Self-Driving Car Racing Game: A Reinforcement Learning Approach

CSE440,
Section-01
Group-10
S. M. Samiul Haq - 2011103042, Jannatul Ferdous Prome - 2021770042, Md. Rajib Hossain-1921038042, MD Mohaiminul Islam-2012138042,
Department of ECE, *North South University*

***Abstract*— This project aims to develop a self-driving car racing game where AI agents learn to navigate racetracks and improve lap times using reinforcement learning. The objective is to simulate an environment where players can train their own AI drivers, competing based on learned driving strategies. This report summarizes our progress so far, initial implementation, and challenges, while outlining our roadmap for the final system.**

Keywords: Reinforcement Learning, Q-Learning, PyGame, AI Racing, Game Simulation, Autonomous Agents.

I. Introduction

The "Self-Driving Car Racing Game" project blends gaming with artificial intelligence by enabling autonomous car agents to learn racing behavior via reinforcement learning. The goal is to allow AI agents to learn through trial and error, receiving rewards for efficient driving and penalties for collisions or going off track. Ultimately, players will be able to train, customize, and compete with their own AI drivers in simulated races.

This system simulates core principles of reinforcement learning in a visually interactive environment, making complex AI behavior tangible through gameplay. This report provides an update on recent progress, challenges, and next works.

II. Objectives

The major goal of this project will focus on training an AI agent to learn how to drive in a car on a racecourse, all on its own with the help of reinforcement learning. Particularly, it targets the following goals:

- Autonomous Navigation: Create an AI agent that can navigate through a 2D racetrack on its own It should not need a human to move the agent
- Learning With Experience: Introduce to it a reinforcement learning system that will enhance working in scenarios of driving-learning through trial-and-error actions
- Lap Completion: Train the agent to make complete laps of the race track without fail
- Performance Optimization: Get able to improve their lap times as well as driving efficiency performance measurably throughout the training process
- Real-time Decision Making: Allow the agent to make real time decisions (acceleration, braking and steering) with respect to the environmental observations

The study aims to demonstrate the effectiveness of Q-Learning algorithms to control autonomous vehicles where the researchers aim to offer an understanding of how AI systems can develop motor control tasks using reinforcement learning paradigms.

III. Progress

A. Game Environment Design

We have chosen PyGame to develop a lightweight 2D racetrack environment. The basic race map has been created with walls and checkpoints. Car agents can currently move forward and turn. Collision detection with track borders has been implemented.

B. Reinforcement Learning Framework

We have implemented a simplified Q-Learning algorithm. The state space includes position, velocity, and orientation, while actions consist of move forward, turn left, and turn right. A preliminary reward system has been defined:

- +1 for moving forward without collision
- -100 for collision
- +100 for completing a lap

IV. Challenges

- **State Complexity**: Designing a minimal yet meaningful state space for the agent is difficult without overfitting.
- **Training Time**: Even simple models take many episodes to show improvement. Computational limitations slow down testing.
- **Reward Tuning**: Finding the right reward balance between speed and safety is ongoing.
- **Collision Logic**: Detecting and reacting to wall collisions in a 2D map needed custom logic.
- **User Customization**: We're still evaluating how much user control to allow over agent parameters within time constraints.

V. Future Work

a. **Multiplayer Game:** Add several AI bots in a race together

b. **Advanced Physics:** Model in real world vehicle dynamics with friction, momentum and aerodynamics

c. **Complex Track Design**: Create more complex tracks which have different terrain and obstacles

d. **Weather Conditions:** Include on environmental effects such as rain, wind, and things such as changing track surfaces

e. **Deep Q-Networks (DQN**): Use neural network underlying Q-learning to improve over the approximation of functions

f. **Policy Gradient Methods:** Discover actor-critic methods in space continuous action

g. **Transfer Learning:** Allows transferring knowledge between two tracks

h. **Hierarchical Reinforcement Learning:** Reinforcement of high level strategy learning and low level control

i. **3D Environment:** 3d Environment: Renovate to 3d graphics in Unity or Unreal Engine

j. **Dynamic in Real-time Adaptation:** Utilize the on-line learning of dynamic track conditions

k. **Human-AI Interaction**: create interfaces that allow human players to take part in an AI competition

l. **Performance Analytics**: Advanced metrics and visualization tools for training analysis

VI. Conclusion

The present project shows that it is possible to use reinforcement learning in the control of autonomous vehicles on a simplified race track. The Q-Learning algorithm demonstrated that it is effective to use it to train an AI agent to travel through a racetrack, completing laps and making improvement in the performance measurable and constant. The project has interesting lessons to offer on the potential cons and pros of using RL in real-life situations of autonomous driving.

Conclusions confirm the potentiality of reinforcement learning to control autonomous vehicles, whereas the revealed limitations indicate the research directions in order to improve that. This project is reasonably well-documented, with a modular design that may serve as a good basis of more research and development with regard to advanced autonomous driving.

References

[1] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," 2nd ed. Cambridge, MA: MIT Press, 2018.

[2] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[3] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, no. 3-4, pp. 279–292, 1992.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.

[5] S. Levine et al., "End-to-end training of deep visuomotor policies," Journal of Machine Learning Research, vol. 17, no. 1, pp. 1334–1373, 2016.

[6] M. Bojarski et al., "End to end learning for self-driving cars," arXiv preprint arXiv:1604.07316, 2016.

[7] A. Dosovitskiy et al., "CARLA: An open urban driving simulator for autonomous driving research," Proceedings of the 1st Annual Conference on Robot Learning, pp. 1–16, 2017.

[8] P. Wolf et al., "Learning to drive in a day," arXiv preprint arXiv:1807.00412, 2018.

[9] A. Kendall et al., "Learning to drive in a day," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 8248–8254, 2019.

[10] J. Chen et al., "DeepRacing: Learning to race autonomously using visual deep reinforcement learning," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6186–6193, 2021.