A Deep Q-Learning-Based Adaptive Traffic Light Control System for Urban Safety

Ashish Tigga, Lopamudra Hota, Sanjeev Patel and Arun Kumar
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, India 769008
118cs0224@nitrkl.ac.in, 519cs1015@nitrkl.ac.in, patels@nitrkl.ac.in, kumararun@nitrkl.ac.in

***Abstract*—Traffic congestion is a significant and recurring issues in today's urbanised world, caused by an increase in the number of vehicles. While vehicle density fluctuates on temporally short and geographically small scales, efficient traffic signaling system helps in avoiding traffic congestion. An inefficacious traffic system can lead to congestion and delays resulting in high pollution, and fuel wastage. The Deep Reinforcement Learning (DRL) method provides an excellent approach to solve the problem involving complex relations such as traffic flow and congestion. Recent development in Deep Neural Network (DNN) further enhances the learning capabilities of an agent with complex real-time data. The paper presents an intelligent Traffic Light Control System (TLCS) built on a Deep Q-Learning (DQL) model that accurately represents the problem's components: agents, environment, and actions. The proposed model aims to minimize the traffic queue length and delay in terms of waiting time. The model is implemented using Simulation of Urban MObility (SUMO) for traffic generation in an urban scenario. The performance of the proposed model is compared with a traditional traffic light control system. The simulation results show that the proposed DQL-based model can significantly reduce the delay compared with the traditional model.**

***Index Terms*—Traffic light, Urban mobility, Congestion, Deep Reinforcement Learning, Deep Q-learning**

I. Introduction

A rapidly increasing urban population demands good transportation infrastructure of public transportation systems, road networks, and metros. Expanding cities require a good amount of travelling distance for a daily routine of an urban resident [1]. Although public transportation systems have improved in many countries, a significant part of cities population still prefer the convenience of private vehicles [2]. An increasing number of personal vehicles on the road involves primary concerns of environmental issues. Congestion in city traffic worsens this issue of exhaust emission with more fuel burnt per kilometre [3]. It also affects the Quality of life when people want efficient mobility.

The problem of congestion is firstly due to a poorly designed road network and secondly the number of vehicles on the road. A better traffic signal mechanism can reduce the problem of traffic congestion. Most of the conventional traffic signal systems in the world use a repetitive pattern for the duration of the green signal with a sequence of green signalling the direction at the intersection [5]. These systems do not consider real-time congestion, leading to unnecessary queues in one direction of the junction and no vehicle in another direction. This proposed work implements a system that utilizes real-time traffic data to improve the overall efficiency of the traffic signal. An efficient traffic signal adapts to the traffic conditions to minimize the queue length and waiting time. Various traffic control systems have been developed which utilize the statistical data to adjust the signal timing. However, these systems tend to perform poorly with dynamic traffic [6].

The proposed adaptive Traffic Light Control System (TLCS) learns from the environment using the DQL technique. Reinforcement learning (RL) is a reward-based learning technique. Each State-Action transition leads to a reward used to learn and optimize the model. In RL, the agent, environment, and actions associated with the problem are accurately represented [7]. In this case, the state is the vehicular queue present at the intersection. The DQL model acts and decide to turn a signal green, based on the current state of the junction. DQL is a modified Q-learning algorithm in which a Neural Network is used instead of a Q-table. The proposed work uses SUMO to generate traffic simulation [8].

A. Motivation

Urban mobility is a significant topic for countries with a growing urban population. Transportation is an important aspect of day-to-day urban life. The current transportation system needs improvement to meet the requirement. As private vehicles become more affordable, due to the increase in traffic stress on the current infrastructure is to test with an increasing number of vehicles on the road.

Some improvements can be made like developing more lanes to accommodate more vehicles. However, building more lanes in already planned cities is not feasible. It will require significant relocation, and the result can not guarantee congestion-free roads. Another improvement can be to improve traffic signal control at the intersection. Fixed time signal tends to waste time and contribute in air pollution. This work intends to improve the TLCS to optimize the waiting time at the junction for each vehicle to improve the overall traffic flow for the better transportation time, less congestion, and reduce vehicle emissions.

B. Contribution

Most of the present adaptive traffic light uses complex mathematical analysis to adapt to traffic rate changes. The dynamic change in traffic results in low performance by the existing system. A reinforcement learning adapts better to the traffic changes. The proposed work contributes towards:

1) The proposed TLCS implements the DQL technique to achieve better results. It focuses on adaptability to achieve higher throughput. This system aims to improve the average waiting time of each vehicle at the junction.
2) The proposed work introduces a fair traffic system with random sequence signalling on a fixed time traffic signal.
3) The method in the work is flexible to different environments with few changes. The proposed traffic control system can be implemented with multiple techniques to collect real-time data—for example, Inductive-loop vehicle detectors.

C. Paper Organization

The organization of rest of the paper is; section II presents the related works, followed by proposed work in section III. Section IV demonstrates the results and discussion, and finally section VI presents the conclusion and future work.

II. Related Works

Split Cycle Offset Optimization Technique (SCOOT) and Sydney Coordinated Adaptive Traffic System (SCATS) are two existent traffic control systems, based on complicated mathematical models [1]. However, they suffer from a lack of real-time flexibility and adaptability. As a method of determining the optimal signal cycle time, Webster's method [4] is an analytical approach to determining the least total delay for all vehicles approaching the intersection. Self-organizing Traffic Signal (SOTL) is a vehicle actuated control method. With self-organizing control, the signal controllers at each intersection communicate with their neighbours and use this input to organize themselves organically [5].

RL-based methods observe numerous aspects of the intersection's traffic and respond to the present situation. Some employ computer vision to detect the length of a backlog of automobiles [9]. Other methods used are based on inductive-loop sensors to detect a stationary vehicle. All these methods effectively calculate the data as queue length waiting time [10], [11]. The goal of the agents is to maximize the throughput and minimize the waiting time [12], [13]. Most RL-based traffic signal control methods aim to reduce the average vehicle travel time. However, improving travel time is sometimes difficult since it is a long-term measure which dependent on a succession of actions [10]. By optimizing green signal timing in an urban arterial road network, Q-learning reduces travel time and vehicle delays. The reward is often designed from traffic metrics like waiting time and vehicle speed. Previous research has looked into two types of reinforcement learning methods: value-based and policy-based models. In previous studies, the value-based DRL was the most commonly used. Using sampled traffic state/control inputs and associated traffic system performance outputs, Li et al./cite[b10] developed a Dense Neural Network to estimate reinforcement learning's Q-function. Their study found that DRL outperformed traditional reinforcement learning in terms of queue length.

III. Proposed Work

The proposed work focuses on developing an adaptive TLCS for intersections. The method used for this is RL. DQLs determine the best course of action based on intersection state.

The system counts the vehicles on each arm of the intersection using an inductive loop sensor to formulate the current environment state. A traffic simulator collects the data and evaluate the proposed model. The model aims to minimize the waiting time of the vehicles.

A. Problem Statement

The problem with the conventional fixed-sequence traffic signal is that they are inefficient in many cases: Traffic lights can glow green without waiting in a direction; it will glow for a fixed amount of time, subsequently increasing congestion in other directions. The other adaptive traffic signal discussed earlier performs poorly in dynamic traffic conditions. The problem caused by an inefficient traffic control signal includes-

- Congestion on one side of the intersection, uneven traffic flow.
- Increase in average traffic delay for the vehicles.

B. Approach

The DQL approach uses a Q-learning algorithm with Deep Neural Network (DNN). It is a reinforcement learning technique, where an agent learns how to acquire the goal through interaction with an environment; If the model's action contributes to the goal, the model will be rewarded, as shown in Fig.1.

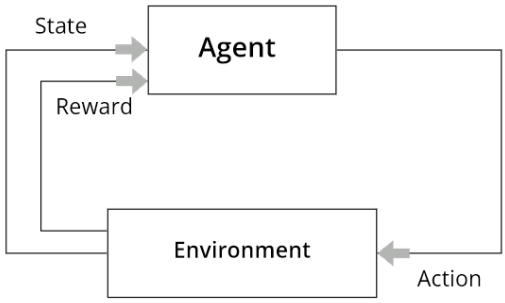


Fig. 1: Block Diagram of Reinforcement Learning Technique.

This is done in order to guide the model on the correct path. If it does an activity that does not lead to the goal, it receives a negative reward to prevent it from learning in the incorrect way. The Q-value is associated with a state-action pair to determine how good the action is on the state. Agent

learns by updating a Q table of state-action pairs using the Bellman equation given in equation 1.

$$\underbrace{\text{New}Q(s,a)}_{\text{New Q-Value}} = Q(s,a) + \underbrace{\alpha}_{\text{New Q-Value}}[\underbrace{R(s,a)}_{\text{Reward}} + \underbrace{\gamma}_{\text{Discount rate}} \overbrace{\max Q'(s',a')}^{\text{Maximum value of predicted reward}}] \quad (1)$$

The DQL model uses a DNN instead of the q-table. With each action, the neural network inputs the state and outputs the q-value. The highest q-value is the best-known action. The action is chosen using the epsilon greedy exploration strategy and update the network's weights using the Bellman equation. ϵ-greedy is an exploration strategy in RL that takes an exploratory action (most unlikely action) with ϵ probability and a greedy action (most likely action) with probability $1-\epsilon$.

The work consists of traffic simulation using Simulation of Urban Mobility (SUMO) and implementing the DQL model that interacts with SUMO.

SUMO is used in this work to simulate four way intersection with four incoming lanes in each arm as shown in Fig.2.

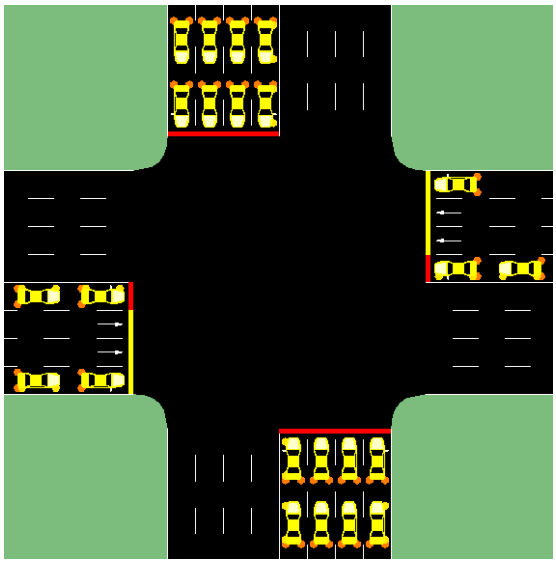


Fig. 2: Four-way Intersection simulation using SUMO

TraCI is used as interface between the program and SUMO application. The environment has the following condition:

- The environment is left-hand drive.
- The intersection considered is isolated, so adjacent intersections are not considered.
- The leftmost lane is for left turn only. The middle two lanes are for going straight. The rightmost lane is for a right turn or going straight.
- The leftmost lane has an independent signal from the other three lanes.

SUMO is configured to generates vehicles as per Weibull distribution, rapid increase in the start and a gradually decreasing till the end.

The environment is the intersection and the vehicles on it, the agent in the proposed model is a traffic light control system and the state is defined by the presence of a vehicle in a predetermined location that is spread across the incoming lanes.

Action is signal to turn green. In environment considered, there are eight signals, and they turn green in pairs, so the total number of actions is four. The reward equals a change in cumulative waiting time after each action.

The Deep Q-Network (DQN) is a fully connected neural network with input as states and output as Q-value for the actions. The action with the highest Q-value is most likely. Experience replay mechanism stores state, action, Q-value and next state calculated using the current neural network with Epsilon Greedy strategy. The memory from the experience replay is used to train the neural network at the end of each episode. An episode is a sequence of state action transitions till the terminal state.

The proposed model includes two neural networks, one the primary network and the target network. The primary network predicts the Q-value for the current state for each sample in the memory (state, action, reward, next state). The target network predicts the best Q-value on the next state. The primary network is updated and trained at each step, but the target network is not updated till the end of the episode. This is done to avoid chasing after the changing Q-value..

The current Q-value calculated is updated with the reward and Q-value of the next state according to the Bellman equation and the network is trained for the updated Q-value as given in equation 2.

$$\underbrace{\text{New}Q(s,a)}_{\text{New Q-Value}} = \underbrace{Q(s,a)}_{\text{Q-value(Primary NN)}} + \underbrace{\alpha}_{\text{Learning rate}}[\underbrace{R(s,a)}_{\text{Reward}} + \underbrace{\gamma}_{\text{Discount rate}} \overbrace{\max Q'(s',a')}^{\text{Next state Max(Q) in Target NN}}] \quad (2)$$

The reward in proposed model is cumulative waiting time. The queue length is calculated for each step of an episode. Queues length is defined as the total time a vehicles wait at a particular step. The sum of queue length is equal to the cumulative waiting time for all the vehicles. Algorithm 1 describes the method used in this approach. Fig.3 shows the flow diagram of the adaptive TLCS training procedure. Algorithm 1 explains the training process and the bellman equation to update Q-value for training.

IV. Results and Discussion

SUMO is used with the TraCI interface to retrieve the real-time data from the simulator. The implemented model has a fixed state size. Fig. 4 shows all the possible actions which is phases of green signal; North South Left (NSL), North-South (NS), East-West (EW) and East West Left (EWL).

The training hyper-parameters are presented in Table 1. There are 100n iterations, where each iteration runs for 5400secs. Cars are generated per iteration and thus here 1000

Algorithm 1 DQL algorithm for TLCS

Definition:
E = Maximum number of episode
max_steps = Maximum numer of steps per episode
Q = Q-value in primary neural network
$\bar{Q}$ = Q-value in target neural network
R = Experience replay memory
ϵ = for epsilon-greedy exploration
Initialization:
$Q \leftarrow$ Initialization of primary Q-value with random weights w
$\bar{Q} \leftarrow$ k Initialization of target Q-value with weights $\bar{w} = w$
for $episode = 1, E$ **do**
    Generate traffic simulation with seed = $episode$
    $\epsilon \leftarrow$ 1- $epsiode/E$
    $step \leftarrow 0$
    while $step < max\_steps$ **do**
        $S \leftarrow$ Current state from SUMO using TraCI
        $current\_wait\_time \leftarrow$ Waiting time from SUMO using TraCI
        $reward \leftarrow old\_wait\_time - current\_wait\_time$
        $R \leftarrow$ add_sample$(old_state, old_action, reward, S)$
        $A \leftarrow$ With probability ϵ choose random action OR choose $argmax_a(Q)$
        Simulate the yellow signal steps if $A \neq old\_action$
        Simulate the A green signal steps in SUMO
        $old\_state \leftarrow S$
        $old\_action \leftarrow A$
        $old\_wait\_time \leftarrow current\_wait\_time$
        $sum\_reward \leftarrow sum\_reward + reward$
        $cumulative\_wait\_time \leftarrow cumulative\_wait\_time + current\_wait\_time$
    end while
    for $epoch = 0, training\_epochs$ **do**
        $batch \leftarrow$ get_samples$(R, batch_size)$
        $states, next\_states \leftarrow batch$
        $Q\_state \leftarrow model.predict\_batch(states)$
        $Q\_next\_state \leftarrow target\_model.predict\_batch(next\_states)$
        for i, b in $batch$ **do**
            $state, action, reward \leftarrow b[0], b[1], b[2]$
            $current\_q \leftarrow Q\_state[i]$
            $current\_q[action] = reward + \gamma * argmax_a(Q\_next\_state[i])$
            $x[i] \leftarrow state$
            $y[i] \leftarrow current\_q$
        end for
        model.train(x,y)
    end for
    $w \leftarrow model.get\_weights$
    $target\_model.set\_weights(w)$
end for

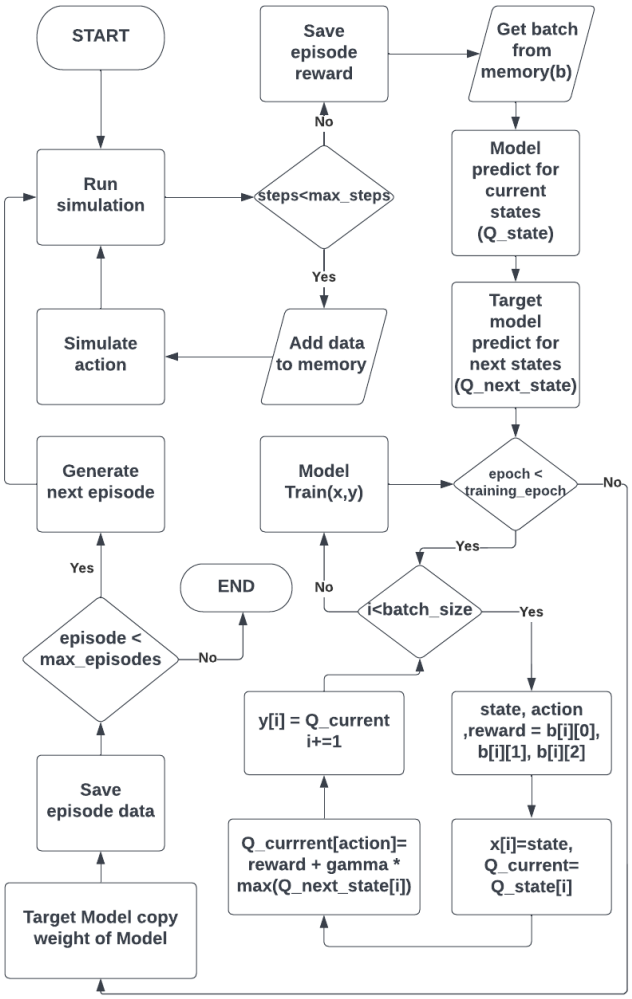
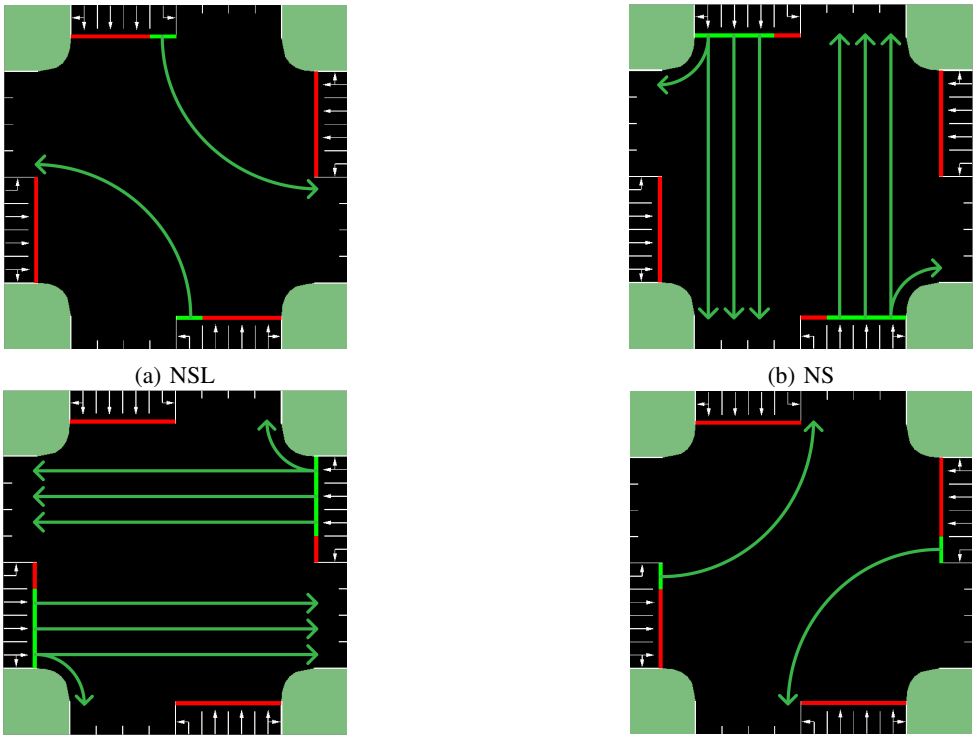


Fig. 3: Flow diagram of the model training process using bellman equation

cars are generated in 5400secs. For the training process the batch size is taken to be 64. The replay memory size i.e. the queue containing the agent experience is taken to be 50000. The hidden layer is assumed to be 4 for getting the optimal result. As for data with large dimensions and features the optimal hidden layer is 3-5, and increasing it increases the complexity.

The model is trained on the data generated by the SUMO. The traffic condition is generated from Weibull-distribution as illustrated in the Fig.5. The training reward with each episode is illustrated in Fig.6 which shows the reward converges to around 5000 negative reward after 100 episodes. The cumulative reward increases as the model learns. Also, the average queue length decreases with each training episode demonstrated in Fig.7. The proposed model converge towards positive reward at around 100 episode.

a) Performance analysis:: The proposed method is compared with a fixed time cyclic signal under the same traffic



(a) NSL (b) NS

(c) EW (d) EWL

Fig. 4: Actions: Phases of green signal for all directions

TABLE I: Training hyper-parameters used in the model

Parameters	Value	Description
ϵ	1-0	Epsilon reduces for each episode starting from 1
Total episodes	100	Number of episodes the model is trained
Max. steps per episode	5400	One step equals one second in simulation
Cars generated per episode	1000	No. of cars passing the intersection per episode
Number of hidden layers	4	Hidden layers in the dense network
Width of each layer	120	Number of neurons per layer
Training epochs	800	Total training epochs per episode
Batch size	64	Batch size of samples taken from memory
Memory size	50000	Maximum size of the replay memory
Input layer width	80	Input size equals size of states
Output layer width	4	Output size equals number of actions

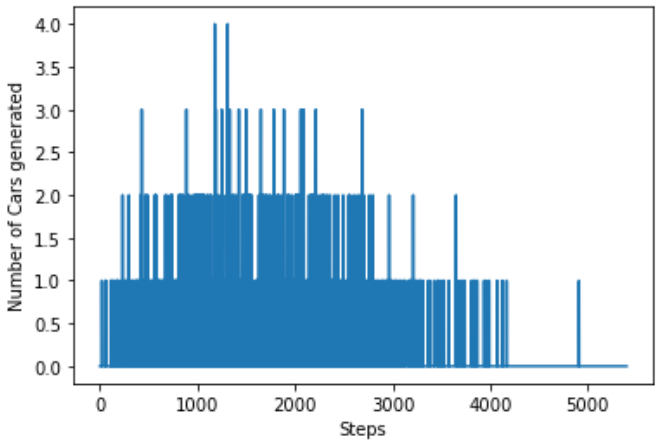


Fig. 5: Traffic generation using Weibull distribution for each episode

condition generated by SUMO. The model is tested for 100 episodes of unique seed and the queue length and cumulative delay is recorded.

The DQL model outperforms the other method with reduced queue length and delays time, as illustrated in Table 2. In proposed method of data collection queue length is directly related to cumulative waiting time for all vehicles. The results show there is approximately 32% improvement of DQL proposed model from traditional model in terms of waiting time and queue length.

TABLE II: Result comparison between simple and DQL TLCS for 100 episodes

Method	Average queue length	Average waiting time(s)
DQL TLCS	2.36	12.76
Simple TLCS	3.46	18.68

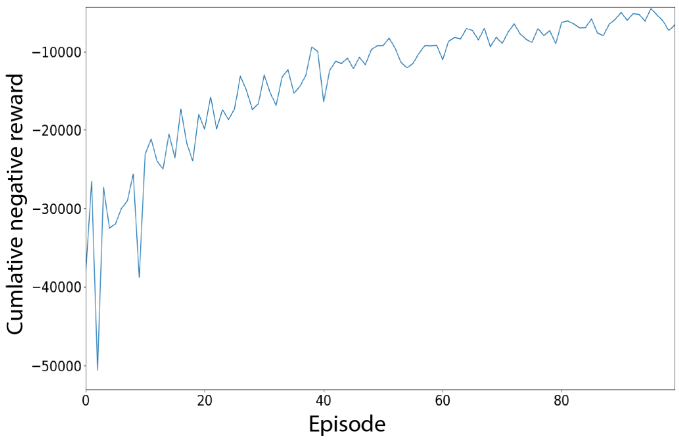


Fig. 6: Training: Cumulative reward

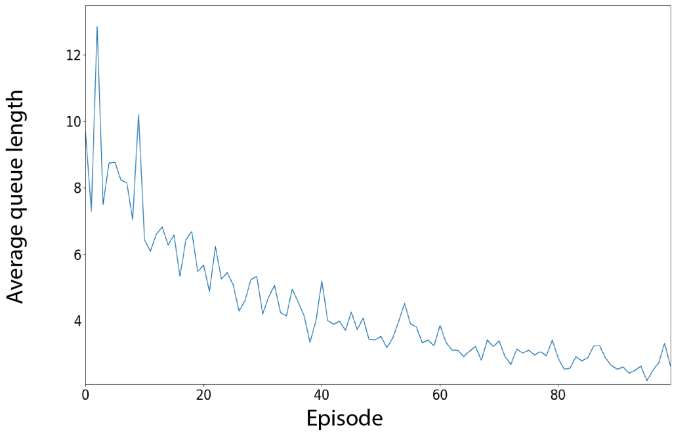


Fig. 7: Training: Average queue length

V. Conclusion and Future Work

This paper proposes an adaptive TLCS based on deep reinforcement learning. The Agent learns to choose optimal action using the deep Q-learning technique to reduce queue and waiting time for the incoming vehicles. The proposed model compares with the traditional traffic light control system; it results in reduced waiting time and queue length.

Future works aim to improve the Deep Q-learning methods with other reinforcement learning techniques. The work can be extended by developing real-world robust data collection techniques. As, the proposed work focuses on a single intersection environment, evaluating the model against a network of intersections can result in performance issues.

References

[1] Stevanovic, Aleksandar, Cameron Kergaye, and Peter T. Martin. "Scoot and scats: A closer look into their operations." In 88th Annual Meeting of the Transportation Research Board. Washington DC. 2009.

[2] Linda, S. T. E. G. "Can public transport compete with the private car?." IATSS research 27, no. 2 (2003): 27-35, https://doi.org/10.1016/S0386-1112(14)60141-2.

[3] Bharadwaj, Shashank, Sudheer Ballare, and Munish K. Chandel. "Impact of congestion on greenhouse gas emissions for road transport in Mumbai metropolitan region." Transportation Research Procedia. 25 (2017): 3538-3551.

[4] Webster, F.V. Traffic Signal Settings; Technical Report; Transportation Research Board: Washington, DC, USA, 1958.

[5] Cools, Seung-Bae, Carlos Gershenson, and Bart D'Hooghe. "Self-organizing traffic lights: A realistic simulation." In Advances in applied self-organizing systems, pp. 45-55. Springer, London, 2013.

[6] C. Chen, H. Wei, N. Xu, et al. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In AAAI, 2020.

[7] Genders, Wade, and Saiedeh Razavi. "Using a deep reinforcement learning agent for traffic signal control." arXiv preprint arXiv:1611.01142 (2016).

[8] Lopez, Pablo Alvarez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. "Microscopic traffic simulation using sumo." In 2018 21st international conference on intelligent transportation systems (ITSC), pp. 2575-2582. IEEE, 2018.

[9] Zaatouri, Khaled, and Tahar Ezzedine. "A self-adaptive traffic light control system based on YOLO." In 2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), pp. 16-19. IEEE, 2018.

[10] Balaji, P. G., X. German, and Dipti Srinivasan. "Urban traffic signal control using reinforcement learning agents." IET Intelligent Transport Systems 4, no. 3 (2010): 177-188.

[11] Ikidid, Abdelouafi, Abdelaziz El Fazziki, and Mohammed Sadgal. "Multi-agent and fuzzy inference-based framework for traffic light optimization." International Journal of InteractiveMultimedia and Artificial Intelligence (2021).

[12] Wei, Hua, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. "Intellilight: A reinforcement learning approach for intelligent traffic light control." In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2496-2505. 2018.

[13] Xu, Ming, Jianping Wu, Ling Huang, Rui Zhou, Tian Wang, and Dongmei Hu. "Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning." Journal of Intelligent Transportation Systems 24, no. 1 (2020): 1-10.

[14] Li, Li, Yisheng Lv, and Fei-Yue Wang. "Traffic signal timing via deep reinforcement learning." IEEE/CAA Journal of Automatica Sinica 3, no. 3 (2016): 247-254.