International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

A Review on Deep Reinforcement Learning Induced Autonomous Driving Framework

Gopika Gopinath TG[1], AnithaKumari S[2]

[1]Student, Dept. of Computer Science and Engineering, LBS institute of technology for women, Kerala, India
[2]Professor Dept. of Computer Science LBS Institute of Technology for women, Kerala, India

---

Abstract - *A car that runs automatically is the perpetual goal of reinforcement learning, an Artificial intelligence paradigm. The Reinforcement Learning is being used in many control tasks. The influence of RL has a secure foundation in the area of Atari games. The planning part is done by RL and RNN were integrated for learning partially observable scheme. This paper aims to provide an outline and idea of deep reinforcement learning as it is the new area in the research field for autonomous driving and later describes the proposed framework. The framework for autonomous driving is build using Beta simulator and the training and learning part is done by the Concurrent neural network. Information integration can be done by Recurrent Neural Networks, it enables car to handle moderately apparent scenarios. The purpose of this work is to develop a deep learning framework for autonomous driving using the framework called UDACITY, an open source driving simulator which uses deep learning for the training of an autonomous driving agent. The view point is considered from behind the car. Data from sensors are fused so that information from other vehicles are aggregated including their speed and relative position. The simulator provides a 3D graphical interface which displays multi-lane roads, vehicles, obstacles etc.*

***Key Words*: Reinforcement Learning, Recurrent Neural Network, Convolutional Neural Network, Autonomous Driving**

1. INTRODUCTION

Driving any kind of vehicle is a task which requires skill and experience along with presence of mind from a human driver. Autonomous driving being done by AI is an extreme challenge. The Reinforcement Learning is used in control tasks for a long time. To achieve human level control the combination of Reinforcement Learning with Deep Learning is used as one of the most assuring approaches. This successful demonstration was already done in Atari games using the Deep Q Networks. RL and DL parts are responsible for planning part and learning part respectively. Later Recurrent Neural Network is used for information integration. The information is aggregated from camera which are high dimensional but the useful informa- tion for autonomous driving requires only lower dimensional images. Fine details of the vehicle are not important, the interest lies in the bounding box information. The relevant information is only extracted and all other non-relevant parts are neglected which in turn improves accuracy and efficiency of the autonomous driving system thus reduces the memory requirements and computation complexity which are considered to be the critical parameters of the embedded system that will deploy the autonomous driving control unit. In this paper main focus is on end to end autonomous driving model which inputs images and gives driving actions as outputs. The Reinforcement Learning framework provide a model with best policy where an agent follow a best action in a given particular state so that the total rewards value is estimated and compare it whether it is maximum or minimum and thus a terminal state is reached. The Reinforcement Learning paradigm models to learn from its own involvement by taking actions. The model learns from making mistakes. In RL this can be handled by reward signal in which it can model decision between whether to drive (action) or to plan (where to drive). It is very challenging to make it work on a real car thus most of the current RL research is done in game simulation environment. In this work the simulator is UDACITY which illustrates driving scenarios where the car will be able to navigate with sharp turns which can all be illustrated in RL.

2. REINFORCEMENT LEARNING MODEL

Autonomous driving aims for a safe and secure comfortable approach that drives an autonomous vehicle to the desired location. Before sending orientation to the vehicle control autonomous motion planners need to understand the environment where the vehicle is supposed to make its control over. The steps involving understanding the environment are state of vehicle, interaction with obstacles traffic constraints etc. A mapping from current state to the intended space where the vehicle tends to move is discovered by the motion planner. Haoyang Fan et.al [7] and Shai et.al [8] proposed their work about the approaches made to develop mapping scenario. There exists three major approaches to develop mapping system as discussed above: Learning via Demonstration or Imitation learning or through optimizing the current cost/reward function or through inverse reinforcement learning.

2.1 Imitation Learning or Learning via Demonstration

In this Imitation learning system otherwise called Learning via Demonstration, expert demonstration induced state to action mapping is learned directly. The direct application of imitation learning leads to further complication in robotic systems such as autonomous driving since imitation learning has seldom knowledge in understanding the environmental changes. Also, it only works for limited scenarios and requires large amount of dataset since the coverage of date is critical in imitation learning. Another issue in imitation learning is that it requires special issues to covariate shifting issues as the environment changes rapidly. This can be solved if and only if more demonstration data are collected from an expert. Related data collection for extensive problems are of no use. In autonomous driving applications, states are hard to produce since it involve interaction among obstacles in surrounding or constraints. When erroneous behavior occurs an intended imitation framework is found to be difficult.

2.2. Optimization through reward function

This process is done for driving actions through maximizing the reward function. The motion planning approaches derive the policies with a pre-stated cost/reward function. In this approach search procedures such as dynamic programming or direct optimization are done after space is discretize into lattice. Experts are providing the cost/reward functions for inverse reinforcement learning.

2.3 Inverse Reinforcement Learning

By comparing the expert demonstration, Inverse Reinforcement Learning learns the reward function. Expert demonstration is generated with policies that optimizes the reward function. Reward extraction can be done via feature matching. Most IRL methods are found to be expensive in terms of computation since it is required to generate policies in each iteration via reinforcement learning sampling. Majority of the IRL approaches are effective for specific tasks with adequate training data and time. But it becomes challenging when these learning based methods are applied to autonomous driving. In the conventional reinforcement learning model [9], there exists 3 key terms namely state, action and reward. State refers to the current situation, action refers to which the action that perform by agent in a a particular situation and finally reward refers to the feedback depending upon the action and it determines whether it is valuable or cheap. Depending on the value of the reward function next state is determined. Most importantly an agent is connected to the environment via action. The input, current state, action and agents behavior are all indicated using the letters i, s, a, and B respectively. The status of the surrounding is changed via the actions and the state transition value is fed to the agent through a scalar reinforcement signal say r. The actions behavioral change r chooses the action by the well planned trial and error methods. Therefore the conventional reinforcement model consist of a distinct set of states S and actions A. The agent responsibility is to find a policy which maps states into actions. In general, the expectations lies in the fact that the non-deterministic nature of the environment and the act of taking the similar action in the same state on two different instances which in turn results in next different states or different reinforcement values or together. Reinforcement learning is different from supervised learning in wide variety of ways [10]. The main difference is that there is no input/output pairs. It is seen as a trial and error approach. The agent is not directly informed about the action it is supposed to take. It can only evaluate the actions according to the rewards it received. Despite that, after choosing the action the reward is told to the agent and the corresponding state. But the information regarding the action in its long run is not revealed. It is required that the agent should gather useful information and experience regarding the system and its states and actions and its rewards. In the paper proposed by Takafumi Okuyama et.al [13] they dealt with the simulation results in autonomous driving in a simple environment with obstacles and lane markings. The learning process is done by Deep Q Network. The output of CNN in Deep Q Network is not classification which is not same in reinforcement learning. The Q values or rewards are captured by the front camera placed on the car corresponding to the actions of the vehicle. The actions provided are discrete angles where the car steers with fixed speed. Based on the rewards maximum value corresponding action is enforced. This work implemented gives high accuracy in learning by observing obstacles and the lanes. Recently DQN has proved its successful demonstration in Atari games by Google DeepMind. DQN enables the agent to learn from its behavior rather than labelled data. Abdur R. Fayjie et.al [14] proposed a work with Deep Reinforcement Learning to solve complex control and navigation related tasks. This paper mainly deals with automatic steering and obstacle avoidance in urban environment. The difference when compared with other autonomous driving framework is that it uses two sensor data as input. A camera sensor and laser sensor is placed in front of the car. It also built a real time prototype capable of running the same algorithm in low cost. It incorporates GPU (Nvidia-TX2 for running deep learning algorithms). In the network architecture a deep neural fully connected network is designed to approximate Q function. This network consist of 3 convolutional layers and 4 dense layers. Two types of data namely front camera image and lidar data are fed into the neural network. The neural network accepts 5 actions with defined reward as trained.

3. REINFORCEMENT LEARNING INDUCED SIMULATORS AND FRAMEWORKS

3.1. ALVINN (Autonomous Land Vehicle in a Neural Network)

A related work ALVINN [1] developed neural network which trained a policy to control driving by mapping from images. It is designed for the road tasks is a three layer BP(Back Propagation) network. Input layer consist of two retinas and a feedback unit. Each of the input layer has connection to 29 hidden layer units which is connected to the resultant layer. It consist of 46 units of two groups. First 45 units is the linear demonstration of the curvature of the road through which the vehicle have to travel and head towards the middle part of the road. The middle unit is the representation of the straight travel condition along with the right and left conditions The current architecture consist of only single hidden layer feedback network. In ALVINN it take image input from camera and the output is the guidance to which the vehicle travel. Training is done with simulated road images. The final output is a feedback unit which check whether the road is brighter or cloudy. In the testing stage the activation of the output unit is given to the input layer as a feedback mechanism. In order to check the effectiveness and accuracy, several tests were conducted and have succeeded in the Carnegie Mellon autonomous navigation tests vehicle. It has proved that the network has effectively followed real roads under real time conditions. The representations may differs accordingly as the network is trained under certain conditions.

3.2 TORCS (The Open Race Car Simulator)

Bernhard Wymann et.al [2] proposed an open source driving simulator TORCS which is used for autonomous driving from real generalized images. It is a modern highly portable, modular multi agent simulator for cars. TORCS simulator is an ideal platform for artificial intelligence since it is of highly modular and portable. It was used to create agents for Artificial Intelligence (AI). For autonomous driving vehicles new modules designed for simulation are included such as intelligent control systems. The API gives moderate access to the simulation state This makes it effortless to develop scratch from mid-level control systems to sophisticated driving agents which reacts successfully in unexpected situations. the components of the simulation engine includes basic features such as the rotational inertia , mass, engine, wheels along with mechanical details like links and differential, suspension types etc. Static and dynamic friction of tyres for different ground types are all checked. Since TORCS is completely modular simulation can be easily replaced. In this paper the participating players or agents which are called as robots are loaded as external modules in this framework. These agents are developed independently which satisfy only the basic API requirement of the robot code. Each robot has ability to collect and process information about the geometry and surface of tracks. The robots interact with the simulation every 0.02 sec. The API gives information regarding the racing status, position and distance of robot from the track's edges and position with respect to other cars. The robots might lack direct access to the simulation state. Robots use

3D projection instead of low-level API. The mission of this paper was to maintain a stable API in order to avoid distraction for many users.

3.3 The DARPA Autonomous Vehicle

The DARPA Automatic Vehicle DARPA project [4] is an off-road robot that drive avoiding obstacles on an open terrain form visual output. This system was trained by a human driver from hours of data training under variety of scenarios. The network is a 6 layer convolution network. In Ahmad El Sallab et al. [3] paper they introduced a robot car that drives autonomously which is the ultimate goal of Artificial Intelligence. In this paper the levels of AI task involved in Autonomous driving is divided into Recognition, prediction and planning. Recognition is used to identify surrounding environment. For example, pedestrian and traffic sign detection etc. With deep Learning algorithms human level recognition is relatively an easy task. In this framework Convolutional Neural Networks are the most successful deep learning model. The success of this demonstrations are inherited in autonomous driving , lane and vehicle identification. In Prediction the main purpose is to predict the imminent states of the environment. In order to predict the subsequent state the past information is needed to integrate. Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) are used for end to end labelling system. RNN is used to enhance object tracking performance in the Deep Tracking model recently. In planning it incorporates recognition and prediction to plan the subsequent stages of driving actions that enables the vehicle to steer. This is most complicated task of the three. The mixture Reinforcement Learning and Deep Learning is designed to be the most valid approach to achieve human level control in autonomous vehicles. Human level control is already achieved on Atari Games with the Deep Q Networks model. RL is responsible for planning part and DL is responsible for learning part. Finally RNNs are integrated for partially observable schemes.

3.4 FODS and DeepGTAV Framework

First order driving simulator (FODS) was introduced by Wesley Hsieh [5] which is an open source driving simulator designed for data collection purpose and bench marking performance for autonomous driving experiments First order driving simulator (FODS) was introduced by Wesley Hsieh [5] which is an open source driving simulator designed for data gathering purpose and bench marking performance for autonomous driving experiments Another framework called DeepGTAV [11] is an interface which communicates with an instance of Grand Theft Auto which is a popular 3D open source sandbox game with driving component. . Its communication is possible through a client-server interface in python. The environment is realistic graphics and can include other cars too. This simulator is more complex and its environment is built mainly for gaming purposes rather than for driving experiments. OpenAI Gym [12] exhibits many built-in environments. The driving environment is an open-source 2D driving simulator implemented with Box2D. The state space is top down view with the camera centered on the car. There exists many driving simulators developed by many research groups and labs of varying complexities. For evaluating inverse reinforcement learning algorithm Sadigh et al. created their own 2D driving simulator. The viewpoint is a 2D bird's eye view centered on the car. This dynamic model was specified symbolically through Theano. Marcelo Paulon J. V. [6] wrote against autonomous vehicles. One of the example portrayed was Teslas autopilot car crash in 2016. It brought up controversies in safety and reliability and liability in terms of legal functions. As these vehicles become main stream topics like security and transparency is a subject of heavy discussion. Autonomous vehicles are sophisticated systems that works on technologies such as LIDAR, GPS, Artificial Intelligence and high definition maps for navigation and collision avoidance. This refers to the fact that each autonomous vehicle is collecting and analyzing large amount of data which is about 2.6 terabytes per hour as per Intel. If this data is used for exchange between vehicles then it is possible for coordinating movement which in turn helps reduction of crashes by increasing safety thereby optimize traffic problems on cities and highways.

4. OBSTACLE AND LANE DETECTION USING DEEP LEARNING

In self-driving vehicles, on road obstacle identification and classification is the key task in prediction and perception system since object localization and classification of vehicles and lane detection is required for efficient and smooth execution of autonomous vehicles. In the paper done by Gowdham Prabhakar et.al [15] proposed a deep learning scenario for the identification and classification of on road vehicles and obstacles like vehicles, pedestrians and other static dynamic objects using a region based CNN trained with PASCAL VOC image dataset. This system was implemented on a Titan X GPU. Frame rate is considered to be 10fps for a VGA resolution image frame. Because of the powerful GPU with high frame rate, the system was best suited for autonomous cars in high way driving. Faster Region-based CNN (abbreviated as R-CNN), a variant of CNN is used to address detection and classification of on-road objects. The training was done in a pre-trained network model ZF Net, which was trained for 20 different objects of PASCAL VOC dataset for the detection and classification system. In the real-time detection phase, the system filtered the object detection in such a way that the entire system was made to detect only the classes corresponding to the on road objects. The output generated was a rectangular bounding boxes and classification data of objects were saved since it is the useful parameter for motion planning of self-driving vehicle. Inwook Shimet et.al [17], proposed an obstacle detection module on road using Lidar sensors assuming that the road is flat. The obstacles which are not a direct distraction for driving vehicles such as hills and speed bumbs are distinguished from real obstacles. For faster processing 3D point clouds are projected to a 2D grid other than processing thousands of points from eight scan lines. Probabilistic tem- plate matching method was used to detect traffic lights. Prior information such as scale and position of lights were used to check the state of traffic lights showed a good result. For several illumination conditions Hue Saturation Value domain image was used for getting robust results. Zheng-Hao Chong et.al [16], used Scale Invariant Feature Transform (SIFT) was used to scan every traffic signs and are preprocessed to extract the key points. These key points were saved in the database for future reference and comparisons. At each0.01 seconds a particular image is captured by the camera and are processed with SIFT to extract the key points which are then compare with the saved key points. Lane change is considered to be the crucial vehicle maneuver which requires coordination among surrounding vehicles. In [17] road lane detection modules provides information to extract feature map of road markers. Denoising is done by dilation, erosion and blob labelling which is calculated by analyzing the width of each blob. If the width of blob is greater than width threshold value then the blob is not considered as the road lane. In the work done by Pin Wang et.al [21], formulated a reinforcement learning oriented methodology for lane change in an intelligent way by making the vehicular agent to grasp lane change under various unseen and diverse scenario. The driving environment involves several other vehicles and obstacles in which their behavior may change according to the environmental conditions. For instance when a vehicle is supposed to change its lane, it may indicate it by turning on the turning signal and the lag vehicle in the intended lane may cooperatively change its path or may accelerate just to hinder the vehicle's action of motion. In order to get a model free access and to find excellent policy, an RL based approach was adopted since it in not trivial to model the environment with all possible future situations. MPC based controllers were used for explicit sensor inputs. A new method was proposed by Junqing Wei et.al [18] to enhance the level of perception and quality of driving in autonomous vehicles. Here after clearing the path with no obstacles a reference planning layer generates dynamically feasible paths. After that a behavioral planning layer considers fixed and mobile obstacles into account. It also takes the social cooperation between the autonomous vehicles and the surrounding vehicles. Due to the real time scenarios, it is unable for the motion planner to consider the effects of vehicle controllers with imperfection. In [18], a framework for behavioral planning is proposed with the combination of parallel and hierarchical architecture in order to accomplish high level intelligence in fully autonomous driving. Behavioral planning focuses on road traffic including static and dynamic obstacles and all road blockages as input. The output generated are controller directives including the desired vehicle, driving bias, aggressiveness of distance keeping and maximum speed. Prediction and Cost-function Based (PCB) algorithm was used to implement behavioral planning. Sonke Eilers [19] explained a framework for path planning for autonomous planning in static environment. This can be achieved by decoupling search based path planning and expansion strategies. In the architecture of autonomous vehicle for planning framework, the overall system is divided into 3 sections namely Sense, Think and Act. The localization component is connected to the think section every section of the think segment needs information of the position of the vehicle. Sense segment is used for generating representation of the environment and to localize the vehicle. The act section consist of actuators like servos for steering acceleration and braking. Jesse Levinson et.al [20] proposed trajectory planning towards handling realistic road traffic where other dynamic traffic participants are considered. This situations include driving scenarios like merging and passing with traffic flow, changing lanes or avoiding vehicles. The trajectory planning works in these situations on the planning and execution.

5. CONCLUSION

This paper contributes to the survey of recent advances in deep reinforcement learning and introduces framework for end to end autonomous driving using deep reinforcement learning. It discusses about the various frameworks and simulators for autonomous driving along with the com- parison and differences made by various work based on the reinforcement learning induced simulators, games and autonomous driving. This paper also showed that Deep Q-Networks can be an effective means of controlling a vehicle from high dimensional sensory inputs. Reinforcement learning approach for lane change behavior, obstacle and traffic detection and behavior planning are also discussed. In addition, review of overall system with architecture and sensor network including modules for obstacle detection and information collection are also included.

REFERENCES

[1] Dean A. Pomerleau, ALVINN: An Autonomous Land Vehicle in a Neural Network, Pomerleau Carnegie Mellon University Pittsburgh, 2015.

[2] Bernhard Wymann Christos Dimitrakakisy, Andrew Sumnery, TORCS: The open racing car simulator, March. 2015

[3] Ahmad El Sallab1, Mohammed Abdou1, Etienne Perot2 and Senthil Yogamani, Deep Reinforcement Learning framework for Autonomous Driving, arXiv: 1704.02532v1 [stat.ML] 8 Apr 201

[4] Lu Chi, and Yadong Mu, Member, IEEE, Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues, arXiv: 1708.03798v1 [cs.CV] 12 Aug 2017

[5] Wesley Hsieh, Electrical Engineering and Computer Sciences University of California at Berkeley, First Order Driving Simulator, Technical Report No. UCB/EECS-2017-102 http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-102.html May 12, 2017.

[6] Marcelo Paulon J. V. AVCP: Autonomous Vehicle Coordination Pro- tocol, Pontifcia Universidade Cat olica do Rio de Janeiro (PUC-Rio) Advisor: Markus Endler, December 2017.

[7] Haoyang Fan1, Zhongpu Xia2, Changchun Liu2, Yaqin Chen2 and Qi Kong ,An Auto tuning Framework for Autonomous Vehicles , arXiv:1808.04913v1 [cs.RO] 14 Aug 2018.

[8] Shai Shalev-Shwartz Shaked Shammah Amnon Shashua, Safe, Multi- Agent, Reinforcement Learning for Autonomous Driving, arXiv: 1610.03295v1 [cs.AI] 11 Oct 2016.

[9] Leslie Pack Kaelbling, Michael L. Littman, eComputer Science Department, Box 1910, Brown University Providence, USA Reinforcement Learning: A Survey.

[10] Manon Legrand,Deep Reinforcement Learning for Autonomous Vehicle Control among Human Drivers, Faculty of Sciences Department of Computer Science 2016.

[11] Conrado Mateu Gisbert, Novel synthetic environment to design and validate future onboard interfaces for self-driving vehicles, Project in Computer Communication ICT October 2018.

[12] Robert Chuchro, Deepak GuptaGame, Playing with Deep Q-Learning using OpenAI Gym, cs231n.stanford.edu/reports/2017/.

[13] Takafumi Okuyama, Tad Gonsalves, Jaychand Upadhay,Autonomous Driving System based on Deep Q Learnig, International Conference on Intelligent Autonomous Systems 2018.

[14] Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee,Driverless Car: Autonomous Driving Using Deep Reinforcement Learning In Urban Environment, 2018 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawai'i, USA, June 27-30, 2018.

[15] Gowdham Prabhakar, Binsu Kailath, Sudha Natarajan, Rajesh Kumar, Obstacle Detection and Classification using Deep Learning for Tracking in High-Speed Autonomous Driving, 2017 IEEE.

[16] Zheng-Hao Chong, Ling Yee, Albert Causo1, I-Ming Chen, Autonomous Robot Driving Decision Strategy Following Road Signs And Traffic Rules: Simulation Validation , 2016 16th International Conference on Control, Automation and Systems (ICCAS 2016) Oct. 16-19, 2016 in HICO, Gyeongju, Korea.

[17] Inwook Shim , An Autonomous Driving System for Unknown Environments Using a Unified Map, IEEE, Jongwon Choi, Student Member, IEEE, IEEE TRANSACTIONS ON INTELLIGENT TRANS-PORTATION SYSTEMS, VOL. 16, NO. 4, AUGUST 2015

[18] J. Junqing Wei, Jarrod M. Snider, Tianyu GU, John M. Dolan and Bakhtiar Litkouhi, A Behavioral Planning Framework for Autonomous Driving 2014 IEEE Intelligent Vehicles Symposium (IV) June 8-11,2014. Dearborn, Michigan, USA.

[19] Sonke Eilers2, Jurgen Boger1 and Martin Franzle1 A Path Planning Framework for Autonomous Vehicles , Proceedings of the 9th Inter- national Workshop on Robot Motion and Control, Wasowo Palace, Wasowo, Poland, July 3-5, 2013.

[20] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun Towards Fully Autonomous Driving: Systems and Algorithms, IEEE Intelligent Vehicles Symposium (IV) Baden-Baden, Germany, June 5-9, 2011.

[21] Pin Wang Ching- Yao Chan Arnaud de La Fortelle, S Reinforcement learning Based Approach for Automated Lane Change Maneuvers, 2018 IEEE Intelligent Vehicles Symposium (IV)