Behavior Cloning

Train a model to drive the vehicle autonomously

The goals / steps of this project are the following:

- Collecting training data.
- Data processing.
- Design the Model Architecture.
- Train the model.

Reflection

1. Data Collecting

The Data should get with the simulator by driving it manually, but I am not a good game player and always drive the car into the pool or off the road even though I spend such a lot of time to train myself that my fingers of my left hand got spastic. At last I decided to use the data stored in “opt” folder in workspace. This dataset has about 18070 images. In these images, the car was driven as in the middle of the track as possible. Before the car leaving the track, the car was steered back towards the center of the road with the steering angles adjusting. When passing the curve, the car was driven smoothly.

The center images were flipped and the opposite of steering angles were taken to augment my training data.

Because that each frame has three images(from left, right and center camera), to enhance the model’s accuracy, All these three images are used in my model. For an offset angle has been added to left and right images, the steering angles were amended with +0.20 for left images and -0.20 for right images.

2. Data processing

Step 0: Cropping the Images

The images were cropped with the first 50 row pixels and the last 20 row pixels to capture useful features faster.

Step 1: Normalization of the Images

With the normalization, on the one hand, the neuron output saturation caused by excessive absolute value of input is prevented; on the other hand, some small values in the output data are not swallowed. The pixels values of the images were divided by 255.0 and subtracting 0.5 to center them around zero.

3. Design Model Architecture

My model architecture is as below.

Layer		Output Shape
Two 3×3 ×16 convolution layer	First	43X159X16
	Second	42X157X16
BatchNormalization layer		42X157X16
2×2 strides Maxpooling layer(dropout)		21X79X16
Two 3×3×32 convolution layer	First	19X77X32
	Second	17X75X32
BatchNormalization layer		17X75X32
2×2 strides Maxpooling layer(dropout)		9X38X32
Two 3×3×32 convolution layer	First	7X36X32
	Second	5X34X32
BatchNormalization layer		5X34X32
2×2 strides Maxpooling layer(dropout)		3X17X32
Flatten layer		1632
Fully connected 1024 (dropout)		1024
Fully connected 512(dropout)		512
Fully connected 100(dropout)		100
Fully connected 1		1

The convolution layers were employed firstly for feature extraction。

Batch normalization layers was added to accelerate training and to reduce the influence of the covariate shift, the model will get more strong and robust。

Dropout layers have also been applied to each hidden layers to avoid overfitting.

At last the fully connected layers are used for regression.

4. Training the model

The 20% of the samples were split for validation set.

The generator was used to save memory space. The train generator and valid generator batch size were set to 32.

The adam optimization algorithm was used to train the model with 8 epochs.

The mean squared error was selected as the metric of the model. And finally, the training mean squared error and the validation mean squared error is less than 0.016.

My model is trained as “mymodel.h5”.

5. Testing the model

With the drive.py, my model is used to drive the car autonomously in the simulator and the effect is OK and the video “run1.mp4” show the result.