Project 3: Behavioral Cloning

Q1. Are all required files submitted?
A1. Yes files have been submitted alongside this writeup

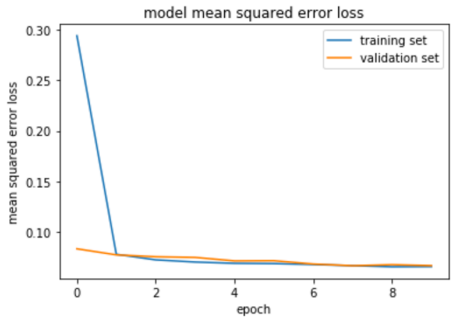
Q2. Is the code functional?
A2. Yes the code was used to train the model to generate the video submitted.

Q3. Is the code usable and readable?
A3. Comments have been included where considered appropriate.

Q4. Has an appropriate model architecture been employed for the task?
A4. The NVIDIA model architecture has been used as the base for this project. Overfitting was not observed in the model.

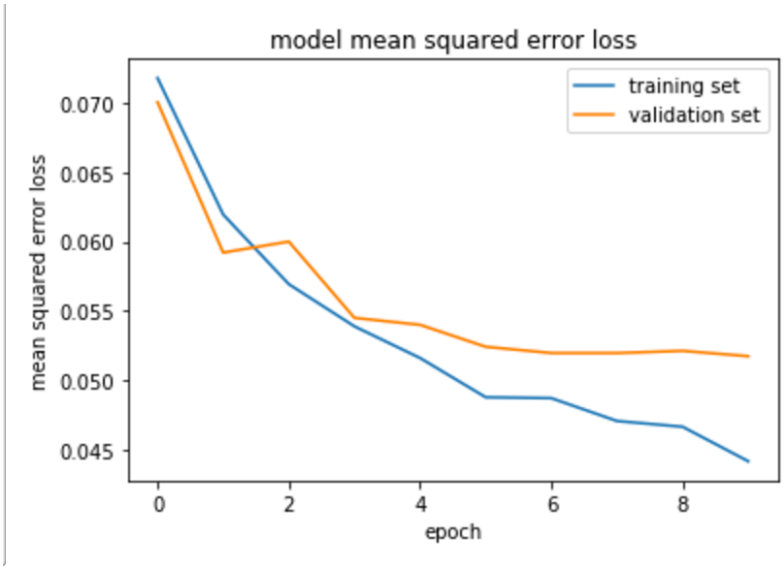
- **Regularization was tried out but it increased the loss.**

```
216/215 [==============================] - 359s - loss: 0.0686 - val_loss: 0.0712
Epoch 6/10
216/215 [==============================] - 359s - loss: 0.0684 - val_loss: 0.0713
Epoch 7/10
216/215 [==============================] - 359s - loss: 0.0674 - val_loss: 0.0681
Epoch 8/10
216/215 [==============================] - 359s - loss: 0.0663 - val_loss: 0.0664
Epoch 9/10
216/215 [==============================] - 359s - loss: 0.0653 - val_loss: 0.0674
Epoch 10/10
216/215 [==============================] - 359s - loss: 0.0654 - val_loss: 0.0665
dict_keys(['val_loss', 'loss'])
```

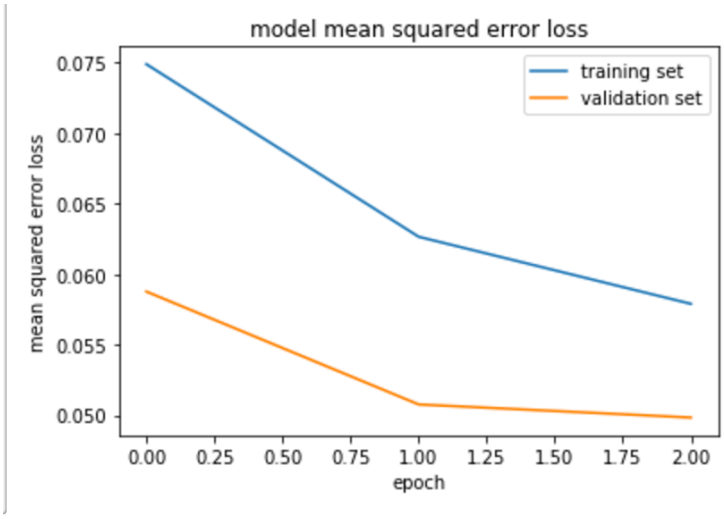


- **Correction parameter for left and right steering angles was tuned by hit and trial. Parameters of .2, .15, .1, .075 were tried. Only stable model was achieved with parameter value of .1. For the other parameters the trained model drove the car off track.**

- **Different Epochs were tried and for higher epochs overfitting was observed as an example see the graph below.**

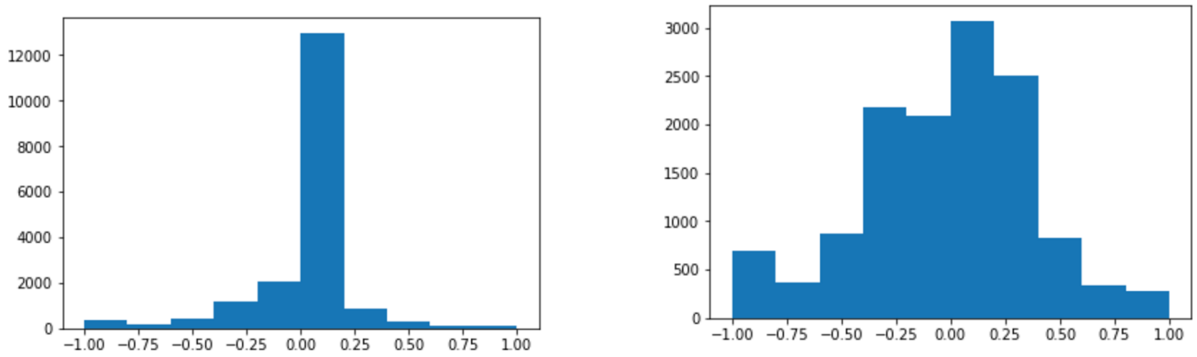


For training the model 3 epochs were used with the loss graph illustrated below.

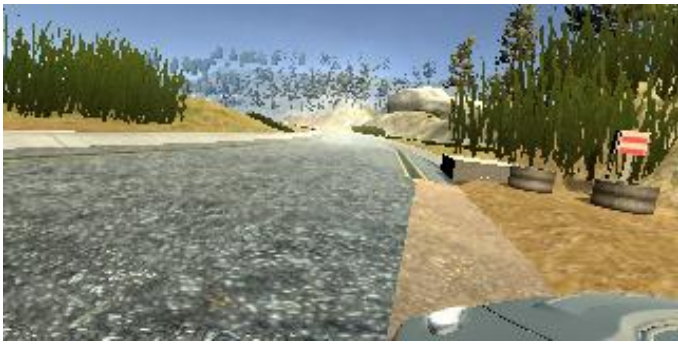


Q5. Has an attempt been made to reduce overfitting of the model?
A5. The training data was split into training and validation data. Regularization was not used for the submitted solution as overfitting was not observed. Regularization was tried with varying parameter values, but it increased the loss/cost function value.

Q6. Is the training data chosen appropriately?
A6. The more training data was collected especially to train for the sharp turns. Training data was also balanced for training examples as the 0 angle images dominated the training data. The balancing removes any bias due the dataset during training. Attempts made to train the model without balancing were unsuccessful. The training data is comprised of the data provided for the project, data collected by driving car in clockwise and counter clockwise around the track, data collected specifically for sharp turns and data collected for recovering if the car drives over an unsafe surface.

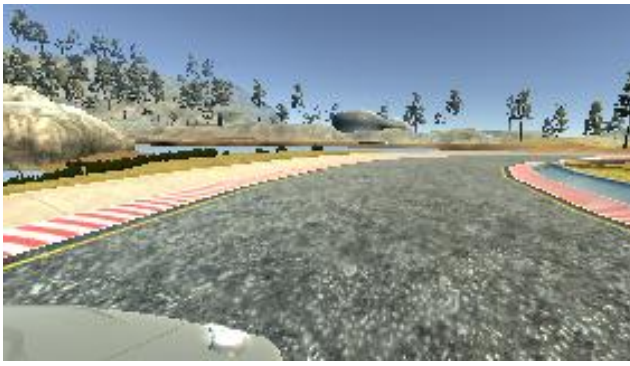


Histograms of training data before and after balancing. Original dataset is on the left. Balanced dataset on the right.



Recovery from off road state above

Recovery from off road state above

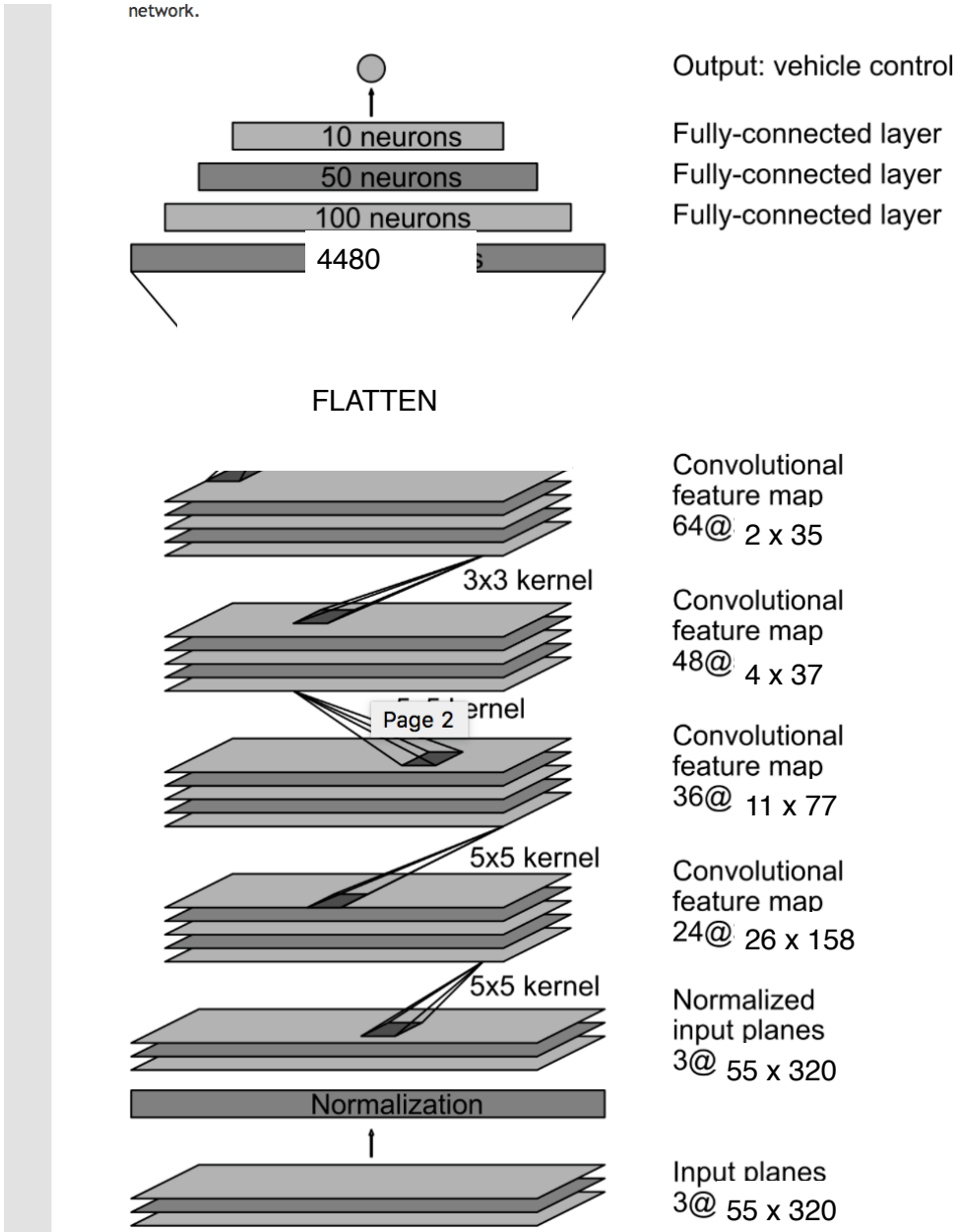


Sharp turn on track above

Q7. Is the solution design documented?
A7. Answers 4, 5, and 6 cover this question.

Q8. Is the model architecture documented?
A8. The NVIDIA architecture was used as a base with python generator to handle the data. The input size was different which will create different node sized CNN. One convolutional layer was removed from the network for this project.

- **Input image size is 160 x 320 x 3.**
- **Cropped image size 55 x 320 x 3**
-



Q9. Is the creation of the training dataset and training process documented?
A9. Elaborated in answer 6.

Q10. Is the car able to navigate correctly on test data?
A10. At no time the car goes over the red or brown ledges.