Sultan Moulay Slimane University

NATIONAL SCHOOL OF APPLIED SCIENCES KHOURIBGA

Report

Module: Research Methodology

1st year Master's Big Data and Decision Support :

Under the Theme :

Reinforcement Learning in the Transportation Sector : Autonomous Vehicle

Carried out at ENSA, Khouribga

From 03/03/2024 to 09/06/2024

Framed by :

Bella Abdelouahab

Supervised By :

Pr. My Saddik KADIRI

Academic Year : 2023-2024

Acknowledgment

I extend my deepest gratitude to my supervisor, Pr. My Saddik KADIRI, for his guidance and support throughout the development of this project. I also thank my family, friends, and peers for their encouragement and feedback.

Abstract

This report explores the application of Deep Reinforcement Learning (DRL), specifically through the Double Deep Q-Network (DDQN), in advancing autonomous vehicle technologies. DDQN is utilized to optimize the decision-making processes in self-driving cars, facilitating rapid and efficient response to dynamic driving environments. The study investigates the theoretical underpinnings of DRL and its practical implications in real-world scenarios, analyzing how the integration of DDQN enhances vehicular navigation systems. The research not only tests the efficacy of DDQN in optimizing route choices and driving strategies but also examines the potential for reducing accidents and improving traffic efficiency. By analyzing performance metrics under various simulation conditions, this study provides insights into the scalability and adaptability of reinforcement learning models in autonomous driving, offering a pathway to more sophisticated and safer transportation solutions.

Keywords: Deep Reinforcement Learning, DDQN, Autonomous Vehicles, Decision Making, Traffic Efficiency, Simulation, Artificial Intelligence, Real-Time Data Processing.

Résumé

Ce rapport explore l'application de l'apprentissage par renforcement profond (Deep Reinforcement Learning, DRL), spécifiquement à travers le réseau Double Deep Q-Network (DDQN), pour faire avancer les technologies des véhicules autonomes. Le DDQN est utilisé pour optimiser les processus de prise de décision dans les voitures autonomes, facilitant une réponse rapide et efficace aux environnements de conduite dynamiques. L'étude examine les fondements théoriques du DRL et ses implications pratiques dans des scénarios du monde réel, analysant comment l'intégration du DDQN améliore les systèmes de navigation véhiculaire. La recherche teste non seulement l'efficacité du DDQN dans l'optimisation des choix d'itinéraire et des stratégies de conduite, mais examine également le potentiel de réduction des accidents et d'amélioration de l'efficacité du trafic. En analysant les métriques de performance sous diverses conditions de simulation, cette étude fournit des perspectives sur la scalabilité et l'adaptabilité des modèles d'apprentissage par renforcement dans la conduite autonome, offrant un chemin vers des solutions de transport plus sophistiquées et plus sûres.

Mots-clés : Apprentissage par renforcement profond, DDQN, Véhicules autonomes, Prise de décision, Efficacité du trafic, Simulation, Intelligence artificielle, Navigation véhiculaire.

Contents

List of Figures

General Introduction

The integration of artificial intelligence (AI) into automotive technology represents a pivotal shift in how vehicles operate and interact with their environment. The advent of deep reinforcement learning (DRL), particularly through algorithms such as the Double Deep Q-Network (DDQN), has opened new frontiers in the development of autonomous driving systems.[4] This report delves into the mechanics of DDQN and its applications in self-driving cars, focusing on its role in enhancing navigational algorithms and decision-making processes. The study elaborates on the advantages of DDQN over traditional control systems, including its ability to learn from the environment through a reward-based system without human intervention. The exploration covers various aspects of DRL, such as the algorithmic challenges, computational requirements, and the ethical implications of deploying autonomous vehicles on public roads. By providing a comprehensive analysis of DDQN's impact on safety, efficiency, and reliability, the report aims to contribute to the ongoing discourse on the future of transportation, highlighting the transformative potential of AI in reshaping how vehicles perceive, decide, and act in an ever-changing world.

Chapter1

General Context

1.1 Introduction

The transportation sector is at the forefront of technological innovation, driven by the need for more efficient, safe, and sustainable mobility solutions. The integration of artificial intelligence (AI) and Big Data is playing a pivotal role in this transformation. These technologies are not only enhancing the operational efficiency of transportation systems but also enabling the development of autonomous vehicles. Among the various AI techniques, Deep Reinforcement Learning (DRL) has emerged as a key player, offering robust solutions for complex decision-making processes. This chapter provides a general context for understanding the role and impact of the Double Deep Q-Network (DDQN) in the transportation sector.

1.2 Big Data in Transportation

Big Data refers to the vast volume of data generated from various sources, including sensors, GPS devices, social media, and transactional records. In the transportation sector, Big Data is used to analyze traffic patterns, predict travel times, optimize routes, and improve overall efficiency. The ability to process and analyze large datasets in real-time has significant implications for traffic management, logistics, and public transportation systems.

Big Data analytics in transportation involves collecting data from diverse sources, processing it to extract meaningful insights, and using these insights to inform decision-making. For example, traffic management centers use real-time data from sensors and cameras to monitor traffic flow and make adjustments to traffic signals to reduce congestion. Similarly, public transportation systems use data from ticketing systems and GPS devices to optimize schedules and routes.

1.3 Artificial Intelligence in Transportation

Artificial Intelligence (AI) encompasses a range of technologies that enable machines to mimic human intelligence. In the transportation sector, AI is used for various applications, including predictive maintenance, autonomous driving, and traffic management. Machine learning, a subset of AI, involves training algorithms to recognize patterns in data and make predictions or decisions based on these patterns.

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. This approach is particularly well-suited for applications where the optimal solution is not known in advance and must be discovered through trial and error.

1.4 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) combines the strengths of deep learning and reinforcement learning. Reinforcement learning is like giving your car a personal coach. It learns from experience and improves its driving efficiency, just like an athlete refining their skills over time.[1] Deep learning involves training neural networks with multiple layers to recognize complex patterns in data, while reinforcement learning focuses on learning optimal actions through interaction with an environment. DRL algorithms, such as the Deep Q-Network (DQN), use deep neural networks to approximate the Q-value function, which represents the expected reward for taking a given action in a given state.

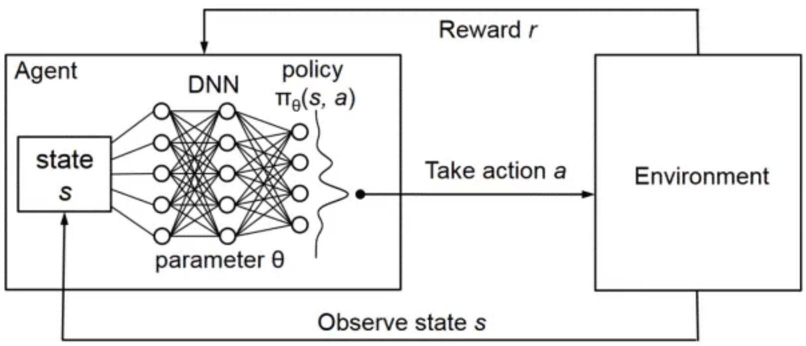

Figure 1.1: DQN Architecture

The DQN algorithm has been successfully applied to various tasks, including playing Atari games at a superhuman level. However, DQN suffers from overestimation bias, where the algorithm tends to overestimate the value of certain actions. To address this issue, the Double Deep Q-Network (DDQN) was introduced.

1.5 Double Deep Q-Network (DDQN)

The Double Deep Q-Network (DDQN) is an extension of the DQN algorithm that reduces overestimation bias by decoupling the selection of actions from the evaluation of their values. In DDQN, two networks are used: a primary network for selecting actions and a target network for evaluating the value of the selected actions. The target network is updated less frequently than the primary network, providing a more stable target for learning.

DDQN has been shown to improve the stability and performance of DQN in various tasks, making it a promising approach for applications in autonomous driving. By enabling more accurate decision-making in dynamic environments, DDQN can enhance the safety and efficiency of autonomous vehicles.

1.6 Applications of DDQN in Transportation

The application of DDQN in the transportation sector is primarily focused on autonomous driving. Autonomous vehicles must navigate complex and dynamic environments, making real-time decisions based on sensor data and predefined rules. DDQN can optimize these decision-making processes, allowing autonomous vehicles to respond more effectively to changing traffic conditions and unexpected obstacles.

For instance, DDQN can be used to optimize route planning for autonomous vehicles, ensuring that they take the most efficient path while avoiding congestion and minimizing fuel consumption. Additionally, DDQN can be applied to improve the coordination of multiple autonomous vehicles, enhancing overall traffic flow and reducing the likelihood of accidents.

Beyond autonomous driving, DDQN can also be applied to other areas of transportation, such as traffic signal control, public transportation scheduling, and logistics optimization. By leveraging the capabilities of DDQN, transportation systems can become more responsive, efficient, and reliable.

1.7 Challenges and Future Directions

While DDQN offers significant potential for improving transportation systems, several challenges remain. One major challenge is the need for large amounts of high-quality data to train DRL algorithms. Collecting and processing this data can be resource-intensive, and ensuring its accuracy and relevance is crucial for the success of DDQN applications.

Another challenge is the computational complexity of training and deploying DDQN models. Advanced hardware and software infrastructure are required to support the real-time processing and decision-making capabilities of DDQN. Additionally, the deployment of DDQN in real-world transportation systems must address issues related to safety, reliability, and ethical considerations.

Despite these challenges, the future of DDQN in transportation looks promising. Ongoing research and development efforts are focused on improving the efficiency and scalability of DDQN algorithms, as well as exploring new applications and use cases. As technology continues to advance, DDQN is expected to play an increasingly important role in shaping the future of transportation.

1.8 Conclusion

The integration of Big Data and AI, particularly through the use of DDQN, is driving significant advancements in the transportation sector. By enabling more accurate and efficient decision-making, DDQN has the potential to transform autonomous driving and other transportation applications. This chapter has provided a general context for understanding the role and impact of DDQN in transportation, setting the stage for a deeper exploration of its technical aspects and practical implementations in the following chapters.

Chapter2

Methodology

Because it makes decisions based only on the present system state, the double deep reinforcement-learning model in this study is constructed by combining a neural network with a standard reinforcement-learning model to build a real-time controller. Double deep Q-network, or DDQN, is the name of this novel algorithm model that was initially suggested and used for video game play (Mnih et al., 2015). Remarkably, the dynamic model of the controlled vehicle and the specific control variables do not need to be taken into account while developing a reinforcement learning based control strategy. All that is needed to specify the control agent's inputs and outputs are the environmental conditions and actions, respectively. This is the major difference comparing the conventional control strategies like model predictive control (MPC) that is based on a full dynamical model of the controlled vehicle. The selection of definition of environmental states, actions and reward in this work is based on domain expertise. The actions, states, and reward for this specific problem are defined in the following sections[3].

2.1 Model-free Control

2.1.1 Double Deep Q-Network (DDQN) Model

The DDQN model integrates a neural network with a standard reinforcement learning model to build a real-time controller. Unlike traditional control strategies such as model predictive control (MPC), which rely on a full dynamic model of the controlled vehicle, DDQN learns optimal policies through interactions with the environment. This adaptability makes it suitable for complex and dynamic scenarios commonly encountered in autonomous driving. The advantages of DDQN include reduced overestimation bias through the use of two separate networks for selecting and evaluating actions, improved stability due to the decoupling of action selection and evaluation, and scalability as its model-free nature allows it to be applied across various scenarios without needing a detailed vehicle model.

2.1.2 States, Actions, and Immediate Reward

In the DDQN model, the state space (S) encompasses all necessary information for vehicle decision-making, including the current position (p) and speed (v) of the vehicle, sensor data (s_{sensor}) from LiDAR, cameras, and radar, traffic information regarding the positions (p_{traffic}) and velocities (v_{traffic}) of nearby vehicles, road conditions such as lane markings, traffic signals, and road curvature (c), and temporal information like time of day (t) and weather conditions (w). The action space (A) represents the possible maneuvers the vehicle can perform, such as adjusting the steering angle (a_{steering}), controlling the speed through acceleration ($a_{\text{acceleration}}$) and braking (a_{braking}), moving to adjacent lanes, and modifying speed to maintain safe distances or adhere to speed limits. The reward function (R) provides feedback to guide the learning process, encouraging safe driving by maintaining safe distances and adhering to traffic rules, efficiency by minimizing travel time and fuel consumption, smoothness by penalizing abrupt or unsafe maneuvers, and goal achievement by reaching the destination or following planned routes effectively.

2.2 DDQN Algorithm

The DDQN algorithm consists of several key steps. Initialization involves setting up the online network Q with random weights and the target network Q' with the same initial weights as the online network. The replay memory is also initialized to store experiences. During the experience replay phase, experiences (s_t, a_t, r_t, s_{t+1}) are stored in the replay memory, and a minibatch of experiences is sampled from this memory for training. For action selection, the online network Q is used to choose an action a_t based on the current state s_t. Action evaluation is then performed using the target network Q' to evaluate the value of the selected action. The Q-value ($Q(s_t, a_t)$) is calculated using the Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma \max_a Q'(s_{t+1}, a)$$

where r_t is the immediate reward, γ is the discount factor, and $\max_a Q'(s_{t+1}, a)$ represents the maximum Q-value of the next state s_{t+1} estimated by the target network Q'. The main difference between DQN and DDQN lies in the action evaluation step, where DDQN uses the target network to provide more stable targets for learning, reducing the overestimation bias commonly encountered in DQN. By following these steps, the DDQN algorithm improves the learning stability and decision-making accuracy of the reinforcement learning model, making it a powerful tool for real-time control in autonomous vehicles.

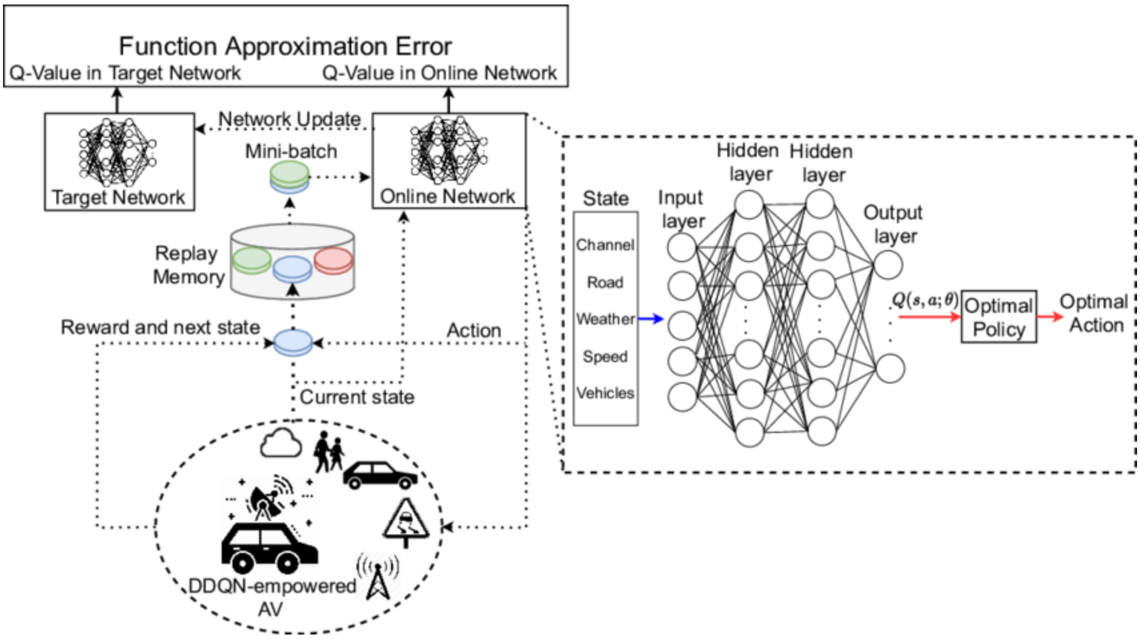

Figure 2.1: DQDN Architecture

Chapter 3

Case Study

3.1 Introduction

In this chapter, we delve into the realm of autonomous driving, focusing on the implementation of the Double Deep Q-Learning Network (DDQN) algorithm. Our aim is to explore how DDQN can enhance self-driving capabilities within simulated environments. Motivated by the transformative potential of autonomous technology, our project seeks to overcome dataset limitations inherent in traditional approaches, fostering agile and robust autonomous systems. Through a comprehensive methodology encompassing model development, simulation environment creation, and iterative training and evaluation, we endeavor to push the boundaries of technology and redefine the future of transportation.

3.2 Simulation Environment Development

In the pursuit of advancing autonomous vehicle technologies, the creation of robust and adaptable simulation environments stands as a cornerstone. These environments serve as the digital playgrounds where artificial intelligence algorithms learn to navigate the complexities of real-world scenarios. In this segment, we delve into the meticulous process of crafting a simulation environment tailored to facilitate the training of our model.

The selection of an appropriate framework emerged as a pivotal decision in our journey towards creating an effective simulation environment. After careful consideration, we opted for Pyglet, a versatile Python framework renowned for its prowess in game development. While competing frameworks boasted compelling features, Pyglet's unique ability to provide precise control over frame rates resonated with our project's computational demands.

Upon laying the groundwork with Pyglet, we embarked on the iterative process of environment refinement. From the initial blueprint to the final rendition, each iteration aimed to strike a delicate balance between functionality and user experience. The environment, rendered in a 2D canvas spanning 1000 by 500 pixels, featured a meticulously designed landscape, complete with a racing truck and maneuverable car.

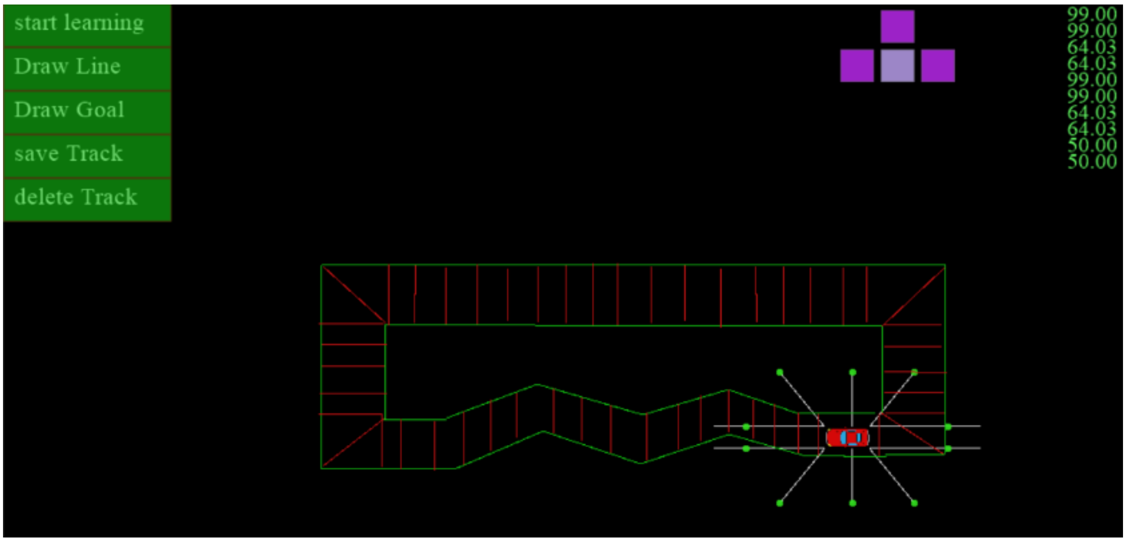

Figure 3.1: game simulation using pyglet

One of the defining features of our environment was the integration of a comprehensive control panel. This intuitive interface not only facilitated seamless interaction but also empowered users to exert granular control over various simulation parameters. From initiating learning sessions to saving and loading track configurations, the control panel served as the nerve center of our simulation environment.

Furthermore, the environment's evolution was punctuated by the introduction of novel functionalities aimed at enhancing its utility and versatility. Among these enhancements was the implementation of a sophisticated model-controlled car, responsive to nuanced keyboard inputs. Additionally, the incorporation of sensors and distance calculations facilitated precise obstacle detection and navigation.

In summary, the development of our simulation environment represents a testament to our unwavering commitment to excellence in autonomous vehicle research. Through meticulous attention to detail and a relentless pursuit of innovation, we have succeeded in crafting a platform poised to propel the frontiers of artificial intelligence in transportation.

3.3 DDQN model TensorFlow Architecture

In any model in machine learning and deep learning have an input we can say that Model Input refers to data that is sent to a Model either electronically or manually, such as data from a servicer, data from financial services information providers, cash adjustments (such as reimbursable expenses), and information from intermediate calculation programs. For this input it can be an array, picture, ... or like in our case it is an array containing numerical values representing the distances. So our model takes three distances, which are the distance of the left side, right side and front side. The same thing about output. It move a car left, right or speed break, so to build our model neural network we need to also define the hidden layers, these are neurons that correlate between a given input and the correct output, and each one layer has an activation function,we choose the number 3 to be the number of hidden layers and each layer with 32 neurons the following schema shows the structure of the neural network:

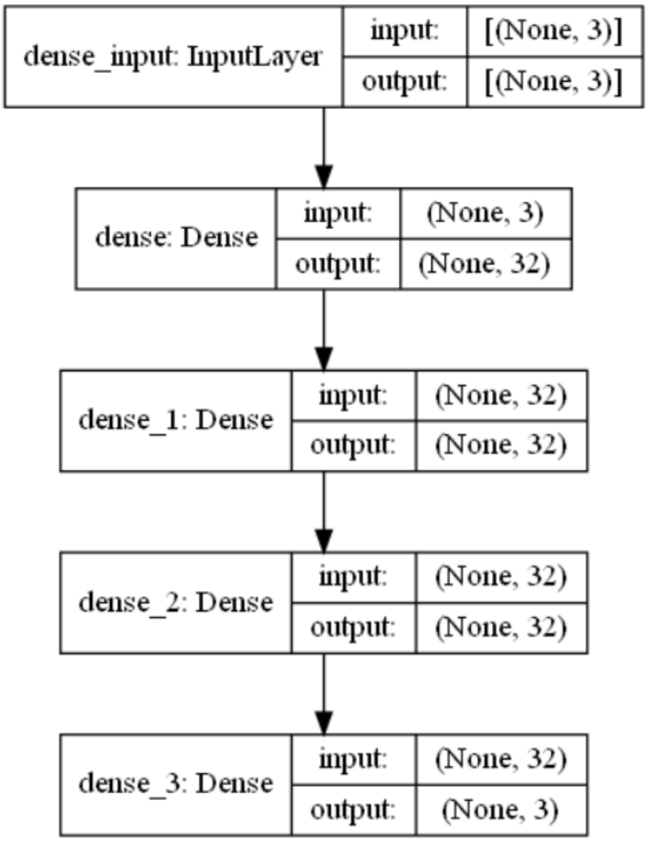

Figure 3.2: DDQN TensorFlow Architecture

3.3.1 DDQN Training and Findings

3.3.1.1 Model Configuration

In this section, we discuss the specific hyperparameters utilized to control the learning process within our environment. Given the complexity of the environment, careful selection of hyperparameters is crucial for achieving optimal performance.

- **Epsilon**: Epsilon, a value ranging between 0 and 1, dictates the frequency of random actions taken by the agent during training. To encourage exploration, we set the epsilon discount to 0.998. Additionally, to ensure effective training, we maintain a minimum epsilon value of 0.1.

- **Gamma**: Gamma determines the discount factor, influencing the agent's consideration of future rewards. By reducing the contribution of rewards from future states, the agent can focus on immediate rewards. We set the gamma value to 0.99 to strike a balance between short-term and long-term rewards.

3.3.1.2 Model Learning Graph

To visualize the learning progress of our DDQN model, we plot the training graph using default hyperparameters.

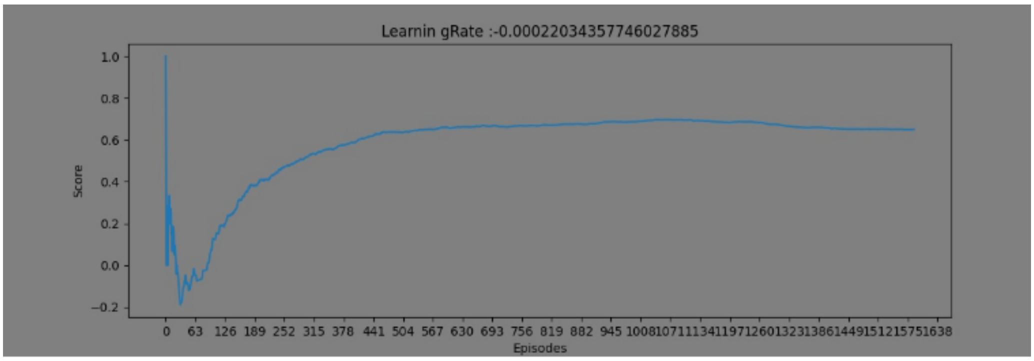

Figure 3.3: Training Graph with Default Hyperparameters

The above figure illustrates the training graph, showcasing the progression of rewards per episode over the course of training. The hyperparameters used for this training run are as follows:

- Learning Rate: 0.01
- Epsilon Discount: 0.9
- Batch Size: 128
- Inputs: 11
- Output (Number of Actions): 5

Next, we experiment with custom hyperparameters to observe their impact on model performance.

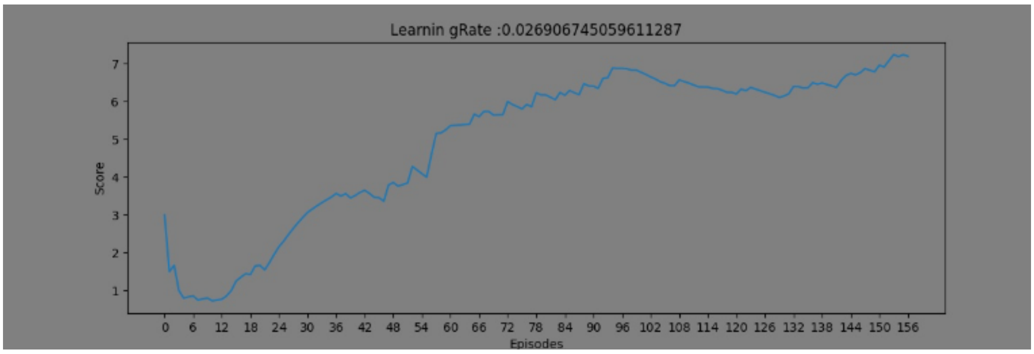

Figure 3.4: Training Graph with Custom Hyperparameters

In this graph, we adjust hyperparameters such as learning rate, epsilon discount, batch size, and gamma to assess their effects on the model's learning trajectory. The hyperparameters used for this training run are as follows:

- Learning Rate: 0.001
- Epsilon Discount: 0.998
- Batch Size: 64
- Inputs: 11
- Output (Number of Actions): 5
- Gamma: 0.99

Additionally, we increase the number of actions taken per second from 10 to 60 to allow the agent more time to evaluate the state.

As demonstrated, when compared to a model that uses default hyperparameters, the model that uses custom ones gradually increases the rewards per episode and results in a greater reward peak.

3.4 Discussion

The integration of Double Deep Q-Networks (DDQN) in autonomous vehicles holds significant promise for advancing the capabilities of self-driving technology. By employing reinforcement learning techniques within complex environments, DDQN facilitates autonomous decision-making, thereby enhancing the intelligence and adaptability of autonomous vehicles.

Autonomous vehicles stand to benefit immensely from the implementation of DDQN, particularly in scenarios requiring real-time decision-making and navigation through dynamic environments. Through continuous learning and interaction with the surroundings, DDQN-equipped vehicles can optimize their behavior to navigate safely and efficiently, thus reducing the likelihood of accidents and improving overall traffic flow.

Furthermore, the synergy between DDQN and big data analytics amplifies the potential of autonomous vehicles by enabling them to harness vast amounts of data for informed decision-making and predictive analysis. This integration not only enhances the safety and efficiency of autonomous driving but also opens avenues for novel applications such as predictive maintenance and route optimization.

In conclusion, the application of DDQN in autonomous vehicles represents a significant step towards realizing the vision of fully autonomous transportation systems. By leveraging reinforcement learning techniques and big data analytics, DDQN empowers vehicles to navigate complex environments with enhanced intelligence and autonomy, thereby reshaping the future of mobility.

Conclusion

In the realm of autonomous vehicles, Double Deep Q-Networks (DDQN) and big data analytics stand as formidable pillars of innovation. Our case study illuminates the power of DDQN in enhancing the intelligence and adaptability of self-driving models within simulated environments. By harnessing reinforcement learning principles and the insights gleaned from big data, DDQN-equipped vehicles promise safer, smarter, and more efficient transportation solutions.

Looking ahead, DDQN represents more than just automation; it signifies a transformative leap towards intelligent and sustainable mobility. As we embrace the possibilities of DDQN and big data analytics, we embark on a journey towards a future where transportation is not just autonomous but a catalyst for innovation and human progress.

In essence, our study underscores the transformative potential of DDQN in shaping the future of autonomous vehicles and invites us to envision a world where transportation is synonymous with safety, efficiency, and intelligence.

Bibliography

[1] Chen, Chen, Jiange Jiang, Ning Lv, and Siyu Li. "An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge." *IEEE Access* 8 (2020): 99059-99069. doi:10.1109/ACCESS.2020.2998015.

[2] Denis Ushakova, Egor Dudukalov, Ekaterina Mironenko, and Khodor Shatila. *Big data analytics in smart cities' transportation infrastructure modernization*. In *X International Scientific Siberian Transport Forum*, 2024. Address: Suan Sunandha Rajabhat University, 1 U-Thong Nok rd., Dusit, Bangkok, Thailand, 10300; Hardrain Technologies LLC, 30 North Gould Street, Suite 4000, Sheridan, Wyoming, United States of America, 8.

[3] Xuewei Qia, Yadan Luo, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew Barth. "Deep reinforcement learning enabled self-learning control for energy efficient driving." Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

[4] Pan, X., Yang, H., and Liu, D. (2020). A survey on deep reinforcement learning. *Neurocomputing*, 399, 49-69.

[5] Silver, D., Hubert, T., Schrittwieser, J., et al. (2018). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *Science*, 362(6419), 1140-1144.

[6] Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

[7] Hessel, M., Modayil, J., Van Hasselt, H., et al. (2018). Rainbow: Combining Improvements in Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

[8] Castro, R., Oliveira, A., and Ferreira, J. (2021). Autonomous vehicles and deep reinforcement learning: A survey. *Neurocomputing*, 443, 275-291.

[9] Schrittwieser, J., Antonoglou, I., Hubert, T., et al. (2019). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604-609.