Autonomous Driving in a Multi-Lane Highway Environment

Krithika Balasubramanian, Akhil Kothari, Vijayakumar Kuppusamy

Abstract: Our goal through this paper is to figure out if it is possible to create an autonomous driving environment with a self-governing car with the help of a Q learning algorithm, a variant of Reinforcement Learning. To prepare and test-driving calculations, we convey a reproduced traffic framework simulation. We plan to split the environment around the agent vehicle into 16 states. The Q learning algorithms calculations, which are based on the Bellman's Equations, will help quantify the quality of each state, helping the agent make the right decisions in the environment to avoid collisions. The World health organization reports highlight that in 2019 there have been over 5 million reported road accidents with approximately 1.5 million causalities and an increase of 167% in road accidents over the last 15 years. Through this paper, we want to push the envelope concerning creating a more secure driving environment and help avoid unfortunate accidents and loss of lives.

Keywords: Autonomous driving, Q Learning, Multilane, Reinforcement learning.

I. INTRODUCTION

The focus of every new scientific finding and invention revolves around making human life safe and comfortable. We have almost replaced human labor involved in various day to day activities to reduce our strain with the invention of machines like washing machines and dishwashers. Although driving a car that requires immense concentration and skill, it is one domain where complete efforts have not been made to help overcome human errors.This paper explores how and to what degree a vehicle can be driven without collision in a multilane highway environment, a recreated traffic condition is set up to achieve this. Artificial Intelligence calculations require extensive preparation and testing. Since the expense and dangers of preparing calculations of individual vehicles in certain rush hour gridlock situations are restrictive, we have set up a reenacted traffic condition to test how practical the reinforcement learning approach is to create an autonomous environment.

II. LITERATURE SURVEY

Reinforcement learning is a very active area of research interest in data science. It helps an agent to reach full potential concerning the total amount of reward it can receive while interacting with a sophisticated, new, and uncertain environment through a computational approach of learning. The authors Richard Sutton and Andrew Barto provide a simplified format of their key ideas and algorithms towards the implementation of reinforcement learning. Their discussion covers a varied range of approaches from the field's history to the most recent developments and applications. The only fundamental necessary mathematical background is knowledge concerning elementary concepts of probability. [7]

Q-learning is a form of reinforcement learning approaches and also easily one of its most applied representatives and one of the off-policy strategies. After the development of Q-learning, many research problems have implemented their algorithms in various data science problems. With the vast advances in the field, more variants of Q-learning, especially like Deep Q-learning, which combines the basics of Q learning with deep neural networks, this new variant has been the most successful in helping solve problems and develop new applications. The paper thoroughly explains how they have used the evolved Q-learning algorithm by unraveling the mathematical complexities in its algorithms as well as its flow from the family of reinforcement algorithms to help handle its existence in solving new problems. [8]

This paper has worked creatively on developing a smart autonomous driving model. Their key idea was to implement their concept with the help of a reinforcement learning approach to understand the extent to which they can successfully implement an autonomous model. Simulations show that their method can learn feasible overtaking policies in different traffic environments, and the performance is comparable or even better than manually designed decision rules. Their paper lacked the idea of a 4-Lane highway environment model to test the vehicle's decisions to move lanes rather than depending on varying velocity. [3]

This paper works on implementing Deep Q learning to solve problems cart pole swing-up and legged-locomotion. The algorithm proposed provides strategies that have a competitive execution, which is established by an algorithm that gives an entire overview of the domain and its subparts. [10] This paper describes the working of a deep neural network and Q-Learning. It also talks about the Double Q-Learning algorithm's fundamental theory, which was first found in a tabular environment. The next talks about how Double Q learning is used in real-world cosmic problems. [11] This paper works on deep Q network and tests this agent on the classic Atari 2600 games.

Revised Manuscript Received on June 30, 2020.
* Correspondence Author

Krithika Balasubramanian*, Associate Professor, in Department of Information Security, at School of Computer Science and Engineering

Akhil Kothari, Associate Professor, in Department of Information Security, at School of Computer Science and Engineering

K. Vijayakumar, Associate Professor, in Department of Information Security, at School of Computer Science and Engineering

Getting just the pixels and the game score as information sources, the algorithm was able to outperform the working of every algorithm introduced over the last few years and accomplish a level practically identical to that of an expert human game analyzer. This analysis was done for almost 49 games sets with the help of the same algorithm. [12] The proposed system in this paper works on deep reinforcement learning and provides an insubstantial framework that works on the principles of asynchronous gradient descent. These principles help in the better development of the deep neural network. From this proposed system, it shows that a parallel actor-learners have a steady effect on training, which gives the ability for all four reinforcement learning methods to train neural network controllers. [13]

This paper explains the concept of Q learning. Q learning is used for a dynamic environment by rewarding the agent based on the interactions it makes with the environment. In this paper a system is proposed where the agent should provide continuous actions to the reaction given by continuous states. The method comprises a neural network that is integrated with an unconventional interpolator. [15]

III. PROBLEM STATEMENT

The existing systems use machine learning concepts to achieve control over an autonomously controlled agent car in a multi-lane environment with the help of actions such as changing of lanes, accelerating along with the ability to decelerate steadily. Existing models have made these decision-making skills a possible reality with the help of data science algorithms, which would help avoid accidents in a multi-lane environment. Still, the results obtained by these algorithms lack accuracy and hence show a large collision percentage.

The system that we are proposing through this paper, woks towards analyzing and figuring out if reinforcement learning can be a long-term solution towards achieving an autonomous driving environment. We will create a traffic simulation to explore the range of help reinforcement learning can offer to achieve autonomous driving. One of the prominent Reinforcement learning approaches is Q learning, which will be used to maneuver the autonomous car in our simulation environment. We plan to pursue and explore how effective the deep Q-learning algorithm is to create an autonomous driving environment.

IV. PROPOSED SYSTEM

Our system is based on reinforcement learning, where the agent will train and adapt to the environment by rewarding itself. Our Artificial Intelligence calculations are to help the agent vehicle make better decisions are based on Bellman's Equations. In the proposed system, which is illustrated in Figure 1, Q learning is a reinforcement learning which does not involve any model for the learning of the algorithm. It gives operators the ability to figure out how to act ideally by encountering the activities' results without expecting the vehicles to manufacture direction in the nature area. The car movements are decided to depend upon the Q esteem concerning the state's activity pair. Q esteem is the deciding factor that aids the agent in traveling from one place to another and decides on the most suitable arrangement to determine the subsequent actions. The activity which provides the most significant hike in the Q esteem value from all the available activities is picked as the driving choice for the relating starting or current state.

Towards the beginning of our preparation, the agent operator is ignorant of making the right decisions. Initially, a large segment of the steps is made indiscriminately, which is indicated by the Exploration rate. The refreshment of the Q table is vastly dependent on determining Q esteem values. With progress in our preparation, the investigation rate diminishes, hence the progressive moves are a result of the Q esteems and not the arbitrary. For our driving domain, the range of conceivable info states is very high. Thus, it's hard to keep up with a Q table.

Along these lines, for the development of the agent, Deep Q Network is incorporated. The Neural framework takes in all the information acquired. This framework consists of different hubs in different layers—the hubs in the yield layer contrast with the driving administrator's potential activities. During the preparation time frame, many of these hubs are changed to such an extent that the yield layer's hubs relate to the Q-esteems for the activities.

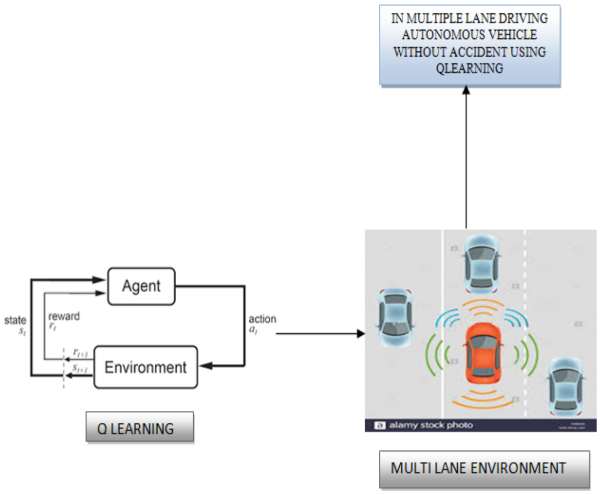


Figure 1: System Architecture for the proposed system

V. AUTONOMOUS VEHICLE'S MODULES

A. Activities of the driving expert

Possible activities for any mode of transport in a roadway circumstance:

1) Alternate lane to evade a crash.
2) Accelerate.
3) Decelerate.
4) Idle Action.

The inert activity communicates that the independent vehicle can remain in a comparable way and with a relative speed that it had during the past advance.

B. Internal properties of the driving agent

The significant tasks of a mode of transport in a highway environment are either moving to another way or altering its speed. In the arrangement illustrated in Figure 2, the autonomous vehicle is just utilizing the car data of the cars in a split-second encompassing.

The data involves the current detachment, di, to and the speed, vi, of all of the including vehicles (a = 1, 2, 3, 4, 5, 6), where a = 1 and a = 2 mean the vehicle in front and behind in a comparative way, a = 3 and a = 4 demonstrating the vehicle in front and behind one way to the other side, and a = 5 and a = 6 indicating the vehicle in front and behind one way to the other side. On the off chance that such a path to one side or to the privilege doesn't exist, because the self-governing vehicle is as of now driving in the furthest left or furthest right path, the significant separation is set to the highest worth, and the pace is valued to 0. At long last, the present condition of the self-ruling car, in other words, its speed, va, increasing speed rate, aa, and path list, La, are utilized as info.

C. Reward - a significant instrument for training the agent

We can say, Rewards can be considered as a significant piece of fortification. Based on the rewards obtained, the training of the agent keeps developing for the better. Given the reward framework, the driving specialist can be extremely wary, attempting to stay away from impacts by moving genuinely moderate. The agent usually trains itself to arrive at the highest velocity and decrease driving duration at all expense. Planning of the reward component is a vital procedure. Rewards are given with the end goal of the independent vehicle attempting to adhere to the fundamental traffic rules and start a surpass at whatever point conceivable. A high negative reward is given if there should arise an occurrence of a crash, as it is the most horrible outcome. Additionally, a negative reward is given if the speed is "0" to demoralize the agent specialist from halting the interstate's agent. Negative rewards are additionally given if the agent is near the car in front or on the off chance that it is in the overwhelming paths pointlessly, for example, on the off chance that no vehicles are present at the close by the path. To urge the agent to move at as far as possible, a positive reward is given in the event that it keeps up as far as possible, and also, the agent receives a negative reward if it deviates from the proposed speed.

Likewise, the positive reward is awarded if the agent attempts to overwhelm more slow-moving vehicles. The complete rundown of remunerations and the factors utilized are given in Table 1. We see L0 is currently the furthest right path in the street. Also, Lmax is present in the most distant left path. Distance proximity is the separation between the vehicles that are viewed as excessively close.

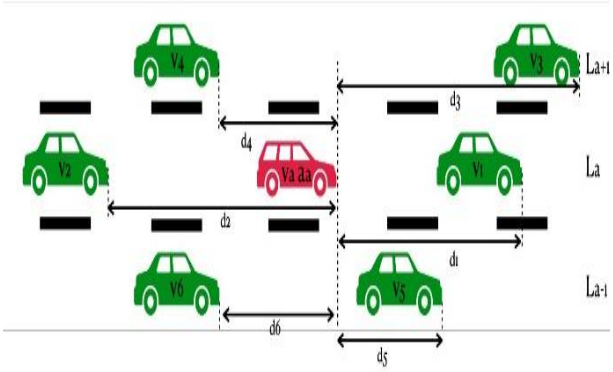


Figure 2: Calculate the proximity between cars

Table 1: Reward Calculation method for the proposed scenario

S. No	Reward	Reward Condition
1	-101	Collision
2	-50	Else & $v_a = 0$
3	-5	Else & La f= $L\max$ & $d_1 < d_{\text{proximity}}$
4	$50 - d_5$	Else & L_a f= L_0 & $d_5 < d_{\text{proximity}}$ & $a_a > 0$
5	$-1.5 \times d_5$	Else & L_a f= L_0 & $d_5 > d_{\text{proximity}}$
6	0.5	Else & $L_a = L_{\max}$ & $d_1 < d_{\text{proximity}}$ & $a_a < 0$
7	-0.5	Else & $L_a = L_{\max}$ & $d_1 < d_{\text{proximity}}$ & $a_a > 0$
8	-1	Else & $v_a > SpeedLimit$
9	1	Else & $a_a > 0$
10	2	Else & $v_a = SpeedLimit$
11	0	Else

VI. MATHEMATICS INVOLVED

Reinforcement learning works based on the action-reward principle. Here, the agent interacts with the environment and learns the working of the system. Based on the learning and decisions the agent takes, it rewards itself, and a new state is created. An agent may or may not be completely aware of the environment. Reward calculation is something the agent is already aware of, which is considered a function of its actions. A reward is a function in which the agent cannot be changed randomly. In certain situations, the agent finds it challenging to calculate the maximum reward despite knowing the entire environment well.

Mathematically speaking, reinforcement learning can be formulated using the Markov Property,

$$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots\ldots.., S_t] \qquad (1)$$

Which states that "Future is Independent of the past given the present." In equation 1, S[t] denotes the current state of the agent, and S[t+1] denotes the next state. And this property can be used to derive the Markov Reward Process as mentioned in equation 2.

$$R_s = E[R_{t+1} \mid S_t] \qquad (2)$$

The Bellman equation is used to reward the agent and make the learning process for the agent more efficient.

$$V(s) = E[R_{t+1} + \gamma\,(S_{t+1}) \mid S_{t=s}] \qquad (3)$$

In equation 3, we can see that the value of a state can be decomposed into immediate reward(R[t+1]) plus the value of successor state (v [S (t+1)]) with a discount factor(γ).

VII. PROJECT MODULES

The project had two significant modules:

A. Creating the environment

A 4-lane roadway condition is created utilizing the Simulation of Urban Mobility. Here our self-governing agent car is constrained because of fortification sort of learning. All the parallel vehicles in the environment are constrained by the Simulation of Urban Mobility itself. All these parallelly running vehicles are viewed as a similar length, which is 3 meters.

Different cars in the environment are embedded into the reproduction on irregular occasions following Poisson dissemination. It is a more slow-moving sort of car where the self-ruling agent car can surpass quickly with the most extreme speed like the self-sufficient vehicle. The likelihood of embeddings the more slow-paced agent (speed = 11.1 m/s) into the reenactment is around 0.1 every passing second and for the other kind of cars (speed = 55.55m/s) is set to 0.01 every passing second. All the cars in the environment are configured to maintain all the traffic rules and also keep up a base separation with the car ahead of them. This implies that the cars in the environment would not start any crash; henceforth, impacts must be brought about by the self-ruling vehicle. In the self-ruling vehicle, a crash evasion framework outside the AI learning calculation is utilized. These calculations help identify if the car ahead is exceptionally near and consequently slows down to dodge backside impacts by the self-governing vehicle. This is being used as attempting to evade the effects using only the learning calculation was insufficient, because of the number of crashes being exceptionally heavy.

Henceforth, the impacts now are just conceivable while car changes path. As far as possible, the speed on the thruway is maintained at 22.22m/s. Every new simulation scene recreation comprises of 160-time ventures with each time step sticking to one second. Our self-ruling car is gone into the recreation at the 60th time step, so there will be vehicles out and about before the independent vehicle. As the learning calculation is for controlling the self-ruling vehicle, it is dynamic just while the self-driving car is in the reenactment. If there should be an occurrence of a crash, the present scene is finished.

B. Autonomous Car Training Module

A replay of experience is utilized efficiently to prepare the neural system for the agent operator. The information state, new destination state, activities, rewards, and also the scene end status during each time step is spared in their memories, and some of the past encounters are picked indiscriminately to prepare a neural system as back to back time steps are exceptionally connected.

The investigation rate is set to a high estimation of 0.9 toward the beginning of preparing and it is diminished exponentially during each time step by a factor of 0.9992 to such an extent that before the finish of the preparation the vast majority of the activities by the operator are taken dependent on the most extreme Q esteem for the activities. The entire implementation was done using Python.

VIII. RESULTS AND DISCUSSION

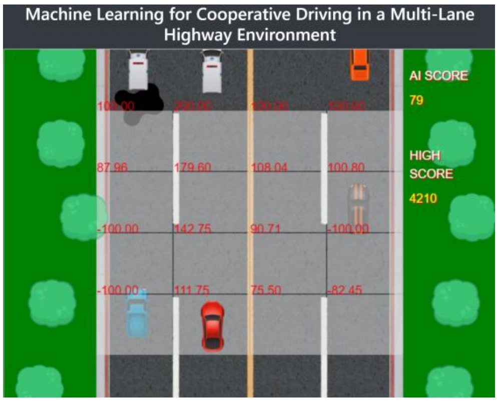


Figure 3: Agent vehicle in motion

Figure 3 is a screenshot of the multi lane highway simulation, the 4-grid blocks are the surroundings where the agent is evaluating its environment and the lane changing decisions are bound to be taken based on their results.

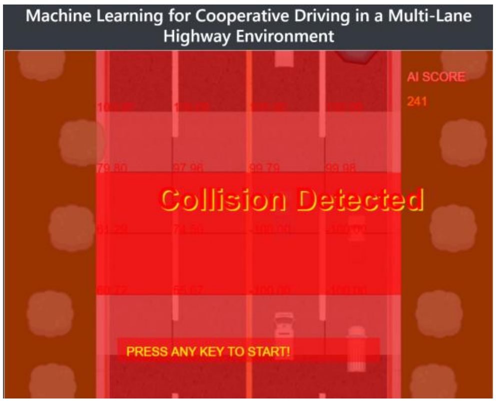

Machine Learning for Cooperative Driving in a Multi-Lane Highway Environment
AI SCORE
241
79.80
97.96
99.79
Collision Detected
PRESS ANY KEY TO START!

Figure 4: Agent Vehicle Collision detected

Figure 4 is a screenshot of the simulation ending when a collision between the agent vehicle and another object/ vehicle is detected during the drive.

IX. CONCLUSION

In this paper, our goal was to implement the Q-learning algorithm, which works on Bellman's Mathematical Equations, to create a simulation model where a vehicle can navigate autonomously in a multilane highway environment. The agent is wholly trained based on the Q-learning algorithm. The results obtained from the simulations show a decrease in collision probability with every successive simulation run. The simulated environment is created, keeping in mind the basic traffic rules implemented in a real-life highway scenario. This gives the agent an ideal environment to train. But this system lacks certain factors that involve slowing down when it does not have the luxury of changing lane, resulting in a collision at some point in every simulation. The environment created is a straight lane highway. This gives the system more scope to develop on different highway types ranging from a single lane highway, creating traffic signals, etc. The proposed method provides a positive step towards achieving an autonomous driving environment using Q- Learning in the near future.

AKNOWLEDGEMENT

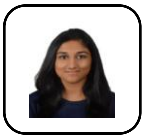
We would like to thank the Vellore Institute of Technology, Vellore for giving us this opportunity and support throughout the course of this work. We would like to extend our sincere gratitude to our professor Prof. Vijaya Kumar K and all other faculties for the knowledge and information that was taught and shared in class.

REFERENCES

1. S. K. Gehrig and F. J. Stein, “Dead reckoning and cartography using stereo vision for an autonomous car,” in Intelligent Robots and Systems, 1999. IROS’99. Proceedings. 1999 IEEE/RSJ International Conference on, vol. 3. IEEE, 1999, pp. 1507–1512.
2. S. Ishikawa and S. Arai, “Cooperative learning of a driving strategy to suppress phantom traffic jams,” in Agents (ICA), IEEE International Conference on. IEEE, 2016, pp. 90–93.
3. X. Li, X. Xu, and L. Zuo, “Reinforcement learning based overtaking decision-making for highway autonomous driving,” in Intelligent Control and Information Processing (ICICIP), 2015 Sixth International Conference on. IEEE, 2015, pp. 336–342.

4. M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo–simulation of urban mobility,” in The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain, vol. 42, 2011.
5. F. Chollet et al., “Keras,” 2015. [Online]. Available: https://keras.io
6. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., “Tensorflow: a system for largescale machine learning.” in OSDI, vol. 16, 2016, pp. 265–283.
7. R. S. Sutton, A. G. Barto et al., Reinforcement learning: An introduction. MIT press, 1998.
8. C. J. Watkins and P. Dayan, “Q-learning,” Machine learning, vol. 8, no. 3-4, pp. 279–292, 1992.
9. Szepesvári, C. (2010). “Algorithms for reinforcement learning”, Synthesis lectures on artificial intelligence and machine learning, 4(1), 1-103.
10. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015), “Continuous control with deep reinforcement learning”.
11. Van Hasselt, H., Guez, A., & Silver, D. (2016, March), “Deep reinforcement learning with double q-learning”, In Thirtieth AAAI conference on artificial intelligence.
12. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015), “Human-level control through deep reinforcement learning”, Nature, 518(7540), 529-533.
13. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June), “Asynchronous methods for deep reinforcement learnin”, In International conference on machine learning (pp. 1928-1937).
14. Ng, A. Y., & Russell, S. J. (2000, June), “Algorithms for inverse reinforcement learning”, In Icml (Vol. 1, p. 2).
15. Gaskett, C., Wettergreen, D., & Zelinsky, A. (1999, December), “Q-learning in continuous state and action spaces”, In Australasian Joint Conference on Artificial Intelligence (pp. 417-428). Springer, Berlin, Heidelberg.

AUTHORS PROFILE



Krithika Balasubramanian, currently pursuing B-Tech in Computer Science and Engineering in Vellore Institute of Technology, Vellore. Notably worked in Obstacle detection, Deep learning - Plant species Recognition, Smart Highways: Harvesting Electricity using speeding vehicles. Inclined towards upcoming fields like Data Science and thus have done technical projects related to the same.

Akhil Kothari, Final year Computer science student at Vellore Institute of Technology, Vellore. Notable projects include Question Generating System, Emotion analysis of Textual Documents, Text Summarizing Application, and Climate Model Simulation Prediction. Working on pursuing master's in data science and analytics to have a career in the same field.

K. Vijayakumar received his M.Tech. and Ph.D. Degree in 2006 and 2018 from VIT, Vellore, India. Research interests are Data Mining and Missing data. Working as Associate Professor, in Department of Information Security, at School of Computer Science and Engineering (SCOPE), in Vellore Institute of Technology, Vellore-632014.