UDACITY

Back to Self-Driving Car Engineer

Use Deep Learning to Clone Driving Behavior

REVIEW

CODE REVIEW 5

HISTORY

Meets Specifications

Excellent work, you nailed it!
👏👏👏

I can see you put a lot of effort in your project and advanced a lot, you should be really proud! You have shown a firm grasp of the concepts presented here and are good to go.

If you curious, check my take on this project here.

On further work on end-to-end driving I recommend this article which adds temporal visual cues using LSTM.

Keep going and good luck!
Paul

PS. If you have further questions remember you can find me on Slack as `@viadanna`

Required Files

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

All required files are present.

Qualify of Code

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

The provided `model.h5` can be used to successfully drive the autonomous car.

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.

Awesome

Excellent work implementing a generator to train the model.

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.

Nice choice of architecture.

Suggestion

Alternatives here are nVidia's PilotNet and comma.ai.

I highly recommend using nVidia's here.

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

Excellent work using pooling, dropout and dataset augmentation to reduce variance.

Suggestion

Another technique commonly used here is L2 regularization.

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

Good idea using the AdamOptimizer that actually takes care of updating the learning rate.

Suggestion

If you want to know more about optimizers, check this article for a nice description and comparison of different algorithms.

Instead of using a fixed number of epochs one alternative is using Keras' EarlyStopping callback which stops training the model when it stops improving.

Another useful callback is ModelCheckpoint which can be used to save the best model found during training.

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

Excellent work here, your model developed nice driving skills based on the training data.

Awesome

Excellent work augmenting the training dataset.

Suggestion

For better results you might want to balance the amount of examples based on the steering angles.

Architecture and Training Documentation

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

Awesome

Excellent work doing lots of experimentation here, this shows an inquisitive mind that'll be of great benefit on the course and career 👍

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.

Nice description of the architecture.

Suggestion

If you desire to deep-dive into the architecture using TensorBoard, Keras includes a callback to simplify saving the logs.

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.

Good job documenting how the training dataset was generated, preprocessed and augmented.

Suggestion

I recommend plotting a histogram of the angles in the training dataset to check the balance of the output. Remember that an unbalanced dataset can result in a biased model.

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 model correctly navigated the required course, congratulations!

DOWNLOAD PROJECT

5 CODE REVIEW COMMENTS ›