Modeling and Simulation 建模与仿真, 2024, 13(1), 136-150
Published Online January 2024 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2024.131014

Hans汉斯

基于改进的深度强化学习策略的交通信号控制

徐晴晴，韩天立，胡林治

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2023年11月21日；录用日期：2023年12月31日；发布日期：2024年1月10日

摘　要

交叉口的交通信号控制是治理交通拥堵的重要组成部分，而现有的交通信号大多采用循环控制，效率低下且会造成长时间的车辆延迟和能量浪费。针对此问题，采用深度强化学习算法与环境之间进行互动来学习最佳策略。具体地，在智能体学习的初始阶段，创建了一个动作价值评估网络，以增加智能体的学习经验，帮助智能体更快的掌握缓解交通拥堵的技能。提出的模型基于双决斗深度Q网络(Double Dueling Deep Q-Network, 3DQN)算法，车辆的位置信息作为模型的输入，交叉口的四种相位为动作空间，执行动作前后的累积等待时间差被定义为奖励。在城市交通模拟器(Simulation Of Urban Mobility, SUMO)中对模型进行评估。实验结果表明，提出的模型在累积奖励方面相较于DQN、Double DQN、Dueling DQN、3DQN分别增加了58.9%、51.9%、51.3%、48%，证明改进的学习策略可以有效地提升各项交通指标。

关键词

深度强化学习，交通信号控制，SUMO，智能交通，机器学习

Traffic Signal Control Based on Improved Deep Reinforcement Learning Strategy

Qingqing Xu, Tianli Han, Linzhi Hu

School of Optoelectronic Information and Computer Engineering, Shanghai University of Science and Technology, Shanghai

Received: Nov. 21st, 2023; accepted: Dec. 31st, 2023; published: Jan. 10th, 2024

Abstract

Traffic signal control at intersections plays a crucial role in managing traffic congestion. However, the conventional cycle control used in existing traffic signals is inefficient and often leads to significant vehicle delays and energy wastage. To address this issue, a deep reinforcement learning

文章引用: 徐晴晴, 韩天立, 胡林治. 基于改进的深度强化学习策略的交通信号控制[J]. 建模与仿真, 2024, 13(1): 136-150. DOI: 10.12677/mos.2024.131014

algorithm was employed to interact with the environment and learn the optimal control strategy. In the initial stages of the agent's learning, an action-value evaluation network was established to enhance the agent's learning experience and facilitate the rapid acquisition of skills for mitigating traffic congestion. The proposed model was based on the double dueling deep Q-Network (3DQN) algorithm, utilizing vehicle position information as input and the four phases of the intersection as the action space. The reward was defined as the difference in cumulative waiting time before and after executing an action. The model's performance was evaluated using the simulation of urban mobility (SUMO) city traffic simulator. Experimental results demonstrated that the proposed model achieves a substantial increase in cumulative rewards, surpassing DQN, double DQN, dueling DQN, and 3DQN by 58.9%, 51.9%, 51.3%, and 48%, respectively. These findings validated the effectiveness of the improved learning strategy in enhancing various traffic indicators.

Keywords

Deep Reinforcement Learning, Traffic Signal Control, SUMO, Intelligent Transportation, Machine Learning

Copyright © 2024 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

随着经济和社会的发展，机动车数量和相应的出行需求不断增加。我们在享受汽车带来便利的同时，也在承担其带来的后果，最为明显的就是造成了严重的交通拥堵。交通拥堵会导致交通事故频繁发生、车辆尾气排放量不断增加、社会经济损失日益上升，所以，有效的交通管理和控制对于缓解逐渐恶化的交通状况至关重要。解决上述问题通常有两种可能的解决方案：第一种是直接改造道路基础设施，例如扩大道路面积，增加道路通行能力；第二种是优化交通信号控制(Traffic Signal Control, TSC)系统。显然，第一种方案不仅要浪费大量的人力、物力、财力，而且还会导致短期内道路通行能力变差。第二种方案可以直接在现有的架构上实现，在不破坏当前的道路结构的前提下，缓解交通拥堵问题。因此，TSC 系统的优化是提高交通效率的重要方法。

长期以来，许多研究人员在 TSC 系统的优化方面进行了大量的研究，并提出了许多 TSC 系统。大致可以分为三大类：固定配时系统，使用 Webster 公式[1]确定绿灯的持续时间；驱动控制系统，如 MOVA [2]、LHVORA [3]和 SOS [4]都是典型的驱动控制系统；自适应控制系统，例如 SCATS [5]和 SCOOT [6]。但是，随着交通状况的复杂度升高，这些传统的 TSC 系统不能实时适应交通状况从而调整交通信号，导致交通效率低下。

到了 20 世纪 90 年代，人工智能方法开始用于控制交通信号，其中强化学习(Reinforcement Learning, RL)作为一种实现交通信号控制的方法在学术界得到了广泛的应用。但是，早期强化学习技术仅限于表格 Q-learning 算法，通常使用线性函数来估计 Q 值，因此仅适用于小规模的状态空间，由于交通道路系统的复杂性导致状态空间也异常复杂，此时就不再适合使用表格 Q-learning。随着近年来深度学习(Deep Learning, DL)的快速发展，研究人员开始将深度学习与强化学习相结合，称为深度强化学习(Deep Reinforcement Learning, DRL)，通过使用神经网络来近似给定状态的 Q 值来选择最佳动作，成功解决了交通状态空间复杂的问题。近年来，将 DRL 应用到交通信号控制问题上也取得了很大的进展。

比如，Wang 等人[7]提出了一种基于 3DQN 的交通信号控制方法，使用了基于高分辨率事件的数据，该方法以两种常用的交通信号控制策略为基准进行对照实验，即固定时间控制策略和驱动控制策略，实验证明所提出的基于事件数据的状态表示优于一般状态表示。Luo 等人[8]提出了一种自适应道路划分策略和基于深度 Q 网络(Deep Q Network, DQN)的改进神经网络模型。提出了一种基于 Fibonacci 序列的自适应道路划分方法，但是收敛速度还需要改善，而且并没有使用更加先进的深度强化学习算法。唐慕尧等人[9]将传统 DQN 与长短期记忆网络(Long Short-Term Memory, LSTM)相结合，使用 LSTM 预测未来交通状态，与当前状态串联作为传统 DQN 的输入。这种结合方法使得状态空间维度大大增加，而且使用合成数据集进行预测并不准确，反而会使训练结果很难收敛。任安妮等人[10]提出了一种基于注意力机制的 DRL 交通信号控制算法，使得神经网络去关注重要的输入状态，但是引入的注意力机制大大增加了算法的复杂度。

可以看到，利用 DRL 进行交通信号控制都取得了很好的效果，有效的降低了交叉口车辆累计等待时间、车辆排队长度等指标。基于此，本文利用 DQN 训练智能体进行交通信号控制，但是为了提高学习速度和价值函数的估计精度，对 DQN 进行了一些改进，包括双重网络结构、经验池抽样策略和优势函数。学习策略的选择也是影响学习效果的重要因素，许多教育工作者提倡从典型的经验中学习，然后通过学习不同的经验来提炼知识。从人工智能的领域来看，一些特殊的样本确实可以加快学习速度，样本的全面性可以减少过拟合，提高模型的准确性。但是大多数先前的研究在训练的初始阶段并没有给智能体提供丰富的经验样本，使智能体不能学习到正确的控制策略，导致在初始阶段由于经验和探索机制不足评价指标不稳定的问题。除此之外，本文根据建模的交通环境重新对状态空间、动作空间和奖励函数进行了精心设计，将会进一步提升模型的性能和适用性。综上所述，本文的主要贡献如下：

a) 对 DQN 算法进行一些改进，包括双重网络结构、经验池抽样策略和优势函数，有效的解决了 Q 值的高估问题，从而使得模型快速收敛；

b) 创建了一个动作价值评估网络，在训练初始阶段，该网络用于增加 3DQN 的经验，并帮助智能体更快的掌握缓解交通拥堵的技能，解决了由于初始阶段经验不足导致的非平稳性问题；

c) 在 SUMO 建立了一个单交叉口模型，模仿现实世界的车流量和交通规则，以验证所提模型的正确性和有效性，实验结果证明，该模型优于传统的固定配时算法和现有的流行方法。

2. 研究背景

2.1. 强化学习

强化学习的本质是根据与环境之间的互动学习最佳策略。具体的，RL 智能体(Agent)与环境交互，在观察到执行动作的结果后，通过学习改变动作，以此响应获得的奖励。强调在与环境的交互中学习，利用评价性的反馈信号来实现决策的优化。

图 1 为强化学习的工作循环过程，在时间步 t 时，智能体观察到环境的状态 s_t，根据状态 s_t 执行动作 a_t 与环境交互，当智能体执行动作后，环境基于所选动作转换到新状态 s_{t+1}，同时向智能体提供奖励 r_t 作为反馈，整个过程可以用四元组 $\langle s_t, a_t, r_t, s_{t+1}\rangle$ 表示。策略 π 由一系列动作组成，RL 任务通常使用价值函数来表示在一个状态下执行一个策略取得的累积预期回报，价值函数包括动作状态价值函数和状态价值函数。在状态 s 下执行动作 a 的累积预期回报被定义为动作状态价值函数 $Q^{\pi}(s,a)$：

$$\begin{aligned} Q^{\pi}(s,a) &= \mathrm{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s_t = s, a_t = a, \pi\right] \\ &= \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a, \pi\right] \end{aligned} \tag{1}$$

其中 γ 为折扣系数，$\gamma \in [0,1)$，表示当前的奖励比未来的奖励更有价值。

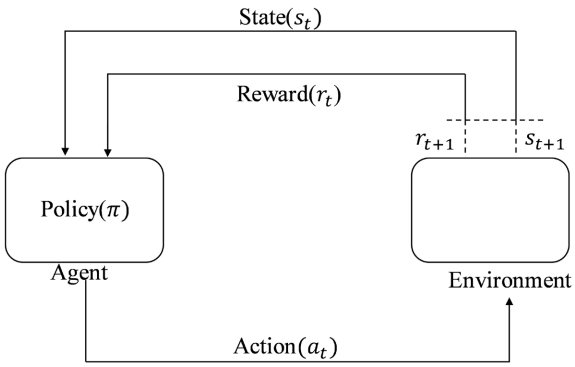

Figure 1. The work cycle of reinforcement learning
图 1. 强化学习的工作循环过程

智能体的目标是从初始状态开始学习一种最优控制策略，以最大化累积预期回报。如果智能体已知后续状态的最优 Q 值，那么最优策略 π^*将只选择使得累积预期回报最高的动作。因此，根据后续状态的最优 Q 值计算最优 $Q^{\pi^*}(s,a)$ 可以使用最优贝尔曼方程：

$$Q^{\pi^*}(s,a) = \mathrm{E}_{s_{t+1}}\left[r_t + \gamma \max_{a_{t+1}\in \mathrm{A}} Q^{\pi^*}(s_{t+1}, a_{t+1}) \mid s, a \right] \tag{2}$$

其中，最优策略 π^*为：

$$\pi^* = \arg\max_{\pi} Q^{\pi}(s,a) \tag{3}$$

状态价值函数 $V^{\pi}(s)$表示从状态 s 开始，智能体获得的累积预期回报，定义为：

$$V^{\pi}(s) = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right] \tag{4}$$

在传统的 RL 算法中，例如 Q-learning 算法，使用 Q 表格存储每个状态下每个动作的最优动作状态价值函数 $Q^{\pi^*}(s,a)$，但是当环境变得非常复杂时，很容易导致状态空间或动作空间剧增，产生维数灾难。为了解决这个问题，很多研究在 RL 中引入 DL，利用 DNN 近似 RL 中的动作状态价值函数，抽取高维数据的抽象特征，以实现降维。这种将深度学习和强化学习结合起来的方法就称为 DRL。

2.2. 深度强化学习

DRL 将深度学习的特征表示能力与强化学习的决策能力相结合，从而实现强大的端到端学习控制能力，DRL 在许多需要感知高维输入和做出最佳或接近最佳决策的任务中取得了实质性进展[11]。Mnih 等人[12]最先提出了深度 Q 网络(Deep Q-Network, DQN)，环境的原始图像作为 DQN 的状态输入，使用卷积神经网络(Convolutional Neural Network, CNN)估计 Q 值，DQN 估计的近似目标值为：

$$y_t = r_t + \gamma \max_{a} Q(s_{t+1}, a; \theta) \tag{5}$$

其中 θ 表示神经网络的参数。文献[12]的另一个贡献是运用了目标网络和经验回放两种技术，从而稳定 CNN 的学习过程。

由于在每次迭代的过程中，DQN 的参数 θ 都会更新，使得相邻时间步长的 Q 值由具有不同参数的同一网络获得，这可能会导致输出不稳定等问题。为了解决这个问题，文献[12]引入了与主网络结构相同、

但参数不同的目标网络。在训练刚开始时，主网络和目标网络使用相同的参数。主网络负责与环境交互并获取训练样本，在每个训练步骤中，目标网络输出的目标值和主网络估计的 Q 值用来更新主网络的参数。每隔几个训练步骤，目标网络的参数与主网络同步，目标网络在一段时间内保持目标值不变，有助于增强 DQN 的稳定性。此时 DQN 输出的近似目标值为：

$$y_t = r_t + \gamma \max_a Q\left(s_{t+1}, a; \theta^-\right) \tag{6}$$

其中 θ^-是目标网络的参数。

智能体在与环境交互的过程中会产生许多运动轨迹$\left(s_t, a_t, r_t, s_{t+1}\right)$，相邻的运动轨迹在时域上存在一定的相关性，直接使用这些运动轨迹进行训练可能会导致估计值与期望值之间的差值增大。为了避免这种时间相关性的影响，利用经验回放技术将最近一段时间的运动轨迹储存在回放存储器中，然后在训练过程中从回放存储器中随机采样批次，用于训练深度神经网络。同时，这种小批量的学习方式也大大提高了训练的效率。

在 DQN 算法中，如等式(5)和等式(6)所示，动作选择和动作评估都是使用相同的 Q 值实现的，如果在不准确或有噪声的情况下，可能会导致 Q 值的高估。为此，Van Hasselt 等人[13]将 Double Q-learning 算法[14]与 DQN 结合，构造了一个新的算法，称为 Double DQN。Double DQN 使用主网络选择动作、目标网络评估动作，动作选择和目标网络解耦，使 Q 值估计更加准确：

$$y_t = r_t + \gamma Q\left(s_{t+1}, \arg\max_{a_{t+1}} Q\left(s_{t+1}, a_{t+1}; \theta\right); \theta^-\right) \tag{7}$$

与大多数专注于改进 RL 算法的研究不同，Wang 等人[15]专注于神经网络结构的创新，提出了 Dueling 网络结构，Dueling DQN 与 DQN 的唯一区别就在于神经网络的结构不同。Dueling 网络结构可以同时估计每个动作的状态价值函数 $V^{\pi}(s)$和优势函数 $A^{\pi}\left(s,a\right)$，其中，优势函数定义为 $A^{\pi}\left(s,a\right)=Q^{\pi}\left(s,a\right)-V^{\pi}\left(s\right)$，表示在特定状态下执行每个动作的相对优势。而 Dueling DQN 的输出 $Q\left(s,a\right)$则是上述两个分支的组合：

$$Q\left(s,a;\theta\right)=V\left(s;\theta\right)+A\left(s,a;\theta\right)-\frac{1}{|A|}\sum_{a_{t+1}}A\left(s,a_{t+1};\theta\right) \tag{8}$$

研究表明，与直接使用优势函数相比，再减去所有优势值的平均值可以提高优化的稳定性。

3. 基于 3DQN 和动作价值评估网络的交通信号控制模型

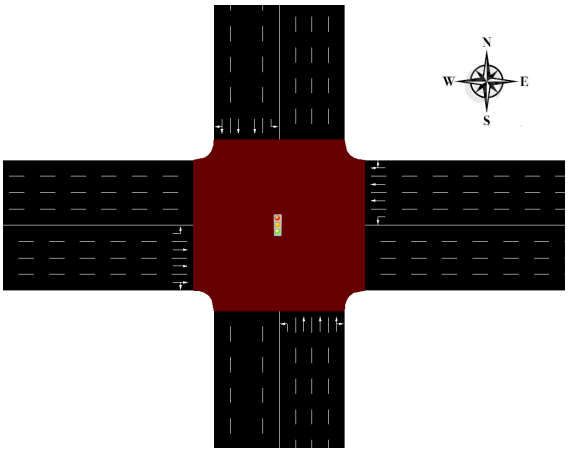

Figure 2. Four-way four-lane intersection
图 2. 4 向 4 车道交叉口

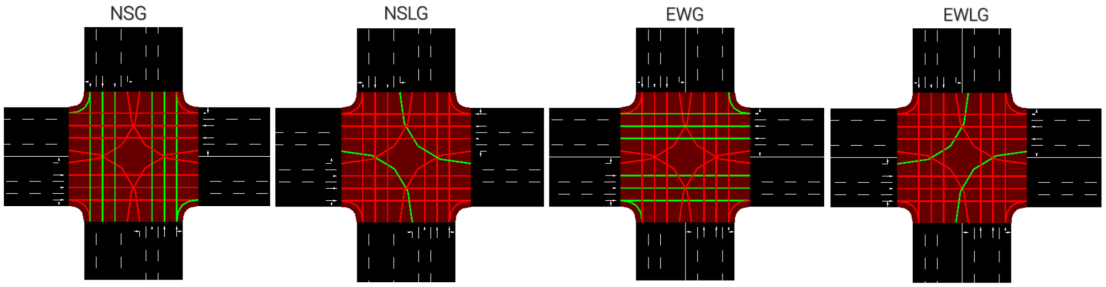

Figure 3. Four traffic phases at intersections
图 3. 交叉口 4 种交通相位

为了使用深度强化学习解决交叉口的交通信号控制问题，首先建立了如图 2 所示的交叉口模型，其中每条道路由四条车道组成，包括一条左转车道、两条直行车道、一条右转和直行共用车道。交叉口中的所有车辆均遵守图 3 所示的四种交通相位规则行驶，交通相位按照 NSG、NSLG、EWG、EWLG 的顺序循环工作，为了确保司机有足够的反应时间，在切换下一相位之前设置了黄灯时间。其中，NSG 表示南北方向直行和右转绿灯，NSLG 表示南北方向左转绿灯，EWG 表示东西方向直行和右转绿灯，EWLG 表示东西方向左转绿灯。其次，还需要定义 DRL 所需要的状态空间、动作空间、奖励函数以及控制交通信号的核心算法。

3.1. 状态空间

关于状态空间的表示方法多种多样，刘智敏等人[16]将交叉口的路段均匀分割，即将从停车线开始的车道离散为长度相同的单元格。并使用三个矩阵表示状态空间，分别是位置矩阵、速度矩阵和交通信号相位矩阵。但是这种分割方法是不合理的，它忽略了车辆与交叉口之间的距离对交叉口交通状况的影响，越靠近交叉口的车辆对交叉口的交通状况影响越大，反之则越小。Mousavi 等人[17]将 SUMO 的模拟快照作为状态输入，但是这种复杂图像的状态表示可能会导致学习过程非常缓慢，而且不一定会带来显著的性能提升。

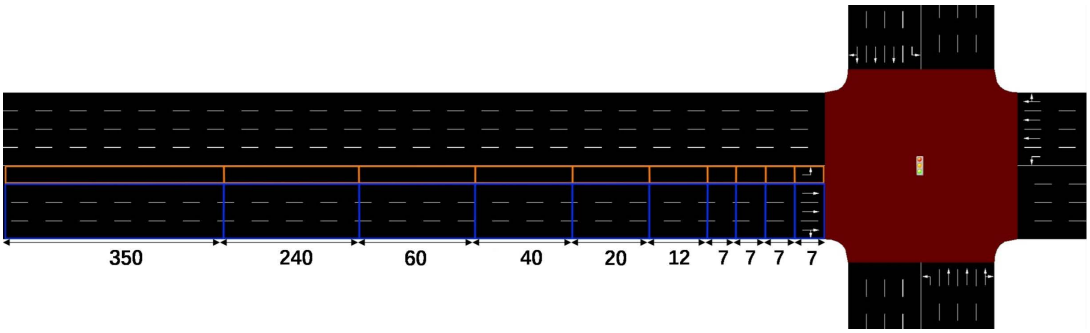

Figure 4. Schematic diagram of section division of west entrance lane at intersection
图 4. 交叉口西入站车道路段划分示意图

因此，本文基于路段不均匀分割的方法，将从停止线开始的车道离散为长度不同的单元格，离交叉口越远，单元格的长度越长。如图 4 交叉口西入站车道路段划分示意图所示，左转车道单独为一组，直行和右转三条车道为一组，车道长度 750 米，然后将每组划分成长度不同的 10 个单元格，因此，整个交叉口四个入站车道共有 80 个单元格，状态空间则由这些单元格中车辆的存在信息组成，若单元格中存在车辆，则状态值取 1，否则取 0。其中，靠近交叉口的最短单元格的长度正好比一辆车的长度长 2 米，这

样接近交叉口的车辆就能被及时的检测到。这种状态空间表示方法不仅能够获得主要的交通信息，而且还缩小了状态空间的规模，降低了智能体学习的复杂度。

3.2. 动作空间

在强化学习的工作循环过程中，智能体收到环境的状态和奖励后，需要选择合适的动作去执行，这里所有可能的动作组成的集合就称为动作空间。本文中，将图 3 所示的交叉口的 4 种交通相位定义为动作空间，用集 $A=\{NSG,NSLG,EWG,EWLG\}$ 表示。每个交通相位的持续时间为 τ_g，在切换到下一个交通相位之前，设置了一个时间为 τ_y 的黄灯时间，确保驾驶人能够及时减速并准备停车，若选择的动作和上一个时间步骤中的动作相同，则不需要设置黄灯时间。

3.3. 奖励函数

奖励函数的设计对于强化学习的学习结果非常重要，一个好的奖励函数可以准确地反映智能体所选动作的好坏，从而帮助更新策略，以实现最佳控制。

为了缓解交叉口的拥堵情况，选择减少车辆在入站车道的等待时间来实现。在时间步长 t 时，入站车道上速度小于 0.1 m/s 的所有车辆的累积等待时间定义如下：

$$awt_t=\sum_{i=1}^{n}wt_{i,t} \tag{9}$$

其中，$wt_{i,t}$ 表示在时间步长 t 时，第 i 辆车的等待时间，n 表示所有等待车辆的数量。基于此，t 时刻的奖励函数 r_t 定义为执行动作前后的累积等待时间差：

$$r_t=awt_{t-1}-awt_t \tag{10}$$

其中，awt_{t-1} 和 awt_t 分别表示在时间步长 $t-1$ 和时间步长 t 时入站车道中所有等待车辆的累积等待时间。

3.4. 交通信号控制模型

在本小节中，将介绍 3DQN 和动作价值评估网络相结合的信号控制模型(3DQN-EN)，整体网络结构如图 5 所示。

如图 5 的上半部分所示，输入的是一个 8 × 10 的状态矩阵，经过一个输入层展开成了维度为 80 的一维数组，然后是 4 个具有 400 个神经元的全连接层。最后一个全连接层被分成了两个分支，分别用来估计状态价值函数 $V(s)$和优势函数 $A(s,a)$，这两个函数最后组合在一起就是最终估计的 Q 值，如等式(8)所示。所有的激活函数都为 ReLU。智能体的学习目标就是为了让网络估计的 Q 值接近目标值 y_j，故定义损失函数：

$$L(\theta)=\frac{1}{B}\sum_{j=1}^{B}\left[Q(s_j,a_j;\theta)-y_j\right]^2 \tag{11}$$

其中，B 为批量处理的大小，目标值为：

$$y_j=r_j+\gamma Q\left(s_{j+1},\arg\max_{a_{j+1}}Q(s_{j+1},a_{j+1};\theta);\theta^-\right) \tag{12}$$

主网络参数 θ 通过自适应矩估计(Adaptive Moment Estimation, Adam)更新。文献[18]对 Adam 算法进行了评价并与其他反向传播优化算法进行了比较，结果表明 Adam 算法具有较快的收敛速度和自适应学习率，整体性能最好。

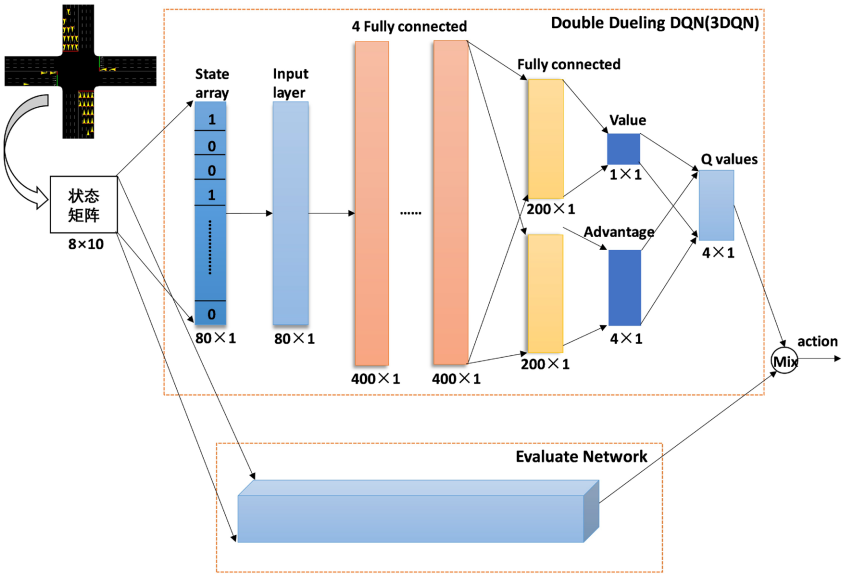

Figure 5. Network structure based on 3DQN-EN signal control model
图 5. 基于 3DQN-EN 信号控制模型的网络结构

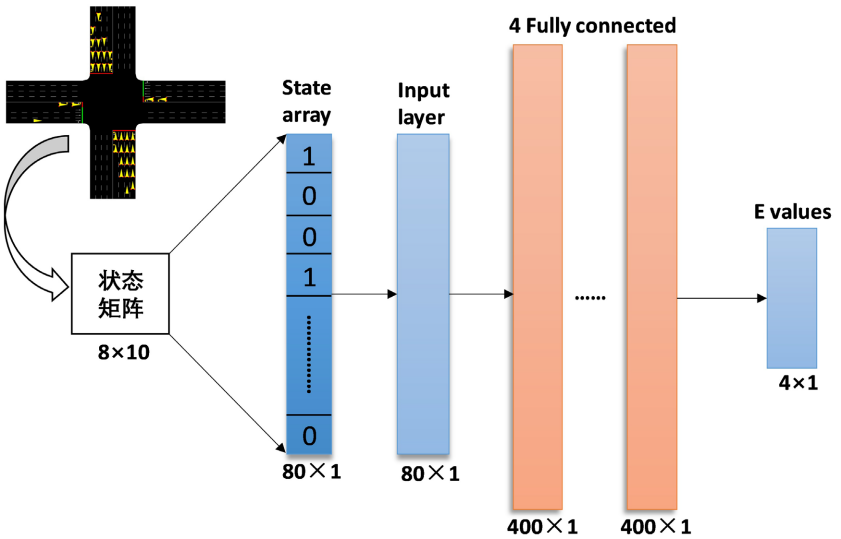

Figure 6. Structure of action value evaluation network
图 6. 动作价值评估网络的结构

除此之外，另外一个重要的问题就是学习过程中的动作选择，存在两种选择动作的方法，分别是探索和利用。探索是指在存在的动作空间中随机选择动作以学习更多的知识，而利用则是使用已经学习到的知识选择回报最大的动作，所以平衡好这两者之间的关系对学习结果非常重要。大多数的研究都采用了 ϵ 贪婪策略选择动作：

$$a_t = \begin{cases} \arg\max_a Q(s_t, a; \theta), \mu \geq \epsilon \\ \text{从动作空间中随机选取动作，} \mu < \epsilon \end{cases} \tag{13}$$

其中，μ 为[0,1)之间的一个随机整数，ϵ 为探索率，在训练过程中，由初始值线性衰减到最终值。但是这种方法在前期探索时，往往会使学习结果非常不稳定，尤其是在交通信号控制中，如果因为探索导致信

号相位混乱，可能会产生一些严重的交通事故。所以，本文利用动作价值评估网络来弥补 ϵ 贪婪策略前期探索中的不足，增加一些特殊的经验样本，有助于模型更快地学习环境规则。动作价值评估网络 E 的结构图如图 6 所示，E 是一个全连接网络，其输入和 3DQN 的输入完全相同，经过 1 个输入层、4 个全连接层，最后输出估计的每个动作的 E 值。全连接网络相对于其他神经网络结构，如 CNN、循环神经网络，没有特定的输入形状要求，可以通过增加网络层数和神经元数量增加模型的容量，从而提高模型的拟合能力。并且全连接网络的结构相对简单，易于理解和实现，每个神经元都与前一层的所有神经元相连，可以进行并行运算提高计算效率。

为了评估每个动作的好坏，直接将奖励函数作为评估网络的目标值：

$$y_j^E = r_j \tag{14}$$

则评估网络 E 的损失函数定义为：

$$L_E(\theta) = \frac{1}{B}\sum_{j=1}^{B}\left[E\left(s_j, a_j; \theta^E\right) - y_j^E\right]^2 \tag{15}$$

动作选择规则改进为：

$$a_t = \begin{cases} \arg\max_a \left[Q(s_t, a; \theta)\cdot E\left(s_t, a; \theta^E\right)\right], \mu \geq \epsilon \\ \text{从动作空间中随机选取动作，} \mu < \epsilon \end{cases} \tag{16}$$

基于 3DQN-EN 的交通信号控制的整体算法流程见附录中的算法 1。首先初始化一些变量和超参数，初始化交叉口环境(1~4 行)，获得初始状态 s_1 (7 行)。从预训练中获得的经验样本 (s_t, a_t, r_t, s_{t+1}) 会存储在经验回放器中，但是经验回放器的容量是有限的，当存储样本达到容量最大值时需要将最旧的经验样本删除(11~12 行)。然后在预训练中使用动作价值评估网络积累一些经验，通过 *Adam* 反向传播算法更新评估网络的参数 θ^E (15~19 行)。正式开始训练时，需要利用已经训练好的评估网络选择动作并执行(25~30 行)，然后继续使用 *Adam* 反向传播算法训练并更新主网络参数 θ，每隔 C 步将主网络参数复制给目标网络的参数 θ^- (33~36 行)。每一个回合训练结束后，探索率 ϵ 就会做一次更新：

$$\epsilon = 1 - \frac{h}{H} \tag{17}$$

其中，h 为当前训练回合，H 为总回合数。刚开始时，$\epsilon = 1$，智能体随机从动作空间中选择动作，随着训练的进行，智能体越来越多的利用自己已经学到的知识取选择动作，直到训练结束。

4. 模拟仿真实验

本节中，在合成的交叉口的情景下进行了一系列模拟实验，验证所提出的 3DQN-EN 算法的有效性，并与固定配时算法(Fixed-time)、DQN_AM、DQN 算法、Double DQN 算法、Dueling DQN 算法以及 3DQN 算法进行对比实验。

4.1. 实验设置

本文所有的实验评估是在 SUMO [19]平台上进行的。SUMO 是由德国航空航天中心交通研究所设计的，它是一个在时间上离散、空间上连续的微观交通模拟软件。由于它包含一个名为 Traci (Traffic Control Interface)的接口，可以直接在 Python 中调用，从而使用户更加方便地查询和修改交通模拟状态。此外，TensorFlow 框架用于实现基于深度强化学习的交通信号控制。

交叉口的模拟交通网络环境如图 2 所示，这是一个 4 向 4 车道的交叉口，最左侧的车道只允许左转，

中间两条车道只允许直行，最右侧的车道允许右转和直行，每条道路的长度为 750 米。如图 3 所示，定义了 4 种交通相位，其中设置 $\tau_g = 10$ s，$\tau_y = 4$ s。车辆的长度为 5 米，车辆之间的最小距离为 2.5 米，车辆跟车模型采用 SUMO 中默认的 Krauss 模型，车辆的加速度为 1.0 m/s^2，减速度为 4.5 m/s^2，所能达到的最大速度为 70 km/h。

在模拟环境中，车流量的生成方式非常重要，为了和真实世界更加接近，本文选择了 Weibull 分布来模拟车辆的生成，其形状参数设置为 2。进入路网的车辆数目共有 1000 辆，由于路网中的车流量服从 Weibull 分布，所以，一开始进入路网的车辆数目呈递增趋势，然后就会达到一个峰值，即现实世界中的高峰期，度过了高峰期后，车辆数目渐递减。其中，进入路网中的车辆有 75%的概率直行，25%的概率左转或者右转。

实验共进行了 200 个回合，每个回合包括 5400 个时间步长，迭代训练 800 次，详细的模型参数见表 1。整个实验过程所使用的硬件设备是一台带有 NVIDIA GeForce RTX 3050 GPU 的个人计算机，同时使用 Python 3.8 和 TensorFlow-gpu 2.4.0 来实现这个模型，在 SUMO 1.14.1 中进行仿真实验。

Table 1. Parameters of the model
表 1. 模型的参数

参数	值
经验回放器 M 的空间大小	50,000
批处理大小 B	32
折扣系数 γ	0.75
学习率 α	0.0001
目标网络更新步长 C	150
总回合数 N	200
评估网络 E 的训练回合数 P	100
迭代次数	800

4.2. 评价指标及对比研究

首先最重要的一个评价指标就是本文所设计的奖励函数。奖励函数可以直接反映出所提算法的好坏，本文的奖励函数定义为执行动作前后车辆的累积等待时间差。但是，我们只记录奖励函数为负值的情况，因为当奖励函数小于 0 时，说明智能体做出了错误的决策，而学习的过程就是使这个负值越来越大，逐渐向正值靠近。所以，第一个评价指标就是每一回合的累积负奖励值。其次，通过观察靠近交叉口处的车辆队列长度也能判断出算法是否有效，因此将每一回合的平均车辆队列长度作为第二个性能指标。最后一个性能指标为一个回合中车辆的累积等待时间。为了证明本文所提出模型的正确性和有效性，与一些基准算法和其他改进算法进行比较：

固定配时算法：使用 Webster 公式确定绿灯的持续时间，按照固定的相位顺序和绿灯时间进行交通信号控制，无法根据交叉口的交通状况进行改变。实验中按照图 3 所示的 4 种交通相位周期性的变化，每种相位的持续时间为 30 s，切换到下一相位之前会有 4 s 的黄灯时间。

DQN：最原始的深度强化学习算法，没有本文所提算法中的经验回放、双网络目标值估计、决斗网络结构等技巧。本实验中使用与文献[8]相同的状态、动作，除了批处理大小不同，其余超参数设置完全相同。

Double DQN：只在 DQN 的基础上使用了双网络目标值估计的方法，实验中其余设置和上述 DQN 算法完全相同。

Dueling DQN：只在 DQN 的基础上改变了网络结构，采用了双决斗网络结构，实验中其余设置和 DQN 完全相同。

DQN_AM：使用了文献[10]中提出的新的改进算法，在 DQN 算法的基础上增加了注意力机制，实验中所有超参数的设置、动作、状态、奖励和 3DQN-EN 算法完全相同。

此外，我们还将 3DQN 算法与 3DQN-EN 算法单独进行对比实验，两者唯一的区别就是动作价值评估网络 E，以呈现本文所提出模型的良好性能。

4.3. 实验结果与分析

本文有三个评价指标：累积负奖励值、累积车辆延迟时间和平均队列长度，我们通过这三个评价指标来验证所提模型的正确性与有效性。图 7 展示了 3DQN-EN 及其对比算法的累积负奖励，我们的目标是最大化累积负奖励。从图中可以看出，由于经验价值评估网络的存在，在一开始的训练回合中，3DQN-EN 的累积负奖励值就大于−9000，然后经过不断地迭代训练，在大约 80 回合时收敛到−3000。而 DQN、Double DQN、Dueling DQN 虽然最后也向−3000 收敛，但是在前期的训练回合中累积负奖励值都小于−12,000，且波动较大，尤其是 DQN 算法，在 80 回合之前一直处于不稳定的状态。固定配时算法的累积负奖励值一直处于−21,000 和−18,000 之间，虽然非常稳定，但是它无法通过多次迭代训练获得更高的累积负奖励值。而 DQN_AM 则是所有算法中表现最差的，大约在 90 回合时才慢慢开始收敛。

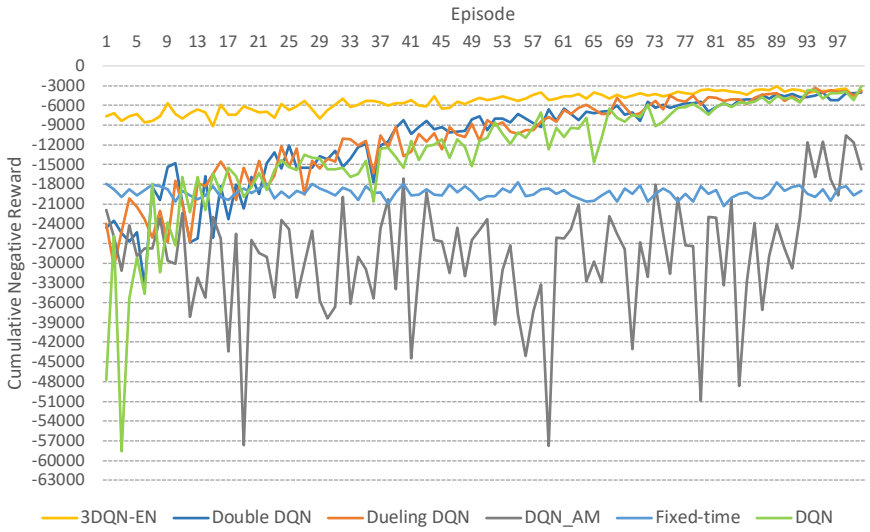

Figure 7. Comparison of cumulative negative reward of each algorithm (the higher the better)
图 7. 各算法的累积负奖励对比(越大越好)

就累积车辆延迟时间来说，如图 8 所示，3DQN-EN 的累积车辆延迟时间最短，总体小于 20,000 s，在大约 80 回合收敛到 10,000 s。Double DQN 和 Dueling DQN 在大约 60 回合时开始稳定下来，而 DQN 到 80 回合才开始稳定。而固定配时算法的累积车辆延迟时间一直保持在 40,000 s 附近。DQN_AM 算法则一直难以收敛，处于巨大波动中。

图 9 为各算法的平均队列长度对比，可以看到在 3DQN-EN 模型的控制下，平均队列长度维持在 2 到 4 辆车之间，而固定配时算法的平均队列长度在 6 到 8 辆车之间。同样地，DQN 算法到 80 回合才开始稳定，然后收敛到 2 辆车，Double DQN 和 Dueling DQN 到 60 回合开始稳定，最终也收敛到 2 辆车，

DQN_AM 的效果仍然是最差的。

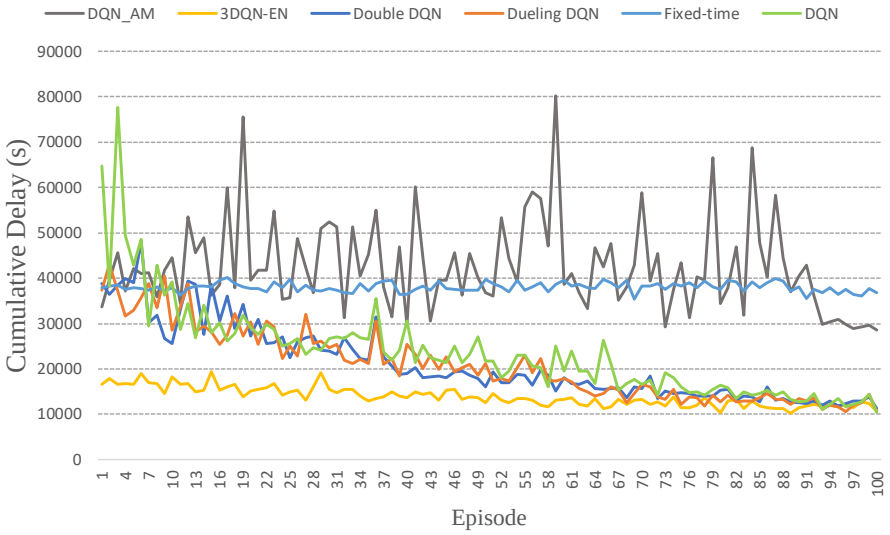

Figure 8. Comparison of cumulative vehicle delay time of each algorithm (the lower the better)
图 8. 各算法的累积车辆延迟时间对比(越小越好)

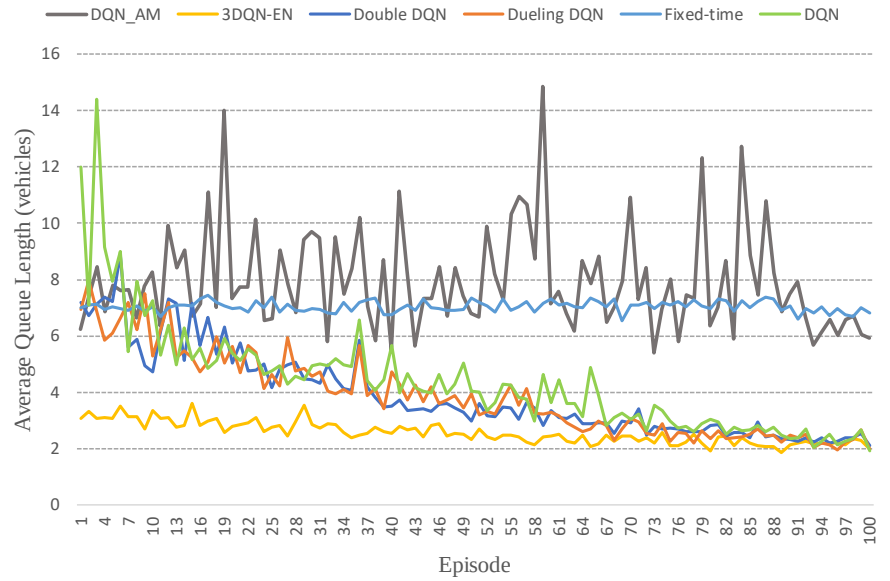

Figure 9. Comparison of average queue length of each algorithm (the lower the better)
图 9. 各算法的平均队列长度对比(越小越好)

此外，本文还总结了各算法的平均性能指标和算法收敛时间，如表 2 所示。所提的 3DQN-EN 模型的平均累积奖励相较于 DQN 来说增加了 58.9%，相较于 Double DQN 来说增加了 51.9%，相较于 Dueling DQN 来说增加了 51.3%。就平均累积车辆延迟时间来说，3DQN-EN 相较于 DQN 来说减少了 41.3%，相较于 Double DQN 来说减少了 34.5%，相较于 Dueling DQN 来说减少了 33.9%。对于平均队列长度，3DQN-EN 相对于 DQN、Double DQN、Duleing DQN 都减少了 25%。从算法收敛时间上看，本文所提的 3DQN-EN 模型也优于其他算法。由于固定配时算法没有所谓的算法收敛概念，而 DQN_AM 算法一直难以收敛，所以表 2 中没有标明其算法收敛时间。

综上所述，本文所提的模型 3DQN-EN 在各个评价指标和算法收敛时间上均优于其他对比算法。

Table 2. Average performance and convergence time of each algorithm
表 2. 各算法的平均性能和收敛时间

指标	平均累积奖励	平均累积车辆延迟时间(s)	平均队列长度(辆)	算法收敛时间(s)
Fixed-time	−19288.20	38030.69	7	/
DQN_AM	−28733.50	42652.57	8	/
DQN	−13128.20	23616.87	4	5339.84
Double DQN	−11227.60	21144.17	4	6810.92
Dueling DQN	−11097.50	20971.76	4	4670.25
3DQN-EN	**−5399.41**	**13853.09**	**3**	**2623.50**

除此之外，本文还单独验证了动作价值评估网络的存在价值。具体地，将 3DQN-EN 模型与无动作价值评估网络的 3DQN 模型进行对比实验，实验评估结果如表 3 所示。从表 3 中可以看出，3DQN-EN 模型在各个性能指标上都优于 3DQN 模型。其中，平均累积奖励提高了 48.8%，平均累积车辆延迟时间减少了 66.5%，平均队列长度减少了 75%。主要差异在于 3DQN-EN 模型可以借助预训练阶段训练动作价值评估网络积累经验，从而解决了初始阶段因 ϵ 贪婪策略造成的训练效果不稳定的问题。表 3 中的算法收敛时间是指算法的累积负奖励达到−6000 所耗费的时间，3DQN-EN 的算法收敛时间减少了 46.4%。从以上实验结果分析来看，本文所提出的 3DQN-EN 模型在所有评价指标上都取得了最佳性能。

关于实验方案的公平性与合理性，一方面，在本文的实验方案中，3DQN-EN 相对于其他对比算法来说，由于需要预先训练经验价值评估网络，所以存在预训练阶段。预训练阶段仅用来在所有可能的动作空间中探索效果最好的动作，然后提供给主网络再进一步探索和选择最优动作，这样能够避免主网络前期的盲目探索。在实验过程中，训练经验价值评估网络所耗费的时间远小于其他算法在正式训练阶段收敛前耗费的时间，因此在我们的实验方案中指出 3DQN-EN 比其他对比算法收敛更快是合理且公平的。另一方面，在正式训练过程中，依然是 3DQN 算法与环境交互学习，以获取最大奖励，并不存在动作价值评估网络的作用盖过主网络的情况。

Table 3. Experimental evaluation results of 3DQN and 3DQN-EN
表 3. 3DQN 与 3DQN-EN 的实验评估结果

指标	平均累积奖励	平均累积车辆延迟时间(s)	平均队列长度(辆)	算法收敛时间(s)
3DQN	−11057.10	20829.11	4	5653.62
3DQN-EN	**−5399.41**	**13853.09**	**3**	**2623.50**

5. 结论

本文中，提出了一个基于 3DQN-EN 算法的交通信号控制模型。该模型在学习初始阶段建立了一个动作价值评估网络，以增加智能体学习的深度经验。除此之外，还对 DQN 算法进行了一些改进，如经验回放、双网络目标值估计、双决斗网络结构。实验结果表明，动作价值评估网络可以帮助减少前期学习阶段的波动、加快收敛速度，且所提出的算法在平均累积奖励、平均累积车辆延迟、平均队列长度等指标上均优于 DQN、Double DQN、Dueling DQN、DQN_AM 以及固定配时方法，能够有效的减少交通拥堵情况。但是，在本文中仅在单交叉口的情景下进行了实验。在未来的工作中，需要研究更加先进的多

智能体深度强化学习算法，在现实世界中的多交叉口环境下进行验证。

参考文献

[1] Webster, F.V. (1958) Traffic Signal Settings. https://trid.trb.org/view/113579

[2] Vincent, R.A. and Peirce, J.R. (1988) "MOVA": Traffic Responsive, Self-Optimising Signal Control for Isolated Intersections. https://trid.trb.org/view/295257

[3] Kronborg, P. and Davidsson, F. (1993) MOVA and LHOVRA: Traffic Signal Control for Isolated Intersections. *Traffic Engineering and Control*, **34**, 195-200.

[4] Kronborg, P. and Davidsson, F. (1996) Development and Field Trials of the New SOS Algorithm for Optimising Signal Control at Isolated Intersections. *IEE Conference Publication*, **42**, 80-84. https://doi.org/10.1049/cp:19960295

[5] Sims, A.G. (1979) The Sydney Coordinated Adaptive Traffic System. *Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems*, Pacific Grove, 1979, 12-27.

[6] Hunt, P.B., Robertson, D.I., Bretherton, R.D., *et al*. (1981) SCOOT-A Traffic Responsive Method of Coordinating Signals. https://trid.trb.org/view/179439

[7] Wang, S., Xie, X., Huang, K., *et al*. (2019) Deep Reinforcement Learning-Based Traffic Signal Control Using High-Resolution Event-Based Data. *Entropy*, **21**, 744. https://doi.org/10.3390/e21080744

[8] Luo, J., Li, X. and Zheng, Y. Researches on Intelligent Traffic Signal Control Based on Deep Reinforcement Learning. 2020 16*th International Conference on Mobility, Sensing and Networking* (*MSN*). Tokyo, 17-19 December 2020, 729-734. https://doi.org/10.1109/MSN50589.2020.00124

[9] 唐慕尧, 周大可, 李涛. 结合状态预测的深度强化学习交通信号控制[J]. 计算机应用研究, 2022, 39(8): 2311-2315. https://doi.org/10.19734/j.issn.1001-3695.2021.12.0704

[10] 任安妮, 周大可, 冯锦浩, 等. 基于注意力机制的深度强化学习交通信号控制[J]. 计算机应用研究, 2023, 40(2): 430-434. https://doi.org/10.19734/j.issn.1001-3695.2022.06.0334

[11] Wang, X., Wang, S., Liang, X., *et al*. (2022) Deep Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, **33**, 1-15. https://doi.org/10.1109/TNNLS.2022.3207346

[12] Mnih, V., Kavukcuoglu, K., Silver, D., *et al*. (2015) Human-Level Control through Deep Reinforcement Learning. *Nature*, **518**, 529-533. https://doi.org/10.1038/nature14236

[13] Van Hasselt, H., Guez, A. and Silver, D. (2016) Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 2094-2100. https://doi.org/10.1609/aaai.v30i1.10295

[14] Hasselt, H. (2010) Double Q-Learning. *Advances in Neural Information Processing Systems*, **23**, 2613-2621.

[15] Wang, Z., Schaul, T., Hessel, M., *et al*. (2016) Dueling Network Architectures for Deep Reinforcement Learning. *ICML*'16: *Proceedings of the* 33*rd International Conference on International Conference on Machine Learning*, New York, 19-24 June 2016, 1995-2003.

[16] 刘智敏, 叶宝林, 朱耀东, 等. 基于深度强化学习的交通信号控制方法[J]. 浙江大学学报(工学版), 2022, 56(6): 1249-1256.

[17] Mousavi, S.S., Schukat, M. and Howley, E. (2017) Traffic Light Control Using Deep Policy-Gradient and Value-Function-Based Reinforcement Learning. *IET Intelligent Transport Systems*, **11**, 417-423. https://doi.org/10.1049/iet-its.2017.0153

[18] Haji, S.H. and Abdulazeez, A.M. (2021) Comparison of Optimization Techniques Based on Gradient Descent Algorithm: A Review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, **18**, 2715-2743.

[19] Monga, R. and Mehta, D. (2022) Sumo (Simulation of Urban Mobility) and OSM (Open Street Map) Implementation. 2022 11*th International Conference on System Modeling & Advancement in Research Trends* (*SMART*), Moradabad, 16-17 December 2022, 534-538. https://doi.org/10.1109/SMART55829.2022.10046720

附 录

算法 1 用于交通信号控制的 3DQN-EN 算法

输入：探索率 ϵ，经验回放器的空间 M，批处理大小 B，折扣率 γ，评估网络 E 的训练回合数 P，训练总回合数 N，每个回合的时间步长 T。

1. 用零元素初始化经验回放器 m；
2. 用随机值初始化主网络参数 θ 和动作价值评估网络参数 θ^E；
3. 初始化目标网络参数 $\theta^- = \theta$；
4. 初始化交叉口环境数据，导入车流；
5. for *episode* = 1 to N do
6. if *episode* ≤ P then
7. 观察交叉口当前的状态 s_1；
8. for t = 1 to T do
9. 随机选择一个动作 a_t；
10. 交叉口环境转变成一个新的状态 s_{t+1}，并根据式(10)得到奖励 r_t；
11. if *len*(m) > M then
12. 将经验回放器中最旧的经验样本删除；
13. end if
14. 将样本 (s_t, a_t, r_t, s_{t+1}) 存储到经验回放器 m 中；
15. if *len*(m) >= B then
16. 从 m 中随机选择 B 个经验样本，并根据式(14)计算评估网络的目标值；
17. 根据式(15)计算评估网络 E 的损失函数 $L_E(\theta)$；
18. 使用 *Adam* 反向传播算法训练并更新 E 的参数 θ^E；
19. end if
20. $s_t = s_{t+1}$；
21. end for
22. else
23. 观察交叉口当前的状态 s_1；
24. for t = 1 to T do
25. 根据式(16)选择一个动作 a_t；
26. if $a_t == a_{t-1}$ then
27. 继续保持当前交通相位 τ_g s；
28. else
29. 将信号灯转变为黄灯并保持 τ_y s；
30. 改变到下一个交通相位并保持 τ_g s；
31. 执行步骤 10；
32. 执行步骤 11-14；
33. 从 m 中随机选择 B 个经验样本，根据式(12)计算目标值 y_i；
34. 根据式(11)计算损失函数 $L(\theta)$；
35. 使用 *Adam* 反向传播算法训练并更新主网络参数 θ；
36. 每 C 步更新目标网络的参数 $\theta^- = \theta$；
37. 执行步骤 20；
38. end for
39. 更新 $\epsilon = 1 - \dfrac{episode}{N}$；
40. end for