

Deep Reinforcement Learning Framework for Navigation in Autonomous Driving

Gopika Gopinath T G

M.Tech Scholar, Computer Science and Engineering
LBS Institute of Technology for Women
Trivandrum, India

Anitha Kumari S

Associate Professor, Computer Science and Engineering
LBS Institute of Technology for Women
Trivandrum, India

Abstract— Reinforcement Learning resides in the scope of Machine Learning which allows software agents and machine to automatically manipulate the behavior of a specific environment and perform accordingly. The success of Atari games has proved the influence of RL in gaming environments. From the inspiration of the success in RL induced games, the idea of a car that runs automatically has become the goal of Reinforcement Learning. The purpose of this paper is to provide idea of how Reinforcement Learning can be implemented for the purpose of navigation in autonomous car in game environment that is the proposed work here. The classification of the images which are the primary dataset is done by Convolutional Neural Network (CNN). The purpose of this work is to implement navigation in autonomous car using MXNet, an open source reinforcement learning framework which is primarily used to train and deploy deep neural networks. In assistance with the Beta simulator made by the open source driving simulator called UDACITY is used for the training of the autonomous vehicle agent in the simulator environment. The agent here is a car that navigates without driver intervention and proceeds to move autonomously by learning thoroughly about the environment surroundings

Keywords— Beta Simulator, Convolutional Neural Network, MXNet, Reinforcement Learning, Udacity Framework
Introduction

I. INTRODUCTION

Driving a vehicle requires skill, expertise and presence of mind from a human driver. The driving scenario is a complicated challenge when it comes to incorporate Artificial Intelligence in automatic driving schemes. In order to bring human level talent for machine to drive vehicle, then the combination of Reinforcement Learning (RL) and Deep Learning (DL) is considered as the best approach. This combination has already proved the success in Atari games. RL is responsible for planning part where as DL is responsible for learning part. The information needed for manipulation is collected form images of high dimension but the information needed for autonomous driving requires only low dimension images. The relevant information is only extracted and all other non-relevant parts are neglected. The accuracy and efficiency of the system is thus improved. The main parameters like memory requirements and computational complexities are also reduced. This work portrays an end to end autonomous driving model that takes images as input and outputs driving actions. The RL model works in such a way that it learns from making mistakes i.e. it learn from its own involvement by taking actions. This particular scenario is handled by reward signal or in other

words based on these reward signal the driving agent can take appropriate decision like whether to move (action) or where to drive (plan). This is difficult to implement on a real car as it requires time and huge cost thus this current RL research is done in game simulation environment. Beta simulator is used to illustrate driving scenarios in this work. The car is able to navigate with sharp turns and it adjusts the speed in the curves and humps which is illustrated by RL. This enacts how self-driving is implemented using Behavioral Cloning. The whole process include Convolutional Neural Network, for feature extraction and continuous regression for getting steering angle.

II. RELATED WORK

Artificial Intelligence based autonomous vehicles have already explored the existence in different levels of autonomous driving scenario. Following are the related projects in autonomous driving done by various simulators.

A. ALVINN (Autonomous Land Vehicle in a Neural Network)
ALVINN [1] was developed by neural network which control driving by the use of images. Input layer consists of two retinas and feedback unit in which every layer in the input is connected to 29 hidden layer unit. It is divided into 2 groups each with 46 units such that first 45 units demonstrates the curvature of the path through which the vehicle navigate. Input image is taken from the camera and the output is the guidance for the vehicle to travel. Simulated road images are taken and thus training is done. The final output is a feedback unit which check whether the road is good to travel i.e. it check whether the condition of the road is cloudy or bright. Several tests were conducted in order to check the accuracy and effectiveness and have succeeded in the Carnegie Mellon autonomous navigation tests vehicle. It is technically proved that the network have followed real on road conditions.

B. TORCS (The Open Race Car Simulator)
TORCS [2] is an open source driving simulator proposed by Bernhard Wymann et.al and is used for autonomous driving from real generalized images. It is a modular multi-agent simulator for cars and is highly portable. In this simulator the agent are loaded as external modules in the framework. This agents are developed independently which satisfies the basic API requirements of the robot code. Robots are designed in such a way that it has the ability to collect and process the information about the geometry and surface of the path. The

information regarding the racing status, distance and position of robot from the edge of the track and position with respect to other cars are given by the API. The purpose of this work was to provide an API which is stable enough to avoid distraction for many other users.

C. The DARPA Autonomous Vehicle

This project is based on off-road robot that proceeds to move by avoiding obstacles on a terrain from visual output [4]. Human driver trains the system under real constraints. The network is built with 6 convolutional network. Ahmad El Sallab et.al introduced a robot car with AI having tasks like recognition, prediction and planning [3]. Recognition is done so that surrounding environment can be identified such as pedestrian, traffic sign detection etc. prediction is used to predict the states. Past information is needed to predict the subsequent states. Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) are used to end to end labelling process, in planning part recognition and prediction parts are incorporated to plan the subsequent stages of driving action that enables the steering of the vehicle. In order to achieve human level control in the autonomous vehicle, combination of RL and DL is designed.

D. FODS & DeepGTAV Framework

Wesley Hsieh introduced an open source simulator called First Order Driving Simulator (FODS) [5]. It is designed for data collection purpose and bench marking performance for automatic driving experience. DeepGTAV [11] is another framework which communicates with Grand Theft Auto instance which is a popular 3D open source sandbox game with driving component. Client-server interface in python is used for communication. Realistic graphics is used for environment and can include other cars too. The environment is built mainly for gaming purpose rather than real time driving experiments.

III. REINFORCEMENT LEARNING

In autonomous driving the vehicle should reach the destination safely. In order to achieve that goal motion planners of the vehicles must understand the environment. Understanding the environment means understanding the state of vehicle, interaction with obstacles, traffic signals etc. The mapping from current space to intended region where the vehicle is supposed to move is done by the motion planner. The mapping scenarios and its regarding approaches [7] [8] were discussed in the work done by *Haoyang Fan et.al and Shai et.al*. There exists several approaches for mapping systems. Reinforcement learning is one among the approach which is done via reward function. By maximising the value of reward function driving actions can be done. The pre-stated cost/reward functions with policies are derived by the motion planners.

Following are the steps in RL using reward function:

- Input: It is the initial state of the model in which it is about to start
- Output: It is the solution of the problem
- Training: This is based on the input as the model returns the state and user will decide to reward the model based on its output or not.
- The model continues to learn.

- On the basis of the value of the reward best solution is made provided the value of the reward should be maximum.

IV. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks also known as CNNs, also known as ConvNets is one of the main stream in neural networks. CNNs are widely used in the areas such as Image recognition and classification, object detection and face recognition. CNNs takes input as images but the computer recognizes it as array of pixels and process it and classify as different categories. HxWxD is the image resolution which is Height, Width and Dimension respectively. Each input image that is taken at the time of training mode is fed to the CNN and it will pass through a stream of layers in CNN with kernel (Filters), pooling layers and fully connected layers. Classification techniques such as Softmax Function is used to classify an object.

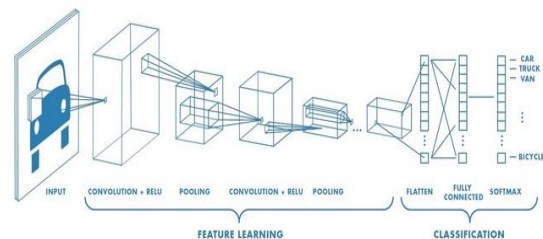


Fig 1: Neural network with convolutional layers

Following are the brief description of the steps in CNN:

- Feed input image to the convolutional layer.
- Determine the parameters needed and apply filters with strides also apply padding if needed.
- Convolution is applied to the image with ReLU activation to the matrix.
- Pooling is done to reduce dimensionality size.
- Convolutional layers are added until perfection is obtained
- Feed the output into fully connected (FC) layer after the output is flattened.
- Classify images and output the class using activation function.

The convolutional layer extracts the features when the input image is fed to it. This layers helps to learn about the features of the image and keeps the relationship between the pixels by using image matrix and kernel (filter). When the filter does not fit the image perfectly the alternate option is either do padding, i.e.; zeroes are padded to the picture thus it fits to the filter perfectly. This is the perfect way of padding and it is called valid padding. In order to introduce non-linearity in our ConvNets Rectified Linear Unit (ReLU) is used and thus CNN would only learn non-negative linear values. Another layer present in CNN is the pooling layer that is used to reduce the number of parameters when the image is large. In other words this layer reduces the dimensionality of each map but preserves the important features and information. Finally the Fully Connected layer called as FC layer is used to flatten the matrix into vector.

V. CONVOLUTIONAL NEURAL NETWORK

A. Data Collection

The training data is collected by driving the car in the training track inside beta simulator built using Udacity. The car in the simulator is driven using the keyboard keys. Depending on the driving that is done using the keyboard keys, it is copied to autonomous mode. In the training track as the car proceeds to move images are taken at each instance using the virtual camera built on left, right and center of the car. These image data are recorded using the record button. Later the images are saved in a specified folder. The images represents the training dataset.

B. Training Process

The label of the images taken are considered to be the steering angle of the vehicle at particular instance. The training images are then fed to convolutional neural network in order to allow it to learn how to navigate the car autonomously same as that of the behavior of manual driver. The most important variable is the steering angle that learns to adjust the car at any given instance and thus eventually learn to adjust appropriate degree based on the circumstances that it find on any particular instance. By the behavior of the user the car learns to drive and navigate autonomously thus the name Behavioral Cloning and it is the technique that plays a vital role in real self-driving too.

VI. PROPOSED WORK

The model includes RELU layers to introduce non linearity and the data is normalized. The model is trained and validated on different data sets. The model is tested by running via the simulator and ensure that the vehicle can stay and move on the track perfectly. The design of the car includes 3 virtual cameras which is used to take input images for training, along with that steering wheel angle is recorded and stored at the time of manual training mode and are considered as the desired steering command. Images are then fed into the CNN. The proposed steering command is calculated form the output of the CNN. This steering command is then compared with the desired steering command and the error is calculated and weights are adjusted via back propagation so that the output received finally will be closer to the desired output. Finally it generates the steering command for perfect driving. After that it is supposed to run the car in autonomous mode of the simulator and the car will start driving.

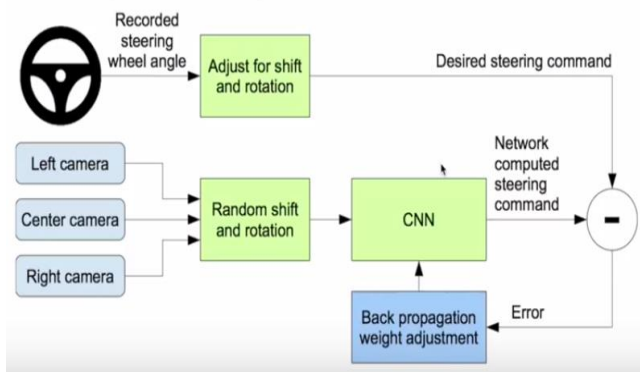


Fig 2: Design of the Architecture

A. Data Collection

After the training of the autonomous model, for testing the simulator is considered to be as the client-server model. The server is the simulator itself and the client is the python program. This client server model is considered as the feedback loop i.e. client is piping in the steering angle and throttle to the server and the server is piping back the images from the car and steering angle so that it can train it right.

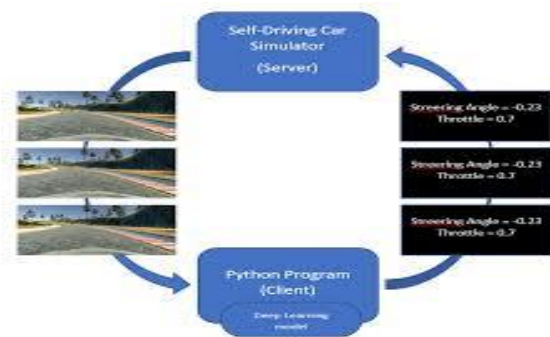


Fig 3: Feedback Loop by Client Server model

VII. RESULT

The navigation of autonomous car in beta simulator made by Udacity has successfully ran on the track by adjusting speed and acceleration when detecting curves and humps. The training is done with the images taken in the training mode and it is fed to the Convolutional Neural Network for classifying the training data set consisting of the images taken by the virtual camera built on the car. Reinforcement learning makes the vehicle to learn about the environment where the vehicle is supposed to navigate and thus it understand to move on the track without failure.

VIII. CONCLUSION

This paper describes the implementation of navigation in autonomous car with the help of Deep Reinforcement Learning framework, Convolutional Neural Network and the driving environment called Beta Simulator made by Udacity. The training approach for the entire process along with operation on convolutional neural network is also discussed. A survey on recent advances in deep reinforcement learning and also framework for end to end autonomous driving using this technology is discussed in this paper. Along with different frameworks, a comparison and differences between the autonomous driving simulators induced by reinforcement learning are also discussed.

REFERENCES

- [1] Dean A.Pomerleau, ALVINN: An Autonomous Land Vehicle in a Neural Network, Pomerleau Carnegie Mellon University Pittsburgh, 2015.
- [2] Ahmed, M. S., Mohammed, A. S., & Agusiobo, O. B. (2006). Development of a Single Phase Automatic Change Over Switch. AU Journal of Technical Report, 10(1), 68-74.
- [3] Ahmad El Sallab1, Mohammed Abdou1, Etienne Perot, Senthil Yogamani, Deep Reinforcement Learning Framework Autonomous Driving, 8 April2017.
- [4] Lu Chi, and Yadong Mu, Member, IEEE, Deep Steering: Learning End-to-End Driving Model from spatial and Temporal Visual Cues, Aug 2017

- [5] Wesley Hsieh Electrical Engineering and Computer Sciences University Of California at Berkley, ley, First Order Driving Simulator technical Report May 2017
- [6] Marcelo J.V.AVCP: Autonomous Vehicle Coordination Protocol, December 2017 M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [7] Shai Shalev-Shwartz Shaked Shammah Amnon Shashua Safe, Multi-Agent Reinforcement Learning for Autonomous Driving, Oct, 2016
- [8] Haoyang Fan¹, Zhongpu Xia², Changchun Liu², Yaqin Chen² and Qi Kong. An Auto tuning framework for Autonomous Vehicles, Aug 2014
- [9] Manon Legrand, Deep Reinforcement Learning for Autonomous Vehicle among Human Drive Faculty of Science Dept, of Science
- [10] Leslie Pack Kaelbling, Michael L. Littman, eComputer Science Dept. Box 1910, Brown University Providence, USA Reinforcement Learning: A Survey
- [11] Conrado Mateu Gisbert, Novel synthetic environment to Design validate future onboard interfaces for Self-driving Vehicles, Project in CS ICT OCT 2017.
- [12] Robert Chuchro, Deepak GuptaGame, Playing with Deep Q- Learning Q using OpenAIGym cs23 In.stanford.edu/2017