Self-Driving Car

Tasnim Sakib Apon

Email: sakibapon7@gmail.com

Github

I. The system we want to model:

We want to model an autonomous vehicle driving simulation. The idea is to simulate an environment and drive a vehicle autonomously with maximum safety. The challenge is to mimic the driving behavior of a human on the simulator with the help of a model trained by machine learning algorithms. Our project is based on an open source pre-built simulator. Here we try to clone the behaviour of a car. It has three python files. Mode.py, Drive.py, Utils.py. The utils.py is totally pre built.We do not have to do anything on it. We had to work on the model file and on the drive file. Here we will have to build a model in the model file,reshape the images and do other necessary things. In the drive file we set our logics.

ii. What do you want to simulate using this model?

We want to simulate an autonomous vehicle. We will use unity as our simulator. With unity, we will manually drive the car within its training mode and in this mode, there will be three cameras to shoot the surrounding environment. After this step, we will process those data of images. Finally, based on these training data and by applying our algorithms we will be able to drive the vehicle autonomously with maximum safety.

iii. What questions will this simulation answer about the system?

This simulation will answer the following sorts of questions about the system:

- How will an agent within the system read the data from its surrounding environment?
- What are the variables, which will be the reacting factor of this system?
- How will image translation be key to generate real-world images from virtual images?
- How much efficiency will the output show in accordance with the input within the system?
- Last but not the least, whether this simulation will be considered as a progressive model or not to establish the necessity of autonomous driving?

Iv. Why is this simulation important?

The importance of this simulation:

- An autonomous vehicle simulation is important because it can be put into realistically risky driving situations to see how it reacts.
- Comparisons can be made across facilities, time, date, illness, injury, or weather conditions. Simulators provide an excellent window into perceptual and cognitive functioning, excellent at tactical control.

- This simulation is important to support the ground on autonomous driving and its key reason for maximum safety. If our simulation is assumed to be successful then it would be great for the above ground.
- The purpose of a Self-driving car project is to build a better autonomous driver. The car should be able to drive itself without falling off the track, with accelerating and braking at all appropriate places.
- Finally, this simulation will cost less than on-road studies.

Part 2

1. Project Task: Model formulation:

a. Problem Analysis: Write down the questions you asked to understand the system?

- How to decide when to turn, accelerate, or use breaks?
- How can the model be efficient for simulation?
- What are the things that we take as input from the surrounding environment?
- How do those three camera sensors help the system to generate efficient output?
- When and how much autonomy does an agent keep while driving the car?

c. Formulate the abstract model by answering the following questions.

i. What are the conditions of the experimental frame of this system?

- Surrounding Environments status & noticing it's changes
- The steering angle turns always right whenever the agent is trying to go forward and to overtake.
- Always preload the destination address in this system, and keep it variable for any last moment changes.
- Through neural optimizer, the CPU needs to select prior deciding factors to overcome the sudden danger.
- Vehicle's speed at any given time.
- Road turns.

ii. Define the system state variables, units, and events.

- Steering Left
- Steering Right
- Acceleration
- Break

<u>Events</u>: At any given time our agent captures three photos from its three sensors and it should be continuous shot. So, The system processes all incoming data without any disruption and makes the best decision and runs most efficiently throughout the whole time. That's why the events of this system are continuous and stochastic.

iii. Explain the relationships between these system state variables and explain how the variables change when events occur.

- Basically, when we talk about a roadway transport system from the perspective of a driver, there's always a steering wheel. A steering wheel is an essential tool of a vehicle, by turning it right or left we drive a vehicle on a road.

- Additionally, accelerations and brakes relations between each other are like inverse equations. Whenever a vehicle increases or decreases its speed acceleration is the reason behind its increase of speed while the brake is the reason behind its decrease in speed.

- We knew that our agent captured three photos of its surrounding environments at any given time. By analyzing the photos through CPU and GPU, the agent will get to know many deciding factors of its system variable and will proceed autonomously in order to drive the vehicle.

iv. Using the above answers define the Mathematical Model for this system.

<u>The mathematical model we have seen for this system is:</u>

autonomy=(1 - ((number of interventions) * 6 seconds/
elapsed time [seconds]))*100.

- By the above mathematical equation, we can simply measure the percentage of autonomy having by the network agent while driving the car.

- Here, the intervention occurs every time the car takes a change in its steering angle based on the path of the centerline of a certain lane more than one meter.

- In addition, we take 6 seconds as by the statistics and report from their experimental data, one needs 6 seconds to change its steering wheel and again back to its previous mid label path of a lane and then restart the steering autonomy mode.

- Finally, we can state that the mathematical equation stands highly efficient as it is determined by counting human interventions.

v. What percentage of your model can you implement? (Which parts of your model can you implement) Justify your expectation.

In the beginning, we were more interested in simulating our idea through Carla where we could implement multiple models. In Carla, we have to deal with every single aspect of a self-driving car starting from path planning, longitude control, lateral control, pedestrian movements, surrounding vehicles movement, environmental change. However, after doing several days of research, we found out that to train models on Carla we need a better computer with a GPU. As I am in my hometown I do not have access to better hardware. Therefore, it takes a huge amount of time to run Carla and in the end, we were forced to stop using Carla. Also as we did not have much time in hand, we decided to go on with the unity of TORCS. We have chosen unity. Here the path planning was not possible. Also obeying the traffic law was not possible in unity as the simulator does not have any traffic system.
In our proposed model we try to run an autonomous car using convolutional neural networks based on the photos it generates through its camera.

vi. What percentage of the simulation can be done using your expected model implementation?

We believe we can implement our model and run it on the simulator. We have generated our training data for the simulation. We are already on our way to simulate our model. Currently, we are working on a code to implement our convolutional neural networks that take the training data, and based on the training data it decides when to turn, accelerate, or use breaks. However, we are still not sure about our data validation part as it will take a long time and I do not have fast hardware that can catalyze the process. Still, we are very dedicated to finishing the simulation with data validation results.

vii. The papers you read to formulate the model of your system should be referred to within the document.

- Autonomous Driving in Reality with Reinforcement Learning and Image Translation Nayun Xu Bowen Tan Bingyu Kong Shanghai Jiao Tong University.

- DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving Chenyi Chen Ari Seff Alain Kornhauser Jianxiong Xiao Princeton University.
- End to End Learning for Self-Driving Cars Mariusz Bojarski NVIDIA Corporation Holmdel, NJ 07735 Davide Del Testa NVIDIA Corporation Holmdel, NJ 07735 Daniel Dworakowski NVIDIA Corporation Holmdel, NJ 07735 Bernhard Firner NVIDIA Corporation Holmdel, NJ 07735 Beat Flepp NVIDIA Corporation Holmdel, NJ 07735 Prasoon Goyal NVIDIA Corporation Holmdel, NJ 07735 Lawrence D. Jackel NVIDIA Corporation Holmdel, NJ 07735 Mathew Monfort NVIDIA Corporation Holmdel, NJ 07735 Urs Muller NVIDIA Corporation Holmdel, NJ 07735 Jiakai Zhang NVIDIA Corporation Holmdel, NJ 07735 Xin Zhang NVIDIA Corporation Holmdel, NJ 07735 Jake Zhao NVIDIA Corporation Holmdel, NJ 07735 Karol Zieba NVIDIA Corporation Holmdel, NJ 07735.

- Learning a Driving Simulator Eder Santana ∗ University of Florida eder@comma.ai George Hotz comma.ai george@comma.ai.

- CARLA: An Open Urban Driving Simulator Alexey Dosovitskiy1 , German Ros2,3, Felipe Codevilla1,3, Antonio Lopez ´ 3 , and Vladlen Koltun1 1 Intel Labs 2Toyota Research Institute 3Computer Vision Center, Barcelona.

Part 3

1. Project Task: Implementation of the Model

a. Which simulation techniques (names of numerical methods, modern techniques) Will you need to simulate the system?

Our problem has turned out to be a supervised learning problem as we already know the states at any given time and we need to find out the states in future.

- We will be using deep learning(convolutional neural network/CNN) to solve our problem.
- The model type will be Sequential.
- We will be using ELU/RELU as it takes care of the vanishing gradient descent.
- We will also be using Dropout too.
- We will use Flatten followed by Dense because we will be feeding fully connected layers.

b. Describe the organization of your program.

Description of our program organization:

- First, we drive our agent to the udacity's training mode
- By using three cameras on the front of our agent,it will generate three photos at any given time and later will be added to a CSV file.
- Then, we push the data set to our python client.
- After that, through CNN we write a code to train our model in order to drive our agent on track 1.
- Then, we store the data we get from our track 1 of our agent's autonomous driving.
- Now, we compare our prediction with the actual value we get and then find the accuracy of our techniques.

c. Implement your model as per your expectation.

The implementation of our model is mainly a simulation in unity simulator. By using convolutional neural network/CNN we have written our simulation logic and pushed it in the simulator.
Here a video can be found where we implemented our model and ran it on our simulator: ***Video***

2. Project Task: Validation and Verification (simpler way)

a. Run the model with a known dataset.
Note: Dataset must be referred.

Data generated on our own by driving the car in the simulator in training mode.
We drove around 25% of the track. Later trained our model on this ***Data.***
A ***Video*** can be found here to show how we generated this data.

After training the model we simulated our model throughout the entire track. Here's a Video that was ran with this dataset: ***Video***

b. Compare the outputs of your model with the outputs of your dataset.

After comparing, we can say that our tested output result is very promising as it is very similar to our dataset.Our data worked very efficiently on track 1 autonomous mode as it was driven well on track 1 by turning left and right according to the dataset we gathered from training mode.

c. Based on your comparison, what updates do you think any update is necessary in your model?

After comparing my simulation output with my dataset,we realised some changes are necessary in order to make our agent more useful. It starts with data augmentation. Since our training dataset is not very big,we can do data augmentation on it to make the dataset more useful. We could easily make a bigger training dataset. However this can be taken as a challenge. We also have to improve our model architecture.

d. Based on your comparison, what mistakes did you find and edit in your code.

Our data worked very efficiently . However it sometimes failed to take a specific turn at the beginning. The error was unpredictable. To fix this issue we played around with the model architecture.

- Added two more Conv2D layers.
- Added Dense.
- Changed activation function RELU to ELU.
- Added Dropout.
- Added MaxPooling2D. Which was later dropped.(Has very little/no effect)

After changing in the model architecture we again trained the data and simulated our model. This resulted in favour of our simulation.

3. Project Task: Simulation and Result analysis

a. Run the simulation based on your problem statement.

Unity simulator has two tracks for self driving simulation. We trained our model by driving 25% of track one. We also simulated our model on track one.We were saving track two for this specific problem. Finally we simulated our model on track two. Here a ***Video*** can be found to show our model against a new map.

b. What are the outputs of your simulation?

After training the data we had an accuracy of 57%. For a self driving car it is a pretty good score The output of our simulation can be divided into two parts.For the first track when we run our agent on track one autonomously, it gives us an effective result by completing the whole track one. However later we needed to bring change in the model architecture. For the 2nd track when we simulated our agent autonomously, it gave us an average result as it could not complete the track.

c. Analyse the output of your simulation.

From the above immediate question answer on point no 1, we mention that for track-1 our test result is very effective as we trained our agent on track-1 previously. So, when we test our agent on track-1 it finds a similar data of the roads and surrounding and thus can act effectively.
On the other hand, for track-2, our test shows an average output as it fails to complete the whole track and also falls out of the track when to shift steering angle for sharp turning. The reason behind this ineffective output is as our test agent has no prior experience about track-2 roads and surrounding.

d. What conclusions can you draw from the analysis? What are your

recommendations for the system?
Our agent worked as intended. However it did not show an effective result for the 2nd track. This problem can be solved by generating sufficient training data. Also the model architecture was not perfect. This issue can be solved by doing hyper parameter optimization and using neutron.
We recommend more complex and long listed data generation for this model. As we trained our model only based on 25% of the map input,it reacted irrationally in some turns.If more training data can be generated our model can have more detailed knowledge about the dataset we provide to it and acts more versatile. Also hyper parameter optimization can be recommended to make the model architecture perfect.

4. Self Assessment:

a. Write three ways you could have improved your model and simulation.

There are several ways we could improve our simulation. We actually had plans to implement more than one model for our final project. However lack of time and good hardware became an issue.

- Advanced lane detection can be implemented.
- Path planning could be added.
- Traffic Light classification,pedestrian movement detection or vehicle detection could be implemented too.