

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.

link to files:

<https://github.com/martonistvan/CarND-Behavioral-Cloning-P3.git>

Model Architecture and Training Strategy

Criteria	Meets Specifications
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.

my network consists of the following layers:

1, input shape is 160x320x3

2, "lambda" normalization layer, mean centered

3, cropping layer to remove the upper (70 pixels) and bottom part (25 pixels) of the image as they are not relevant for the training. output shape: 65x320x3

4, 1st convolution layer with kernel size 5x5, stride 2x2, depth 24, and with RELU activation. Output shape: 31x158x24

5, 2nd convolution layer with kernel size 5x5, stride 2x2, depth 36, and with RELU activation. output shape: 14x77x36

6, 3rd convolution layer with kernel size 5x5, stride 2x2, depth 48, and with RELU activation. output shape: 5x37x48

7, 4th convolution layer with kernel size 3x3, stride 1, depth 64, and with RELU activation. output shape: 3x35x64

8, 5th convolution layer with kernel size 5x5, stride 1, depth 64, and with RELU activation. output shape: 1x35x64

9, flatten layer. output shape: 2112

10, 1st fully connected layer with output size 100

11, 2nd fully connected layer with output size 50

12, 3rd fully connected layer with output size 10

13, final fully connected layer with output size 1 as we have 1 label

I tried various of network combinations to enhance training, like max pooling between convolution layers with filter 2x2, dropout layers. At the end the above described layers gave me the best result.

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.
---	--

I split dataset into training and validation set with a validation ratio of 30 percent.

I also shuffled the data to avoid overfitting, result better generalizations.

Have the model parameters been tuned appropriately?	Learning rate parameters are chosen with explanation, or an Adam optimizer is used.
--	---

I used “adam” optimizer for the learning process as it is well suited for problems with large dataset. Used to minimize the loss function similarly to what SGD does. Learning rate is the default “0.001). I applied mean squared error (MSE) for loss function.

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

For training, I recorded data in the simulator.

1, 2 full laps driving in the middle



2, 1 lap driving on the side line



3, 1 lap recovery driving. Right before going off the road I switched off recording and switched it on again when the car was recovering, so the model knows what to do when the car is about to leave the road.



I tried to use left and right camera images for the training, but at the end it was sufficient to use only the center camera image.