UDACITY

Back to Self-Driving Car Engineer

Use Deep Learning to Clone Driving Behavior

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on successfully completing this project. This is one of the important step towards full autonomy and you certainly did a good job by implementing deep learning algorithm well to train a car think like a human while driving on road.

Required Files

The submission includes a model.py file, drive.py, model.h5 a writeup report and video.mp4.

All files are present.

Quality of Code

The model provided can be used to successfully operate the simulation.

The model was able to successfully operate the car in the simulation.

The code in `model.py` uses a Python generator, if needed, to generate data for training rather than storing the training data in memory. The `model.py` code is clearly organized and comments are included where needed.

Well done on the quality of code!

Your model is able to operate the simulation and your `model.py` code is easy to read and uses a Python generator appropriately to generate data for training.

Model Architecture and Training Strategy

The neural network uses convolution layers with appropriate filter sizes. Layers exist to introduce nonlinearity into the model. The data is normalized in the model.

Train/validation/test splits have been used, and the model uses dropout layers or other methods to reduce overfitting.

Learning rate parameters are chosen with explanation, or an Adam optimizer is used.

The Adam optimizer was used as an adaptive learning rate method

Training data has been chosen to induce the desired behavior in the simulation (i.e. keeping the car on the track).

Good job getting additional data to train your model and using all available cameras.
Furthermore, you have used data augmentation on those images to improve even more the data diversity.

Architecture and Training Documentation

The README thoroughly discusses the approach taken for deriving and designing a model architecture fit for solving the given problem.

✔ You described the process that led you to the final implementation.

I recommend watching the following talk given by Andrej Karpathy at Deep Learning School in Stanford to revise the design of convolutional neural network architectures if needed: https://www.youtube.com/watch?v=u6aEYuemt0M

The README provides sufficient details of the characteristics and qualities of the architecture, such as the type of model used, the number of layers, the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

✔ The final model is presented. The architecture was explained in sufficient details.

Suggestion :

- Including some visualizations of the used architecture would be nice. Using `graphviz`, keras provides utility functions to plot a Keras model.
- Here is a link you can explore more: Model visualization introduction Link

The README describes how the model was trained and what the characteristics of the dataset are. Information such as how the dataset was generated and examples of images from the dataset must be included.

✔ The writeup contains required information regarding the training and the dataset.

Optional video about training neural networks (and much more):

- Nuts and Bolts of Applying Deep Learning (Andrew Ng): https://www.youtube.com/watch?v=F1ka6a13S9I

Simulation

No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle).

The car successfully drives around the track. Well done.

DOWNLOAD PROJECT

Rate this review