Project 3 Report – Behavioral Cloning

Rafael Barreto

INTRODUCTION

This third project of this Self-driving car Nanodegree was the most tiring and difficult project that I did from Udacity. And one of the most interesting too.

This report contains only the successful and final model description. Over the past two weeks, some approaches and architectures were used without good results but provided practical insights that were fundamental to the success of the project.

Before explaining my project I would like to mention some points:

Python generators: Generators are very good to memory management but when I use the the compilation speed decrease so much that I preferred sometimes stay without them. The computer stay useless without memory but is much faster. I have a Geforce GTX660 with 2Gb of dedicated memory and my computer has i5 processor and 8 Gb of memory.

Number of epochs: Around 5 epochs seems to be enough training for my architecture. Any more does not reduce the MSE much if at all. I used only 3 epochs and 5 epochs in the last training.

Numbers of camera: I stay with only the center camera and I think that with this simple circuit it is enough to training a vehicle.

GPU: Tensorflow installed on GPU help a lot speed the project development.

Graphic quality and Screw resolution: I read that the fastest graphic quality and lowest screen resolution helped the model to perform better so I did this way after the first week and things worked better.

Quality and amount of the data: The amount of the images generated by the training mode of the simulator necessary for a good dataset is relative and five or six laps for this project generate a good dataset amount for one training for my project. The quality of the data is more important than the amount driving properly with especially attention with the curves was very important.

Analog steering variation: I used the mouse to do the curves. With it I could keep the angles like a steering wheel of a car and the angles variation is "close to analog". Using the keyboard keys is not a good approach.

Nvidia pipeline: This architecture is not too complex and with excellent proven performance.

MODEL ARCHITECTURE AND TRAINING STRATEGY

The weights of this network were trained to minimize the mean-squared error between the steering command output by the network. Figure 5 shows the network architecture, which consists of 9 layers, including a normalization layer, 5 convolutional layers, and 3 fully connected layers. The model includes RELU layers to introduce nonlinearity, and the data is normalized in the model using a Keras lambda layer.

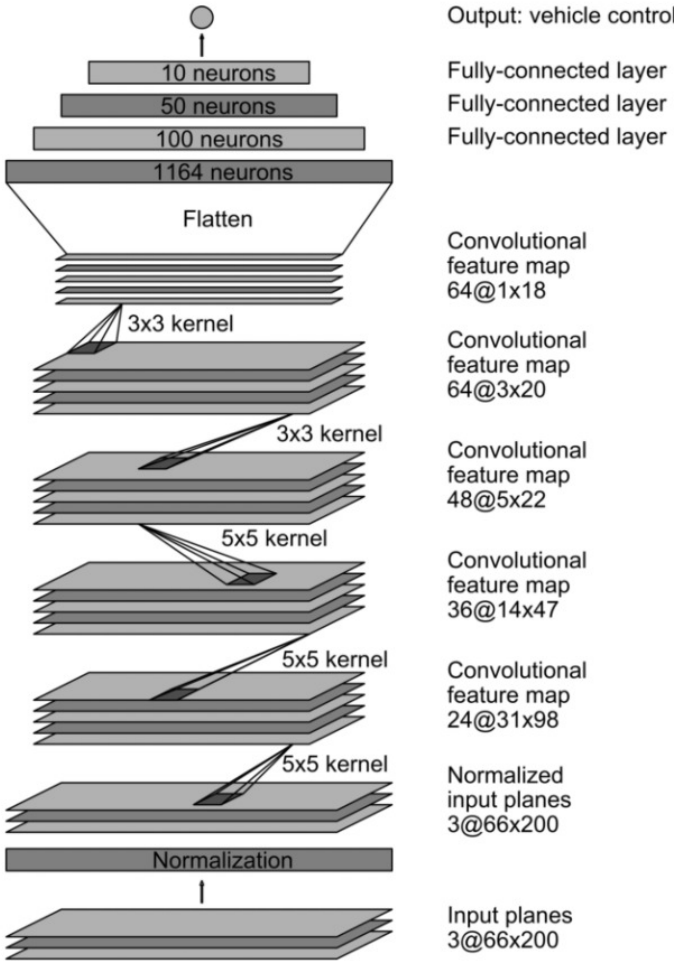


Fig 1. Nvidia CNN architecture. The network has about 27 million connections and 250 thousand parameters.

Based on the Nvidia CNN architecture, above the architecture of the project that was build. The final model is showed below in Fig. 2 and Fig. 3:

```
model.add(Lambda(lambda x: x/127.5 - 1, input_shape=(160, 320,3)))
model.add(Cropping2D(cropping=((70,20),(0,0))))
```

Fig 2. Data is normalized and cropped in the model using a Keras lambda layer.

```
# NVIDIA
model.add(Convolution2D(24, 5, 5, subsample=(2,2), activation="relu"))
model.add(Convolution2D(36, 5, 5, subsample=(2,2), activation="relu"))
model.add(Convolution2D(48, 5, 5, subsample=(2,2), activation="relu"))
model.add(Convolution2D(64, 3, 3, activation="relu"))
model.add(Convolution2D(64, 3, 3, activation="relu"))
model.add(Flatten())
model.add(Dense(1164))
model.add(Dense(100))
model.add(Dropout(0.6))
model.add(Dense(50))
model.add(Dense(10, activation="relu"))
model.add(Dense(1))
```

Fig 3 - Layers of the CNN.

As we can see on Fig. 2 the images were normalized and cropped using Keras lambda layer. The cropping is a good approach because in the top and the bottom of all images there are irrelevant information that undermines the model, making it more difficult to generalize. The landscape as edifices, trees, and mountains do not add value to the model.

The model contains one dropout layer in order to reduce overfitting with a rate of 60% and was trained and validated on different data sets to ensure that the model was not overfitting and it was observed that 3 epochs were enough for training the model. More than that the validation accuracy does not decrease.

3. Model parameter tuning

The model used an “Adam” optimizer with learning rate tuned manually. After several attempts an initial learning rate of 0.0001 achieved good results. I mapped all parameters that I could tune and this approach works fine as the table I shows.

The parameters to tuning are the near angle to zero that part data should be deleted, the percentage of the data with measurements near to zero that should be deleted, the number of epochs, the dataset used and the learning rate.

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. To capture good driving behavior, I first recorded two laps on track one using center lane driving. Here is an example image of center lane driving:

Fig. 4 - Example image of center lane driving

Then two laps were recorded with the vehicle along the lap in a zigzag course even in the curves, alternating the direction abrupt for right and left turns. For the first training session, I got better results with this approach. Only for the next sessions, I used the approach of recording only recovering from the left side and right sides of the road back to center with the start record angle always very large, for instance, as angles with magnitude bigger than 8 degrees and very slow speed. This recovering approach was emphasized in the points where the car does not behave properly.

The images below show what a recovery looks like starting from the right side of the road:

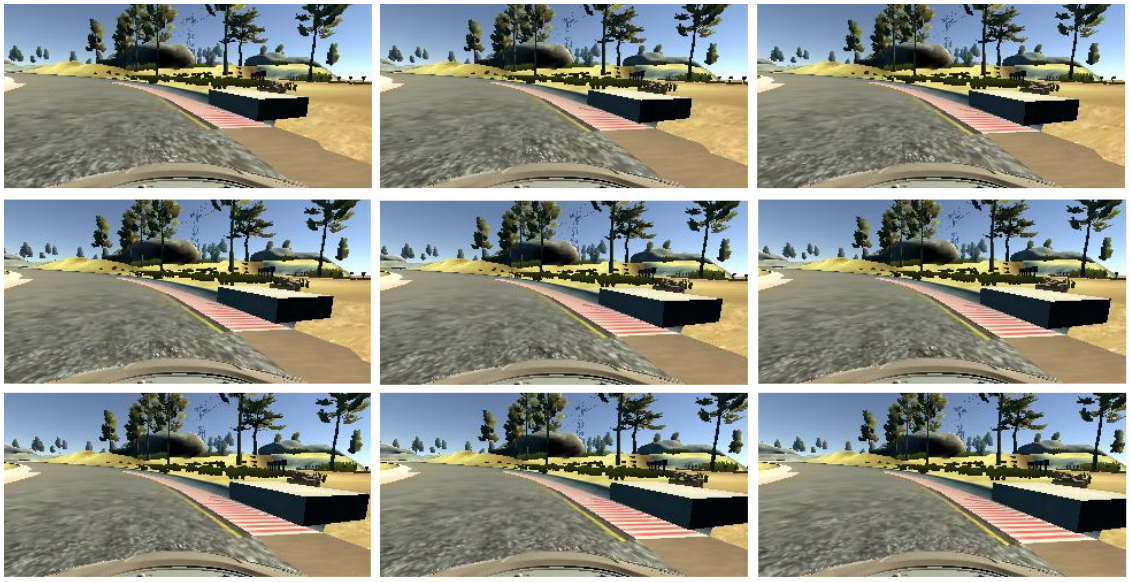
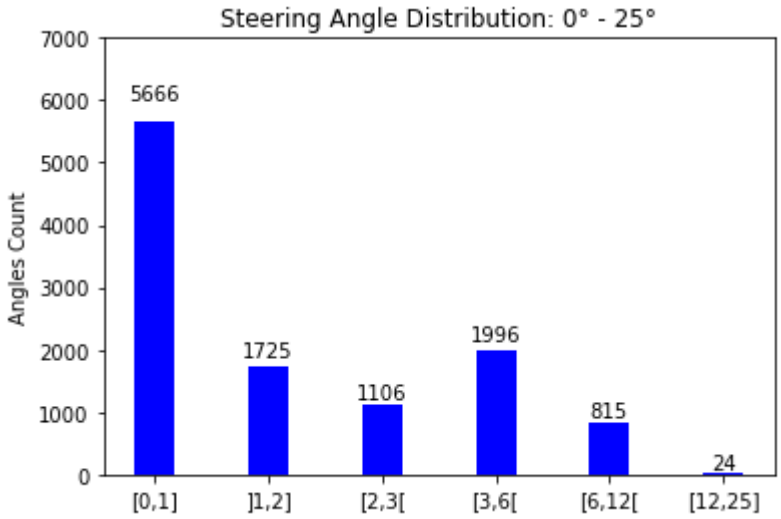


Fig. 5 – Recovering from the right side of the road.

To complete the first dataset I made one more lap with slowly focusing on a "perfect curve behavior", driving with speed around 5 to 7 mph.

DATA REDUCTION

Analyzing the angles measurements collected in the first dataset we observe that the angles from zero to two degrees make up the vast majority of measures. The consequences of it will be a bias associated with going straight and the vehicle will have the difficulty of making sharp curves. In this way, was necessary to remove the most part of data. This one parameter that was tuned according to the Table I.



DATA AUGMENTATION

To augment the dataset, all images generated after the training were flipped and her related angles were inverted to avoid this time an bias to the left due the circuit is kind “circular”.

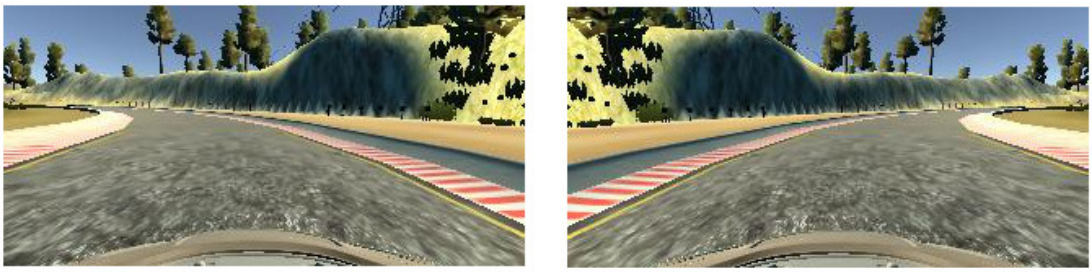


Image captured by the center camera

Flipped image from the center camera

Fig. 6 – Flipped image technique.

START MODEL

The better approach that worked for this project is a start model that at least would be able to pass in 3 curves. I got a good start model after 10 attempts and I put in a table all parameters that I can tune one for each time.

	Attempts	Parameters					Results			
		near	% to delete	epochs	Dataset	LR	1º curve	2º curve	3º curve	4º curve
First Training	1	0.08	70%	3	1	0,0001	ok	No	-	-
	2	0.08	70%	3	2	0,0001	ok	No	-	-
	3	0.08	70%	3	3	0,0001	ok	No	-	-
	4	0.15	70%	3	3	0,0001	ok	No	-	-
	5	0.08	85%	3	3	0,0001	ok	ok!!!!	no	-
	6	0.08	85%	3	4	0,0001	ok	No	-	-
	7	0.08	90%	3	3	0,0001	ok	ok!!!!	no	-
	8	0.08	95%	3	3	0,0001	Ok	No	-	-
	9	0.08	95%	3	4	0,0001	No	No	-	-
	10	0.06	85%	3	3	0,0001	ok	ok!!!!	ok!!!!	ok!!!!
Second Training	11	0.06	85%	3	5	0,0001	ok	No		
	12	0.06	90%	3	6	0,0001	ok	ok	ok	ok
	13	0.06	95%	3	6	0,0005	ok	ok	No	-
	14	0.06	95%	3	6	0,00001	ok	ok	ok	ok
	15	0.06	95%	3	7	0,00001	ok	ok!!!!	ok!!!!	ok!!!!

Table I - Parameter tuning register and attempts to training the vehicle properly.

For each attempt I save the model and the dataset. The first and second training have all new images.

First Training

With this start model, I finished the first training with the vehicle able to go through all the lap, but with moments the car did not stay in the middle of the road. The dataset 3 of the attempt 10 has 12786 samples.

Second Training

I tried making some improvements as keep the car in the center of the road and better the performaces on curves. I decided delete 95% of the data with angles less than 1.5 degrees enabling the car to make sharp curves. The vehicle completed well the lap without touch the yellow lanes.

CONCLUSION

The car could make all circuit safely without touch any lane of the road during the test. The speed was set to 15 mph in the final and the car performs well. The car was tested till a speed of 20 mph and still performs well.