Intelligent Urban Navigation Agent

Daniel Alejandro Presiga - 20212020116
Systems Science Foundations
Universidad Distrital Francisco José de Caldas
Professor: Carlos Andres Sierra

***Abstract*—This paper presents the design and implementation of an intelligent autonomous agent for urban navigation using Deep Q-Learning (DQN). The agent interacts with a simulated environment built with Gymnasium and trained using the Stable-Baselines3 library. It learns to make decisions that optimize travel time while respecting traffic rules. Key metrics such as reward per episode, average completion time, and navigation errors demonstrate the agent's ability to adapt and learn effective policies.**

***Index Terms*—Deep Q-Learning, Reinforcement Learning, Urban Navigation, Autonomous Agents, Gymnasium, Stable-Baselines3**

I. Introduction

Urban traffic poses a significant challenge for autonomous navigation systems due to its dynamic and often chaotic nature. Traditional rule-based systems struggle to adapt in real time to unpredictable events such as congestion, detours, or sensor noise. Deep Q-Learning (DQN), a reinforcement learning technique that combines Q-learning with deep neural networks, offers a promising solution by allowing agents to learn from interactions and improve policies through experience.

This project aims to implement a DQN-based intelligent agent capable of navigating a simulated city environment. Building on previous work, we developed a modular architecture that integrates essential components such as sensors and actuators. Existing literature supports the application of DQN in similar contexts, with emphasis on decision-making under uncertainty and dynamic optimization.

II. Methods and Materials

The agent was implemented using Gymnasium as the simulation environment and Stable-Baselines3 for reinforcement learning algorithms. The state space was defined as a 4-dimensional vector: vehicle speed, proximity to obstacles, traffic light status, and time. The discrete action space consisted of: accelerate, maintain speed, and decelerate.

We designed a reward function with both positive and negative reinforcements:

- Positive rewards for reaching destinations quickly and obeying traffic rules.
- Negative rewards for collisions, running red lights, or inefficient routes.

Funded by Universidad Distrital Francisco José de Caldas.

The learning process minimized the following loss function adapted from the Bellman equation:

$$L(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(Q(s,a;\theta) - (r + \gamma \cdot \max_{a'} Q(s',a';\theta^{-}))\right)^2\right] \quad (1)$$

We included a target network to stabilize training and executed multiple scenarios:

- **Baseline**: Moderate traffic and simple roads.
- **Heavy Traffic**: High density, forcing adaptive behavior.
- **Complex Routes**: Intersections and dead ends.
- **Sensor Degradation**: Added noise to test robustness.
- **Dynamic Rewards**: Changed mid-training to test adaptation.

The agent progressively learns how to move and orient itself better by analyzing its environment during each attempt. It considers traffic light status, nearby buildings or obstacles, and its own speed to refine its decisions over time.

III. Results

Multiple experiments were carried out by varying the number of training episodes (i.e., map repetitions). As expected, the results confirmed that the agent's learning quality improves with a higher number of episodes. With few repetitions, the agent often gets stuck or loops in place. However, with a significant number—such as 30,000 episodes or more—the agent demonstrates clear improvements in navigating the city more effectively and reaching target destinations with greater consistency.

IV. Conclusions

Although the final version of the system does not meet all initially expected goals—particularly due to the basic interface and pending features—the project forms a solid base. The agent successfully performs its main function: it learns from repeated interactions with the virtual environment and improves its ability to reach goals over time. This adaptive behavior proves the viability of the approach and provides a foundation for future enhancements in visualization and agent capabilities.

References

[1] Farama Foundation, "Minigrid," [Online]. Available: https://github.com/Farama-Foundation/minigrid
[2] DigitalOcean, "Getting Started with OpenAI Gym," [Online]. Available: https://www.digitalocean.com/community/tutorials/getting-started-with-openai-gym