International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume- 9, Issue- 3 (June 2019)
www.ijemr.net https://doi.org/10.31033/ijemr.9.3.01

Self-Navigation Car using Reinforcement Learning

Dr. Rafi U Zaman[1], Syed Mujtaba[2], Mirza Jawad Ali[3] and M. Saaduddin Ahmed[4]
[1]Associate Professor, Department of IT, Muffakham Jah College of Engineering and Technology, Hyderabad, INDIA
[2]Student, Department of IT, Muffakham Jah College of Engineering and Technology, Hyderabad, INDIA
[3]Student, Department of IT, Muffakham Jah College of Engineering and Technology, Hyderabad, INDIA
[4]Student, Department of IT, Muffakham Jah College of Engineering and Technology, Hyderabad, INDIA

[1]Corresponding Author: rafi.u.zaman@mjcollege.ac.in

ABSTRACT

In this paper, a project is described which is a 2-D modelled version of a car that will learn how to drive itself. It will have to figure everything out on its own. In addition, to achieve that the simulator contains a car running simultaneously &can be controlled by different control algorithms - heuristic, reinforcement learning-based, etc. For each dynamic input, the Reinforcement- Learning modifies new patterns. Ultimately, Reinforcement Learning helps in maximizing the reward from every state. In this first Part, we will implement a Reinforcement-Learning model to build an AI for Self Driving Car. Project will be focusing on the brain of the car not any graphics. The car will detect obstacles and take basic actions. To make autonomous car or self-driving car a reality, some of the factors to be considered are human safety and quality of life.

Keywords— Heuristic, Reinforcement Learning, Reward Function, Self-Driven Car

I. INTRODUCTION

As we know the drones have become familiar these days, the concept of autonomous cars also sounds cool**.** Self-Navigation car takes input dynamically and make actions accordingly**.** As per perception-based model, ithas cameras/sensors to detect objects and after detecting it navigate either forward, left or right. These types of project implementation can help in safe transportation, extemporize fuel efficiency and lower traffic. In this project, our neural network contains angle and rotation with some numeric, which helps the car to take better decision in real time. By using Reinforcement learning we are focusing on two main principles max reward and shortest path. The car is simulated such that it should reach destination point by making better decisions. Lastly, a graph is generated based on reward gain and time spend i.e. it ranges from -1 to 1. For further upgrading computer vision model can be used by using Convolution Neural Network (CNN) to extract features from raw images.

The focus of this project is to develop an autonomous car, which will be able to drive on its own without any human guidance once they are trained. Road traffic injuries caused an estimated 1.40 million deaths worldwide in the year 2015 and more than 50 million injured.[5] A study using British and American crash reports as data, found that 87% of crashes were due solely to driver factors.[6]. This autonomous cars be safe and would be beneficial for humans. To process this an Artificial Intelligence is required for car to propose a path in its own. Additionally a self-driving car can reduce the no of traffic jams and human errors. The distance between the cars can be reduced. The project will help in freeing parking places and the fuel will be saved. Self Navigating car is the future of the upcoming years. In the year 2013, the concept of autonomous car was introduced. It is spreading worldwide in order to avoid the risk of accidents especially by youngsters. Artificial intelligence is the trending topic, which can be used in order to achieve the result.

II. METHODOLOGY

A vehicle that can sense environment and navigate without any human interference is known as autonomous car. In real time, autonomous car takes dynamic inputs and based on max reward gain and shortest path followed the car reaches the destination. In our neural network, a function select_actionis defined to train the model and generate the output and simultaneously the reward is added. So we have designed a car which is simulated and has 3 sensors in it, so we find a diagonal way as the shortest path to reach the destination point. Here the user is given the authority to create obstacle in the simulator, based on reinforcement leaning the car detect the angle and rotation and make a decision to move either straight, left or right for getting shortest path. For every action taken by car the reward either gets added or subtracted until it reaches destination point. The algorithm get smoother by detecting obstacles one after the other. In addition, since this is perception-based model, more the obstacle more time is required to train the model and vice-versa. The analysis begin from negative point i.e. -1 when it is initializing the model to train and once the model is completely trained it end up with a positive point i.e. 1. If no user-defined obstacle is created the model is trained without taking much time, and if more complex obstacle is created, the model takes more time to find the shortest path. The graph from negative to positive shows that the model gets trained finding the shortest path to reach the destination point.

III. PRIOR APPROACH

Hyundai motors working on Integration of HD maps in autonomous cars for 2018 Olympics. Japan will be completely implementing self-driving car by 2020 for Tokyo Olympics. Baidu company china working on deep learning concepts for Apollo project. Presently in India TATA and Mahindra both are working on autonomous car. Intel working on multiple processors, which are much, needed for automated driving. Apart from Asian companies the European commission funded Eureka PROMETHEUS project for the development of driver assistant system. These projects were attracted by the German car manufacturer Volkswagen and leads to further growth in development of driver assistant technology. The US Company like Cruise, Waymo, Uber, and Lyft are working on commercial ride sharing services, which limits complicity, and avoid cost constraints. Many companies are developing LiDAR systems for improving object deduction accuracy and specifically Waymo has spent over $1.1 billion on its real world driving, training and testing. The autonomous companies like Intel and NVidia are playing major role in developing self-driving system.

IV. SYSTEM ANALYSIS

Existing System

Human Factor In Vehicle Collisions Contribute A Major Part In A Collision. Human Reaction Speed Is Higher Than 200ms [1]

Drawbacks

In 2015 almost 140,000 people were injured each day worldwide in traffic accidents .[5]. According to world health organization, road traffic injuries caused an estimated 1.35 million deaths worldwide.[2]. Stress and discomfort can in introduce.

Proposed System

An autonomous car in 2D simulator uses recurrent neural network. A path is being recognize using AI for the intelligent car to follow. No input data is collected, it takes immediate action based on reinforcement learning algorithm. Temporary storage of trained data can be loaded and cleared

Advantages

- Our simulated autonomous car can be used in gaming consoles in which our AI has to take better decisions.
- The disable person or the person who are not allowed to drive the car can also make better use of this autonomous car. This can provide benefits to the travellers.
- Human Reaction Speed Is Higher Than 200ms whereas the AI an reach up to the reaction speeds of up to 10ms. [4]
- A computer as a driver will never make an error.[3]
- A self driving car can be very safe and useful for the entire mankind.[3]
- On the other hand, self-driving vehicles can introduce new stresses and discomforts.[3]

Hardware Requirements

- RAM: At least 4GB
- OS: Linux based OS preferable (UBUNTU)

Software Requirements

- IDE:SPYDER
- Language: Python

V. PERFORMANCE EVALUATION

- Performance evaluation is done based on maximum reward gain and shortest path followed to reach the destination.
- Analysing max reward and shortest path
- After creating the architecture of neural network we then implement Reinforcement learning by defining select_action() function.
- This function is used for initializing model i.e. by taking inputs dynamically which car faces in real time.
- The function learn is making model learn by itself which generates output and a reward is added to it. Hence this output and rewards are updates dynamically.
- The Bellman Equation:

$$V(s) = max_a(R(s,a) + \gamma V(s')) \qquad \textbf{Eq. (1)}$$

Where, s –State
a –Action
R – Reward
γ - Discount
s'- Expected State

Reward is gained when the agent is in the state 's' and can take any number of action 'a'. The value of current state 's' is summed with the product of the next state ' s' ' and the discount. The maximum value is considered from all the possible actions plus the state of discount.

- The Markov Decision Process:

$$V(s) = max_a(R(s,a) + \gamma \sum P(s,a,s')V(s')) \qquad \textbf{Eq. (2)}$$

Where, s –State
a –Action
R – Reward
γ - Discount
s'- Expected State

Here, the evaluation of Markov decision process is modeled after the Bellman Equation with the extension of the summation of all the possible states when the action 'a' is taken.

$$TD(a, s) = R(s, a) + \gamma max_{a'} Q(s', a') - Q_{t-1}(s, a) \quad \textbf{Eq. (3)}$$

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha TD_t(a, s) \quad \textbf{Eq. (4)}$$

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha(R(s, a) + \gamma max_{a'} Q(s', a') - Q_{t-1}(s, a)) \quad \textbf{Eq. (5)}$$

Temporal Difference

The value of the action taken is the result of the sum of the reward for that action and the product of the discount value 'γ' and the maximum reward is taken into consideration. Equation **(4)** is the simplified version of equation **(3).** Equation **(5)** is derived from the equation **(4)** which gives us the clear understanding of the temporal difference.

Simulator Setup

● Simulator is setup in such a way that a car model from source point has to reach destination point
● Our aim is to find shortest path, we find a diagonal way as a shortest path to reach destination point,
● The model is designed in car. ky file. After getting our AI, which contains neural network, we set angle and rotation with some numeric property.
● Model has three sensors, which make a decision of moving straight or right or left. With change in positions of car the reward also get updated simultaneously until it reaches destination point
● Simulator also contains three API buttons i.e. save, clear and load.
● On clicking save ,it saves the current AI and displays the obstacle created by user with a performance graph
● With load () function /button we retrieve/load the last saved AI.
● Clear button clears all the obstacles created by user and re-initialize the AI.

Results Discussion

● This 2-D graph is generated with time in ms on X-axis and reward on Y-axis. This graph is nothing but analysis our whole project which starts with negative points and ends with positive.
● Based on the how complex the obstacle is created, accordingly the model takes times.
● The reward range is from -1 to +1, -1 indicates no AI is created and couldn't find shortest path. Similarly, +1 indicates AI is build i.e. training on user defined obstacle and is so close to reach destination. As shown in fig.1 the graph initially is negative and slowly moves towards position indicating the model is training and gives us a success.

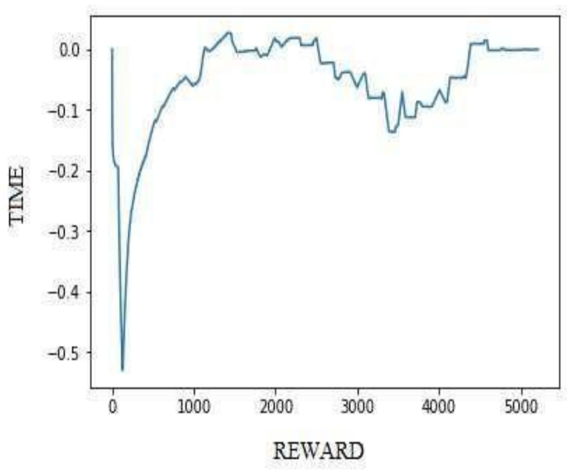

Fig 1 .Learning Graph b/w Time(ms) Vs Reward

VI. EXPERIMENTAL RESULTS

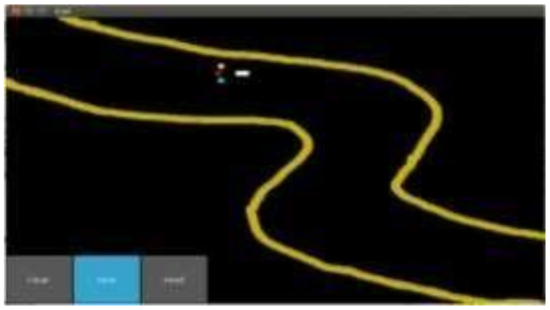

Fig2. Simulation of the proposed protocol

The car/agent moving from one state to another by taking action based on reinforcement learning reaches destination diagonally. These sensors on the car plays an important role in making decision of moving either straight, left or right. Based on computational ability, the car takes longer time to take decisions. The API buttons present on the simulator are Clear, Load, Save woks until simulator is

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume- 9, Issue- 3 (June 2019)
https://doi.org/10.31033/ijemr.9.3.01

running. More the complexity of obstacle created more the agent takes the time.

When the user wants to save his /her obstacle the save API button comes into picture. It temporary saves into memory and generate the performance graph of the AI. Suppose the user wants to redesign or to create more complex obstacle then the load function retrieves the last saved AI. Clear button clears all the obstacles created by user and re-initialize the AI. Finally, if user is ready to produce the trained output, a performance graph of reward and time is generated.

VII. CONCLUSION

As we can see many socio-economic motivators in adoption of self-navigating cars, our project with well-designed trained and tested with a recurrent neural network model helps in predicting angle and rotation and thus improvising human safety, quality of life and infrastructure efficiency. Given a complex situation, the self-driving model takes time to process and make a better decision. The importance of platooning, pooling and improvising the object-detecting feature will definitely make an autonomous car a reality in future.

REFERENCES

[1] Arthur Juliani. (2016). *Simple reinforcement learning with tensorflow. (Part 4)*. Available at: https://medium.com/@awjuliani/simple-reinforcement-learning-with-tensorflow-part-4-deep-q-networks-and-beyond-8438a3e2b8df.
[2] W. H. Organization. (2015). *Global status report on road safety*. Available at: https://www.who.int/violence_injury_prevention/road_safety_status/2015/en/.
[3] Daily M., Swarup, M., Trivedi, M. (2017). Self-driving cars. *Computer*, *50*(12), 18-23. Available at: http://ieeexplore.ieee.org/document/8220479/.
[4] Gargi Sharma. (2017). How artificial intelligence is outpacing humans. Available at: https://dzone.com/articles/how-artificial-intelligence-is-outpacing-humans.
[5] D.J White. (1993). *A survey of applications of markov decision processes*. Available at: http://www.it.uu.se/edu/course/homepage/aism/st11/MDPApplications3.pdf.
[6] https://www.bbc.com/news/world-asia-india-36496375.
[7] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.