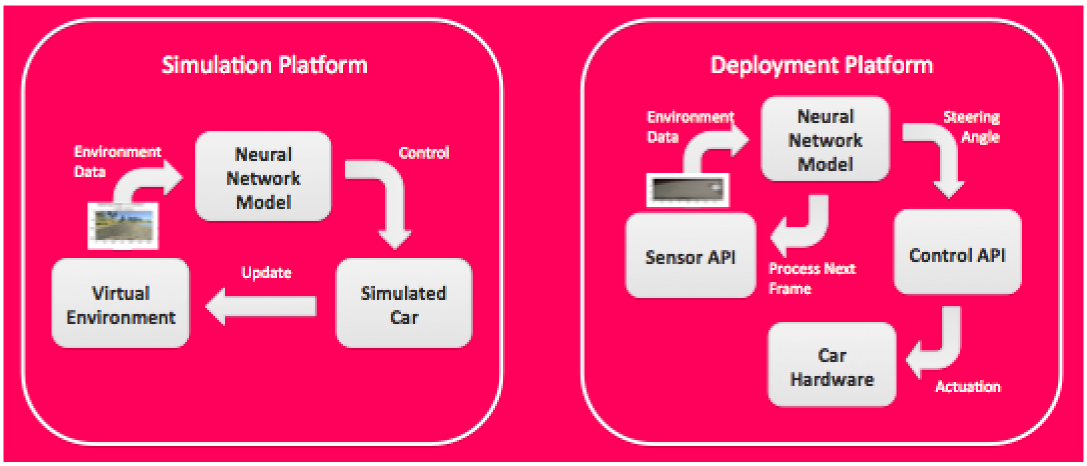
Team Pink Flamingo - Jianjie Liu, Michael Morscher

AutoDrive Final Project Status Report - April 12th, 2019

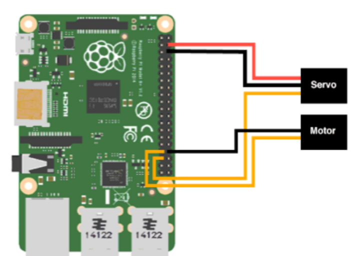
Introduction:

With the advent of autonomous vehicle, it is sensible for us to examine the technology powering what would be a revolution in transportation and in our society through training a self-driving model incorporating this behavior, and building a physical proof-of-concept miniature self-driving car.

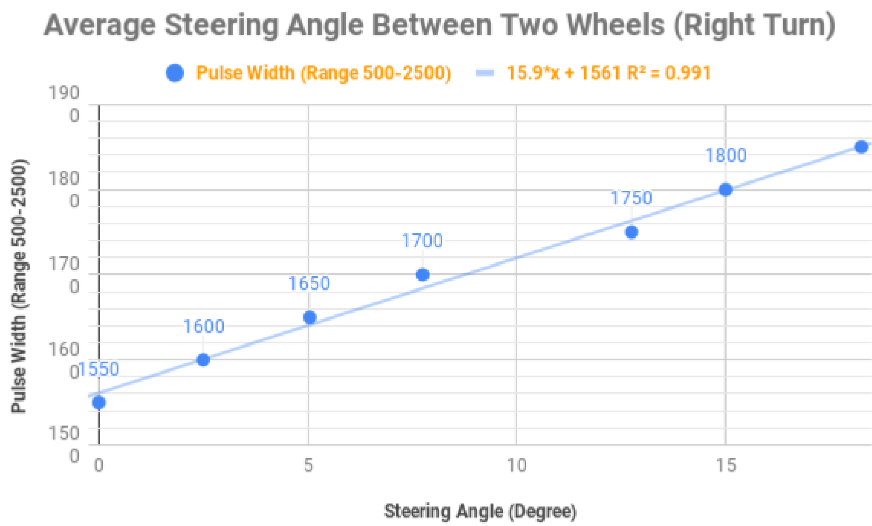
Design and Implementation:



We will be using two different platforms to build our prototype. We will use a simulator (see reference Item 3) to produce data to train a deep neural network published by Nvidia (see reference Item 4) which will be responsible for controlling the steering angle of the vehicle from images of the road condition. The simulation records a video of the user driving the vehicle and tags each video frame with the respective steering angle of the vehicle. We have collected two full loops of data (about 400MB) and trained the network for 8 hours on a 1.4 GHz Intel Core i5 CPU.



Once the neural network model is trained, we then deploy the model onto a raspberry pi which is wired to both the motor and servo of a RC car. To actuate the servo on the hardware for the desired steering angle, we conducted a series of experiments to determine the relationship between the steering angle and the pulse width of the signal needed to induce it.



With these information, we build low level control APIs for turning the vehicle and spinning up its motors through producing the desired pulse width signal. Finally, we establish a pipeline to feed real-time image from the camera on the vehicle to the neural network which controls the steering of the vehicle to stay on the designated track.

Deviations from Original Plan:

We have originally planned to incorporate more sensors (i.e. ultrasonic sensor) into the prototype, but we are restrained from doing so by limited physical space on the car and the low power from the raspberry pi.

Results:

The prototype is able to drive successfully under good light condition in a reasonable track (without curves of turning radius smaller than 30 inches). See project video at https://www.youtube.com/watch?v=xME5ly80URw.

Successes:

The model is able to drive the prototype in deployment which is very different from the simulation track. The model is doing reasonably well given the little training time and without abundant amount of data. Although the model only controls the steering angle, it is able to successfully control the vehicle under steady speed. We have set out to optimize our software pipeline to reduce the response latency of the vehicle and we have improve the latency from 0.5 second to 0.05 second.

Failures (and more importantly, why):

The large body of the prototype prevents it from driving in a smaller track. The car cannot turn its wheels more than 19 degrees and its minimum turning radius is 30 inches. This creates a discrepancy with the simulation because the simulated car can steer up to 30 degrees and the model often actuates steering angle higher than the physical limits of the prototype. This seriously affects the consistency of performance from the prototype.

Another factor that damages the consistency of the prototype is light condition of the environment. The raspberry pi camera has very limited dynamic range where if part of the track is overcasted even by a slight shadow, it is possible for the camera to underexpose the image, rendering anomaly in its driving behavior. At the current state, the model fails to interpret road intersections. Road intersections are absent throughout the training as all of the tracks are racing track cleared of all obstacles.

Conclusion:

Overall, this project is a successful proof-of-concept for an autonomous vehicle in small scale using end-to-end neural networks. The raspberry pi can fully support all the computation needed to steer the vehicle at functional level and additional hardware (Intel Movidius Compute Stick) can be put on to accelerate the computation. Certainly the low-powered setting prevents the use of a better sensor and complicated algorithm, but state of this project shows that a working prototype is possible. The inconsistency between the simulated car and the prototype has always been an issue. If this project were to be repeated again, we shall find a smaller chassis for the prototype and use a more sophisticated simulator where we can customize the training tracks. In addition, we should also modify the neural network to control both steering angle and throttle of a vehicle.

Reference:

1. AutoDrive Project Repo
   https://github.com/JohnGee96/AutoDrive
2. Neural Network Model Training Project Repo
   https://github.com/JohnGee96/CarND-Behavioral-Cloning
3. Udacity Autonomous Vehicle Simulator
   https://github.com/udacity/self-driving-car-sim
4. Nvidia's End-to-End Learning for Self-Driving Cars:
   http://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf