CarND: Behavioral Cloning

Model Architecture and Training Strategy

1. Model architecture

I used the same CNN model Nvidia introduced for end-to-end deep learning. It first batch normalizes the input and crop the upper half of the image that are not necessary for our task. Then, the following 5 layers are convolutional layers (feature map) that use 2x2, 2x2, 2x2, 1x1, 1x1 filter size respectively. In the end, there are 4 dense layers which have length of 100, 50, 10, and 1 since we only output one value. All conv layers use RELU activation to introduce nonlinearity.

2. Attempts to reduce overfitting in the model

First of all, I applied data augmentation, which randomly flip the vehicle horizontally and negate the measurement. I also randomly used the left camera and the right camera instead of using the centre photo all the time. The correction I applied is 0.1 for right shot and -0.1 for left photo.

3. Model parameter tuning

The model is compiled with mean square error loss function and adam optimizer. The learning rate is the default value of adam.

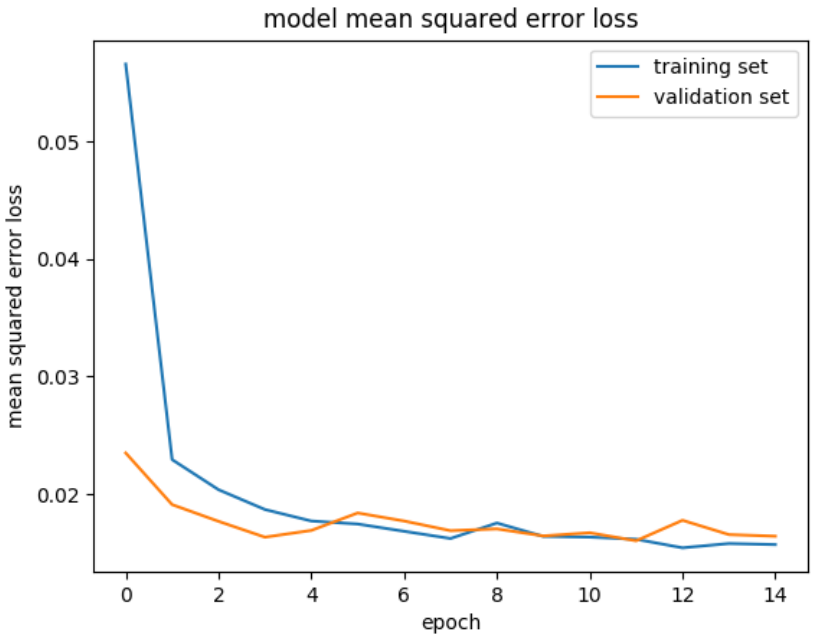
4. Creation of dataset

I ran the simulator vehicle for two laps in the training mode to increase amount of data. To ensure model performs well, I tried to drive the vehicle on the centre of the road.

I implemented a data generator to avoid loading all dataset into memory. The generator also randomly shuffled the dataset for each batch.

5. Training

Below is a plot of loss.



After training, I run the autonomous mode using the model I trained. It finish the lap greatly.