Behaviour Clonning Project

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
* model.h5 containing a trained convolution neural network
* writeup_report.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 model3.h5

3. Submission code is usable and readable

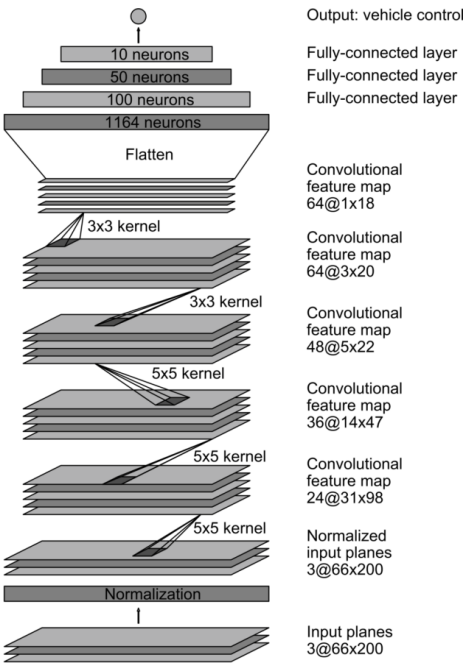
The model.py file contains the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

Model Architecture and Training Strategy

1. An appropriate model architecture has been employed

The network architecture, which consists of 9 layers, including a normalization layer, 5 convolutional layers (3 layer having kernel size (5,5) and two layers having kernel size(3,3)), and 3 fully connected layers.

The model includes ELU layers to introduce nonlinearity , and the data is normalized in the model using a Keras lambda layer .



In the above model first five convolutional layers will extract the features of the image and the remaining four fully connected layers will predict the correct steering angle.

Layer 1: Convolutional. The output shape 24@31x98.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Layer 2: Convolutional. The output shape 36@14x47.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Layer 3: Convolutional. The output shape 48@5x22.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Batch Normalisation

Layer 4: Convolutional. The output shape 64@3x20.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Layer 5: Convolutional. The output shape 64@1x18.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Batch Normalisation

Layer 6: Fully Connected. 100 outputs.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Layer 7: Fully Connected. 50 outputs.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Layer 8: Fully Connected. 50 outputs.

Activation. ELU activation.

Dropout : Probability of Dropping = 0.5

Final layer (9): 1 output

Data Set Generation:

Using Udacity Car simulator data is collected by driving car in the centre of the track.To generalise the model data is collected in both clockwise and anti clockwise direction.

Data Preprocessing:

1. First the image is cropped to avoid trees and some unwanted pixels in the top using Cropping2D in keras.
2. Then the image is normalised using keras lambda layer.

$$x = (x/255.0 – 0.5))$$

3. Attempts to reduce overfitting in the model

The model contains dropout layers after each convolutional and fully connected layers in order to reduce overfitting.

The model was trained and validated on different data sets to ensure that the model was not overfitting . The model was tested by running it through the simulator and ensuring that the vehicle could stay on the track.

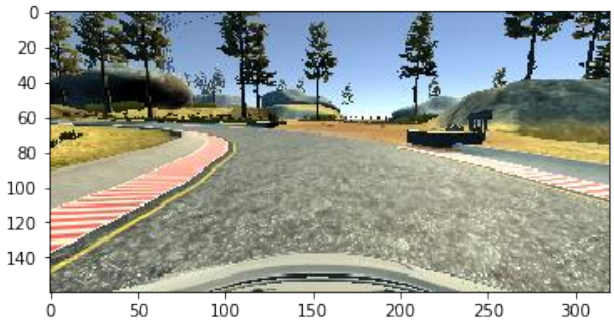
4. Model parameter tuning

The model used an adam optimizer, so the learning rate was not tuned manually .

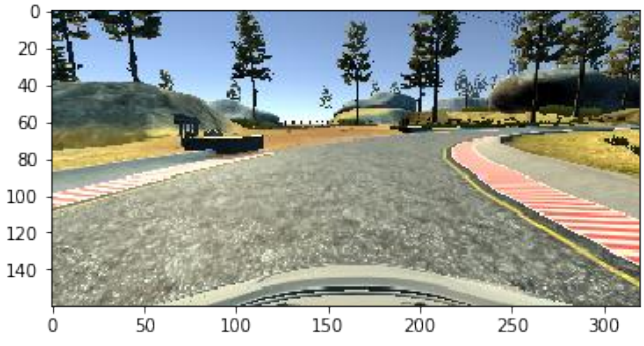
5. Solution Design Approach

The training data set contains four types of data ,

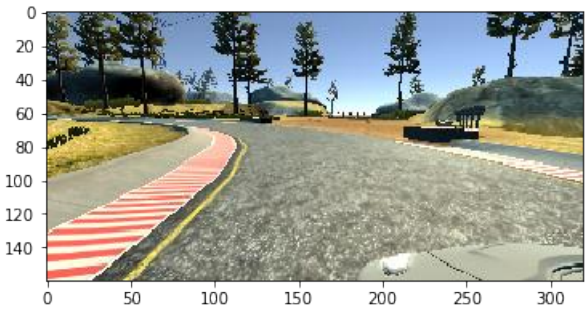
1. Centre camera images



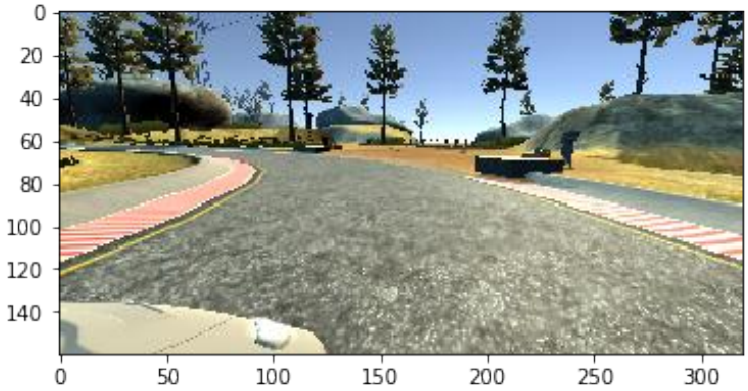
2. Flipped centre camera images



3. Left Camera images



4.Right Camera images



The reason for using the flipped center camera images is to generalise the model to drive in both clockwise and anticlockwise direction of the track.

Left and Right Camera images will make the car to drive back to the centre of the road when it drifts off from the center.

Then the occurance of the images having large steering values are increased to tackle the sharp turns in the track.

While testing my model in the simulator i found out that the car didn’t drive well in the bridge so i increased the no of images of the bridge.

By using validation set of images I found that my model is overfitting and i introduced dropout after every layer.