Deep Reinforcement Learning for Autonomous Driving Strategy

Ashwin S H
Reg. No. 180953352

Under the guidance of

Dr. Rashmi Naveen Raj
Associate Professor
Dept. of I&CT
MIT Manipal

Presentation Overview

Introduction

Autonomous vehicles:

- Class of vehicles that can perceive its surroundings and move with little or no human interaction.
- Utilize multiple sensors such as LiDARs, RADARs, GPS, cameras etc.

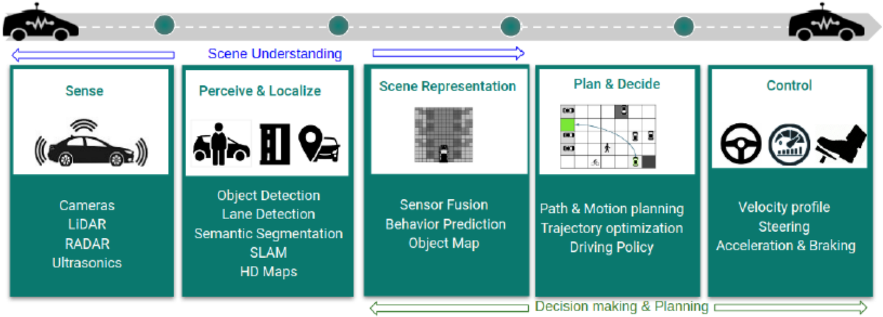

Figure 1: Standard blocks of autonomous driving system [1]

Reinforcement Learning:

- Branch of Machine Learning in which an autonomous agent learns and improves its performance by interacting with its environment.
- The agent chooses an action for each state it encounters and receives a reward.
- The goal for the agent is to maximize the cumulative rewards received over its lifetime.

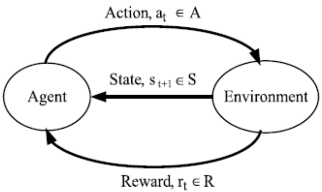

Figure 2: General scenario of Reinforcement Learning [2]

Introduction

Deep Reinforcement Learning (DRL):

- Combines Reinforcement Learning with Deep Learning.
- Sometimes, the states are very high dimensional and cannot be solved by traditional RL algorithms.
- DRL algorithms represents the policy as a neural network.

Introduction

Commonly used Deep RL Algorithms

- Deep SARSA
- Deep Q Learning
- Double Deep Q Learning
- Deep Deterministic Policy Gradient

Introduction

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q)^2)$
    Update the actor policy using the sampled gradient:

$$\nabla_{\theta^\mu}\mu|_{s_i} \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

Figure 3: DDPG algorithm[6]

Literature Survey

- Aims to model the decision making and interactions between various vehicles which run on highways.
- Double Deep Q-Network (DDQN) is employed to train the host vehicle.
- An open source simulation platform called 'SUMO - Simulation of Urban Mobility' is used to implement the work.

[Zhao et al., 2020 [3]]

Literature Survey

- Deep Deterministic Policy Gradient (DDPG) algorithm is throughly explained.
- This work establishes the mapping of driving state to driving action using DDPG algorithm on TORCS simulation software.
- Single agent scenario.

[Huang et al., 2019 [4]]

Literature Survey

- Aimed at steering the vehicle in its path with the help of Deep Q-Learning algorithm.
- The model uses raw images, sensor inputs and calculated rewards to create a Q-value approximator which is used to steer the car in TORCS simulator.

[Chopra et al., 2020 [5]]

Literature Survey

- Car-following simulation model
- Used the distance between the two cars to calculate jerk in driving, time for collision etc.
- Reward function is developed by capturing real-time driver data from the lead car.
- Implemented in 'Next Generation Simulation' software

[Zhu et al., 2020 [8]]

Literature Survey

- Used Double Deep Q Learning to build vehicle speed control model.
- Implemented in SUMO software which has only straight roads and no turns.
- Reward function - each state is mapped to a reward expectation.

[Zhang et al., 2018 [7]]

Literature Survey

- Focuses on scene understanding, decision making motion planning and vehicle control.
- Changes that need to be made for environmental challenges.
- Talks about the need of various sensors for different scenarios.

[Elallid et al. 2022 [9]]

Literature Survey

- Talks about explainability of decisions taken by the Autonomous Vehicles(AV).
- AVs must be able to explicate what the see, do and might do when they interact with the environment.
- A conceptual framework for explainable AVs

[Omeiza et al. 2021 [10]]

Literature Survey

Table 1: Comparison of states, actions and rewards

Work by	**State space**	**Action space**	**Reward function**
Zhao et al., 2020[3]	$S = \{pos_x, pos_y, v_x, v_y\}$	$A = \{acceleration\ value\}$	$R = 1 - \frac{V_{max} - V_x}{V_{max}}$
Huang et al., 2019[4]	$S = \{v_{x,y,z} \epsilon R^3, \zeta \epsilon R^{19}, \delta, \theta\}$	$A = \{accelerate, brake, steer\}$	$R = vcos\theta - vsin\theta - v\delta$
Zhang et al., 2018[7]	GPS position, acceleration, relative speed and position	$A = \{a \mid a \epsilon \{accelerate, decelerate, maintain\}\}$	R = {-2, -1, 0, 1, 2} depending upon the action taken
Chopra et al., 2020[5]	Camera images	$A = \{left, half\_left, no\_action, half\_right, right\}$	$R = vcos\theta$

Problem Statement

- Most of the work focuses on only one car in the track.
- Also reward function is solely dependent on the speed of the vehicle.
- The problem statement of this work is to detect other cars and obstacles present on road and drive the autonomous vehicle keeping in mind all those obstacles.

Objectives

#1

To deploy a suitable RL algorithm to control the speed and facilitate lane keep operation of the car.

#2

To avoid collisions by detecting obstacles using sensors.

Methodology

Requirements

- TORCS-1.3.7
- Ubuntu 16.04
- Python with keras and tensorflow framework

Methodology

Figure 4: Driving Environment

Figure 5: During Training

Methodology

Table 2: Table of States [4]

States Symbol	Description	Range
$v(km/h)$	agent's velocity	$(0, 300)\ \epsilon R^3$
δ	offset between centre of the lane and the agent horizontally	$(-1, 1)\ \epsilon R^1$
$\theta(radian)$	departure angle between the lane and agent	$(-\pi, \pi)\ \epsilon R^1$
$\zeta(m)$	distance between lane and agent	$(0, 200)\ \epsilon R^{19}$
$DistF(m)$	Distance between the agent and obstacle(s) ahead of it	$(0, 200) \epsilon R^4$
$DistR(m)$	Distance between the agent and obstacle(s) on the right side	$(0, 200) \epsilon R^{14}$

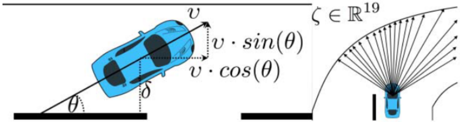

Figure 6: Driving Environment[4]

Methodology

Table 3: Table of Actions [4]

Action Symbol	Description	Range
accelerate	current acceleration	$(0, 1) \epsilon R^1$
brake	current braking	$(0, 1) \epsilon R^1$
steer	current steering angle: negative for right, positive for left	$(-1, 1) \epsilon R^1$

Reward function $R = v_x \cos(\theta) - v_x \sin(\theta) - v_x|\delta|$

Implementation

Algorithm Used:

- DDPG, an off policy algorithm
- Experience Replay and slow-learning target networks.

Functionalities Implemented:

- Lane Keep
- Front obstacle detection and avoidance
- Overtaking
- Side obstacle detection and avoidance

Implementation

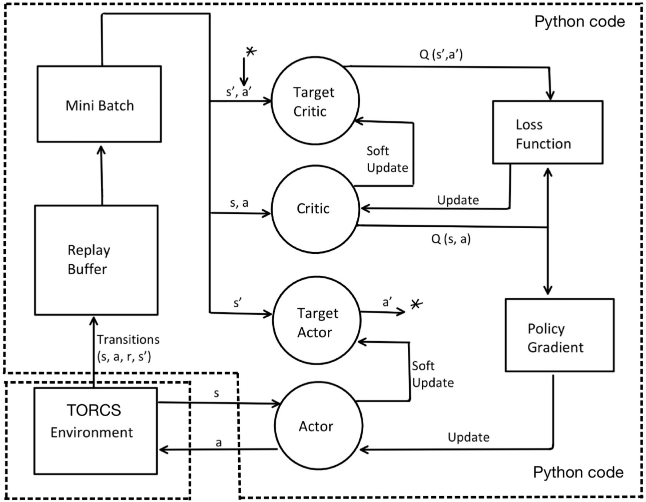

Figure 7: DDPG block diagram

Implementation

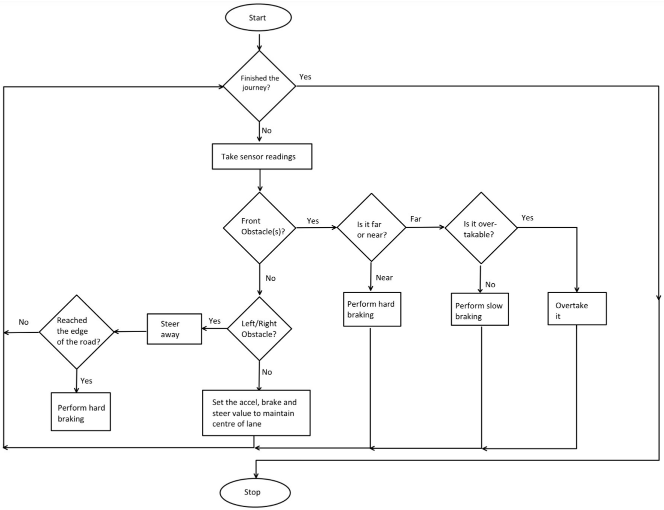

Figure 8: Obstacle detection and collision avoidance complete block diagram.

Implementation

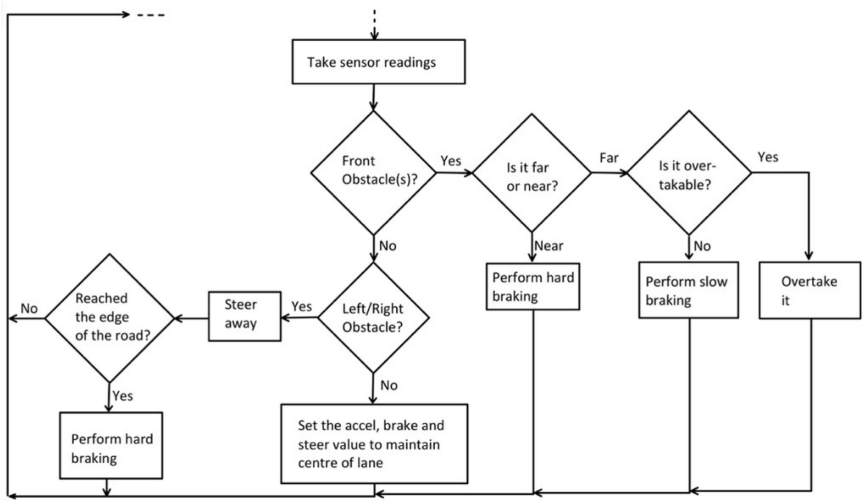

Figure 9: Obstacle detection and collision avoidance partial block diagram.

Results

Implemented DDPG algorithm to train the car to move correctly in the track.

1. Video footage during training
2. Video footage after the car has learnt to drive
3. Video footage of all obstacle detection and avoidance

Results

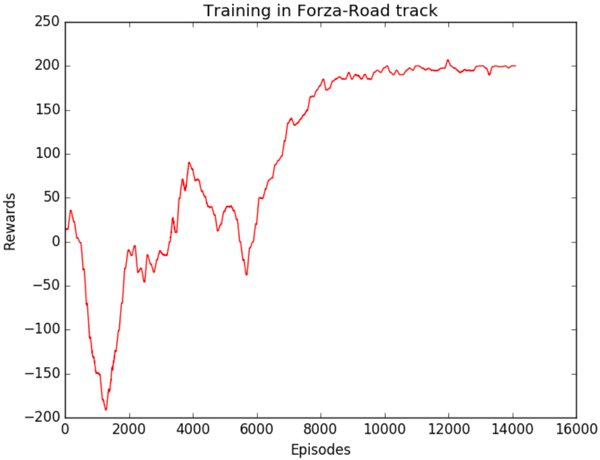

Figure 10: Training graph (Rewards vs Episodes)

Results

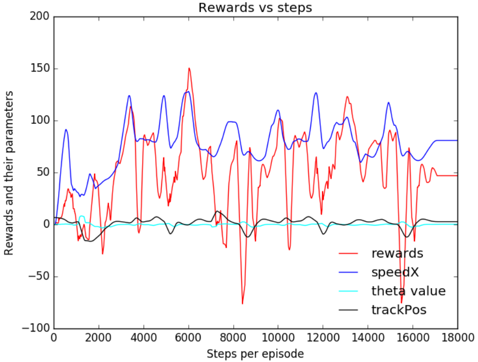

Figure 11: Rewards vs Speed, Angle and Lane-Position (track 1)

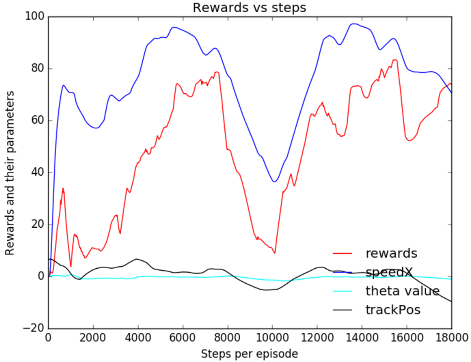

Figure 12: Rewards vs Speed, Angle and Lane-Position (track 2)

Results

Figure 13: Rewards vs Speed, Angle and Lane-Position (track 3)

Figure 14: Rewards vs Speed, Angle and Lane-Position (track 4)

Scope (who will benefit from this work?)

- Elderly people, patients on medication, persons with disabilities who will have to depend on other persons for mobility can now move around independently.
- Busy executives can attend to their work undisturbed while travelling
- Traffic accidents can be reduced.

Conclusion

- Self driving cars are the need of the hour due to increasing traffic and related accidents.
- DDPG algorithm is used to train the agent
- The agent can detect obstacles at the front as well as sides and take necessary actions.

References

Kiran, B Rav et al., (2021)
Deep Reinforcement Learning for Autonomous Driving: A Survey
IEEE Transactions on Intelligent Transportation Systems

Naveen Raj, Rashmi and Nayak, Ashalatha and Kumar, M Sathish (2020)
A Survey and Performance Evaluation of Reinforcement Learning Based Spectrum Aware Routing in Cognitive Radio Ad Hoc Networks
International Journal of Wireless Information Networks 144–163

Huang, Zhiqing and Zhang, Ji and Tian, Rui and Zhang, Yanxin (2019)
End-to-end autonomous driving decision based on deep reinforcement learning
2019 5th International Conference on Control, Automation and Robotics (ICCAR) 658–662

Zhao, Junwu and Qu, Ting and Xu, Fang (2020)
A Deep Reinforcement Learning Approach for Autonomous Highway Driving
IFAC-PapersOnLine 542–546

Chopra, Rohan and Roy, Sanjiban Sekhar (2020)
End-to-end reinforcement learning for self-driving car
Advanced computing and intelligent engineering 53–61

References

- Lillicrap, Timothy P et al., (2015)
  Continuous control with deep reinforcement learning
  arXiv preprint arXiv:1509.02971
- Zhang, Yi and Sun, Ping and Yin, Yuhan and Lin, Lin and Wang(2018)
  Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning
  IEEE Intelligent Vehicles Symposium (IV) 1251–1256
- Zhu, Meixin and Wang, Yinhai and Pu, Ziyuan and Hu, Jingyun and Wang, Xuesong and Ke, Ruimin (2020)
  Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving
  Transportation Research Part C: Emerging Technologies - 117
- Elallid, Badr Ben and Benamar, Nabil and Hafid, Abdelhakim Senhaji and Rachidi, Tajjeeddine and Mrani, Nabil (2022)
  A Comprehensive Survey on the Application of Deep and Reinforcement Learning Approaches in Autonomous Driving
  Journal of King Saud University-Computer and Information Sciences - Elsevier
- Omeiza, Daniel and Webb, Helena and Jirotka, Marina and Kunze, Lars (2021)
  Explanations in autonomous driving: A survey
  IEEE Transactions on Intelligent Transportation Systems

Thank you