Behavioral Cloning

End to End Learning for self-driving cars

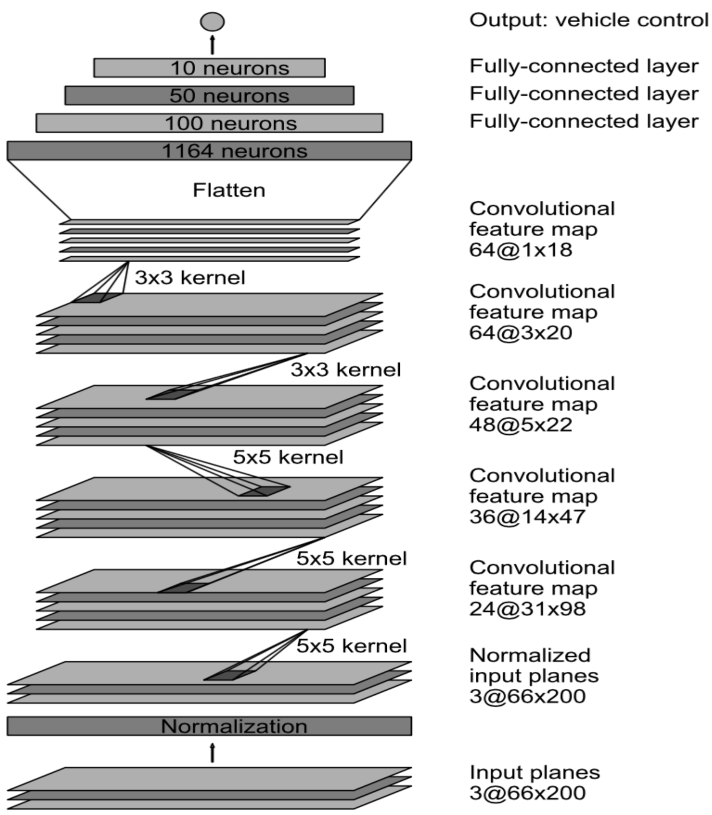
By Jingqiang Zha

The goals / steps of this project are the following:

* Use the simulator to collect data of good driving behavior (I don't have a hand shank and use keyboard to train the simulator is quite difficult so finally I decided to use the data given.)
* 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

Model Architecture and Training Strategy

1. Model Architecture



A similar architecture to NVIDA is used. Normalization layer→Conv(5,5,24) →MaxPool(2,2)→Dropout(0.5)→Conv(5,5,36)→MaxPool(2,2)→Dropout(0.5)→Conv(5,5,48)→MaxPool(2,2)→Dropout(0.5)→Conv(3,3,64)→MaxPool(2,2)→Dropout(0.5)→Conv(3,3,64)→MaxPool(2,2)→Dropout(0.5)→RELU→Full(240)→Dropout→RELU→Full(120)→Dropout→RELU→Full(20)→Dropout→RELU→Full(1)

The model includes RELU layers to introduce nonlinearity, and the data is normalized in the model using a Keras lambda layer. Also Keras cropping is used to select an effective area of interest.

2. Attempts to reduce overfitting in the model

The model contains dropout layers in order to reduce overfitting.
The model was trained and validated on different data sets to ensure that the model was not overfitting (0.2). The model was tested by running it through the simulator and ensuring that the vehicle could stay on the track.

3. Model parameter tuning

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

4. Appropriate training data

Training data was chosen to keep the vehicle driving on the road. I used a combination of center lane driving, recovering from the left and right sides of the road. Data augmentation techniques including brightness adjustment, shifting and image flipping, and adding shadow are used.

After training, the car is able to remain on the track though the training and validation error is still high (around 0.02). Also, the car oscillated around the center of the lane a lot. When I tested the car in the other track, it failed.