Deep Q Network Safety Decision-making Model of Autonomous Vehicle Based on Trajectory Prediction

Shucheng Zhang
MEMS Department
Duke University
shucheng.zhang@duke.edu

Abstract

To overcome the shortcomings of handcrafted decision methods in field of autonomous vehicle, we developed a Safety Decision-making Model based on Deep Q Network (DQN) to complete safe and high-speed driving in the highway environment. The experiments showed that our model greatly increased average speed while keeping vehicles safe. Moreover, we added the predicted trajectories of surrounding vehicles into the original input and proved their importance in improving the risk forecast ability. More details are shown in https://github.com/Shuchengzhang888/Safety-Decision-Model.

1 Introduction

In recent years, autonomous vehicles have been one of the most popular research directions in academics and industry because of their huge potential in reducing travel costs and increasing driving safety [1]. According to W. D. Montgomery's estimation [13], the widespread application of autonomous vehicles is expected to have annual social benefits of nearly 800 billion in the aspect of congestion mitigation, road casualty reduction, decreased energy consumption, and increased productivity caused by the reallocation of driving time. However, robust automated driving in unknown environments has not been achieved yet. One of the most significant concerns is how to guarantee every decision the self-driving agent makes is safe, in the case of rapidly changing driving surroundings and the influence of other road users [2].

Traditionally, the criterion of safety is based on handcrafted rules [3]. This method requires people to foresee every potential traffic situation, and code how to handle them in advance, which is time-consuming. More importantly, the decision system of the agent cannot learn from past experience and is not robust in new unknown environments. The appearance of reinforcement learning brings a new solution to this problem, due to its self-learning ability. Thus, in this project, we developed a safety decision-making model based on Deep Q Network (DQN) to achieve collision avoidance and high-speed driving in a highway environment [14].

Also, as an important part of the autonomous vehicle perception module, the information on objects' behavior tracking and prediction has been proven to be important in safety decision-making, and as input, it is widely used in the decision-making models of many large-scale self-driving vehicle platforms, like Apollo or Waymo [15]. However, so far, the prediction information of surrounding objects is rarely used in reinforcement learning models. We believe that the lack of this information will make the model lose the ability to predict risks in advance and change strategies on time. Therefore, in this project, we use the prediction information of the surrounding object's trajectories as input to train the model to study whether it will improve the robustness of the model.

The main works of this project are as follows:

- Built a simple highway environment based on OpenAI/Gym for training and testing the reinforcement learning model.
- Implemented DQN algorithm to train safety decision-making model to achieve collision avoidance and high-speed driving in the simulated environment and showed its advantages in increasing average speed while keeping vehicles safe.
- Added predicted trajectories of surrounding vehicles into the original input and proved this kind of prediction will make the model forecast the danger to a certain degree.

2 Related works

A great number of research has been done on the topic of trajectory prediction over a long period of time. Traditional methods include Monte Carlo simulation [16], Bayesian networks [17], and Hidden Markov Models (HMM) [18]. These techniques typically concentrate on analyzing objects based on their prior movements and can only be used in straightforward traffic situations with few interactions between vehicles. However, they may not perform as well in situations with various types of vehicles and pedestrians. In recent years, deep learning-based methods, like Long Short-term Memory (LSTM) and Graph Convolutional Networks (GCN), have been proposed for trajectory prediction. Xin Li proposed an encoder-decoder GRU-based model that can simultaneously predict the trajectories of all observed objects by using a graph to represent the interaction between all nearby objects.

Currently, most of the research on safe autonomous driving is based on the rule method. The rule method makes decisions and limits driving behaviors to keep safe by using human-made empirical criteria, such as maximum speed, and minimum following distance [3]-[5]. For instance, [10] tries to define the broad safety standards that an autonomous vehicle must meet. These requirements are referred to as Responsibility, Sensitivity, and Safety (RSS). However, there is no assurance that rule-based techniques would stop undesired behaviors in a highly dynamic and changing environment. Additionally, the rule-based method cannot be learned from different situations and there is a terrible performance in unseen scenarios. Therefore, because of the lack of adaption, the rule-based method can just be the baseline to keep the autonomous vehicle out of danger. A more advanced decision-making technologies should be developed to suit different driving situations.

Reinforcement learning (RL), as a learning-based approach, has got a huge success in robotics control and path planning, because complete knowledge of the environment is not necessary and experiences in new environments can be learned [6]-[8]. With the combination of deep learning, the DQN network [11] solved the problem in which high dimensional data cannot be the input. Xi Xiong developed a decision-making system by combining deep reinforcement learning and safety-based control [19]. They first use the DDPG algorithm to get the driving policy using partial state inputs and then combine the policy network and safety-based control, including artificial potential field and path tracking, to avoid collision and drive following the lane line. The three algorithms work well together in the simulator, according to tests.

3 Methods

3.1 Reinforcement Learning and Deep Q Network

In this project, we formalize the safety decision-making model as a Markov Decision Process (MDP) [21]. The optimal action-value function, $Q^*(s,a)$, follows Bellman equation. This is described by,

$$Q^*(s,a) = \underset{s'}{\mathbb{E}}\left[r + \gamma \max_{a'} Q^*\left(s', a'\right) \mid (s,a)\right].$$

In the DQN algorithm [13], the optimal action-value function is combined with deep learning. we define the deep neural network with weights θ as a function approximator of the optimal value function. In this case, the network is then trained by adjusting its weights, θ_i, at every iteration, i, to minimize the error in the Bellman equation. This process can be done by stochastic gradient descent and the loss function at iteration i will be defined as

$$L_i\left(\theta_i\right) = \mathbb{E}_{\mathrm{M}}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right) - Q\left(s, a; \theta_i\right)\right)^2\right]$$

During the training process, the actions of each interaction will be selected by ϵ-greedy policy, which means there are ϵ probability that the action will be selected randomly, $1-\epsilon$ probability that the the highest rewarded action will be implemented.

3.2 Reference Model

To study the performance of our model, we combined IDM and MOBIL decision models, which are widely used in intelligent vehicle research, as our reference [22, 23]. IDM is commonly used to determine the longitudinal dynamics of a vehicle. In IDM, the agent's velocity v is related to the distance to the vehicle in front d and the velocity difference between the agent and surrounding vehicles Δv. The function is as follows.

$$\dot{v} = a\left(1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{d^*(v,\Delta v)}{d}\right)^2\right)$$

$$d^*(v,\Delta v) = d_0 + vT + v\Delta v/(2\sqrt{ab})$$

Line-changing action is decided by the MOBIL. To be specific, if the agent wants to change line and overtake, the acceleration of rear vehicles should satisfy a safety criterion,

$$\tilde{a}_{\mathrm{e}} - a_{\mathrm{e}} + p\left((\tilde{a}_{\mathrm{n}} - a_{\mathrm{n}}) + (\tilde{a}_{\mathrm{o}} - a_{\mathrm{o}})\right) > a_{\mathrm{th}}$$

where $a_{\mathrm{e}}, a_{\mathrm{n}}$ and a_{o} are the accelerations of the ego vehicle, the rear vehicle in the target lane, and the rear vehicle in the current lane, assuming that the ego vehicle stays in its lane. Also, $\tilde{a}_{\mathrm{e}}, \tilde{a}_{\mathrm{n}}$ and $\tilde{a}_{\mathrm{o}}$ are the accelerations if the agent change its line.

3.3 Agent Module

Perception The input of the safety decision model includes ego vehicle and the top 4 nearest surrounding vehicles' position, velocity, and acceleration of each state. Furthermore, the kinematic predicted positions after t seconds of surrounding vehicles will be added into input of the model. The details are shown as follows.

Table 1: **The input of decision model**

P_xe, P_ye	Ego vehicle's position
V_xe, V_ye	Ego vehicle's velocity
P_xi, P_yi	Surrounding vehicles' relative position $i = 0, 1, 2, 3$
V_xi, V_yi	Surrounding vehicles' relative velocity $i = 0, 1, 2, 3$
P_xt, P_yt	Surrounding vehicles' relative predicted position $t = 1, 2, 3$

In this project, we defined 4 types of the model input. The basic input is composed of ego vehicles' position and speed, and the positions and speeds of the nearest 4 vehicles, with a total dimension of 20. On this basis, by adding the position prediction information of surrounding vehicles in the next t, $2t$, and $3t$, three other sets of inputs with dimensions 28, 36, and 44 are formed. These inputs will be used to study whether the prediction information will affect Model performance has improved.

Decision The safety decision model based on DQN will be trained following the parameters in Table 2 and the architecture shown in Figure 1. Moreover, the output of the model includes 5 different discrete actions. They are shown in the Table 3.

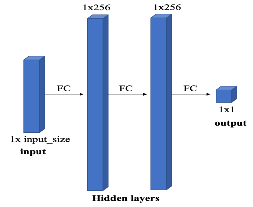

Figure 1: DQN Architecture

Table 2: **Training Details**

Epoch e	200000
Learning rate r	0.0005
Batch size b	256
Discount factor γ	0.8
Target network update frequency	500
Replay buffer size	15000
Learning start step	200

Table 3: **The input of decision model**

Num	Action	Description
0	LANE LEFT	Change to the left lane
1	IDLE	Keep current driving
2	LANE RIGHT	Change to the right lane
3	FASTER	Speed up $a = 5m/s^2$
4	SLOWER	Slow down $a = -5m/s^2$

The reward function is composed of 2 components: high speed reward and collision reward.

$$r = r_{collision} + r_{high\_speed}$$

$$r_{collision} = \begin{cases} -1 & If\ deviates\ from\ the\ road\ or\ collides \\ 0 & Otherwise \end{cases}$$

$$r_{high\_speed} = \lambda \frac{v_{current} - v_{min\_reward}}{v_{max\_reward} - v_{min\_reward}}$$

Where, the $[max\_reward, min\_reward]$ is the reward interval. λ is reward radio. Here, we set it to 0.6

Control We control all vehicles in the simulated environment following the Kinematic Bicycle Model.

$$\begin{aligned} \dot{x} &= v\cos(\psi + \beta) \\ \dot{y} &= v\sin(\psi + \beta) \\ \dot{v} &= a \\ \dot{\psi} &= \frac{v}{l}\sin\beta \\ \beta &= \tan^{-1}(1/2\tan\delta) \end{aligned}$$

where (x, y) is the vehicle position; v its forward speed; ψ its heading; a is the acceleration command; β is the slip angle at the center of gravity and δ is the front wheel angle used as a steering command.

3.4 Trajectory Prediction

As we mentioned, there already have been several ways to predict the trajectories of the surrounding vehicles. Although deep learning-based trajectory prediction methods can generate more accurate predictions in complex traffic environments, their relatively slow response speed and excessive resource occupation are not suitable for billions of training in reinforcement learning. Thus, to simplify the input and accelerate the training process, we implement the kinematic theory to generate prediction information. The kinematic functions are as follows,

$$\begin{aligned} v &= v_0 + at \\ p &= p_0 + vt \end{aligned}$$

Where v_0 and p_0 are the current state velocity and position, v and p are the next state velocity and position.

4 Experiments

4.1 Experiments Setup

Metrics Inspired by [3], we use a self-defined index to evaluate the model's performance.

$$\tilde{p} = (d/d_{\max})\,(\bar{v}/\bar{v}_{\text{ref}})$$

Where, d and $\bar{v}$ are the driving distance and velocity respectively. $d_{\max}$ is the road length. $\bar{v}_{\text{ref}}$ is the driving velocity of reference model.

Setup Both training and testing processes are completed in a 4-ways highway environment. The totall length is $1000m$ and the limited speed is $45mile/h$. The number of NPC, which is controlled by reference model, is 30.

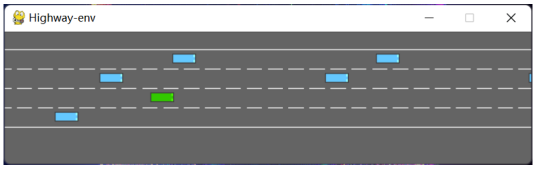

Figure 2: Experiment Environment

4.2 Experimental Result

Compared with reference model The Figure 3 shows our model's performance, compared with reference model. Overall, DQN-based decision models outperform reference models after training for 100,000 epochs. After 200,000 epochs training, its performance index has reached up to 1.208. Figure 4 shows that this performance is mainly due to the higher average speed during driving. It can also be seen from the training results that for the reference model, the handcrafted decision-making function allows it to follow the traffic flow for more time. Although this decision reduces collisions, it also greatly reduces traffic efficiency. However, our model will try to maintain a high speed during the test. Although the number of collisions has increased, we believe that this limit can be improved by optimizing the training process and reward mechanism. (More details can be seen in 'DEMO' folder.)

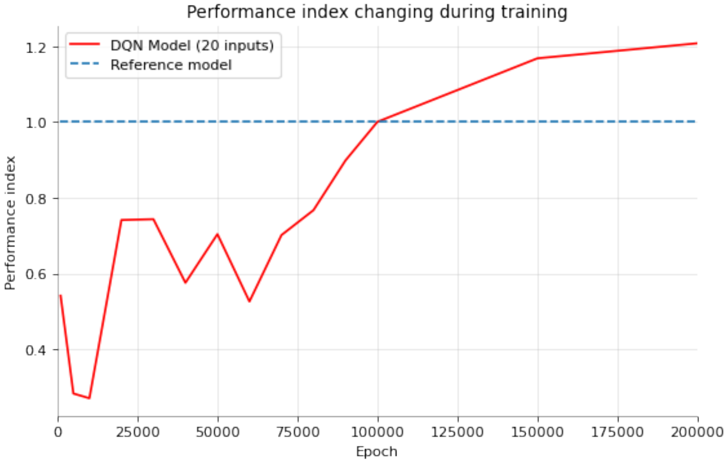

Figure 3: Performance index changing during training

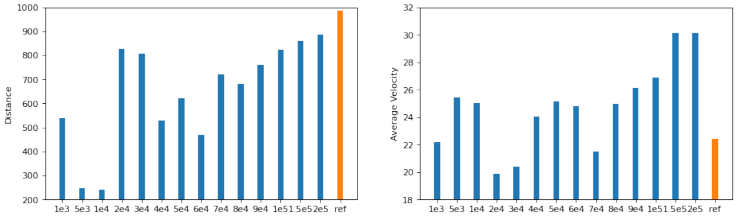

Figure 4: Average maximum driving distance and average velocity changing during training

Compared with adding predicted information Figure 5 shows the average speed and average driving distance of the vehicles after taking the predicted trajectories of different stages of the surrounding vehicles as input. We found that after adding the forecast information of t, the performance of the model will be slightly improved; after adding $2t$ and $3t$, there will be a slight decrease. We believe that this result shows the potential of using prediction information as input in vehicle safety decision-making, but more advanced model structures and training methods, such as CNN, should be used to enable the model to extract prediction information more accurately.

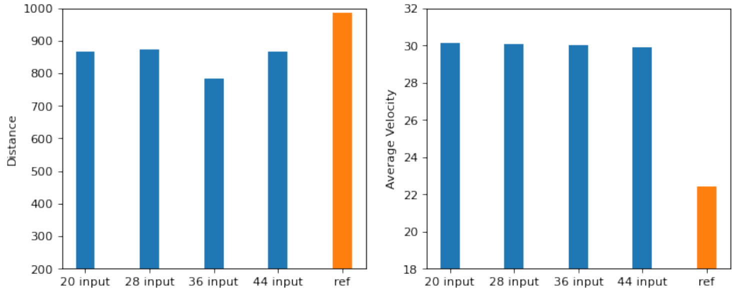

Figure 5: Average maximum driving distance and average velocity adding different stage prediction

5 Conclusion & Limitation

In this project, we developed a safety decision-making model of autonomous vehicle based on Deep Q Network (DQN) to achieve collision avoidance and high-speed driving in a highway environment. Moreover, we use the prediction information of the surrounding object's trajectories as input to train the model and show their potentials in RL-based safety decision model.

However, we do have several limitations due to the time and academic background. Future work can be done following the below direction.

- As we mentioned,the DQN is ineffective in solving issues involving high-dimensional action state spaces. Thus more advanced reinforcement learning algorithm or network structures should be implemented in this topic.
- The reward mechanism now only punish when the collision happen. An additional punishment should be added to penalize closed following distance.
- A decision-making system combining handcrafted rules and RL-based model may improve the performance of the model and accelerate the training process.

References

[1] J. Garcıa and F. Fernandez, "A Comprehensive Survey on Safe Reinforcement Learning," p. 44.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," IEEE Access, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.

[3] C.-J. Hoel, K. Wolff, and L. Laine, “Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Nov. 2018, pp. 2148–2155. doi: 10.1109/ITSC.2018.8569568.

[4] S. Nageshrao, E. Tseng, and D. Filev, “Autonomous Highway Driving using Deep Reinforcement Learning.” arXiv, Mar. 29, 2019. Accessed: Oct. 21, 2022. [Online]. Available: http://arxiv.org/abs/1904.00035

[5] X. Xiong, J. Wang, F. Zhang, and K. Li, “Combining Deep Reinforcement Learning and Safety Based Control for Autonomous Driving.” arXiv, Dec. 01, 2016. Accessed: Oct. 19, 2022. [Online]. Available: http://arxiv.org/abs/1612.00147

[6] A. Baheri, S. Nageshrao, H. E. Tseng, I. Kolmanovsky, A. Girard, and D. Filev, “Deep Reinforcement Learning with Enhanced Safety for Autonomous Highway Driving,” in 2020 IEEE Intelligent Vehicles Symposium (IV), Oct. 2020, pp. 1550–1555. doi: 10.1109/IV47402.2020.9304744.

[7] J. Chen, B. Yuan, and M. Tomizuka, “Model-free Deep Reinforcement Learning for Urban Autonomous Driving.” arXiv, Oct. 21, 2019. Accessed: Oct. 21, 2022. [Online]. Available: http://arxiv.org/abs/1904.09503

[8] B.-Q. Huang, G.-Y. Cao, and M. Guo, “Reinforcement Learning Neural Network to the Problem of Autonomous Mobile Robot Obstacle Avoidance,” in 2005 International Conference on Machine Learning and Cybernetics, Aug. 2005, vol. 1, pp. 85–89. doi: 10.1109/ICMLC.2005.1526924.

[9] A. Baheri, “Safe Reinforcement Learning with Mixture Density Network: A Case Study in Autonomous Highway Driving.” arXiv, Nov. 17, 2020. Accessed: Oct. 21, 2022. [Online]. Available: http://arxiv.org/abs/2007.01698

[10] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. CoRR, abs/1708.06374, 2017.

[11] Nguyen L T, Schmidt H A, Von Haeseler A, et al. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies[J]. Molecular biology and evolution, 2015, 32(1): 268-274.

[12] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

[13] W. D. Montgomery, R. Mudge, E. L. Groshen, S. Helper, J. P. MacDuffie, and C. Carson, “America’s workforce and the self-driving future: Realizing productivity gains and spurring economic growth,” Securing America’s Future Energy, Washington, DC, USA, Tech. Rep., 2018.

[14] Li, Xin, Xiaowen Ying, and Mooi Choo Chuah. "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving." arXiv preprint arXiv:1907.07792(2019).

[15] Xu, Jiaxuan, et al. "An automated learning-based procedure for large-scale vehicle dynamics modeling on baidu apollo platform." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.

[16] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. icml, vol. 30, no. 1, 2013, p. 3.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[18] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model mobil for car-following models,” Transportation Research Record, vol. 1999, no. 1, pp. 86–94, 2007.

[19] Xiong, Xi, et al. "Combining deep reinforcement learning and safety basedcontrol for autonomous driving." arXiv preprint arXiv:1612.00147 (2016).

[20] Baheri, Ali. "Safe reinforcement learning with mixture density network, with application to autonomous driving." Results in Control and Optimization 6 (2022): 100095.

[21] Li, Yuxi. "Deep reinforcement learning: An overview." arXiv preprint arXiv:1701.07274 (2017).

[22] M. Treiber, A. Hennecke, and D. Helbing, “Congested Traffic States in Empirical Observations and Microscopic Simulations,” Phys. Rev. E, vol. 62, pp. 1805–1824, 2000.

[23] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model mobil for car-following models,” Transportation Research Record, vol. 1999, pp. 86–94, 2007.