Project Writeup

Anirudh AV (anirudh.av93@gmail.com)

---

1. Project Overview

- I personally felt that 50% of this project is based on data collection more than tweaking the model architecture. Thus it took many iterations of disciplined data collection and book keeping to get the ”right” data for the model to ”work”.
- The data I collected manually did not keep the car to the center (even after selectively flipping some wide steering angle images.)
- So the udacity supplied data gave me a model that did that smoothly (still a little rough when compared to humans driving)
- However the udacity data did not provide the angles for aggressive maneuvers to tread the sharp turns on the track, So I appended to the udacity data by manually training the network by collecting data in the training mode.
- I stuck to the outer edge of the road so that the red stripes on the sharp turn were seen by all 3 cameras.
- This additional data along with reduced speed in the testing phase worked like a charm.
- It was a lot of work but after multiple iterations of data collection I was able to find the ”right” dataset.
- On the model Architecture side of things I used the nVidia architecture as suggested. Further details provided in the below sections.

2. Model Architecture

- As mentioned , I used the nVidia architecture for the network.
- To combat over-fitting I wanted to use dropout.
- Numerically , the loss with the dropout stagnated at a value and during testing it was able to manipulate the first two sharp turns but was not aggressive enough to handle the ”immediate” right sharp turn.
- Decreasing the dropout rate made the model perform better numerically during training, but was much worse during testing.
- This is where I would like to thank **Jeremy Shannon** for the idea of Using regularization and ELU to make the model smoother.
- Dropout is a form of regularization in some sense , hence it’s obvious why that worked. However, the reason that dropout did NOT work is still gray and I am further studying into it.

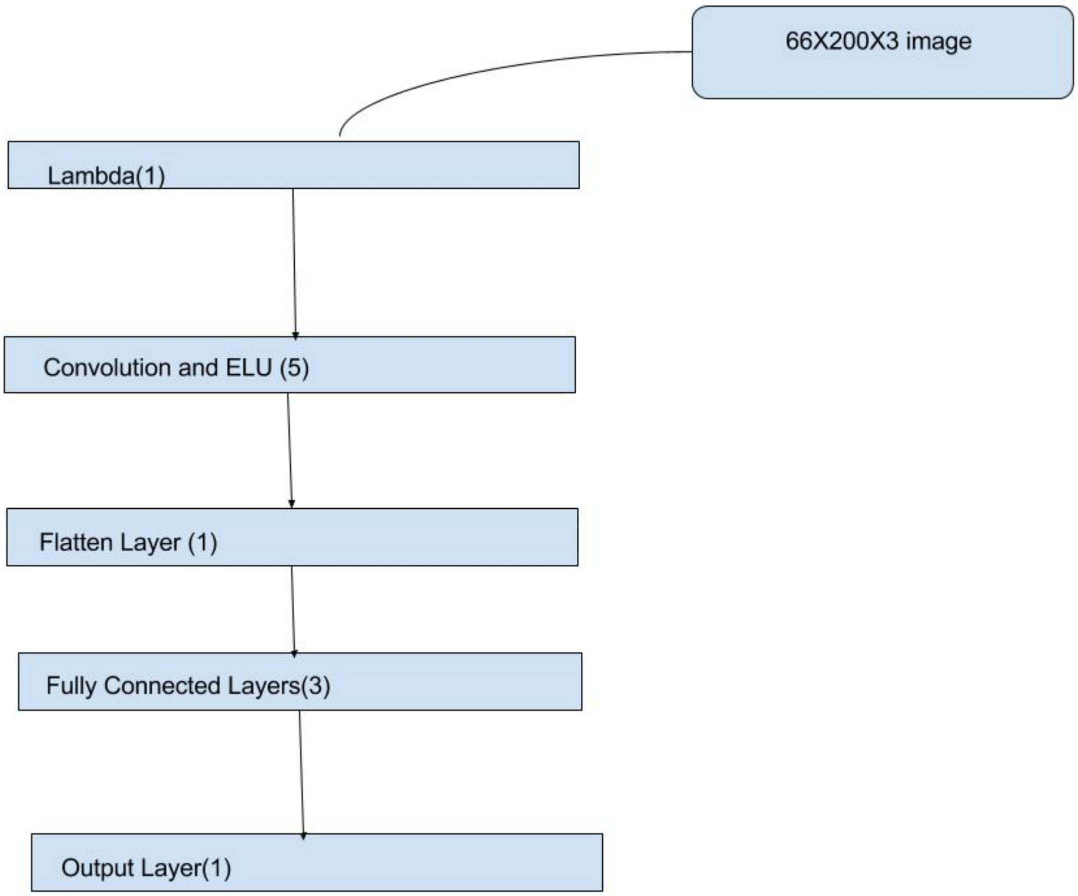

Figure 1: Model Schematic

3. The Data

3.1. Data Preprocessing

Preprocessing is done to accommodate the suggestions by the nVidia paper.

- Cropped the image so it takes 90 rows in the middle which leave out irrelevant data.
- Gaussian Blur with a kernel size of 3X3
- Resize the image to a 66X200 that the nVidia architecture takes
- Convert to the YUV color Space.

3.2. The distribution

Just like the traffic signs classifier project where some classes had too many examples compared to the others, here the steering angle 0 was the most common output class, hence the data was redistributed to flatten the dataset.

3.3. Additional Data

- I used images from all 3 cameras with a correction of 0.25 for the steering angle
- I augmented the images by flipping them and negating the angle as suggested. Helped me stay in the center.
- I applied some distortion to the image as suggested by a fellow student on slack. It did not change the behavior much , but I am sure it makes the model more general.

Figure 2: before flipping (left) after flipping (right)

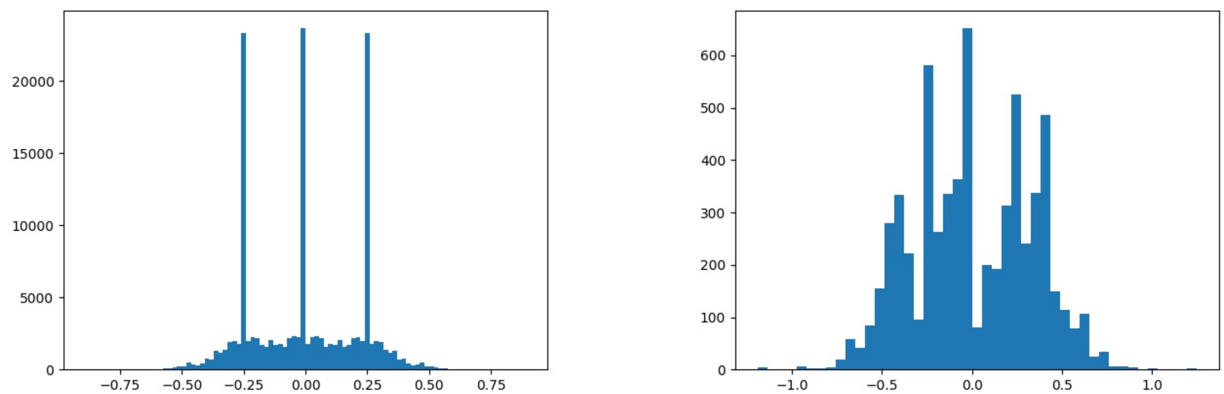

Figure 3: Before flattening (left) after flattening (right)

4. Conclusion

I got a self driving car to stay on the road in track 1 and a little more work needed for track 2. It was really fun and I would have a lot more to write about ”Ways of not making a self-driving car”.