Behavioural Cloning Project Write up

Files Submitted & Code Quality

My project includes all the default python files provided in the beginning of the task which are helping to fulfil the task but also I have attached these certain files:

- train.py which preprocesses the data and trains the Convolutional Neural Network (CNN).
- model.h5 and model.json which contains the trained CNN
- Behavioural Cloning Project Write up.pdf which potentially is the file which I am currently writing.
- run1.mp4 is the video of the car driving autonomously.

By using the python drive.py model.h5 and the simulation software provided by Udacity, my model performance could be tested and the car should drive autonomously within the first track. Not fast but steady. I am explaining the speed of driving also with the speed I was able to drive the car while recording the data.

One of the scripts which lead to the car driving autonomously and my main contribution coding wise is the train.py where I have extracted the recorder data from my initial manual drive and I have preprocessed the data towards CNN training, validation data and finally fitting and saving the model. I have included comments but certainly more comments could be provided.

Model Architecture and Training Strategy

I have used as suggested the NVIDIA convolutional network with the following architecture which seems to be a great architecture for autonomous car driving environment.

Layer (type)	Output Shape	Param #
lambda_1 (Lambda)	(None, 160, 320, 3)	0
cropping2d_1 (Cropping2D)	(None, 65, 320, 3)	0
conv2d_1 (Conv2D)	(None, 33, 160, 24)	1824
activation_1 (Activation)	(None, 33, 160, 24)	0
max_pooling2d_1 (MaxPooling2	(None, 32, 159, 24)	0
conv2d_2 (Conv2D)	(None, 16, 80, 36)	21636
activation_2 (Activation)	(None, 16, 80, 36)	0
max_pooling2d_2 (MaxPooling2	(None, 15, 79, 36)	0
conv2d_3 (Conv2D)	(None, 8, 40, 48)	43248

```
_________________________________________________________________
activation_3 (Activation)    (None, 8, 40, 48)         0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 39, 48)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 7, 39, 64)         27712
_________________________________________________________________
activation_4 (Activation)    (None, 7, 39, 64)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 38, 64)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 6, 38, 64)         36928
_________________________________________________________________
activation_5 (Activation)    (None, 6, 38, 64)         0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 5, 37, 64)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 11840)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1164)              13782924
_________________________________________________________________
activation_6 (Activation)    (None, 1164)              0
_________________________________________________________________
dense_2 (Dense)              (None, 100)               116500
_________________________________________________________________
activation_7 (Activation)    (None, 100)               0
_________________________________________________________________
dense_3 (Dense)              (None, 50)                5050
_________________________________________________________________
activation_8 (Activation)    (None, 50)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                510
_________________________________________________________________
activation_9 (Activation)    (None, 10)                0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 11
=================================================================
```

Model parameter tuning

The model used an adam optimizer, so the learning rate was not tuned manually which could be seen on line 135 from the train.py script.

Appropriate training data

I have divided my data to training and validating data where my validation data was 20 % of all the used images.

Model Architecture and Training Strategy

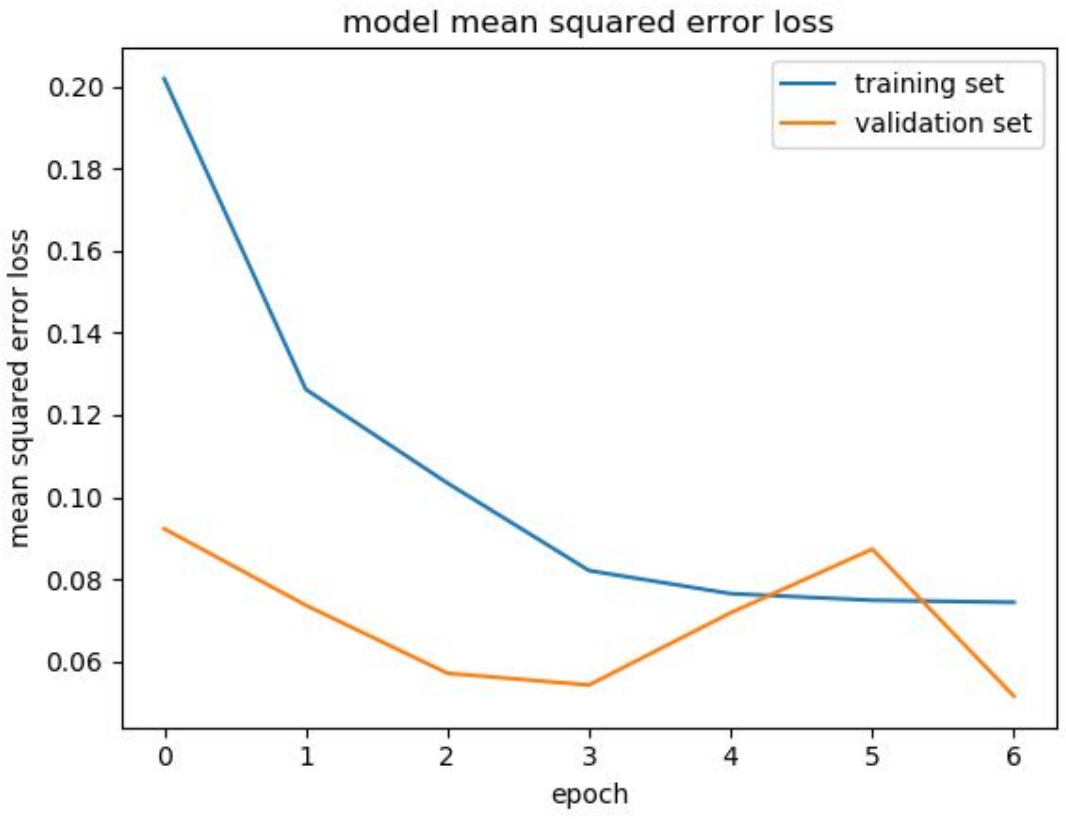
1. Solution Design Approach

As a first step I have looked at the data that I was able to use and more particular the centre, left and right images and also the steering angles. Based on that information I needed to train a CNN to drive simultaneously. My first strategy was to ‘normalise’ the data as the

steering angle distribution was uneven or with other words most of the data was coming from near zero steering angle (the values for steering angles were already normalised). I have mainly reduced angles smaller than 0.1 and -0.1. That has also deducted my data approximately 4 times.

My next step was to pre-process the data and more specifically I have played with the brightness of the images randomly within some limits. But also I have flipped randomly images and reverse their steering vector.

The following step was to define the architecture of the convolutional neural network. There I have chosen to use the CNN that has been proposed by Nvidia. I believe that is a great CNN fitting the purposes of training a vehicle to drive autonomously. I have divided my data to training (5705 images) and 20% of validating (634 images). The Model mean squared error loss from the training and validation data has been presented in the figure below for the 7 epochs that I have used.



Creation of the Training Set & Training Process

To capture good driving behavior, first I recorded a lap on my own but my driving skills were hurrendes so I made the decision to use the driving data provided by Udacity as presented below.

I have added the left and right images as well with a +- 20 degrees correction to the steering angle. Some the other pre-processing steps were discussed in the aforementioned pages.

The training process took not long time and showed good results as shown previously in the document.

Testing Process

I have tested my model by driving in autonomous mode in the simulation software with the first track. I have recorded and attached the drive called run1. The car has drove steadily for 2 laps within the street. The only drawback that I would say is that the car is driving particularly slowly.

Future Work

Work more over the potential overfitting issues of the model. Would test and make sure it works with the other track in autonomous mode.