Behavioral Cloning

Writeup Template

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

Behavioral Cloning Project

The goals / steps of this project are the following: * Use the simulator to collect data of good driving behavior * Build, a convolution neural network in Keras that predicts steering angles from images * Train and validate the model with a training and validation set * Test that the model successfully drives around track one without leaving the road * Summarize the results with a written report

Rubric Points

Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

Files Submitted & Code Quality

1. Submission includes all required files and can be used to run the simulator in autonomous mode

My project includes the following files:

- model.py containing the script to create and train the model

- drive.py for driving the car in autonomous mode

- track1_model.h5 containing a trained convolution neural network for track1

- track2_model.h5 containing a trained convolution neural network for track2

- writeup_Omkar_Behavioural_cloning.pdf summarizing the results

2. Submission includes functional code

Using the Udacity provided simulator and my drive.py file, the car can be driven autonomously around the track by executing

```
python drive.py track1_model.h5 #For track1

python drive.py track2_model.h5 #For track2
```

3. Submission code is usable and readable

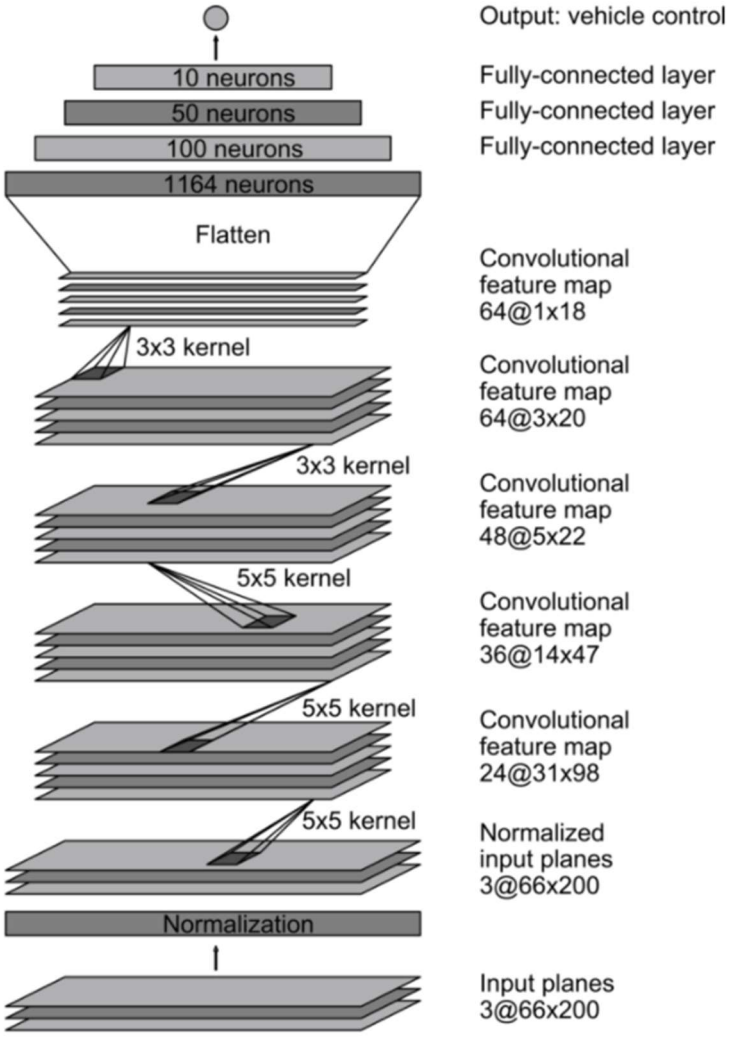
The model.py file contains the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

I initially tested the whole code in a Jupyter Notebook. The notebook Training.ipynb is also in the submission.

Model Architecture and Training Strategy

1. An appropriate model architecture has been employed

My model is based on a well known Nvidia published architecture. This was published in the following link: https://devblogs.nvidia.com/deep-learning-self-driving-cars/



As a part of iteration, in the beginning my model was based on watching the tutorial videos in the lectures and mimicking the network that is shown. This network yielded only slightly acceptable results on track1. The resulting drive from this network is included as track1_run1.mp4

Subsequent improvements with the Nvidia network resulted in better runs as shown in videos track2_run1.mp4 and track1_run2.mp4.

2. Attempts to reduce overfitting in the model

The model contains dropout layers in order to reduce overfitting (model.py lines 21).

The original Nvidia model shown above does not contain any drop out layers. I added drop out layers after each dense layer.

3. Model parameter tuning

The model used an adam optimizer, so the learning rate was not tuned manually (model.py line 25).

4. Appropriate training data

This is where the success of my model comes from. Initially when I only recorded a lap while driving in the middle of the lane on track1, the results showed that the autonomous runs were driving off the road. Then I concatenated the drive_log with following drives:

- Clockwise middle of the lane driving
- Clockwise zigzag driving – this is where I recorded driving away from the edges all along the track towards the middle of the lane. When I was driving away from middle of the lane recording was stopped.
- Same as above two but in counter-clockwise direction

The same recording was done for track 1 and track 2

A separate validation set was recorded by driving in the middle of the lane on both track1 and track2.

5. Image Augmentation

By augmenting incoming images to the generators, I was able to speed up the network performance. Initially, without image augmentation I needed 10 epochs to train the network to enable satisfactory autonomous drive, but after image augmentations, I was

able to generate the neural network with only 2 epochs. Instead of reinventing the wheel, I used some image augmentation from the traffic signal classifier project and some from the link below: https://chatbotslife.com/using-augmentation-to-mimic-human-driving-496b569760a9

Each image undergoes augmentation in following sequence:

Original image -> translated image -> brightened image -> image with random shadow -> flip the image (half of the images get flipped)

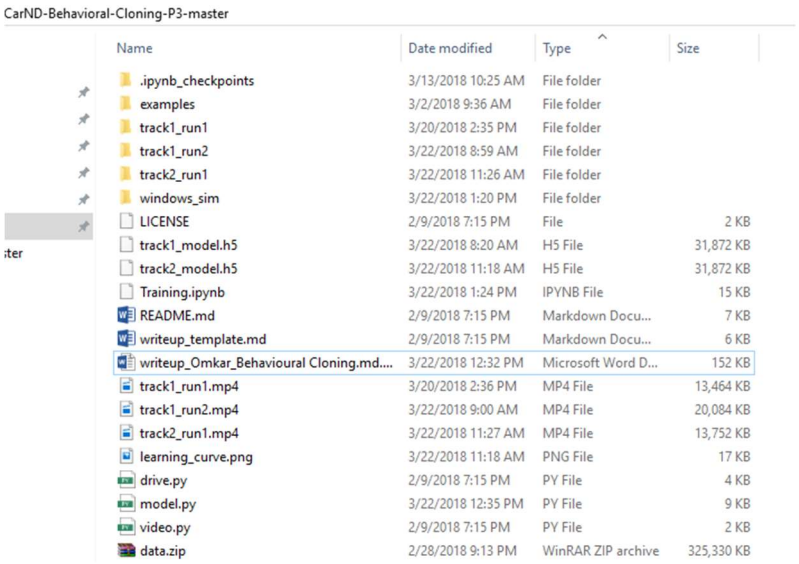
This kind of augmentation generated data which was richer than the original recorded data.

6. Data generators

As suggested in the tutorials, the amount of data generated was fairly large. There were approximately 50000 images to crunch. Loading this amount of data in memory would have been very inefficient. Instead, I created data_generators which provide batched image/steering angle data to the model. While initially loading the data, I only read the file locations. The data generators use the file locations to load the images and then before passing the images to the network, they augment the images as explained before.

7. Miscellaneous

Following images show the folder structure of the project detailing the image storage on local PC. been very inefficient. Instead, I created data_generators which provide batched image/steering angle



CarND-Behavioral-Cloning-P3-master › windows_sim ›

Name	Date modified	Type	Size
IMG	3/22/2018 10:47 AM	File folder	
windows_sim_Data	3/22/2018 8:42 AM	File folder	
track1_ccw_middle_train.csv	3/20/2018 11:14 AM	Microsoft Excel C...	572 KB
track1_ccw_zigzag_train.csv	3/20/2018 11:31 AM	Microsoft Excel C...	262 KB
track1_cw_middle_train.csv	3/20/2018 11:21 AM	Microsoft Excel C...	861 KB
track1_cw_zigzag_train.csv	3/20/2018 11:28 AM	Microsoft Excel C...	550 KB
track1_drive_log_train.csv	3/20/2018 2:01 PM	Microsoft Excel C...	2,250 KB
track1_drive_log_validate.csv	3/22/2018 1:20 PM	Microsoft Excel C...	562 KB
track2_ccw_middle_train.csv	3/22/2018 10:08 AM	Microsoft Excel C...	929 KB
track2_ccw_zigzag_train.csv	3/22/2018 10:22 AM	Microsoft Excel C...	544 KB
track2_cw_middle_train.csv	3/22/2018 10:28 AM	Microsoft Excel C...	696 KB
track2_cw_zigzag_train.csv	3/22/2018 10:35 AM	Microsoft Excel C...	364 KB
track2_drive_log_train.csv	3/22/2018 11:01 AM	Microsoft Excel C...	2,719 KB
track2_drive_log_validate.csv	3/22/2018 11:02 AM	Microsoft Excel C...	731 KB
windows_sim.exe	1/19/2017 7:11 AM	Application	17,786 KB

CarND-Behavioral-Cloning-P3-master › windows_sim › IMG

Name	Date	Type	Size
left_2018_03_20_11_28_16_309.jpg	3/20/2018 11:28 AM	JPG File	17 KB
left_2018_03_20_11_25_05_544.jpg	3/20/2018 11:25 AM	JPG File	17 KB
center_2018_03_20_11_31_43_782.jpg	3/20/2018 11:31 AM	JPG File	17 KB
left_2018_03_20_11_28_16_383.jpg	3/20/2018 11:28 AM	JPG File	17 KB
left_2018_03_20_11_25_05_473.jpg	3/20/2018 11:25 AM	JPG File	17 KB
center_2018_03_20_11_25_05_544.jpg	3/20/2018 11:25 AM	JPG File	17 KB
center_2018_03_20_11_31_44_074.jpg	3/20/2018 11:31 AM	JPG File	17 KB
center_2018_03_20_11_31_43_856.jpg	3/20/2018 11:31 AM	JPG File	17 KB
left_2018_03_20_11_25_05_618.jpg	3/20/2018 11:25 AM	JPG File	17 KB
left_2018_03_20_11_25_05_401.jpg	3/20/2018 11:25 AM	JPG File	17 KB
center_2018_03_20_11_25_05_473.jpg	3/20/2018 11:25 AM	JPG File	17 KB
center_2018_03_20_11_25_05_618.jpg	3/20/2018 11:25 AM	JPG File	17 KB

Following image shows the model.py being run on the anaconda prompt with track1 training:

```
(carnd-term1) C:\Users\omkar.karve\CarND-Behavioral-Cloning-P3-master>python model.py
C:\Users\omkar.karve\AppData\Local\Continuum\anaconda3\envs\carnd-term1\lib\site-packages\h5py\__init__.py:36: FutureWarning: Conversion of the
second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`
.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
starting
No of training steps 80
No of validation steps 20
Epoch 1/2
80/80 [==============================] - 567s 7s/step - loss: 0.0808 - val_loss: 0.0329
Epoch 2/2
80/80 [==============================] - 590s 7s/step - loss: 0.0662 - val_loss: 0.0332
Saving model to model.h5

(carnd-term1) C:\Users\omkar.karve\CarND-Behavioral-Cloning-P3-master>
```