

DeepTesla

开题报告

Fei He

2017 年 10 月 16 日

一、项目背景

近年来，深度学习神经网络在计算机视觉领域，特别是图像识别上取得的突破性进展无疑是科技界的热点之一。诚然，神经网络并非新兴技术，从时间线上可以看出它的发展脉络：1958 年康奈尔心理学家罗森布拉特推出感知机；1986 年 Hinton 等人发明训练多重神经网络纠错的方法；1989 年 LeCun 用卷积神经网络识别手写体；1991 年递归神经网络发明；2007 年李飞飞创立 ImageNet；2012 年谷歌大脑识别猫脸（6 月），8 月谷歌将神经网络引入语音识别，10 月 Hinton 的学生在 ImageNet 竞赛夺冠，成绩大幅提升；2015 年 12 月微软 ResNet 图像识别准确率超越人类。在将近 50 年的发展历史中，其迎来爆发的根本原因就在于现今计算机强大的计算力和海量的数据。现在，这项技术已经延伸到医疗、游戏、公共服务等等和人们生活息息相关的各个层面之中。

在众多深度学习神经网络应用场景之中，无人驾驶（自动驾驶）无疑是中短期内最有前景的之一。现阶段，以摄像头作为道路图像传感器，结合人工智能和视觉处理技术实现无人驾驶（自动驾驶）在成本上体现了巨大的优势。不同的神经网络模型已经应用到车辆转向控制和速度控制中。如英伟达（NVIDIA）提出的端到端¹（end to end）卷积神经网络模型（Convolutional Neural Network，以下简称 CNN）；Comma.ai 提出的²变分自动编码器（Variational Autoencoder，以下简称 VAE）及基于生成式对抗网络（GAN generative adversarial network，以下简称 GAN），用于实现道路视频预测的代价函数（Cost Function），在此基础上结合了递归神经网络（Recurrent Neural Networks，以下简称 RNN）的转换模型（Transition Model）等。

在这些已经实现的案例中，与传统的方法相比，神经网络和深度学习展现了惊人的效率和优势。本项目就是基于 MIT 6.S094 这门公开课中的 Tesla 数据集训练深度学习模型，根据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，从而对不同的深度学习模型有一个初步的探索。

二、问题描述

该项目的目的在于利用深度学习模型：端到端 NVIDIA end-to-end 卷积神经网络模型（CNN）；VGG16 + Nvidia model 迁移学习（Transfer Learning）模型。尝试使用 VAE + GAN 模型生成训练数据图像、CNN + RNN model（包括一对一，多对一）和 CNN+RNN seq2seq model（RNN seq2seq, Recurrent Neural Networks Sequence

¹M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.

²E. Santana and G. Hotz. Learning a driving simulator. arXiv



Figure 1: 视频帧图片

to Sequence)。根据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，并评估不同模型的表现。

该项目中涉及到近两年无人驾驶中几个先进的模型，将试图整体上对无人驾驶转向角度预测问题有一个全面的了解与探究。

三、输入数据

主要包括 tesla 在两种不同驾驶模式（human driving 和 autopilot）下的前置相机录制的视频和车辆的转向控制信号。数据可以从这里下载:[百度云](#)，数据集包含 10 段 H264/MKV 格式视频，视频帧为 30 fps，截取画面像素为 1280*720，视频数据格式如图“Figure 1: 视频帧图片”：

每一个视频都对应一张存储有对应时间戳（ts_micro）和帧编号（frame_index）和转向角度（wheel）的 CSV 格式表格。表格如下：

数据使用的基本思想为：将所有视频帧全部转换为 JPEG 格式 PNG 格式保存于内存（如有需要，保存于硬盘文件中），每幅图片对应于相应的 CSV 格式表格一行（只是提取出转向角度）。用于模型的训练数据和测试数据。对于图像数据，可以转化为灰度图片和进行数据归一化加快模型训练收敛速度；对于目标变量 CSV 表格中的转向角度，则首先需要查看其平衡性：如图“Figure 3: 转向角度数据分布”：

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

Figure 2: CSV 表格

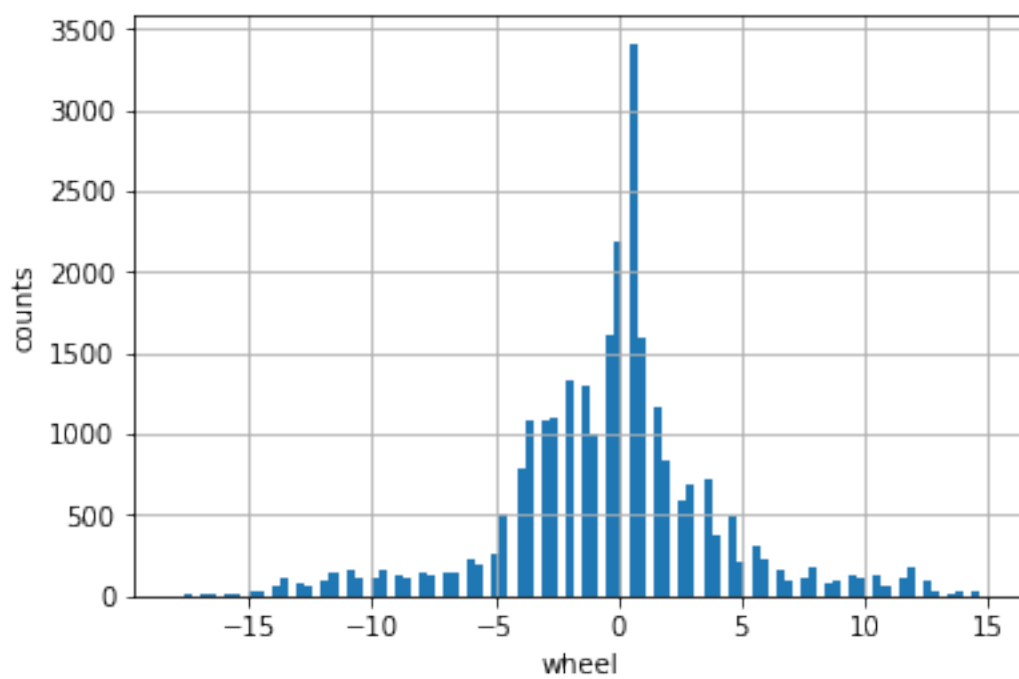


Figure 3: 转向角度数据分布

可以看出, 整个转向角度数据分布基本为正态分布, 可以发现车辆的控制信号中角度为 0 附近的数据比较多, 意味着得到的模型可能对比较大的转弯反应“迟钝”, 解决方法则可以通过复制转向角度较大的数据以增加训练数据来解决, 使得模型学到有价值的转向策略 (一般情况, 一个好的数据集应该是顺时针、逆时针、直行的转向信号频次足够接近的数据集, 即“对称”的数据集。但从逻辑上, 人类开车时需要转弯的频率是要远低于直行微调方向盘的频率, 增加整体训练数据也可以让模型学到转弯的策略)。

对数据集大致了解后, 就可以进行数据预处理: 通过平移, 翻转, 加噪声等方法从已有数据中创造出一些“新”的数据。也就是 Data Augmentation; 通过裁剪、重设图像大小或灰度化来简化训练数据集。另外, 可以使用 VAE + GAN 网络通过非监督学习生成训练用的数据图像。

四、解决办法

在本项目中的解决办法 (必做) 为: 使用 NVIDIA 提出的结构端到端 (end-to-end), 所谓端到端指的是神经网络输入为原始图片, 神经网络输出为直接控制的命令。其特点在于, 特征提取层和控制输出层的分界并不明显, 网络中的每一个部分对于系统来说都起着特征提取和控制作用; 另一种方法是利用迁移学习 (Transfer Learning) VGG16 在第一种方法的基础上优化神经网络模型。通过训练数据训练模型, 然后使用模型完成从测试图像到转向角度的映射。

另外的方法 (选做) 为: 在卷积神经网络 (CNN) 基础上引入循环神经网络 (RNN) 一对一、多对一以及序列到序列模型 (RNN seq2seq, Recurrent Neural Networks Sequence to Sequence), 将视频帧的时间序列作为参考因素, 实现图像对转向角度的预测。

五、基准模型

1、NVIDIA end-to-end model (必做): 英伟达 (NVIDIA) 一直以来致力于深度学习和无人驾驶的研究。在 2016 年 NVIDIA 发表的论文³中介绍了如何利用 CNN 实现端到端 (end to end) 的无人驾驶。其实现可以简单概括为利用卷积层提取图像特征, 在卷积层后接全连接层对卷积层提取到的特征进行训练并最终输出转向控制信号。该模型使用 keras 实现。

2、VGG16 + Nvidia model (必做): 剑桥大学和 Google DeepMind 的研究人员在 2014 年论文⁴中提出的一种用于大规模图像识别的深度卷积神经网络, 大体思想是用较小的卷积核 (3x3) 和较深的卷积层 (16-19 层) 对图像通用特征进行提取。利用迁移学习 (Transfer Learning: 把预先训练好的模型, 应用到类似的处理任务上, 达到节约训练时间和提高模型表现的目的), 使用 VGG16 优化模型。由 Alex-net⁵发展而来的 VGG 网络主要修改了两个地方: (1) 在卷积层使用更小的 filter 尺寸和间隔; (2) 整个图片和 multi-scale 上训练和测试图片。VGG16 初始模型的.py 文件来源于: <https://github.com/fchollet/deep-learning-models/blob/master/vgg16.py>, 训练集是 ImageNet, 训练好的模型权重参数保存为.h5 文件中, 分为 notop 和 withtop 两种, 其中 notop 不包

³End-to-End Deep Learning for Self-Driving Cars, Blog Post: [End-to-End Deep Learning for Self-Driving Cars](#)

⁴Very Deep Convolutional Networks for Large-Scale Image Recognition: <https://arxiv.org/pdf/1409.1556.pdf>

⁵AlexNet: http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf

含原 VGG 结构的 3 个全连接层，可以广泛应用在特征提取中。从链接⁶下载了 keras 使用的 VGG16 notop 的权重。该模型使用 keras 实现。

3、VAE + GAN model（选做，用于生成训练和测试图像）：Comma.ai 在文献⁷中提出，使用自编码器（Autoencoder）来实现道路预测图像的生成。简单来说就是 VAE 与 GAN 网络通过训练道路图像，这样编码器能够同时优化隐藏层的高斯先验模型以及由 GAN 网络提取出特征上的相似性。发生器（generator）会将隐藏层输出的随机样本作为输入，并输出编码器网络，在经优化后即可“欺骗”辨别器（fool discriminator），并尽量减少原始图像与解码图像的相似性。判别器则始终训练以区分输入图片的真实性——判别真伪。训练完成后，使用生成器就可生成道路驾驶图像。由于该模型较复杂，在项目中参考 Comma.ai 源代码实现简单的模拟。该模型使用 keras + Tensorflow 实现。

4、CNN+RNN 一对一、多对一 model（选做）：参考 UDACITY 的“深度学习纳米学位”⁸中“循环神经网络”课程，将问题简化为 RNN 的一对一模型，即一幅图像预测一个转向角度，使用 LSTM（Long Short-Term Memory），实现信息持久化。另外在此之上，多对一模型则是通过每幅图像“向前看”来实现几幅图像预测一个转向角度，同样需要使用 LSTM（Long Short-Term Memory），实现信息持久化。该模型使用 keras 实现。

5、CNN+RNN seq2seq model（选做）：参考 UDACITY 的开源挑战项目⁹“Teaching a Machine to Steer a Car”，其中第一名“Team Komanda”利用了循环神经网络序列到序列模型（RNN seq2seq, Recurrent Neural Networks Sequence to Sequence），实现了一个不同长度序列输入图像到输出转向角度的映射模型，将时间序列的因果关系引入了模型。由于该模型较复杂，将参考源代码进行修改，通过 Tensorflow 来实现模型。

说明：在选做的模型中，至少完成两个选做模型。在该项目中使用多种模型的目的在于，通过大量的实验、实践不同的神经网络从而对深度学习有一个更深入的理解和掌握。

六、评估指标

就整个项目来说，这是一个回归问题，均方误差（Mean Squared Error, MSE）可以作为评估指标。

$$MSE = \frac{1}{N} \sum_{i=1}^N \sqrt{y_p - y}$$

其中， y_p 是预测的转向角度， y 是真实的转向角度，样本数量为 N ，均方误差低则模型表现优。

⁶https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5

⁷E. Santana and G. Hotz. Learning a driving simulator. arXiv

⁸优达学城（Udacity）深度学习纳米学位：<https://cn.udacity.com/dlnd>

⁹Udacity. An open source self-driving car, 2017: <https://medium.com/udacity/teaching-a-machine-to-steer-a-car-d73217f2492c>

七、设计大纲

第一部分：数据分析与探索。这一步骤主要是读取数据（包括 csv 文件和视频文件）、分析数据（如转向角度的数据分布等）。

第二部分：数据预处理。在这部分中讲前九段视频数据（转化为图片后）划分为训练集和验证集，将第十段视频（转化为图片后）划分为测试集。因为深度学习在大量的数据集下才能发挥出性能，所以对训练集和验证集运用数据增加技术：数据归一化、剪切及重设尺寸，随机翻转和更改亮度等。另外，尝试使用 VAE + GAN 来生成训练用的道路图像。

第三部分：利用 Keras 以及 Tensorflow 训练模型并导出模型参数。使用基准模型对训练集进行训练，并通过验证集评测模型表现。期间涉及到技术有：卷积网络 and 全连接网络搭建与调参、迁移学习网络搭建与参数固定和调参、Convolution3d 的使用、seq2seq 模型搭建与调参、残差网络的使用、训练数据 RELU 和 ELU 激活函数运用、运用 Droupout 层防止过拟合以及训练测试数据生成技巧等。最后选取最佳模型，并将训练好的模型和参数保存为 model.json 和 model.h5 两个文件。

第四部分：测试模型表现。在测试集上测试模型表现，评估人工转向与模型转向误差。最后对测试视频 epoch10_front.mkv 生成结果视频并保存在 ./output 中。