UDACITY

Logout

PROJECT

Use Deep Learning to Clone Driving Behavior

A part of the Self Driving Car Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Congratulations, your project meets all requirements!

I've enjoyed very much reviewing your project, you can be proud of the hard work you've done!

Keep *deep*learning and stay ***Udacious***! :D

Quality of Code

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

The provided trained model works very well in the simulation, great job!

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.

The code is presented in a very good quality, well organized, commented, and formatted, a great job indeed!
Also, a very good job using a Python generator to avoid excessive memory consumption during the training process. :)

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.

- A very good job using both convolution and nonlinearity layers!
- Data is properly normalized prior training the model, great job here as well! :)

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

- Great job splitting the training dataset into training and validation subsets!
- Dropout layers were properly used to reduce overfitting, nice job here as well!

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

Adam optimizer was used, nice job!

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

Architecture and Training Documentation

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

An excellent job here this time, explaining your approach until reaching and functional model. A great job indeed!

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 topology of the implemented model is perfectly detailed in the README, also the included diagram is very clarifying, great job here as well! :)

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 should be included.

The training process is properly described in the README. Also, information and the examples of images from the dataset were included, good!

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).

Excellent job here, the car drives itself across the entire circuit and always maintaining the required trajectory, on the drivable surface even on those places where cars normally tend to slip out of the road! :)

DOWNLOAD PROJECT

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ