

一．项目背景

随着时代的发展，随之而来的交通事故发生率也随之上升，而其中很大一部分是由驾驶者的不正当驾驶行为所导致，而无人驾驶技术的出现，是可以起到由人为导致的交通事故率的，其作为一项跨学科的存在，其中作为行车安全的因素，路面探测和动作选择占了十分重要的地位。而当车辆的前置摄像头把车辆行驶中的前方图像传回控制终端，就需要控制终端作出合适的判断，控制方向盘的转动方向，以保证行车安全，本项目是通过 Tesla 的行驶数据，对其方向进行模拟和预测。

二．问题描述

我计划，根据公开的 Tesla 数据集对车辆的前置相机所拍摄的路况图像，实现对车辆转向角度的预测。同时保证不在商业行为上使用这些数据集。

三．输入数据

数据来源：https://github.com/nd009/capstone/tree/master/deep_tesla

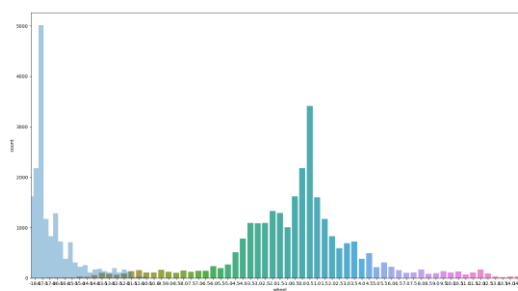
整个 repository 有几个子文件夹组成，其中包含

1) epochs:

包含了 10 代的数据集，其中分为操纵（steering）数据集和车辆前方状况（front）的数据集，其中 steering 保存为 CSV 格式的文件，其中包含 3 个 label：

ts_micro	frame_index	wheel
1464305394391807	0	-0.5
1464305394425141	1	-0.5
1464305394458474	2	-0.5

ts_micro 是时间戳，frame_index 是帧编号，wheel 是转向角度（以水平方向为基准，+为顺时针，-为逆时针），每个 steering 数据集有大概 1000~3900 行数据，总共有 27000 行数据，是足够训练一个普通的 CNN 模型的。经过合并数据集后，10 个 steering 数据集的 wheel 的值分布如图：



可见数据集的值分布是相对不平均的

2) images

驾驶程序的图片资源文件

3) models

保存程序训练好的模型

4) output

保存模型在数据集 epoch10_front.mkv 上的表现效果

5) utils.py

预置的读取模型、生成视频等函数 python 文件

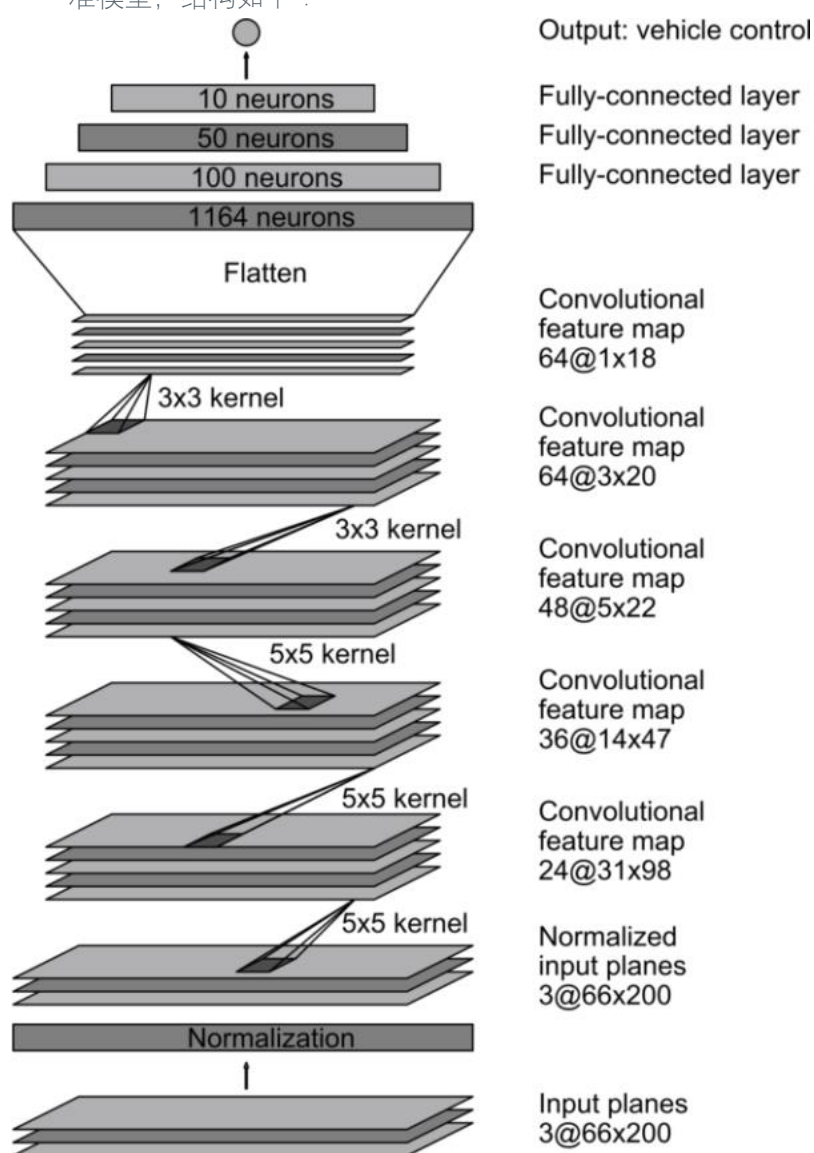
- 6) params.py
设置参数 python 文件
- 7) run.py
结果视频的主程序 python 文件

四 . 解决办法

通过对模型输入 front 和 steering 数据集，对模型进行训练后，得出被保存的模型文件 model.json 和 model.h5, 并可在网页应用 <http://selfdrivingcars.mit.edu/deepteslajs/> 进行模拟测试

五 . 基准模型

在 Nvidia 的无人驾驶的一篇论文 ([论文正文](#)) 中，提出了一个无人驾驶可用的 CNN 基准模型，结构如下：



本项目最终的模型结构将与该模型进行比较

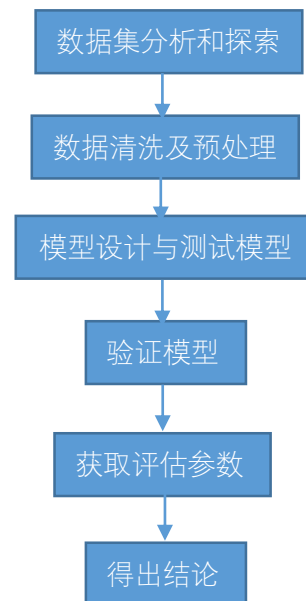
六．评估指标

我计划使用以下参数值对模型进行评估：

- 1) 向前/向后反应耗时值：forward/back pass ms
- 2) 平均误差值（实际和预测间的差值，据对数据集的观察，值存在正和负，因此此处将取绝对值以防止数据被抵消）：mean absolute error

七．设计大纲

经过初步分析各 front 和 steering 数据集，我计划使用以下步骤进行模型探索：



具体步骤为

- 1) 数据集分析和探索
使用第三方库对数据集和视频文件进行展示，使之有较为简单可视的展示
- 2) 数据清洗及预处理
分析和探索完数据集后，我将对数据集有更初步的了解，接下来，将先把前 8 个 front 和 steering 数据集作为测试和训练集，后 2 个作为验证集，而对于一辆无人车来说，仅 10 套数据肯定是不能满足它即将遇到的路况，因此，可能需要在对数据集进行一系列的加工
- 3) 模型设计与测试模型
探索完数据集后，将在此基础上设计神经网络模型，由于要对测试数据集尝试不同的层结构和参数调试，此步骤耗时估计将比较久
- 4) 验证模型
使用验证数据集对模型进行验证，并在 <http://selfdrivingcars.mit.edu/deepteslajs/> 进行模拟验证
- 5) 获取评估参数
根据模型在验证集上的验证结果，获取 forward/back pass ms 和 avg error 的评估指标
- 6) 得出结论
根据模型结果得出结论