Use Deep Learning to Clone Driving Behavior

Required Files

CRITERIA	MEETS SPECIFICATIONS
Are all required files submitted?	The submission includes a model.py file, drive.py, model.h5 a writeup report and video.mp4. The following files are present in the zip file. 1. model.py 2. model.h5 3. drive.py 4. final_run.mp4 5. Writeup_Behavior_Cloning.pdf

Quality of Code

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Is the code functional?	The model provided can be used to successfully operate the simulation.	The model is working fine. Please refer to the final_run.mp4 in the zip file submitted
Is the code usable and readable?	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 model.py mainly consists of 8 parts 1. all imports (lines 1-13) 2. reading the measurements (lines 19-24) 3. a generator function (lines 44-96) 4. generating the train_generator, validation_generator (lines 99, 100) 5. Commented out LeNet model (lines 101-122) 6. Nvidia model (lines 124-155) 7. running the model with fit_generator (lines 173-176) 8. saving the model and model weights (182-188)

Model Architecture and Training Strategy

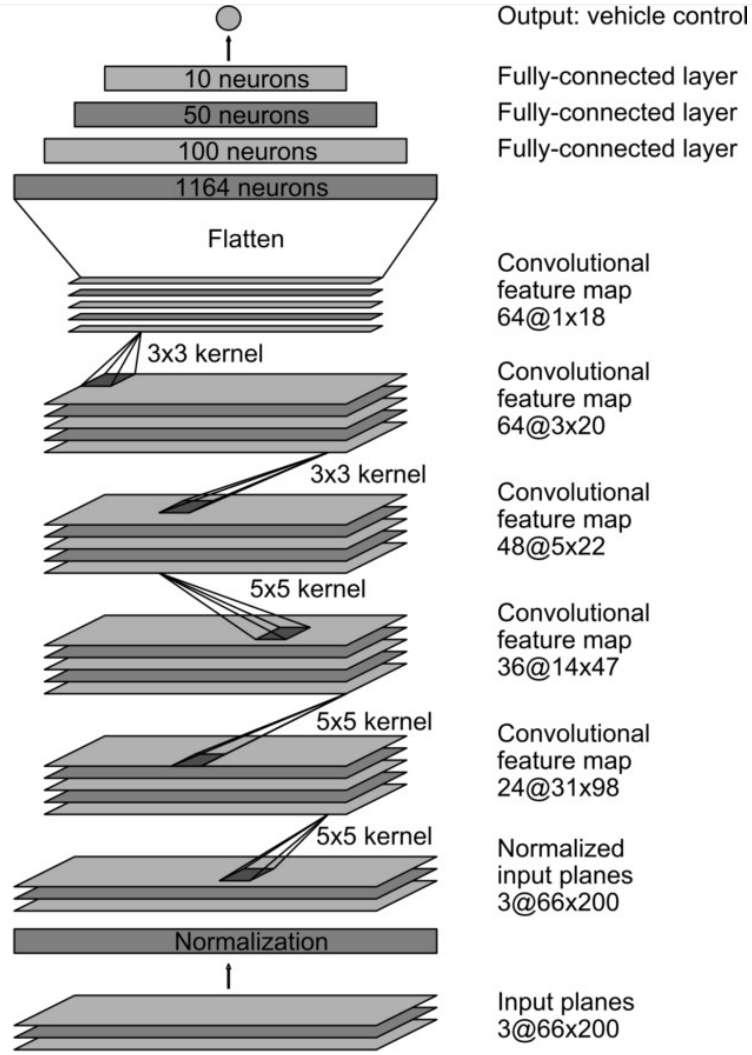
CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Has an appropriate model architecture been employed for the task?	The neural network uses convolution layers with appropriate filter sizes. Layers exist to introduce nonlinearity into the model. The data is normalized in the model.	As a trial LeNet model was used and but decided on the Nvidia model.
Has an attempt been made to reduce overfitting of the model?	Train/validation/test splits have been used, and the model uses dropout layers or other methods to reduce overfitting.	This has been done in line 31. The dropout regularization was used to avoid over fitting The track was driven close to three laps in both directions. Some data was collected from edges. Also, in the initial training, the car was going to gravel after the different colored bridge. At that time more data was taken and retrained the model
Have the model parameters been tuned appropriately?	Learning rate parameters are chosen with explanation, or an Adam optimizer is used.	An adam optimizer is used
Is the training data	Training data has been chosen to induce the	The track was driven close the three laps in

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
chosen appropriately?	desired behavior in the simulation (i.e. keeping the car on the track).	both directions. Some data was collected from edges. Also, in the initial training, the car was going to gravel after the different colored bridge. At that time more data was taken and retrained the model

Architecture and Training Documentation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Is the solution design documented?	The README thoroughly discusses the approach taken for deriving and designing a model architecture fit for solving the given problem.	Based on the training lectures, I trained using LeNet but soon realized the model is not working well. A couple of reasons why this could be the case is that the error is not at all decreasing after only a few iterations. Even after 20 epochs, both the training loss and validation loss is stagnated around 0.04. As soon as I realized that the lenet is not satisfactory, as recommended in the lectures quickly moved to nvidia model. Once moved to the Nvidia model, quickly the task become the tuning of parameters. Originally, I just trained the model without any regularization techniques. The model performed well in only a few iterations. As part of the review comments from my first submission, I added a few drop out layers. I tested the dropout layers between every layer. But then I didn’t get enough stability as I trained by tuning, * drop probability * number of epochs * batch sizes. Later I commented out some drop out layers and finally achieved some stability with only 3 drop out layers between and after the convolutional layers. As I would describe in the next part in training, the model is

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		really stable and also didn’t find any over fitting.
Is the model architecture documented?	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 final model used is nvidia model with add-on dropout layers. The following is a depiction of the nvidia model. This model is nothing but the nvidia model described in https://devblogs.nvidia.com/deep-learning-self-driving-cars/

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS

```
Layer (type)                 Output Shape              Param #
=================================================================
lambda_1 (Lambda)            (None, 66, 200, 3)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 31, 98, 24)        1824
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 47, 36)        21636
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 47, 36)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 5, 22, 48)         43248
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 20, 64)         27712
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 20, 64)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 18, 64)         36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
dropout_3 (Dropout)          (None, 1152)              0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               115300
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 11
=================================================================
Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0
```

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		Network Layer Description The network consists of 1 Normalization layer, 5 convolutional layers, 3 dropout layers and 3 fully connected layers. **Layer1: Normalization Layer** The first layer is taking an image of size 66x200x3 RGB (the image is converted from BGR2RGB as part of preprocessing during generator step) image and normalizes it so that the values are between -1.0 and 1.0 **Layer2: Convolutional Layer** The first convolutional layer applies a filter size of 5x5 and stride of 2x2, resulting in output of 24@31x98. **Layer3: Convolutional Layer** The first convolutional layer applies a filter size of 5x5 and stride of 2x2, resulting in output of 36@14x47. **Layer4: Dropout Layer:** The first drop out layer is applied with a drop probability of 0.3 **Layer5: Convolutional Layer:**

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		The third convolutional layer applies a filter size of 5x5 and a 2x2 stride resulting in an output of 48@5x22 **Layer6: Convolutional Layer:** The third convolutional layer applies a filter size of 3x3 and a 2x2 stride resulting in an output of 64@3x20 **Layer7: Dropout Layer:** The first drop out layer is applied with a drop probability of 0.4 **Layer8: Convolutional Layer:** The third convolutional layer applies a filter size of 3x3 and a 2x2 stride resulting in an output of 64@1x18 **Layer9: Dropout Layer:** The first drop out layer is applied with a drop probability of 0.5 **Layer10: Flatten Layer:** The output is then flattened to 1164 neurons. **Layer 11-13: Fully-connected Layers** Each of the fully-connected layers then results in 100, 50 and 10

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		neurons respectively. In addition, each of the convolutional later is activated using relu activation function. As an optimizer, I have used adam optimizer to get the benefit of adaptive learning rate. The model finally ended up having over 250k parameters.
Is the creation of the training dataset and training process documented?	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.	**Creation of the training dataset and Training process:** The training and test sets are created using the data obtained via data collection, data augmentation techniques. **Data Collection:** 1. The original data provided as part of the data was used. 2. In addition to the original data, additional data is collected via the following means; a. 3 laps of data in forward direction is collected. b. 3 laps of data was collected in the reverse direction.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		c. Data was collected from the edges going to the middle of the road. 4. Some data was collected near the gravel after the different colored bridge. After initial training was done, the car was taking a deviation and going on the gravel road. That effect is negated by collecting additional data at that point by diverting the car away from the dirt/gravel road. The final data is split in to test and train set into 80:20 ratio. **Data Augmentation:** The collected data is augmented using the following methods 1. Data from all three cameras was used. 2. Each image is flipped 180 degrees. That way we have 6 images for each image. **Data Processing:** Each images is processed as per the steps below: 1. Each image is converted in to RGB from BGR. 2. Each image is cropped to process only the road image and not the surrounding areas.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		3. Then each image is resized to match the nvidia model. 4. The data is shuffled before the generator and at the time of yielding by the generator. Once we have the data, then most of the time is spent in training the model. The model is trained by changing the following parameters. 1. Batch size 2. Drop probability 3. Epochs As mentioned earlier, the additional data was collected when the car was taking deviations like going to gravel road. Again, as mentioned earlier, The model was originally trained without regularization. The model showed good progress around 2-4 epochs when training without regularization. But training with more epochs causing overfitting and the car was not at the center of the road.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		To train with regularization, I have introduced dropout layers all across the model. At each stage the car tended to go away from the road or beyond the road. By using multiple trials, I arrived at a decently performing model using 720 as batch size, drop probability of .3, .4, and .5 at each of the drop out layers respectively. Now the remaining work is to determine the number of epochs. Since originally, the training converged with less number of epochs, I started with 3 epochs and slowly adding epochs. Based on the tests, the model seems to show improvements as we increase the number of epochs. Around 8-10 epochs, the model seems to reach less than 2% validation loss. In the initial training, the training is stopped, as soon as the validation loss started increasing. But then when drop out regularization is introduced, I concentrated to reach 2% or less validation loss. As we use dropout regularization, minor changes(increases) in validation loss is acceptable. The advantage of regularization techniques is prevention over fitting. As I gained the confidence on the model, I went on training up to 50 epochs. Still the model performed nicely. But stuck to 10 epochs for the final result.

Simulation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Is the car able to navigate correctly on test data?	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).	Please check the video final_run.mp4. One interesting this about this project is as it can be observed from the video, the model performed for long hours in the middle of the road.