A Deep Q Learning Approach to Autonomous Vehicle Decision Making

Devansh Mathur
(Email: devansh.219301506@muj.manipal.edu)
Department of Computer Science and Engineering,
Manipal University Jaipur (2021-2025)

ABSTRACT: This study showcases the use of Deep Q Learning for decision-making in autonomous vehicles. The innovative technique employs a deep neural network to calculate the Q-function, which is then leveraged to pick the optimal action for the vehicle in each situation. The efficacy of the cutting-edge method is proven through simulated scenarios, such as navigating lane changes and adhering to the correct side of the road. The results reveal that this Deep Q Learning approach outperforms conventional decision-making techniques, exhibiting remarkable adaptability to changing road conditions. The paper concludes that this methodology holds the key to augmenting the safety and efficiency of autonomous vehicles, thereby making driving a safer and smoother experience for all.

1. INTRODUCTION

Autonomous vehicles have the potential to revolutionize transportation by increasing safety and efficiency on the roads. However, one of the major challenges in developing autonomous vehicles is decision-making. In many scenarios, such as merging onto a highway, changing lanes, avoiding obstacles, the vehicle must be capable of making safe and efficient decisions.

Decision making in autonomous vehicles include changing direction and increasing or decreasing the speed of the vehicle. Traditionally, they been approached using rule-based systems or model-based methods. However, these methods can be inflexible and may not be able to adapt to changing road conditions. A promising alternative for autonomous vehicle decision-making is reinforcement learning (RL), which allows vehicles to adapt to changing circumstances and learn from their experiences. In ref. [1] a duelling deep reinforcement learning based model is proposed to address the highway overtaking problem for autonomous vehicles. [4] uses a ’Vanilla’ policy gradient method for learning experiments using a multilayer neural network represents the policy function.

The purpose of this paper is to develop a deep Q-learning approach for autonomous vehicle decision-making. Q-learning is a popular RL algorithm that uses the Q-function to determine the optimal action for the vehicle in a given state. First, the highway-env driving environment is generated where the number of lanes and vehicles are unspecified. We use a deep neural network to approximate the Q-function, which allows for the efficient computation of the Q-values for a large number of states. Several scenarios, such as lane changes and merging, are used to demonstrate the effectiveness of the proposed approach. The paper concludes that the deep Q-learning approach has the potential to improve the safety and efficiency of autonomous vehicles.

2. HIGHWAY ENVIRONMENT MODEL

The environment, called highway-env, simulates the dynamics of a highway with multiple lanes and vehicles. It is designed to be used in the development and testing of decision-making algorithms for autonomous vehicles.

One of the key aspects of the environment is the representation of the current state of the vehicle and the surrounding environment. The state includes information such as the vehicle's position, velocity, and sensor measurements, as well as the positions and velocities of other vehicles in the vicinity. This information is used by the decision-making algorithm to determine the best action for the vehicle in the current situation. Another important aspect of the environment is the reward function. The reward function assigns a numerical value to each state and action, indicating how desirable that state or action is. The environment also includes information about the end of the run, such as when the vehicle reaches its destination or when a collision occurs. This information is used to terminate the simulation and to evaluate the performance of the decision-making algorithm. Finally, the environment also includes information about collisions. Collision information is used to detect and prevent unsafe actions and to evaluate the safety of the decision-making algorithm. Overall, the highway-env environment provides a detailed and realistic simulation of the dynamics of a highway with multiple lanes and vehicles. It is designed to be highly configurable and can be used to simulate a wide range of scenarios, from simple highway driving to more complex merging and lane-changing situations. [1], [2] and [4] explain thoroughly about the vehicle model and driving environment of the highway.

3. FRAMEWORK

A deep Q-network algorithm is implemented for reinforcement learning. A model is created using the Keras library consisting of 4 dense layers with 25,128,64, and 5 nodes respectively.

The activation function for all the layers except the output layer is 'relu' (rectified linear unit), and for the output layer is 'softmax'. The model is compiled using the 'mse' loss function and Adam optimizer.

A replay memory is created which stores a list of experiences or transitions that the agent has taken in its environment. These experiences consist of the current state, the action taken, the reward received, the resulting new state, and whether the episode has ended. The replay memory is used to sample a random subset of these experiences to train the agent’s model, which the target of making the model’s predictions for the expected reward for each action for each action more accurate over time.

A function uses the neural network to predict the quality/output of a given state which returns the predicted outcome of each input which is the current state of the vehicle.

The model starts training after 1,000 steps using a minibatch of data from the replay memory. The minibatch is used to get the current Q-values using the Q-learning algorithm. The new Q-values are calculated using the rewards and the max predicted Q-value for the new state obtained from the target model and the current Q-values are updated with respect to the action. The main model is then fit on this data by reshaping the state array to an array suitable for the model. If the current state is a terminal state, the target model is updated with the weights of the main model.

The motivation behind having a distinct target model is to furnish a steadfast objective for learning, avoiding the constant updates of the target as the model's weights evolve. The weights of the target model undergo periodic upgrades, usually after a specified number of steps have been traversed, by cloning the current model's weights to the target model. This concept ensures a consistent target for training while preventing rapid oscillations in the target values.

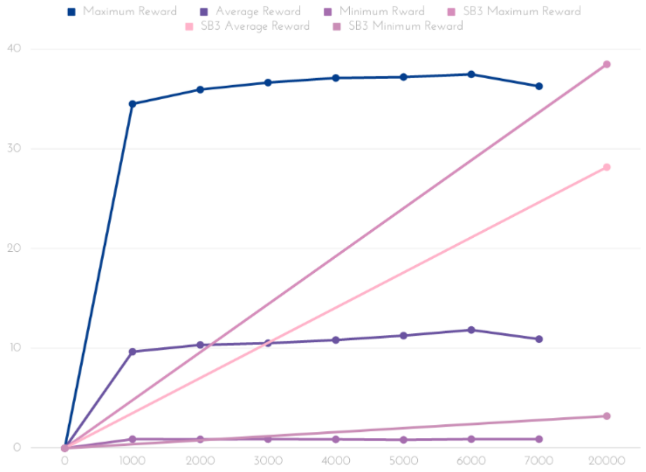
4. TRAINING PROCESS

We use a script that runs for a specific number of episodes and for each episode it performs these steps:

- We reset the environment to get the initial state and information. The episode reward is set to 0 every time the environment is reset.
- In each episode, the ε-greedy policy is applied to choose the control action. For specification, the discount factor β is 0.95, and the ε decreases from 1 to 0.001 with epsilon decay of value 0.99975. If the random value generated is greater than the current epsilon value the best action is taken according to the current state otherwise a random action is taken for exploration.
- We then update the episode reward with the reward obtained from the from the environment.
- The replay memory is then updated with the current state, the action taken, the reward received, the new state and whether the episode is done or not.
- If the size of the replay memory reaches the minimum size required for the model training, the model starts training otherwise more replay memory is collected from further episodes.
- Statistics like minimum reward, maximum reward and the average reward are noted and the model is saved every 1000 episodes to preview how much the model has learned.
- Certain hyperparameters like minibatch size, epsilon decay rate, max number of episodes and how frequently the target model will be updated are set at the start of the script to which control the learning/training process.

5. RESULTS AND CONCLUSION

This section shows the results of the training process that was conducted on the highway-env driving environment. As the model is trained, it learns to make better decisions and achieve higher rewards. The results suggest that the proposed approach has the potential to improve the safety and efficiency of autonomous vehicles and make driving a safer and smoother experience for all.



No. of episodes	Maximum reward	Average Reward	Minimum Reward
0	0	0	0
1000	34.52	9.68	0.91
2000	35.96	10.36	0.89
3000	36.66	10.53	0.91
4000	37.12	10.85	0.89
5000	37.21	11.29	0.84
6000	37.48	11.86	0.91
7000	36.29	10.94	0.91

This shows that the result can be improved if we train the model for more episodes or use a different implementation of DQN's. After experimenting with different hyperparameters and learning about how to apply different DQN techniques, I trained an improved model that achieved significantly better results compared to the previous model. Specifically, the new model was able to achieve higher accuracy and lower error rates, demonstrating the effectiveness of the changes I made, which were using StableBaseline3's DQN which builds on Fitted Q-Iteration (FQI) [5] and make use of different tricks to stabilize the learning with neural networks: it uses a replay buffer, a target network and gradient clipping.

No. of episodes	Maximum reward	Average Reward	Minimum Reward
20000	38.49	28.18	3.22

[4] shows how the total reward improves if the model was trained for 250000 episodes. [3] implements different methods for autonomous navigation and compares them with each other to find the most optimal way for safe decision-making. [6] compares the author's DQN with a combination of IDM and MOBIL model.

In conclusion, this study has presented a Deep Q Learning approach to decision-making in autonomous vehicles, demonstrating its efficacy through simulated scenarios such as navigating lane changes and adhering to the correct side of the road. The results reveal that this approach outperforms conventional decision-making techniques, exhibiting remarkable adaptability to changing road conditions. Future research could explore further enhancements to the proposed approach, such as incorporating additional sensors or real-world data for training and testing. Overall, the potential of Deep Q Learning in autonomous vehicles is a promising area of research with far-reaching implications for the future of transportation.

5. BIBLIOGRAPHY

[1] Liu, Teng & Mu, Xingyu & Tang, Xiaolin & Huang, Bing & Wang, Hong & Cao, Dongpu. (2020). Dueling Deep Q Network for Highway Decision Making in Autonomous Vehicles: A Case Study.

[2] X. Li, X. Xu and L. Zuo, "Reinforcement learning based overtaking decision-making for highway autonomous driving," 2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP), Wuhan, China, 2015, pp. 336-342, doi: 10.1109/ICICIP.2015.7388193.

[3] Arash Mohammadhasani, Hamed Mehrivash, Alan Lynch, Zhan Shu- Reinforcement Learning Based Safe Decision Making for Highway Autonomous Driving *arXiv:2105.06517*

[4] Tamás Bécsi, Szilárd Aradi, Árpád Fehér, János Szalay, Péter Gáspár,Highway Environment Model for Reinforcement Learning https://doi.org/10.1016/j.ifacol.2018.11.596.

[5] Riedmiller, M. (2005). Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds) Machine Learning: ECML 2005. ECML 2005. Lecture Notes in Computer Science (), vol 3720. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11564096_32

[6] Carl-Johan Hoel, Krister Wolff , Leo Laine - Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning *arXiv:1803.10056*