Non-cooperative Automated Merging in Autonomous Driving for Highways Using Deep Reinforcement Learning

Amin Tabrizian [1]

Abstract

Despite the recent successes of AI in autonomous driving, on-ramp merging remains a major challenge. In this paper, we propose a deep reinforcement learning approach based on DQN for safely merging a vehicle onto a highway in a non-cooperative environment. Our online merging planning algorithm considers both minimum speed changes and safe distances to other vehicles. The results of our experiments demonstrate the strong performance of the proposed algorithm.

1. Introduction

Autonomous driving technology has made significant progress in recent years, with many advanced AI algorithms achieving impressive results in a variety of driving tasks. However, one area where autonomous vehicles still struggle is on-ramp merging, where a vehicle must safely enter a highway from an on-ramp. This is a complex task that requires the vehicle to carefully plan its trajectory and speed, taking into account the surrounding traffic and the constraints of the road. There are multiple research studies that are focused on different aspects of this driving task.

In (Chen et al., 2021), a multi-agent reinforcement learning (MARL) approach is proposed for a high-way on-ramp merging where both autonomous vehicles (AVs) and human-driven vehicles (HDVs) exist. The proposed algorithm was able to outperform the existing works in terms of collision rate and training efficiency.

Some other research studies focused on cooperative planning in the highway merging scenario. A Monte-Carlo tree search (MCTS) approach was proposed in (Lenz et al., 2016) for 3 different merging scenarios. It was shown that the algorithm is able to generate cooperative behaviors that can be observed as human driving styles. However, the results and evaluation of the performance of the algorithm were not discussed in the paper. An optimal trajectory planning algorithm was proposed in (Ntousakis et al., 2016) based on the minimization of vehicle acceleration, jerk, and its first derivative for ensuring passenger comfort and less engine effort in a cooperative single-lane highway vs. single-lane on-ramp merging scenario. The results demonstrated the strength of the proposed method in an automated merging procedure where a set of vehicles are under merging control. Extending the state-action space of the merging task was the aim of some other studies. In (Schester & Ortiz, 2021), a deep MARL approach was presented for a continuous space of states and actions for the problem of highway on-ramp merging. They claimed that the proposed algorithm was essentially optimal in the context of the evaluation environment. Another anti-jerk DDPG-based approach was presented by (Lin et al., 2019) for both ensuring collision avoidance and passenger safety.

In this paper, we propose a DQN-based approach for safe on-ramp merging in a non-cooperative scenario, where the mainstream vehicles do not have a cooperative manner for the merging task. In this scenario, all of the mainstream cars have a constant random speed regardless of the position of the merging vehicle. The merging vehicle has to adjust its speed to ensure both safety and less engine effort for the merging task. The contributions of this research are

1. Completing the merging task in a non-cooperative environment, where agents do not have a cooperative manner.

2. Imposing uncertainty in the parameters of the vehicles in each episode.

The structure of the paper is as follows. In Section 2, we will cover reinforcement learning (RL) preliminaries and DQN formulations. The problem framework will be introduced in Section 3. Then, the results will be discussed in Section 4, and we will conclude the paper in Section 5.

2. Reinforcement Learning Preliminaries

In this section, RL preliminaries and DQN will be briefly discussed.

[1]Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, D.C. Correspondence to: Amin Tabrizian <amin_tabrizian@gwu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

2.1. Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning that focuses on learning how to make optimal decisions in a given environment through trial and error. The goal of RL is to find the best policy, or sequence of actions, that maximizes the expected cumulative reward over time. In RL, an agent interacts with its environment by selecting actions from a set of possible actions, and receives rewards based on the actions it takes. The state of the environment at any given time is represented by a vector of features, and the actions taken by the agent determine the next state of the environment. The agent uses the rewards it receives to learn the optimal policy for maximizing its cumulative reward. The key components of RL are the state, action, and reward. The state represents the current state of the environment, and the action is the decision made by the agent based on the current state. The reward is the feedback provided by the environment based on the action taken by the agent. The agent uses the rewards to update its policy and learn the optimal actions for maximizing its cumulative reward. Q-values, also known as action-values, represent the expected future reward for taking a given action in a given state. The Q-value for a given state-action pair is calculated using the Bellman equation:

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

where s and a are the current state and action, r is the reward received, s' is the next state, and γ is the discount factor that determines the importance of future rewards. The agent uses the Q-values to select the action with the highest expected future reward. The optimal policy, π^*, is defined as the policy that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}[R_t|\pi]$$

where R_t is the cumulative reward at time step t, and π is the policy.

2.2. Deep-Q Network

DQN is a deep learning algorithm used for reinforcement learning which was proposed by Minh et al. (2013). It combines a deep neural network with Q-learning to find an optimal action-selection policy. To improve stability and performance, DQN uses fixed-target networks and experience replay. The former decouples the training of the Q-network from its evaluation, while the latter improves the efficiency of training by sampling randomly from a replay buffer of experiences. These techniques help make DQN a more robust and effective solution for reinforcement learning problems. The loss function used in DQN is the mean squared error (MSE) between the predicted and target values. This loss function is defined as follows:

$$L(\theta) = \frac{1}{N} \sum (Q_{predicted}(s,a) - Q_{target}(s,a))^2$$

Table 1. Parameter values for the highway and vehicles

PARAMETER	VALUE
MAXIMUM ACCELERATION a_{max}	$5\frac{m}{s^2}$
MAXIMUM DECELERATION a_{min}	$6.5\frac{m}{s^2}$
MAXIMUM ROAD SPEED LIMIT V_{max}	$25\frac{m}{s}$
MINIMUM ROAD SPEED LIMIT V_{min}	$5\frac{m}{s}$

where $L(\theta)$ is the loss function, N is the total number of experiences used in training, $Q_{predicted}(s,a)$ is the predicted value for a given state-action pair, and $Q_{target}(s,a)$ is the target value for the same state-action pair. The goal of training is to minimize the loss function, by adjusting the network parameters to reduce the error between the predicted and target values.

3. Merging Problem Formulation

3.1. Merging Environment

For this problem, the merging environment is created in the Simulation of Urban Mobility (SUMO) driving simulator (Krajzewicz et al., 2002). The environment consists of a single-lane highway and an on-ramp. The schematic representation of the merging scenario is in Figure 1.The vehicles on the highway have a constant speed which is randomly selected between a lower and an upper bound in each episode. The initial speed of the ego vehicles is also selected randomly within a predefined limit. The safety rules for all vehicles are terminated to ensure having a correct training with making collisions.

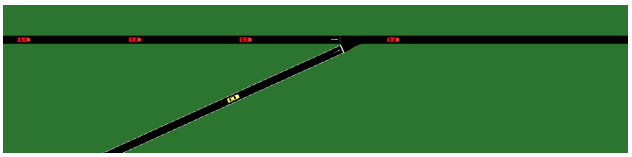

Figure 1. Entrance ramp road scene: automated driving vehicle (yellow), referred as the ego vehicle; main traffic vehicles (red). The ego vehicle is under the traffic controller but other vehicles have a constant random speed.

The highway and vehicle parameters are shown in Table 1.

3.2. Environment State

The environment state of the framework consists of 18 different values, the position and velocities of all of the existing vehicles.

$$s = [x_{v_0}, y_{v_0}, ..., x_{v_5}, y_{v_5}, V_{v_0}, ..., V_{v_5}]$$

Table 2. DQN Hyper Parameters

PARAMETER	VALUE
HIDDEN LAYERS' SIZE	256
LEARNING RATE α	0.001
BATCH SIZE	64
DISCOUNT FACTOR	0.99
EPSILON START	1
EPSILON END	0.01
EPSILON DECAY RATE	5×10^{-5} FOR $v_{1,\ldots,3}$ AND 10^{-5} FOR v_4

3.3. Action

The action space for the ego vehicle is to either increase/decrease its speed by a value of 1 or not change the previous speed.

3.4. Reward

In order to ensure both safety and efficiency of the merging task, a multi-objective reward function is designed for this problem. There will be a penalty of 1 for changing the speed, a penalty of 15 or 30 for violating the safe distance between the ego and front/back vehicle, and a penalty of 100 for making a collision. The reward function is provided as follows.

$$R(s,a) = \begin{cases} -1 & \text{if } a = \pm 1 \\ -15 & \text{if } d_1 < d_{safe} \\ -30 & \text{if } d_1 \text{ and } d_2 < d_{safe} \\ -100 & \text{if collision} \end{cases}$$

In the above equation, d_1, d_2 are the smallest distances between the ego and mainstream vehicles, and d_{safe} is the minimum safe distance that is defined externally.

3.5. DQN Structure and Hyper Parameters

A four layer neural-network is used for estimating the Q-values. All of the experiences are stored in a replay buffer and used randomly for training the Q-networks. The network structure and hyper parameters are stated in Table 2.

4. Results

4.1. Training

The DQN agent was trained in 4 different versions of the merging scenario for $20,000$ episodes. The cumulative reward of these training episodes are provided in 2. Here is a brief description of each version.

- Version 1: Initial speed of mainstream vehicles are sampled from a uniform distribution between $[12, 15)\frac{m}{s}$. The ego vehicle initial speed is sampled in a same manner between $[11, 14)\frac{m}{s}$.
- Version 2: Initial speed of mainstream vehicles are sampled from a uniform distribution between $[13, 16)\frac{m}{s}$. The ego vehicle initial speed is sampled in a same manner between $[11, 14)\frac{m}{s}$.
- Version 3: Initial speed of mainstream vehicles are sampled from a uniform distribution between $[12, 15)\frac{m}{s}$. The ego vehicle initial speed is sampled in a same manner between $[9, 12)\frac{m}{s}$.
- Version 4: Initial speed of mainstream vehicles are sampled from a uniform distribution between $[12, 15)\frac{m}{s}$. The ego vehicle initial speed is sampled in a same manner between $[11, 14)\frac{m}{s}$. The epsilon decay rate is different in this version as stated earlier.

It is worth noting that version 2 and 3 are more difficult scenarios because of the faster speed in mainstream vehicles and slower speed in ego vehicle respectively.

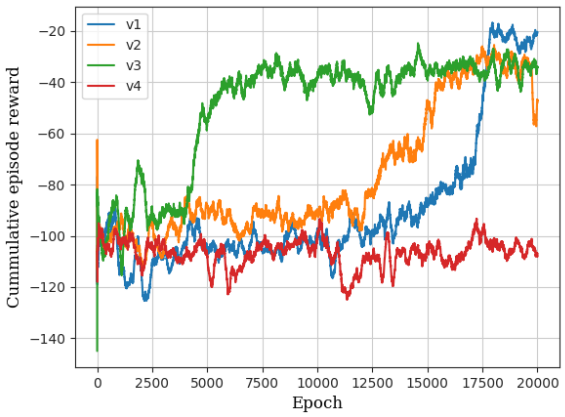

Figure 2. Cumulative episode rewards for training episodes.

4.2. Testing

After all agents were trained, they performance were evaluated on 300 more episodes with the same conditions. The results are represented in Table 3. As mentioned earlier version 2 and 3 are more challenging comparing to other versions, which is obvious in the collision rate of the testing episodes. Version 4 did not demonstrate an acceptable performance perhaps because of the excessive amount of exploration rate. Except the 4th version, all other agents demonstrated a strong performance considering the uncertainties and non-cooperativeness of the scenario.

Table 3. Testing Results

AGENT	SUCCESS RATE (%)
VERSION 1	**94.00**
VERSION 2	92.67
VERSION 3	91.67
VERSION 4	28.00
RANDOM	18.00

As a sample, a time speed of the first agent in one of the testing scenarios are shown in Figure 3. In this episode, ego vehicle will first slow down to reach the appropriate time slot for merging. After reaching to the merging point, it will speed up to avoid making collision with the rear vehicle.

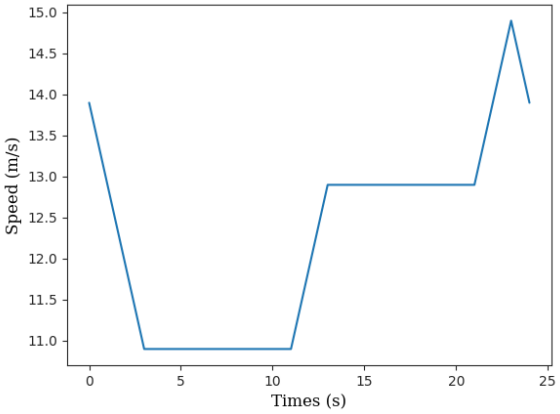

Figure 3. Time speed of the ego vehicle in one testing scenario

5. Conclusion

A DQN-based algorithm for a non-cooperative merging scenario was proposed in this paper. The results demonstrated the strength and robustness of the proposed algorithm for the defined merging scenario. Here are some possible contributions for future works:

1. Expand the existing model to environments with higher dimensions.

2. Limit the state information to the position and velocity of a certain number of vehicles.

3. Update the reward function to ensure passenger comfort.

4. Explore the scalability and safety of the current algorithm in new environments.

References

Chen, D., Hajidavalloo, M., Li, Z., Chen, K., Wang, Y., Jiang, L., and Wang, Y. Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic, 2021. URL https://arxiv.org/abs/2105.05701.

Krajzewicz, D., Hertkorn, G., Feld, C., and Wagner, P. Sumo (simulation of urban mobility); an open-source traffic simulation. pp. 183–187, 01 2002. ISBN 90-77039-09-0.

Lenz, D., Kessler, T., and Knoll, A. Tactical cooperative planning for autonomous highway driving using monte-carlo tree search. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 447–453, 2016. doi: 10.1109/IVS.2016.7535424.

Lin, Y., McPhee, J., and Azad, N. L. Anti-jerk on-ramp merging using deep reinforcement learning, 2019. URL https://arxiv.org/abs/1909.12967.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.

Ntousakis, I. A., Nikolos, I. K., and Papageorgiou, M. Optimal vehicle trajectory planning in the context of cooperative merging on highways. *Transportation Research Part C: Emerging Technologies*, 71:464–488, 2016. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2016.08.007. URL https://www.sciencedirect.com/science/article/pii/S0968090X16301413.

Schester, L. and Ortiz, L. E. Automated driving highway traffic merging using deep multi-agent reinforcement learning in continuous state-action spaces. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 280–287, 2021. doi: 10.1109/IV48863.2021.9575676.