机器学习工程师纳米学位毕业设计

自动驾驶汽车深度学习：DeepTesla

杨奕

2018 年 02 月 5 日

项目背景

自动驾驶是近年来人工智能研究领域里比较热门及相对成熟的技术，已经有很多实用的产品出现，如 Nivdia 公司的 High-level view of the data collection system 及参考训练模型。通过模仿人类视觉观察道路来判断驾驶方向是其中最基础的功能，DeepTesla 这个项目就是根据人类驾驶或机器驾驶录制的车辆前方视频及相应的车辆转弯角度数据来训练的端到端的汽车自动驾驶，即通过车辆行驶前方图像来控制车辆运行方向。

这个项目来源于 MIT 6.S094 公开课，这门课程包括 DeepTraffic，DeepTesla 等部分，DeepTraffic 主要是通过强化学习训练车辆识别信号灯、障碍物，而 DeepTesla 主要是使用深度学习的图像识别技术控制车辆行驶的方向。通过这个项目可以加强对卷积神经网络知识的学习，同时了解深度学习如何在实际应用中使用。

问题陈述

人类驾驶车辆在道路上行驶是通过眼睛来观察道路的弯曲程度，然后通过大脑处理后控制手来操控方向盘改变行车角度。深度学习的计算机视觉也可以达到这种程度，即通过计算机视觉对大量实际道路的行驶训练来学习到模型，通过这个模型可以计算出不同道路场景下车俩行驶的角度，从而达到与人类驾驶相同的效果。这种通过图像直接判断行驶方向的端到端的自动驾驶技术，如果模型设计合理及通过大量数据的训练应该可以达到或超过人类的驾驶水平。这个项目与其他图像分类问题不同的是，输出的结果是一个角度数据，而不是分类，因此是一个深度学习的回归问题。

数据和输入

这个项目的训练数据同样来自 MIT 6.S094 课程，其中 epochs 目录下包括 10 段在多种道路上行驶的视频，文件从 epoch01_front 到 epoch10_front，格式为 mkv，这些视频可以直接

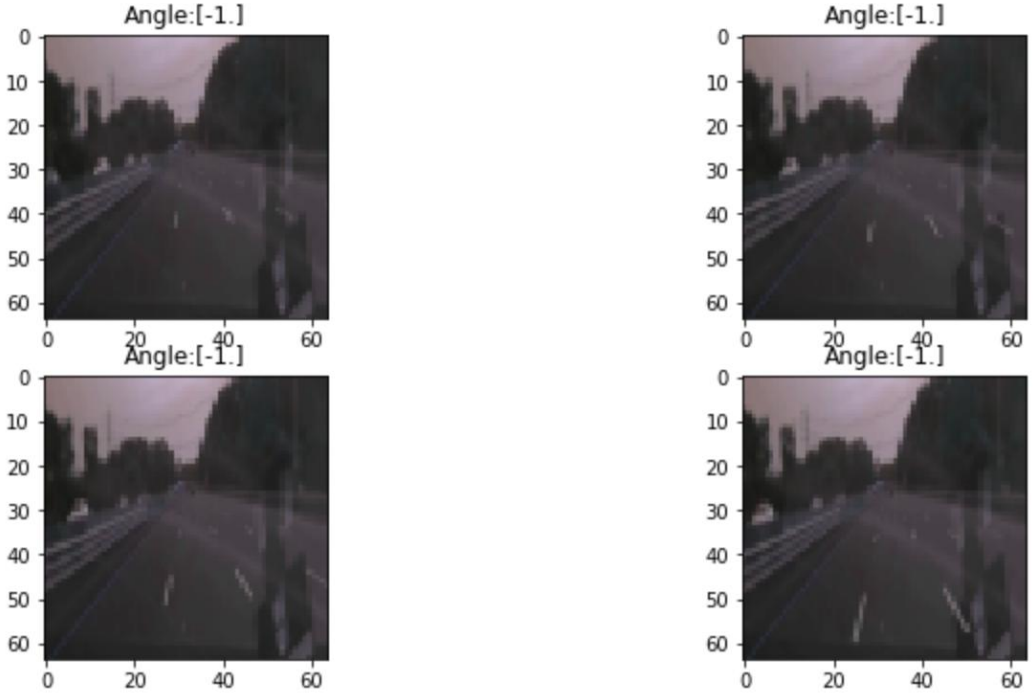
通过视频软件播放。

另有 10 个对应的控制信号 CSV 格式文件，文件的内容为 ts_micro 时间戳，frame_index 帧编号，wheel 转向角度（以水平方向为基准，+为顺时针，-为逆时针），这里对我们有用的是 wheel，即训练数据的标注信息。其中视频文件中的每一帧在 csv 文件中都有对应的转向角度。

我们训练时就是读取视频文件中的每一帧图像作为特征数据，其对应的转向角度作为标注来训练。每帧图片大小为 720*1280*3，在实际使用中要进行剪裁，每个视频文件的情况如下，可以看出视频包括不同的道路情况：

文件	帧数	场景	自动驾驶-1	Loss
epoch01_front.mkv	1500	高速公路，前方车俩少	1	11
epoch02_front.mkv	3900	高速公路，前方车辆较多	1	4
epoch03_front.mkv	2700	桥梁上	0	7
epoch04_front.mkv	2700	弯路多	0	73
epoch05_front.mkv	2700	二车道，在里侧通行	0	23
epoch06_front.mkv	2700	傍晚，在里侧通行，有隔离网	0	39
epoch07_front.mkv	2700	在中间车道通行，两边是树木	1	7
epoch08_front.mkv	2700	二车道，在里侧通行	0	23
epoch09_front.mkv	2700	在中间车道通行，两边是树木	1	3
epoch10_front.mkv	2700	在中间车道通行，两边是树木	1	

剪裁为 32*32*3 的图片样本如下：



解决方案

对于这个项目可以编写模型通过项目提供的训练数据进行训练。训练结束后可以通过验证数据集来验证验证数据与实际结果的准确率。最终的模型应该可以在模拟驾驶程序中判断道路视频的转向角度或通过摄像头实时拍摄道路的转向角度。在图像识别方面可以使用在 keras 中提供的成熟的模型，如 VGG16、RESNET50 等，由于这些模型都是作为分类问题存在的，而我们的项目是一个回归问题，因此需要进行改造，即保留原模型中卷积层、最大池化层，对于输出层则使用线性激活函数。

基准模型

对于行车道路的识别是一个典型的图像识别技术，可以采用深度学习中的卷积神经网络，通过设置多个卷积层，池化层等来训练。从 2012 年的 AlexNet 到 2015 年的 Residual Net，卷积神经网络的准确率越来越高，因此我们可以模仿或直接采用一个已有的模型来训练数据。根据此项目对于图像数据的要求及资源的限制，准备采用对资源要求较小而准确率稍高高的 VGG16 模型进行迁移学习，以达到良好的训练效果。同时还可以参考上节提到的 Nvidia 关于端到端自动驾驶的论文编写模型，比较两种方法的实际效果。

评价指标

对于模型的训练效果可以通过几种方法验证：

1、 在训练中观察每轮次的损失系数 loss，这里使用回归模型使用的 mse,正常情况

下 loss 应随着训练次数增加逐渐降低并最终收敛；

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

2、还可以通过计算视频验证预测结果与实际结果的决定系数 r2,正常情况下这个系数在 0-1 之间，越接近 1 越好；

- The total sum of squares (proportional to the variance of the data):
  $SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2,$
- The regression sum of squares, also called the explained sum of squares:
  $SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2,$
- The sum of squares of residuals, also called the residual sum of squares:
  $SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$

The most general definition of the coefficient of determination is

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}.$$

项目设计

DeepTesla 项目实际开发过程包括项目环境准备、特征和标注数据的准备和读取、模型的建立、参数的配置、模型训练、验证数据集验证结果分析、模型保存、验证数据输出和模拟驾驶环境下的模拟测试，以下分别介绍各部分内容。

1) 项目环境搭建

这个项目采用 python3 + keras + tensorflow + jupyter notebook 来完成，因此首先需要安装 python 3.5.4、keras 2.1.2、tensorflow 1.4.0 和 jupyter notebook,为了解决 python2 和 3 的版本问题，可以为 python3 建立虚拟环境。

2) 特征和标注数据准备和读取

在项目提供的源代码中 utils.py 已经提供了读取模型、生成视频等函数，可以直接使用，但需要根据训练模型设计一次读取几个特征文件，读取特征文件的大小。

3) 模型建立

可以直接使用 keras 中的 VGG16 模型进行改造，这里使用这个网络进行训练，而不是使用已经训练好的模型。

4) 参数配置

对于网络需要设置输入数据的 shape,训练的轮次 epochs,每批次数量 batch_size

5) 模型训练

使用模型训练并实时打印训练结果，观察损失率，根据结果调整训练参数，由于每个视频文件包含的图像较多，可以将这些图像分批次进行训练，如将视频数据的 01-08 作为训练数据，09 作为验证数据。

6) 测试

根据测试视频数据 epoch10_front.mkv 实际输出预测数据，即角度，同时计算实际结果和预测结果的决定系数。

7) 模型保存

使用 keras 保存功能保存模型 model.json 和 model.h5

8) 验证数据输出

根据模型生成 epoch10_front.mkv 的结果视频并保存在./output 中

9) 模拟环境测试

将网络输入课程所提供的网页应用中，利用 ConvNetJS 来测试

参考文献

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (10 Dec 2015).Deep Residual Learning for Image Recognition

[2] Lex Fridman,cars.mit.edu(Jan 2017).Course 6.S094: Deep Learning for Self-Driving Cars

[3] NVIDIA Corporation,End to End Learning for Self-Driving Cars

[4] Karen Simonyan,Andrew Zisserman,VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

[5] Coefficient of determination, https://en.wikipedia.org/wiki/Coefficient_of_determination

[6] Mean squared error, https://en.wikipedia.org/wiki/Mean_squared_error