强化学习 道路交通

摘要

第1章 绪论

## 1.1 研究背景和意义

强化学习算法最早在

## 1.2 国内外研究现状

## 1.3 主要研究内容

## 1.4 本文结构安排

第2章 相关概念与理论基础

## 2.1深度学习

### 2.2.1 多层感知机网络

2.2.2 卷积神经网络

2.2.3 循环神经网络

## 2.2强化学习

2.2.1 马尔科夫决策过程

马尔科夫决策过程是强化学习算法应用过程中的。强化学习的要素包括智能体和环境。智能体根据环境状态输出动作，而环境根据动作得到下一个状态和奖励值。智能体通过和环境的不断交互，最终学习到一个相对稳定的策略，以最大化奖励值作为目标，达到学习效果。一般情况下，强化学习的模型可以表示为马尔科夫决策过程（MDP），其主要包括了<,,,>在内4个元素。智能体在时刻时，具有状态，当该智能体选择行为后，获得来自环境的奖励，并得到下一个状态[1]。如图1为马尔科夫决策过程下，强化学习智能体与环境的交互图。

环境

智能体

行为

奖励

下一状态

图1 智能体和环境的交互图

在智能体和环境的不断交互过程中，智能体的主要目标为，在一个时间周期内，获取到最大的累积收益，智能体在t时刻做出决策的过程依据为最大化的期望回报。期望回报本身不仅与当前的时刻获得的奖励有关，还与在该时间范围内后续一系列奖励有关。期望回报公式如下：

（1）

由于未来奖励具有一定的概率性和不确定性，因此一般情况下，期望回报在对奖励的累积过程中添加衰减因子作为超参数，表示未来奖励对当前期望回报本身所占权重的降低。最终周期内的期望回报表示公式如下：

（2）

2.2.2 策略迭代与值迭代

2.2.3 Sarsa算法与Q学习

2.2.4 Dyna算法与其拓展

## 2.3深度强化学习

### 2.3.1 DQN算法与其拓展

Q-Learning算法结合了神经网络能够实现更复杂的状态表示，另外计算机运算能力大幅提升，使得基于值迭代的强化学习算法的应用范围变得广阔。以DQN算法为主，各种改进方案。

目前，普遍使用的DQN的算法描述如下：

1. 对每个回合进行循环
2. 初始化状态
3. 对每个时间步进行循环
4. 使用的概率选择随机行为，否则选择
5. 执行行为，获得奖励，得到下一个状态和
6. 存储目前获取到的信息：，,,到缓存中
7. 设置
8. 使用梯度下降更新网络的参数：
9. 每步重置
10. 结束该回合
11. 结束

DQN算法存在过估计问题，且采用单个进程进行计算，效率较低，具有改进空间。

### 2.3.2 DPG算法与其拓展

传统的值迭代算法由于输出内容为每个行为s,a的价值估计，状态可以使用神经网络的输入表示，行为a却往往表示为离散形式。在具体的算法实现上，神经网络的输入层的每个神经元代表状态s的一个维度，输出层表示每个动作的价值估计，离散动作a的取值数量和输出层相关。若动作a的取值为多个维度的连续值，则基于价值估计的算法则无法适用。因此引入了基于策略梯度的一些算法。

强化学的目标为最大化奖励值

<https://www.cnblogs.com/bai2018/p/12888101.html>



在智能体经历的一条马尔科夫决策轨迹过程中，使用表示该条轨迹，表示该条轨迹出现的概率，表示出现状态的概率，表示在状态下，选择行为的概率。则可以表示为如下所示。



在每条轨迹中，产生了累积奖励，最终的期望奖励表示如式子所示。



强化学习算法的目标是最大化期望奖励，因此策略梯度算法的目标则是采用梯度上升的方法，对参数求导，获取最大化的期望奖励。如下的求导过程中，



由于：



因此：



通过以上，得到策略梯度的训练更新过程如下式子所示。



然而，在实际执行过程中，若奖励值都为正值，且存在采样偏差，从而导致动作概率分布函数估计误差，因此增加了偏置项，使得奖励值出现负值，只增加高于平均奖励R对应的轨迹策略，如下式所示。





另外，整条轨迹的累积奖励不足以度量每个输出行为的概率好坏，因此在计算过程中，应对应到每个动作选择的评价。在轨迹中的行为一定程度上决定了后续的状态行为对，因此使用衰减因子来表示后续的动作价值。更新公式变为如下所示。



以上的更新方法在每次算法更新过程中都需要采样轨迹，采样的轨迹被学习后就直接被丢弃，造成了交互经验的浪费和采样频率增高。

因此使用重要性采样，将在线策略转为离线策略。其中TRPO和PPO算法都将其进行了转换，更新公式。





### 2.3.3 Actor-Critic类算法

DDPG算法

### 2.3.4 多智能体的强化学习

$\nabla $

### 2.3.5 元强化学习

近年来，元学习的一些思路应用到了强化学习领域。传统强化学习算法从特定的环境中进行学习，当智能体根据环境训练完成后，若环境发生内部变化，则难以适应于环境变化，模型需要重新训练，造成计算计算成本的损失。

元学习的目标是在各种学习任务上训练一个模型，这样它就可以只使用少量的训练样本来解决新的学习任务。有监督的问题的训练过程中，损失函数如下式所示。



和分别是从任务中采样得到的输入和输出，从而获得了任务的损失，最小化损失，以对回归模型进行训练。

为每个任务更新过程的步长，每个函数的损失值为



元学习的目标是获得以下函数的最小化。



参数更新的过程中，



输入：元神经网络参数

输出：

1. 随机初始化元神经网络参数
2. 循环迭代一定次数
3. 从中采样任务
4. 对每个训练得到一个参数
5. 对采样的任务进行参数的更新
6. 返回元学习参数

通过元学习获取到的参数由于最小化了多个任务的目标函数，因而最终得到的神经网络对各个任务更加敏感，通过少量训练即可拟合到单个任务中。

由于基于策略梯度的强化学习算法在计算价值函数的目标值时，采用从后向前的过程，得到的状态价值属于无偏估计，因此应用效果更佳。

而基于值迭代的强化学习算法得到的价值估计，具有方差较小，偏差较大的特征，因此在应用元学习方法上需要进行改进。

<https://zhuanlan.zhihu.com/p/106210146>

第3章 模型构建与求解

## 3.1 模型符号说明

### 3.1.1 名词解释

信号灯相位：在单个路口的交通信号系统中，针对所有路口的交通信号灯组合，为该时刻的信号灯相位。当交通环境的信号灯发生变化，对应的信号灯相位发生变化。通常情况下，国内采用靠右行驶的交通规则，右转信号灯默认为常绿而被省略，因此，理论上在一个十字路口中，每个路口的信号灯相位个数为2\*2，共有（2\*2）4种可能性的信号灯相位表示。

相位周期：也叫信号周期。在传统的交通信号控制系统中，信号灯相位的变化具有一定的时间规律。从一个起始的信号灯相位开始，经过一定的时间，按照固有的信号灯变化情况进行信号灯相位变化，最终又回到起始相位，为一个相位周期。

全红时间：在一个信号灯路口上，当该路口上的所有信号灯都为红色的持续时间，为信号灯的全红时间。（全红时间也是绿灯间隔时间的重要组成部分，在3秒的黄灯时间不足以完全清空交叉口的车辆时，设置全红时间非常必要，合理的全红时间可以大大提升交叉口的通行效率，保障交叉口的行车安全。但是全红时间会增加交叉口的控制延误，特别是控制方案中相位数较多时，延误增加显著，因此必须合理测算全红时间。https://www.sohu.com/a/128695015\_455170）

路口吞吐量：也称为交通流量，指在一定时间内，通过路口的车辆总数。

饱和流量：在有效绿灯时间内的最大车辆通行量。受到车道宽度、车道坡度、大车混入率、转弯车流比率、车道功能**等**，单位为量/有效绿灯小时

### 3.1.2 变量表示

交通信号灯控制环境在建模过程中，往往需要采用形式化的方式进行描述，以便于理清求解思路和方法。下表列出了交通信号灯控制过程的基本变量和定义。其中的变量表示仅适用于该小结，用于交通系统中更深入的变量名进行解释。

|  |  |  |
| --- | --- | --- |
| 变量名 | 变量表示 | 自变量说明 |
| 绿信比 |  | 为有效绿灯时间，为信号周期时间，为周期损失时间 |
| 有效绿灯时间 |  | 绿灯时间，黄灯时间，部分损失时间，前损失时间，后补偿时间，后损失时间 |
| 绿灯间隔时间 |  | 黄灯时间，全红时间 |
| 周期损失时间 |  | 为前后损失时间，为相位数  为全红时间 |
| 前后损失时间 |  | 启动损失时间，停车损失时间 |
| 周期有效绿灯时间 |  | 为信号周期时间，为周期损失时间 |
| 有效绿灯时间 |  | 相位有效绿灯时间，黄灯时间，  为前后损失时间 |
| 饱和流量 |  | 车道的基本饱和流量（常数，查表得），校正系数（和车道宽度、大车概率、专用车道、自行车、行人、等有关） |
| 通行能力 |  | 为车道的通行能力，为车道的饱和流量，为车道的绿信比，为车道的有效绿灯时间，为信号周期时间 |
| 饱和度（一般设置在0.8-0.9，运行条件较好，接近1则交叉口通行条件迅速恶化） |  | 车道的实际流量  车道的通行能力 |
| 总饱和度 |  | 为车道的饱和度 |

一般情况下，交通流量较小，相位数较少则取70s左右，较大则取180s左右。

张勇刚 道路交通控制技术及应用[M] 55-82

泊松分布Poisson分布。

泊松分布的参数λ是单位时间(或单位面积)内随机事件的平均发生次数。泊松分布适合于描述单位时间内随机事件发生的次数

适合于泊松分布的事件需要满足下述 3 个条件：

1. 这个事件是一个小概率事件。

2. 事件的每次发生都是独立的。

3. 事件的概率是稳定的。

例子：公交车站有很多不同线路的公交车，平均每 5 分钟会来两辆公交车，求解 5 分钟内来 5 辆公交车的概率有多大？

解析：在这里问题里，随机事件（公交车来到）拥有固定的平均概率（每 5 钟 2 辆），每次事件随机且独立（不同的线路），所以这个随机事件在单位时间（5 分钟）内出现的次数就近似的服从语泊松分布。

最终代入公式求解得：P(X=k=5) = (2^5 / 5!) \* e^(-2) 概率约为 0.0361

————————————————

原文链接：<https://blog.csdn.net/Jmilk/article/details/79798170>

伯尔分布

概率密度函数

累积概率分布

到目前为止，定时信号的配时方法在国际上主要有英国的TRRL法（也称Webster法）、澳大利亚的ARRB法以及美国的HCM法等。在我国有“停车线法”和“冲突点法”等方法。随着研究的不断深入，定时信号的配时方法也在进一步的改进之中。

<https://baike.baidu.com/item/%E9%9F%A6%E4%BC%AF%E6%96%AF%E7%89%B9%E9%85%8D%E6%97%B6%E6%B3%95/16168485?fr=aladdin>

## 3.2 传统交通控制算法

### 3.2.1 基于经验性的配时算法

（1）黄灯时间的确定



为驾驶人的反应时间，经验取值为0.8-1秒，进口道速度，为减速度，为道路宽度，为车身长度。

（2）全红时间的确定



为绿灯车头停止线到冲突点距离，为上一相位尾车从停止线到冲突点的距离，为车辆长度，为交叉口设计速度

### 3.2.2 韦伯斯特配时法

根据韦伯斯特算法，最佳的相位周期可以表示为如下形式，



其中，为周期损失时间，为临界车道组流量比，、为系数项

当为1，为0时，有最短信号周期可以表示为如下形式，



为周期损失时间，为临界车道组流量比。交叉路口的饱和度为1，控制效果不理想。

韦伯斯特平均延误。



绿信比为，为周期时长，为总饱和度，为绿信比之和。

1. 计算临界车道组流量比
2. 计算最佳相位周期
3. 对最佳相位周期进行剪裁clip(,,)
4. 绿灯时间
5. 返回相位周期，每个相位对应的绿灯时间

### 3.2.3 SOTL算法

输入：某个时间点的内部车辆总数，当前绿灯相位持续时间，最小绿灯时间，

输出：

1. foreach (timestep) do
2. 
3. if >then
4. if not （0<<）then
5. if（）then
6. 设置交通信号相位到相位i
7. 
8. end
9. end
10. end
11. end

在SOTL方法中，每个路口的每条线路，使用一个内部车辆计数器。初始时刻和该线路交通灯变为红色时设置为0，然后再每个时刻统计到达该车道的车辆数。若车辆数到达车辆数阈值，则信号灯转换到该相位，前一个相位变为黄灯并持续一定时间。

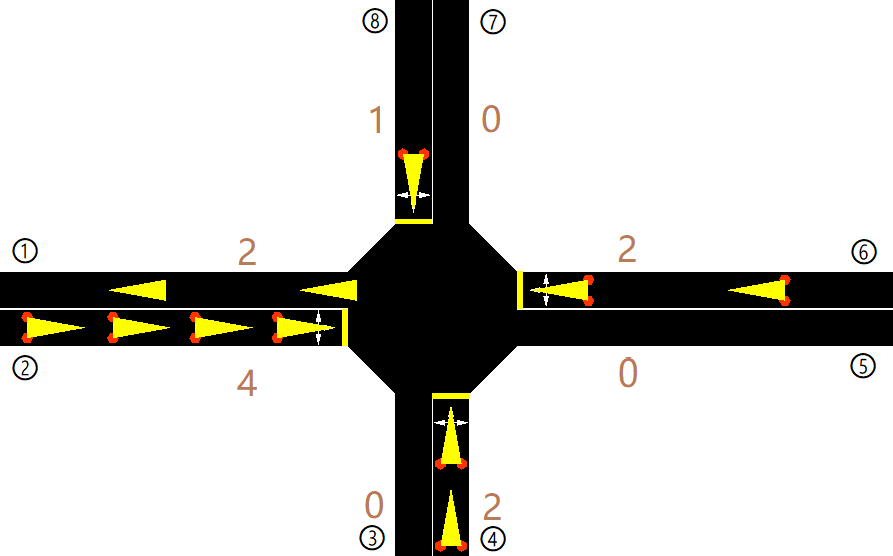
### 3.2.4 最大压力算法

输入：最小绿灯时间

输出：

1. foreach (timestep) do
2. if < then
3. =+1
4. else
5. =0
6. return argmax({Pressure(p) for p in P})
7. end if
8. end

在最大压力方法中，设置最小绿灯时间后，使用作为当前相位的绿灯持续时间计数器。当达到最小绿灯时间后，将清零，然后输出当前压力最大的相位。最大压力的计算方式可以采用相位对应线路的入口车辆数减去出口车辆数来表示，如图所示所示的交通路口，路线2到路线5的压力为4（4-2），而路线6到路线2的压力为0（2-2）



## 3.3 交通环境构建

### 3.3.1 平台和编程语言

强化学习的测试平台有很多，如gym、mujoco、

强化学习的算法实现平台也有很多。

目前在交通领域，常用的交通仿真软件有SUMO、VISSIM等，其中SUMO提供了TRACI编程接口，可供Python、C++、Java等语言进行调用。这些仿真软件较为权威，但仿真速度有待提高。CityFlow提供了一种更快的仿真速度，但是具有限制，如仿真环境不支持Windows操作系统，仿真参数和效果有限等。

在选择过程中，考虑到Python编程语言的用户友好型，SUMO软件资料和文档较多、CityFlow软件计算速度较快的因素，配合使用以上方法，开发了一套易用的交通环境测试平台。

代码量大约

### 3.3.2 软件平台架构

3.3.2算法实现

由于交通环境模型的建立是为了最大化路口吞吐量，缓解交通阻塞。出于这样的目的，去掉了相位设置的不同分类，采用较为通用的8相位设计，使得智能体决策空间较为自由。而元强化学习在应用过程中，在各个任务之间的切换过程时，动作空间的表示一致性，使得最小化多任务之间的时序差分误差时，神经网络的设计和表达更合理。

相位设置：

|  |  |  |
| --- | --- | --- |
| 原始相位 | 拓展相位-1 | 拓展相位-2 |
| ①⑦ | ①② | ①⑥ |
| ③⑤ | ③④ | ③② |
| ②⑧ | ⑤⑥ | ⑤⑧ |
| ④⑥ | ⑦⑧ | ⑦④ |





只考虑②③④⑤

相位设置：

|  |  |
| --- | --- |
| 原始相位 | 拓展相位-1 |
| ② | ②③ |
| ③⑤ | ③④ |
| ④ |  |

强化学习应用时，环境搭建和模拟是一个不可或缺的部分。

代码进行了大量的清洗和重构。

分层次的结构调用

参数书写的一致性（形参书写顺序的一致性）

遵循的编程基本规范

3.1.2 数据正则化处理

3.1.3 复权方法统计周期

3.2 模型构建方案

3.2.1 状态空间构建

3.2.2 动作空间构建

3.2.3 奖励函数构建

3.3 网络和参数选择

3.3.1 交易费用设置

3.3.2 网络模型构建

3.3.3 交易算法应用

3.4 本章小结

第四章 交易策略实现与验证

4.1 实验准备工作

4.1.1 编程工具选择

4.1.2 实验硬件参数

4.1.3 数据下载处理

4.2 算法实现对比

4.2.1

第五章 结束语

5.1 主要工作和创新点

5.2 后续研究工作

强化学习在1956年被首次提出后，在上世纪后期开始了第一次研究热潮，以Q-learning和Sarsa为经典的算法，能够解决一些实际的问题。当计算能力得到大幅提高，大数据科学提供数据样本，强化学习在2014年开始了新的研究高潮，代表的算法如PG算法和DQN算法，增强了强化学习在连续动作和空间情况下的表示和应用效果【强化学习原理与实践P9】。

SFFFFFFF

FFFFFFFF

FFFHFFFF

FFFFFHFF

FFFHFFFF

FHHFFFHF

FHFFHFHF

FFFHFFFG

Q学习和Sarsa都属于时序差分的学习方法。

强化学习和其它种类的机器学习方法不同，强化学习不具有任何先验知识，也不具有特征描述，而仅仅是通过奖励函数的设置，使得智能体与环境不断的交互过程中，提高智能体的获得奖励的能力。获得奖励的能力常常使用累积奖励来衡量。而在深度强化学习中，降低神经网络的损失也可以作为衡量一个标准。

**背景知识**

强化学习的要素包括智能体，智能体的策略，智能体的对价值的判断，智能体的在线学习与离线学习，智能体的环境接口。其中智能体的环境接口常常表示为马尔科夫过程，包括了SARP在内的环境要素。

强化学习的环境种类可以分为确定性的环境，该种环境没有任何变化的情况，如固定的走迷宫的环境，该类环境常常表现出完全的确定性。不确定的环境常常以一种概率分布的形式体现在环境对行为作出选择的状态下，如投掷硬币和骰子的结果。完全可观察的环境是指MDP环境中的状态表示是否是完全的信息，如在Atari游戏的环境是整个屏幕，则为完全的状态表示。而部分可观测的模型是指多人扑克时只知道自身的牌而无法获取对手的牌，又或者是在应用强化学习进行量化交易过程中，对环境的描述无法做到完全掌控和描述，而只能以已知的部分信息作为当前状态。离散和连续在强化学习环境表示上，具有行为和状态两种维度。另外也有单个智能体和多个智能体的区别。

环境也可以分为分幕式环境和保持性环境。

在以解决MDP环境问题的强化学习目标是在智能体与环境的不断交互下获取最大化的奖励和。在时刻的奖励可以表示为：



式子中表示在时刻下获取到的奖励，是当前和未来的所有奖励之和。而引入折扣因数主要是表示未来奖励和及时奖励的重要性。因为MDP模型中存在转移概率P，奖励概率R，导致了未来的不确定性增多，获得收益的风险增大，添加折扣因子则针对复杂多变的环境下，对未来期望的更少，而环境确定性更多，则可以适当增大折扣因子使得其影响减小。折扣因子用表示。因此时刻的累计收益表示为：



智能体的策略常常用表示，体现出智能体做出选择的决定。

强化学习可以利用估计值作为策略的决定方式。

强化学习的状态函数有和行为无关的，和与行为有关的表示。

如下为状态值函数表示：



上式表示了状态值函数为在当前状态下得到多次累积奖励的一个期望值。替换为每次的奖励值，状态值函数表示为：



状态行为值函数在状态值函数的基础上，将当前行为也作为状态的一部分，以表示在该状态下，执行该行为的价值。公式表示为



强化学习的目标为获得最大的值函数，以表示。





根据值函数的贝尔曼方程，



Q函数的贝尔曼方程



得到



上述方程称为贝尔曼最优方程

推导值函数和Q函数的贝尔曼方程：

定义为转移概率：



奖励概率为：





已知值函数可表示为：







Q学习更新公式：



强化学习的基本图示为：

环境

智能体

行为

奖励、

下一状态

模拟退火：



T永远大于0.

当为负，则表示能量降低，完全接受改变（探索）。P>1

当为正，则P在（0,1）之间，随着温度T降低，则概率P也降低，即接受改变（探索）的概率降低。P>random 则接受改变。因为T的降低，导致P无法大于random，则很难接受改变。

当最终，不接受改变，表示解的形成。

基于模拟退火策略的Sarsa强化学习算法， = Q\_new - Q\_old



将为正，则进行探索。P>1

当为负，则开始可能不探索。随着T的降低，要保证P探索概率降低。而T降低本身，导致了P降低，最终到0.当P>random很难成立，表示解的形成。

基于深度强化学习的股市超盘手模型研究，采用了DQN基础算法，结合了股市基本面指标和技术面指标共计37个特征作为状态，将买入和卖出限制在10手以内共计21个行为，在设计奖励的过程中，添加了短线操作失误的惩罚等，认为将DQN算法参与到量化交易中能够实现一定的收益回报。

熵的概念

相对熵、KL散度

*为真实分布，**为预测分布*

用于衡量分布的差异性。



信息熵，用于衡量系统的信息量

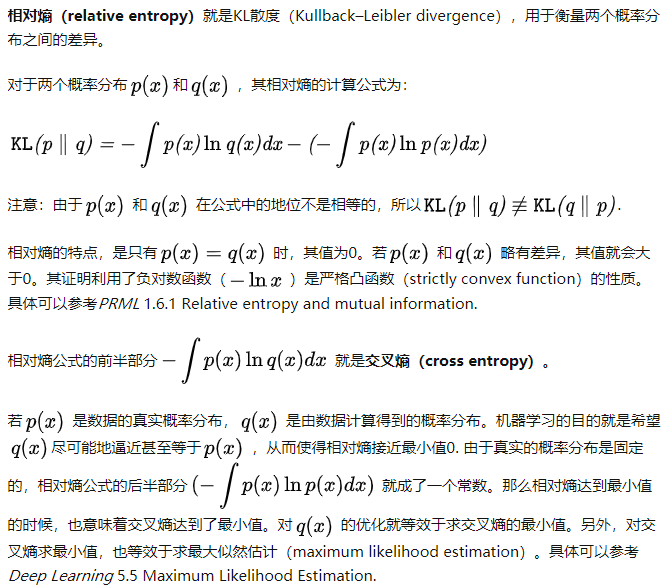


交叉熵，用于衡量分布的差异性



相对熵=交叉熵-信息熵

由于在机器学习学科中，需要解决的问题所服从的概率分布假设是确定的，因此使用相对熵和使用交叉熵作为度量方式上，在方向上都是正确的。



22,5