**Udacity Self-Driving Car Engineer**

**Report**

* **Project:**  Behavioural Cloning
* **Name:**  Shao Hongxu

**Let me just use some brief sentence to explain the thing I did, basically these are 6 big parts here:**

**Part 1:** import necessary lib

**Part 2:** using training data generator function (read data real time while testing & spilt training & validation data)

**Part 3:** collect training data (most important part for this project)

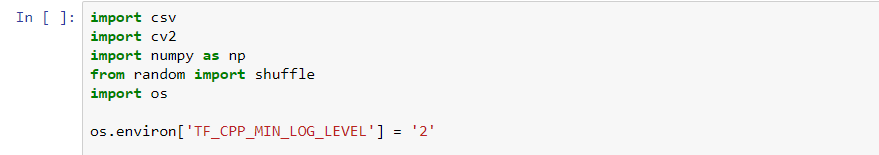
**Part 4:** identify model

**Part 5:** use trained model to run simulation in Auto- mode.

**Part 6:** Summary

**To be specific as below, you will find some note on Script, key take away and Area could be improved:**

**Part 1:** import necessary lib:



**Part 2:** using training data generator function (read data real time while testing & spilt training & validation data)



**-> Key take away:** This is a beautiful function especially for situation of a big data but a normal computing memory. Learnt from Udacity, it makes few Machine learning possible for me to have a try. (I have only one NVIDIA GTX 1080 with 8GB memory)

In the end, I select **batch size** as 32, which suits my GPU not running more than 50%.

**Part 3:** collect training data (most important part for this project)

1. Track one - one complete round. (forward) -> keep the car on the middle line
2. Track one - one complete round. (backward) -> keep the car on the middle line
3. Track one – collect data on scenario of car moving from one side into middle line. (most important data)
4. Training data provided by project
5. Track two - one complete round. (forward) -> keep the car on the middle line
6. Track two - one complete round. (backward) -> keep the car on the middle line

After all these data, still car sometime driving out of road at the point of road side next to lake & desert. After that, I was collecting a group of additional training data specific at the point of road side next to lake & desert. And replicate the data for these scenarios 4 time and put it into the training data. Finally, it works in the end.

**Part 4:** identify model



Machine learning model structure:

* Input layer: 160\*320\*3
* 1st layer: cropping layer, to crop the image into area needed.
* 2nd layer: Convolution2D layer, 24, 5, 5
* 3rd layer: Convolution2D layer, 36, 5, 5
* 4th layer: Convolution2D layer, 48, 5, 5
* 3rd layer: Convolution2D layer, 64, 5, 5
* 4th layer: Convolution2D layer, 64, 5, 5 , with 0.5 dropout
* 5th layer: flatten layer
* 6th layer: 100 unit, with 0.1 dropout
* 7th layer: 50 unit, with 0.1 dropout
* 8th layer: 10 unit
* Output layer: 1

**Learning rate parameters** is Adam optimizer for this model.

**Dropout** (0.5, 0.1) has been selected to protect overfitting.

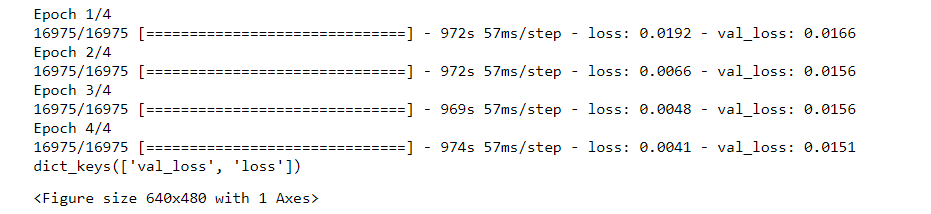
In the end, only 4 epochs give a good enough result for simulation track one.

Training & validation set has been spilt into 80% and 20%

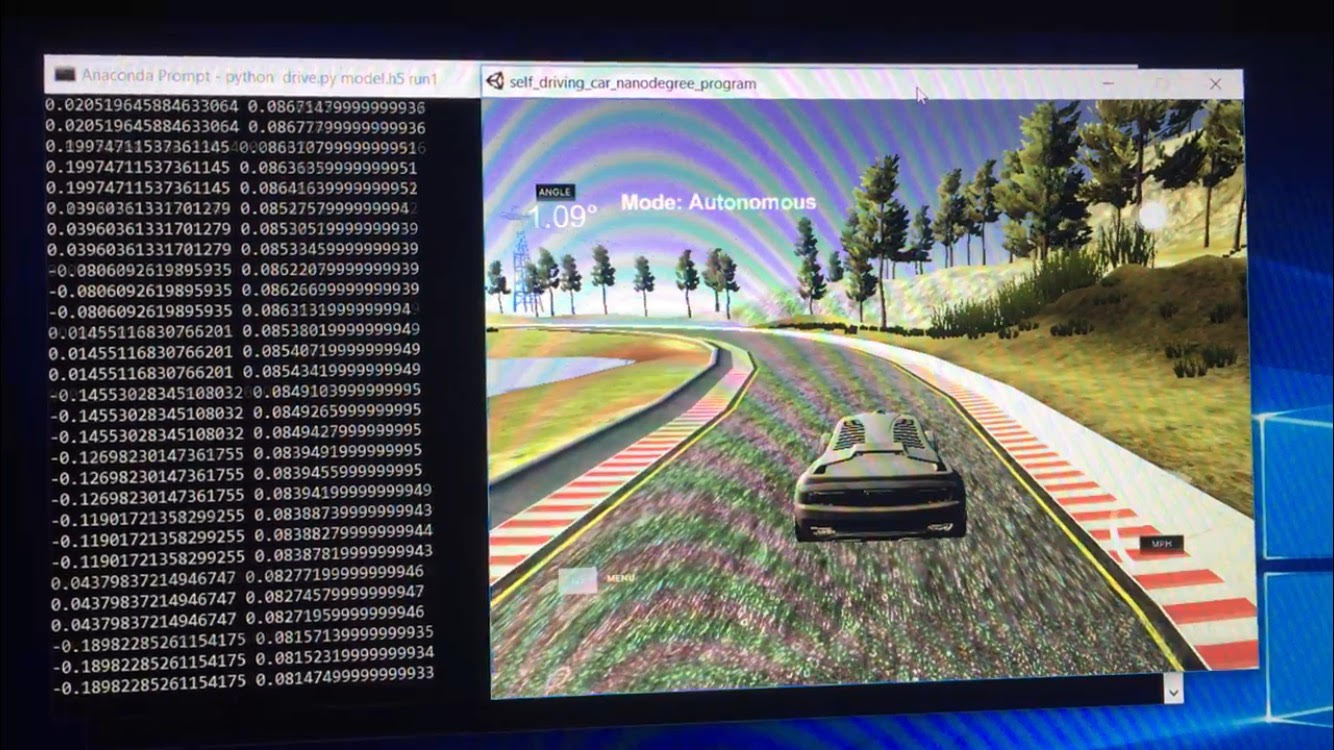
**-> Key take away:** dropout is needed, I tried model without dropout, validation set loss never decreased.

**-> Area could be improved:** last 3 layers (100, 50, 1) architecture is slightly simple, maybe it can be improved a little bit to make it more intelligent.

**-> Result:**



**Part 5:** use trained model to run simulation in Auto-mode.



In the end, car is able to run for one complete round in track one. You can check video attached in detail.

**Part 6:** Summary

Personally, I really like this cloning project very much. It brings me the sense of the most important area for machine learning/ deep learning project. Most of the time, for a machine learning competition, data has been provided. In that case, as a individual, most of the time we even couldn’t start the project from data collection.

And project make me feel about the improvement from data you put in addition. That is really great!