机器学习纳米学位

毕业项目

龚宇海

2018年5月4日

I. 问题的定义

项目概述

猫狗大战来自于Kaggle上举办的一次有趣的竞赛，目标是训练一个模型，然后通过上传一张猫或狗的照片，来分辨它是猫，还是狗，涉及到机器学习领域中的深度学习和图像分析方向。

项目需要使用到卷积神经网络（Convolutional Neural Network, CNN），是神经网络的一种，它在图像识别和分类领域已被证明非常有效。在卷积神经网络中，卷积的主要目的是从输入图像中提取特征。通过使用输入数据中的小方块来学习图像特征，卷积保留了像素间的空间关系。

一般的神经元是由输入值和激活函数给出感知器最终的输出。如图1-1所示：

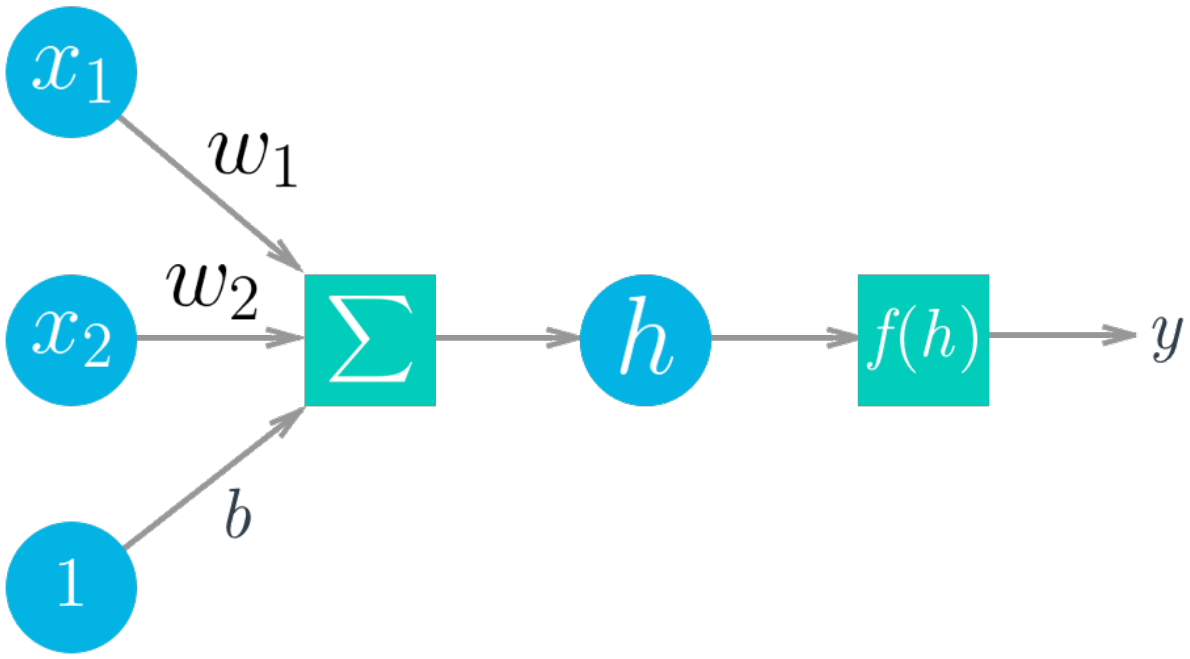

图1-1 图片引用udacity

输入值可以是由权重、输入和偏置项的线性函数组成，如**sigmoid函数**：

sigmoid(x) = 1/(1+e−x)

另一种**ReLU函数**:

ReLU(x) = max(0, x)

ReLU函数被更多的用于神经网络的激活函数。

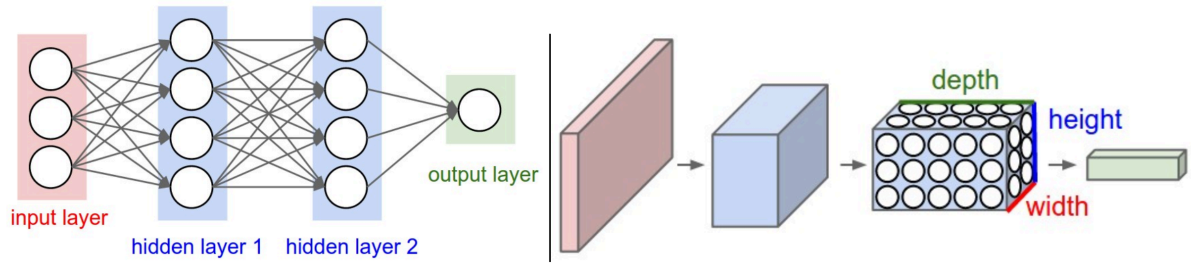

图1-2 图片引用cs231n

由多个神经元组成了神经网络，如图1-2左侧所示。在输入和输出之间的都是隐藏层，隐藏层中经过大量的计算，最后输出得到结果。图1-2右侧部分是卷积神经网络，它利用输入是图片的特点，把神经元设计成三个维度： **width**, **height**, **depth**，depth是用来描述神经元 。比如输入的图片大小是 32 × 32 × 3 (rgb)，那么输入神经元就也具有 32×32×3 的维度。这样的一种网络结构，正好对应图片分类的需求。

构建一个卷积网络需要3个最基本的层，分别是**卷积层（Convolutional Layer）**, **池化层（Pooling Layer）**，**全连接层（Fully-Connected Layer）** 。

卷积神经网路中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法优化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征。

池化层得作用在于减少输出值的大小和避免过拟合，通过特征切分取平均值或最大值。

全连接层会把所有局部特征结合变成全局特征，用来计算最后每一类的得分。

项目的数据集由Kaggle竞赛项目中提供，训练文件夹包含12,500张标记为猫的图像和12,500张标记为狗的图像。测试文件夹包含12,500个图像，根据数字ID命名，未做标记。对于测试集中的每个图像，模型需要预测图像是狗的概率（1=狗，0=猫），很明显，这是一个二分类问题。项目的目的是尽可能的提高准确的预测结果。

问题陈述

从Kaggle上获得的数据来自于真实世界中不同地区不同人所拍摄的猫或狗的图像。图像的分辨率质量参差不齐，猫和狗的形态各异，品种多样，背景环境也非常复杂，包括可能含有的异常图片，比如图像中根本就没有猫或狗，而是人或建筑物。这些都增加了分类的难度。

评价指标

训练一个可以有效运行的神经网络，需要使用到损失函数。对于二分类问题，常使用交叉熵作为损失函数对模型的参数求梯度进行更新：

$$\$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]\$$

其中n是样本数量，y是标签，y=1表示狗，y=0表示猫，$\hat{y}$ 表示样本为狗的概率。

损失函数的值越小，说明模型对于数据集的分类效果越好。

II. 分析

数据的探索

项目数据集有两个子集，分别是训练数据集和测试数据集。其中训练数据集中的图片按照`'(dog/cat).(num).jpg'`，如`'dog.132.jpg'`或`'cat.435.jpg'`，相当于对图片做了标签，明确说明图片中的是狗还是猫。

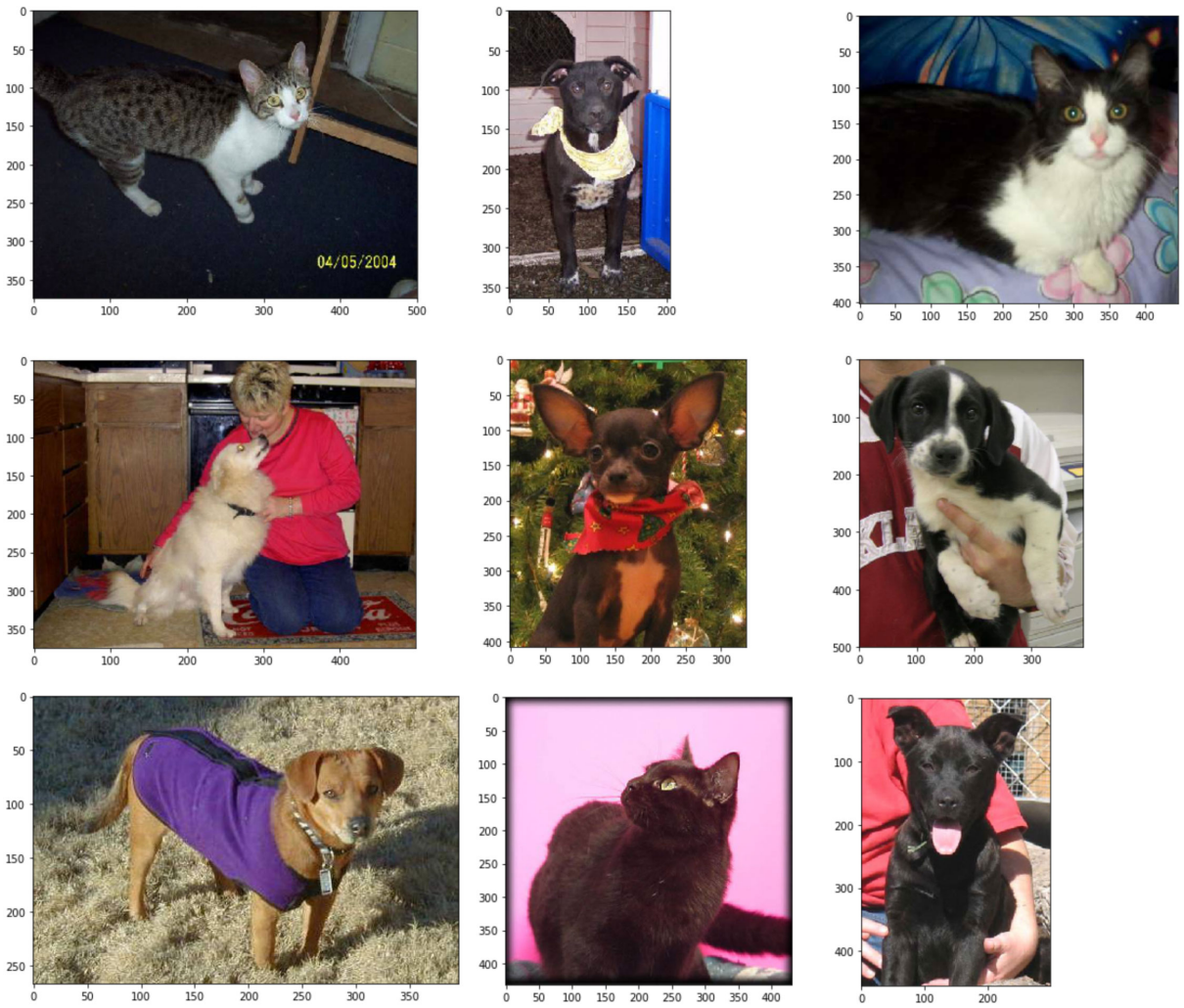

图2-1 train中的部分图片

训练集总共有25000张图片，猫和狗各12500张，分别占一半。从图2-1中可以看出，图片拍摄的背景丰富多样，拍摄的光线明暗不均，图片的大小也是各不相同，图2-2给出了训练集中图片大小的分布情况，不过大部分图片用肉眼来看，还是可以直观的分辨出猫狗的。

探索性可视化

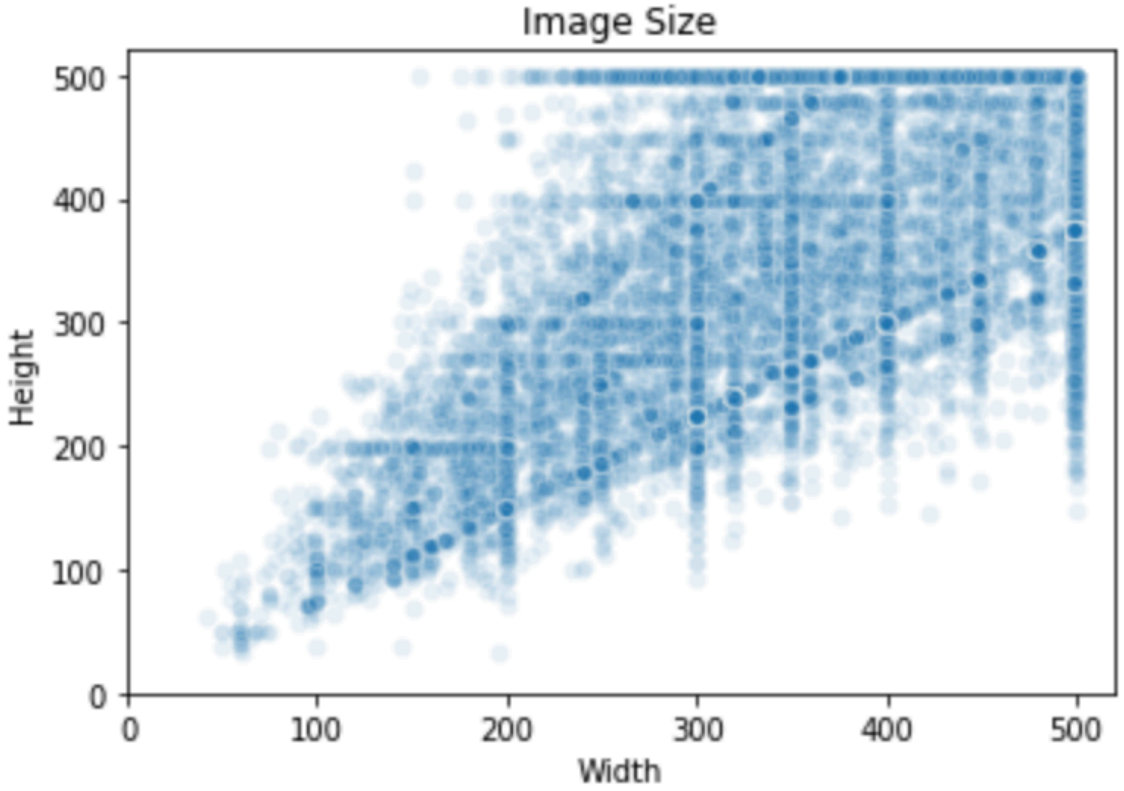

图2-2 图片大小分布

对于训练模型来讲，图片大小的不一致，会给构建模型带来很大的困难，所以在构建模型之前，需用对所有的图片的尺寸进行统一化伸缩。使用openCV对图片进行处理，在训练集中，大部分的数据都可以作为神经网络的特征提取，但也有部分的异常值，如`dog.1773.jpg`、`dog.12376.jpg`。

图2-3 异常图片，左dog.1773.jpg，右dog.12376.jpg

因为是人为上传的图片，最初标记这些图片的也是人为操作的，难免会有一些错误信息。在实际生活中，遇到的异常值可能还要多，这就需要模型具备一定容错能力。在数据量更大的时候，一个模型的好坏，容错能力也至关重要。这里对这些异常值不做处理，但我们知道这种情况在很多时候是实际存在的，而且很难避免。

算法和技术

项目使用迁移学习(Transfer learning) 的方法来构建猫狗分类的神经网络。迁移学习顾名思义就是就是把已学训练好的模型参数迁移到新的模型来帮助新模型训练。具体地，在迁移学习中，我们已有的知识叫做源域(source domain)，要学习的新知识叫目标域(target domain)。迁移学习研究如何把源域的知识迁移到目标域上。特别地，在机器学习领域中，迁移学习研究如何将已有模型应用到新的不同的、但是有一定关联的领域中。传统机器学习在应对数据的分布、维度，以及模型的输出变化等任务时，模型不够灵活、结果不够好，而迁移学习放松了这些假设。在数据分布、特征维度以及模型输出变化条件下，有机地利用源域中的知识来对目标域更好地建模。另外，在有标定数据缺乏的情况下，迁移学习可以很好地利用相关领域有标定的数据完成数据的标定。

迁移学习按照学习方式可以分为基于样本的迁移，基于特征的迁移，基于模型的迁移，以及基于关系的迁移。基于样本的迁移通过对源域中有标定样本的加权利用完成知识迁移；基于特征的迁移通过将源域和目标域映射到相同的空间（或者将其中之一映射到另一个的空间中）并最小化源域和目标域的距离来完成知识迁移；基于模型的迁移将源域和目标域的模型与样本结合起来调整模型的参数；基于关系的迁移则通过在源域中学习概念之间的关系，然后将其类比到目标域中，完成知识的迁移[1]。

ImageNet上已经完成训练的神经网络模型有很多，它们对自然图像特征具备很好的提取和筛选能力。相对于本项目中，会产生干扰问题的背景环境、人物和其他自然实物，这些经过训练的模型都能够很好的将主要物体提取出来，然后可以迁移到新的模型中训练。这样的好处是可以不用自己从零开始构建和训练模型，况且即使是自己从零开始构建的模型，也不一定会比现有的更好，所以选择迁移模型，建立在现有的工具上开始训练新模型会是一个不粗的选择。

引用：[1].什么是迁移学习 (Transfer Learning)？这个领域历史发展前景如何？ - 王晋东不在家的回答 - 知乎

基准模型

在使用Keras深度学习库的官网上可以看到，已经有人给我们提供了一些可使用的模型，并且标注了模型的准确率，利用现有的模型以及预训练出的权重，就能很好的表示猫和狗的特征。考虑到准确率，模型大小等因素，基准模型选择的是ResNet50[1]、InceptionV3[2]、和Xception[3]这个三个模型组合。结构图如下：

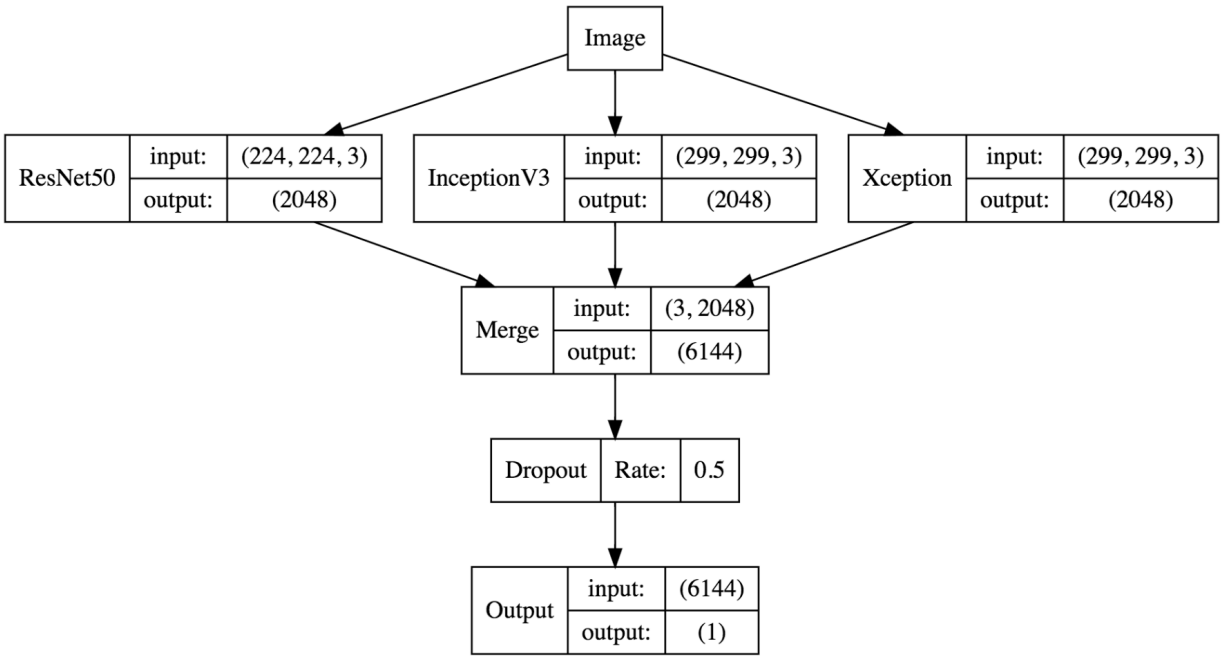

图2-4 基准模型结构图

各个子模型的初始权重采用ImageNet权重，禁用了各个模型分类器的Top层，将全局平均池化层作为输出层。在复现模型的过程中，构建好模型后直接训练模型，第一次就在Kaggle上得到0.04008的分数。相对于项目示例中作者获得0.04141略微好一点，因此希望能够有更好一点的成绩。

III. 方法

数据预处理

基于现有的模型进行迁移计算，输入的数据需要与模型相匹配，ResNet50、InceptionV3和Xception这三种模型对应输入的数据是不同的，需要满足各个模型的默认输入要求。图像数据的表现形式在计算机中表现为：(weight, height, channel), 每个元素的取值范围在0～255之间。预训练模型先进行归一化处理，处理完成后作为输入数据，传递给模型；添加全局平均池化层GlobalAveragePooling2D，作为模型最后的输出数据。ImageDataGenerator作为keras提供的图像预处理函数，可以将传递过来的图像数据根据指定的大小，传递给模型需要的输入数据。

查阅Keras API，Application模块为部分模型提供预处理好的输入值，如InceptionV3和Xception。在训练模型时，直接使用preprocess_input作为输入参数。

执行过程

根据预先构想好的迁移模型方案，分别训练并导出模型ResNet50、InceptionV3和Xception三个处理好的模型，然后开始聚合模型。

Dropout(0.5)->acc=0.9650

Dropout(0.25)->acc=0.9816

```
model.fit(X_train, y_train, batch_size=16, epochs=5, validation_data=(X_valid, y_valid))
```

```
Train on 20000 samples, validate on 5000 samples
Epoch 1/5
20000/20000 [==============================] - 229s 11ms/step - loss: 0.1194 - acc: 0.9509 - val_loss: 0.0685 - val_a
cc: 0.9750
Epoch 2/5
20000/20000 [==============================] - 227s 11ms/step - loss: 0.0817 - acc: 0.9691 - val_loss: 0.0502 - val_a
cc: 0.9816
Epoch 3/5
20000/20000 [==============================] - 227s 11ms/step - loss: 0.0786 - acc: 0.9700 - val_loss: 0.0647 - val_a
cc: 0.9770
Epoch 4/5
20000/20000 [==============================] - 227s 11ms/step - loss: 0.0754 - acc: 0.9702 - val_loss: 0.0848 - val_a
cc: 0.9740
Epoch 5/5
20000/20000 [==============================] - 227s 11ms/step - loss: 0.0768 - acc: 0.9702 - val_loss: 0.0770 - val_a
cc: 0.9768

<keras.callbacks.History at 0x7f31c32849b0>

 20000/20000 [==============================] - 226s 11ms/step - loss: 0.0712 - acc: 0.9728 - val_loss: 0.0688 - val_a
 cc: 0.9780
 Epoch 2/5
 20000/20000 [==============================] - 227s 11ms/step - loss: 0.0678 - acc: 0.9741 - val_loss: 0.0639 - val_a
 cc: 0.9792
 Epoch 3/5
 20000/20000 [==============================] - 227s 11ms/step - loss: 0.0706 - acc: 0.9724 - val_loss: 0.0688 - val_a
 cc: 0.9788
 Epoch 4/5
 20000/20000 [==============================] - 227s 11ms/step - loss: 0.0691 - acc: 0.9736 - val_loss: 0.0600 - val_a
 cc: 0.9810
 Epoch 5/5
 20000/20000 [==============================] - 227s 11ms/step - loss: 0.0687 - acc: 0.9746 - val_loss: 0.0595 - val_a
 cc: 0.9802
```

完善

在考虑权重更新的时候，检查权重点ModelCheckpoint函数可以帮助在每次的epoch中，保存获得分数最好的权重，然后通过获得最优权重来训练模型，最后获得了0.03843的成绩。

!image-20180514012158058

根据毕业项目要求，kaggle_train中所构造的模型即已达到Kaggle的前10%的分数，得分小于原作者的0.04141。

IV. 结果

（大概 2-3 页）

模型的评价与验证

在这一部分，你需要对你得出的最终模型的各种技术质量进行详尽的评价。最终模型是怎么得出来的，为什么它会被选为最佳需要清晰地描述。你也需要对模型和结果可靠性作出验证分析，譬如对输入数据或环境的一些操控是否会对结果产生影响（敏感性分析sensitivity analysis）。一些需要考虑的问题：

- *最终的模型是否合理，跟期待的结果是否一致？最后的各种参数是否合理？*
- *模型是否对于这个问题是否足够稳健可靠？训练数据或输入的一些微小的改变是否会极大影响结果？（鲁棒性）*
- *这个模型得出的结果是否可信？*

合理性分析

在这个部分，你需要利用一些统计分析，把你的最终模型得到的结果与你的前面设定的基准模型进行对比。你也分析你的最终模型和结果是否确确实实解决了你在这个项目里设定的问题。你需要考虑：

- *最终结果对比你的基准模型表现得更好还是有所逊色？*
- *你是否详尽地分析和讨论了最终结果？*
- *最终结果是不是确确实实解决了问题？*

V. 项目结论

（大概 1-2 页）

结果可视化

在这一部分，你需要用可视化的方式展示项目中需要强调的重要技术特性。至于什么形式，你可以自由把握，但需要表达出一个关于这个项目重要的结论和特点，并对此作出讨论。一些需要考虑的：

- *你是否对一个与问题，数据集，输入数据，或结果相关的，重要的技术特性进行了可视化？*
- *可视化结果是否详尽的分析讨论了？*
- *绘图的坐标轴，标题，基准面是不是清晰定义了？*

对项目的思考

在这一部分，你需要从头到尾总结一下整个问题的解决方案，讨论其中你认为有趣或困难的地方。从整体来反思一下整个项目，确保自己对整个流程是明确掌握的。需要考虑：

- *你是否详尽总结了项目的整个流程?*
- *项目里有哪些比较有意思的地方?*
- *项目里有哪些比较困难的地方?*
- *最终模型和结果是否符合你对这个问题的期望?它可以在通用的场景下解决这些类型的问题吗?*

需要作出的改进

在这一部分，你需要讨论你可以怎么样去完善你执行流程中的某一方面。例如考虑一下你的操作的方法是否可以进一步推广，泛化，有没有需要作出变更的地方。你并不需要确实作出这些改进，不过你应能够讨论这些改进可能对结果的影响，并与现有结果进行比较。一些需要考虑的问题：

- *是否可以有算法和技术层面的进一步的完善?*
- *是否有一些你了解到，但是你还没能够实践的算法和技术?*
- *如果将你最终模型作为新的基准，你认为还能有更好的解决方案吗?*