# SELF DRIVING CAR USING DEEP Q-LEARNING

**Akhilesh Thete[1], Chinmay Toley[2], Shreyas Inamdar[3]**

*[1,2,3]NBN School of Engineering & Technology, Pune*

---

**Abstract:** *In this paper, a self-driving car prototype using Deep Neural Network on Simulator is proposed. Self-driving cars are one of the most increasing interests in recent years as the definitely developing relevant hardware and software technologies toward fully autonomous driving capability with no human intervention. Level-� autonomous vehicles are potentially turning into a reality in the near future. Deep Neural Networks (DNNs) have been shown to achieve significant Performance in various perception and control tasks in comparison to other techniques in the latest years. The key factors behind these impressive results are their ability to learn millions of parameters using a large amount of labeled data. Deep Q-Networks (DQN), a reinforcement Learning algorithm that achieved human-level performance across various fields*

## Introduction

Self-driving cars have been one of the most promising prospects for Artificial Intelligence research, it would be the greatest technological revolution of the near future. There are several technologies making up an autonomous vehicle - laser, radar, GPS, image processing computer vision, machine vision, etc. Compared with other sensors like LIDAR or ultrasonic, cameras are lower-cost and can provide more information on the road (traffic signs, traffic lights, pedestrians, obstacles, etc. Reinforcement learning (RL) agents incrementally update their parameters (value function or model) while they observe a stream experience. In their simplest form, they discard incoming data immediately, after a single update

## Literature Review

2.1 Prioritized experience replay", Tom Schaul, John Quan, Ioannis Antonoglou and David Silver "This paper demonstrates the effectiveness of navigating autonomous vehicles using reinforcement learning methods. We have shown that Deep Q-Networks can be an effective means of controlling a vehicle directly from high-dimensional sensory inputs, and we used a novel combination of CNN and RNN networks to achieve this.

2.2 "Real time self-driving car using Deep Neural Network", Truong-Dong Do, Minh-Thien Duong, Quoc-Vu Dang and My-Ha Le - The results obtained from the experiments look promising which suggests that alancing the Exploration/exploitation ratio based on value differences needs to be further investigated.

2.3 Open CV based autonomous RC Car", B,Sabith, K.Akila, S,Krishna Kumar, D.Mohan.– "The evidence from neuroscience suggest that a prioritization based on episodic return rather than expected learning progress may be useful too Atherton et al. (2015); Olafsd ́ ottir et al. (2015); Foster & Wilson (2006).

2.4 "Formulation of Deep Reinforcement Learning Architecture Toward Autonomous Driving for On-Ramp Merge", Pin Wang, Ching-Yao Chan In this work, they propose a Deep Reinforcement Learning architecture for learning an on- ramp merge driving policy.

**Analysis Model: SDLC Model to be applied**

The Waterfall Model is sequential design process, often used in Software development processes, where progress is seen as flowing steadily download through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. This one of the oldest models followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, construction and deployment of the software.

1. Communication

In communication phase the major task performed is requirement gathering which helps in finding out exact need of customer. Once all the needs of the customer are gathered the next step is planning.

2. Planning

In planning major activities like planning for schedule, keeping tracks on the processes and the estimation related to the project are done.

Planning is even used to find the types of risks involved throughout the projects. Planning describes how technical tasks are going to take place and what resources are needed and how to use them.

3. Modeling

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed.

Different models for developing software is created depending on the requirements gathered in the first phase and the planning done in the second phase.

4. Construction

The actual coding of the software is done in this phase. This coding is Done on the basis of the model designed in the modeling phase. So in This phase software is actually developed and tested.

5. Development

In this last phase the product is actually rolled out or delivered installed at customers end and support is given if required. A feedback is taken from the customer to ensure the quality of the product. From the last two decades Waterfall model has come under lot of criticism due to its efficiency issues. So let's discuss the advantages and disadvantages of waterfall model.

**Implementation Planning**

**Purpose**

Autonomous cars create and maintain a map of their surroundings based on a variety of sensors situated in different parts of the vehicle. Radar sensors monitor the position of nearby vehicles. Video cameras detect traffic lights, read road signs, track other vehicles, and look for pedestrians.

3.5.2 Domain Area of Project

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction.

3.5.3 Feasibility Study

This project studied the feasibility of constructing an autonomous vehicle controller based on probabilistic inference and utility maximization. Several theoretical and algorithmic advances were required in order to create an inference system capable of handling vehicle monitoring in a real-time fashion. New methods were also developed for learning probabilistic models from data, and for learning control policies given reward/penalty feedback

3.5.4 Risk Management

An Unregulated Industry. More Accidents Blending Self-Driving and Manual Cars Vulnerability to Hacking & Remote Control. Computer Malfunctions.

**Deep Q Learning:**

With Deep-Q Learning we can program AI agents that can operate in environments with discrete actions spaces. A discrete action space refers to actions that are well-defined, e.g. moving left or right, up or down. Atari's Breakthrough is a typical example of an environment with a discrete action space. The AI agent can move either left or right. The movement in each direction is happening with a certain velocity. If the agent could determine the velocity, then we would have a continuous action space with an infinite amount of possible actions (movement with a different velocity). This case will be considered in the future.

**Conclusion**

This project has demonstrated the effectiveness of navigating autonomous vehicles using reinforcement learning methods. We have shown that Deep Q-Networks can be an effective means of controlling a vehicle directly from high-dimensional sensory inputs, and we used a novel combination of CNN and RNN networks to achieve this. While currently it seems as though a well-designed, low-dimensional discrete state-space agent is able to more stably control a car compared to a more complex DQN agent, we believe our work could be extended in several ways. Namely, it would be nice to find a better alternative of defining our reward function that still maintains the careful balance between maximizing speed while guaranteeing car stability. Similarly, it would be interesting to generalize our work to continuous action spaces. Despite these limitations, we are still proud that our agents were able to successfully control a car without any explicit notion of the car's underlying dynamics.

**REFERENCES**

[1] Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097- 1105.
[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in International conference on machine learning, 2018, pp. 647-655.
[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.
[4] N. Otterness, M. Yang, S. Rust, E. Park, J. H. Anderson, F. D. Smith, A. Berg, and S. Wang, "An evaluation of the NVIDIA TX1 for supporting real-time computer-vision workloads," in Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE, 2017, pp. 353-364.
[5] https://en.wikipedia.org/wiki/Deep_learning