Technical Report

Intelligent Urban Navigation Agent

Daniel Alejandro Presiga – 20212020116
Universidad Distrital Francisco José de Caldas
Professor: Carlos Andres Sierra

July 2025

Contents

Abstract

This technical report presents the design, modeling, implementation, and evaluation of an intelligent autonomous navigation agent based on Deep Q-Learning (DQN). The system was built on the foundations of reinforcement learning, using Gymnasium and Stable-Baselines3 to simulate urban traffic dynamics. The agent learns to navigate by maximizing travel efficiency while minimizing rule violations, showing promising results across multiple test scenarios.

1 Introduction

Urban mobility remains a central challenge in modern smart cities. Designing autonomous agents that can safely and efficiently navigate chaotic traffic scenarios requires advanced techniques such as reinforcement learning. This project proposes the implementation of a Deep Q-Network agent that adapts to traffic rules and unpredictable conditions, improving its decisions over time.

2 Literature Review

Previous works have demonstrated the effectiveness of Q-learning and DQN in control and navigation tasks. Notable references include the Farama Foundation's Minigrid environments, OpenAI Gym tutorials, and implementation guides on Stable-Baselines3. Key concepts include reward maximization, policy learning, and convergence under uncertainty.

3 Background

We modeled the agent's environment based on sensors (proximity, camera, tachometer, clock) and actuators (steering, wheels). The reward system penalizes infractions and rewards legal, efficient driving. The Bellman equation was adapted to form the basis of the DQN loss function. The architecture includes a policy network and a target network.

4 Objectives

- To implement an intelligent autonomous agent for urban navigation.
- Implement Deep Q-Learning in a simulated environment.
- Evaluate performance in a simulated urban scenario.

5 Scope

This project focuses on single-agent navigation in a simulated 2D environment using Gymnasium. Physical robotics, real-world sensors, or external APIs are not included. The agent operates in discrete time steps with defined observation and action spaces.

6 Assumptions

- The simulation accurately represents key urban driving conditions.
- Sensors provide relevant information with limited noise.
- The reward structure aligns with desirable driving behavior.

7 Limitations

- The agent may overfit to specific routes.
- Sensor degradation was artificially modeled.
- Multi-agent interaction was not explored in this iteration.

8 Methodology

The environment was constructed in Gymnasium with a discrete action space (accelerate, maintain, decelerate) and a 4D state vector. The reward function rewarded legal and efficient driving and penalized collisions and delays. The agent was trained with Stable-Baselines3 using DQN with epsilon-greedy exploration and replay buffers.

During training, the agent gradually learns to move and orient itself better in the environment. With each attempt, it "analyzes" its surroundings — considering traffic lights, nearby obstacles, and its own speed — to make smarter decisions in the next steps.

9 Results

Several tests were conducted by varying the number of training episodes (i.e., repetitions through the virtual map). As expected, the results confirmed that a higher number of repetitions led to better learning outcomes. With very few repetitions, the agent often entered loops or got stuck in certain areas. However, when increasing the number to values such as 30,000 or more, the agent clearly demonstrated learning and began achieving more consistent navigation toward the goal.

10 Discussion

The number of training repetitions had a direct and observable impact on the agent's performance. Low episode counts often led to poor generalization and repetitive behaviors. In contrast, long training sessions allowed the agent to explore more states and develop stronger policies. These repetitions help the agent refine its internal policy, gradually improving its movement and orientation strategies in response to the dynamic environment around it.

11 Conclusion

Although the project did not reach the originally envisioned level—mainly due to a basic interface and pending features—it provides a solid foundation for future development. The agent fulfills its core objective: learning from the environment and progressively improving its pathfinding. This iterative improvement shows that the project can evolve into a more complete solution with better visualization and interaction features.

References

- Farama Foundation. Minigrid. `https://github.com/Farama-Foundation/minigrid`
- DigitalOcean. Getting started with OpenAI Gym. `https://www.digitalocean.com/community/tutorials/getting-started-with-openai-gym`
- FSNDZOMGA. Reinforcement Learning using Stable Baselines 3. `https://medium.com/...`
- Amin, S. Deep Q-Learning (DQN). `https://medium.com/...`