Project report: Course project EE597

Ritwik Ashok
ID:959361627
Department of Electrical Engineering
Pennsylvania State University
State College, PA 16803

Ratnesh Yadav
ID:962052529
Department of Electrical Engineering
Pennsylvania State University
State College, PA 16803

Real-Time Autonomous Intersection Management Using Deep Q-Learning

1 Motivation

Urban congestion remains one of the most pressing challenges in modern cities, leading to significant economic and environmental impacts. Our project introduces a solution to this persistent problem through the development of a real-time autonomous intersection management system. Utilizing Reinforcement Learning (RL) techniques, our system is designed to overhaul the traditional methods of intersection control, aligning with the emerging era of connected autonomous vehicles (CAVs).

The role of connected autonomous vehicles (CAVs) is increasingly crucial in the landscape of urban traffic management. As these vehicles become more common, they bring the potential for a substantial shift in how traffic flows are managed. Our project envisions a scenario where both CAVs and traffic infrastructure operate as intelligent, collaborative agents. This partnership is pivotal for realizing a traffic management system where decisions are dynamically adjusted based on real-time traffic conditions.

Incorporating intelligent algorithms, our system leverages the capabilities of CAVs to interact seamlessly with traffic signals and other infrastructural elements. This interaction allows for the continuous monitoring and adjusting of traffic flows at various intersections. By doing so, our system addresses congestion precisely at the urban intersections that often become bottlenecks, significantly disrupting city traffic flow and contributing to increased travel times and vehicle emissions.

Our approach creates a symbiotic relationship between the technological advances in autonomous vehicles and the existing urban traffic infrastructure. This integration not only improves the efficiency of traffic management but also enhances the overall utility of urban road networks. By reducing the frequency and severity of congestion, our system contributes to decreased overall travel times, lower emissions from idling and stop-and-go traffic, and improved air quality.

The broader implications of our project extend beyond immediate traffic improvements. By significantly reducing congestion, we contribute to enhanced urban mobility, which is essential for the economic vitality and quality of life in urban areas. The success of this project could serve as a model for other cities, demonstrating the potential benefits of integrating advanced technology with traditional traffic management practices.

Ultimately, our goal is to set a new standard in intersection management that could be adopted widely across various urban settings. This project not only addresses current urban transportation issues but also anticipates future developments in vehicle technology and urban planning. By pioneering a scalable and adaptive traffic management solution, we aim to lead the way in sustainable urban transportation innovations, setting a benchmark that could transform the core infrastructure of city traffic systems worldwide.

2 Objective

The primary objective of this project is to design and implement a Reinforcement Learning (RL)-based policy, specifically utilizing Q-learning, to enhance the capability of autonomous agents navigating complex urban intersections. This initiative is focused on empowering these agents, particularly connected autonomous vehicles (CAVs), to manage intersection crossings with an unprecedented level of efficiency and safety. The integration of this sophisticated RL technique is pivotal for developing an autonomous intersection management system that is capable of making dynamic, intelligent decisions in real-time scenarios, thus catering to the rapidly evolving landscape of urban mobility.

In pursuit of this goal, the implementation strategy involves meticulously equipping the driving agents with the capability to evaluate and strategically exploit available gaps in traffic flows. This requires establishing a finely tuned balance between caution and assertiveness within the decision-making protocols of the autonomous agents. Such calibration is essential to ensure that these agents can optimize traffic throughput while upholding stringent safety standards. The ability of the system to accurately assess and adapt to changing traffic conditions is a critical component towards achieving an equilibrium in traffic management, facilitating smoother flows and reducing the likelihood of traffic jams and accidents.

The broader aim of this project extends beyond mere technical implementation; it seeks to enable autonomous agents to effectively utilize existing opportunities to traverse intersections without undue delays. This functionality is expected to enhance overall traffic throughput significantly and decrease congestion in urban environments. By streamlining the process of intersection crossing, the project endeavours to deliver substantial improvements in urban traffic conditions, including the reduction of vehicle idling times and associated emissions, thus contributing positively to environmental sustainability and urban quality of life.

3 Methodology

3.1 Data exploration and Visualization

In the development of our Reinforcement Learning (RL) model, the data exploration and visualization processes are pivotal for understanding and interpreting the environmental dynamics that our autonomous agents encounter. Within the scope of our project, the dataset consists predominantly of environmental states representing a simulated urban intersection. Specifically, the environment is structured as a 21x21 grid, where the vertical axis simulates a roadway navigated by our autonomous agent, and the horizontal axes are occupied by lanes with simulated vehicles. This setup is illustrated in **Figure 1** of our report, which depicts a world with two traversable lanes marked by directional arrows that indicate the flow of traffic.

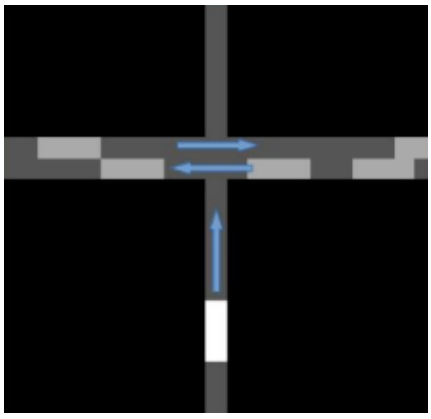

Figure 1: Environment design

During the initialization of our simulation, traffic configurations are randomly generated across both lanes, setting the stage for the autonomous agent to begin its journey from the bottom border of the grid. The simulation is designed to track the agent's progress as it attempts to traverse the intersection successfully. The termination of the simulation is triggered by one of two outcomes: the agent successfully crossing the intersection or a collision occurring with another vehicle. Although a substantial portion of the vertical space within the grid is dedicated to displaying the vehicle's approach or passage through the intersection, it also serves a secondary purpose of helping the neural network recognize spatial relationships, which are critical for the agent's decision-making process.

Dynamic interactions within the environment further enrich the dataset. Each time a grey vehicle fully emerges onto the grid from the periphery, the system calculates a variable random number to determine the likelihood of another vehicle appearing in the subsequent time step. This stochastic element introduces variability in the average gap size between vehicles, thereby mimicking the unpredictable nature of real-world traffic flows. This aspect of the simulation is crucial for training the RL model to adapt to changing conditions and to make informed decisions based on the state of the environment.

The set of actions available to the agent is intentionally limited to enhance the focus on strategic decision-making. Initially, the agent can only move forward as it approaches the intersection. However, once it reaches the intersection, it must choose between continuing to advance or remaining stationary. This decision is critical as it must ensure safe passage through the intersection without unnecessary delays. The strategic choice of when to move or pause is central to the RL policy's goal of optimizing traffic throughput while maintaining safety.

By simulating various traffic scenarios and visualizing the agent's interactions with the environment, we can refine our RL models to better handle the nuances of real-world driving conditions. This foundation supports our broader objective of developing a sophisticated autonomous intersection management system that can significantly improve urban traffic flow and safety.

3.2 Environment Design

In the Environmental Design section of our methodology, we describe the configuration of the simulation environment that serves as the testbed for our autonomous intersection management system, designed to replicate realistic urban intersections. Our environment is tailored to accommodate both single-lane and multi-lane traffic scenarios; each modelled on a separate learning code to ensure each scenario's specific challenges and dynamics are addressed effectively. The environment is constructed as a grid, measuring 21 pixels in height and width, where vehicles, including the autonomous agents (CAVs), have a uniform span of 3 pixels. This design facilitates precise control over vehicle movements and interactions within the simulated environment.

For single-lane scenarios, the simulation focuses on the interactions of the CAV with the ongoing traffic within a single lane, which simplifies the complexities associated with intersection navigation. This setup allows for focused testing and tuning of the Reinforcement Learning (RL) algorithms specific to environments where vehicle interactions are more predictable and contained. In contrast, the multi-lane scenarios are designed to simulate more complex traffic patterns involving multiple lanes and a variety of vehicular behaviours and interactions. This requires a separate set of RL algorithms tailored to manage the increased complexity and to optimize decision-making in a dynamically changing traffic environment.

The decision to develop distinct learning models for single-lane and multi-lane scenarios allows us to customize the RL approaches according to each scenario's specific requirements and challenges. This distinction is critical for achieving high accuracy and efficiency in the autonomous navigation of intersections under varying traffic densities and configurations. The single-lane model hones in on the precision of movements and decision-making in a controlled setting, while the multi-lane model addresses the broader challenges of coordinating multiple agents and traffic flows simultaneously.

By employing a grid-based model and dedicating different learning codes to single and multi-lane scenarios, our project creates a versatile and robust testing ground. This environment not only allows autonomous agents to demonstrate their capabilities in navigating intersections under both simplified and complex conditions but also serves as a crucial framework for observing and enhancing the agents' strategic decision-making processes.

3.3 Traffic Dynamics

The operational framework of our simulation is designed to emulate the complex and dynamic conditions typical of urban intersections. The simulation progresses through discrete time steps, with each vehicle advancing one pixel per step, including the autonomous agent (CAV). This methodical progression ensures consistent timing of vehicle interactions and aids in the accurate analysis of traffic flow and agent behaviors. In this simulated environment, the autonomous agent navigates vertically toward the intersection while other vehicles traverse horizontally, creating a dynamic setting in which the agent must continuously adapt its navigation strategy for safety and efficiency.

At the core of the simulation is the intersection, which serves as a critical arena for testing the decision-making capabilities of the autonomous agent. As the agent approaches this juncture, it encounters variable traffic patterns due to the horizontal movements of other vehicles. This necessitates a robust decision-making process where the agent must evaluate its surroundings and make real-time decisions to avoid collisions and minimize transit delays. The effectiveness of the agent in navigating this complex scenario is pivotal to demonstrating the practical utility of our Reinforcement Learning (RL) models in managing urban traffic efficiently and safely.

To increase the realism and challenge of the simulation, we incorporate variations in traffic density and vehicle speeds to mimic real-world traffic conditions closely. These variations compel the autonomous agent to continually adjust its decision-making strategy based on the evolving traffic scenarios it faces. This aspect of the simulation is crucial for testing and refining the RL policies under realistic conditions, ensuring that the solutions developed are both effective and scalable.

3.4 Q-Learning Implementation

The Q-learning algorithm operates by iteratively updating the Q-values associated with specific state-action pairs. These Q-values essentially serve as indicators, guiding the connected autonomous vehicle (CAV) toward the most beneficial action in any given state. This approach allows the CAV to learn from its interactions within the simulation, continuously improving its ability to navigate intersections with increased safety and efficiency.

State Space: The state space within the framework is defined by the pixel data in the simulation environment, which captures the positions of the agent and other vehicles. It also includes the direction of traffic flow and the sizes of gaps between vehicles, which are crucial for making real-time navigation decisions.

Action Space: The action space is limited to two actions: "go forward" or "stay in place." The decision to take a particular action is based primarily on the evaluation of traffic gaps at the intersection, ensuring that movements are made only when there is a sufficient gap to do so safely and efficiently.

Reward System: A reward system that incentivizes the CAV to cross the intersection as efficiently as possible while avoiding collisions. This system reinforces behaviours that contribute to smooth traffic flow and increased safety.

At the heart of the model, the Q-function, *Q(s, a)*, calculates the maximum discounted future reward expected for performing an action *a* in a state *s* and continuing optimally thereafter. This function is pivotal in optimizing the CAV's decision-making process, ensuring that it responds appropriately to immediate conditions and aligns with long-term objectives of minimizing delays and maximizing safety at intersections.

$$Q\left(s_t, a_t\right) = \max\left(R_{t+1}\right) \tag{1}$$

We select the action with the highest Q-value. The policy is

$$\pi = \operatorname{argmax}_a Q(s, a) \tag{2}$$

Using the Bellman equation:

$$Q(s,a) = r + \gamma \max_{a'} Q\left(s', a'\right) \tag{3}$$

The network is trained with the loss function:

$$L = \frac{1}{2}\left[r + \max_{a'} Q\left(s_1, a_1\right) - Q(s,a)\right]^2 \tag{4}$$

3.5 Training and Testing

During the training process, the neural network adopts a trial-and-error approach facilitated by Q-learning. Experience replay is employed to diversify training samples, enhancing the efficiency of the learning process. Optimization techniques involve training the network with a mean squared error loss function. Additionally, random actions are introduced to foster exploration, regulated by the epsilon parameter. These strategies collectively contribute to refining the network's performance and decision-making capabilities.

Code Listing 1: Pseudo Code(Training Phase)

```
for e in epochs:
    # Instantiate the environment
    env = initialize_environment()

    # Continue running the trial until it is terminated
    while not trial_terminated(env):
        # Initialize the environment for the trial
        initialize_environment_state(env)

        # Check the agent's position relative to the intersection
        if at_intersection(env):
            # If at the intersection, move forward
            move_forward(env)
        else:
            # If not at the intersection
            if random_float() < epsilon:
                # Explore: Choose a random action
                choose_random_action(env)
            else:
                # Exploit: Choose the best known action from the policy model
                choose_policy_action(env)

    # Use experience replay to learn from past experiences
    experience_replay(env)

    # Update the model based on the experience
    update_model_from_experience(env)
```

The training methodology employed in our Q-Learning algorithm involves iterative epochs, each comprising the instantiation of a simulated environment resembling a road network. This environment is represented as a 21x21 NumPy array, with road segments denoted by a value of 1 and traffic introduced randomly in horizontal lanes. The agent, embodied as a vehicle, progresses within this environment, advancing until encountering an intersection where it must make a decision: either proceed forward or remain stationary. Initially, to ensure exploration of diverse states and actions, the agent's decisions are governed by a balance between random exploration and exploitation of learned policies, determined by an epsilon-greedy strategy. Specifically, with a probability dictated by epsilon, actions are chosen randomly, fostering the discovery of optimal strategies. Conversely, when epsilon permits, actions are determined by predictions from a neural network model, trained to estimate Q-values—expected cumulative rewards associated with taking specific actions in given

states. These Q-value estimations guide the agent's decision-making process, facilitating efficient navigation through the environment. After each action, the resulting state, action taken, received reward, and subsequent state are stored in an ExperienceReplay memory, enabling the agent to learn from past experiences. Periodically, the model is updated by sampling a random minibatch from the ExperienceReplay memory, utilizing these experiences to refine its predictions and adjust its parameters via techniques such as gradient descent. Training iterations continue until convergence, defined by either successful traversal through the intersection or a collision event, ensuring the model iteratively learns and improves its decision-making capabilities over multiple epochs.

Code Listing 2: Pseudo Code(Testing Phase)

```
    # Load the pre-trained model
model = load_pretrained_model()

# Iterate over each trial
for t in trials:
    # Set up the environment
    env = instantiate_environment()

    # Continue the trial until it is terminated
    while not trial_terminated(env):
        # Initialize the environment for this step of the trial
        initialize_environment_state(env)

        # Check if the agent is at the intersection
        if at_intersection(env):
            # Determine action based on Q-values from the model
            action = determine_action_from_q_values(model, env)
        else:
            # If not at the intersection, move the agent forward
            move_agent_forward(env)
```

The testing phase commences with the loading of the pre-trained model into memory. Subsequently, a series of trials are conducted wherein an environment is instantiated, mirroring the road network configuration utilized during training. Within each trial, the environment is initialized, and the agent's position is monitored to ascertain if it has reached an intersection. At the intersection, the trained model's Q-values are leveraged to deduce the optimal action for the agent to take. Conversely, when the agent is not at an intersection, it continues to progress forward. Throughout this process, a tally of results is maintained, crucial for subsequent evaluation metrics. Notably, during testing, the model's predictions are consistently invoked at intersections via the model.predict() method, ensuring the agent's actions are guided by the learned policies. This approach enables a systematic evaluation of the trained model's performance under real-world conditions, providing insights into its efficacy and generalization capabilities beyond the training data.

4 Results and Insights

The performance outcomes of our trained models within both single-lane and multi-lane intersection environments were recorded, demonstrating substantial improvements in the autonomous vehicle's (CAV's) ability to navigate intersections.

Evaluation Metrics:

A script was written to evaluate the learned model and the Q-value implementation that instantiates an environment and loads the trained model. This testing program presents the agent with thousands of randomly generated environments and records the following two evaluation metrics:

- Number of missed crossing opportunities
- Percentage of successful crossings

Single-Lane Intersection Analysis:

In the single-lane intersection scenario, the results were less than optimal when employing a random policy as a baseline for comparison.

Missed Opportunities: 6.14 %, Success Rate: 21.30 %

Specifically, the model exhibited a 6.14% rate of missed opportunities, where the CAV failed to navigate the intersection when it was actually safe to do so, and achieved a success rate of only 21.30% in safely navigating the intersection without collisions. In contrast, when applying the learned Q-learning policy, the missed opportunities dramatically decreased to 0.17%, and the success rate impressively reached 100%. These results unequivocally demonstrate that the RL model successfully learned an effective crossing policy, significantly enhancing the CAV's decision-making capabilities in a controlled environment.

Missed Opportunities: 0.17 %, Success Rate: 100 %

Two-Lane Intersection Analysis:

Transitioning to the more complex two-lane intersection environment, the application of learned policy showcased its effectiveness.

Missed Opportunities: 5.52%, Success Rate: 99.9 %

The rate of missed opportunities was observed at 5.52%, which, while higher than in the single-lane scenario, remains impressively low considering the increased complexity of navigating multiple lanes. The success rate here was 99.9%, indicating that the CAV almost always successfully navigated the intersection safely. This high success rate in a multi-lane environment underscores the adaptability and effectiveness of our Q-learning-based policy, even under more challenging traffic conditions.

Comparative Insights and Model Efficacy:

The comparison between the baseline random policies and our learned policies across different traffic scenarios clearly illustrates the superiority of the reinforcement learning approach. The dramatic reduction in missed opportunities and the near-perfect success rates achieved with the learned policies confirm the efficacy of the RL model in learning and applying complex decision-making strategies. It highlights the potential of such systems to be implemented in real-world autonomous driving applications, where decision-making accuracy is critical for safety and efficiency.

Key observations include:

- **Adaptive Behavior**: The CAV learned to adapt its strategy based on traffic conditions, effectively identifying and utilizing safe gaps to cross the intersection.
- **Balance Between Safety and Efficiency**: The model successfully achieved a balance, minimizing waiting times at the intersection without compromising safety.
- **Scalability Potential**: The results indicate the potential scalability of the approach to more complex scenarios, including multiple lanes and varied traffic dynamics.

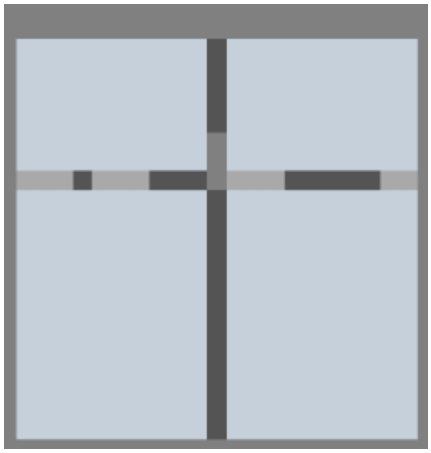

Figure 2: One lane Crossing

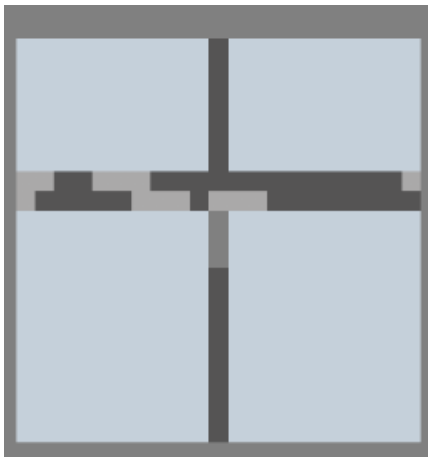

Figure 3: Two lane crossing

5 Future Directions

One significant improvement could involve increasing the resolution of the simulation environment by utilizing a higher number of pixels and shorter time intervals. This adjustment would allow for a more detailed and granular analysis of traffic dynamics and vehicle interactions, potentially leading to even more precise control and decision-making by autonomous agents. Additionally, integrating real-world vehicular constraints, such as acceleration and deceleration dynamics, would provide a more realistic simulation of vehicle behavior. Introducing the requirement for the agent to accelerate from a standstill at intersections gradually would add another layer of complexity and realism to the model, enhancing its applicability to real-world scenarios.

By enhancing the sophistication and accuracy of our system, we aim to evolve its capabilities to manage real-world traffic conditions effectively. To ensure the feasibility and effectiveness of these enhancements, we plan to initially test our refined model on smaller robot agents that emulate similar traffic conditions. This step will serve as an intermediary phase, allowing us to validate and refine the system's performance in a controlled yet realistic setting before transitioning to full-scale real-world applications. These advancements build on the strong foundation laid by the current model, pushing the boundaries of what is possible in autonomous traffic management and extending its potential for integration into actual traffic systems where enhanced safety and efficiency are crucial.

6 Conclusion

In this project, we concentrated our work on developing and analyzing traffic management systems within single-lane and dual-lane configurations, laying a foundation for exploring more complex applications in the future. This work will help us delve deeper into optimizing traffic management systems to harness the potential of connected autonomous vehicles (CAVs). The current results have helped us extract valuable insights into the capabilities of Q-learning methodologies. These insights are instrumental in guiding the development of intelligent and adaptive traffic management solutions, essential for the evolving landscape of urban transportation.

The results obtained from our simulations demonstrate the capabilities of Q-learning in improving traffic management. This advancement is not merely a theoretical achievement; it represents a big step in improving the efficacy and sustainability of urban mobility frameworks. As technology continues to evolve, the insights gained from this work will undoubtedly contribute to the broader application and refinement of intelligent transportation systems, ensuring that urban infrastructure can adapt to and thrive in the face of technological advancements.