# Traffic Sign Recognition

### Model Architecture and Training Strategy

#### 1. Solution Design Approach

(1) At first, I only used the center data, I wanted to build my model as a single layer deep neural network before I started adding convolutions and layers. I worked on the pipeline and generating images from the drive log first.

(2) I was sure that the pipeline was in a good place, I added a convolution layer to the model.

(3) With the model above, I was able to complete just one small part. Then I added the left images and right images.

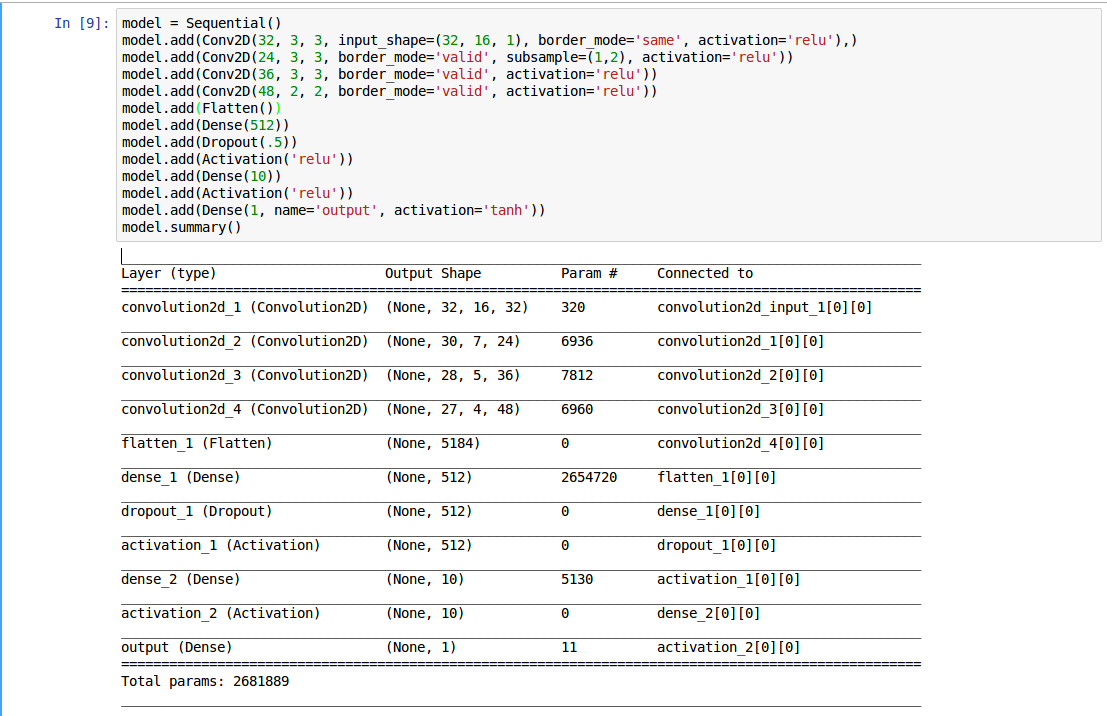
(4) Then I started out with a more complex model, but I found that this took much longer to train and didn't produce much better results, so I pruned down the model for simplicity and speed, while still keeping accuracy high.

(5) At last, I trained more recovery data, then used them to train the model.

At the end of the process, the vehicle is able to drive autonomously around the track without leaving the road.

#### 2. Final Model Architecture

The final model architecture consisted of a convolution neural network with the following layers and layer sizes:



For the model architecture

(1) My model consists of a convolution neural network with 3x3 filter sizes and depths between 32 and 48. I chose four convolutional layers to extract features from the camera images, followed by four fully-connected layers to . The last two layer are activated by a ReLU, while the last layer is activated by tanh to keep the prediction angles between -1 and 1.

(2) I introduced two dropout layers - one after the first four convolutional layers and one after the second two - to reduce the possibility of overfitting.

(3) I optimized the model with an Adam optimizer over MSE loss.

(4) For hyperparameters, I chose a learning rate of 0.0001 rather than the default adam optimizer rate of 0.001. because I found that the higher rate plateaued at a higher loss and produced worse qualitative driving results. I trained for 5 epochs because performance increase started diminishing when training for longer.

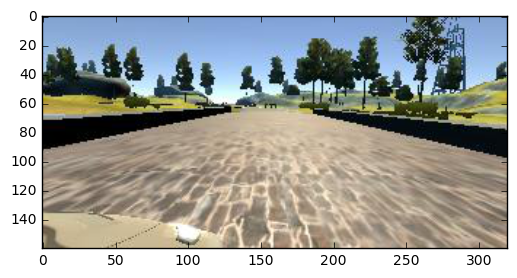
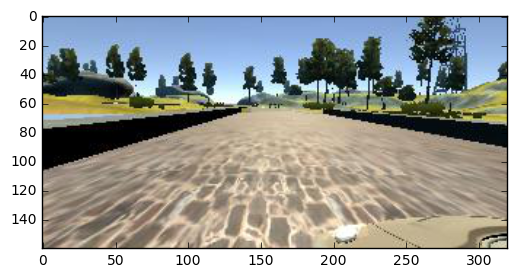
#### 3. Creation of the Training Set & Training Process

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.

Here is an example image of center lane driving:



I then display the vehicle recovering from the left side and right sides of the road.



After the collection process, I had X number of data points. I then preprocessed this data according to:

(1)Resize the image to (32, 16, 3)

(2)Convert image to grayscale

(3)Normalize the image

I finally randomly shuffled the data set and put 0.05% of the data into a validation set. The model was trained and validated on different data sets to ensure that the model was not overfitting.

The left, right and center images are used in training. I've added 0.08 to the left images and -0.08 to the right images to compensate for the difference in camera angle.