UDACITY

PROJECT

Use Deep Learning to Clone Driving Behavior

A part of the Self Driving Car Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW 5

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Congratulations on passing the behavioral cloning project! Through this project you became familiar with Keras, deep neural networks, and more. Not many people in the world can say they made an artificially intelligent car, so be sure to celebrate.

Happy learning and keep up the good work.

Required Files

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

All required files are present.

Quality of Code

✓ **The model provided can be used to successfully operate 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.**

Nice work using a generator. You could have also used Keras useful `ImageDataGenerator` that does a lot of the image augmentation techniques that you use. Here are the docs related to the generator.

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.**

Please see my comments in your `model.py`. The way you declared your generator for the validation data actually did not completely split your data, which could have caused problems.

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

✓

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.

✓

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 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.

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).

Great work!

DOWNLOAD PROJECT

5 CODE REVIEW COMMENTS ›

RETURN TO PATH