机器学习纳米学位

万立佳

2018 年 10 月 29 日

I.问题的定义

项目概述

项目的目的是识别驾驶员在驾驶过程中是否有分心的行为。驾驶员长时间驾驶很难保证一直处于安全驾驶的状态，根据美国的疾病防控中心的分析，车祸中有大约五分之一是由于司机分心导致的，分心动作有：打电话、发短信、吃东西、聊天等，因此准确识别驾驶员在驾驶过程中是否分心可以有效的提醒驾驶员规范驾驶，以降低车祸的发生概率。

这个项目涉及图像识别领域，此项目可有效的帮助降低车祸的发生，是非常有意义的项目，并且此解决方案亦可用于其他领域。该项目是 state farm 公司在 kaggle 上发起的，数据也由该公司准备，数据中包含训练数据和测试数据，训练数据中包含若干驾驶员模拟驾驶的拍照图像，且在训练数据中已对用户的行为打标。

问题陈述

该问题是通过学习已标注的 10 类驾驶员的分心动作图像，来预测测试集图像中驾驶员的状态，是图像分类的问题。

图像中驾驶员状态有：

c0.安全驾驶

c1.右手打字

c2.右手打电话

c3.左手打字

c4.左手打电话

c5.调收音机

c6.喝饮料

c7.拿后面的东西

c8.整理头发和化妆

c9.和其他乘客说话

卷积神经网络是近年迅速在普及的新型图像识别算法和架构，由于imagenet的推波助澜，至今已发展出许多不同的版本，在图像识别领域取得出色成绩，该项目将使用卷积神经网络来识别这些图像属于哪种状态。

该项目的希望的结果是，对测试集中的每张图片预测出对应十种类别的概率值。例如：对于安全驾驶类别的图像来说，结果可能是c0类的概率为0.95，其余类别不足0.5。即结果图像是在每个类别上的概率值，基本上不会出现在某一类别上的概率恰好为0的情况。

评价指标

采用kaggle中相同的度量指标：multi-class logarithmicloss，公式如下：

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

其中N代表测试集样本的数量，M代表分类的个数，本项目中即为10.log是自然对数，y_{ij}有两种取值，当样本i属于j类时取值为1，否则取值为0。p_{ij}是样本i属于j类的概率值。logloss指标比accuracy考虑的更加细致，因为不仅考虑是否预测该类别的概率最大，而且考虑到了和其他类别的区分度。在结果中并不要求对每张照片的预测结果的和必需等于1，因为每个概率会除以所有图片所有概率的和。

II.分析

数据的探索

数据集源于 state farm 上传到 Kaggle 的数据。包括三个文件，

driver_imgs_list.csv 说明：训练数据的映射文件，描述训练数据集中：驾驶员、类别、路径的对应关系。

sample_submission.csv 说明：提交结果的样例文件，其中包括了每个图片对应在十种类别上的概率值。

imgs.zip 说明：图像数据压缩包，其中有 test（测试集）、train（训练集）两类数据，其中训练集中对应十种类别有具体的十个子文件夹，每种文件夹对应一种类别。

图像数据集中包含大量车载摄像头对驾驶员位置的摄影截图，可清楚看到驾驶员的各种行为，包括打电话、喝饮料、拿后面的东西、发消息等。训练集中已将图像标记分类，分为 c0 到 c9 一共十个文件夹存放，共 22,424 张图片。测试集中有 79,729 张未标记分类的图片。数据集中每一张图片大小为 640*480 像素。图片中的驾驶员的外貌区别很大，胖瘦、高矮、性别、肤色、年龄都有所区分，同时图片的光线、窗外的背景、车内饰的颜色也有所不同。

图 1 原始图像的呈现

探索性可视化

分析训练集的映射文件 driver_imgs_list.csv，可得以下总结：

1. 总共有 22,424 张已标记图像。
2. 总共有 10 种不同的驾驶状态。且总体来看，每种状态的图片数量差别不大，即它们呈均匀分布。
3. 总共有 26 名驾驶员，每名驾驶员的图像数量并不完全相同，且每个驾驶员在不同类别上的图像数量分布也有所不同，其中分布最不均匀的是驾驶员 p72。

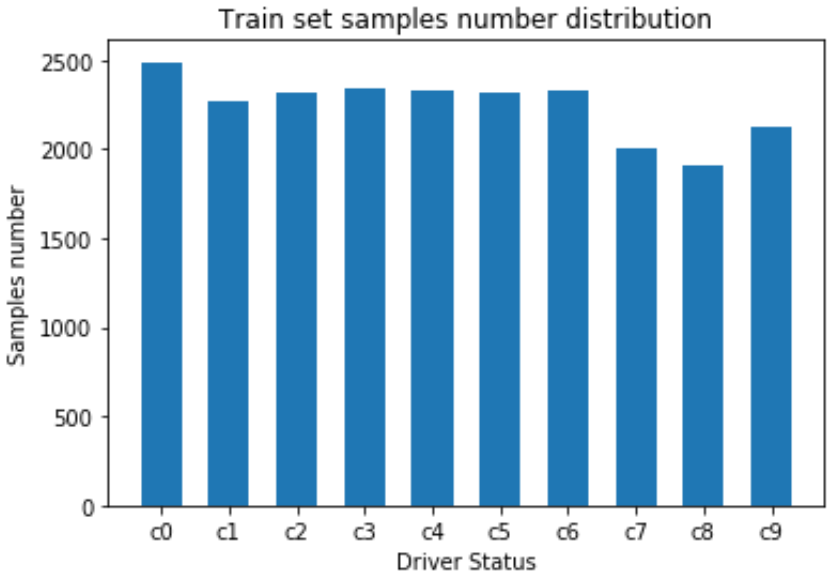

图 2 训练集总体样本各类别数量分布

由上图可知，训练数据中所有司机状态分类呈均匀分布。

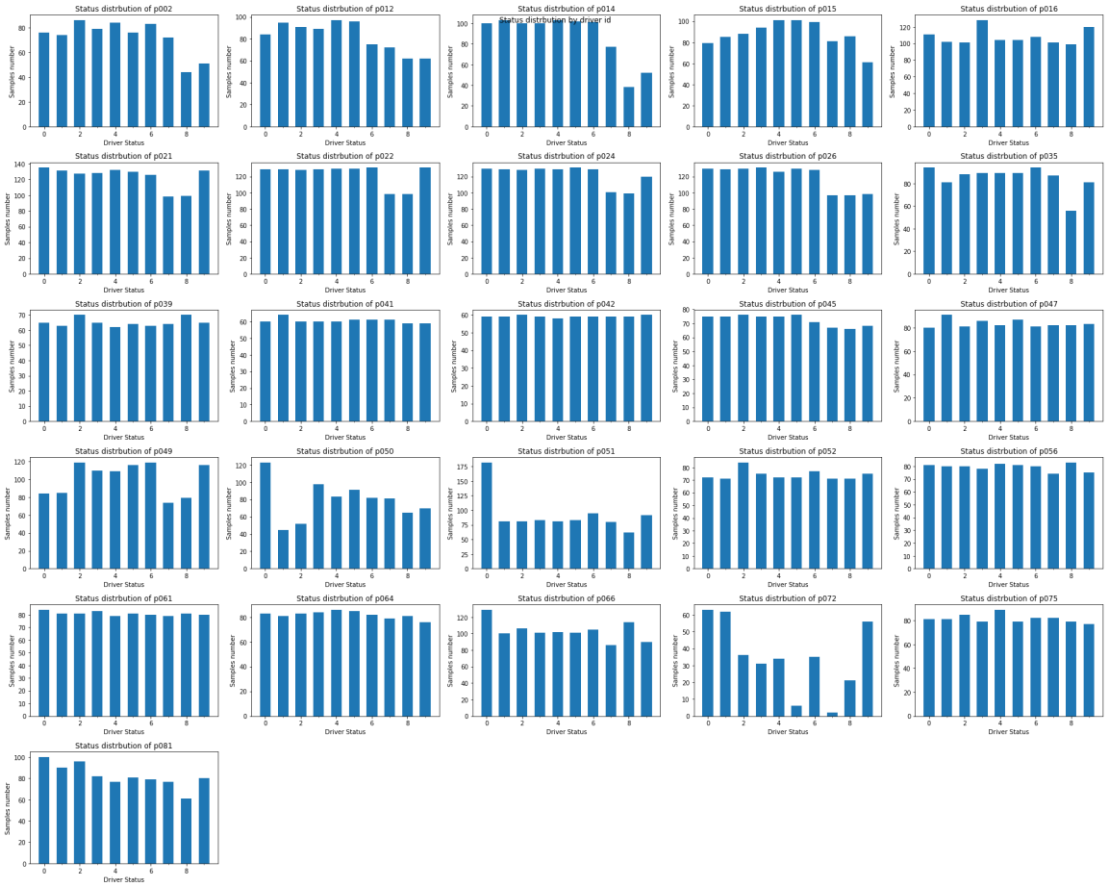

图 3 各司机各类样本数量分布

由细分到每个司机的分类图像数量可以看出，虽然每个驾驶员的图像总量分布还算均匀，p72 驾驶员在各个分类上的数据严重不均匀，同时 p14、p50、p51 也存在不均匀的情况。

算法和技术

TensorFlow

TensorFlow 是一个 Google 公司开发的基于数据流图计算的人工智能神经网络处理框架，可广泛应用于如语音识别、自然语言理解、计算机视觉、广告等领域。

Keras

Keras 是一个高层神经网络 API，基于 Tensorflow、Theano 以及 CNTK 后端框架的上层框架，让 Tensorflow 等基础神经网络框架更易于使用。

数据增强

简单来说，数据增强是通过图像变换，例如：缩放、旋转、去燥、模糊等操作，生成更多的训练样本数据，用以增强模型的泛化能力，降低模型的过拟合的可能性。项目使用 Keras 中的 ImageDataGenerator 对训练的图像增强。

InceptionV3

InceptionNet 由 Google 提出，以较大优势取得了当年 ImageNet 比赛的第一名，当前为 InceptionV1，而 InceptionV3 则为它的后期改进版本。InceptionV3 拥有比 AlexNet 和 VGG19 更大的网络，但其计算量只有 15 亿次浮点运算，同时只有 500 万的参数量，准确率却大大胜于 AlexNet 和 VGG19。拥有着计算量小、模型训练快、分类准确的特点。

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

图 4 Inception **的网络层次结构**

InceptionNet 内部多处使用 InceptionModule，总共拥有 3 种，共 10 个 InceptionModule，多个小的网络反复使用一起堆叠成了一个大的神经网络。InceptionNet 在 V2 版本时还提出了 BatchNormalization 方法，是一个非常有效的正则化方法，可以让大型卷积网络的训练速度加快很多倍，同时收敛后的分类准确率也可以得到大幅提高。

ResNet50

ResNet（ResidualNeuralNetwork）由微软研究院的 KaimingHe 等 4 名华人提出，通过使用 ResidualUnit 成功训练 152 层深的神经网络，在 ILSVRC2015 比赛中获得了冠军，取得 3.57%的 top-5 错误率，同时参数量却比 VGGNet 低，效果非常突出。

ResNet 的结构可以极快地加速超深神经网络的训练，模型的准确率也有非常大的提升。ResNet 最初的灵感出自这个问题：在不断加神经网络的深度时，会出现一个 Degradation 的问题，即准确率会先上升然后达到饱和，再持续增加深度则会导致准确率下降。

这并不是过拟合的问题，因为不光在测试集上误差增大，训练集本身误差也会增大。假设有一个比较浅的网络达到了饱和的准确率，那么后面再加上几个的全等映射层，起码误差不会增加，即更深的网络不应该带来训练集上误差上升。而这里提到的使用全等映射直接将前一层输出传到后面的思想，就是 ResNet 的灵感来源。假定某段神经网络的输入是 x，期望输出是 H(x)，如果我们直接把输入 x 传到输出作为初始结果，那么此时我们需要学习的目标就是 F(x)=H(x)-x。

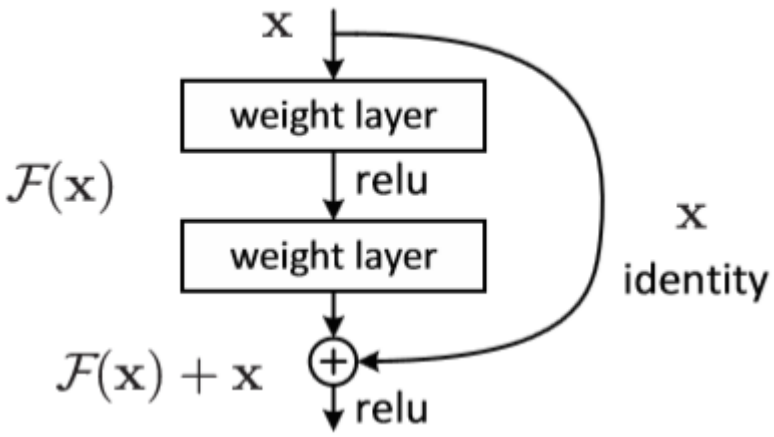

图 5 ResNet 残差学习单元示例

这就是一个 ResNet 的残差学习单元（ResidualUnit），ResNet 相当于将学习目标改变了，不再是学习一个完整的输出 H(x)，只是输出和输入的差别 H(x)-x，即残差。

Xception

Xception 也是由 Google 发布的深度神经网络架构。Inception 模块是一大类在 ImageNet 上取得顶尖结果的模型的基本模块，例如 GoogLeNet、InceptionV2/V3 和 Inception-ResNet。有别于 VGG 等传统的网络通过堆叠简单的 3*3 卷积实现特征提取，Inception 模块通过组合 1*1，3*3，5*5 和 pooling 等结构，用更少的参数和更少的计算开销可以学习到更丰富的特征表示。
通常，在一组特征图上进行卷积需要三维的卷积核，也即卷积核需要同时学习空间上的相关性和通道间的相关性。将这两种相关性显式地分离开来，是 Inception 模块的思想之一：Inception 模块首先使用 1*1 的卷积核将特征图的各个通道映射到一个新的空间，在这一过程中学习通道间的相关性；再通过常规的 3*3 或 5*5 的卷积核进行卷积，以同时学习空间上的相关性和通道间的相关性。

但此时，通道间的相关性和空间相关性仍旧没有完全分离，也即 3*3 或 5*5 的卷积核仍然是多通道输入的，那么是否可以假设它们可以被完全分离？显然，当所有 3*3 或 5*5 的卷积都作用在只有一个通道的特征图上时，通道间的相关性和空间上的相关性即达到了完全分离的效果。

若将 Inception 模块简化，仅保留包含 3*3 的卷积的分支，再将所有 1*1 的卷积进行拼接，进一步增多 3*3 的卷积的分支的数量，使它与 1*1 的卷积的输出通道数相等：此时每个 3*3 的卷积即作用于仅包含一个通道的特征图上，作者称之为“极致的 Inception（ExtreamInception）”模块，这就是 Xception 的基本模块

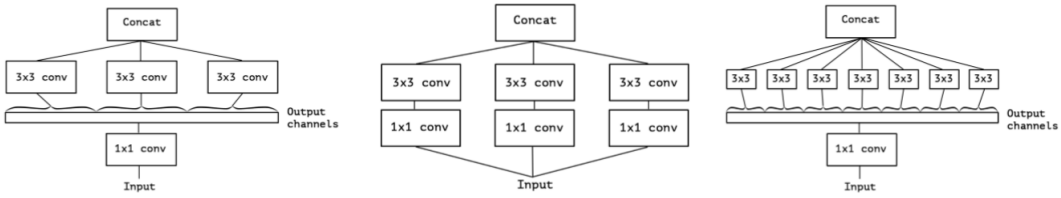

图 6 ExtreamInception 示例

事实上，调节每个 3*3 的卷积作用的特征图的通道数，即调节 3*3 的卷积的分支的数量与 1*1 的卷积的输出通道数的比例，可以实现一系列处于传统 Inception 模块和“极致的 Inception”模块之间的状态。

Xception 网络则由这些“极致的 Inception”模块构成，以下是结构图：

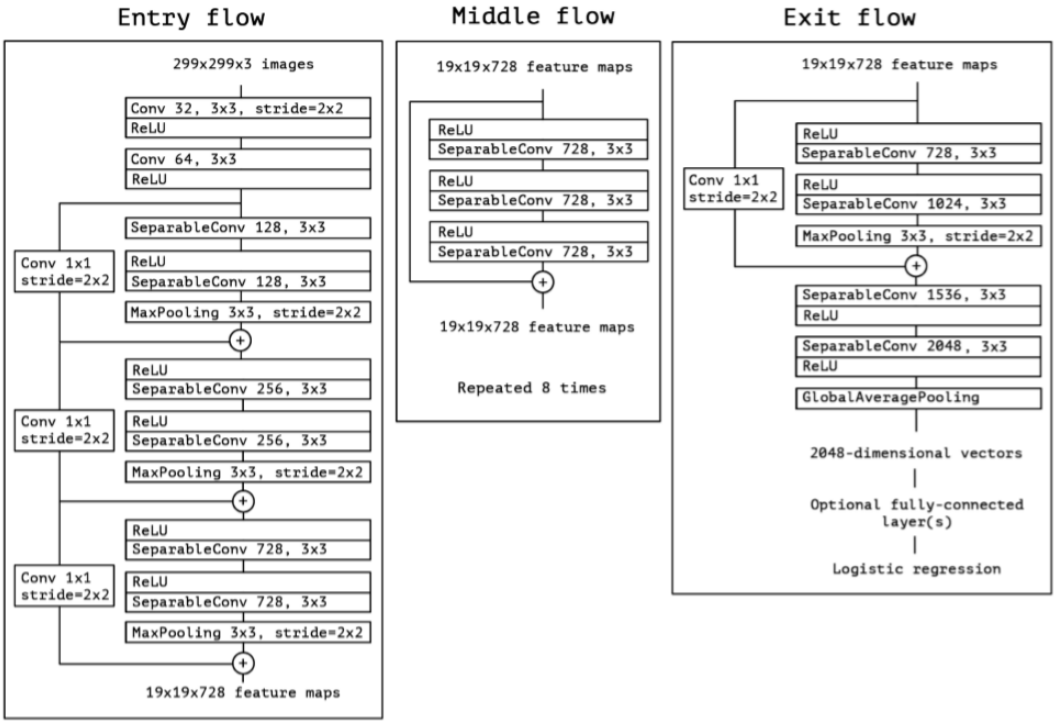

图 7 Xception 网络构成图

DenseNet

DenseNet 中的 DenseBlock 的一个重要思想就是对前每一层都增加一个单独的 shortcut，使得任意两层网络都可以直接“沟通”，也是参考了 ResNet 的残差网络思想，但使用的更多、更广，不过由此带来的代价就是训练时占用的显存更多。

DenseNet 和 ResNet 的一个明显区别是，ResNet 是求和，而 DenseNet 是做一个拼接，每一层网络的输入包括前面所有层网络的输出。第 L 层的输入等于，其中 k 是生长率，表示每一层的通道数，比如下图网络的通道数为 4。

DenseNet 提升了信息和梯度在网络中的传输效率，每层都能直接从损失函数拿到梯度，并且直接得到输入信号，这样就能训练更深的网络，这种网络结构还有正则化的效果。其他网络致力于从深度和宽度来提升网络性能，DenseNet 致力于从特征重用的角度来提升网络性能

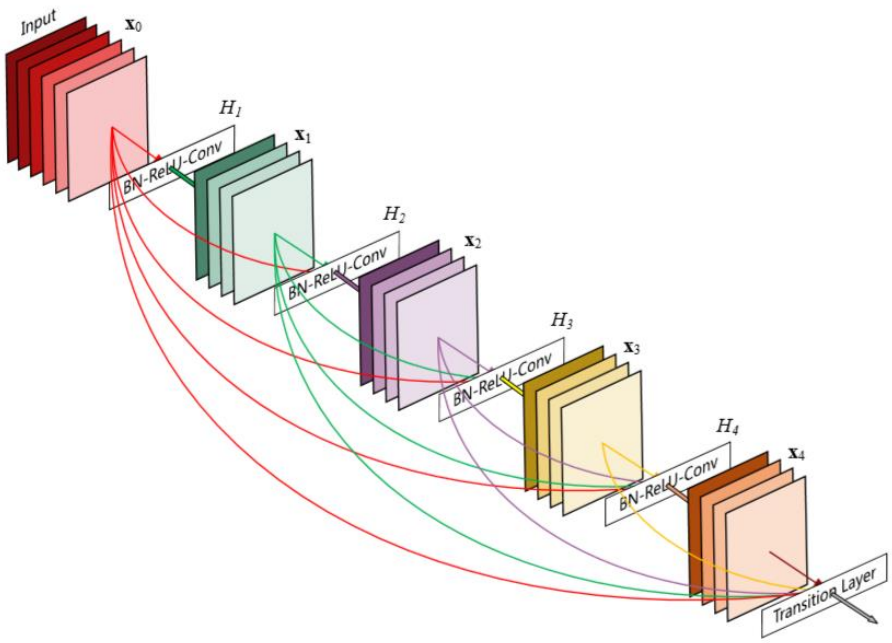

图 8 DenseNet **网络结构展示图**

这种结构的好处是可以缓解梯度消失，省参数省计算，特征重用可以起到抗过拟合的作用。达到相同的精度，densenet 只需要 resnet 一半的参数和一半的计算量。

SGD

随机梯度下降优化器。和 BGD 的一次用所有数据计算梯度相比，SGD 每次更新时对每个样本进行梯度更新。对于很大的数据集来说，可能会有相似的样本，这样 BGD 在计算梯度时会出现冗余，而 SGD 一次只进行一次更新，就没有冗余，而且比较快，并且可以新增样本。

Adam

Adam 是另一种计算每个参数的自适应学习率的方法。存储了过去梯度的平方的指数衰减平均值，也保持了过去梯度的指数衰减平均值：如果和被初始化为 0 向量，那它们就会向 0 偏置，所以做了偏差校正，通过计算偏差校正后的和来抵消这些偏差：然后使用这些变量来更新学习参数：

实践表明，Adam 比 SGD 优化方法效果更好，学习速度更快，也不像 SGD 容易受鞍点影响导致停留在局部极小值。

基准模型

使用 Kaggle 中该项目的排名分数作为基准模型。使用前 10%的分数作为基准，第 144 名，最小损失值为 0.25634。

III.方法

数据预处理

经过抽察数据集发现，ID 为 p081 的这名司机在 c0 安全驾驶类别中存在多张图像能判断为与其他乘客聊天的状态，属于异常数据，决定弃用 p081 在 c0 类别中的所有数据。

driver_imgs_list.csv 数据表就像一个训练图像集的索引文件，通过它可引用到对应的图像文件。使用该文件中的司机 ID 的指引，排除掉 ID 为 p081 的司机在 c0 分类中的图像数据，此后的数据处理和训练都将基于去除异常值后的数据集处理。去除 p081 的司机在 c0 分类中的图像索引后，所有样本中各类数据分布仍然呈平均分布：

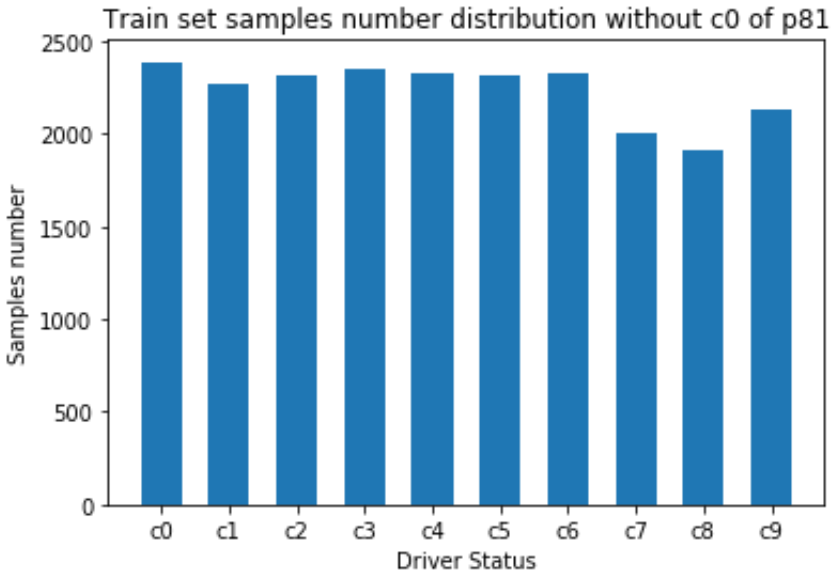

图 9 去除 p81 异常类别 c0 后，所有样本各类别图像数量分布

排除 p081 的 c0 数据后，训练集中共有 22,342 张图像。同一司机在同一类别下的图像的相似度非常高，虽然图像量可观，但缺少一些不确定性，参考了原比赛中一些优秀方案，决定使用图像拼接的方式对其进行数据增强处理，目

的是为了让模型在训练时更多地关注司机的行为，以得到更好效果的模型。图像拼接后的效果：

图 10 图像拼接以后的示例图

在对图像进行拼接增强后，训练集数据增加到 44,648 张图像，大大丰富了训练集。

除此之外，像素归一化也是非常重要的方法。不同模型的预处理方式不同，我们需要严格按照模型初始训练时的处理方法做归一化，InceptionV3, Xception 的归一化方式都是线性约化到[-1,1]， ResNet 则是减去训练集均值。另外，TensorFlow 中给出了多种归一化的方法：

1. local_response_normalization:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(0,i+n/2)} \left(a_{x,y}^{j}\right)^{2}\right)^{\beta}}$$

i 代表第 i 层

$a^{i}_{x,y}$ 就代表 第 i 层的 (x,y)位置所对应的值

n 个相邻 feature maps.

K, α , n, β 是 hyper parameters

可以看出,这个函数的功能是, $a^{i}_{x,y}$ 需要用他的相邻的 map 的同位置的值进行 normalization。

2. batch_normalization

$$\frac{\gamma\ (x-\mu)}{max\ (\sigma,\varepsilon)} + \beta$$

参考以下源代码：

```
@tf_export(v1=['layers.batch_normalization'])
def batch_normalization
```

```
  """Functional interface for the batch normalization layer.
  Reference: http://arxiv.org/abs/1502.03167
  "Batch Normalization: Accelerating Deep Network Training by Reducing
  Internal Covariate Shift"
  Sergey Ioffe, Christian Szegedy
  Note: when training, the moving_mean and moving_variance need to be
updated.
  By default the update ops are placed in `tf.GraphKeys.UPDATE_OPS`, so
they
  need to be added as a dependency to the `train_op`. Also, be sure to add
  any batch_normalization ops before getting the update_ops collection.
  Otherwise, update_ops will be empty, and training/inference will not
work
  """
```

3. moments

   参考以下源代码：

```
@tf_export("nn.moments")
def moments(
    x,
    axes,
    shift=None,  # pylint: disable=unused-argument
    name=None,
    keep_dims=False):
  """Calculate the mean and variance of `x`.
```

执行过程

该项目执行时将使用 Keras 框架中各带有 imagenet 预训练权重的深度学习模型，如：InceptionV3、Xception、ResNet50、DenseNet201。使用这些模型进行训练时都将排除掉原模型中的顶层全边接层，加入全局平均池化层对模型的高层特征向量进行降维，并加入 Dropout 层，Dropout 层将在训练过程中按一定的概率将网络中的神经元暂时丢弃，减少模型的训练参数，防止过拟合。经过多次测试 Dropout 选择在 0.5 最合适，Dropout 参数过大，丢弃率过高，学习速度慢，同样的精度需要的训练代数更多，容易欠拟合；Dropout 参数过小，丢弃率过低，学习速度快，模型更早进入过拟合状态。

在模型的最顶部加入 10 分类的全连接层，去掉全连接层的偏移项，并加入 L2 正则对训练参数进行惩罚，以此来防止过拟合。

因数据集较多，为了减轻资源的负载压力， fit 训练数据时使用 ImageDataGenerator 来分批 fit 图像，并对图像做旋转、缩放、翻转等数据增强处理

完善

验证集分割

使用司机 ID 来做验证集分割。如果同一司机的图像数据在测试集和验证集中都存在，可能会影响验证集上测试的评分。为了更加客观的评价模型，根据司机 ID 来切分数据会更合理。通过软连接的方式将图像链接到新的目录，可以减少存储以及提升计算效率。

IV.结果

单模型分析

通过训练 InceptionV3 模型，来得到优化器、学习率等参数在防止模型过拟合方面的作用。

表 1 InceptionV3 训练过程数据

	实验 1	实验 2	实验 3	实验 4	实验 5	实验 6
是否使用拼接图像	否	否	否	是	是	是
优化器	SGD	SGD	SGD	SGD	Adam	Adam
学习率	0.0003	0.0002	0.0002	0.0003	0.0001	0.0001
衰减	9.00E-08	9.00E-08	2.00E-07	3.00E-07	6.00E-08	2.00E-08
全连接层偏置项	未排除	未排除	未排除	未排除	未排除	未排除
L2 正则化	否	否	否	否	是	是
val_loss	0.8(过拟合)	0.7 (过拟合)	0.5 (过拟合)	0.11	0.15	0.02

通过多组实验对比，降低学习率，提高衰减，排除全连接层偏置项是可以防止过拟合的。拼接图像可以有效的增强模型的泛化能力。

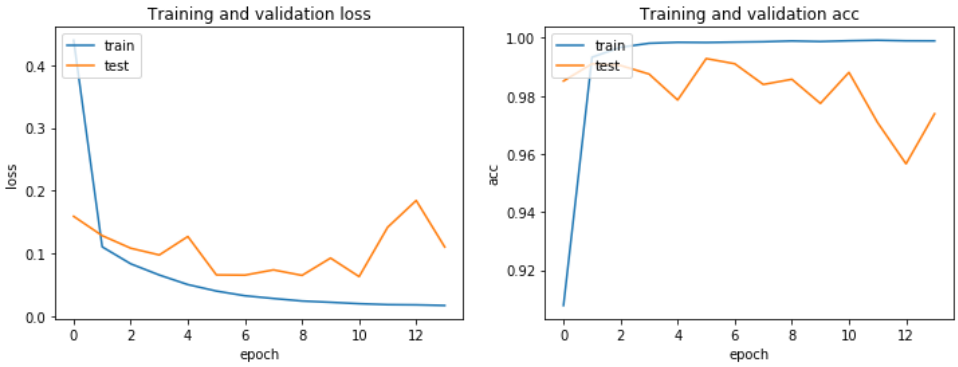

图 11 DenseNet201 Training and Validation loss

上图为 DenseNet201 的 train val loss 图，训练参数为：

表 2 DenseNet201 训练参数

模型	DenseNet201
优化器	Adam
学习率	0.00003
衰减	1.00E-08
batch_size	32
kaggle 分数	0.30165

可以看到 val loss 在降低到 0.1 附近时相对稳定，但在 epoch 10 的时候出现了明显的波动，说明无法继续拟合出更好的效果，出现了过拟合现象，可以通过进一步降低学习率、尝试正则化、调整 dropout 等方式来防止过拟合。

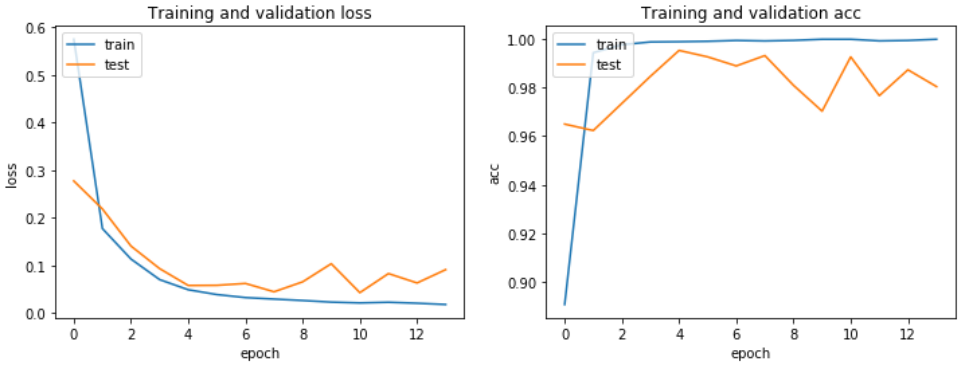

图 12 Xception Training and Validation loss

上图为 Xception 的 train val loss 图，训练参数为：

表 3 Xception 训练参数

模型	Xception

优化器	Adam
学习率	0.00003
衰减	1.00E-08
batch_size	32
kaggle 分数	0.26151

可以看到在该参数方案下，val loss 降低到 0.1 之前比较稳定，在 0.1 附近出现小幅波动，总体还是比较稳定的，并且没有出现明显的过拟合现象。

表 4 多个模型的效果对比

模型	InceptionV3	Xception	ResNet50	DenseNet201
优化器	Adam	Adam	Adam	Adam
学习率	0.0001	0.00003	0.00005	0.00003
衰减	2.00E-08	1.00E-08	2.00E-08	1.00E-08
batch_size	32	32	32	32
kaggle 分数	0.31175	0.26151	0.37406	0.30165

通过对比可得出，单模型中 Xception 的效果最好。虽然参数仍不是单模型最优的效果方案，但可以看出通过单模型想达到预期的目标有一定难度，因此需要考虑使用融合模型来进一步提升模型的效果。

融合模型分析

表 5 不同的融合方案的结果对比

	实验 1	实验 2	实验 3
模型	InceptionV3 Densenet201	InceptionV3 Xception Densenet201	InceptionV3 Xception InceptionResnetV2 Densenet201
kaggle 得分	0.23062	0.2037	0.21049

做融合模型分析时，需要去掉最后的全连接层，且在外层增加 Dropout 层级全连接分类层。通过对比发现，实验 2：InceptionV3、Xception、Densenet201 三种模型的融合方案效果最好，kaggle 得分为 0.2037，低于目标：0.25634。

Submission and Description	Private Score	Public Score	Use for Final Score
mix_pred.csv 4 hours ago by onroad mix model 2	0.23062	0.23444	☐
mix_pred.csv 4 hours ago by onroad mix model 1	0.20370	0.23426	☐
mix_pred.csv 4 hours ago by onroad mix model	0.21049	0.23359	☐

图 13 融合模型在 Kaggle 上的得分

V.项目结论

卷积神经网络的参数调优、模型选择是一个很重要的方面，但更基础的是需要对数据及数据处理的方法有充分的了解，例如：司机数量少导致司机导致图像内容会过于一致，缺少多样性，因此需要根据司机来筛选验证集。再如：打标数据中存在误判及各类别数据分布不均的情况。通过数据增强的手段，包括：拼接数据、旋转、缩放等，可以有效的增强模型的泛化能力，使模型的预测结果有了很大的提升，才得以达到预期的目标。

另外，我的工作行业是互联网金融，职责是评价用户的信用，映射到我的工作中，该项目给了我一些启发。以往的机器学习方法需要严重依赖专家意见，从运营商、淘宝、信用卡等事务型的数据中提取统计型的特征，进而通过机器学习的方法来做分类，这种方法存在的缺点是难以甄别养出来的假资料。我的想法是：将运营商、淘宝、信用卡等具有时间戳的事务型数据，转变为一张图像，再用卷积神经网络去学习。把 6 个月的数据，转化为 144（十分钟一区间）x 180（180 天）的图像，内容是将每一条事务性的数据调整为三元组（动作、对方、地点），通过转换可以表达出数据中更细的内容，说不定可以得到更好的结果，仅是个想法，还没有验证。

需要作出的改进

可以做的改进还有很多，主要集中在数据处理和模型调优两个大的方面。数据处理方面可以做的更细致，图像拼接明显有拼的不准确的情况，可能会影

响拟合效果，应对拼接的图像做一些初步的评价甚至是过滤。其他的数据增强的手段尝试也没有那么多，可以尝试的更多。

另外在模型方面，单模型经过调优应该也可以达到 top10%的效果，可以调节的参数组合仍有很多。另外就是融合模型也是一个非常重要的方向，通过融合模型应该可以得到更好的结果。

参考文献

[1]Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567, 2015.
[2]François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. arXiv prepr int arXiv:1610.02357, 2016.
[3]Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv prepr int arXiv:1602.07261, 2016.
[4]Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arXiv prepr int arXiv:1512.03385, 2015.
[5]Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. arXiv prepr int arXiv:1608.06993v5. 2016.