Deep Q-Learning For Adaptive Speed Control In Autonomous Vehicles

#exploreCSR

Abdul Rehman1, Shruti Sharma[2] , Ariba Khan[3] , Priya Kumari[4], Indukuri Mani Varma[5]

1) Sant Longowal Institute of Engineering and Technology, SLIET, Punjab ,2) Madhav Institute of Technology and Science ,Gwalior(Madhya Pradesh) 3) Aligarh Muslim University, Uttar Pradesh 4) Guru Gobind Singh Educational Society's Technical Campus, Jharkhand 5)Indian Institute of Technology Roorkee

INTRODUCTION

Autonomous vehicles are revolutionizing transportation by enhancing safety and efficiency, with precise speed control under varying driving conditions being a critical challenge. The dynamic traffic conditions, characterized by fluctuating vehicle densities, unpredictable maneuvers of neighboring vehicles, and varying speeds, pose significant challenges for maintaining safe and efficient driving speeds. To address these challenges, a Deep Q-Neural Network (DQN) model is employed to optimize speed management in autonomous vehicles by enabling them to learn from their experiences and improve decision-making over time. The model is trained in a simulated environment that closely mirrors real-world driving scenarios. The environment provides observations of the current state, enables actions to adjust vehicle speed, and compute rewards based on speed regulation performance using Deep Reinforcement Learning (DRL). The agent's objective, running in the vehicle, is to maintain a high speed while making room for the vehicles so that they can safely merge in the traffic. The simulation is done in the Highway env simulator to create high-dense vehicular scenarios as shown in Fig. 1 and run the Deep Reinforcement Learning model such that the task is achieved.

Fig 1

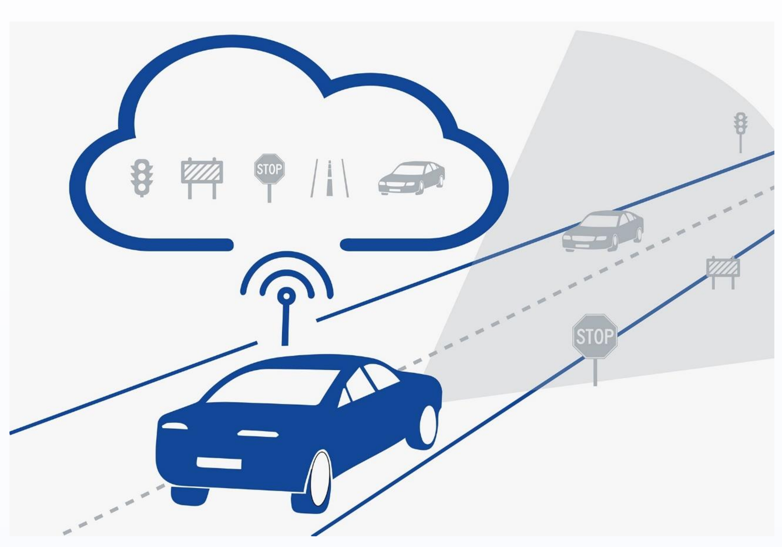


Fig 2

OBJECTIVES

1) Vary the number of controlled vehicles with multiple agents and analyze their speed control mechanism in the scenario.
2) Vary the density of traffic and other vehicles' speed based on the environment.
3) Model Hyperparameters Tuning in the DQN model.
4) Create scenarios where there is a high probability of collision between the vehicles.

METHODOLOGY

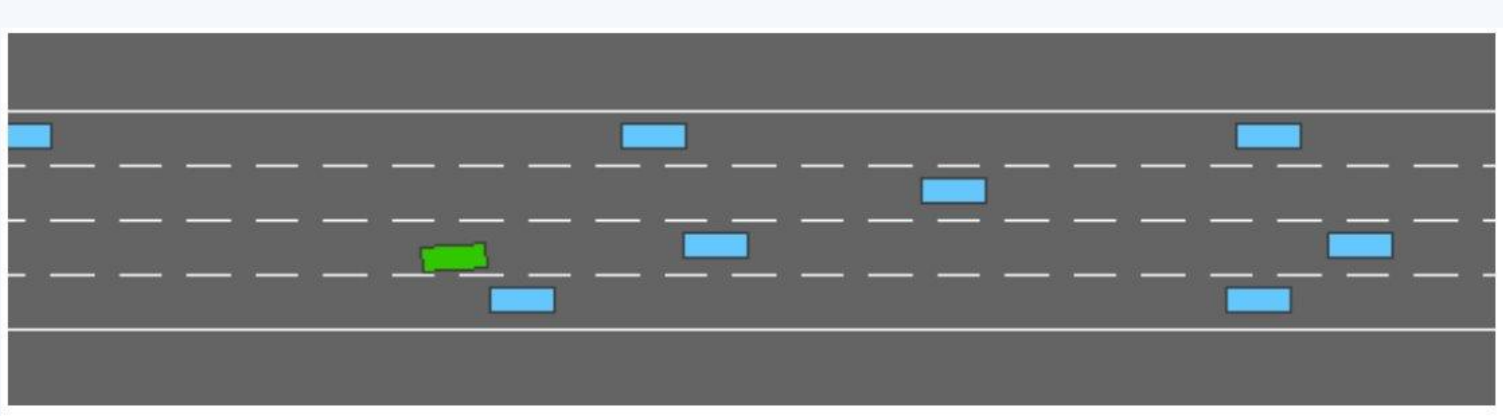


Fig 3

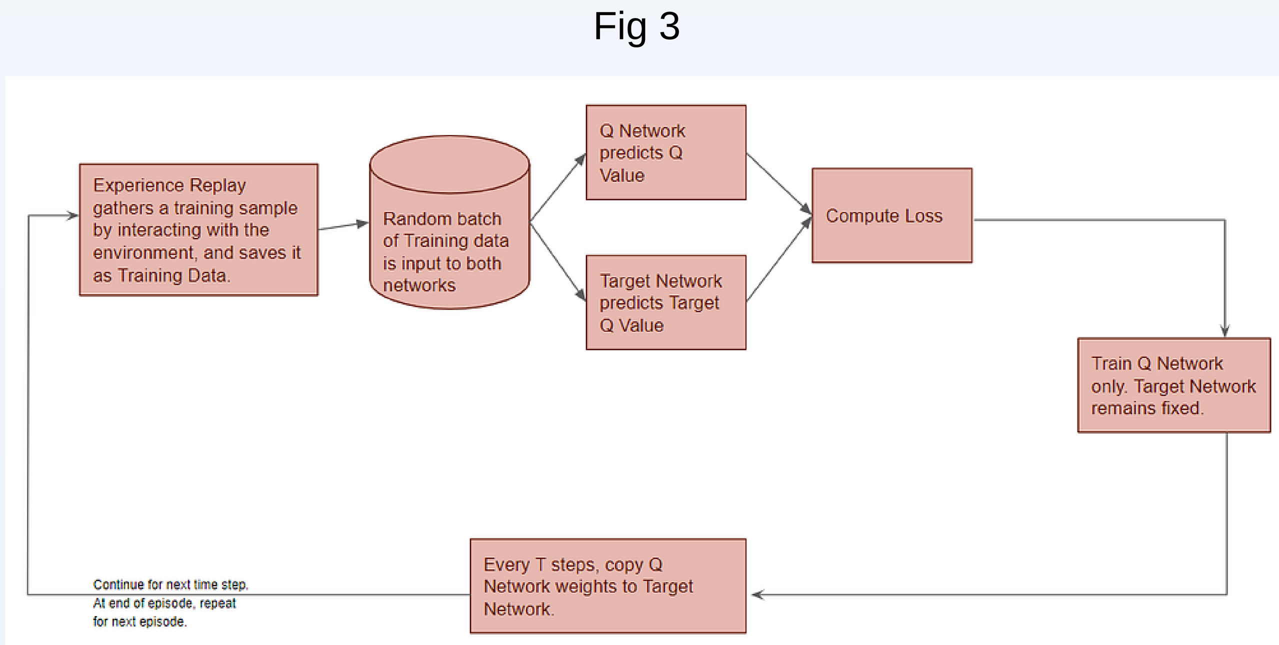


Fig 4

Fig. 3: The ego/controlled-vehicle (in green colour) is driving on a multilane highway populated with other vehicles. The objective of the agent, running in the ego vehicle, is to reach a high speed while avoiding collisions with neighboring vehicles. Driving on the right side of the road is also rewarded.

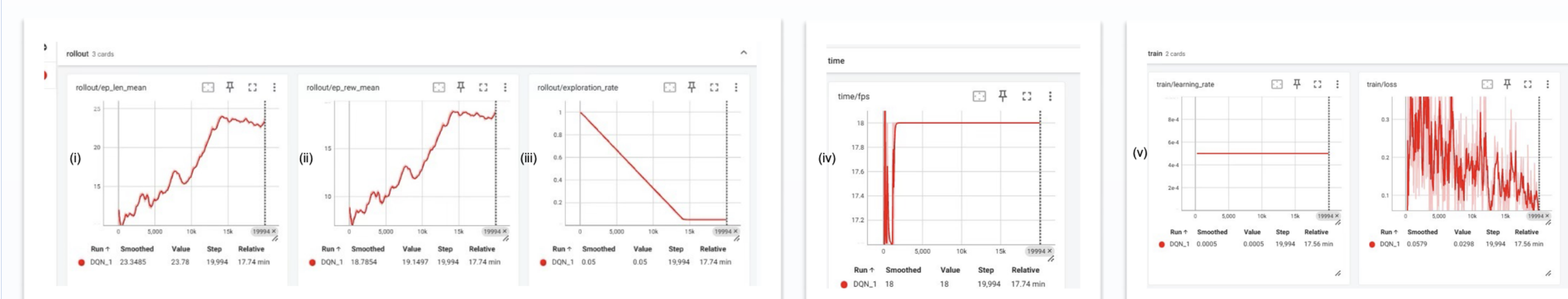
Fig4:In deep reinforcement learning, particularly in the context of Deep Q-Learning, the training process is inherently iterative. After the neural network computes the loss and updates its weights based on the predicted and actual Q-values, the cycle begins anew. This cyclical process incorporates a technique known as experience replay, which significantly contributes to the stability and efficiency of the training procedure.

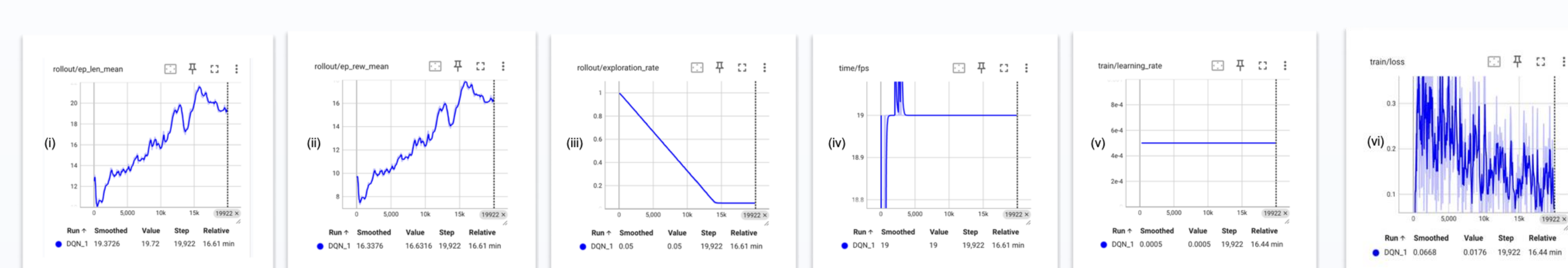
Experience replay involves storing the agent's experiences, which consist of state, action, reward, and next-state tuples, in a replay buffer. During training, random samples from this buffer are used to update the neural network. This method has several advantages: Decorrelation of Data, Efficient Use of Data, Breaks Temporal Correlations.
By leveraging experience replay, the agent can effectively learn optimal speed control policies through continuous interactions with the environment, ensuring that it not only reacts appropriately to immediate situations but also makes decisions that align with long-term objectives. This results in smoother, safer, and more efficient driving behaviors, which are essential for the practical deployment of autonomous vehicles.

RESULTS

Based on Model Hyperparameter Tuning:

- **Learning Rate =5e-4** , We lowered it to stabilize training (learning_rate=1e-4). It did prevent large updates but also led to unstable learning.
- **Batch Size =32**, We Increased the batch size to 64 and then to 128, It provided stability in training by providing better gradient estimates, though it will use more memory.
- **Buffer Size=15000**, Increasing it to 50000 or 100000 can help the model learn from a larger pool of experiences.
- **Exploration Fraction 0.7**, We Lowered it to around 0.4, which allowed the model to exploit learned policies sooner while still exploring adequately.
- **Gamma=0.8**, We increased it to 0.99 to focus on long-term rewards that lead to more strategic behavior.
- **Train_freq=1, gradient_steps=1,** We set Train_freq=4 and gradient_steps=4 to allow for more robust training cycles.





(i) rollout/ep_len_mean: This graph shows the average episode length over time during the training process.
(ii) rollout/ep_rew_mean: This graph shows the average episode reward over time during the training process.
(iii) rollout/exploration_rate: This graph shows the exploration rate over time during the training process.
(iv) time/fps:This graph shows the frames per second (FPS) during the training process over time.
(v) train/learning_rate:This graph shows the learning rate used during the training process over time.
(vi) rain/loss:This graph shows the loss over time during the training process.

CONCLUSION

This study demonstrates that Deep Q-Learning significantly enhances the speed control of autonomous vehicles, improving their safety and efficiency in various driving conditions. The DQN model, trained in a simulated environment, effectively adapts to changes in traffic density, vehicle speeds, and the number of controlled vehicles. Hyperparameter tuning further optimizes the model's performance. The results show substantial improvements in episode lengths, rewards, and training stability. These findings underscore the potential of Deep Q-Learning in advancing autonomous vehicle technology for real-world applications.

REFERENCES

1) Highway-env, Leurent, Edouard, An Environment for Autonomous Driving Decision-Making, 2018, GitHub, https://github.com/eleurent/highway-env.

2) Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. No. 1. 2016.

ACKNOWLEDGEMENT

The authors would like to thank the Google for Research opportunity. Also, the author is very grateful to the Indian Institute of Technology Roorkee for providing the necessary infrastructure to carry out this research work. The author would also like to thank Devki Nandan Jha for guiding and supervising during the research internship.