游戏特定的信息或手工设计的视觉特征，也没有提供给仿真器的内部状态；它只是从视频输入，奖励和终端信号，以及一系列可能的行为中学到东西就像人类玩家一样。

### 4.1.2 深度强化学习中的梯度下降算法

人们已经研究了许多梯度下降的变化，有一个主要的缺点：它有必要计算二阶偏导数，这可能是非常费力的。为了看看它为什么费力，假设想要计算所有的第二个偏导数。如果有一百万个这样的变量，那么就需要计算一个像万亿分之一，即一百万平方秒的偏导数那样的计算成本会很高。如果说，有避免这种问题的技巧，寻找梯度下降的替代方案是一个活跃的调查领域。很好的使用梯度下降的变化作为在神经网络中运用的主要方法。

如何应用梯度下降在神经网络中学习这个想法是使用梯度下降来找到最小化费用的权重w和偏差b。如公式4-1所示为成本函数：

(4-1)

表示网络中所有权重的集合，b是所有的偏差都是训练输入的总数，n是网络输入时的输出向量，a总和是所有训练输入的总和。目标是使用代价函数的大小来衡量深度学习网络中权重与偏差的调整到合适状态的好坏。

要查看它是如何工作的，让重新设置梯度下降更新规则，并用权重和偏差替换变量。换句话说，的位置现在具有分量，并且梯度向量具有相应的分量和。在组件方面，已经提出了梯度下降更新规则：

(4-2)

(4-3)

通过重复应用此更新规则找到最低的成本函数。换句话说，这是一个可用于在神经网络中学习的规则[43]。不幸的是，当训练输入的数量非常大时，这可能需要很长时间，并且学习缓慢发生。

一个叫随机梯度下降(Stochastic Gradient Descent, SGD)的想法可以用来加速学习。这个想法是通过计算随机选择的训练输入的小样本来估计梯度。通过对这个小样本进行平均，可以快速得到对真实梯度的良好估计，并且这有助于加速梯度下降并进而学习。

为了使这些想法更加精确，随机梯度下降通过随机挑选少量随机选择的训练输入来工作。 将这些随机训练输入标为标签，并将它们称为mini-batch。认为样本尺寸m足够大，认为平均值将大致等于平均值，也就是说：

(4-4)

为了将其与神经网络中的学习联系起来，用和表示的神经网络中的权重和偏差。然后随机梯度下降法是通过挑选一个随机选择的最小批次的训练输入，并用这些输入进行训练。即：

(4-5)

(4-3)

总的结果是在当前小批量中的所有训练样本上。然后挑选另一个随机选择的小批量培训。依此类推，直到用完了训练输入，就可以说完成了一次训练。那时开始了一个新的训练循环。

## 4.2 改进深度Q学习优先经验重放算法分析研究

### 4.2.1 深度 Q 学习

深度Q学习（Deep Q-learning，DQN）是由深度学习和强化学习Q学习算法结合产生的新的算法。Mnih教授等把卷积神经网络与强化学习的Q学习相融合[44]，提出了深度Q网络模型。该模型可以用在处理基于视觉感知的控制任务等诸多方面，是深度强化学习学习领域的开创性工作。

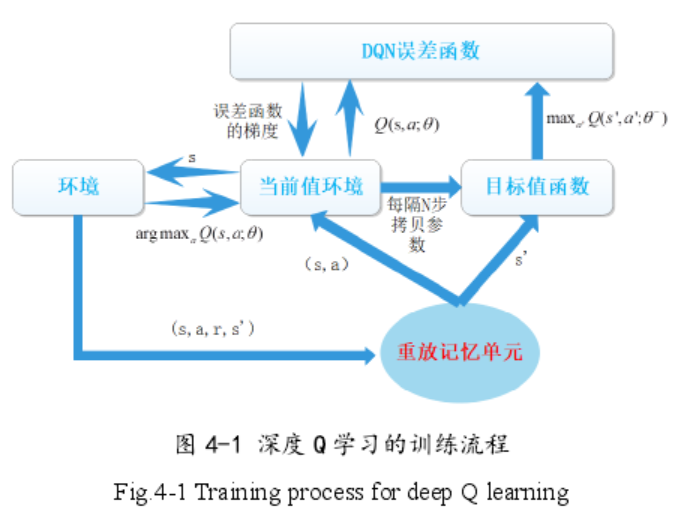
在最基本的层面上，Q-Learning利用一个表格来存储队列。相反，可以使用非线性函数近似来近似。这就是所谓的深度Q学习。描述近似的可调参数（权重）。人工神经网络（Artificial Neural Network，ANN）用作从Q表中检索值的近似方法，但是以稳定性为代价。使用ANN很像在JPEG图像中找到的压缩。压缩是有损的，并且在压缩时信息丢失。这使得DQN不稳定，因为值可能在训练中被错误地编码。除了值迭代之外，还必须为更新参数的反向传播过程定义代价函数。代价函数另外一种表示形式为：

(4-7)

公式4-7说明了提出的代价函数，也可以叫做损失函数。它使用贝尔曼方程来计算梯度下降的损失。为了提高训练稳定性，使用经验重放。这是一个内存模块，用于存储已经探测到的状态空间部分的内存。经验通常随机选择，然后作为训练数据重放到深度神经网络。

首先是将深度神经网络用作Q值网络[45],用w表示，然后用Q值利用均方差定义目标函数，同时可以称之为损失函数，能够知道这里即是利用Q学习来更新的Q值作为目标值的[46]。现在有目标值和当前值后，他们之间的偏差可以运用均方差得到。接下来就能够直接算出和损失函数相关的梯度了，最后使用SGD实现优化目标。已经拥有之前计算得出的梯度值，就可以在深度神经网络中来计算了，因而能够利用SGD随机梯度下降来跟新参数并最终获取最优的 Q值。

图4-1重点表述了深度Q学习的相关训练流程。为了解决非线性网络在对值函数的表达时存在的不太稳定等问题，深度Q学习算法对之前的Q学习算法做了三点改进。



##### 图 4-1 深度Q学习的训练流程

第一点是深度Q学习在训练过程中要用到经验重放机制，在线处理得到的转移样本是。在当中的每个时间步 t 时刻，把智能体与环境相互融合所拿到的转移样本存储到回重放记忆单元中去。当训练时从重放记忆单元中随机的选择小批量转移样本，然后利用随机梯度下降（SGD）算法用来更新网络参数θ。在训练深度网络时，通常都是要保证训练样本相互之间都是相互独立的。

第二点是深度 Q学习除了应用深度卷积网络近似表达当前的值函数以外，还另外使用了另外的一个网络用来产生目标 Q 值。用当做当前值网络的输出，可以来估计当前状态动作队列的值函数[47]用来表示目标值网络的输出，通常用近似值函数的优化目标，就是目标Q值。现在值网络的参数θ想要做到实时更新的效果，每次在经过N轮的迭代以后，就会把当前值网络的参数传递给目标值网络。通过最小化当前Q值和目标Q值之间的均方误差来达到更新网络参数的目的。误差函数为：

(4-8)

对参数θ求偏导，得到以下梯度：

(4-9)

拥有目标值网络之后，相对时间段内目标Q值的数值不会有变动，这也在相当程度上降低现在的Q值与目标Q值之间相关性，提升算法的文稳定性。

第三点是深度Q学习[48]将奖励值和误差项缩小到有限的区域内，从而有效的确保了Q值和梯度值都能处在合理的范围内，提高了算法的稳定性。实验表明，深度Q学习在处理例如Atari游戏等类真实环境的比较复杂问题时，表现出与人类玩家不相上下的技术水平。在处理各种基于视觉感知的深度强化学习任务时，深度Q学习使用了同一套网络模型，参数设置与训练算法，这些都能够证明深度Q学习方法拥有很强的适应性和通用性。

接下来的章节会重点分析之前的深度Q学习与经验重放算法结合运用到智能体决策处理中的算法以及改进后的算法，通过对比后会发现本文所提出来的改进后的算法相较之前的算法在智能体决策处理方面有了很好的提升。

### 4.2.2 改进深度 Q 学习优先经验重放算法分析

深度强化学习算法（Deep reinforcement learning, DRL）是Deep Mind最近时间以来重点研究且会有美好前景的机器学习算法架构。两篇 Nature 上的奠基性论文（DQN和Alpha Go）使得 DRL这一深度学习（Deep learning, DL）和强化学习（Reinforcement learning, RL）的交叉方向变得炙手可热。之前有论文在从重放经验记忆（replay memory）抽样时选择了随机抽样，这样做可能会导致最后的结果不是最优的。

运用经验重放算法，可以把智能体所有的时间步骤内的经验全部存储起来，，将许多片段（episodes）存储进一个重放记忆中。而位于算法内部的循环中，采取Q-learning更新，mini-batch小批量更新等来采样经验，并从存储的样本中随机的提取。在完成经验重放后，智能体依照贪婪算法，选择并且执行一个动作[49]。

上述算法存储最近N个经验元组（experience tuples），当执行更新任务时，随机从经验重放数据集上均匀的采样。这种新的强化学习深度学习模型仅使用原始像素作为输入，使用在线Q学习的一个变体，它将随机小批量更新与经验重放记忆相结合，以便轻松培训RL的深层网络。这种方法在测试的游戏中获得了最先进的成绩，没有对架构或超参数进行调整。

一部分经验比其它经验要对参数的训练效果发挥出更大的作用。基于此，

Tom·Schaul和John·Quan等人在2016年提出使参数更新倾向使值估计变化更大的经验。如何确定一些经验比另一些经验对值估计的贡献更大，就需要有定量的测度。这个测度一般选用时间差错误法,一般来说当TD-error越大的时候，就是expected learning progress越高的经验数据，这样就能够让它们重放的次数更加的频繁。基于这种思想实现的greedy TD-error prioritization算法把经验数据和它的TD-error按序存放入重放经验记忆中，每次取最大TD-error的经验单元进行重放，与此同时参数更新的量也与之同比。除此之外新的经验单元会被设定成最大优先级以保证它至少能够被训练一次，从而最大限度的减少过拟合的出现。

但是之前的方法在定义每个状态转移的抽样概率P(i)的时候没有进行充分的数学论证与理论指导。基于以上考虑，本节提出了一种改进的优先经验重放技术来弥补这个问题。该方法通过大量科学数据计算对比得出最优的抽样概率P(i)。并采用Open AI Gym提供的小车爬山等 5 个游戏强化学习环境来验证该算法，通过改变优先级到概率的映射函数，得到更重视最大优先级的函数表达式。实验表明，该方法可以在保证游戏得分的情况下有效地降低学习玩游戏的时间，算法的性能得到明显提高，使游戏学习最优策略的速度显著提升。

### 4.2.3 改进深度 Q 学习优先经验重放算法实现

根据上述问题，针对优先级pi出现的概率，改变这个概率求解公式，可以多尝试几种，只要满足条件1：，条件2:。然后进行实验对比看应用哪种概率选择公式能使游戏学习的更快。

发现函数选择时函数的导数（或者是变化率）很重要。，，等不同的函数表达式会成数量级的影响最大优先级出现的概率的大小。通过大量实验可以得到的结论是变化率大的函数表达式会更重视最大优先级，使得最大优先级的概率大增。

Log 函数以2为底的表达式为：

(4-10)

时，，为优先级，差值越大，即各个优先级之间的差距越大，概率映射函数对最大优先级越重视。就越可以使得最大优先级的概率越大，使得游戏能够从以往的经验池中更多的学习到奖励最大的状态转移，使得学习玩游戏得到的奖励值最大，也就是在最少的时间内将游戏玩的结果每次都能快速的达到要求。

提出的改进的优先重放算法就是针对优先级到概率的映射函数，通过实验选定函数f(x)=ax，它的数学曲线图像的变化率随着变量 x 的增大而快速增大，可令a等于自然数e,同时通过大量实验对比得到最优的映射函数是：

(4-11)

其中，是转移i的优先级。指数α决定了使用多少优先级，当α等于0的时候是均匀的情况。

通过改变优先级到概率的映射函数，在α不变时，得到了更优的实验结果，对比原始优先经验重放，改进的优先经验重放找到正确的完成游戏或任务的方法的速度大幅提升。

本文的最新算法与深度Q学习结合随机抽样算法[50]相类似，利用了一种称为经验回放的技术，将代理人的经验存储在数据中的每个时间步，设置，将许多轮次汇集到一个重放记忆中。在内部循环算法中，应用Q学习更新或最小化更新，从存储样本池中随机抽取经验样本e〜D。不同的是在在执行重放经验时，将按照最新的优先级与相对应的概率关系进行策略选择并执行操作[51]。完整的算法，可以称之为新改进深度Q学习优先经验重放算法，在算法4-1中给出。

|  |
| --- |
| 算法4-1：改进深度Q学习优先经验重放算法 |
| 初始化重放记忆D到内存N中  使用随机权重初始化动作价值函数Q  for episode=1,M do  初始化序列并且预处理此序列  for t=1,…,T do  根据概率选择一个随机动作  其他情况下令  在模拟器中执行动作并观察及时回报与图像  令并且预处理  保存经验单元组到历史经验队列D中  从D中随机采样mini-batch大小的样本  如果 ,for j=1 to M do  抽取transition  计算重要抽样权重  计算TD-error )  更新transition优先级  累计权重变化)  当达到最终的时，令否则令  在执行梯度下降操作  end for  end for |

新改进深度Q学习优先经验重放算法与深度Q学习结合随机经验重放算法还有深度Q学习优先经验重放算法相比，通过改变优先级到概率的映射函数，在α不变时，得到了更优的实验结果，对比原始优先经验重放，改进的优先经验重放找到正确的完成游戏或任务的方法的速度大幅提升。实验结果与分析将在第五章详细介绍。

## 4.3 本章小结

本章开始阐述了深度强化学习背景并分析了深度强化学习中的梯度下降算法。然后重点分析研究深度Q学习与经验重放，并在之前传统算法的基础上对深度Q学习优先经验重放算法进行改进分析研究，最后形成改进的深度Q学习优先经验重放算法。

# 第五章 实验及结果分析

借助深度学习，强化学习以及深度强化学习算法，本章将通过实验仿真对比分析各个算法在智能体决策中所发挥的作用，并将会证明第四章中新的深度Q学习改进优先经验重放算法比之前的算法在智能体决策中发挥的作用能够使智能体更加快速高质量的完成游戏。

## 5.1 实验软硬件环境

在实验的软件与程序方面，深度学习这一概念虽然出现时间不是很长，但是现在它的发展势头很猛，和它相关的框架算法与工具也是层出不穷日新月异。目前正在使用keras框架。

Python在目前的深度学习领域成为了最受欢迎的编程语言。除此之外还有着不一而足的深度学习框架算法等能够为所用，而且可以发现全部大型科技公司都在使用着他们各不相同的框架算法。

在实验中，计算机选择Ubuntu 16.04 系统版本，使用Python计算机语言进行编程开发试验，然后在其中添加很多必要的开发包，其中最重要的就是keras框架。

## 5.2 实验方案设计

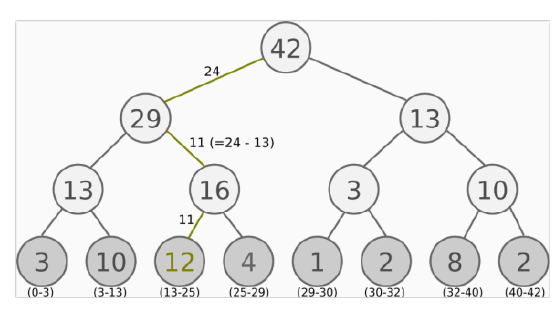
### 5.2.1 实验软件程序架构

在本论文中的所有实验所使用的Python语言版本均为 3.5.3 版本，在程序中的一些架构可在之后的图片中看到。gym版本为0.8.0版本。Open AI Gym在研发和比较强化学习算法时是一种比较实用的工具包。

强化学习和有监督学习的评测不一样。有监督学习的评测工具是数据。一旦拥有一些有标注的数据就可以实施有监督学习的评测。强化学习的评测工具是环境。能够提供给一个环境给智能体来分析运行，才能评价智能体的策略的好坏。Open AI Gym提供各种环境的开源工具包,它最重要的功能就是提供各种强化学习环境，可以被Python程序直接调用实现游戏环境的载入。

如图5-1所示，Sum Tree类是一种树形结构, 每片树叶存储每个样本的优先级p，每个树枝节点只有两个分叉，节点的值是两个分叉的和，所以Sum Tree的顶端就是所有p的和。正如下所示，最下面一层树叶存储样本的p,叶子上一层最左边的13=3+10，按这个规律相加，顶层的root就是全部p的和了。

抽样时，会将p的总和除以batch size，分成batch size那么多区间,(n=sum(p)/batch\_size)。如果将所有node的priority加起来是42的话,如果抽6个样本,这时的区间拥有的priority可能是这样：[0-7],[7-14],[14-21],[21-28],[28-35],[35-42]。



##### 图 5-1 Sum Tree 树形结构

然后在每个区间里随机选取一个数。比如在第 4 区间[21-28]里选到 24，就按照这个 24 从最顶上的 42 开始向下搜索。首先看到最顶上 42 下面有两个子节点，拿着手中的 24 对比左边的子节点 29，如果左边的子节点比自己手中的值大，那就走左边这条路，接着再对比 29 下面的左边那个点 13，这时，手中的24比13大，那就走右边的路，并且将手中的值根据 13 修改一下，变成 24-13=11。接着拿着 11 和 13 左下角的 12 比，结果 12 比 11 大，那就选 12 当做这次选到的priority，并且也选择 12 对应的数据。所以Sum Tree结构用来确定优先级概率分布情况。

然后Memory类是用来落实经验重放方法的类，类中包含存储，抽样，更新，优先级等的具体操作方法等等功能。

最后DQNPrioritized Replay类用来处理深度强化学习算法中的各种小细节问题，包括神经网络层数和类型，状态转移序列的存储，智能体动作的选择，各种超参数的配置等问题。

### 5.2.2 实验方案设计

之前各章中所提到的算法有深度Q学习随机经验重放，传统深度Q学习优先经验重放算法和改进深度Q学习优先经验重放算法等。对于这些算法的评价对比要有之后一些过程。第一步是要创立两个游戏集，其中一个要用来训练，另外的会用来测试。然后接下来要运用训练集游戏调节各项参数与改进网络层次架构设计，之后要运用测试集游戏评估对比文章中所使用的各种算法的实际性能。

在实验中所用到的环境来自Open AI Gym，共有7款游戏，分为训练用的2款游戏和测试用的5款游戏。下面来分析介绍一下这几款游戏。

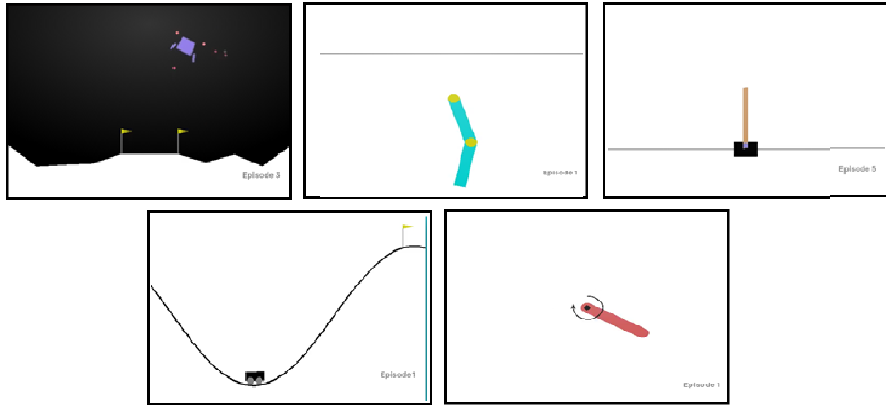
用作训练集的两款游戏是Sea quest（深海游戈）和Breakout（打砖块），测试的5款游戏是Lunar Lander（飞行器着陆）、Acrobot（多节摆动器）、Cart Pole（倒立摆）、Mountain Car（小车爬山）、Pendulum（平衡木）。

如图 5-2 是作为训练集的两款游戏 Sea quest 和 Breakout：



##### 图 5-2 训练用两款游戏

如图 5-3 是作为测试集的五款游戏Lunar Lander、Acrobot、Cart Pole、Mountain Car、Pendulum：



##### 图 5-3 测试用五款游戏

在运用两个训练集游戏训练完成后，在全部的 5 个测试集游戏中，所运用的深度学习神经网络层次结构，学习算法还有各种超参数的设置基本保持不变。其中超参数是开始学习之前就已经设定好值的参数，而别的参数要通过训练出现。然后需要改变的地方是根据测试集各个游戏的不同的状态改变游戏评分得分与输出的节点数量。

在各个游戏中的游戏得分状况差别会比较大，的目标是让所有的游戏都能够在同一个架构中进行测试，在对游戏操作的得分机制进行设计时，当游戏得分正时奖励加十，游戏得分为负时要惩罚，奖励为负十，分数不变的时候奖励也不变，为零。

一般来说各个游戏都有它们自己不同的操作方法，所以要保证在玩各个游戏时需要改变输出层的节点的相关参数数量。通常情况下向左向右向上向下移动的游戏操作应在共同的输出单元，而其他操作比如说突然加速减速，攻击等游戏操作应放置在相对应的输出单元中。

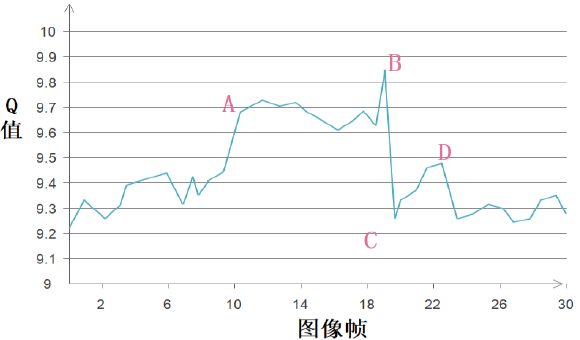
前面提到了三种用于智能体玩游戏的算法，包括深度Q学习随机经验重放，传统深度Q学习优先经验重放算法和改进深度Q学习优先经验重放算法。这几种算法在各个游戏中要用到之后描述的评价方式。在游戏环节中，每一轮游戏成为一个episode过程，每一个episode过程都起始于重置命令后的那一帧画面图像，最后在收到相应的游戏结束标志条件后终结这一episode过程，亦或是超出5分钟的实际智能体玩游戏时间的最后一帧画面图像。

在一个episode时期内智能体每隔4帧画面执行一次操作，各个算法在不同游戏的很多个episode过程中学习并进行比较评估。最后运用各种图表来直观展示各个不同的算法在智能体玩游戏时的情况并进行比较从而得到最优的算法。

## 5.3 实验结果分析

### 5.3.1 改进算法的直观模型策略分析

为了更好的理解算法模型学到的策略，把以训练集Sea quest游戏为例子。



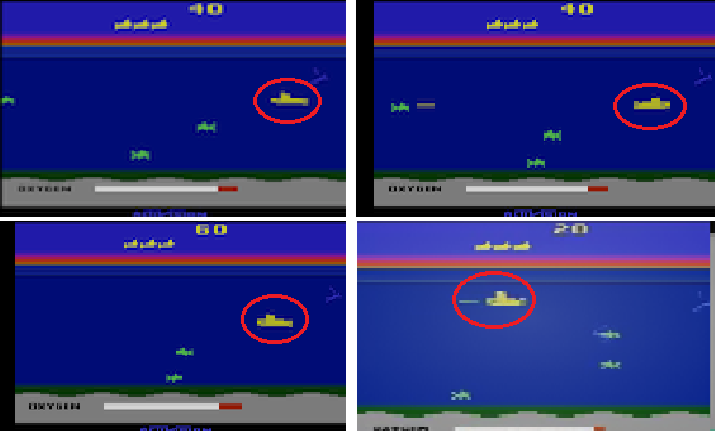
##### 图 5-4 Sea quest 游戏连续 30 帧画面的 Q 值曲线

在游戏进行时其中的随机一个点开始连续的 30 图像帧画面截取并与之相匹配的Q值用曲线图表示出来。任意一个图像帧画面的Q值都根据公式 3-4来计算。如图 5-4 为Sea quest 游戏中的 Q 值曲线图。

下面是四个时间点分别匹配的游戏图像帧画面，与 Q 值曲线相互比对。

A 点时黄色的小潜艇发现目标并锁定，并随时准备发射鱼雷将其打掉，所以此时的 Q值明显升高了；B 点时潜艇已经发射了鱼雷并且马上就会击中目标，所以此时的奖励会是最大的，随意这时的 Q 值最大；

C 点时潜艇一直在寻找目标并且没有找到，此时 Q 值会大幅下降；最后在 D 点时潜艇发现目标并刚刚发出鱼雷但并不知道会不会命中目标，所以此时 Q 值有跃升但没有达到最大值。



##### 图 5-5 游戏画面分析

游戏画面中的潜艇只占据整个画面的不多的像素值，深度强化学习网络能够从不多的像素改变中学习到正确的策略，这表明本文所改进的算法应用于网络结构中能够发现并提炼出有效的并且高层次的特征（潜艇与目标相对位置），而且算法能够从这些特征环境中学习到最优的动作价值函数并应用到决策中。

### 5.3.2 各算法 CNN 网络层数架构选择

现对测试集两个游戏进行测试并运用打砖块游戏进行分析。在同一个训练集游戏打砖块（Breakout）环境中，分别运用DQN随机经验重放、DQN优先经验重放和改进DQN优先经验重放三种算法在在6种含有不同隐含层的深度神经网络中进行测试。

|  |  |
| --- | --- |
| 模型编号 | 模型描述 |
| CNN1 | 单隐层，只有 1 层卷积层 |
| CNN2 | 1 层卷积层，1 层池化层，1 层全连接层 |
| CNN3 | 2 层卷积层，2 层池化层，1 层全连接层 |
| CNN4 | 3 层卷积层，3 层池化层，1 层全连接层 |
| CNN5 | 5 层卷积层，5 层池化层，1 层全连接层 |
| CNN6 | 7 层卷积层，7 层池化层，1 层全连接层 |

##### 表 5-1 各不同隐含层深度卷积神经网络结构描述

首先在三种算法中，当达智能体在游戏达到100分时不计算所用时长。所用的episode（轮次）各是多少。由表5-2可知：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| （/轮次） | CNN1 | CNN2 | CNN3 | CNN4 | CNN5 | CNN6 |
| DQN 随机经验重放 | 270 | 210 | 106 | 74 | 37 | 26 |
| DQN 优先经验重放 | 208 | 130 | 85 | 64 | 31 | 21 |
| 改进 DQN 优先经验重放 | 193 | 119 | 73 | 51 | 24 | 19 |

##### 表 5-2 达到相同分数时各算法所用的episode大小

在游戏达到相同的episode等于50时不计最后得分，各个算法在不同CNN模型中训练所用的对应时间如表5-3所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| （/s） | CNN1 | CNN2 | CNN3 | CNN4 | CNN5 | CNN6 |
| DQN 随机经验重放 | 674 | 1091 | 1680 | 2450 | 4540 | 13370 |
| DQN 优先经验重放 | 537 | 910 | 1530 | 2324 | 3820 | 11485 |
| 改进 DQN 优先经验重放 | 497 | 848 | 1468 | 2090 | 3429 | 9863 |

##### 表 5-3 达到相同 episode 时各算法所用时间

Tab.5-3 Time spent by each algorithm when reaching the same episode

最后在游戏的运行相同时间等于800s时，忽略此时各算法运行了多少episode，各个算法在不同CNN模型中训练所得分数如表 5-4 所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| （/s） | CNN1 | CNN2 | CNN3 | CNN4 | CNN5 | CNN6 |
| DQN 随机经验重放 | 17 | 28 | 37 | 51 | 68 | 75 |
| DQN 优先经验重放 | 18 | 24 | 37 | 57 | 76 | 76 |
| 改进 DQN 优先经验重放 | 16 | 20 | 38 | 55 | 73 | 83 |

随着时间的推移，单位时间内对智能体玩游戏的得分影响也逐渐减小，这是因为在游戏刚开始时智能体会犯的错误较多，所以学习的也最快，之后错误越来越少，学习的也就越来越慢了。

根据以上的表格与分析，可以看出在智能体玩游戏时也不是深度卷积神经网络的隐藏层越多越好。层数越多所消耗的时间也越多，同时对硬件的要求也越高。但为了保证深度学习的效果也不能太少，结合本文的实际情况，发现在应用 CNN5模型时深度学习对强化学习情况下选择最优游戏动作从而得到较高分数的效果最佳，同时所用时间也在可以接受范围内，所以本文选择使用CNN5 深度卷积神经网络模型用于深度学习发现图像游戏画面中的环境状态。

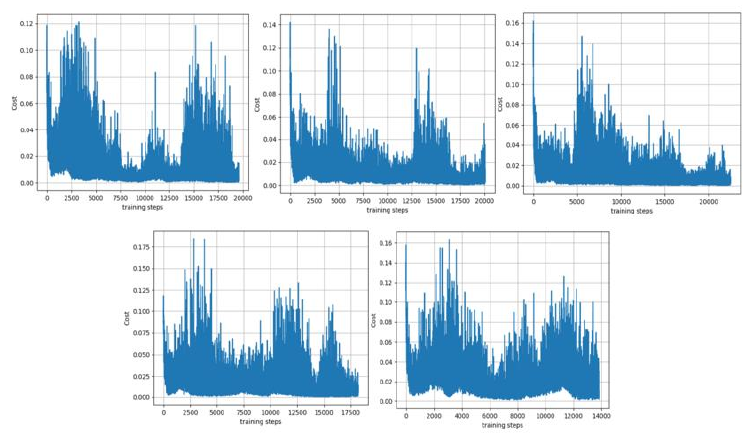
同时也可以从各个图表中看出，相较于深度 Q 学习随机经验重放算法与深度Q 学习优先经验重放算法，改进深度 Q 学习优先经验重放算法无论在所用训练时间长度与所用 episode 轮次，还是在相同训练所用时间与轮次时所得到的游戏分数上都比之更有优势。

在训练集游戏将深度网络中的权重参数训练的比较成熟之后就可以用测试集游戏来检验三种算法在游戏中智能体的表现了。

### 5.3.3 各算法代价函数分析

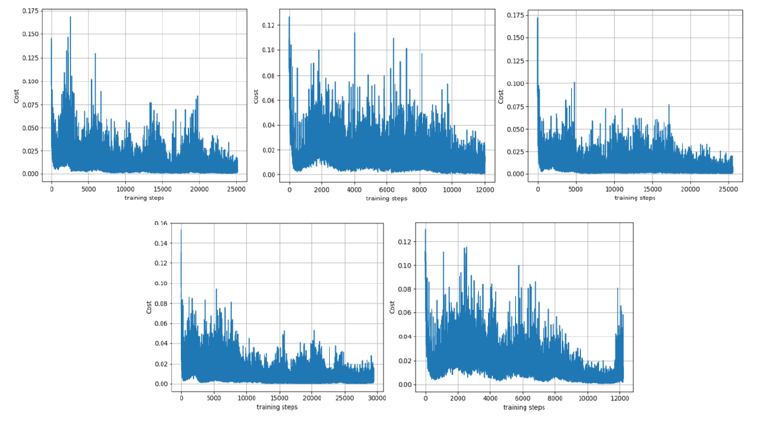
公式 4-7 说明了提出的代价函数，也可以叫做损失函数。它使用贝尔曼方程来计算梯度下降的损失。为了提高训练稳定性，使用经验重放。这是一个内存模块，用于存储已经探测到的状态空间部分的内存。经验通常随机选择，然后作为训练数据重放到深度神经网络。误差大小与权重更新速度成正相关分布。这是一个很好的性质。

下面是用于测试集 5 个游戏三种算法的代价函数曲线，如果抛开可能存在的 误差，可以看出随着训练的持续，代价函数数值总体是下降的趋势，从中可以看出深度卷积神经网络中权重的改变越来越精确的反应出强化学习中智能体对游戏的操纵越来越向着正确的方向改变，代价函数越小，说明深度学习网络神经元之间的权重参数更加合理，从而更加准确的发现游戏图像的特征并输出最优的强化学习环境。然后智能体就可以更迅速和准确的从深度学习提供的游戏图像环境中选择最优的动作，使得游戏的得分越来越高。



##### 图 5-6 DQN 随机经验重放算法的 5 个测试集游戏代价函数曲线

图 5-6 为运用 DQN 随机经验重放算法时的代价函数曲线。 纵坐标为 Cost代价函数值，横坐标为训练步数。



##### 图 5-7 DQN 优先经验重放算法的 5 个测试集游戏代价函数曲线

图 5-7为运用 DQN 优先经验重放算法时的代价函数曲线。 纵坐标为 Cost代价函数值，横坐标为训练步数。

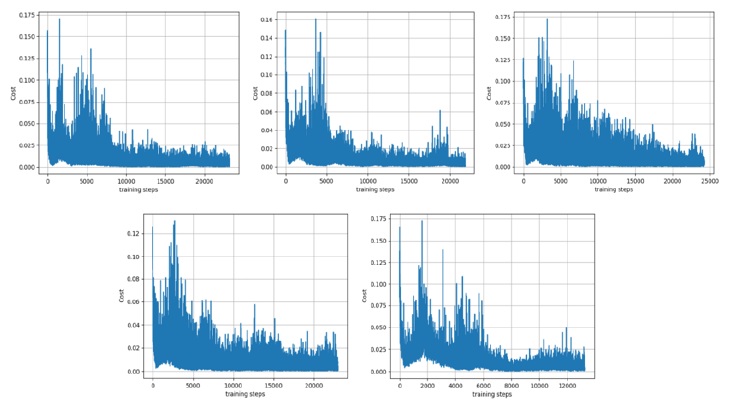


图 5-8 改进 DQN 优先经验重放算法的 5 个测试集游戏代价函数曲线

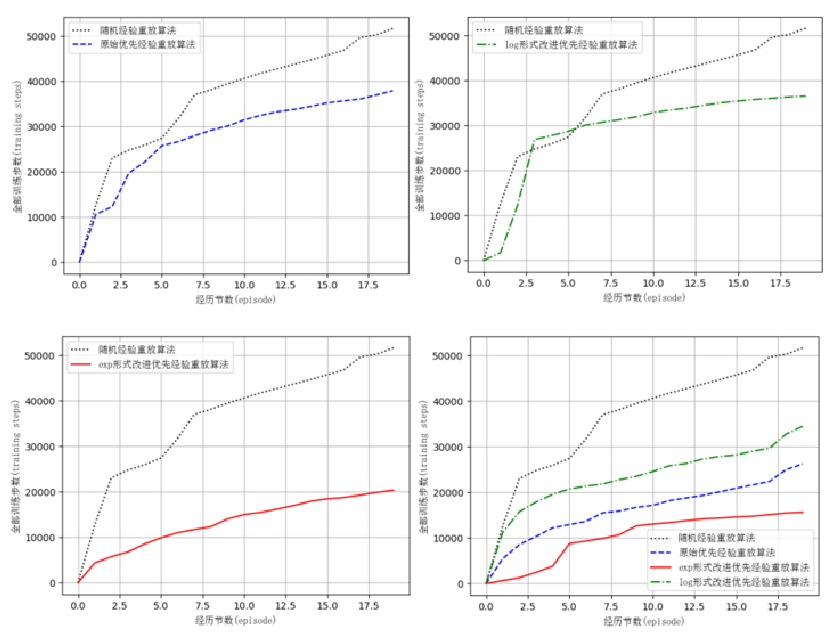
图 5-12 为运用改进 DQN 优先经验重放算法时的代价函数曲线。 纵坐标为Cost 代价函数值，横坐标为训练步数。

总体上可以看出改进深度 Q 学习优先经验重放算法的代价函数变化曲线比较深度 Q学习优先经验重放算法和深度 Q 学习随机经验重放算法，新的算法代价函数曲线下降更加平缓稳定，不会有太大的中段跃升，下降幅度也更大从而更好的保证智能体在动作选择时更加准确做出正确动作从而在游戏中获得更高的分数。

### 5.3.4 各算法效率分析

在本小节中，将对比之前三种算法，即深度 Q 学习随机经验重放算法，深度Q学习优先经验重放算法与改进深度 Q 学习优先经验重放算法在测试集 5 个游戏中智能体对游戏取得相同游戏目标的效率分析。设定当智能体在 5 个游戏中完成游戏的设定目标后为一个 episode，比如小车爬山游戏中小车到达山顶拿到小旗子后就结束一轮 episode 并开始新的一轮游戏。所以可以通过每一轮中完成智能体游戏所花费的步数（transitions）来判断这个算法在游戏中的速率表现。

首先对原始实验进行模拟，比较优先经验重放与随机经验重放的差别，运用针对优先级到概率的映射函数（3-5），通过计算机的仿真得到对比曲线，它的纵坐标是训练所用步数，横坐标是循环节数（episode）。然后类似的，通过公式（4-10）与公式（4-11）得到 log 形式的映射函数与随机经验重放对比和 exp 形式的映射函数与随机经验重放对比。最后得到全部映射函数代表的三种算法与原始的深度 Q 学习随机经验重放算法对比曲线图。如下图 5-13 所示：



##### 图 5-8 各算法取得相同游戏目标的效率表现

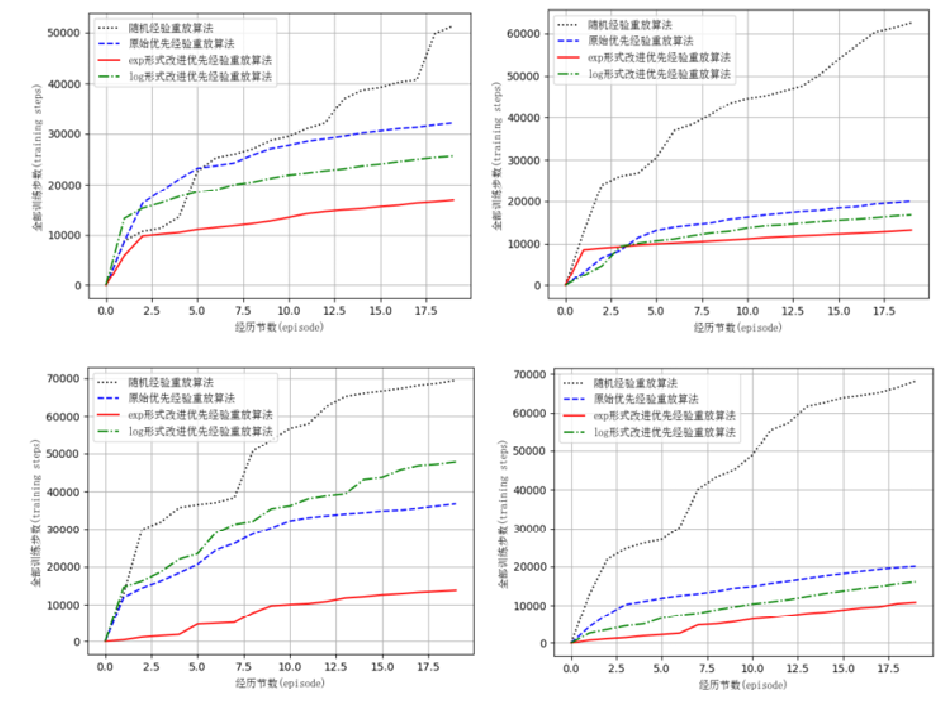
从左到右，从上到下依次是原始深度 Q 学习随机经验重放算法与深度 Q 学习优先经验重放算法智能体学习效率对比；原始深度 Q 学习随机经验重放算法与基于公式 4-10 的深度 Q 学习优先经验重放算法智能体学习效率对比；原始深度Q 学习随机经验重放算法与基于公式 4-11 的深度 Q 学习优先经验重放算法智能体学习效率对比；所有算法智能体学习效率对比；智能体学习效率对比即是智能体对游戏画面环境的感知并作出相应的最优的动作改变游戏图像环境并返回奖励的过程的能力对比。

从左到右，从上到下依次是游戏 Lunar Lander（飞行器着陆）、Acrobot（多节摆动器）、Cart Pole（倒立摆）、Pendulum（平衡木）前文中提到的各个算法智能体学习效率对比。

通过图 5-7 与 5-8 可以看出相比于深度 Q 学习随机经验重放，优先经验重放的运用对智能体学习玩游戏的速率有了一些提升，也就是智能体基于深度学习对神经网络各隐含层中的神经元的权重参数的调整反映出的对游戏环境的影响改变，然后通过 Q 学习算法做出更加优化的动作选择策略。

同时基于公式（4-10）的深度 Q 学习优先经验重放的运用对学习速率有了一些提升但是不明显。最后基于公式（4-11）的改进深度 Q 学习优先经验重放算法的运用相比于其他算法对学习速率有了明显的提升。

剩余 4 个游戏的各算法效率对比如下图 5-14 所示：



##### 图 5-9 各算法在另外 4 个游戏中取得相同游戏目标的效率表现

像游戏的深度强化学习改进算法研究 54

而保证将算法性能发挥到极致（满分设定为 100 分）。

图 5-15 各算法在游戏中最高得分

Fig. 5-15 The highest score of each algorithm in the game

由图表可知虽然在比如

Crat Pole

游戏中的得分没有比深度

Q

学习有限经验重

放算法高，但改进的深度

Q

学习优先经验重放算法在总体上对比其他两种之前的

算法来说能在游戏中取得更高的分数。

与深度

Q

学习随机经验重放算法和传统深度

Q

学习优先经验重放算法相比，

改进的深度

Q

学习优先经验重放算法在智能体决策智能体的游戏策略中的表现更

加出色。相较于之前的深度

Q

学习优先经验重放算法在达到相同游戏结果的情况

下游戏运行所用时长平均能够减少 41.6%；所花费的训练轮次平均减少 29.4%；在

最终游戏得分方面相较于之前的深度

Q

学习优先经验重放算法平均能够提升

9.43%。

5.4 本章小结

本章先介绍实验软硬件环境，实验软件程序架构与实验方案设计并进行实验

结果分析，通过一些游戏截图说明改进的算法能学习到图像信息并做出合理的决

策方案，进行各算法 CNN 网络层数架构选择与算法代价函数分析，各算法效果分

析还有各算法游戏分数对比，并由实验结果的分析研究最后得到结论。 用于智能体决策的深度强化学习改进算法研究

56

1.在接下来的研究中将会寻找更加有效的降低代价函数的深度学习神经网

络结构并持续调整各个隐层神经元之间的权重参数。

2.寻找性能更加强劲的硬件设备加速深度学习算法的训练过程并在同时减

少训练时间。软件方面也会尝试不同的编程语言与版本，对比发现能否有效提高

训练效率。

3.继续寻找可能存在的更加优化的经验重放单元与选中概率的映射函数，从

而可以更加高效的完成深度强化学习算法的训练。

4.本文中的智能体是单智能体，只能在比较低端的游戏中发挥作用，下一步

计划在更加复杂的游戏环境中使用多智能体来加速训练过程。

青岛科技大学研究生学位论文

55

总结与展望

1.总结

进入二十一世纪以来一种叫做深度强化学习的算法开始迅猛发展，深度强化

学习算法是将深度学习与强化学习的特点结合起来形成的一种新的有效的人工

智能学习算法。它可以运用到诸如机器人行为控制，汽车自动驾驶，图形游戏控

制等诸多领域研究之中。本文主要研究将深度强化学习算法运用于操控图形游戏

使得智能体能够在游戏中取得更好的成绩。

智能体决策也能够理解为视频游戏，视频中的帧画面即为一张图像。在本文中，

研究了如何更好地教机器玩游戏并取得最好的成绩。

在之前的相关研究中对于智能体决策的算法有强化学习与浅层神经网络的结

合，也有一些智能体决策运用了深度强化学习算法。不过直到 2013 年才出现了第一

种应用深度 Q 学习与经验重放算法相结合的新算法，即深度 Q 学习经验重放算

法。但是这种算法也存在一些瑕疵，知道一些经验要相比于别的经验对参数的改

变发挥更多的作用，因此 Tom·Schaul 等人提出要把参数更新更多的倾向使值估

计变化更大的经验。然后深度 Q 学习优先经验重放算法应运而生。

深度 Q 学习优先经验重放算法比较好的解决了增加更有意义的经验的重放

次数，但是只是模糊的给出了一个概率公式，虽然也达到了目的，但是还会有更

好的效果可以实现。所以在此基础上本文提出了改进深度 Q 学习优先经验重放算

法。改变经验单元的重放的概率公式，从中选取更优的概率公式并实现更好的经

验重放效果。

最后本文通过一系列实验，在训练集与测试集共七个游戏中的实际测试，包

括包括通过一些游戏截图说明改进的算法可以学习到图像信息并做出合理的决

策方案，还有各算法 CNN 网络层数架构选择，各个算法代价函数分析，各算法

效果分析还有各算法游戏分数对比，可以得出结论。

2.展望

本课题对用于智能体决策的深度强化学习算法进行了研究，提出了一种新的深

度强化学习算法用于智能体的训练完成游戏。目前来看已经获得较好的结果，但

仍然有在其他方面持续改进的余地。

参考文献

Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.（第一篇）

刘全, 翟建伟, 章宗长,等. 深度强化学习综述[J]. 计算机学报, 2018(1):1-27.