UDACITY

< Return to "Self-Driving Car Engineer" in the classroom

DISCUSS ON STUDENT HUB

Use Deep Learning to Clone Driving Behavior

审阅

代码审阅 17

HISTORY

Meets Specifications

Good effort! 👏
Congratulations on successfully completing this project! 🎉
The car drives well on both the tracks. You have done a commendable job using good set of techniques.

If you have time then kindly go through the following resources . You can read about various approaches which were adopted by other nanodegree students to aid the performance of their model.

- https://jacobgil.github.io/deeplearning/vehicle-steering-angle-visualizations
- https://chatbotslife.com/using-augmentation-to-mimic-human-driving-496b569760a9
- https://medium.com/@fromtheast/you-dont-need-lots-of-data-udacity-behavioral-cloning-6d2d87316c52

Keep up the good work and all the best for your future projects! 👍

Required Files

The submission includes a model.py file, drive.py, model.h5 a writeup report and video.mp4.
All the required files are present.

Qualify of Code

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

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.

- Code is organized and comments have been used appropriately.
- Nice work using generator as it is especially useful to avoid "Out of memory" issues.

Resources

Few helpful resources on python generators are given below:

- https://jeffknupp.com/blog/2013/04/07/improve-your-python-yield-and-generators-explained/
- https://www.youtube.com/watch?v=bD05uGo_sVI

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.

- Network has adequate number of convolution layers.
- Nice work using activation function(ReLU) to achieve Nonlinearity.
- Good job normalizing the data in the model.

PS. Faster alternative to ReLU activation is ELU. For more information on ELU refer https://arxiv.org/pdf/1511.07289v1.pdf

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

- Data has been split properly.
- Nice job using dropout layers to reduce overfitting .

Resources

Few resources on ways to reduce overfitting are given below:

- http://neuralnetworksanddeeplearning.com/chap3.html#overfitting_and_regularization
- https://www.jeremyjordan.me/deep-neural-networks-preventing-overfitting/
- https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d

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

Nice job using Adam optimizer!

Few helpful resources on Adams optimizer are given below:

- http://cs231n.github.io/neural-networks-3/#ada
- https://www.youtube.com/watch?v=JXQT_vxqwIs

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

Good job using all camera images from sample data (provided by udacity), collecting new data and augmenting the dataset with flipped camera images to improve 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.

Kudos for discussing your approach and it is nice to see that your model architecture design is based on the NVIDIA model.

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 architecture has been described thoroughly and visualization of the model was provided along with its layers and parameters. Well done!

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.

- Model training has been described along with appropriate images. Nice Job!

- Excellent job including the plot of steering angles distribution before and after filtering the steering bias.

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 is very stable and navigates around drivable portion of both tracks flawlessly. Well done 👏

下载项目

17 代码审阅评注 ›

返回 PATH