CS350 Reinforcement Learning for Adaptive Traffic Light Control

Data Science Project Specification

1. Introduction

With the widespread use of automobiles, traffic congestion has become increasingly severe. Faced with this challenge, we need innovative and dynamic traffic control systems to reduce the impact of congestion between regions. Despite various proposed solutions, including machine learning and urban planning, traffic congestion remains a significant problem.

This project aims to utilize reinforcement learning algorithms to regulate multiple traffic signals, optimizing traffic flow within regions. Especially in specific road sections and unique situations, our system can react in real-time, alleviating traffic congestion. To enhance the system's intelligence, the project incorporates technologies such as Memory Palace and Phase Gate [1]. This comprehensive approach, considering various traffic scenarios, makes our traffic control system more flexible and efficient.

2. Objective

The objective of this project is to optimize the urban traffic system based on different time flows. The primary goal is to create an adaptive traffic light control system that responds in real-time to varying traffic conditions, effectively reducing congestion, and improving overall traffic flow. Since we focus more on reinforcement learning algorithm, the Image Recognition part will be ignored here.

- The first milestone of the project is building up a simulator (environment) of road situation which will be used in algorithm test later. The simulator will first be 1 intersection, then later it can generate cars and with more intersections.
- The second goal, come out the algorithm of the control system and train with apply them on the simulator. This section will address the decisions made by single traffic light in response to real-time traffic flow.
- Apply **Multiagent System (M.A.S)** to the control system, enhancing the Interconnectedness of Multiple Traffic Lights, make control to different traffic light.
- Add more complex road situation to the simulator and M.A.S, such as traffic light of footpath and bend.

3. Methodology

To achieve the main goal of this project, plan-driven and agile methods will be used, this can easily help the project is always on progress and complete objectives step by step. This also performs well to Supervisor or other support member understand the

progression. Due to some unpredictable situation, plan-driven may not fit that well, agile then can help with catch up with the plan.

To build the control system visualisable and self-adaptive traffic light, a lot of tools might be used, reinforcement learning algorithm, python and GitHub is mainly used in this project. Research also part of this project, before producing the control system, research provides information about reinforcement learning for adaptive traffic lights, involve its validation. SUMO may also be used as a implementation for the environment.

Algorithm, Deep Q-learning (DQN) will be used, which is one approach to approximate the Q-learning algorithm using neural networks to estimate the value function[2], used to help traffic light make decisions. For some basic part in this model, Q-learning is also feasible here. This will be the main part of the model, metrics, value function and reward will be designed here. Algorithm will come out come out value function by using value iteration method. The reward of Value function is mixed by several features, *Waiting Time of Vehicles, Queueing length of vehicles, Number of Vehicles passed, Time allow Vehicle pass and action of Traffic Light.*

Python is the mainly programming language used in this project. It uses a simplified syntax with an emphasis on natural language. Algorithms are easier to be implemented by python since libraries in python are powerful enough. Pytorch will be used, it has powerful capabilities in building and training deep learning models. Computer-generated simulations are simply imitation of a situation or a process[3], it performs what decisions made by agents and helps to improve the control system. In case, this will also be developed in python base with SUMO.

SUMO (Simulation of Urban Mobility) is an open-source traffic simulation tool used to simulate urban traffic flow. It is employed for researching and evaluating traffic planning and control strategies. The simulator will be built based on SUMO packages.

Research, to successfully build up the control system with visualisation, research is necessary. GitHub and Google Scholar will provide a lot of information relates project, both code and publication. From GitHub, code related potential issue can be found a solution. Code reading helps code more readable to people other than me. Publication reading provides general adivice to the whole project, idea can be raised after reading and analysis.

Jira is a comprehensive project management and issue tracking tool; project information can be recorded on Jira. On Jira, it can record issues, errors, and requirements, and track the status and progress of issues at any time, this will provide a great help to complete the progression and final report.

4. Timetable

Majority of project will base the timetable and goal driven; goals should be accomplished in given time. Weekly job will follow the framework of weekday timetable.

Meeting with supervisor will be on Wednesday Noon every two week and will have meeting will a Phd. Support every week. Meeting will give advice on current state of project, help solve difficulties and push on project progression.

Goals	Best End Time	DDL
Build Simulator and RL tutorial	End of October	First week of November
RL model	End of November	Mid of December
Add M.A.S to model	End of February	End of February
Apply to Real Road	Not necessary, do it if have enough time.	

The table below will contain the information about weekly job.

Monday	Plan this week project state
Tuesday	Research, Tutorial
Wednesday	Weekly meeting, slightly replan
Thursday	Coding follows the plan
Friday	Coding follows the plan
Saturday	Ending weekly plan

5. Reference

[1] H. Wei, G. Zheng, H. Yao and Zhenhui Li, 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. Available at: https://dl.acm.org/doi/pdf/10.1145/3219819.3220096

[2] Ankit Choudhary, 2023. A Hands-On Introduction to Deep Q-Learning using OpenAI Gym in Python. Available at: https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/#h-what-is-deep-q-learning

[3] Terence Shin, 2020. Building Simulations in Python — A Step by Step Walkthrough. Available at: https://towardsdatascience.com/building-simulations-in-python-a-complete-walkthrough-3965b2d3ede0