Behavioral Cloning

Video of project: https://www.youtube.com/watch?v=KpcbalVUCUA

Behavioral Cloning Project

The goals / steps of this project are the following:
* Use the simulator to collect data of good driving behavior
* Build, a convolution neural network in Keras that predicts steering angles from images
* Train and validate the model with a training and validation set
* Test that the model successfully drives around track one without leaving the road
* Summarize the results with a written report

Rubric Points
###Here I will consider the [rubric points](https://review.udacity.com/#!/rubrics/432/view) individually and describe how I addressed each point in my implementation.

###Files Submitted & Code Quality

####1. Submission includes all required files and can be used to run the simulator in autonomous mode

My project includes the following files:
* model.py containing the script to create and train the model
* drive.py for driving the car in autonomous mode
* v2model.h5 containing a trained convolution neural network
* writeup_report.md or writeup_report.pdf summarizing the results

####2. Submission includes functional code
Using the Udacity provided simulator and my drive.py file, the car can be driven autonomously around the track by executing
```sh
python drive.py model.h5 imgs
```

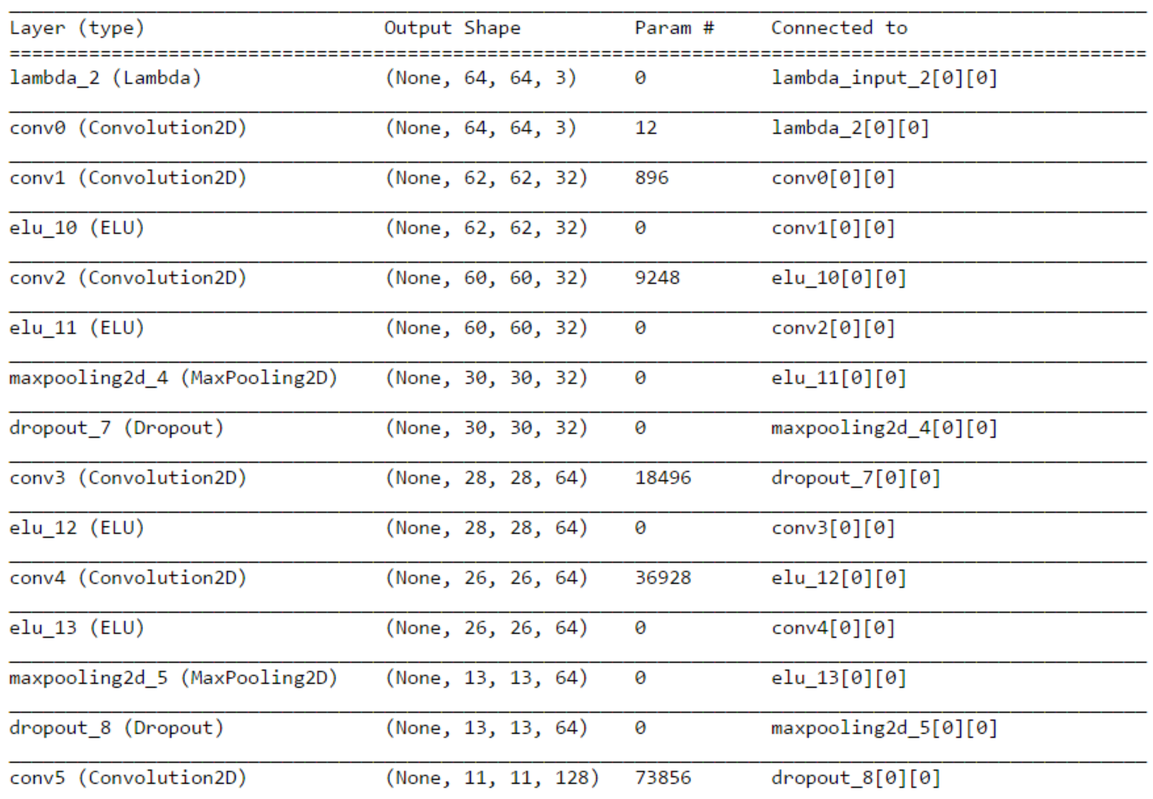
####3. Submission code is usable and readable

The model.py file contains the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

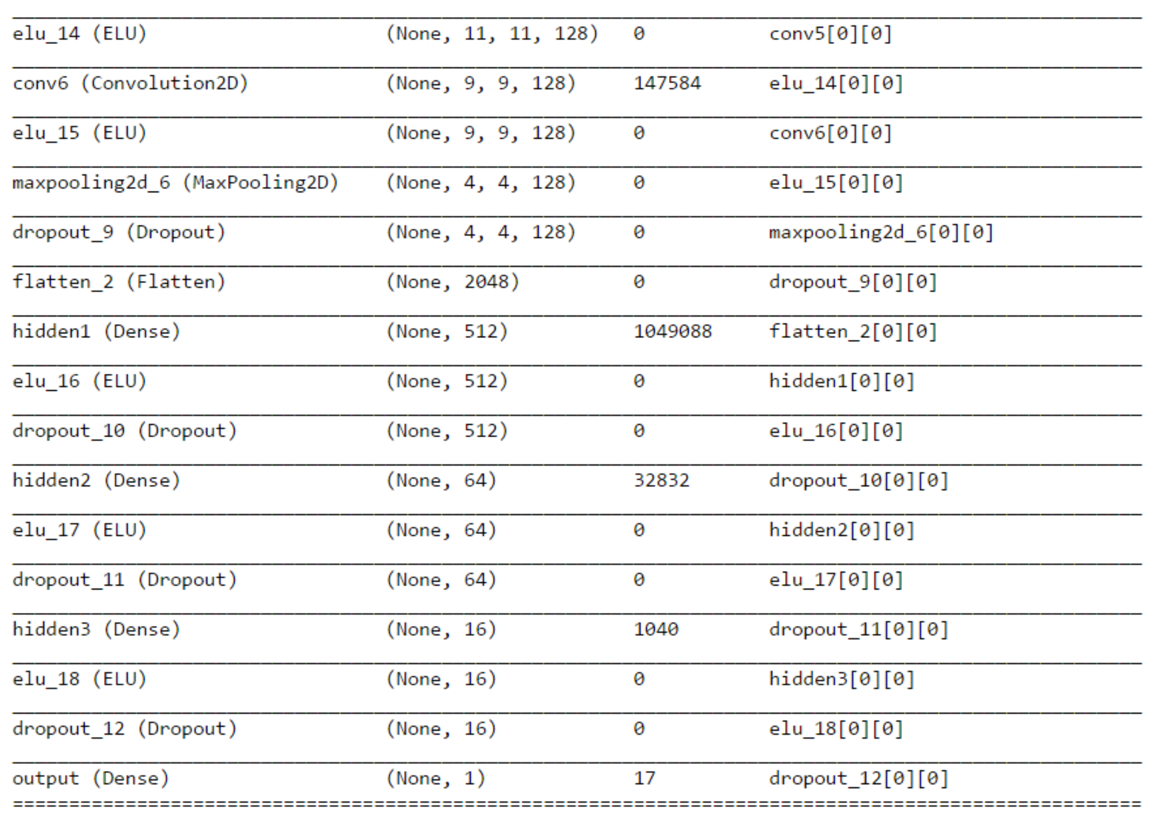
###Model Architecture and Training Strategy

####1. An appropriate model architecture has been employed

(code line 24).



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
lambda_2 (Lambda)                (None, 64, 64, 3)     0           lambda_input_2[0][0]
____________________________________________________________________________________________________
conv0 (Convolution2D)            (None, 64, 64, 3)     12          lambda_2[0][0]
____________________________________________________________________________________________________
conv1 (Convolution2D)            (None, 62, 62, 32)    896         conv0[0][0]
____________________________________________________________________________________________________
elu_10 (ELU)                     (None, 62, 62, 32)    0           conv1[0][0]
____________________________________________________________________________________________________
conv2 (Convolution2D)            (None, 60, 60, 32)    9248        elu_10[0][0]
____________________________________________________________________________________________________
elu_11 (ELU)                     (None, 60, 60, 32)    0           conv2[0][0]
____________________________________________________________________________________________________
maxpooling2d_4 (MaxPooling2D)    (None, 30, 30, 32)    0           elu_11[0][0]
____________________________________________________________________________________________________
dropout_7 (Dropout)              (None, 30, 30, 32)    0           maxpooling2d_4[0][0]
____________________________________________________________________________________________________
conv3 (Convolution2D)            (None, 28, 28, 64)    18496       dropout_7[0][0]
____________________________________________________________________________________________________
elu_12 (ELU)                     (None, 28, 28, 64)    0           conv3[0][0]
____________________________________________________________________________________________________
conv4 (Convolution2D)            (None, 26, 26, 64)    36928       elu_12[0][0]
____________________________________________________________________________________________________
elu_13 (ELU)                     (None, 26, 26, 64)    0           conv4[0][0]
____________________________________________________________________________________________________
maxpooling2d_5 (MaxPooling2D)    (None, 13, 13, 64)    0           elu_13[0][0]
____________________________________________________________________________________________________
dropout_8 (Dropout)              (None, 13, 13, 64)    0           maxpooling2d_5[0][0]
____________________________________________________________________________________________________
conv5 (Convolution2D)            (None, 11, 11, 128)   73856       dropout_8[0][0]
```



```
____________________________________________________________________________________________________
elu_14 (ELU)                     (None, 11, 11, 128)   0           conv5[0][0]
____________________________________________________________________________________________________
conv6 (Convolution2D)            (None, 9, 9, 128)     147584      elu_14[0][0]
____________________________________________________________________________________________________
elu_15 (ELU)                     (None, 9, 9, 128)     0           conv6[0][0]
____________________________________________________________________________________________________
maxpooling2d_6 (MaxPooling2D)    (None, 4, 4, 128)     0           elu_15[0][0]
____________________________________________________________________________________________________
dropout_9 (Dropout)              (None, 4, 4, 128)     0           maxpooling2d_6[0][0]
____________________________________________________________________________________________________
flatten_2 (Flatten)              (None, 2048)          0           dropout_9[0][0]
____________________________________________________________________________________________________
hidden1 (Dense)                  (None, 512)           1049088     flatten_2[0][0]
____________________________________________________________________________________________________
elu_16 (ELU)                     (None, 512)           0           hidden1[0][0]
____________________________________________________________________________________________________
dropout_10 (Dropout)             (None, 512)           0           elu_16[0][0]
____________________________________________________________________________________________________
hidden2 (Dense)                  (None, 64)            32832       dropout_10[0][0]
____________________________________________________________________________________________________
elu_17 (ELU)                     (None, 64)            0           hidden2[0][0]
____________________________________________________________________________________________________
dropout_11 (Dropout)             (None, 64)            0           elu_17[0][0]
____________________________________________________________________________________________________
hidden3 (Dense)                  (None, 16)            1040        dropout_11[0][0]
____________________________________________________________________________________________________
elu_18 (ELU)                     (None, 16)            0           hidden3[0][0]
____________________________________________________________________________________________________
dropout_12 (Dropout)             (None, 16)            0           elu_18[0][0]
____________________________________________________________________________________________________
output (Dense)                   (None, 1)             17          dropout_12[0][0]
====================================================================================================
```

####2. Attempts to reduce overfitting in the model

The model contains dropout layers in order to reduce overfitting (code cell 24). drop = 0.5

The model was trained and validated on different data sets to ensure that the model was not overfitting (code cell 25).

I also used
   callbacks=[ModelCheckpoint(filepath="v2model.h5", verbose=1, save_best_only=True)])
with save best only to reduce overfitting

####3. Model parameter tuning

The model used an adam optimizer, so the learning rate was not tuned manually (cell 24).

####4. Appropriate training data

Training data was chosen to keep the vehicle driving on the road. I used Udacity data only For details about how I created the training data, see the next section.

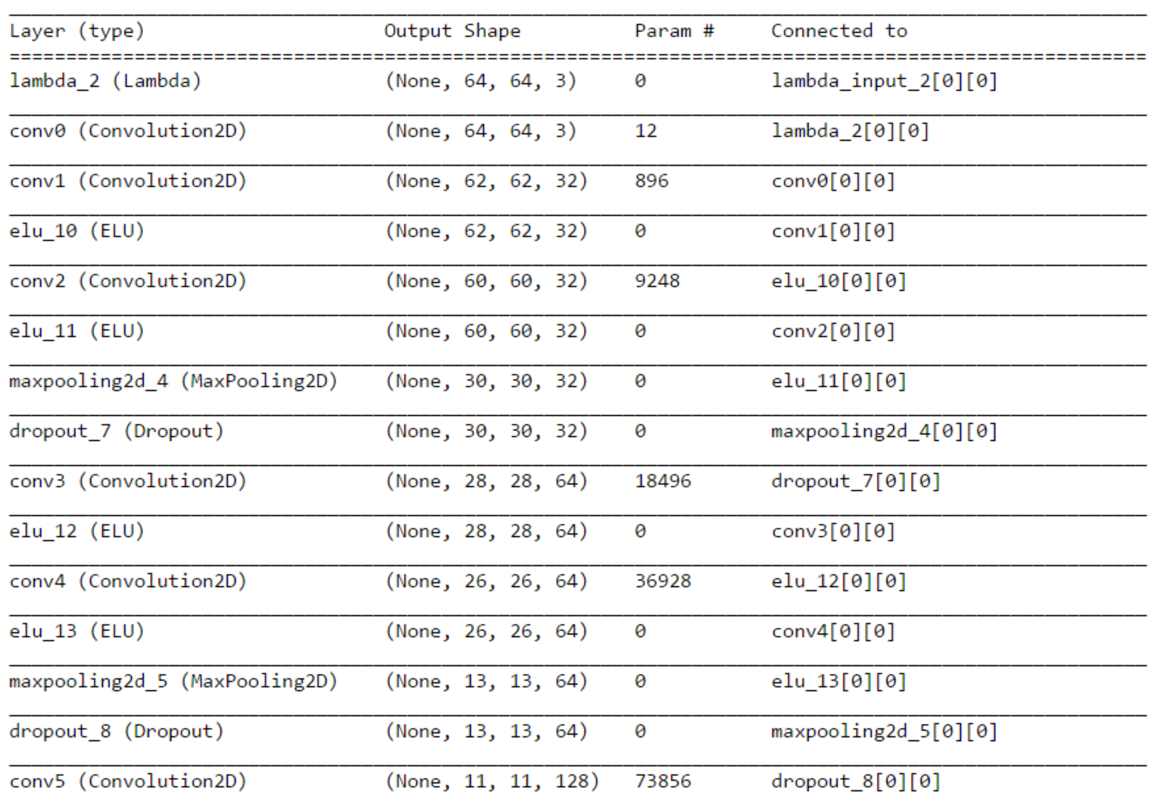
###Model Architecture and Training Strategy

####1. Solution Design Approach

I tried vgg and project 2 archetecture but not get good results so I used this archetecture from Vivek Yadav post

The first layer is 3 1X1 filters, this has the effect of transforming the color space of the images.. As we do not know the best color space apriori, using 3 1X1 filters allows the model to choose its best color space. This is followed by 3 convolutional blocks each comprised of 32, 64 and 128 filters of size 3X3. These convolution layers were followed by 3 fully connected layers. All the convolution blocks and the 2 following fully connected layers had exponential relu (ELU) as activation function. I chose leaky relu to make transition between angles smoother.

####2. Final Model Architecture



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
lambda_2 (Lambda)                (None, 64, 64, 3)     0           lambda_input_2[0][0]
____________________________________________________________________________________________________
conv0 (Convolution2D)            (None, 64, 64, 3)     12          lambda_2[0][0]
____________________________________________________________________________________________________
conv1 (Convolution2D)            (None, 62, 62, 32)    896         conv0[0][0]
____________________________________________________________________________________________________
elu_10 (ELU)                     (None, 62, 62, 32)    0           conv1[0][0]
____________________________________________________________________________________________________
conv2 (Convolution2D)            (None, 60, 60, 32)    9248        elu_10[0][0]
____________________________________________________________________________________________________
elu_11 (ELU)                     (None, 60, 60, 32)    0           conv2[0][0]
____________________________________________________________________________________________________
maxpooling2d_4 (MaxPooling2D)    (None, 30, 30, 32)    0           elu_11[0][0]
____________________________________________________________________________________________________
dropout_7 (Dropout)              (None, 30, 30, 32)    0           maxpooling2d_4[0][0]
____________________________________________________________________________________________________
conv3 (Convolution2D)            (None, 28, 28, 64)    18496       dropout_7[0][0]
____________________________________________________________________________________________________
elu_12 (ELU)                     (None, 28, 28, 64)    0           conv3[0][0]
____________________________________________________________________________________________________
conv4 (Convolution2D)            (None, 26, 26, 64)    36928       elu_12[0][0]
____________________________________________________________________________________________________
elu_13 (ELU)                     (None, 26, 26, 64)    0           conv4[0][0]
____________________________________________________________________________________________________
maxpooling2d_5 (MaxPooling2D)    (None, 13, 13, 64)    0           elu_13[0][0]
____________________________________________________________________________________________________
dropout_8 (Dropout)              (None, 13, 13, 64)    0           maxpooling2d_5[0][0]
____________________________________________________________________________________________________
conv5 (Convolution2D)            (None, 11, 11, 128)   73856       dropout_8[0][0]
```

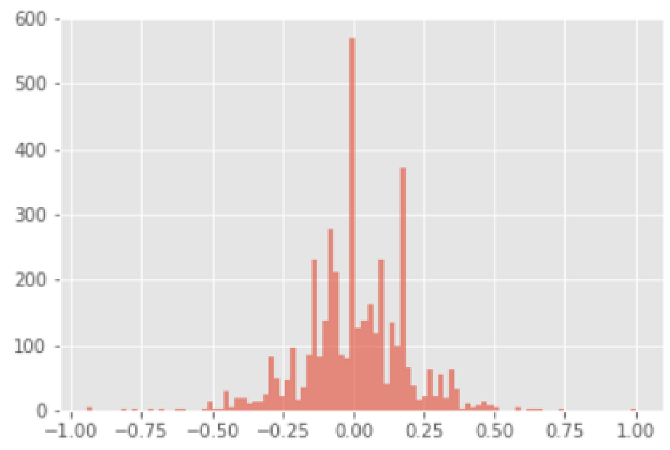
```
____________________________________________________________________________________________________
elu_14 (ELU)                     (None, 11, 11, 128)   0           conv5[0][0]
____________________________________________________________________________________________________
conv6 (Convolution2D)            (None, 9, 9, 128)     147584      elu_14[0][0]
____________________________________________________________________________________________________
elu_15 (ELU)                     (None, 9, 9, 128)     0           conv6[0][0]
____________________________________________________________________________________________________
maxpooling2d_6 (MaxPooling2D)    (None, 4, 4, 128)     0           elu_15[0][0]
____________________________________________________________________________________________________
dropout_9 (Dropout)              (None, 4, 4, 128)     0           maxpooling2d_6[0][0]
____________________________________________________________________________________________________
flatten_2 (Flatten)              (None, 2048)          0           dropout_9[0][0]
____________________________________________________________________________________________________
hidden1 (Dense)                  (None, 512)           1049088     flatten_2[0][0]
____________________________________________________________________________________________________
elu_16 (ELU)                     (None, 512)           0           hidden1[0][0]
____________________________________________________________________________________________________
dropout_10 (Dropout)             (None, 512)           0           elu_16[0][0]
____________________________________________________________________________________________________
hidden2 (Dense)                  (None, 64)            32832       dropout_10[0][0]
____________________________________________________________________________________________________
elu_17 (ELU)                     (None, 64)            0           hidden2[0][0]
____________________________________________________________________________________________________
dropout_11 (Dropout)             (None, 64)            0           elu_17[0][0]
____________________________________________________________________________________________________
hidden3 (Dense)                  (None, 16)            1040        dropout_11[0][0]
____________________________________________________________________________________________________
elu_18 (ELU)                     (None, 16)            0           hidden3[0][0]
____________________________________________________________________________________________________
dropout_12 (Dropout)             (None, 16)            0           elu_18[0][0]
____________________________________________________________________________________________________
output (Dense)                   (None, 1)             17          dropout_12[0][0]
====================================================================================================
```
####3. Creation of the Training Set & Training Process

Augmentation: cell 21

Steering angle s=0 have the highest frequency: more than 20 times larger than the frequency of other angle values. Also, there are more positive than negative angle values.

so I sample 11% only from zero angle data

Angles Distribtion after take 11% of zero angles data only



then used left and right images by create adjusted steering angles for the side camera images

constand used is 0.05 but I tried many things not provide any better result

then Create X ,Y (img,steering) img have all images and steering have all angles

then split (train set and validation set) ,Shuffle train se) and form sets to array

Preprocss functions
Preprocss functions
1- cropresize(img):
Crop non important parts from an image (img) and retain only the track, Then Resize the image to 64*64 dimensions

2- change_colorspace(img, color_space='RGB'):
change from RGB mode to YUV

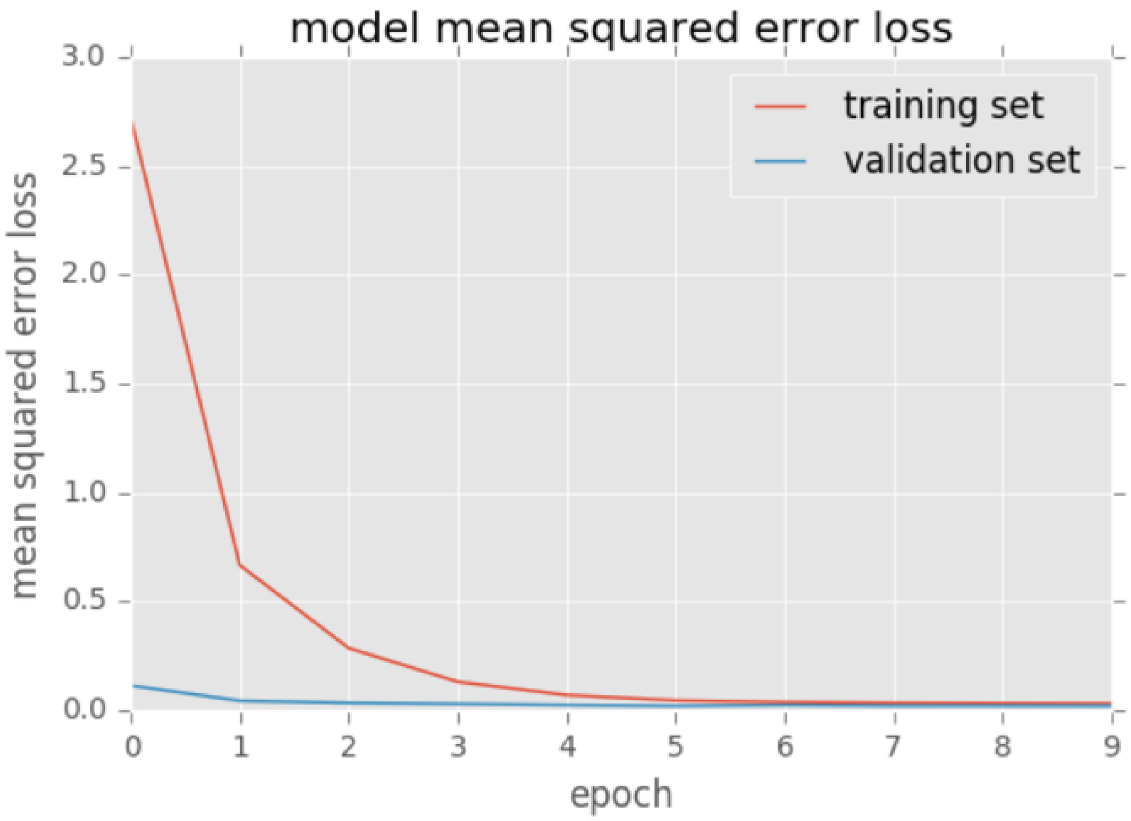
3- flipped_image(img, steering):
flip image (img) horizontally then reverse the steering angle (steering)

4-brightnessed_image(img):
set random brightness to the image (img)

5- translated_image(img, steering):
to generate fair dataset (normal distrebution) generate a new image
from the current image(img) using random shift in horizontal direction.
with change in the steering angle (steering) value is 0.4 per pixel

6- image_augmentation(img, steering):
apply flipped_image with 50% chance then apply Brightness and tranlslate functions

Finally I make function to read and transform (cropresize,change colorspace) images for validation set and function to read and agument and transform for train set. and used generator to pass it to train model.


model mean squared error loss
training set
validation set
mean squared error loss
epoch
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7
8
9