

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 01 月 18 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，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果在实际应用中使用。

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辆行驶的角度，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。

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。另外最后一个视频文件 epoch10_front.mkv 可以作为测试数据, 测试模型的效果。

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, 这个项目与其他图像分类项目不同的是, 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。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。

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。对于 DeepTesla 项目拟采用准确率较高的 Residual Net, 已达到最好的训练效果。

评价指标

Residual Net 是目前准确率比较高的神经网络, 我们可以直接采用这个模型来训练, 训练结果应该与 Residual Net 的准确率类似。Residual Net 的准备率为 96% 以上, 因此我们训练的准备率应与此接近, 或至少大约 90%, 才能满足自动驾驶的需求。

项目设计

DeepTesla 项目是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 `ResNet50` 网络，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。

```
from keras.applications import ResNet50
```

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，观察准确率，根据结果调整训练参数

6) 验证

根据验证视频数据 `epoch10_front.mkv` 验证模型的准确率

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

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

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