SELF DRIVING CARS USING CNN

Batch Name : BDAROOS

SRAVANI NIHARIKA GARAPATI (16328843)

NIKHILA CHIRUMAMILLA (16323180)

SAI RAM CHAGANTI (16327655)

SHRAVANI NALLA(12576204)

Table of Contents

Introduction

- Self driving cars, one of the trending topics all over the world. A car with no steering wheel, pedals and with the most advanced technology autopilot.
- A ***Self driving car***, also known as autonomous vehicle(AV), is a car incorporating vehicle automation, that is a ground vehicle that is capable of sensing its environment and moving safely with little or no human input.
- In our project, we will be using an advanced neural network in combination with Computer vision to build a self driving car.
- It learns the whole set of algorithms required to operate a car's steering wheel.
- In a simulated driving situation, the CNN was trained and evaluated using data.
- A front-facing center camera footage may be used to generate control orders for the car's steering once it has been trained.

Why self driving cars

Below are the main reasons why we need self driving cars

- 90% reduction in traffic deaths
- 60% drop in harmful emissions
- Eliminate stop and go waits by 100%
- 10% improvement in fuel economy
- 500% increase in lane capacity
- 40% reduction in travel time
- Consumer savings of £5bn
- Greater Independence

Companies Investing in Self Driving Cars

44+

Related Works and Leaders in the Innovation

- Waymo

  Previously owned by the GOOGLE

- Tesla

  Already rolling out self-driving cars

- Waymo and Tesla are considered to be the leaders in the innovation of self-driving cars
- Waymo and Tesla use a variety of sensors and cameras to construct a 3D representation of the car's surroundings.
- A 360-degree representation of the environment requires expensive technology like as LIDAR , radar and ultrasonic sensors
- the Tesla Model S makes use of a single back camera as well as broad and narrow front cameras that view within.
- Twelve ultrasonic sensors in the back, plus a RADAR in the front, allow the automobile to measure the speed of things in front of it.

Methodology

Data Collection and Manipulation

- The goal is to forecast the steering wheel movements of a self-driving automobile based on data from a camera in front of the vehicle.
- There are two modes in system's operations

  1)Data Collection Mode

  2)Self Driving Mode
- To train the neural network, depiction of the steering wheel angle neural network, the camera records its pictures and sensors measurements. The camera inputs pictures straight into the neural network while our system is driving the automobile.
- To begin gathering data, we must first alter the surrounding environment to our favor. On various courses with sharp and smooth curves, bumps, trees and traffic signals, the simulated cameras deliver views in front of the car.
- A background script gathers information on the driver's control of the steering wheel, throttle, and brake pedals.
- The photos captured by the camera are processed before they are sent to the network.

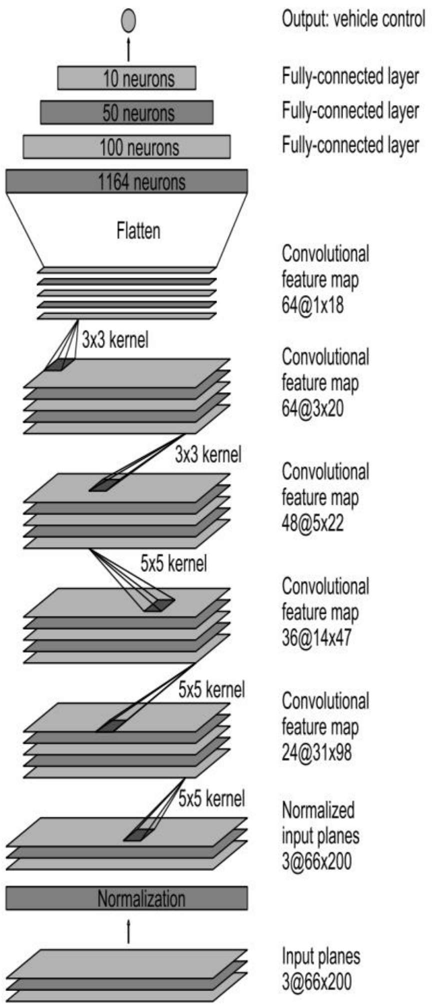
Data Collection and Manipulation(Contd..)

There are various steps before images from the camera are sent to the network:

- Selection procedure: data with desirable qualities is picked and separated into a training and a validation set, the training set consisting of 80% and the validation set consisting of 20% of the data.
- Changing the color scheme of the input photos may be accomplished via the use of the color scheme modification technique.
- The photos are then cropped to eliminate the front of the automobile, resized to 66x200, and converted from RGB to YUV since YUV takes into consideration human perception, enabling decreased bandwidth for chrominance components.
- Afterwards, the pictures are supplemented by adding simulated shifting and rotations to teach the network how to recover from a bad location or orientation. Using a normal distribution with a mean of 0, the magnitude of each of these disturbances is generated at random.
- The last step is to combine the pictures along with their related steering angel, throttle, and brake values and feed to neural networks as arrays.

Approach: Neural Networks

- Neural Networks employ numerical vectors or matrices as input and conduct operations like additions and matrix multiplication, such as matrix multiplication to control the parameters.
- In order to provide output for the next layer, activation functions require weights, biases, and a specified input.
- The findings are shown on the output layer, which is the last stage of the pipeline.
- For each input, the difference between the calculated and intended output is specified by a loss function.
- The optimizer in a neural network is used to minimize the discrepancy between predicted and actual results.
- For training, an optimization strategy starts with a random set of weights.
- Additional layer types and connections between layers are needed to create an array of neural and deep network architectures.



Architecture

- Feed forward neural networks is the NN architectural type we employed in our tests.
- First, images with dimensions of 66X200X3 are used as input, followed by five convolution layers with 24, 36, 48, 64, 64 neurons in each, each with a filter size of 5X5 and a 2X2 pixel jump at each filter pass (strides of sizes 2X2), and two final layers with filter sizes of 3X3 but no strides.
- The final two layers use filter sizes of 3X3 but no strides. Once the convolutional layers have produced their multi-dimensional output, they are followed by a flattening layer.
- There are 5 layers in total each with 100 neurons

Activation Function Used: elu

Optimizer Used: Adam

Model Used : Sequential

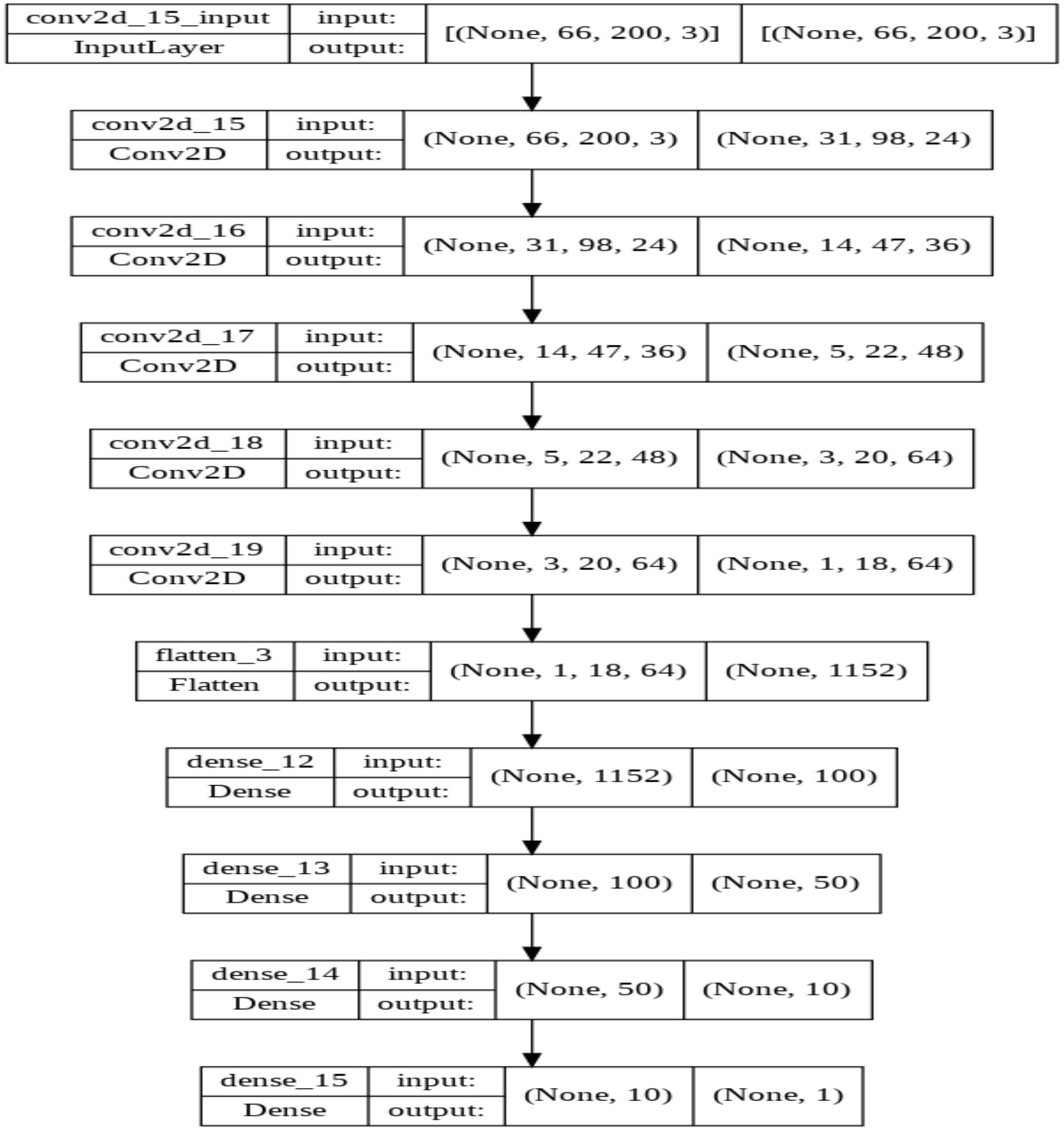
Validation Steps : 200

Number of time periods :10

Steps-per-epoch: 300.

Architecture(Contd..)

- Consider the below architecture model



Software Requirements

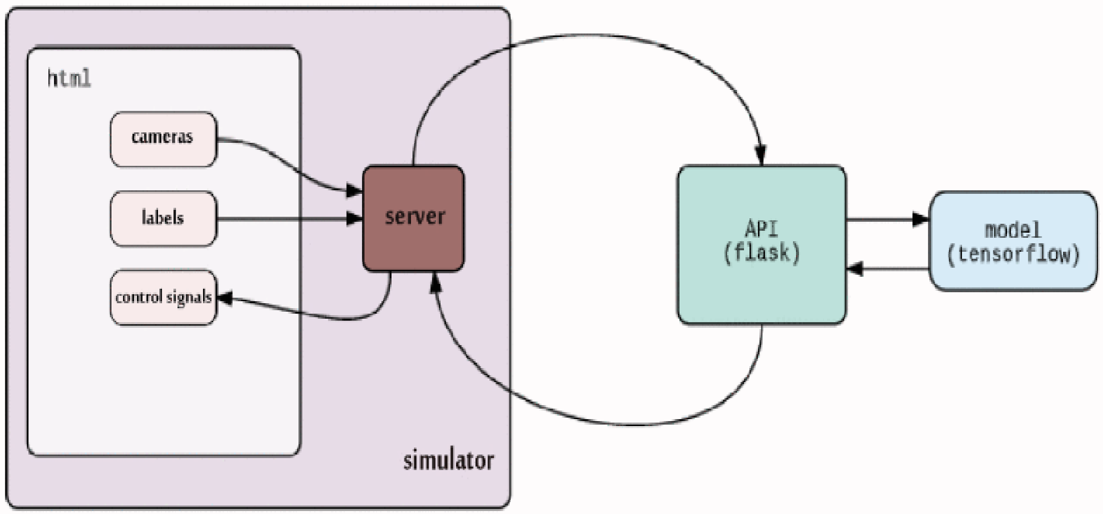
- Simulator
- TensorFlow
- Keras
- SocketIO
- Flask
- Opencv
- Pillow

Model Training

- Once the information has been gathered and analyzed, there are 1200 random samples chosen from CSV, and the network is trained on the training set by drawing random batches until the whole training set has been utilized, then drawing random batches again. The network is tested against the validation set at regular intervals
- We upload the data to a Github account and clone the data to Google Colab before we begin training. After that, Colab is utilized to get the training started.
- Using a batch generator and a model with random weights, input batches are fed into the system before being sent through the layers and compared to the provided labels. The Adam optimizer adjusts the weights.The validation loss is calculated and recorded after each run through the inputs in the first epoch check point function, and if the following epoch yielded a lower value for the validation loss, a copy of the model and its associated weights are saved as model.h5 file.
- Model.h5 file is created to save the weights for future testing and operation of the self-driving automobile when all the epochs have been completed.

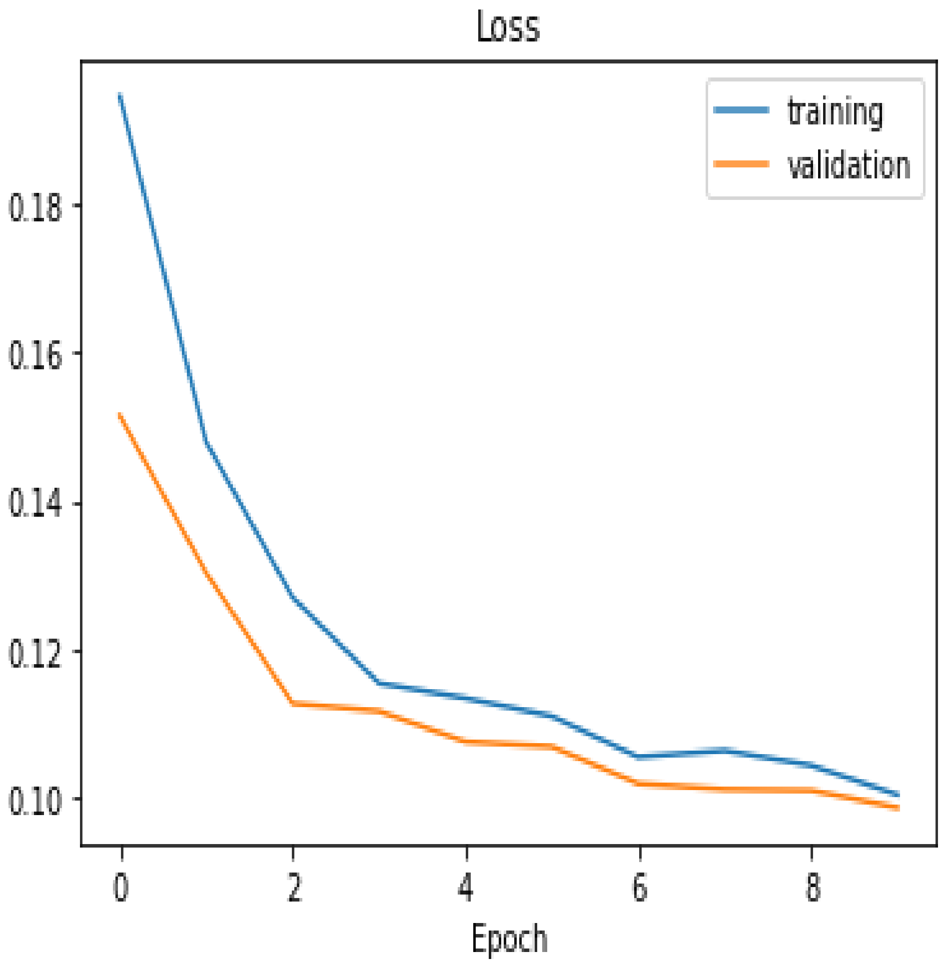
Results

- We are using anaconda tool for running the python code and installed all the required libraries and dependencies.
- We have placed the .py file and the model.h5 file in a folder. First we create an environment and then activate it where we can run the code and execute it smoothly.
- Initialize Socektio webserver which is used to perform realtime communication between client and server.When a client creates a single connection to websocket server, it keeps listening for new events from server, allows us to continuously update the client with server. It establishes bidirectional communication with the server.
- As soon as connection is established, we are setting the initial values and simulator is going to send the image details, based on the image, our model extracts features from the images and predicts steering angle.



Model Testing:

- The training and testing validation loss graph in figure shows that there is not over fitting in the model and the model was able to predict the turns effectively by using the center , left, right, steering angle, break amount and throttle amount retrieved from csv file.
- The loss achieved is 3% which is very less. If large amount of training data is used we can reduce the loss percentage even more.
- For a model to perform efficiently, it should train on huge amount of data to handle all kinds of situations.



Conclusion

- This research reveals that CNN can be taught using simulation data to run autonomous automobiles. We found that CNNs can learn the whole lane and road follow job without the need for human breakdown into lane recognition, semantic abstraction, route planning and control. Deep Neural Networks and classic computer vision methods were used in this study to see whether they might lessen risky driving behavior in self-driving cars. The process of setting up the investigation's environment and assembling the tools required a significant amount of time throughout development. Future autonomous driving initiatives will benefit tremendously from using a system that blends classic computer vision methods with the processing power of DNNs, according to the study.

Future Scope

- It's recommended that we collect additional training data in a variety of lighting settings to improve the CNN's capabilities. Definitely this will increase the precision of the car's navigation system. It is also possible to enhance the model's functionality by adding features such as the ability to avoid obstacles encountered on the course. An accurate model might benefit greatly by the CNN being implemented on a real automobile.

References

[1] G. Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6), 141-142

[2] Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/ challenges/LSVRC/..

[3] Himmelsbach,M.;Mueller,A.;etal.LIDAR-based3Dobjectperception. In Proceedings of 1st international workshop on cognition for technical systems, volume 1, 2008..

[4] Clevert, Djork-Arn, Thomas Unterthiner, and Sepp Hochreiter. ”Fast and accurate deep network learning by exponential linear units (elus).” arXiv preprint arXiv:1511.07289 (2015)..

[5] D.P. Kingma and J. Ba.Adam: A method for stochastic optimization. CoRR, abs/1412.6980,2014.

[6]Giusti A, Cireşan D C, Masci J, Gambardella L M and Schmidhuber J 2013 Fast image scanning with deep max-pooling convolutional neural networks 2013 IEEE Int. Conf. Image Process. ICIP 2013 - Proc. pp 4034–8.


THANK
YOU