基于深度强化学习的城市交通灯控制方法研究

作者姓名 闫呈祥

学校导师姓名、职称 方 敏 教授

企业导师姓名、职称 刘毅明 高工

申请学位类别 工程硕士

学校代码 10701 学 号 1603121646
分 类 号 TP39 密 级 公开

西安电子科技大学

硕士学位论文

基于深度强化学习的城市交通灯控制方法研究

作者姓名：闫呈祥

邻　　域：计算机技术

学位类别：工程硕士

学校教师姓名、职称： 方　敏 教授

企业教师姓名、职称： 刘毅明 高工

学　　院：计算机科学与技术学院

提交日期：2019 年 6 月

Research on Urban Traffic Light Control Method Based on Deep Reinforcement Learning

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Computer Technology

By

Yan Cheng Xiang

Supervisor: Fang Min Title: Professor

Supervisor: Liu Yi Ming Title: Associate Research Fellow

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名：_____　　日　　期：_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在___年解密后适用本授权书。

本人签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘要

21 世纪以来全球人口数都在不断增加，城市人口增加的同时也严重影响到了城市交通的发展，智慧交通的发展和实现也成为了各国急需解决的问题。随着互联网和科技的发展，大数据、云计算、深度强化学习、人工智能等新的科技领域不断成为研究的热点和难点，智慧交通的研究也依据新科技迈向了自适应发展阶段。实现更加高效的城市交通灯控制算法也成为一项具有重要意义和研究价值的工作，本文对现有城市交通灯控制算法存在的不足之处优化，主要内容如下：

（1）首先对城市交通灯控制如何建模做了详细说明，然后针对单路口交通灯控制算法中存在存储 Q 表状态空间爆炸及未考虑历史策略对将来学习的影响等问题做出改进：利用近端策略框架考虑单路口历史动作相位的执行策略对当前时刻选择执行相位造成的影响，通过当前策略与历史策略的比值来优化改变当前路口环境学习率和深度学习采样，为了防止比值出现超域问题采用置信区间的方法，对其比值进行裁剪。通过实验在轻度交通流和重度交通流环境下验证算法的有效性。

（2）目前城市交通灯控制算法研究中虽有涉及到多路口协作的建模方案，但对于邻居之间如何具体协作没有反映出来，本文利用基于分布式深度 Q 网络来实现城市多路口协作控制建模。其中主要考虑了路口自身的历史状态、历史相位动作和该路口一跳邻居路口的上一时刻交通状态、执行相位动作对当前路口的影响，并将路口自身的历史状态-动作和一跳邻居的状态-动作依次通过 MLP 神经网络、LSTM 神经网络、MLP 神经网络计算最终当前路口当前时刻需执行的相位动作。通过实验在轻度交通流和重度交通流环境下验证算法的有效性。

（3）城市交通灯控制系统属于分布式系统，目前城市交通灯控制算法中针对分布式建立多路口多策略的算法较少，无法高效解决城市相关类型车辆优先行驶问题，本文利用分布式 W-Learning 实现交通灯分布式多路口多策略协同控制算法。其中主要通过计算路口本地策略和远程策略的 Q 值（执行相位）和 W 值（重要性权值）及 C 值（协作系数）来选择最大 W 值所对应的 Q 值执行，其中基于协作图原理在各路口之间并行互传状态、动作、奖励、W 等值，减少系统学习时间并了解邻居路口的交通状况。通过实验在轻度交通流和重度交通流环境下验证算法的有效性。

关 键 词：深度强化学习， 交通灯控制， 多路口协作， 分布式系统协作

ABSTRACT

Since the 21st century, the global population has been increasing. The increase of urban population has also seriously affected the development of urban transportation. The development and realization of smart transportation has become an urgent problem for all countries. With the development of the Internet and technology, new scientific and technological fields such as big data, cloud computing, deep reinforcement learning, and artificial intelligence have become the hotspots and difficulties of research. The research of intelligent transportation has also moved toward the adaptive development stage based on new technology. The realization of more efficient urban traffic light control algorithm has become an important and research value. This paper optimizes the existing deficiencies of urban traffic light control algorithms. The main contents are as follows:

(1) Firstly, how to model the urban traffic light control is described in detail, and then the improvement of the storage of the Q table state space explosion in the single intersection traffic light control algorithm and the impact of the historical strategy on future learning are improved: The near-end strategy framework considers the impact of the execution strategy of the single-action historical action phase on the current execution phase, and optimizes the current intersection environment learning rate and deep learning sampling by the ratio of the current strategy to the historical strategy, in order to prevent the ratio from appearing in the super-domain. The problem is solved by using a confidence interval method. The effectiveness of the algorithm was verified by experiments in mild traffic flow and heavy traffic flow environment.

(2)At present, although there are modeling schemes involving multi-junction cooperation in the study of urban traffic light control algorithms, it does not reflect how the specific cooperation between neighbors is, This paper uses the Distributed deep Q network to realize urban multi-road cooperative control modeling. The main consideration is the historical state of the intersection itself, the historical phase action, the traffic state of the previous moment of the one-hop neighbor intersection of the intersection, the influence of the phase action on the current intersection, and the historical state of the intersection itself - the action and the neighbor of the hop. The state-action sequentially calculates the phase action to be performed at the current moment of the current intersection through the MLP

neural network, the LSTM neural network, and the MLP neural network. The effectiveness of the algorithm was verified by experiments in mild traffic flow and heavy traffic flow environment.

(3) Urban traffic light control system belongs to distributed system. At present, there are few algorithms for distributed multi-junction multi-strategy in urban traffic light control algorithm, It is impossible to efficiently solve the problem of priority driving of vehicles of related types in cities, This paper uses distributed W-Learning to realize distributed multi-junction multi-strategy coordination of traffic lights. Control algorithm. The Q value execution corresponding to the maximum W value is selected mainly by calculating the Q value (execution phase) and the W value (importance weight) and the C value (coordination coefficient) of the intersection local strategy and the remote strategy, wherein the collaboration is based on the principle of cooperation diagram Parallel exchange of status, action, reward, W, etc. between the intersections, reducing system learning time and understanding the traffic conditions at neighboring intersections. The effectiveness of the algorithm was verified by experiments in mild traffic flow and heavy traffic flow environment.

Keywords: Deep reinforcement learning, Traffic light control, Multi-junction collaboration, Distributed system collaboration

插图索引

表格索引

符号对照表

符号	符号名称
i	路网中任意路口
$agent_i$	路口 i 的控制器
A_i	路网路口控制器 $agent_i$ 的相位集合
a_i	路网路口控制器 $agent_i$ 在某个时间步(cycle)执行的动作
L	路网中所有车道的集合
TL	城市路网中所有交通灯集合
tla	交通灯 tl 的动作集合
tl_i	道路 l_i 上的交通灯
s_{pl_i}	车辆所在车道交通灯、当前位置、目的地
id	交通灯相位编号
r	奖励值
b	单元格长度
c_1 和 c_2	比例系数
n_i	两个决策点之间通过的车辆数
n_0	单位时间内经过的平均车辆最大数量的一半
r_0	系统额外奖励
q_{i_t}	t 时刻第 i 个路口的四个不同方向中各自的排队长度
ρ	当前策略和历史策略的比值
$\pi_\theta(\cdot)$	当前策略
$\pi_{\theta old}(\cdot)$	历史策略
$clip(\cdot)$	裁剪函数
$L(\cdot)$	损失函数
D	记忆存储单元
$z(\cdot)$	MLP 神经网络计算函数
$h(\cdot)$	LSTM 神经网络计算函数
max	求最大操作
min	求最小操作
LP_i	本地策略
RP_i	远程策略
C	协作系数

缩略语对照表

缩略语	英文全称	中文对照
ATWT	Average Trip Waiting Time	车辆平均行驶等待时间
AJWT	Average Junction Waiting Time	路口平均等待时间
BP	Back Propagation neural network	BP 神经网络
CNN	Convolutional Neural Network	卷积神经网络
DDQN	Double Deep Q-learning	双 Q 学习
DDRQN	Deep Distributed Recurrent Q-Networks	分布式深度循环 Q 网络
DNN	Deep Neural Networks	深度神经网络
DQN	Deep Q-Learning Network	深度 Q 学习网络
DRL	Deep Reinforcement Learning	深度强化学习
DRQN	Deep Recurrent Q-Network	深度循环 Q 网络
DWL	Distributed W-Learning	分布式 W 学习
KNN	K-Nearest Neighbor	K 最近邻
LSTM	Long Short-Term Memory	长短期记忆网络
MDP	Markov Decision Process	马尔可夫决策
MLP	Multi-layer perceptron neural networks	多层感知器神经网络
PG	Policy Gradient	策略梯度
PPO	Proximal Policy Optimization	近端策略优化
RL	Reinforcement Learning	强化学习
SAE	Stacked Autoencoder	栈式自编码
SUMO	Simulation of Urban Mobility	城市交通模拟
TAV	Total Vehicles Arrived	到达目的地总车辆数
TCL	Traffic Light Control	交通灯控制

目录

第一章 绪论

1.1 研究背景与意义

随着世界经济和科技的发展，人口的发展也在快速增长，越来越多的普通家庭有能力去购买私人汽车，在解决个人出行方便的同时也严重增加了城市交通车流控制的压力。我国作为世界的人口大国，在促进经济发展的同时也使得更多的人流进入城市地区，使得城市道路交通供不应求，交通运行状态不断出现由顺畅到拥挤再到拥堵的逐渐恶化现象，不仅导致人们出行时间的浪费、交通管理时间和财务的大力消耗、二氧化碳的大量排放等，也严重影响了城市的可持续发展和国家大力推广的城市智慧交通计划。因此必须管理城市基础设施以有效的应对这样的增长，对任意城市交通而言有效缓解交通压力的途径主要有三种：（1）加快城市道路的建设；（2）加快城市地铁建设；（3）高效控制城市交通灯。其中途径 1 和途径 2 都需要大量的财力和有限的城市土地资源环境做支持，并且在建设初期会进一步加剧城市交通的拥堵，虽然在未来可以适当缓解城市交通压力，但是并不是快速解决交通拥堵的方法，所以高效的城市路口交通灯控制才是缓解交通压力更为快速有效的途径。

目前城市交通灯控制主要有三种途径[1-4]：（1）定时控制系统，每个交通灯控制设备已设置好每个相位执行的顺序和时长，不考虑车流的动态变化；（2）车辆驱动信号控制，通过路口安装的感应设备来动态决定某一相位执行时长的增加或是减少，相比较于定时控制系统已考虑车辆的动态流量；（3）交通灯自适应控制，通过当前路网路口车辆的交通状态动态自适应的管理调配应执行相位和时长[5]。在自适应交通灯控制系统中考虑机器学习和人工智能技术可进一步改善系统的有效性[6][7]。

强化学习和深度学习[8]是目前在机器学习和人工智能领域很热的研究方向，并在城市交通灯控制中取得了不错的效果[2][9]。强化学习和深度学习不需要提前了解城市道路环境和具体的车流状况，仅需要通过与城市道路环境的交互获取先验知识来学习形成一定的智能判别系统，在环境中获取状态后得到的奖励激励城市道路交通灯做出更优的相位动作。随着城市道路环境复杂度的变高导致状态-动作空间状态的指数型增长，利用深度学习有效的解决此类问题[10][11]，目前深度强化学习在围棋[12]和 Atari 2600 游戏[7]中都表现出了良好的效果，可见深度学习在人工智能方向和大数据推理逻辑方向具有很好的表现。自适应城市交通灯控制利用深度学习可进一步优化现有的单路口算法并与实际结合需要考虑多路口之间的协作控制，这将会是国内外在城市智慧交通方向研究的热点，我国智慧交通的发展同样需要深度学习等先进技术的支撑。

1.2 国内外研究现状

1.2.1 国外研究现状

利用强化学习解决城市交通灯控制已经在20世纪出现，文献[13-15]成功将SARSA应用于交通灯控制，这也是世界学术上第一次在交通灯控制算法中应用强化学习。文献[2]利用分布式多agents模型解决交通灯控制问题，每个agent都有独立的Q表学习并判断执行相位，实验证明了Q学习在交通灯控制算法中的有效性。文献[16][17]使用车辆交通状态建模的强化学习模型，通过计算路口在红绿灯的最大收益确定最优动作相位。文献[18]基于文献[16]的方法考虑了全局拥堵因子对路口的影响，导致状态转移概率和值函数的不同，并利用实验证明算法有效性。文献[19][20]考虑多agents协作，利用协作图建模，通过路口邻居之间的互相通信完成多路口之间的相位动作选择，利用实验证明算法有效性。

文献[21]使用深度堆叠自动编码器SAE解决强化学习中城市交通信号控制Q表空间状态爆炸问题；文献[22]利用基于值函数Q最大奖励值和基于策略梯度两种算法解决单路口交通灯控制问题。文献[23]利用协同Q学习算法，计算邻接路口红绿灯带来的收益值选择执行相位，属于利用强化Q-Learning实现多路口交通灯协同控制；文献[24]利用车辆排队长度与车速之间的关系，满足正态分布，预测周期内各相位执行时长和次序，与通常的强化学习建模不同并没有用到神经网络，主要利用正态分布解决路口相位问题。文献[25]利用多智能体协作原理，计算停在某一交通灯上的车辆数以及新到的车辆数及延迟时间选择执行动作；利用最小通信和[26]进行分布式强化学习的模型解决多路口交通灯控制，主要是将分布式系统划分为小系统，通过计算每个小系统的Q通信消耗，然后选择最小值Q，总体系统Q值表示为众多小系统最小Q的和，其中也是利用了协作图的原理。

通过考虑经验样本在时间差异上的误差并利用时间步长t纠正一阶偏置和二阶偏置，利用DQN和竞争架构的DQN解决单路口交通灯控制问题[27]。文献[28]利用分层架构建立交通灯控制模型，主要分为三层，最底层为城市路口，利用KNN算法计算邻居未访问状态的Q值，并通过时间差异函数与合格迹线函数计算单路口的Q值，选择执行最优策略即max策略值，然后通过层次之间的通信完成区域间路口通信，利用欧几里得公式计算各路口间的权重，最后形成城市路网的通信。文献[29-31]分别利用团队奖励协同多agents学习的价值分解网络、深度强化学习的价值自适应孤立路口交通灯控制、分布式分层多智能协作强化学习完成多路口交通灯的控制。利用城市模型中路口坐标位置建立坐标计算模型[32]、基于分布式协同强化学习的交通灯控制集成V2X网络的动态聚类算法[33]、基于协同多智能体框架的交通灯控制群体强化学习（粒子群）算法[34]、协作多策略强化学习算法[35]解决了城市多路口协同控制交通灯的问

题，并通过实验验证算法有效性。

1.2.2 国内研究现状

中国城市人口数的增长已经严重影响到城市交通的发展，如何高效的解决城市交通的拥堵问题及发展智慧城市交通已成为我国急需解决的问题，最近几年我国研究学者也在城市交通灯控制邻域取得了优异的成绩。

文献[36][37]利用强化学习建立状态学习空间、奖励函数、最小路口排队数量的自适应单路口交通灯控制算法，实验表明优于定时控制系统和感应设备控制系统。文献[38]利用 BP 神经网络和 Q-Learning 结合实现单路口交通灯控制，实验证明优于定时控制系统。文献[39]将 agent 概念与路口相结合，利用 Q-Learning 实时更新交通状态，减少车辆排队时长，通过单路口证明算法的有效性。文献[40]将多步强化学习 SARSA 应用于单路口交通灯控制，根据实时路口交通状态决策执行相位，实验表明优于定时控制系统。

文献[41]中利用启发式强化学习计算路口红绿灯收益来获取下一时刻单路口执行相位，并利用协作图和多 agents 协作的概念，考虑多路口之间交通灯的协作调配，实验表明利用强化学习的多 agents 协作算法优于定时系统和 max-plus 等算法。文献[42]考虑了交通突发事件对于交通灯控制的影响、元胞自动机原理实现双排队模型、分布式信号控制策略等三方面实现对交通灯自适应控制并实验证明了算法有效性。文献[43]利用强化学习和 agents 关系建立单路口和多路口控制算法，分别利用分布式 Nash-Q 算法、策略梯度上升等算法改进城市路口交通灯控制算法，实验证明算法有效性。文献[44]利用深度强化学习 DQN 解决单路口控制算法以及利用协作图和多 agents 之间的关系建立多 agents 迁移学习算法实现多路口协作控制，实验证明算法有效性。文献[45]利用红绿灯绿性比和车流比例来实现固定周期内各相位执行时长；文献[46]基于参数融合的 Q-Learning 解决单路口交通灯控制；文献[47]利用相位差和遗传粒子群算法优化单路口相位配时；文献[48]利用路口左转车道和右转车道的规则设置及城市空间布局对路口交通灯控制优化；文献[49]利用栈式自编码模型对交通状态进行估计、利用网络滤波协调非饱和交通状态、利用瓶颈区域协调信号算法解决饱和交通状态等问题。

1.3 本文主要工作和组织架构

城市路网中路口交通灯的有效控制是缓解城市交通压力的重要途径，针对目前已有算法做了以下改进：（1）利用近端策略 PPO 框架，考虑路口执行相位的当前策略和历史策略的比值影响当前的学习率和深度学习的采样，为防止出现超域问题，设置置信区间，优化单路口交通灯控制算法；（2）利用分布式深度循环 Q 网络 DDRQN

框架，考虑路口自身的历史状态、历史相位动作，一跳邻居路口上一时刻的交通状态、相位动作等利用 MLP 神经网络和 LSTM 神经网络计算当前时刻路口需要执行的相位，优化多路口交通灯控制算法；（3）利用分布式 W-Learning 框架，考虑路口本地策略和远程策略对路口执行相位重要性评判，计算重要性 W 值和协作系数 C 值，多路口间互相传递数据加快系统的学习，最终选择最大 W 值所对应的 Q 值相位执行，优化多路口多策略交通灯控制算法。

本文共有六章节，各章节的主要内容概述如下：

第一章：介绍了论文研究背景与意义，以及城市交通灯控制在国内外研究现状分析，同时简述了论文的研究内容与结构。

第二章：介绍了深度强化学习的背景知识，主要包含强化学习与马尔可夫问题、深度学习与深度 Q 学习、策略梯度、卷积神经网络、分布式与强化学习等知识。

第三章：首先叙述了城市交通灯控制问题及算法优越性的评价指标，然后介绍了城市交通灯控制建模方式，并详细说明了解决单路口交通灯控制的近端策略 PPO 控制算法。

第四章：首先介绍了多 agents 系统的协作，然后详细介绍了分布式深度循环 Q 网络及基于分布式深度循环 Q 网络 DDRQN 框架的城市交通灯建模方法，优化多路口协作交通灯控制算法。

第五章：首先详细讲解了分布式 W-Learning 如何体现多 agents 之间的协作及 DWL 算法应用于交通灯建模，然后详细讲解了多路口间传递数据的注意事项等，并解释了基于 DWL 建立的交通灯控制模型，优化多路口多策略协作交通灯控制算法。

第六章：总结本文的研究工作，对本文改进三个算法的优点以及未来工作总结分析。

第二章 深度强化学习背景知识

本章主要是对城市交通建模问题中涉及到的深度强化学习知识进行介绍，主要包括了强化学习与马尔可夫问题、深度学习与深度 Q 学习、策略梯度、卷积神经网络、分布式与深度强化学习。

2.1 强化学习与马尔可夫问题

强化学习（奖励学习、评价学习）是指 agent 通过不断的试学习和环境之间的交互来获取奖励值，依据奖励值来指导 agent 的动作。每一时间步 agent 都会与环境交互并获取奖励直到 agent 到达最终状态或者是 agent 满足了条件，模型如图 2.1 所示。

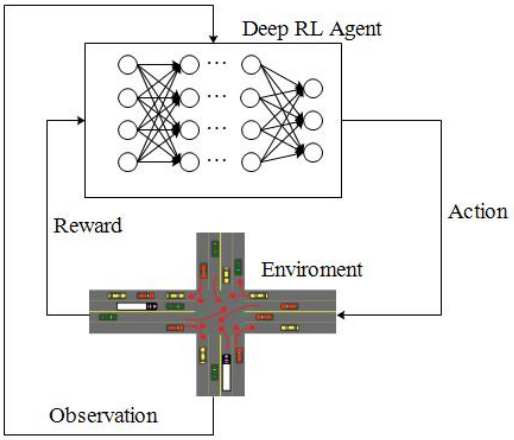

图 2.1 强化学习模型

通常会将强化学习认为是一种马尔可夫决策问题(MDP)，马尔可夫针对随机问题，是未来发展的概率问题。决策者通过周期的或者是连续的观察随机系统并惯性的作出策略，MDP 定义为五元组$<S,A,P,R,\gamma>$，其中 S 是环境状态空间的状态集；A 是代理为了与环境交互可以按顺序使用的动作空间中的一组动作；P 是转移概率，状态转移函数 T 为：$S\times A\to\prod(S)$；R 为即时奖励函数，指系统采用动作 A 从状态S转移为S'所获得的奖赏值；γ为折扣因子，它模拟了未来的重要性并立即奖励，$\gamma\in[0,1]$。在每个时间步 t，代理人通过环境感知状态$s_t\in S$，并根据其观察选择执行动作 A。采取行动后导致环境状态转变到关于转变函数 T 的下一个状态$s_{t+1}\in S$，代理接受由奖励函数 R 确定的奖励r_t。

强化学习中代理的目标是学习最优策略$\pi:S\times A\to[0,1]$，其定义为在状态s_t中选择动作的概率，以便遵循基础策略预期的累积折扣奖励随着时间的推移达到最大化。

时间 t 体现未来奖励 R_t 定义如下：

$$R_t = E\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\right] \tag{2-1}$$

在大多数现实问题中，有很多状态和动作无法应用经典的强化学习，但是在交通灯控制优化问题中，状态空间是连续的，可采用 RL 技术来解决。RL 技术可以使用不同形式的函数逼近器。例如：线性函数近似，状态和动作空间 f 的特征线性组合及学习的权重 w($\sum_i f_i w$)，非线性函数近似（神经网络）。直到最近，RL 的大部分工作都应用线性函数逼近器，最近深度神经网络(DNN)如 CNN 或 RNN、SAE 等也被普遍用于大型 RL 任务。

2.2 深度学习与深度 Q 学习

深度学习是机器学习中一个新的研究领域，主要是通过建立类似于人类可学习分析的神经网络来解释大数据问题，例如：文本、图像、语音、视频等多种类型数据。深度学习是一种无监督学习，通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示，相对于其他机器学习的方法，深度学习的主要优势是可以用更多的数据或更好的算法来提高学习算法的效率和结果的准确率。目前深度学习应用最广的就是在计算机视觉、语音识别、自然语言处理、人工智能 AI 等多种领域。

深度学习技术是解决高维数据并从数据中提取判别信息的最佳解决方案之一。深度学习算法具有从数据中自动提取特征（提取表示）的能力。通过直接馈入深网的数据来学习表示而不使用人类知识（即自动特征提取）。深度学习模型包含多层表示。实际上，它是一堆构建模块，如自动编码器，Restricted Boltzmann 机器和卷积层，在训练期间，原始数据被馈送到由多个层组成的网络中。作为非线性特征变换的每个层的输出用作 DNN(Deep Neural Networks)的下一层的输入，最终层的输出表示可以用于构造分类器或那些可以具有更好的效率和性能的应用程序，其中数据的抽象表示以分层方式作为输入，在其输入的每一层上应用非线性变换以试图学习并提取潜在的解释因子，因此，此过程学习抽象表示的层次结构。

DNN 的主要优点之一是能够从原始输入数据中自动提取特征。深度学习网络(DQN)[50]利用深度学习的这种优势，将代理人的观察结果表示为学习最优控制策略的抽象表示。DQN(Deep Q-Network)通过将 DNN 函数逼近器与 Q-Learning 聚合来学习动作值函数，并结合策略 π，告诉代理应该为每个输入状态选择什么动作。在高维连续状态和动作空间中应用非线性函数逼近器（如神经网络和无模型 RL 算法）会产生

一些收敛问题。这些问题的原因是：（1）RL(Reinforcement Learning)任务中的连续状态具有相关性。（2）由于 Q 值略有变化，代理商的基本策略也会发生变化。为了解决这些问题，DQN 提供了一些可以显著提高算法性能的解决方案。对于相关状态的问题，DQN 使用先前提出的经验重放方法[50]。这样，在每个时间步，DQN 将代理的经验 (s_t, a_t, r_t, s_{t+1}) 存储到数据集 D 中，其中 s_t 、 a_t 、 r_t 分别表示状态、选择的动作和接收的奖励，s_{t+1} 是下一时间步的状态。为了更新网络，DQN 利用随机小批量更新，在训练时从经验重放存储器（先前观察到的转换）中均匀随机采样，这否定了连续样本之间的强相关性，处理上述收敛问题的另一种方法是 Policy Gradient(PG)方法，在 2.3 中会详细说明 PG 方法。

如图 2.2 所示为 DQN 的学习模型，图 2.3 为 DQN 的训练流程。

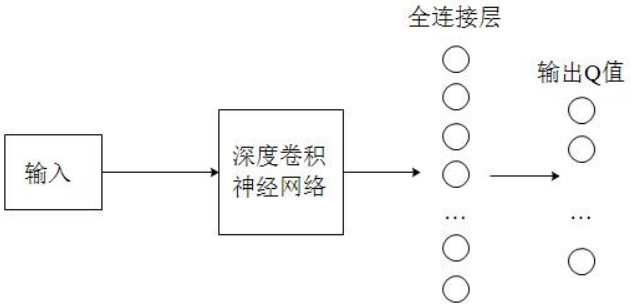

图 2.2 DQN 学习模型

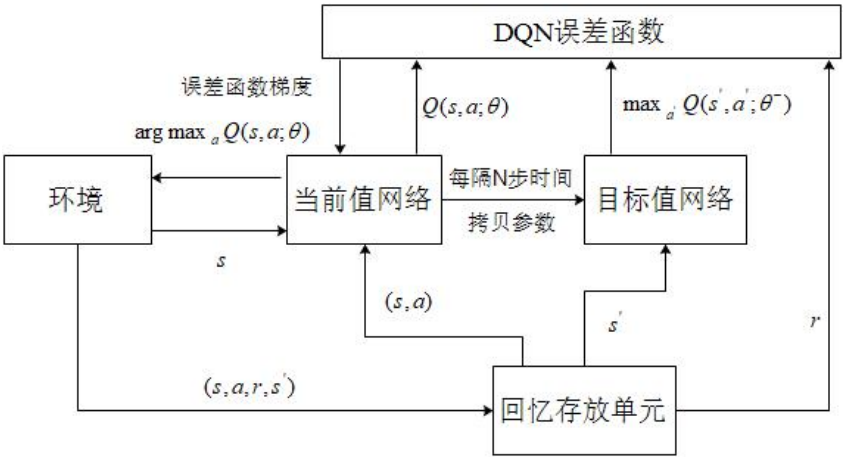

图 2.3 DQN 训练流程

DQN 在游戏和基于视觉感知的场景中有很强的适用性和通用性，但是在 DQN 使用 $Y_t = r + \gamma \max_{a'} Q(s', a' \mid \theta_t^-)$（式中 θ_t^- 表示目标网络参数）近似表示值函数的优化目标时，每次选择最大 Q 值执行动作会导致学习过程中出现过高估计 Q 值问题。在文献[51]中提出了基于双 Q 学习 DDQN(double Q-learning)采用不同的参数 θ 和 θ^-，其中 θ（当前值网络参数）用来选择最大 Q 值对应的动作，θ^-（目标值网络参数）用来评

价最优动作的 Q 值。为选出下一状态的最优动作 a_{t+1}^{*}，构造目标 $Q(s_{t+1},a_{t+1}^{*};\theta_t)$。目标 Q 的公式为(2-2)：

$$Y_t^{DDQN}=r_{t+1}+\gamma Q(s_{t+1},a^{*};\theta_t^{-}) \tag{2-2}$$

其中 $a^{*}=\arg\max_a Q(s_{t+1},a\,|\,\theta_t)$，同样在文献[50]中将 DDQN 分为了基于优势学习的深度 Q 网络和基于优先级采样的深度 Q 网络。

对 DQN 模型改进一般是通过向原有的网络中添加新的模块来实现，在文献[50]中总结了两种 DQN 的改进模型，分别为基于竞争架构的 DQN(dueling network)[51]和深度循环 Q 网络(Deep Recurrent Q-Network，DRQN)[52]。基于竞争架构的 DQN 是将 CNN(Convolutional Neural Network)提取出来的抽象特征分为状态值函数和动作优势函数两种。其 Q 值函数为公式(2-3)，竞争架构 DQN 的模型为图 2.4：

$$Q(s,a\,|\,\theta,\alpha,\beta)=\hat{V}(s\,|\,\theta,\beta)+\hat{A}(s,a\,|\,\theta,\alpha) \tag{2-3}$$

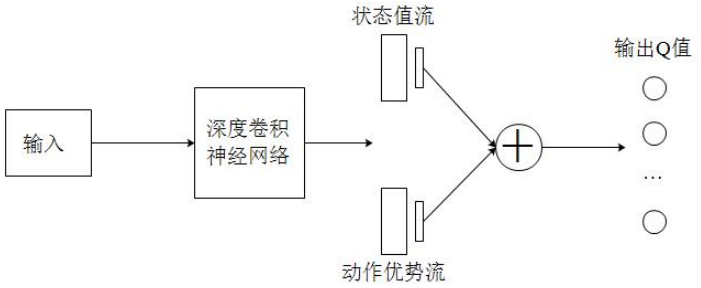

图 2.4　基于竞争架构的 DQN 模型

其中 $\hat{V}(s,|\,\theta,\beta)$ 为状态值函数，$\hat{A}(s,a\,|\,\theta,\alpha)$ 为动作优势流函数，α、β 和 θ 为状态值流、动作优势流和模型剩余部件的参数。

深度循环 Q 网络是通过利用循环神经网络结构来记忆时间轴上连续历史状态信息，对于状态信息部分可观察问题有效的减缓了网络的计算和存储负担。在交通灯控制问题中路口的交通状态是可观察并连续的，因此在论文第四章利用了改进的 DRQN 模型并用到了 LSTM 神经网络，图 2.5 为深度循环 Q 网络模型。

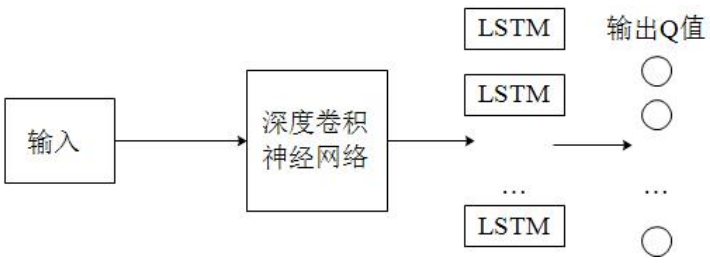

图 2.5　深度循环 Q 网络模型 DRQN

随着深度学习技术研究的深入，还会有更优的模型出现，例如 Open AI 团队提出的 PPO(Proximal Policy Optimization) 模型； Google DeepMind 团队提出的 DDRQN(Deep Distributed Recurrent Q-Networks) 模型、DDPG(Deep Deterministic Policy Gradient)模型、DPPO(Distributed Proximal Policy Optimization)模型，在论文的第三章和第四章分别利用了 PPO 模型和 DDRQN 模型实现交通灯控制算法。

2.3 策略梯度

策略梯度 Policy Gradient(PG)[50]是另外一种不同于 Q-Learning 的强化学习。PG 的输入与强化学习相同都是状态，输出则是该状态采取每个动作的概率。如何判断动作的优劣则是通过获得的奖励，如果某个动作获得的奖励多就增加该动作的概率，相反如果某个动作获得的奖励少就减少该动作的概率。

首先将策略 π 参数化（利用 softmax 函数），即公式(2-4)

$$\pi(a \mid s;\theta) = \frac{e^{\phi(s,a)^T \theta}}{\sum_b e^{\phi(s,b)^T \theta}} \tag{2-4}$$

其中，$\pi(a \mid s;\theta)$ 表示在状态 s 情况下执行动作 a 的概率，θ 为策略 π 性能调整梯度参数。$\phi(s,a)$ 表示关于状态 s 和动作 a 的特征向量。而 PG 目标就是通过调整 θ 让 agent 在策略 $\pi(a \mid s;\theta)$ 的前提下其状态 s 和动作 a 的 $Q^{\pi_\theta}(a,s;\theta)$ 取得最大值。对于梯度 θ 可采用梯度上升方法，即公式(2-5)：

$$\theta \leftarrow \theta + \alpha R \nabla_\theta \sum_{t=0}^{T-1} \log \pi(a_t \mid s_t;\theta) \tag{2-5}$$

其中 α 为学习率，R 为奖励值。所以 PG 是通过改变不同动作概率不断逼近最优策略，获得最大 Q 值。随着时间步的迭代，agent 执行最优动作的概率会不断逼近 1。

策略梯度可以采用梯度上升方法，针对不同问题也可以采用梯度下降方法。实际上，PG 可以搜索策略空间以直接学习策略，而不是估计状态值或动作函数。与传统的 RL 算法不同的是 PG 方法不会遇到在非线性函数逼近下或可能是 POMDP 环境中估计值函数的收敛问题。PG 还可以比纯粹基于价值的方法更好地处理连续状态和动作空间的复杂性[50]。PG 方法使用 Policy Gradients 的蒙特卡罗估计来估计 Policy Gradient，保证这些方法收敛到其参数化策略函数的局部最优值。但是，通常 PG 方法导致其梯度估计的高度变化，因此，为了减少梯度估计器的方差，一些方法从策略梯度中减去基线函数，基线函数可以用不同的方式计算[50]，本章描述一种常见的策略梯度 g 和基线的计算公式(2-6)：

$$g=\nabla_{\theta}\sum_{t=0}^{T-1}\log\pi(a_t\mid s_t;\theta)(R-b) \tag{2-6}$$

式中 R 表示总奖励值，b 是一个与当前轨迹 τ 相关的基线，为减少 R 的方差，通常设置为 R 的一个期望估计。从公式(2-6)中可以看出 R 与 b 的差值越大，对应轨迹 τ 被选中的概率越大。所以在大规模状态的 DRL 任务中，可以将策略通过深度神经网络参数化表示，并采用传统的策略梯度方法来求解最优策略。

本小节认为 PG 是朝着好的方向选择，因此可以通过优势函数来构造策略梯度：

$$g=\nabla_{\theta}\sum_{t=0}^{T-1}\hat{A}_t\log\pi(a_t\mid s_t;\theta) \tag{2-7}$$

$\hat{A}_t$ 表示状态动作对 (s_t,a_t) 优势函数的估计值。通常 $\hat{A}_t$ 可以通过公式(2-8)表示：

$$\hat{A}_t^{\gamma}=r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots-V(s_t) \tag{2-8}$$

$V(s_t)$ 类似于公式(2-6)中的基线 b。当 $\hat{A}_t$ >0 时会增加对应动作选择的概率，当 $\hat{A}_t$ <0 时会减少对应动作的选择概率。

在公式(2-8)中可以选择需要执行的时间步长，可以是一步、两步、多步。若为两步则为公式(2-9)，其他时间步长公式可以此类推：

$$\hat{A}_t^{\gamma}=r_t+\gamma r_{t+1}+\gamma^2 V(s_{t+2})-V(s_t) \tag{2-9}$$

尽管使用了优势函数还是会出现一定的估计偏差，为了缩小方差的同时还能保证偏差较小可以采用公式(2-10)：

$$\hat{A}_t^{\gamma}=\delta_t+(\gamma\lambda)\delta_{t+1}+(\gamma\lambda)^2\delta_{t+2}+\cdots+(\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{2-10}$$

$$\delta_t=r_t+\gamma V(s_{t+1})-V(s_t) \tag{2-11}$$

其中 λ 是一个调节因子，$\lambda\in(0,1)$，当 λ 接近 0 时，$\hat{A}_t^{\gamma}$ 是低方差高误差的；当 λ 接近 1 时，$\hat{A}_t^{\gamma}$ 是高方差低误差的。本论文在第四章会利用梯度策略更新共享网络参数和目标网络参数。

2.4 卷积神经网络

卷积神经网络(Convolutional Neural Network，CNN)[53]是依据生物的视知觉原理

建立的包含卷积计算且具有深度结构的前馈神经网络(Feedforward Neural Networks)，可进行监督学习和非监督学习，其隐含层内的卷积核参数共享和层间连接的稀疏性使得卷积神经网络能够以较小的计算量对格点化(grid-like topology)特征，比如在视频、音频、像素等数据上可进行稳定的学习并且处理的结果也是相当好。卷积神经网络结构主要包含了输入层、卷积层、池化层、全连接层、输出层。

（1）输出层：卷积神经网络的输入层可以处理多维数据，一维卷积神经网络的输入层接收一维或二维数组，其中一维数组通常为时间或频谱采样；二维数组可能包含多个通道，二维卷积神经网络的输入层接收二维或三维数组；三维卷积神经网络的输入层接收四维数组。对于不同的应用场景CNN的数据输入维数也是不同，由于CNN利用梯度策略学习，所以数据输入前都需要统一归一化处理。

（2）卷积层：该层主要功能是数据的特征提取，每层卷积层包含若干个卷积单元，每个卷积单元的参数都是通过反向传播算法最优化得到。卷积层中的神经元与前一层的多个神经元连接，连接区域由卷积核大小决定，定义为“感受野”。该层主要包含了卷积核大小、填充及步长，这三个超参数决定了卷积层输出特征图的大小。

参数共享机制：卷积层中的神经元类似于图像中的滤波器，每个神经元都只关注自己的特征和拥有固定的权重，所有的神经元组合起来就等同与整个图像的特征提取器集合。

（3）激励层：主要作用是非线性映射卷积层结果，通常采用函数 ReLU(The Rectified Linear Unit)，其主要特点为收敛速度快，求解梯度简单。如图 2.6 所示为激励层示意图，不过通常将激励层规划到卷积层里。

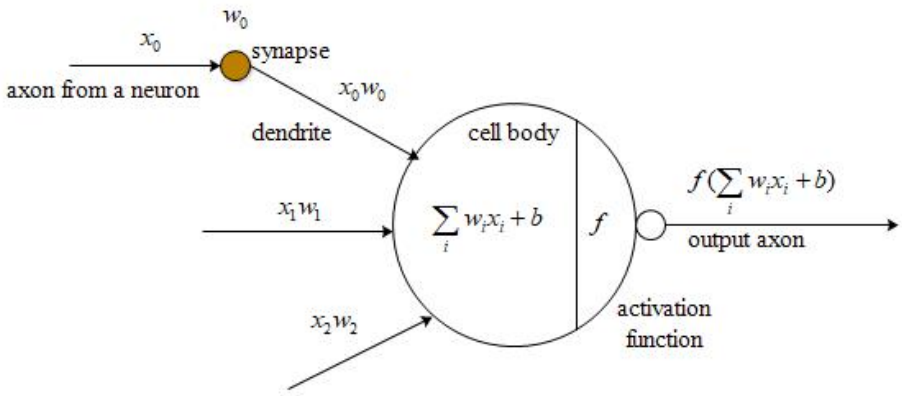

图 2.6 激励层示意图

（4）池化层：该层主要作用是对卷积层输出的特征图进行特征选择及信息过滤，利用池化函数可将特征图中单点的结果转换成其相邻区域的特征图统计量。常见的有Lp 池化、随机/混合池化、谱池化等。Lp 池化一般表现形式为公式(2-12)：Lp 池化中

包含有极大池化和均值池化。

$$A_k^l(i,j)=\left[\sum_{x=1}^{f}\sum_{y=1}^{f}A_k^l(s_0 i+x,s_0 j+y)^p\right]^{\frac{1}{p}} \tag{2-12}$$

其中 $A_k^l(i,j)$ 表示像素 (i,j) 的池化操作，s_0 表示步长。i、j、x、y 均为像素，f 为卷积核大小。p 是预指定参数，当 $p=1$ 时，Lp 池化在池化区域内取均值；当 $p\to\infty$ 时，Lp 池化在池化区域内取极大值。k 为特征图的通道数，l 为通道数最大值。混合/随机池化一般表示形式为公式(2-13)：

$$A_k^l=\lambda L_1(A_k^l)+L_\infty(A_k^l),\lambda\in[0,1] \tag{2-13}$$

式中 $L_1(A_k^l)$ 表示均值池化，$L_\infty(A_k^l)$ 表示极大值池化。

（5）全连接层：这一层类似于前馈神经网络中的隐含层，是卷积神经网络的最后一层并只向其它全连接层传递信号。

（6）输出层：卷积神经网络中输出层的上层通常是全连接层，对于不同应用场景输入的数据不同，输出层的输出数据也会不同。例如：图像分类问题，输出层使用逻辑函数或归一化指数函数(Softmax Function)输出分类标签；物体识别问题，输出层可设计为输出物体的中心坐标、大小和分类；图像语义分割问题，输出层直接输出每个像素的分类结果。

在论文第三章和第五章中使用 CNN 处理数据，第四章中使用 RNN(Recurrent Neural Networks)处理数据，本节不再赘述 RNN，详见可参考文献[54]。

2.5 分布式与深度强化学习

在现实中的很多场景面对多 agents 的情况下，采用单一的深度强化学习虽然在一定程度上优于一般的机器学习算法和其他的非机器学习算法，但是若要取得更好的效果就需要考虑多 agents 之间的协作和分布式的架构。比如在城市路网中的路口就属于类似现实场景。当深度学习模型的训练数据达到数千万级别时，若不做任何剪枝处理，训练参数可能也会达到数百亿级别，为了解决这类问题可以利用多个节点，分布式高效地训练处神经网络，所以分布式可以很大程度的提高深度学习的训练效率。图 2.7 为分布式深度学习框架图。

Google DeepMind 团队在深度强化学习的研究中很突出，DeepMind 团队针对分布式场景提出了如下的分布式强化学习框架：A3C(Asynchronous Methods for Deep Reinforcement Learning) 、 Rainbow(Rainbow: Combining improvements in deep

reinforcement learning)、APE-X(Distributed Prioritized Experience Replay)、Rudder(Return Decomposition for Delayed Rewards)、OpenAI 团队的 PPO(Proximal Policy Optimization Algorithms)等，这些架构在解决不同的实际应用场景中有不同的优势，针对本文研究的交通信号控制场景在论文的第三章和第五章使用 PPO 模型，并且在第五章考虑多 agents 的协作问题，PPO 模型在第三章中详述，本节不再赘述，以下主要简述上述其他几种分布式强化学习框架。

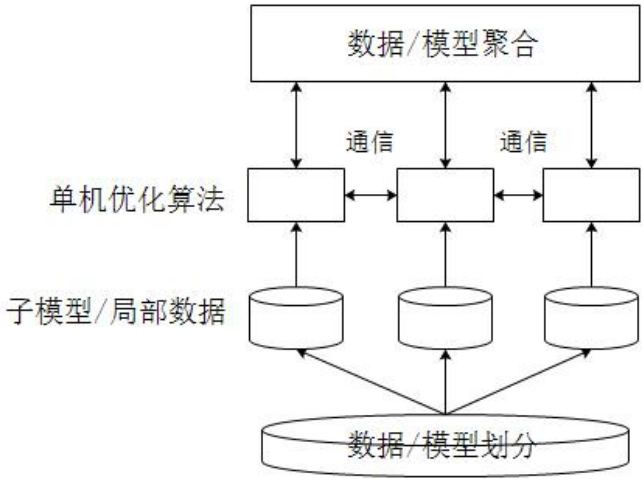

图 2.7 分布式深度学习框架

A3C[55]不仅适用于离散同样也适应于连续动作空间问题，通过创建多个 agents，在多个环境实例中并行且异步的执行与学习，然而 A3C 并不过多的依赖 GPU 或是大型分布式系统，可以运行在一个多核 CPU 上。A3C 可以异步执行多个 agents 的动作探索，通过并行 agents 经历的不同状态减少了数据训练过程中的相关性，不利用 DQN 的经验回放机制也能得到稳定的学习效果，可用于解决 n-step methods、连续性动作控制、actor-critic 等问题。A3C 网络结构使用 CNN，其中一个 softmax output 作为 policy，即 $\pi(a_t|s_t;\theta)$；另一个 linear output 为 value function，即 $V(s_t;\theta_v)$，其余 layer 都是共享。

Rainbow[56]是 DeepMind 整合 DQN 算法中的六种变体：（1）Double DQN(DDQN)通过解耦选择和引导行动解决 Q-Learning 过度估计偏差问题；（2）优先经验回放，通过重放学习到更加频繁的转换，加快学习速率；（3）竞争 DQN 模型，通过状态值函数和动作值函数来评价动作优劣；（4）多步学习，通过权衡偏差-方差，将新得到的奖励讯速传递给旧状态；（5）分布式 Q-Learning，学习折扣回报的分类分布（代替估计平均值）；（6）Noisy DQN，使用随机网络层进行探索。上述六种都是基于同一框架解决不同应用问题或是提升 DQN 某一方面的性能。

APE-X[57]，在分布式中使用多个 actor，分布式中每个 actor 都不完全一样，因此

$\varepsilon-greedy$ 策略中的 ε 也会不同，可以更好的做探索、更全面的搜索优先级最高的 Replay 来训练。

如图 2.8 所示：（1）多个 actor 在各自的环境中产生经验并写入同一个共享经验重放记忆池，同时还可以计算 initial priorities。（2）一个 learner，从记忆池中选择样本，然后更新记忆池中的参数，并更新网络参数。（3）每个 actor 网络从 learner 定期获取最新参数。

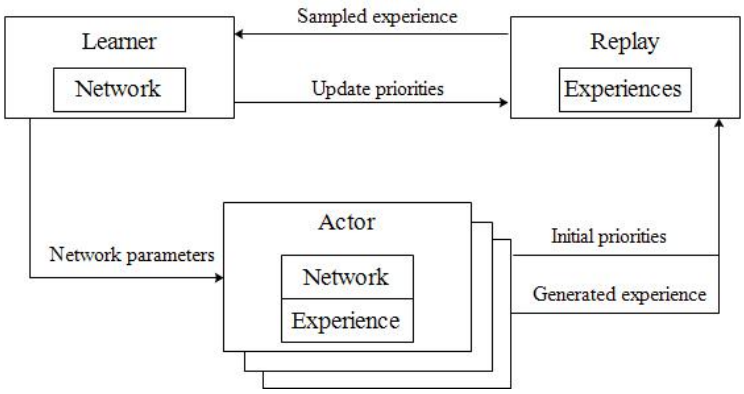

图 2.8　APE-X 框架

Rudder[58]主要是利用基于返回值分解的方法解决强化学习中随着延迟步数的增加，对时间差分(TD)估计偏差的纠正时间的指数级增长，和蒙特卡洛(MC)估计方差的指数级增长问题。

本论文在第五章采用分布式与深度强化学习结构的算法来实现路网中多 agents 之间的协作，其中分布式部分采用 DWL(Distributed W-Learning)来解决，实现多 agents 之间的通信协作；深度强化学习部分采用 PPO 模型，DWL 详解可参见第五章。

2.6 本章总结

本章主要是对深度强化学习的背景知识进行了分析总结，其中 2.1 主要分析了强化学习与马尔可夫问题；2.2 主要分析了深度学习与深度 Q 学习；2.3 主要分析了策略梯度；2.4 主要分析了卷积神经网络；2.5 主要分析了分布式与强化学习。在论文的第三章、第四章、第五章中都是利用该章节中的知识建立交通灯控制模型，为论文算法研究提供了理论依据和保障了学术研究的可行性。

第三章　基于近端策略 PPO 的交通灯控制方法

3.1 引言

城市交通的畅通与否直接影响着人类的生活，日常所见的交通灯多数为定时控制系统。虽然当车流量较小时能有效的解决城市交通相位分时问题，但是当处于车流量的高峰期时定时控制系统就显得捉襟见肘，不仅无法有效的控制交通相位分时，而且严重的影响到人们的出行时间。目前国内外学者提出了很多城市交通灯控制算法，在一定程度上都优于定时控制系统，但是其中基于强化学习的算法需要不断的学习并存储 Q 表，在城市交通灯控制中会存在状态空间爆炸问题；基于深度强化学习的策略梯度优化算法并没有考虑到历史策略对将来学习影响的重要性和实际场景中单方向相位时长控制范围问题。基于上述问题，本章采用基于近端策略 PPO(Proximal Policy Optimization Algorithms)[59]的算法来优化交通灯控制。

3.2 交通灯控制问题描述

交通灯控制在缓解城市交通压力方面发挥着重要的作用，尤其是在交通出行高峰期，良好的交通灯控制策略更能明显的减少城市交通出行拥堵等情况的发生。交通灯控制目的是通过控制一定周期内或不定周期内不同相位的执行时长来解决路口车辆拥堵问题，其中无论是在一定周期内或是不定周期内，文献[23][24][44]等都提及需要设置单相位控制时长范围。因此可将城市交通问题数学量化来分析，通过数学策略来实现交通灯控制的优化问题。

3.2.1 路网中路口通用数据集合

本文在路网路口交通灯的状态选择中只考虑了绿灯和红灯，未考虑黄灯。在实际的城市路网交通中黄灯并不能够带来交通策略上的优化和收益，仅用来缓冲绿灯和红灯之间的动作替换，黄灯的设置一般都是固定的秒数并且黄灯和红灯、绿灯的时间动态变化上没有直接的影响。虽然在部分算法中涉及黄灯，计算不同相位在定周期内红、黄、绿灯的时序分配和时长分配，但本文与其建模方式不同，因此不考虑黄灯对红绿灯的影响。

在基于路网路口车辆交通状态建模中路网的每个路口都是一个 agent，路网车道上的每辆车都是一个 agent，每个路口 agent 和每辆车 agent 相互独立，两者之间没有关联。具体集合如下所示：

（1）i：表示路网中任意一个路口，则交通建模中代表路口 i 的控制器 $agent_i$。

在有 n 个路口的路网环境中，路口控制器的集合为：$I=\{agent_0,\cdots agent_i,\cdots agent_n\}$。图 3.1 所示路网中的一个路口（T 字路口同理）。

（2）A_i：表示路网路口控制器 $agent_i$ 的相位集合，即路口 i 的相位方案。

（3）a_i：表示路口 $agent_i$ 执行的相位动作，$a_i \in A_i$。路网中允许同一时间步内多个入口车道的交通灯为绿灯，与绿灯相冲突的入口车道为红灯。因此 a_i 为路网路口中可以同时满足绿灯的相位组合。图 3.1 路口 i 的动作与相位关系如表 3.1 所示。

（4）L：表示路网中所有车道的集合。L_i 表示路网中某个路口 i 的全部入口车道组成的集合，$L_i \in L$。l_i 表示路口 i 的某一入口车道，$l_i \in L_i$。

（5）TL：表示城市路网中全部交通灯的集合。TL_i：表示任意路口 i 的入口车道 L_i 对应的交通灯集合。tl_i 表示道路 l_i 上的交通灯。

（6）tla：表示交通灯 tl 的动作集合。交通灯的动作对应于交通灯的状态，因为本文中仅考虑红灯、绿灯，所以交通灯的动作集合为 $tla=\{tla_{red},tla_{green}\}$。

（7）s_{pl_i}：由三元组 $[tl_i,pos,des]$ 表示，包含车辆当前时间步所处的位置 pos、目的地 des、以及所处车道 l_i 上的交通灯 tl_i。

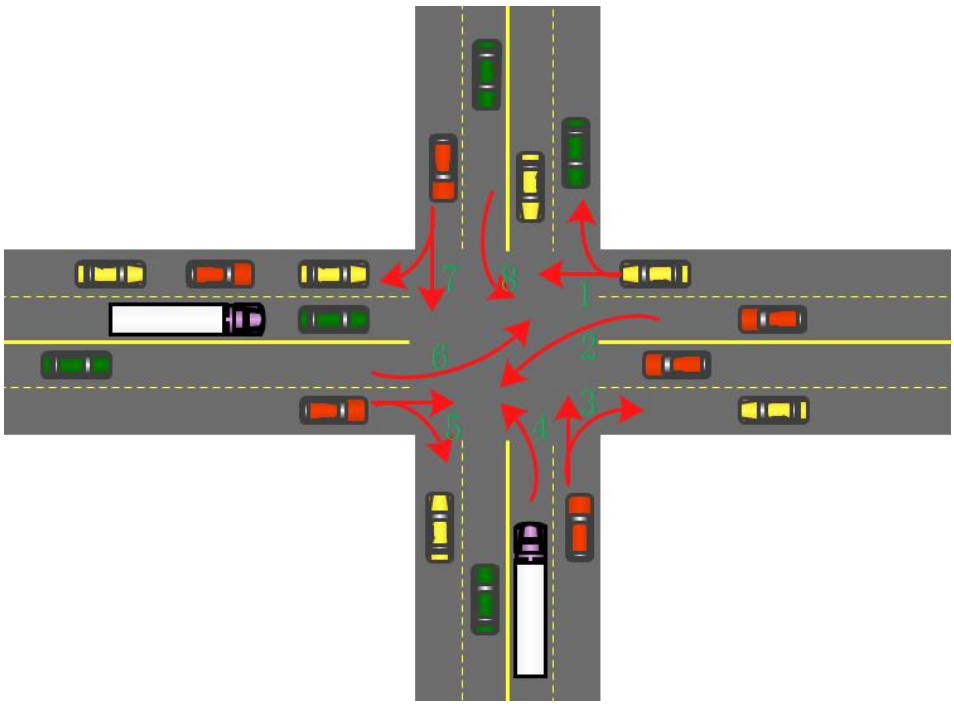

图 3.1　路网中的一个路口

图 3.1 表示了一个路口交通灯含有的八种相位动作，表 3.1 中将路口的动作和相位动作相匹配。上述为本文中通用的建模数据集合，交通灯控制的具体建模方法可详见 3.3 节。

表 3.1 路口动作与相位关系

路口动作（a_i）	相位（id）	
a_0	1	2
a_1	3	4
a_2	5	6
a_3	7	8
a_4	1	5
a_5	3	7
a_6	2	6
a_7	4	8

3.2.2 路网交通性能评价指标

考虑实际路网中判断交通是否拥堵的主要因素为车辆平均等待时间和车辆到达目的地数量等。在比较算法中含有非深度强化学习算法，为便于多个算法之间的比较，选择通用的比较指标，本论文中所有实验的比较将依据下述三个评价指标：

（1）AJWT(Average Junction Waiting Time)：路口平均等待时间。数值上等于通过路口的所有车辆总的等待时间除以通过的总车辆数得到的平均值，反应单个车辆在路口的平均等待时间。若车辆行驶方向与执行相位相同并通过，则该车辆路口等待时间为 0；并只有当路口车辆速度为 0 视为等待。AJWT 值越小，说明车辆路口等待时间越少，反之，车辆路口等待时间越长。

（2）ATWT(Average Trip Waiting Time)：车辆平均行驶等待时间。数值上等于所有到达目的车辆总的等待时间除以车辆总数得到的平均值，反应单个车辆行车全程所需的平均等待时间。ATWT 值越小，说明车辆行驶等待时间越少，反之，车辆行驶等待时间越长。

（3）TAV(Total Vehicles Arrived)：到达目的地的总车辆数。数值上等于已到目的地车辆总数。TAV 值越大，说明到达目的地车辆数越多，反之，到达目的地车辆数越少。

3.3 基于近端策略 PPO 的交通灯控制方法建模

本节通过描述路网交通状态集、路网路口动作集、路网路口动作奖励值将交通灯控制 TCL(Traffic Light Control)问题表示为 DRL(Deep Reinforcement Learning)任务。图 3.2 所示为采用卷积神经网络 CNN 处理 TCL 问题的通用框架：

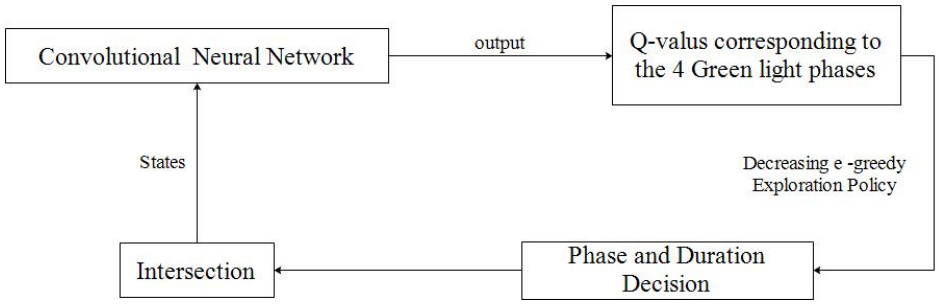

图 3.2 神经网络处理交通灯控制通用框架

路口车道上车辆状态作为 CNN 网络的输入，输出 Q 值表示路口需执行的相位动作，减小 $\varepsilon-greedy$ 策略中的 ε 值，确定相位执行时长。对于任何一个路口都可采用上图框架，采用 DRL 的优势在于可以通过实时交通状态来判断下一时刻执行相位使得路况达到最优，并且不需要像 RL 那样需要不断存储 Q 表而引起空间存储爆炸，下文将详细说明交通建模。

3.3.1 路网交通状态集

路网中交通状态是通过车辆的状态来表示的，因为在路网中可时刻变化的状态与车辆相关，可以体现出路网中交通状态的实时动态性。在上节中有说明路网的 agents 分为两种，一种是车辆，一种是路口，每个车辆是一个 agent，所以可以实时获取任意路口相连接的任意一条车道上的任意车的位置及车道上车的总数。在文献[22][60][61]中将任意路口直接相连的车道从停车线开始至长度为 Length（实验参数部分有设）的车道划分成若干个单元格，每个单元格长度为 b（实验参数部分有设），使得满足每个单元格仅可占据一辆车。用布尔变量来表示单元格内车辆存在与否，1 表示存在车辆、0 表示不存在车辆。

路网中任意路口相连接车道上的车辆都可用 0 和 1 表示，每时刻任意路口直接相连的某些车道上（执行相位的车道）的车辆都会运动，因此每时刻路口的交通状态都会不同。将图 3.3 转化为矩阵表示：

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

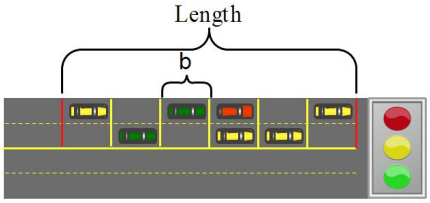

图 3.3 路网交通状态图

3.3.2 路网路口动作集

路网路口的动作集即采取不同动作的相位集，在表 3.1 中已经涉及了所有理论上可能的路口相位集，但是根据实际生活中的中国城市路口相位选择方案，本章节不考虑前四种路口相位集（前四种路口相位集仅开放一个路口方向的车道为绿灯，其他三个方向的车道为红灯，与实际的中国城市交通红绿灯方案不符合），只考虑表 3.2 中后四种动作集。

表 3.2 实际城市路口动作集

路口动作（a_i）	相位（id）	
a_4	1	5
a_5	3	7
a_6	2	6
a_7	4	8

在文献[45][47][48]认为交通灯的周期控制是基于轮播原理，在一定的周期时间内根据车流量来合理的分配每个不同相位所应持续的时间，进而缓解城市交通拥堵的压力。城市交通灯的控制虽然可以采用轮播原理，但是考虑到方法的适应性问题，针对不同的路口不能使用同样的周期时长，那么对于每个城市、每个路口来设计交通灯的轮播周期是件工作量很大的事。

本文对于红绿灯的执行相位时长方法如下：为避免交通灯的频繁切换导致交通混乱及成本代价的提升，设置交通灯任意相位执行时长最大值和最小值，依据实际生活中城市交通灯时长获取平均值，本文统一设置任意相位绿灯最大时长 70s，最小 10s。当选定路口某一相位动作时首先执行 10s 的时长，若下一执行相位不同则 10s 后更换执行相位；若超过 10s 后仍为该执行相位，则在该相位时长上累加 4s，直至执行相位更换。当执行某一相位时长达到 70s 时则不再执行改相位，直接替换为下一相位执行。

按照上述采用相位时长控制并没有涉及到固定的周期时长，相位执行时长完全取决于当时路网路口中车辆的状态即路网的实时交通流状态，并设置相位时长的最大值和最小值来客观的控制车辆在某一相位上的流量。

3.3.3 路网路口动作奖励值

路网作为执行方法的环境，需要在执行动作后对所采取的动作做优劣性评价，文献[22][44][60]等针对路口采用相位动作后所执行的奖励均有所不同，执行奖励涉及到等待时间、队列长度、车流量变化等。首先本章中涉及的底层路网状态是每辆车的实时位置信息，通过车辆状态的统计利用布尔变量转化为某一车道交通的状态，而且本章

算法中涉及的输入状态也就是路段的交通状态，针对路段交通状态变化最明显的两个特征就是在相位响应时间内通过的车流量和排队长度的变化，在执行结束某一相位后路口（路口是一种 agent，上文已经提及并且执行动作的 agent 就是路网的路口）的奖励为:

$$r=\begin{cases} c_1(n_i-t_i n_0)+c_2(\dfrac{\max q_{i_t}-\min q_{i_t}}{\max q_{i_{t-1}}-\min q_{i_{t-1}}})+r_0 & if\ max\ queuing\ length\ is\ reduced \\ c_1(n_i-t_i n_0)-c_2(\dfrac{\max q_{i_t}-\min q_{i_t}}{\max q_{i_{t-1}}-\min q_{i_{t-1}}})-r_0 & otherwise \end{cases} \tag{3-1}$$

其中 r 为路口采用相位动作后状态发生变化所带来的奖励，c_1和c_2为比例系数，c_1和c_2均设为 0.5[62]。n_i为在两个决策点（即选出执行的两个相位动作，可以相同可以不同）之间通过的车辆数，t_i是两个决策点之间的相位执行时间（即绿灯时间）；n_0为单位时间内经过的平均车辆最大数量的一半；r_0表示系统定义的额外奖励；q_{i_t}为在 t 时刻第i个路口的四个不同方向中各自的排队长度；$q_{i_{t-1}}$在 t 时刻第i个路口的四个不同方向中各自的排队长度。

通过式(3-1)反应出执行相位动作后带来的奖励主要和相位执行时间段内通过的车辆数及路口四个方向最大、最小排队长度的变化有关，可以更加明确得评判执行相位动作的优劣。

3.3.4 近端策略 PPO

近端策略 PPO(Proximal Policy Optimization Algorithms)[59]是一种用于强化学习新的政策梯度方法，它通过与环境的相互作用在采样数据之间交替，并使用随机梯度上升来优化“替代”目标函数。 鉴于标准策略梯度方法针对每个数据样本执行一次梯度更新，近端策略(PPO)提出了一种新的目标函数，其允许多个时段的小批量更新，具有信任区域策略优化(Trust Region Policy Optimization,TRPO)的一些优点，但是其实现更简单和通用。

近端策略考虑历史策略对未来状态的影响，本文采用近端策略的思想对深度学习的采样和对未来学习的折扣做评价，ρ表示当前策略和历史策略的比值，如公式(3-2)：

$$\rho=\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta old}(a_t\mid s_t)} \tag{3-2}$$

其中$\pi_\theta(a_t\mid s_t)$为当前采取的策略，$\pi_{\theta old}(a_t\mid s_t)$为历史策略，初始化$\rho=1$，本文采用$\varepsilon-greedy$策略探索。

但是在使用当前策略和历史策略做比值时会出现远远大于 1 的情况或者是无限接近 0 的情况，这样不仅对于未来的学习没有指导作用，可能会应为出现完全信任当前策略（即 $\rho \succ 1$，比值远远大于 1）或者是完全不信任当前策略（即 $\rho \approx 0$，比值无限接近 0）的情况。因此需要对 ρ 进行裁剪，使其在一定范围内影响未来动作的选择，主要是通过削减概率比来修改代理目标。对 ρ 裁剪如公式(3-3)所示：

$$\rho = [\min(\rho, clip(\rho, 1-\xi, 1+\xi))] \tag{3-3}$$

$clip(\rho, 1-\xi, 1+\xi)$ 表示对 ρ 的裁剪[59]，使得 ρ 的取值会固定在该区间，其中 ξ 为超参数。当 ρ 的值远大于 1 时，为避免过分完全的信任当前策略，对其裁剪取值为上限 $\rho = 1+\xi$；当 ρ 的值接近 0 时，为避免完全不信任当前策略，对其裁剪取值为下限 $\rho = 1-\xi$。最终的 ρ 会选择未裁剪和裁剪 ρ 中的最小值作为比例系数。包含 ρ 的损失函数更新公式为(3-4)：

$$L = \rho\alpha[r + \max_{a'} Q(s', a') - Q(s, a)]^2 \tag{3-4}$$

主要算法如下：

算法 1：基于近端策略 PPO 的交通灯控制算法

输入：

路网中路口交通状态转化得到的 01 矩阵，即状态 s

输出：

路网中路口执行相位动作，即动作 a

步骤：

Repeat:

观察路网交通状态 s 并利用 $\varepsilon - greedy$ 策略随机选择一个相位执行或者是选择最大 Q 值表示的相位执行；

执行相位动作，得到环境奖励值 r，观察路口交通新状态 s'；

$$y_i = \begin{cases} r_i & if\ s\ is\ terminal \\ r_i + \max_{a'} Q(s', a') & else \end{cases} ; \tag{3-5}$$

存储单元 D 存放 $< s, a, r, s' >$；

计算神经网络损失函数 $L = \rho\alpha[r + \max_{a'} Q(s', a') - Q(s, a)]^2$ 并更新网络；

路网交通状态由 s 转变为 s'；

end for

算法 1 的主要逻辑如上所示，除此之外需要按照路网状态描述方法初始化路网中路口控制器 agent 的状态矩阵，初始值为 0；初始化折扣因子 γ，$\varepsilon-greedy$ 策略中的 ε，agent 的即时奖励值 r=0，仿真时间步 t 的初始值 0，记忆存储单元为 D。

3.4 仿真实验与结果分析

本文实验通过 SUMO(Simulation of Urban Mobility)软件仿真实现，SUMO 是开源、时间离散和道路空间连续的交通仿真软件，本文使用的版本为 V0.32.0。SUMO 主要功能有创建路网、仿真、交通模拟等，其中图 3.4 是通过 OpenStreetMap 导出，利用 SUMO 命令及 JOSM(Java OpenStreetMap Editor)软件获取路口 ID 合成含有车流及红绿灯的实验路网图；使用 Traci(Traffic Control Interface) API 实现与 SUMO 的交互并获取实时交通状态及按照设计的算法逻辑控制交通灯和仿真数据的采集；为加快神经网络的训练安装 GPU 版本的 TensorFlow；安装集成 Python 环境的 Anaconda，算法逻辑与神经网络利用 Python 实现。

为了验证基于近端策略 PPO 建模的城市交通灯控制算法的性能，在 SUMO 仿真软件中，将 PPO 方法和 TC1 方法、TC-GAC 方法、TC-GAGC 方法、TC-HAQL 方法分别在图 3.4 所示的多个路口交通网络中仿真，并在轻度交通流和重度交通流下分别进行 5 次仿真实验。图 3.5 为图 3.4 中任意一个路口包含交通灯示意图。

在文献[22][44][60]等中针对单路口的路网环境大多数是仅包含一个路口，针对的算法虽然是单路口的交通控制但是却忽略了实际路网的复杂性，因此为了更好的反应算法的优越性，路网采用多个路口的复杂路网图，如图 3.4 所示。路网图中包含多个路口，因此产生车辆的起始位置和目的地也是大有不同，车辆运行线路也会复杂多样化，为了降低运行线路导致的路网车辆线路运行复杂问题，采用最短路径 Dijkstra 算法。上述仅为与实际路网更贴切设计的复杂路网环境，但算法仅针对单路口交通灯控制，不涉及多路口之间的协作。

图 3.4　多个路口的交通网络仿真图

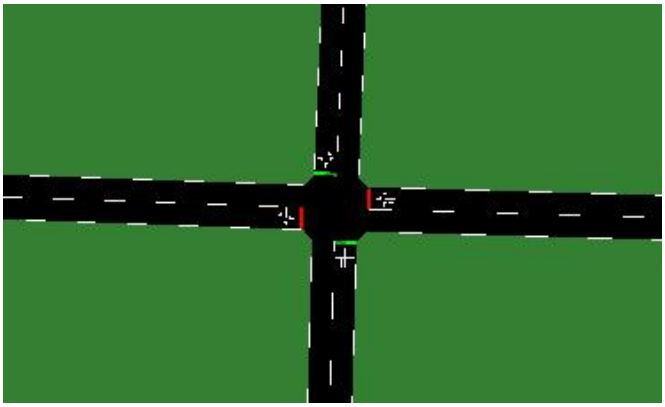

图 3.5 多路口图中的任意单路口示意图

3.4.1 路网环境配置

路口：在图 3.4 中含有 50 个路口，其中包含了十字路口、T 字路口、直角路口等，可根据实验需求自行设计拥有红绿灯的路口。图 3.4 中红色数字标注了 21 个有红绿灯的路口，基本都是主干道上的路口，其交通流相对较大，本章实验主要通过路口 2、3、4、11、17 五处路口做仿真数据采集（通过 JOSM 可获取图 3.4 中的路口 ID）。

车道：图 3.4 中的所有车道都如图 3.5 中所示，设置车道以 6m 为单位划分单元格，车道划分长度 180m，每个单元格同一时间内仅能存在一辆车。

车辆类型：在本文实验中只产生公交车 bus 和小汽车 car 这两种类型车，并不考虑行人、交通事故、自然灾害、雨雪天气等其他不确定性因素造成的影响。car 车长为 4m，bus 车长为 8m，其中 bus 占用两个车道单元格；车间距为最小 1.5m，起步速度为 5m/s，最大速度为 12m/s。

仿真时间：仿真时间设置为 50000 时间步(cycle)，每 5000 时间步对曲线绘制(X 轴)。Y 轴因为评价指标不同和车流量不同，按照实际各项获取数据分隔单位绘制。

车流量设置：路网的车流量设置需要一定的阈值，在文献[73]中说明车辆产生频率值是 0.4 时属于高度交通需求阈值，每条道路车辆产生频率指所有类型车辆的总和，本文中仅包含 car 和 bus 这两种车型，所以仅计算这两种车辆各自频率的和即可。当每条道路产生频率值大于等于 0.4 时交通灯控制方法都不可避免产生交通拥堵或堵塞，当小于 0.4 时则不会造成严重的交通拥堵。因此将每条道路车辆产生频率大于等于 0.4 定义为重度交通流环境，小于 0.4 定义为轻度交通流环境。具体车型产生频率在每章实验环境中都会声明具体值。

比较方法参数设置：TC1 方法、TC-GAC 方法、TC-GAGC 方法、TC-HAQL 方法中初始化参数：折扣因子 $\gamma = 0.9$，ε - *greedy* 策略中的 $\varepsilon = 0.01$。PPO 训练开始时 $\varepsilon = 1$，随后迭代递减至 0.01，折扣因子 $\gamma = 0.9$。

PPO 方法参数设置：

表 3.3　PPO 方法参数设置

参数	值
超参数 ξ	0.2
奖励参数 c_1	0.5
奖励参数 c_2	0.5
系统奖励 r_0	0.4
经验回放单元 D	60000
学习率 α	0.0025
批度大小	32

PPO 方法网络设置：使用 CNN 训练模型，具体网络设置见表 3.4：

表 3.4　PPO 方法网络设置

层	输入	过滤器大小	步长	过滤器数量	激活函数	输出
卷积层 1	84*84*4	8*8	4	32	ReLU	20*20*32
卷积层 2	20*20*32	4*4	2	64	ReLU	9*9*64
卷积层 3	9*9*64	3*3	1	64	ReLU	7*7*64
全连接层 4	7*7*64			512	ReLU	512
全连接层 5	512				Linear	4

3.4.2 轻度交通流

利用 SUMO 产生车辆 car 和 bus，为便于分析算法在不同车流情况下有效性，设置车辆的固定产生频率。在轻度交通流环境下，car 的生成速度设置为 1 分钟 15 辆，即频率为 0.25；bus 的生成速度设置为 1 分钟 3 辆，即频率为 0.05。本章算法 PPO 和 TC1 方法、TC-GAC 方法、TC-GAGC 方法、TC-HAQL 方法在 5 次仿真交通性能评价指标平均值如表 3.5 所示，ATWT 指标对比如图 3.6 所示，AJWT 指标对比如图 3.7 所示。

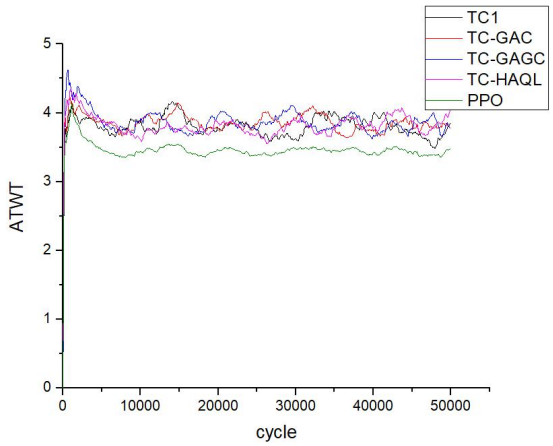

图 3.6 轻度交通流环境下 PPO 与其他控制方法的 ATWT 对比

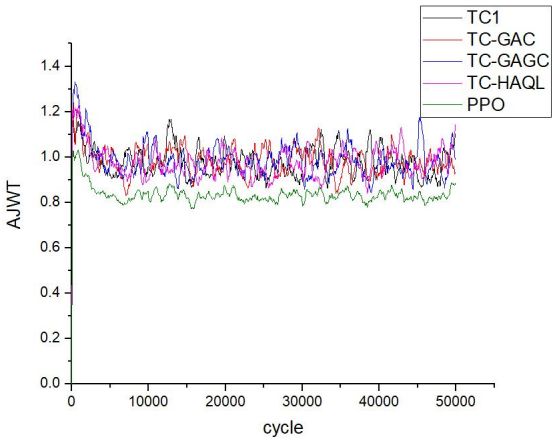

图 3.7 轻度交通流环境下 PPO 与其他控制方法的 AJWT 对比

表 3.5 轻度交通流环境下 PPO 与其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 / 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真时间	预期仿真时间
TC1	3.8033	0.9778	179833	50000	50000
TC-GAC	3.8522	0.9838	179362	50000	50000
TC-GAGC	3.8644	0.9861	179824	50000	50000
TC-HAQL	3.8143	0.9777	179230	50000	50000
PPO	3.4502	0.8341	197518	50000	50000

从表 3.5 和图 3.6、3.7 中可以看出在轻度交通流环境下基于近端策略 PPO 的交通灯控制建模方法和其他方法相比在每个指标上都是略优于其他方法，从 ATWT 和 AJWT 的曲线图可以看出来 PPO 的波动幅度较小并在一段时间内快速减少等待时间并维持相对稳定。但总体而言，对于以上五种方法在轻度交通流中表现的差异不是很

大，都能够对交通灯实现有效的控制，避免城市交通拥堵，使得城市路网中的车辆都可以尽快达到目的地。

3.4.3 重度交通流

在重度交通流环境下因为比较算法都是基于单路口的算法，起初在仿真实验中将car的产生设置为1分钟21辆，即频率为0.35时，TC1算法在仿真时间12576、TC-GAC在仿真时间9712、TC-GAGC在仿真时间30816分别出现了严重交通拥堵导致路网中的车辆堵死无法运行，TC-HAQL方法虽然可以控制交通灯避免发生车辆堵死情况但是算法波动很大，不能平稳的控制交通流。

为避免算法之间比较时出现太大误差，因此单路口仿真在重度交通流环境下，car的产生设置为1分钟18辆，即频率为0.3；bus的产生速度设置为1分钟3辆，即频率为0.05。考虑到公交在实际情况都是按点按班次发车，所以公交车bus的产生频率不发生变化，仅是增加汽车car 的产生频率。

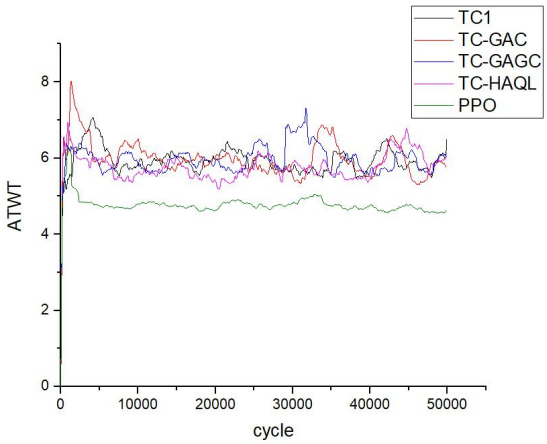

图 3.8　重度交通流环境下 PPO 与其他控制方法的 ATWT 对比

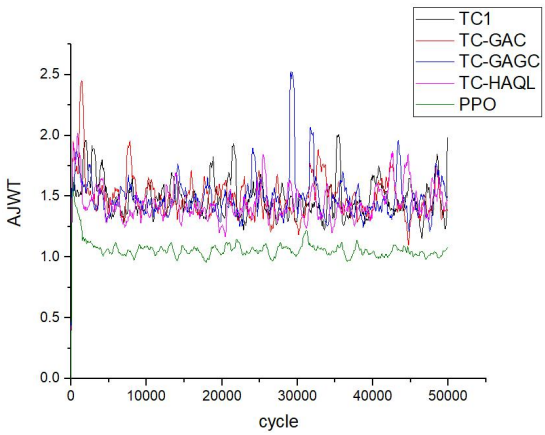

图 3.9　重度交通流环境下 PPO 与其他控制方法的 AJWT 对比

表 3.6 重度交通流环境下 PPO 与其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 / 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真时间	预期仿真时间
TC1	5.9176	1.4834	209225	50000	50000
TC-GAC	5.9828	1.4905	209719	50000	50000
TC-GAGC	5.9588	1.4947	209451	50000	50000
TC-HAQL	5.7652	1.4453	209786	50000	50000
PPO	4.7353	1.0556	230499	50000	50000

从表 3.6 和图 3.8、3.9 可以看出在重度交通流环境下 TC1、TC-GAC、TC-GAGC、TC-HAQL 等方法虽然可以有效的控制交通灯实现路网中车辆的流通，但是 PPO 方法表现的更为优秀，并且相较于其他方法在重度交通流下 ATWT 和 AJWT 都要小，并且波动的幅度也是明显小于其他方法，能够更加平稳的控制交通灯，解决重度交通流环境下路网中车辆的拥堵问题。

综上所述，PPO 方法相比较于其他方法在轻度交通流和重度交通流环境下都能有效的解决交通拥堵问题，尤其是在重度交通流环境下可以高效控制交通灯使得在有限相位执行时间内通过更多的车辆，使得车辆在路网中基本处于行驶状态，避免出现较多车辆等待和排队现象，让车辆在路口的平均等待时间减少，同时全路网减少车辆平均行驶等待时间。

3.5 本章总结

本章主要介绍了基于近端策略 PPO 的交通灯控制建模方法。首先是描述了路网建模中通用的数据集和对算法可靠性及优越性评价的三个指标，然后是描述了路网的建模方法，包括路网状态、路网相位动作、路网即时奖励、相位持续时长等方面。利用 PPO 策略实现对路网交通灯控制算法研究，在实验中分为轻度交通流和重度交通流两种环境来模拟，实验结果表明无论是在轻度交通流环境还是重度交通流环境下都优于其他比较的算法，可见基于深度学习框架和考虑过去策略对未来的影响这两种因素使得交通灯在单路口控制环境下表现更优。

第四章　基于分布式深度循环 Q 网络的交通灯控制方法

4.1 引言

城市交通灯的有效控制能在很大程度上缓解城市交通的压力，将城市路口看作是 agent，那么多个 agents 之间的协作能在很大程度上提升交通的控制能力。在第三章中未考虑到各个路口之间的合作，本章将基于分布式深度循环 Q 网络来实现多路口之间的合作，通过训练学习后各个路口之间会达成一致的协作协议，彼此之间通过满足协作协议来实现多 agents 之间的合作，使得全局的奖励最大，单位时间内全局通过各个路口的车辆更多，最大程度的在重度交通流情况下缓解城市交通的压力。

4.2 多 agents 之间的强化学习

多 agents 系统是指一个系统是在同一环境下由多个 agents 构成，单 agent 系统在解决复杂环境问题上捉襟见肘，因此采用多 agents 系统的协作性来解决复杂环境中的问题。在单 agent 问题中强调的是个体在环境中的优越表现，使得个体在环境中获取的奖励最大；在多 agents 问题中并不是单独的强调个体在环境中的最优，强调的是群体之间的合作使得全局环境下奖励达到最优值或者是次优值，或许对于某一 agent 并不是最优。多 agents 系统的强化学习反应更多的是群体之间的智能性和群体之间的社会性，类似于人类一样，群体之间的协作会让彼此之间进步得更快。

4.2.1 多 agents 协作

在文献[23][41][44]等中，针对路网交通灯控制问题中，多 agents 之间的协作主要是依靠概率函数来实现，比如在此时刻当前路口 agent 的交通信号为向北，则通过目前交通状态需要计算在采取向北动作后变为下一状态的概率，通过概率值的大小来判断下一时刻交通灯采用的执行动作和路口的交通状态。对于其邻居结点都是通过 a_{others} 来代替邻居执行的动作，然而针对邻居采取的动作并没有详细的告知是如何来体现协作和互相之间的协作，并且针对每一个路口其邻居结点最少都会有 2 个邻居路口，那么如何具体得知每个邻居路口的状态和采取的动作呢，用 a_{others} 来代替确实比较模糊，因此本章基于分布式深度循环 Q 网络(DDRQN)实现各路口之间的协作。各路口之间的协作通过自身历史状态、动作值和所有邻居（一跳邻居）的上一时刻的路口交通状态、动作值来反应，一方面可以很明确的体现邻居之间是如何合作以及明确了解到邻居路口的交通状态，另一方面解决强化学习中存储 Q 表导致状态空间爆炸问题。路口多 agents 具体协作方法会在 4.3 节中详细说明。

4.2.2 协作图

协作图[63][64]是一种无向图 G=(V,E)，是一种交互图，强调的是发送和接受消息的对象之间的组织结构。在协作图中，每个节点 i 表示一个 agent，其中结点 $i(i \in V)$、每条边 $e(i,j)$ 表示 *agent i* 和 *agent j* 之间存在的依赖关系，所以在两个 agents 之间存在协作关系。目前协作图已经广泛应用于概率环境中，比如贝叶斯网络模型，将由多个变量组合的去全局函数分解为局部值函数，每个局部值函数依赖于这些变量的子集。

协作图中将全局的协作问题分割为不同的局部问题，在每个局部找到最优解，然后通过局部的最优来获取全局的最优。全局函数 Q 的分解公式如式(4-1)所示：

$$Q(s,a) = \sum_{i,j \in E} Q_{ij}(s, a_i, a_j) \tag{4-1}$$

其中 a_i 和 a_j 分别表示 *agent i* 和 *agent j* 的动作，s 表示状态。

以城市路口为例建立一个多 agents 的协作图，如图 4.1 所示，其中 $e(i,j)$ 表示相邻 agents 之间的依赖性。

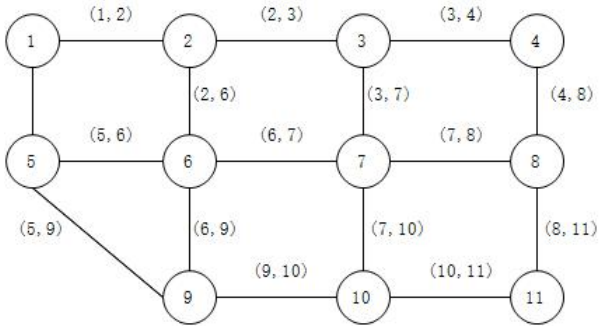

图 4.1　具有 11 个 agents 的城市路口协作图

在本章中依旧对路网采用协作图的思路，针对建立的路网模型将路网的协作图在存储时转换为邻接矩阵来存储，因此在判断某一路口的邻居时可直接查阅邻接矩阵。通过协作图的连通性，如计算某一路口时可知其邻接路口的交通状态和交通灯的动作；其邻接路口在计算时又可知与之相邻的路口，因此可通过局部的计算和累计求和获得全局最优，路网全局的交通状态也可由协作图的连通性和相邻路口之间的协作性来表现出来。

4.3 基于分布式深度循环 Q 网络的交通灯控制方法建模

从路网全局看，交通灯控制问题属于分布式问题，因此我们采用分布式深度循环 Q 网络 DDRQN[65]，其基于组合 DRQN 模型实现多代理之间的深度循环神经网络。

DDRQN 通过代理之间的通信协作使代理团队能学习解决多代理之间的协调任务，在这些任务中，代理没有给出任何预先设计的协作协议，因此为了成功合作，多个 agents 之间首先会自动开发并就自己的协作协议达成一致。

4.3.1 分布式深度循环 Q 网络 DDRQN

在单个代理的环境中已经取得了不错的效果[22]，这些方法中都是假设了每个代理完全可以观察环境状态，在城市交通中每个路口都是相互连接的，因此每个路口都可以实时观察到一跳邻居的交通状态。虽然 DQN 已经扩展到可以解决部分可观察性，但依旧主要是针对单 agent。在合作案例中，多个 agents 必须协调它们的行为，以便在面对不确定性的同时最大化他们的共同收益，不仅关于环境的隐藏状态，还关于它们的邻居观察到的内容以及它们将如何行动。

DRQN（详见本文 2.2 节）不适用于多代理的部分可观察问题，对于多代理部分状态可观察及协作问题可采用分布式深度循环 Q 网络，DDRQN 通过为每个 agent 单独分配 DRQN 训练模块的方法构建多 agents 系统，每个 agent 会单独分配一个 Q 值网络，但是造成了计算和存储资源的大量消耗。所以利用组合 DRQN 训练模块和组合各模块记忆训练的方法无法高效完成多 agents 之间的交流协作。因此将 DRQN 模型做如下改进[50][65]成为 DDRQN 模型：

（1）为了近似估计任意时刻 agent 的历史动作序列，对任意 agent 的输入中都增加上一时刻的动作信息。由于 agents 为了探索而采用随机政策，它们应该根据它们的行动观察历史来调整它们的行为，而不仅仅是观察历史，使得任意 agents 都能近似估计历史状态-动作序列。

（2）为加快网络学习速度和减少网络学习参数数目，agents 之间可共享网络参数，但每个 agent 的策略仍依据自身历史信息产生。实际上，所有 agents 只能学习和使用一个网络。但是 agents 仍然可以表现不同，因为它们接受不同的观察，从而演变出不同的隐藏状态。此外，每个 agent 都接收自己的索引 m 作为输入，使代理更容易专门化。所以权重分享大大减少了必须学习的参数数量，大大加快了学习速度。

（3）基于 DRQN 模型改进的 DDRQN 模型对应的 Q 值函数表示形式为 $Q(o_t^m, h_{t-1}^m, m, a_{t-1}^m, a_t^m; \theta_i)$，其中 m 表示当前处理的 agent 的索引，$o_t^m$ 表示 t 时刻对 agent m 的观察，h_{t-1}^m 表示 t-1 时刻 agent m 的 LSTM 隐藏层状态，a_{t-1}^m 表示 agent m 动作历史序列中的一部分，a_t^m 表示依据当前 Q 值网络的估计值选出 agent m 执行的动作。

（4）参数 θ_i 中并不含有 agent m，不受环境中任何单一 agent 的单独影响，取决于训练网络中的共享权值。

分布式深度循环 Q 网络 DDRQN 算法如下：

算法 2：分布式深度循环 Q 网络 DDRQN 算法

输入：

状态 s_1。初始化权重共享网络参数 θ_1 和目标网络参数 θ_1^-，初始化学习率 α，目标学习率 α^-；初始时间 $t=1$，初始化每个 agent 隐藏层状态 h_1^m

输出：

路口执行动作 a，即 Q 值

步骤：

While 状态 s_t 不是末状态并且 $t<T$；

for 每个 agent m

利用 $\varepsilon-greedy$ 策略随机选择动作 a_t^m

或者 $a_t^m=\arg\max_a Q(o_t^m,h_{t-1}^m,m,a_{t-1}^m,a;\theta_i)$； (4-2)

获得奖励值 r_t，状态转变为 s_{t+1}，$t=t+1$，$\nabla\theta=0$；

for $j=t-1$ to 1，每个 agent m

$$y_j^m=\begin{cases}r_j,if\ s_j\ terminal\\ r_j+\gamma\max_a Q(o_{j+1}^m,h_j^m,m,a_j^m,a;\theta_i^-),else\end{cases}；\quad (4\text{-}3)$$

$$\nabla\theta=(y_j^m-Q(o_j^m,h_{j-1}^m,m,a_{j-1}^m,a_j^m;\theta_i))^2；$$

更新权重共享参数 $\theta_{i+1}=\theta_i+\alpha\nabla\theta$； (4-4)

更新目标网络参数 $\theta_{i+1}^-=\theta_i^-+\alpha^-(\theta_{i+1}-\theta_i^-)$； (4-5)

end for

算法 2 主要描述了 DDRQN 的逻辑，在多 agents 的环境中每个 agent 都需要依照算法 2 执行。DDRQN 应用场景不同网络层的设置也会不同，同样网络层的输入输出设置也会不同，本章基于交通灯控制场景，4.3.2 节中会详细说明 DDRQN 应用于交通灯建模。

4.3.2 DDRQN 交通灯控制方法建模

在上一小节中详细说明了 DDRQN 的主要思路和 DDRQN 的算法，将 DDRQN 方法应用于交通环境中，重要的部分是神经网络的设计。路网采用 DDRQN 设计的详细神经网络如图 4.2 所示。

假设路网中时刻 t 观察的路口 agent 为 m，agent m 的邻接路口有四个，分别为 agent m_1、agent m_2、agent m_3、agent m_4（此四个邻接路口都为 agent m 的一跳邻居并且 agent m 是十字路口。其余的 T 字路口类似为拥有三个路口 agents，直角路口因为仅是作为方向的转变路口不涉及到交通灯相位的选择所以该路口不用做上述假设）。在 t 时刻 agent m 拥有自身的历史数据（agent m 采取的相位动作集和 agent m 四个方向

上的车辆状态集）表示为：$(s_1^m, a_1^m; s_2^m, a_2^m; ...s_{t-1}^m, a_{t-1}^m)$，t 时刻需要获取邻接路口的数据集为：$(s_{t-1}^{m_1}, a_{t-1}^{m_1}; s_{t-1}^{m_2}, a_{t-1}^{m_2}; s_{t-1}^{m_3}, a_{t-1}^{m_3}; s_{t-1}^{m_4}, a_{t-1}^{m_4})$，在 t 时刻对路口 agent m 的观察 o_t^m 表示为自身历史数据集和邻接路口数据集的合集：

$o_t^m = (s_1^m, a_1^m; s_2^m, a_2^m; ...s_{t-1}^m, a_{t-1}^m; s_{t-1}^{m_1}, a_{t-1}^{m_1}; s_{t-1}^{m_2}, a_{t-1}^{m_2}; s_{t-1}^{m_3}, a_{t-1}^{m_3}; s_{t-1}^{m_4}, a_{t-1}^{m_4})$。

对于每个 agent 的 Q 值函数是 $Q(o_t^m, h_{t-1}^m, m, a_{t-1}^m, a_t^m; \theta_i)$，$o_t^m$ 为上述数据的合集，h_{t-1}^m 为当前观察 agent m 在 LSTM 神经网络隐藏层中的状态，a_{t-1}^m 表示 agent m 在历史时刻上采取的交通相位动作，a_t^m 表示根据当前预估 Q 值计算得到 agent m 要执行的相位动作。在每个 agent m 执行完选择的相位动作后获取的奖励按照第三章中提及的奖励公式计算，即公式(3-1)。

在文献[23]中协作关系是考虑了自身历史状态和相位动作序列，没有考虑邻接 agents 的交通状态；文献[44]不再考虑自身的历史状态和相位动作序列，这样选择出来的执行相位动作可能不会是最优情况。因此本章中对于每一个 agent m 在计算时都通过两方面：（1）获取自身的历史状态和相位序列值，加快 agent 的个体学习；（2）获取前一时刻邻接 agents 的交通状态和相位执行值，促进团体 agents 之间的交流，使得在最短的时间内达成一致的协作协议，获得全局奖励的最大值。

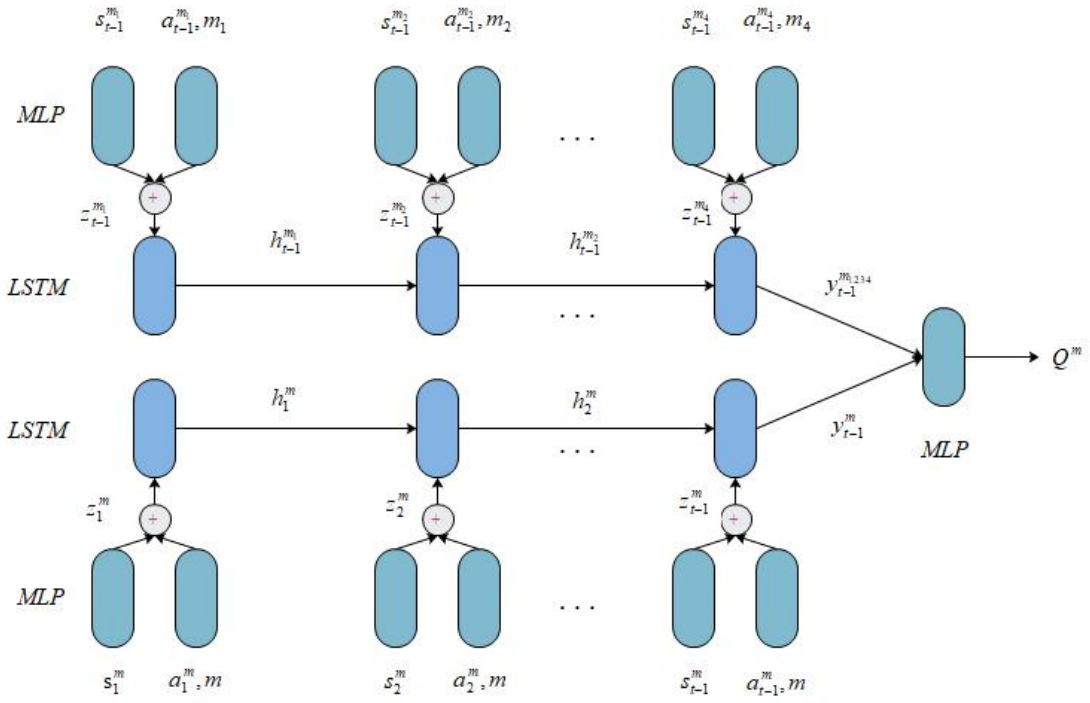

图 4.2 DDRQN 交通灯控制方法建模神经网络模型图

在图 4.2 中设计了将 DDRQN 应用于交通路口 agents 的神经网络。主要分为了两部分来计算 agent m 在 t 时刻需要执行的相位动作，一部分是历史值的计算，一部分是邻接 agents 的计算。

首先将历史状态值 s_1^m 和历史相位值 a_1^m 及当前观察的 agent m 的 m 值作为输入，这时值会通过两个单层的 MLP 神经网络传递得到 z_1^m。z_1^m 计算公式如(4-6)：

$$z_1^m = MLP(s_1^m) \oplus MLP(a_1^m, m) \tag{4-6}$$

z_1^m 经过 LSTM 神经网络得到在第一时刻的隐藏层状态 h_1^m。类似于在第一时刻输入，第二时刻同样可以得到 z_2^m，但此时需要由第一时刻隐藏层状态 h_1^m 和 z_2^m 共同决定隐藏层 h_2^m 的状态，计算公式如(4-7)：

$$h_2^m = LSTM(z_2^m, h_1^m) \tag{4-7}$$

由此可以推导出任意时刻 t，agent m 的计算公式为：

$$z_t^m = MLP(s_t^m) \oplus MLP(a_t^m, m) \tag{4-8}$$

$$h_t^m = LSTM(z_t^m, h_{t-1}^m) \tag{4-9}$$

每一个历史时刻的数据都按照上述的公式(4-8)、(4-9)计算。直到当前时刻 t 的前一时刻 t-1，经过 LSTM 神经网络后计算得到 h_{t-1}^m，此时设 $y_{t-1}^m = h_{t-1}^m$ 并保留当前计算的值。这样 agent m 自身历史数据的全部计算就结束了。对于自身历史数据的计算并不需要从开始一直计算到结束，可自行规定历史数据的时间步长。

第二部分计算的就是邻接路口 agents。针对邻接路口不再需要计算历史时刻的状态数据和相位数据，仅需要考虑当前时刻 t 的前一时刻 t-1 的状态数据和相位数据即可。因为对于邻接路口自身而言，它们在 t-1 时刻的数据一部分也是通过自身历史数据计算得来，若再次考虑计算邻接路口的历史数据就会导致空间状态爆炸并且路网全局的重复计算也会导致算法性能的降低和时间的损耗。在本节的开始已经假设了四个邻接路口 m_1、m_2、m_3、m_4，针对每一个邻接 agent 获取其在 t-1 时刻的状态值和相位值，分别表示为 $s_{t-1}^{m_1}, a_{t-1}^{m_1}$；$s_{t-1}^{m_2}, a_{t-1}^{m_2}$；$s_{t-1}^{m_3}, a_{t-1}^{m_3}$；$s_{t-1}^{m_4}, a_{t-1}^{m_4}$。按照公式(4-8)可计算每个 agent 通过 MLP 神经网络的值 $z_{t-1}^{m_1}$、$z_{t-1}^{m_2}$、$z_{t-1}^{m_3}$、$z_{t-1}^{m_4}$。假设对于 agents 输入的顺序为 m_1、m_2、m_3、m_4，则 m_1 隐藏层状态为 $h_{t-1}^{m_1}$，m_2 隐藏层状态为 $h_{t-1}^{m_2} = LSTM(z_{t-1}^{m_2}, h_{t-1}^{m_1})$，同理 m_3 和 m_4 的隐藏层状态分别为 $h_{t-1}^{m_3} = LSTM(z_{t-1}^{m_3}, h_{t-1}^{m_2})$、$h_{t-1}^{m_4} = LSTM(z_{t-1}^{m_4}, h_{t-1}^{m_3})$。$h_{t-1}^{m_4}$ 作为邻接 agents 的最后一层隐藏层状态可设 $y_{t-1}^{m_{1234}} = h_{t-1}^{m_4}$，由此关于 agent m 的邻接路口 agents 在 t-1 时刻的状态和相位动作就全部计算结束。

最后将 agent m 自身历史数据训练得到的 y_{t-1}^m 和其四个邻接路口 agents 计算得到的 $y_{t-1}^{m_{1234}}$ 经过 MLP 神经网络选择 agent m 在 t 时刻的最佳执行相位。Q 值选择公式为公式(4-10)：

$$Q^m = MLP(y_{t-1}^m \| y_{t-1}^{m_{1234}}) \tag{4-10}$$

通过上述计算得到的 Q 值（即执行相位）即考虑了 agent m 自身的历史影响也考虑了邻接路口 agents 之间的相互协作。相互协作是通过 agents 之间达成一致的通信协议来完成，目的都是为了使得全局奖励能够得到最大值。为了避免红绿灯的频繁切换导致交通事故发生和交通秩序的混乱，选择执行相位后需持续最低时间段 20s，具体相位持续时长可详看 3.3.2 节。

在任意多路口交通路网中，基于 DDRQN 原理交通灯控制建模方法的学习步骤如下：

Step1：依据 3.3.1 节中路网状态描述方法初始化路网中路口控制器 agents 的状态矩阵，初始值为 0。初始化折扣因子 γ，$\varepsilon-greedy$ 策略中的 ε，初始化权重共享参数 θ_1 和目标网络参数 θ_1^-。每个 agent 的即时奖励值 r=0，仿真时间步 t 的初始值 0，对于每一个 agent 的 h_1^m 初始值均为 0。

Step2：$t \leftarrow t+1$，根据 $\varepsilon-greedy$ 策略随机一个相位动作。每个路口控制器观察其入口车道车辆状态变化情况，获取历史车道状态值和相位值，并获取前一时刻与其相邻路口的车道状态值和相位值。

Step3：将获取的值作为 MLP 神经网络的输入，获取的值见图 4.2，通过公式(4-8)和公式(4-9)计算 LSTM 隐藏层需要获取的状态值。获取到邻接路口的隐藏状态值和自身隐藏层状态值后通过 MLP 神经网络和公式(4-10)确定最终在 t 时刻当前计算的 agent 需要执行的相位动作。

Step4：对每一个 agent 执行上述计算并获取即时奖励值。并根据公式(4-4)、(4-5)计算损失函数误差，更新网络权值 θ_1 和 θ_1^-。

Step5：若 $t<T$ 并且 $s_t \neq$ terminal，转 Step2。

4.4 仿真实验与结果分析

为了验证基于 DDRQN 建模的城市交通灯控制算法的性能，将 DDRQN 方法和 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法分别在图 3.4 所示的多路口交通网络中仿真，并在轻度交通流和重度交通流下分别进行 5 次仿真实验。

4.4.1 路网环境配置

本章实验依旧采用 SUMO，实验基本环境、路网中的路口、车道、车辆类型、仿真时间等基本设置与 3.4.1 节中的完全相同，本章中仿真采集数据的路口分为 3 组：（1）图 3.4 中红色数字标记的 2、3、4、10、7 路口采集 2 次；（2）图 3.4 中红色数

字标记的 19、9、2、3、6 路口采集 2 次；（3）图 3.4 中红色数字标记的 10、11、4、15、12 路口采集 1 次。

比较方法参数设置：TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法中初始化参数：折扣因子 $\gamma = 0.9$，$\varepsilon - greedy$ 策略中的 $\varepsilon = 0.01$。

DDRQN 方法参数设置：

表 4.1 DDRQN 方法参数设置

参数	值
折扣因子 γ	0.9
$\varepsilon - greedy$ 策略中的 ε	初始为 1 迭代递减至 0.01
学习率 α	0.0025
目标学习率 α^-	0.0025
批度大小	32
经验回放单元大小	60000

DDRQN 方法中神经网络设置：（假设在任意时刻 t，$0 < t < T$ ）

表 4.2 DDRQN 方法中 agent m 历史数据神经网络设置

agent m 历史数据	参数
输入 MLP 神经网络	$z_t^m = MLP[1\times 64](s_t^m) \oplus MLP[2\times 64](a_t^m, m)$
隐藏层 LSTM 神经网络	$h_t^m = LSTM[64](z_t^m, h_{t-1}^m); h_1^m = z_1^m$

表 4.3 DDRQN 方法中 agent m 邻接路口数据神经网络设置

agent m 邻接路口数据	参数
输入 MLP 神经网络	$z_{t-1}^{m_2} = MLP[1\times 64](s_{t-1}^{m_2}) \oplus MLP[2\times 64](a_{t-1}^{m_2}, m_2)$
隐藏层 LSTM 神经网络	$h_{t-1}^{m_2} = LSTM[64](z_{t-1}^{m_2}, h_{t-1}^{m_1}); h_{t-1}^{m_1} = z_{t-1}^{m_1}$

表 4.4 DDRQN 方法中 Q 值选择神经网络设置

	参数
Q 值选择 MLP 神经网络	$Q_t^m = MLP[128\times 64, 64\times 64, 64\times 1](y_{t-1}^m \,\|\|\, y_{t-1}^{m_{1234}})$

上表均是对于路网中任意 agent 和任意时刻的描述，其中对于 agent m 的邻接路口在上表 4.3 中仅是以 m_1 和 m_2 为例进行说明，其余路口的计算方法和其他所有计算的细节都可参见 4.3.2 节。

4.4.2 轻度交通流

利用 SUMO 产生车辆 car 和 bus，为便于分析算法在不同车流情况下有效性，设置车辆的固定产生频率，在轻度交通流环境下，其中 car 的生成速度设置为 1 分钟 15 辆，即频率为 0.25；bus 的生成速度设置为 1 分钟 3 辆，即频率为 0.05。在轻度交通流情况下，DDRQN 方法和 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法交通性能评价指标的 5 次仿真结果的平均值如表 4.5 所示。ATWT 指标对比如图 4.3 所示，AJWT 指标对比如图 4.4 所示。

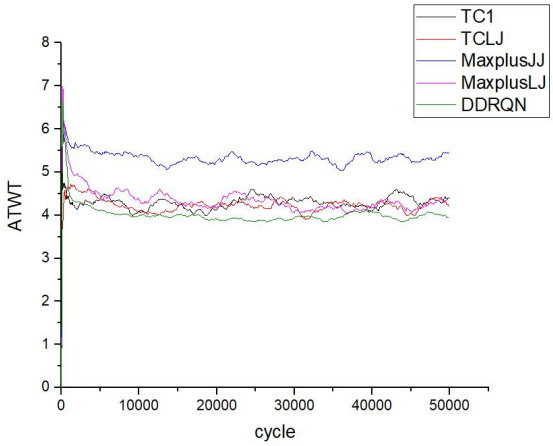

图 4.3 轻度交通流情况下 DDRQN 与其他控制方法的 ATWT 对比

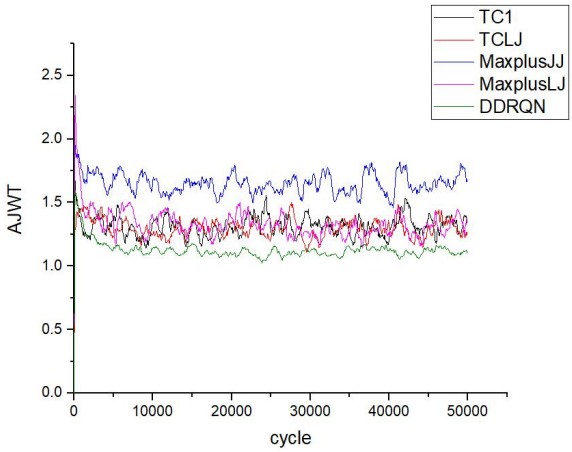

图 4.4 轻度交通流情况下 DDRQN 与其他控制方法的 AJWT 对比

表 4.5 轻度交通流情况下 DDRQN 和其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 / 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真时间	预期仿真时间
TC1	4.2853	1.3133	179686	50000	50000
TCLJ	4.2331	1.2960	179551	50000	50000

MaxplusJJ	5.3241	1.6494	179462	50000	50000
MaxplusLJ	4.3501	1.3307	179765	50000	50000
DDRQN	3.9894	1.1252	197408	50000	50000

从表 4.5 和图 4.3、4.4 中可以看出在轻度交通流环境下 DDRQN 方法和其他方法相比在每个指标上都是略优于其他方法。从 ATWT 和 AJWT 的曲线图可以看出来 DDRQN 的波动幅度较小并在一段时间内快速减少等待时间并维持相对稳定。但总体而言，在多个路口协同控制的情况下对于以上五种方法在轻度交通流中表现的差异不是很大，都能够对交通灯实现有效的控制，避免发生交通拥堵，使得车辆尽快达到目的地。

4.4.3 重度交通流

在章节 3.4.3 中对单路口重度交通流仿真时，car 产生频率为 0.35 时出现了严重的交通拥堵，说明当车流量增多时单路口交通灯控制算法无法有效的控制路口交通灯。本章节仿真算法为多路口协作控制算法，为验证多路口协作控制算法在重度交通流环境下比单路口控制算法更加有效及验证多种多路口协作控制算法的优劣性，重度交通流情况下，car 的产生设置为 1 分钟 21 辆，即频率为 0.35，bus 的产生设置为 1 分钟 3 辆，即频率为 0.05。考虑到公交在实际情况都是按点按班次发车，所以公交车 bus 的产生频率不发生变化，仅是增加汽车 car 的产生频率。

在重度交通流情况下，DDRQN 方法和 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法进行仿真比较，交通性能评价指标的 5 次仿真结果的平均值如表 4.6 所示。ATWT 指标对比如图 4.5 所示，AJWT 指标对比如图 4.6 所示。

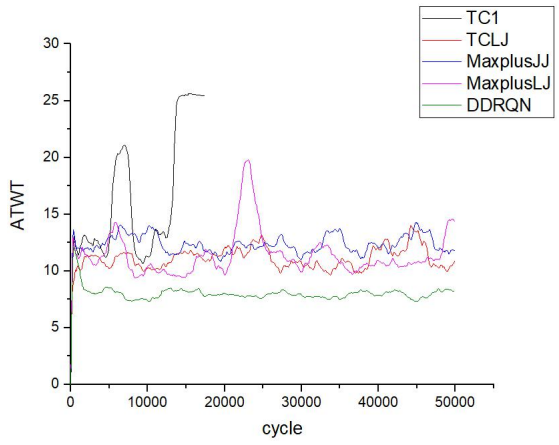

图 4.5　重度交通流情况下 DDRQN 与其他控制方法的 ATWT 对比

从表 4.6 和图 4.5、4.6 可以看出在重度交通流环境下 TC1 方法发生了交通拥堵，

在交通发生拥堵后再无法有效的控制交通灯，使得整个交通网络陷入瘫痪状态。其他三个方法（TCLJ、MaxplusJJ、MaxplusLJ）虽然能在重度交通流下有效的控制交通灯实现车辆的流通，但是 DDRQN 表现的更为优秀，并且相对于其他方法在重度交通流下 ATWT 和 AJWT 都要小，并且波动的幅度也是明显小于其他方法，能够更加平稳的控制交通灯，解决重度交通流环境下路网中车辆的拥堵问题。

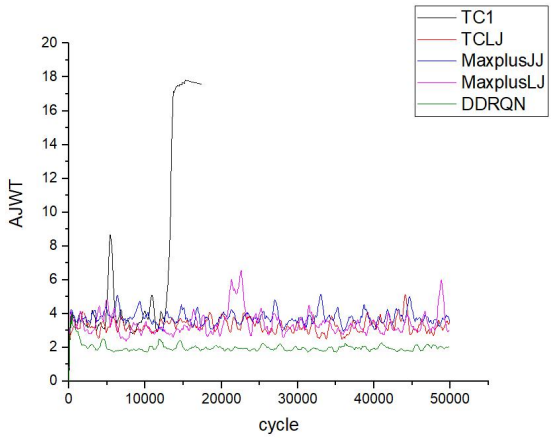

图 4.6 重度交通流情况下 DDRQN 与其他控制方法的 AJWT 对比

表 4.6 重度交通流情况下 DDRQN 和其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 / 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真时间	预期仿真时间
TC1	16.3977	7.1189	64871	17408	50000
TCLJ	11.0840	3.2903	239805	50000	50000
MaxplusJJ	12.3432	3.7755	239847	50000	50000
MaxplusLJ	11.4043	3.4207	239903	50000	50000
DDRQN	7.9061	1.9651	335864	50000	50000

综上所述，DDRQN 方法相比较于其他方法在轻度交通流和重度交通流环境下都能有效的解决交通拥堵问题，尤其是在重度交通流环境下可以高效控制交通灯使得在有限相位执行时间内通过更多的车辆，使得车辆在路网中基本处于行驶状态，避免出现较多车辆等待和排队现象，让车辆在路口的平均等待时间减少，同时全路网减少车辆平均行驶等待时间。

4.5 本章总结

本章主要介绍了基于分布式深度循环 Q 网络的交通灯控制方法。首先是介绍了

多 agents 协作、协作图相关知识并分析了目前多 agents 协作中的不足，然后详细介绍了 DDRQN，并基于 DDRQN 和第三章中的交通建模方法建立了多 agents 之间协作的交通灯控制模型。通过仿真实验结果表明，采用基于分布式深度循环 Q 网络建立的交通灯控制算法优于其他 agents 合作算法，相邻路口之间通过前期的训练和学习达成了一致的合作协议，很大程度的提高了交通网络的整体性能，有效的缓解了重度交通流情况下的交通压力。

第五章　基于分布式 W 学习的交通灯控制方法

5.1 引言

在城市路网中，每个路口交通灯的控制应该是独立并相互协作的，城市交通灯控制系统若为集中式系统，高层控制系统出现缺陷时很容易导致整个城市交通的瘫痪，因此需要采用分布式控制系统。在分布式控制系统中，每个路口相互独立又相互协作，才能更好得控制交通流量，本章单个路口的 Q 学习计算依据第三章的计算方式（基于近端策略 PPO），具体算法可详见第三章；在相互协作方面采用分布式 W-Learning[66]来完成，为加快各路口之间的数据传输和学习模型的学习采用协作图原理来完成，因此本章的重点是如何体现多个路口的协作和加快模型的学习。

5.2 分布式 W-Learning

Distributed W-Learning(DWL)分布式 W 学习是一种强化学习的算法，主要用以解决基于协作代理的普适系统。DWL 支持针对多个异构策略的优化，并解决由于负责实现它们的代理的异构性而产生的挑战。DWL 学习并利用代理之间和策略之间的依赖关系来提高整体系统性能。代理不是总是执行本地最佳操作，而是了解他们的行为如何影响他们的直接邻居并执行邻近代理建议的操作，如果他们的重要性超过了使用预定义或学习的协作系数，那么缩放本地操作的重要性，下文将详细解释 DWL。

城市路网中由于地理位置的不同或者是突发事件紧急事件的影响，导致为了缓解交通压力每个路口的策略都会不同。比如在路口 1 策略是公交车优先；在路口 2 策略是停车场出来的车优先；在路口 3 策略是公交车和救护车优先等；针对不同的城市不同的路口可以设置不同的策略，不同的策略优先级也会不同，在任何路口中救护车或者是消防车、警车等的策略优先级都远高于其他策略。那么当策略众多并且路口众多时仅通过交通路口的状态来分析相位执行动作显然不够，DWL 算法可以解决上述问题，DWL 支持代理之间和策略之间的协作，此外，DWL 学习并利用代理之间和策略之间的依赖关系，以便在保持策略优先级的同时提高整体系统性能。在文献[66]中说明 DWL 不仅可以适应于多 agents 多策略协作也适应于多 agents 单策略协作。

5.2.1 DWL 交通灯协作控制

DWL 中如何体现多 agents 多策略之间的协作是保证分布式系统能稳定高效运行的重要手段，需要考虑哪些 agents 之间进行交互，使得多 agents 之间产生积极的效应。在路网中对于每个路口而言能互相传递数据并且最直观的观察就是其一跳邻居路口，

因此本章中多 agents 之间的 DWL 协作都是通过路口之间的一跳邻居来实现。

DWL 中含有两种策略：一种是本地策略（本章中的本地策略指当前观察路口自身处理交通灯相位和车流量控制的策略，可单策略可多策略），一种是远程策略（本章中的远程策略指当前观察路口的所有一跳邻居路口中包含的处理交通灯相位和车流量控制策略，可单策略可多策略，其一跳邻居路口的策略可相同可不同）。若本地策略为单策略，远程策略中所有一跳邻居都为单策略并且与本地策略相同，则该系统为多 agents 单策略协作系统，同样适应于 DWL 算法。

在路网中对于任何一个路口都可做如下 DWL 协作：对于每个路口都基于本地策略和本地交通状态学习本地 Q 值，并且基于其一跳邻居路口的策略（远程策略）和本地状态来学习 Q 值，以便了解路口自身采取的行动是否适应于一跳邻居路口的策略，是否对其产生了积极的影响。在 DWL 利用策略和状态进行 Q 学习的时候同时进行 W 学习，可基于本地策略和远程策略学习得出不同或者是相同的 W 值，以便了解路口采取的行动（即路口红绿灯相位动作，下同）是如何影响一跳邻居路口并得知影响的程度。通过使用 Q 学习和 W 学习的这种组合，各个路口可以了解到其自身路口的行动是否适应于一跳邻居的策略，并了解到其自身路口如果不执行相对应的行动将如何影响邻居的策略。因此在每个时间步每个路口都将依据其自身状态（路口交通状态，详见论文第三章）、本地策略、远程策略学习 Q 值及 W 值。

上述中所言 Q 值为 Q 学习后路口需要执行的相位动作值，W 值即权重值，指本地策略或者是远程策略的影响值。为了让代理人（即路网中的路口）有动力代表邻居的策略（即一跳邻居的远程策略）执行行动建议，即为邻居的良好表现做出贡献，需要为表现良好的邻居获得奖励，否则只会激励他们最大化本地良好表现所获得的回报。为实现这一目标，我们通过使用“远程策略”的概念，使代理能够了解其代理如何以及在多大程度上影响其邻居的机制，其中每个远程策略对应于其中一个实现的本地策略，并且每当邻近代理的相应本地策略收到奖励时在本地接收奖励。对于远程策略作为协作机制的使用，在一跳邻居之间需要交换的即时观察是每个代理的本地策略的状态以及为了更新 Q 值而在该状态下接收的奖励和远程策略的 W 值。

W 学习与 Q 学习紧密关联，文献[69-71]中详细说明了 W-values 与静态 Q-Learning、动态 Q-Learning、内存空间占比、获取最大集体奖励、多目标等不同条件下 W-values 的取值方式。在路网中路口的交通状态时刻变化并希望获取最大的集体奖励，因此若在 DWL 学习的时间步内任意 $agent_i$ 与其一跳邻居 $agent_j$ 之间本地策略和远程策略协作的重要程度不发生变化，则 $agent_i$ 与一跳邻居 $agent_j$ 之间的任一 W-values 仍为原值，若他们之间本地策略和远程策略协作的重要程度因为路况的原因发生变化，则按照公式(5-1)更新 W 值：

$$W_i(s) := (1-\alpha)W_i(s) + \alpha(Q_i(s,a_i) - (r_i + \gamma \max Q(s^{'}, a_i^{'}))) \tag{5-1}$$

$W_i(s)$ 表示在状态 s 下 $agent_i$ 的本地或远程策略的影响值，α 为学习率，γ 为折扣率。

为了使路口能对远程策略的行动建议给予不同程度的重要性衡量，引入协作系数 C。本地策略的行动建议将以 W 的全值考虑，对于远程策略的 W 值则乘以协作系数 C，$0 \le C \le 1$。当 C=0 表示完全不考虑邻居表现，仅考虑本地策略的影响，属于非协作系统。当 C=1 表示远程策略和本地策略的重要性是一样重要的，关心邻居的表现就和关心自己的表现一样，属于完全协作系统。当$0 < C < 1$时，使得本地代理的本地策略的重要性高于一跳邻居的远程策略，对于每个代理而言都是自私的，本地的策略和远程策略相比也是相对重要的。C 的值可以是预定义的相同值或是不同值，也可以是每个代理学习自身的 C 值，以最大化本地策略和远程策略中的奖励。

协作系数 C 的更新通过 Q 学习实现，因为本章单路口的计算按照第三章的近端策略 PPO 实现，所以协作系数 C 更新公式为(5-2)：

$$Q(s,C) \leftarrow Q(s,C) + \rho\alpha[r + \max_{C^{'}} Q(s^{'}, C^{'}) - Q(s,C)] \tag{5-2}$$

其中 α 为学习率，通过式(5-2)可以看出协作系数 C 与本地策略和远程策略并没有关系，C 的值与当前路口的交通状态有关，图 5.1 为路口间协作系数示意图。

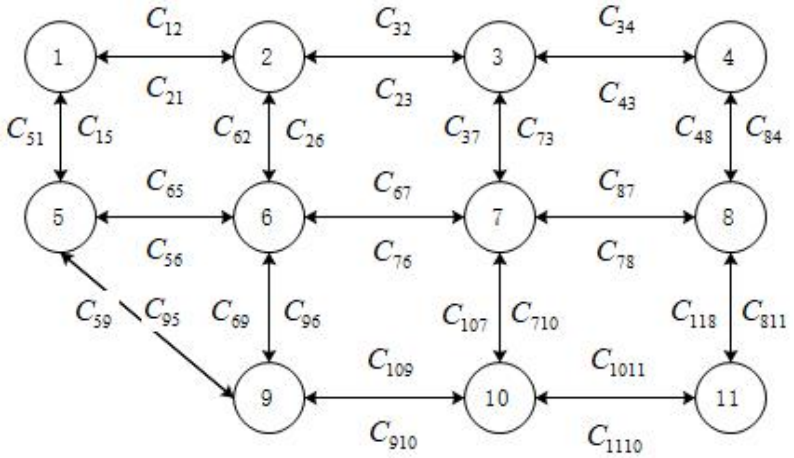

图 5.1 路口间协作系数

其中 C_{ij} 表示 $agent_i$ 对于 $agent_j$ 远程策略的协作系数，以路口 7 为例，C_{73} 表示路口 7 对于路口 3 远程策略的协作系数，C_{37} 表示路口 3 对于路口 7 远程策略的协作系数。通过当前路口 7 的状态计算新的 C 值，那么 $C_{73}=C_{76}=C_{78}=C_{710}$，路口 7 在当前交通状态下对邻居路口 3、6、8、10 远程策略的协作系数相同。同理可根据每个路口的交

通状态计算与其邻居远程策略的协作系数。

综上所述，DWL 的协作特点具体如下表 5.1：

表 5.1 DWL 协作特点体现[66]

	需求	DWL 协作体现
1	分布式	每项政策的本地学习
2	学习原理/基础	Q-Learning、W-Learning
3	同时多个策略	W-Learning
4	多 agents 之间的协作	远程策略
5	多策略之间的协作	远程策略
6	尊重策略优先	远程策略、W-Learning
7	异构策略/代理	远程策略
8	了解与 W-Learning 合作的代理/时间/数量	W-Learning、协作系数 C

5.2.2 DWL 数据集

将 DWL 应用于路网中需要设置额外的数据集合，主要数据集合如下：

（1）每个 $agent_i$ 都有自己的一跳邻居，对于每个 $agent_i$ 其一跳邻居的集合为 $N_i=\{N_{i1},...,N_{im}\}$，对于路网而言 $1\le m\le 4$； $agent_j\in I$， $agent_j$ 为 $agent_i$ 的一跳邻居。

（2） $agent_i$ 的本地策略集合： $LP_i=\{P_{i1},...,P_{ip}\}$；本地策略可为一种也可为多种。

（3）每个 $agent_i$ 都有自己的远程策略集合（即一跳邻居的本地策略）：

$RP_i=\{RP_{ij1},...,RP_{ijr}\}$； $agent_i$ 对应一跳邻居 $agent_j$ 的本地策略 LP_{jk}。

（4）每个代理 $agent_i$ 都需要 Q 学习和 W 学习，当前时刻状态为 s_t，动作为 a_t，上一时刻状态为 s_{t-1}，动作为 a_{t-1}。

5.2.3 DWL 算法

在路网分布式系统中，每个路口都可以看作是独立又相互关联的代理，因此路网中每个路口都需要单独执行 DWL 算法，并通过 W-Learning 和协作系数 C 来体现相互间的协作。

对于每个 agent 都需要先执行初始化，首先创建本地策略并利用本地策略初始化 Q-Learning 和 W-Learning，然后是创建远程策略，初始化后每个代理（路口）与其一跳邻居交换每个策略的状态-动作对，该状态-动作对用于初始化远程策略的 Q-Learning 和 W-Learning，算法 3 为 DWL 在路网任意路口 $agent_i$ 初始化过程：（算法中 RP_{ijk} 表示 $agent_i$ 到 $agent_j$ 的第 k 条远程策略）

算法 3：DWL 在路网任意路口 $agent_i$ 初始化过程

输入：

本地策略、远程策略

输出：

本地策略 Q 值、W 值；远程策略 Q 值、W 值

步骤：

初始化本地策略： 任意 $agent_i$

for 每条策略 P_{il} to LP_i

Q-Learning(P_{il} states, $agent_i$ cation)；

W-Learning(P_{il} states)；

end

创建远程策略：

for 任意 $agent_j$ to N_i

for 每条远程策略 LP_{jk} to LP_j

添加相对应的 RP_{ijk} to RP_i；

end

初始化远程策略：

for 每条 RP_{ijk} to RP_i

Q-Learning(RP_{ijk} states, $agent_i$ actions)；

W-Learning(RP_{ijk} states)；

end

end

初始化后对于每个路口每个时间步都重复做 DWL 学习过程，每个路口使用 Q-Learning 和 W-Learning 学习本地 Q 值和 W 值并获得该状态的奖励，基于获得的奖励更新本地 Q 值和 W 值。同理，每个路口都会学习远程策略-本地操作对的 Q 值以及远程策略的 W 值，路口在每个时间步接收有关其邻居的每个策略的当前状态的信息以及他们收到的奖励，根据收到的奖励，代理更新其远程策略的 Q 值和 W 值。算法 4 为任意路口 $agent_i$ 在一个时间步的学习过程：

算法 4：任意路口 $agent_i$ 在一个时间步上 DWL 学习过程

输入：

本地策略、远程策略，路口交通状态

输出：

奖励值；本地策略 Q 值、W 值；远程策略 Q 值、W 值

步骤：

通过本地策略获取建议相位值：

for 每条策略 P_{il} to LP_i

确定 P_{il} 下的路口交通状态 s_{il}；

$agent_i$ 得到环境的奖励值 r_{il}；

利用 r_{il} 更新本地 Q 值、W 值；

获取路口在状态 s_{il} 执行相位动作 a_{il}，即 Q 值；

通过公式(5-1)获取状态 s_{il} 下的 $W(s_{il})$ 值；

end

通过远程策略获取建议相位值：

for 每条远程策略 RP_{ijk} to RP_i

从一跳邻居 $agent_j$ 获取 RP_{ijk} 下的状态 s_{ijk}；

从一条邻居 $agent_j$ 获取状态 s_{ijk} 下的奖励值 r_{ijk}；

利用 r_{ijk} 更新远程 Q 值、W 值；

获取路口在状态 s_{ijk} 下执行相位动作 a_{ijk}，即 Q 值；

通过公式(5-1)获取状态 s_{ijk} 下的 $W(s_{ijk})$ 值；

end

根据公式(5-3)比较本地策略得到的 W 值和远程策略得到的 W 值；

选择执行最大 W 值所对应的 Q 值相位动作；

end

在每个时间步，更新本地策略和远程策略的 Q 值和 W 值之后需要根据本地策略和远程策略的建议动作选择当前时刻代理执行的最终动作。选择的依据是选择最大 W 值所对应的 Q 值动作，即公式(5-3)：

$$W_{win} = \max(W_{il}, C \times W_{ijk}) \tag{5-3}$$

W_{il} 为本地策略的 W 值，C 为协作系数，W_{ijk} 为代理 $agent_i$ 到 $agent_j$ 的第 k 条远程策略的 W 值。其中 $agent_i$ 的一跳邻居不同所对应的协作系数 C 也不同，C 的值可以是预定义的也可以自学习的，为了能使邻接代理（即路网中单个路口与一跳邻居路口）能更充分的协作并使得本地策略和远程策略获取的奖励和最大，协作系数 C 采用自学。图 5.4 为任意 agent 在一个时间步执行 DWL，算法 5 为协作系数 C 的自学习过程：

算法 5：协作系数 C 自学习过程

输入：

本地策略得到的总奖励；远程策略得到的总奖励；本地和远程策略的 Q 值、W 值

输出：

C 值，最大 Q 值、W 值

步骤：

初始化总奖励值 0

获取所有本地策略得到的总奖励 totalReward+= r_{il} ；

获取所有远程策略得到的总奖励 totalReward+= r_{ijk} ；

本地策略得到总奖励与远程策略得到总奖励求和为 totalReward；

end

利用 totalReward 更新以前使用的 C 的 Q 值： $Q(s_t, C_{t-1})$ ；

根据 Q 值选择 C 的下一个值；

C 值乘以所有远程策略的 W 值： $W(s_{ijk})\times = C$ ；

end

选择所有本地策略和远程策略中执行最大 W 值所对应的 Q 值相位动作；

$W_{\max} = \max(W(s_{il}), W(s_{ijk}))$

执行 $Q_{\max}$ ；

end

DWL 的优势主要体现如下：（1）同时优化多个流量优化目标的能力；（2）路口相互协作的能力；（3）每个路口能够学习与其他路口最佳协作程度的能力；（4）每个目标都被指定为独立的 Q 学习过程，因此可以添加，删除或修改目标，并且并非所有代理都需要实现相同的目标或具有相同数量的目标。在路网中任意路口的相邻路口都有可能会出现交通事故或其他紧急情况、施工封锁单个路口等复杂情况，采用 DWL 学习就可单独删除该路口，重新计算 W 值，在不破坏已有部署系统的条件下仍可最大化调控城市交通流。

5.3 并行传递数据

在大规模多代理的环境中，系统无法及时得知时间步内如何调配才会尽可能达到最优，因此需要在时间步内运行时及时掌握代理之间的状态和调控策略。在路网环境中包含的路口众多并且当车流量多时交通灯控制系统会更加难以有效的控制和调配，并且每个路口所希望达到的处理目标也会不相同。路网路口交通灯控制系统属于复杂

的多代理多目标分布式控制系统，为了加快系统的学习和保障路口之间的协作，需要并行处理数据在代理之间的传递。

为实现多代理系统的快速学习[67]可以采用强化学习、分布式 W-Learning、启发式加速 Q 学习、遗传算法等。当然不同的方法在处理不同的问题时有不同的优势，在路网问题中单 agent 的计算已经使用基于近端策略 PPO（详见第三章）和分布式 W-Learning 来处理，目前需要解决的问题是多个代理之间如何传递数据以及传递什么数据可以加快系统的学习，减少系统的学习时间。本章通过各路口之间并行传递数据完成代理之间的知识共享和重用。

5.3.1 数据传递方式

文献[67]表明强化学习可以通过共享环境状态和不同代理之间的良好动作来实现模型的快速学习，在一个任务（称为源任务）中完成学习，将信息从源传递给目标代理，然后目标代理将其合并进入其状态空间。简单而言就是将源任务的信息传递到目标任务的状态空间中作为学习的新知识，通常理解的数据传递方式有两种，分为单向顺序传递和多向并行传递。

在单向顺序传递中，信息从源目标传向目标任务，图 5.2 中显示源任务 1 将数据传向目标任务 1；执行完任务 1 后，源任务 2 将数据传向目标任务 2；执行完任务 2 后，源任务 3 将数据传向目标任务 3。

在路网环境中对于多代理而言并不能确定哪个路口属于源任务哪个属于目标任务，在任何时候路网中的任意路口都可以成为源任务也可以是目标任务，并且为了实现路口之间的协作需要路口之间相互传递信息，属于双向传递，违反了单向传递的原则，因此针对多代理之间同时互相传递数据采用多向并行传递。在多向并行传递中不区分源任务和目标任务，任何代理在任何时刻都可以是源任务也可以目标任务，双向传递数据，实现多代理同一时间步内的相互协作。图 5.3 所示为多向并行传递。

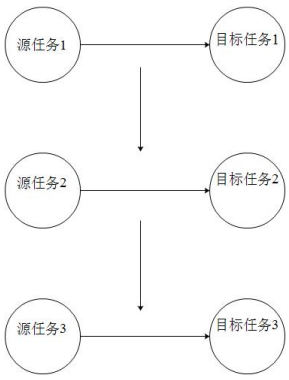

图 5.2　单向顺序传递

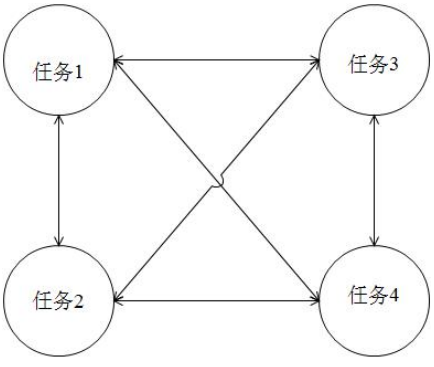

图 5.3　多向并行传递

图 5.3 显示为同一时间步内多代理之间的双向数据传递，任务 1 与任务 2、任务 3、任务 4 之间相互传递。对于路网中的路口而言其一跳邻居的数量范围为1～4 之间，直接相连的路口同一时间内相互传递数据，将多向并行传递应用于路网如图 5.4 所示，可以理解为有向协作图。

5.3.2 数据处理

多代理（多个路口）之间互相传递数据需要考虑如下几个问题（1）传递什么数据？（2）什么时候传递数据并且传递的频率如何？（3）如何将接收的数据纳入目标代理的状态空间中？在上文中已经声明路网系统中任何路口都可以是源任务也可以是目的任务，因此不需要考虑如何选择表现更加良好的代理（路口）作为源任务和目的任务。当然如果路网特别复杂为了简化系统可以考虑将路网中的边界路口不作为代理考虑，下文将详细说明上述中的三个问题。

（1）传递什么数据？

在文献[67][72]中说明了不同系统需要传递的不同数据集，数据集中包含有利于系统学习也有毫无作用的数据，但基本都认可在传递数据时传递最近访问过的状态最有意义，因为这些数据是最新并且已经不太可能转移过。

在路网中需要传输实时数据，因此在处理需要传递的数据之前说明需要用到的数据集，本章节 5.3.2 中声明的 DWL 数据集依然可用，此外针对各 agents 之间需要传递的数据主要如下：1）本地交通状态矩阵计算出的本地 Q 值和 W 值并获得该状态的奖励；2）远程策略-本地交通状态的 Q 值以及远程策略的 W 值，代理在每个时间步接收有关其邻居的每个策略的当前状态的信息以及他们收到的奖励。简言在各 agents 之间需要互相传递的数据就是本地交通状态矩阵、本地 Q 值、W 值以及获得的奖励；远程（一跳邻居）交通状态矩阵、远程 Q 值、W 值以及相对应的奖励。

（2）什么时候传递数据并且传递的频率如何？

在 RL 中，状态的期望值仅在该状态经历多次之后才成为真实值的代表，文献[67]中说明在 RL 传递数据（转移状态信息）时主要有以下方法：1）每次更改时都传输数据；2）仅在值近似收敛时传输数据；3）变化最大的时候传输数据。对于每一种转移数据的方法都有各自的利弊，并且没有任何一种方法适用于所有的状态，具体各 agents 之间传递数据的频率取决于目标代理人访问相关状态的频率和系统中 agents 学习的进度。

针对路网中数据的实时性和复杂性，以及上述三种方法的利弊实验分析[67]，我们采用如下策略：在系统学习的初期需要大量的数据去学习，并且信息的可靠性会比较低，这时需要鼓励 agents 自己去探索更多的行动策略，当然这些传递的数据并不做行动指导，因此最初低转移频率最佳，也可理解为形成大规模系统之前先处理好各自小

规模的系统，这样的低频转移可以认为是对系统的一种初始化过程（可设置为一段时间的低频）。随后数据逐渐可靠，增加传输频率以便加快整个系统的学习，当系统处于稳定阶段即没有需要什么数据学习时逐渐减少传输频率并采用变化最大时传输数据的方法。

（3）如何将接受的数据纳入目标代理的状态空间中？

代理之间需要将邻居传递的数据融合到自己的状态空间中做下一步的学习或是执行动作的基础，需要考虑：1）传递的数据是否已接收过；2）接收的数据是否对代理本身有积极的作用；3）如何将有积极作用的数据合并到代理本身的状态空间中。

在路网中并不需要考虑如此复杂的逻辑，路网中各 agents 的之间的决策依据 W 值的大小，接收的数据主要是影响 W 值的计算和各 agents 之间重要性的判断，并不影响各 agents 本地独立的计算，交通网络的运行不会出现严重的问题。因此对于各 agents 之间传递的数据建立数据缓冲区，将本地状态数据和传递的数据分别存储为两个数据空间，在计算 Q 值和 W 值时从两个数据空间中取相对应的值。本章利用多向传递数据的目的是为了加快系统学习节省时间，所以在考虑多 agents 合作接收数据时不能影响到最基本的单个 agent 处理数据的逻辑，远程（一跳邻居之间）数据和本地数据通过两个数据空间存储和获取。

5.4 基于 DWL 的交通灯控制方法模型

分布式城市交通灯控制系统是未来发展的主要框架模型，也将会基于分布式模型研究多个路口之间相互协作的算法来缓解重度交通流下城市交通的压力。基于分布式 W-Learning 和协作图建立多路口交通灯控制模型如图 5.4 和图 5.5 所示。

假设路网中含有 11 个路口，图 5.4 中显示每个路口都与一跳邻居路口相连接，是一种有向协作图，基于协作图原理和多向并行数据传递可实现路口之间的数据共享，包括路口交通状态 s、路口执行动作 Q 值、奖励值 r、权重值 w、协作系统数值 C。每个路口都可以是源任务也可以是目标任务。

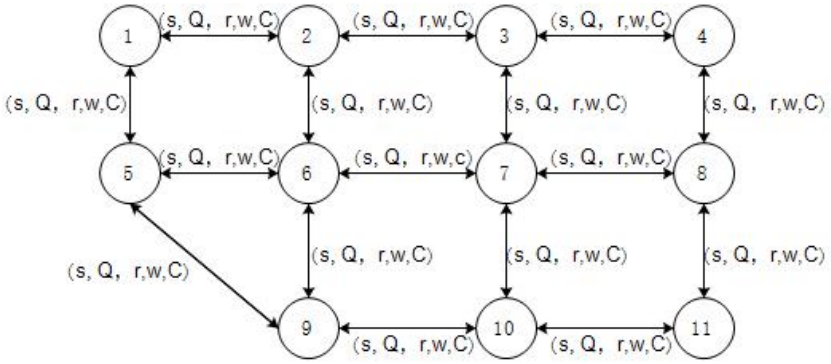

图 5.4　基于协作图的多路口多向并行传递数据

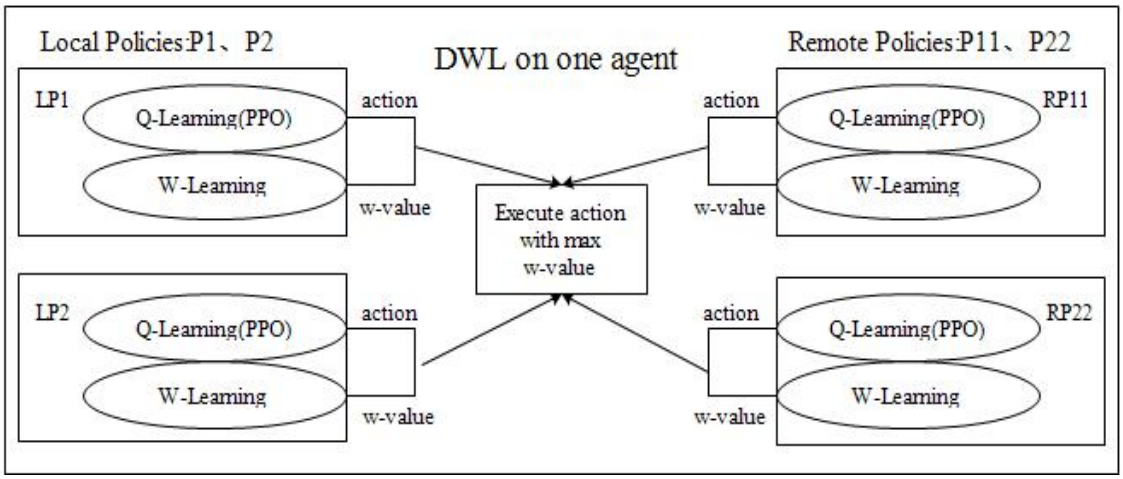

图 5.5 单路口分布式 W-Learning[66]

图 5.5 表示图 5.4 中任意单个路口的 DWL 学习，图 5.5 中仅涉及两种本地策略 $P1$ 和 $P2$，每个路口都需要执行本地策略的 Q-Learning、W-Learning 以及两种远程策略 $P11$ 和 $P22$ 的 Q-Learning、W-Learning。不同本地策略和不同远程策略可以学出不同的 W 值，路口选择最大 W 值所对应的 Q 值表示的相位动作执行即可。全局路网每个路口都采用 DWL 学习并且每个路口都基于协作图原理实现并行传输数据，路网中的各个路口相互独立又相互共享数据。

5.5 仿真实验与结果分析

为了验证基于 DWL 建模的城市交通灯控制算法的性能，将 DWL 方法和 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法分别在图 3.4 所示的具有多个路口的交通网络中进行仿真，并在轻度交通流和重度交通流下分别进行 5 次仿真实验。

5.5.1 路网环境配置

本章实验依旧采用 SUMO，实验基本环境、路网中的路口、车道、车辆类型、仿真时间等基本设置与 3.4.1 节中的完全相同，本章中仿真采集数据的路口分为 3 组：（1）图 3.4 中红色数字标记的 2、3、4、10、7 路口采集 2 次；（2）图 3.4 中红色数字标记的 19、9、2、3、6 路口采集 2 次；（3）图 3.4 中红色数字标记的 10、11、4、15、12 路口采集 1 次。

比较方法参数设置：TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法中初始化参数：折扣因子 $\gamma = 0.9$，$\varepsilon - greedy$ 策略中的 $\varepsilon = 0.01$。

每个单路口采用基于近端策略 PPO 计算，折扣因子 $\gamma = 0.9$，$\varepsilon - greedy$ 策略中 ε 训练初始为 1，随后迭代递减至 0.01。PPO 参数设置见表 3.3，网络设置见表 3.4。

因为在本文实验中涉及的车型只有 car 和 bus 两种，并且不考虑交通事故、行人、

救护车等紧急事件、自然天气灾害影响等诸多因素，为理想型仿真环境，因此为验证多策略对交通灯的影响，本章仅设计两种策略：（1）car 和 bus 的优先性一致；（2）bus 优先性高于 car，优先考虑 bus 多的排队相位。针对数据采集的路口分别设计如下策略安排：（1）所有路口都采用策略 1，全路网为单策略模式，可与 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法进行比较，属于单策略协作类型、单策略非协作类型比较；（2）部分路口采用策略 1，部分路口采用策略 2，具体路口策略分布如表 5.2 所示：

表 5.2　多路口多策略分布表

方案	路口/策略	路口/策略
1	2、4、10、7（策略 1）	3（策略 2）
2	19、9、2（策略 1）	3、6（策略 2）
3	10、4（策略 1）	4、15、12（策略 2）

采用表 5.2 所示分布表利用 DWL 分别在多路口单策略协作类型和多路口多策略协作类型下各仿真实验 5 次求取各评价参数的平均值。当选择多路口多策略仿真情况下可与 DWL 的多路口单策略协作类型比较。

本章中考虑各路口的协作系数 C，首先初始化路口协作系数，然后依据算法 5 对协作系数 C 进行学习变化。表 5.3 中路口对应协作系数以方案 1 为例说明：路口 2、4、10、7、3 对邻居远程策略协作系数分别对应为 0.2、0.4、0.5、0.6、0.8，即 C_{23}=0.2、C_{43}=0.4、C_{103}=0.5、C_{73}=0.6、$C_{32}=C_{34}=C_{37}=C_{310}$=0.8。

表 5.3　各路口协作系数分布表

方案	路口	协作系数
1	2、4、10、7、3	0.2、0.4、0.5、0.6、0.8
2	19、9、2、3、6	0.2、0.4、0.5、0.6、0.8
3	10、11、4、15、12	0.2、0.4、0.5、0.6、0.8

5.5.2 轻度交通流

利用 SUMO 产生车辆 car 和 bus，为便于分析算法在不同车流情况下有效性，设置车辆的固定产生频率，在轻度交通流环境下，car 的产生速度设置为 1 分钟 15 辆，即频率为 0.25；bus 的产生速度设置为 1 分钟 3 辆，即频率为 0.05。DWL 方法和 TCLJ

方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法交通性能评价指标的 5 次仿真结果的平均值如表 5.4 所示。ATWT 指标对比如图 5.6 所示，AJWT 指标对比如图 5.7 所示。本章所有图表中（DWL 单）表示仿真环境采用的是多路口单策略协作模式，（DWL 多）表示仿真环境采用的是多路口多策略协作模式。

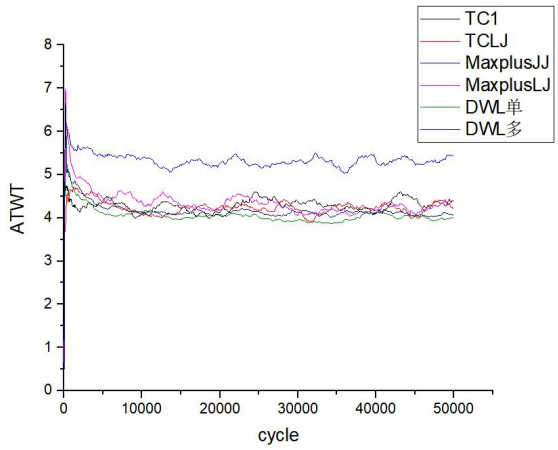

图 5.6 轻度交通流情况下 DWL 与其他控制方法的 ATWT 对比

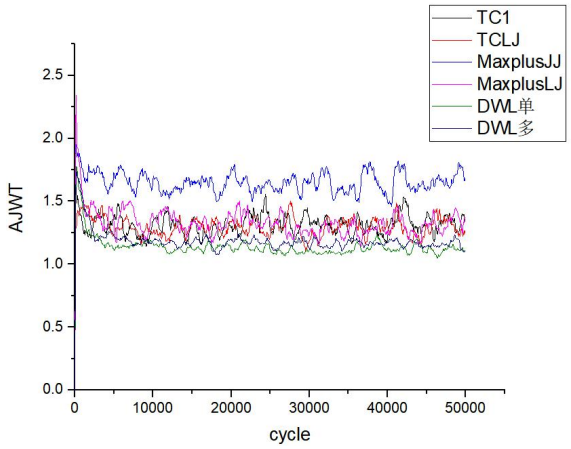

图 5.7 轻度交通流情况下 DWL 与其他控制方法的 AJWT 对比

从表 5.4 和图 5.6、5.7 中可以看出在轻度交通流环境下 DWL 方法和其他方法相比在每个指标上都是略优于其他方法，其中 DWL 多策略仿真效果没有 DWL 单策略仿真效果好，本文认为是考虑了 bus 的优先性使得 car 的等待时间延长，实验中 car 的数量远多于 bus，导致整体指标效果没有单策略的好。其中也反应了多策略对交通灯相位选择的影响，从 ATWT 和 AJWT 曲线图看出基于 DWL 的方法波动浮动小且更能平稳的控制交通灯，总体而言上述六种方法在轻度交通流环境下都可以有效的控制交通灯，避免发生交通拥堵。

表 5.4　轻度交通流情况下 DWL 和其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真 时间	预期仿真 时间
TC1	4.2853	1.3133	179686	50000	50000
TCLJ	4.2331	1.2960	179551	50000	50000
MaxplusJJ	5.3241	1.6494	179462	50000	50000
MaxplusLJ	4.3501	1.3307	179765	50000	50000
DWL 单	4.0541	1.1382	194380	50000	50000
DWL 多	4.1570	1.1839	189634	50000	50000

5.5.3 重度交通流

重度交通流情况下，car 的产生速度设置为 1 分钟 21 辆，即频率为 0.35，bus 的产生速度设置为 1 分钟 3 辆，即频率为 0.05。考虑到公交在实际情况下都是按点按班次发车，所以公交车 bus 的产生频率不发生变化，仅是增加汽车 car 的产生频率。在重度交通流情况下，DWL 方法和 TCLJ 方法、TC1 方法、MaxplusLJ 方法、MaxplusJJ 方法进行仿真比较，交通性能评价指标的 5 次仿真结果的平均值如表 5.5 所示。ATWT 指标对比如图 5.8 所示，AJWT 指标对比如图 5.9 所示。

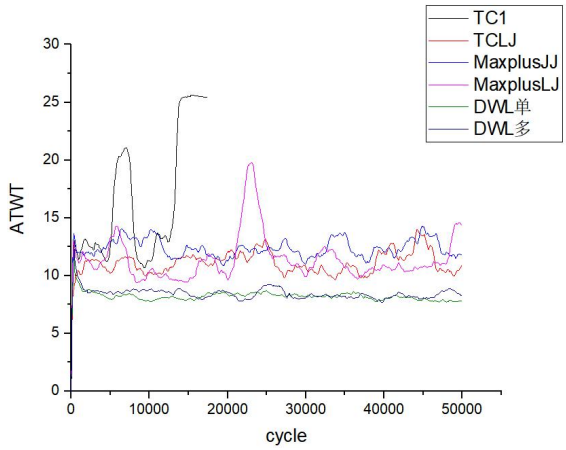

图 5.8　重度交通流情况下 DWL 与其他控制方法的 ATWT 对比

从表 5.5 和图 5.8、5.9 可以看出在重度交通流环境下 TC1 方法已经无法控制交通灯导致车辆严重拥堵、路网瘫痪无法正常运作了。其他三个方法虽然可以有效的控制交通灯使得路网中的车辆正常运行，但是波动幅度很大，不能平稳的控制车流量。基于 DWL 的方法无论是在单策略和多策略的情况下都可以有效并平稳的控制交通灯和车流，并表现出比其他方法更加优秀，其中基于多策略的仿真效果没有单策略好，并

且在中间有一段时间有较明显的波动，多策略中包含了 bus 的优先性，所以本文认为是 bus 优先策略导致整体等待时间上涨，因为 car 滞停了并且数量远超 bus。

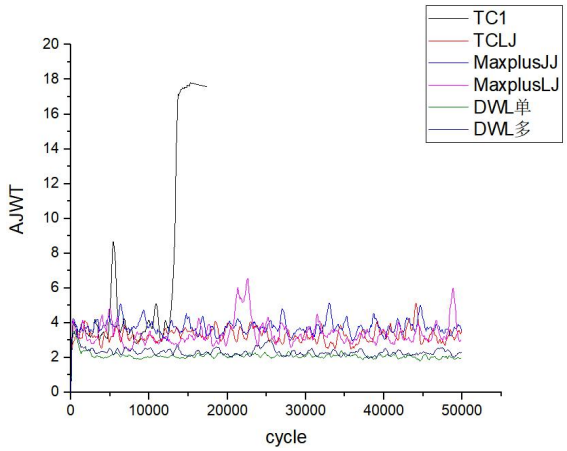

图 5.9 重度交通流情况下 DWL 与其他控制方法的 AJWT 对比

表 5.5 重度交通流情况下 DWL 和其他控制方法 5 次仿真交通性能评价指标平均值

评价参数 / 方法	ATWT (cycle)	AJWT (cycle)	TAV （辆）	实际仿真时间	预期仿真时间
TC1	16.3977	7.1189	64871	17408	50000
TCLJ	11.0840	3.2903	239805	50000	50000
MaxplusJJ	12.3432	3.7755	239847	50000	50000
MaxplusLJ	11.4043	3.4207	239903	50000	50000
DWL 单	8.1677	2.0915	320459	50000	50000
DWL 多	8.4271	2.3413	308912	50000	50000

在轻度交通流和重度交通流中利用了多策略和单策略两种模式，并且在仿真中反应出多策略并没有单策略表现的优秀，为验证单策略和多策略造成的具体影响，分别在上述轻度交通流环境和重度交通流环境仿真实验过程中单独获取了 car 和 bus 的 AJWT、ATWT 等数据，分别求取轻度交通流和重度交通流环境下 5 次仿真数据的平均值，获得如图 5.10 所示对比图。

从图 5.10 和图 5.6、图 5.7、图 5.8、图 5.9 的比较中可以看出，虽然在轻度交通流和重度交通流环境下实验仿真得到的各项指标多策略模式下均劣于单策略模式，但是基于公交车优先的多策略却改善了公交车的运行情况，说明基于某类型车辆建立多策略时可适当改善该类型车辆在路网中的运行情况。如果在多策略中包含救护车、消防车、警车等类型车辆的优先性，那么在城市交通灯控制中不用人为的干预都可满足

相关类型车辆的先行，路口可自行控制交通灯处理紧急事件。

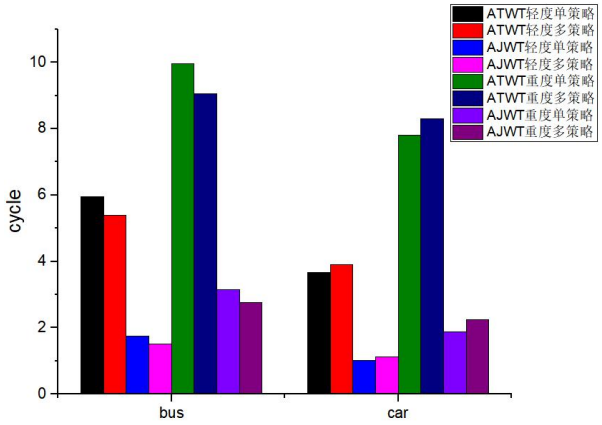

图 5.10　car 与 bus 在单策略和多策略模式下指标对比

本章算法和第四章算法比较虽然各项指标均没有优势，但是本章算法中的多策略模式可优化策略中涉及到的车型的运行情况，多策略协作是未来城市交通控制发展的趋势之一，但是第四章的方法不行，不能优化某类车型的运行情况，不能解决紧急事件优先处理的相位变化，只能整体的分析。

5.6 本章总结

本章开始详细介绍了 DWL 中如何体现多路口多策略之间的协作，涉及到了 DWL 的具体算法及协作系数 C 的自学习算法，DWL 利用 W 值选择最终路口执行的相位动作。为了加快学习的速度和实现路口之间数据的共享，本章基于协作图原理的多向并行数据传递完成，然后介绍了基于 DWL 的交通灯控制模型。最后在实验部分设计单策略和多策略仿真环境，通过实验表明了基于 DWL 的方法在重度交通流环境下明显优于其他方法并说明了多策略在交通灯控制方法中可根据策略设计优化某类车型在路网中运行情况的优点。

第六章 总结与展望

6.1 本文总结

随着科技的不断发展，各国对于城市交通灯控制算法的研究也是不断深入，利用深度强化学习来解决城市交通灯控制算法已经是研究的热点方向。

目前算法中解决城市交通灯控制主要利用蚁群算法、强化学习算法、深度强化学习算法等，并且主要是面向单路口解决交通灯控制，很少涉及到路网中路口之间的协同控制。本文主要目的是提升单路口交通灯控制能力并利用路网中路口间的协同控制机理解决复杂路网中多路口交通灯协同控制问题。本文涉及到单路口、多路口、多路口多策略三方面，具体总结如下：

（1）在单路口交通灯控制问题中利用 Open AI 团队提出的近端策略 PPO 框架，通过考虑历史策略对当前策略的影响程度，即通过比值的形式来影响当前的学习率和深度学习的采样。对于单路口当前时刻 t 的交通执行相位选择不仅和当前路况有密切关系同样和历史策略有关。在当前策略和历史策略做比值计算时发现超域问题，并对其比值进行裁剪划定范围，变为置信区间内的比值计算，解决比值超域问题。

（2）在多路口协同交通灯控制问题中利用 Google Deep Mind 提出的基于分布式深度循环 Q 网络框架，通过考虑当前路口自身历史状态、历史相位以及其一跳邻居路口上一时刻各路口的交通状态、相位动作来综合分析当前路口在当前时刻应该执行的相位动作。利用 MLP 神经网络对当前路口和一跳邻居路口做输入处理，利用 LSTM 神经网络对当前路口和一跳邻居路口做隐藏层状态处理，计算得到历史建议值和邻接路口建议值通过 MLP 神经网络做最终的执行相位选择。

（3）在多路口多策略协同交通灯控制问题中利用分布式 W-Learning 框架，单个路口的相位计算按照第三章中的计算方式，同时增加计算了 W 值，即重要性权重，用来衡量本地策略和远程策略对于路口相位选择的重要性，引入协作系数 C 使路口能对远程策略的相位动作建议给予不同程度的重要性衡量。为了加快系统的学习采用了多向并行传递数据，路网中的各个路口既可以是源任务也可以是目标任务，互相传递数据加快学习，不会影响到分布式 W-Learning 对于路口的 Q 值和 W 值计算，最终路口执行的相位动作是最大 W 值所对应的 Q 值。

6.2 工作展望

本文针对路网中的单路口、多路口协同、多路口多策略协同三方面进行了改进，并取得了较好的性能，但还是存在以下问题可改进：

（1）单路口问题中交通灯的控制问题都是实时计算相位，但是却没有对未来相位动作的预测和未来交通状态的预测。目前单路口的研究是基于单步实现的，多步的实现涉及到奖励值的预估、路口车辆交通流、交通状态的预估等复杂逻辑，研究的目标是预测出路口多步的交通流、交通状态等形成一定程度的轨迹线，通过实时的数据再去不断地修正未来的轨迹线使得未来的预测可以更加准确。如何实现多步单路口交通相位控制的预测是需要解决的问题。

（2）在单路口、多路口协同、多路口多策略协同三方面问题中都仅是考虑了机动车辆，属于一种理想环境并且机动车辆种类数较少，现实中路网环境很复杂，更是包含了非机动车和行人在路口对机动车造成的影响，在本文的算法中都未涉及到行人和非机动车的影响。如何增加行人和非机动在路口对交通灯控制影响的逻辑计算是需要解决的问题。

（3）多路口多策略算法中，在现实路网环境中需要涉及到多种策略，如何选择策略以及分布策略是现实中解决的重点问题，当出现众多策略并在多路口间互传数据时导致存储空间数据爆炸，并且众多策略的运算也将增加单路口运算算法的时间和空间负担，如何面对复杂现实路网的多策略优化单路口交通灯控制算法及解决大数据条件下互传数据的实效性和选择有效数据等问题。

参考文献

[1] Li L, Wen D, Yao D. A survey of traffic control with vehicular communications[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(1): 425-432.

[2] Balaji P G, German X, Srinivasan D. Urban traffic signal control using reinforcement learning agents[J]. IET Intelligent Transport Systems, 2010, 4(3): 177-188.

[3] Abdoos M, Mozayani N, Bazzan A L C. Holonic multi-agent system for traffic signals control[J]. Engineering Applications of Artificial Intelligence, 2013, 26(5-6): 1575-1587.

[4] Li L, Wen D. Parallel systems for traffic control: A rethinking[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1179-1182.

[5] El-Tantawy S, Abdulhai B, Abdelgawad H. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(3): 1140-1150.

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

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

[8] Sutton R S, Barto A G. Introduction to reinforcement learning[M]. Cambridge: MIT press, 1998.

[9] Prashanth L A, Bhatnagar S. Reinforcement learning with function approximation for traffic signal control[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(2): 412-421.

[10] Duggan M, Flesk K, Duggan J, et al. A reinforcement learning approach for dynamic selection of virtual machines in cloud data centres[C]//2016 Sixth International Conference on Innovative Computing Technology (INTECH). IEEE, 2016: 92-97.

[11] Duggan M, Duggan J, Howley E, et al. An autonomous network aware vm migration strategy in cloud data centres[C]//2016 International Conference on Cloud and Autonomic Computing (ICCAC). IEEE, 2016: 24-32.

[12] 赵冬斌, 邵坤, 朱圆恒,等. 深度强化学习综述:兼论计算机围棋的发展[J]. 控制理论与应用, 2016, 33(6):701-717.

[13] Sutton R. RS & Barto, Reinforcement learning: An introduction[J]. 1998.

[14] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of artificial intelligence research, 1996, 4: 237-285.

[15] Rummery G A, Niranjan M. On-line Q-learning using connectionist systems[M]. Cambridge, England: University of Cambridge, Department of Engineering, 1994.

[16] Wiering M A. Explorations in efficient reinforcement learning[D]. University of Amsterdam, 1999.

[17] Wiering M A. Multi-agent reinforcement learning for traffic light control[C]//Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000). 2000: 1151-1158.

[18] Steingrover M, Schouten R, Peelen S, et al. Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion[C]//BNAIC. 2005: 216-223.

[19] Kuyer L, Bakker B, Whiteson S. Multiagent reinforcement learning and coordination for urban traffic control using coordination graphs and max-plus[D]. MS Thesis, 2007.

[20] Kuyer L, Whiteson S, Bakker B, et al. Multiagent reinforcement learning for urban traffic control using coordination graphs[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2008: 656-671.

[21] Li L, Lv Y, Wang F Y. Traffic signal timing via deep reinforcement learning[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(3): 247-254.

[22] Mousavi S S, Schukat M, Howley E. Traffic light control using deep policy-gradient and value-function-based reinforcement learning[J]. IET Intelligent Transport Systems, 2017, 11(7): 417-423.

[23] Rosyadi A R, Wirayuda T A B, Al-Faraby S. Intelligent traffic light control using collaborative Q-Learning algorithms[C]//2016 4th International Conference on Information and Communication Technology (ICoICT). IEEE, 2016: 1-6.

[24] Shen L, Liu R, Yao Z, et al. Development of dynamic platoon dispersion models for predictive traffic signal control[J]. IEEE Transactions on Intelligent Transportation Systems, 2018 (99): 1-10.

[25] Vidhate D A, Kulkarni P. Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control[C]//2017 1st International Conference on Intelligent Systems and Information Management (ICISIM). IEEE, 2017: 325-331.

[26] Chu T, Wang J. Traffic signal control by distributed Reinforcement Learning with min-sum communication[C]//2017 American Control Conference (ACC). IEEE, 2017: 5095-5100.

[27] Liang X, Du X, Wang G, et al. Deep reinforcement learning for traffic light control in vehicular networks[J]. arXiv preprint arXiv:1803.11115, 2018.

[28] Jin J, Ma X. Hierarchical multi-agent control of traffic lights based on collective learning[J]. Engineering Applications of Artificial Intelligence, 2018, 68: 236-248.

[29] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative

multi-agent learning based on team reward[C]//Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018: 2085-2087.

[30] Wan C H, Hwang M C. Value-based deep reinforcement learning for adaptive isolated intersection signal control[J]. IET Intelligent Transport Systems, 2018, 12(9): 1005-1010.

[31] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning[C]//Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018: 443-451.

[32] Dai Y, Duan H, Li Z, et al. A multi-agent coordinate model for urban traffic signal control[C]//IEEE ICCA 2010. IEEE, 2010: 1882-1887.

[33] Liu W, Qin G, He Y, et al. Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering[J]. IEEE Transactions on Vehicular Technology, 2017, 66(10): 8667-8681.

[34] Tahifa M, Boumhidi J, Yahyaouy A. Swarm reinforcement learning for traffic signal control based on cooperative multi-agent framework[C]//2015 Intelligent Systems and Computer Vision (ISCV). IEEE, 2015: 1-6.

[35] Dusparic I, Monteil J, Cahill V. Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning[C]//2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016: 2065-2070.

[36] 温凯歌, 曲仕茹, 张玉梅. 城市单路口信号多相位自适应控制模型[J]. 系统仿真学报, 2009, 21(10): 3066-3070.

[37] 温凯歌, 杨照辉. 基于 CMAC 强化学习的交叉口信号控制[J]. 计算机工程, 2011, 37(17):152-154.

[38] 何兆成, 佘锡伟, 杨文臣, 等.结合 Q 学习和模糊逻辑的单路口交通信号自学习控制方法[J]. 计算机应用研究, 2011, 28(1): 199-202.

[39] 黄艳国, 唐军, 许伦辉. 基于 Agent 的城市道路交通信号控制方法[J]. 公路交通科技(学术版), 2009, 26(10): 126-129.

[40] 李春贵, 阳树洪, 王萌,等. 基于 SARSA(λ)算法的单路口交通信号学习控制[J]. 广西科技大学学报, 2008, 19(2):10-14.

[41] 伦立宝. 基于强化学习的城市交通信号控制方法研究[D]. 西安电子科技大学, 2013.

[42] 俞灏. 动态交通条件下交通诱导与信号控制协同研究[D]. 东南大学, 2016.

[43] 夏新海. 面向城市自适应交通信号控制的强化学习方法研究[D]. 广州: 华南理工大学, 2013.

[44] 王莹多. 基于深度强化学习的路口自适应控制[D].大连理工大学, 2017.

[45] 邱建东, 解小平, 汤旻安, 等. 基于车流量的智能交通信号优化控制研究[J]. 计算机应用与软件, 2018, 35(1): 92-96.

[46] 刘成健, 罗杰. 基于参数融合的 Q 学习交通信号控制方法[J]. 计算机技术与发展, 2018, 28(11): 48-51.

[47] 胡海涛, 罗杰. 基于相位差的子区交通信号协调优化控制[J]. 计算机技术与发展, 2018 (2018 年 06): 151-155.

[48] 王安麟, 孙晓龙, 钟馥声. 一种基于通行优先度规则的城市交通信号自组织控制方法[J]. 重庆交通大学学报 (自然科学版), 2018, 37(02): 96-101.

[49] 项俊平. 城市道路交通信号区域均衡控制方法及应用研究[D]. 中国科学技术大学, 2018.

[50] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.

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

[52] Hausknecht M, Stone P. Deep recurrent q-learning for partially observable mdps[C]//2015 AAAI Fall Symposium Series. 2015.

[53] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436.

[54] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 6645-6649.

[55] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. 2016: 1928-1937.

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

[57] Horgan D, Quan J, Budden D, et al. Distributed prioritized experience replay[J]. arXiv preprint arXiv:1803.00933, 2018.

[58] Arjona-Medina J A, Gillhofer M, Widrich M, et al. Rudder: Return decomposition for delayed rewards[J]. arXiv preprint arXiv:1806.07857, 2018.

[59] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[60] Ha-li P, Ke D. An Intersection Signal Control Method Based on Deep Reinforcement Learning[C]//2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE, 2017: 344-348.

[61] Van der Pol E, Oliehoek F A. Coordinated deep reinforcement learners for traffic light control[J]. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016), 2016.

[62] Abdoos M, Mozayani N, Bazzan A L C. Hierarchical control of traffic signals using Q-learning with tile coding[J]. Applied intelligence, 2014, 40(2): 201-213.

[63] Proper S, Tadepalli P.Solving multiagent assignment markov decision processes[C]//Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 2009: 681-688.

[64] Kok J R, Vlassis N. Using the max-plus algorithm for multiagent decision making in coordination graphs[C]//Robot Soccer World Cup. Springer, Berlin, Heidelberg, 2005: 1-12.

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

[66] Dusparic I, Cahill V. Autonomic multi-policy optimization in pervasive systems: Overview and evaluation[J]. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2012, 7(1): 11.

[67] Taylor A, Duparic I, Galván-López E, et al. Transfer learning in multi-agent systems through parallel transfer[J]. 2013.

[68] Dusparic I, Monteil J, Cahill V. Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning[C]//2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016: 2065-2070.

[69] Humphrys M. W-learning: Competition among selfish Q-learners[R]. University of Cambridge, 1995.

[70] Humphrys M. Action selection methods using reinforcement learning[J]. From Animals to Animats, 1996, 4: 135-144.

[71] Dusparic I, Cahill V. Using distributed w-learning for multi-policy optimization in decentralized autonomic systems[C]//Proceedings of the 6th international conference on Autonomic computing. ACM, 2009: 63-64.

[72] Taylor A, Dusparic I, Galván-López E, et al. Accelerating learning in multi-objective systems through transfer learning[C]//2014 International Joint Conference on Neural Networks (IJCNN). IEEE, 2014: 2298-2305.

[73] Marco Wiering, Jelle van Veenen, Jilles Vreeken et al. Intelligent Traffic Light Control. Technical report, Department of Information and Computing Sciences, University Utrecht. 2004. 1-30.

致谢

时光飞逝，转眼间三年的研究生生活就要结束了，回顾过去的三年时光，收获颇多。在这三年的时光中学到很多知识、遇到很多事，同样有很多的人要去感恩，在此衷心的感谢我的老师同学和亲人们，谢谢你们对我的鼓励和支持！

首先由衷的感谢我的导师方敏教授，在研究生的三年时光中，她悉心教导、耐心督促并给予我很多鼓励，赋予了我优越的学习和科研环境。在确定研究方向时她非常认真负责的查阅了大量国内外文献并结合生活实际列举了 3 个研究方向让我选择，在我选择确定方向后定期的认真听取我的工作汇报并对汇报内容中不恰当的部分做详细的探讨和讲解，然后会对下一步的工作指出方向，让我受益匪浅。方老师严谨的治学态度、敏锐的洞察力和渊博的专业知识、精益求精的工作作风也深深影响着我，让我受益终身！正是方老师的悉心教导让我顺利完成了课题的研究和论文的修改，在此谨向方老师致以诚挚的谢意和崇高的敬意！

其次非常感谢实验室的小伙伴们，感谢你们对我的支持和帮助，感谢平日里在课题研究方面上无私的帮助和辅导，在此向实验室的师兄师姐师弟师妹们表示由衷的感谢！

同时感谢我的家人在读研期间对我的关照和鼓励，坚强的后盾和未来的期望是我求学路上的动力，让我能专心的投入学业中，在学业完成之际向我的家人表示深深的感谢！

最后我要感谢参与我论文评审和答辩的各位老师，感谢你们在百忙之中评阅我的论文。今后的工作中我将加倍努力不负老师同学家人们的期望，愿你们一生幸福安康！

作者简介

1. 基本情况

闫呈祥，男，甘肃通渭人，1993 年 6 月出生，西安电子科技大学计算机科学与技术学院计算机技术专业 2016 级硕士研究生。

2. 教育背景

2012.09～2016.06 中国矿业大学，本科，专业：计算机科学与技术
2016.08～ 西安电子科技大学，硕士研究生，专业：计算机技术

3. 攻读硕士学位期间的研究成果

3.1 参与科研项目及获奖

[1] 西安市科技项目, 网络可靠性分析方法及评价技术, 2017-2020, 未结题, 主要完成人.