机器学习(进阶)纳米学位毕业项目开题报告

项目名称：猫狗大战
作者：Zax
提交日期：180618

一、项目背景

本项目名为“猫狗大战”，原为 Kaggle 上的一个已经结束的竞赛项目[1]。数据源来自 Kaggle 官方网站。

本项目重点在于将深度学习应用于图像识别。“猫狗大战”是个典型的“二分类”问题，预测结果只有“猫”与“狗”两个类别。在完成构建模型后，最终需要针对 Kaggle 所提供的 test 数据集(或图片集)进行预测，并写入一个 csv 文件，提交至 Kaggle 官方网站获取评估分数。评估分数是判断模型是否满足毕业项目要求的唯一标准。

“二分类”问题是所有分类中最为基本的问题，是解决“多分类”问题的基础。而现实生活中更多的是“多分类”问题。

二、问题描述

本项目需要构建一个监督学习的深度学习网络模型，并根据 Kaggle 提供的数据集（分为 train, test 两部分，训练只使用 train 数据集）对模型进行训练，然后使用训练完毕的模型对 test 数据集所有图片进行预测，并按 Kaggle 要求的格式（“sample_submission.csv”，在 Kaggle 官网随 Kaggle 数据集一起提供）在 Kaggle 官方网站提交一个 csv 文件，用于评估模型的分数。

根据项目要求，Kaggle 的评估分数需要达到 kaggle Public Leaderboard 前 10%，根据 Kaggle 官网“猫狗大战”已有的分数排名[2]，10%位置的 score 大约为 0.06149，也即本项目中最终评分不应低于 0.06149，需要说明的是评分数值越小分数越高。

三、输入数据

数据集分为三部分[3]：

- train.zip：训练数据集，一共有 25000 张图片，用于训练搭建的深度学习网络模型；
- test.zip：测试数据集，一共还有 12500 张图片，用于评估模型；
- sample_submission.csv：提交文件的模板，一共有 12500 行数据，用于向 Kaggle 提交预测结果，以便 Kaggle 对模型进行打分。

[1] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
[2] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard
[3] https://www.kaggle.com/c/dogs-vs-cats/data

四、解决方案

本项目计划使用迁移学习和模型融合的方法生成一个预测模型。在进行模型构建之前需要对图片进行清洗，以便清除异常图片，主要的异常图片可能有：非猫非狗的图片、过于模糊的图片、同时出现猫和狗的图片、非实体的猫狗图片（卡通形象等）。

本项目计划执行的步骤：

- 图片清洗：清洗后判断为异常的图片移动到 DATASET/exception 目录下。
  拟使用多个模型对图片进行清洗，取较为平衡（尽可能多地包含异常图片，尽可能少地包含误判的图片，可能需要人工核对）的 top 值，对多个模型获取到的异常图片取并集，作为最终需要清洗掉的异常图片集。
- 搭建模型：
  1） 选择基准模型：InceptionResNetV2、DenseNet201、Xception。
  以上三个基准模型都是 Keras 自带的预训练模型[4]，属于 Keras 的应用模块。Keras 提供了带有预训练权值的深度学习模型，这些模型可以用来进行预测、特征提取和微调（fine-tune）。
  2） 训练特征向量：输入为 train 数据集，采用 1）所述的基准模型分别对输入数据集进行训练，并将训练完成后的特征向量保存在 saved_models 文件夹中。
  3） 载入特征向量：主要目的是将三个模型的特征向量融合为一个完整的向量空间，完成这一步骤后形成新的输入向量和标签，并对输入的特征进行混洗（shuffle）。
  4） 编译模型：选择优化器 optimizer：'adadelta'，损失函数 loss：'binary_crossentropy'，评价指标 metrics：'accuracy'。
  5） 训练模型：对融合后的新模型进行训练，并保存最佳的模型文件（命名为：weights.final_model.hdf5），位置位于 saved_model 文件夹。
  6） 评价模型：使用 Epochs-Loss 曲线及 Epochs-Acc 曲线对模型进行评价，其中 Epochs 为完整训练的轮数，Loss 为评价指标的损失分数，分为 loss（对训练集划分为 train 和 valid 后，train 对应的损失分数）和 val_loss（valid 数据集上的损失分数），Acc 为评价指标的准确率，分为 acc(对应划分后的 train 数据集)和 val_acc（对应划分后的 valid 数据集）。Epochs 为横坐标，各 metrics 指标为纵坐标，观察曲线特征，比如收敛程度、loss 与 val_loss 位置比较、acc 与 val_acc 位置比较，判断模型的曲线是否收敛、是否过拟合或欠拟合。
- 预测：使用上面搭建好的模型对 test 数据集中的图片进行预测，并套用 Kaggle 提供的模板文件 sample_submission.csv 将预测结果写入一个新的 csv 文件（初步命名为 pred_submission.csv）。
- 提交预测文件，获取 Kaggle 评分，将评分结果与项目要求进行对照，如不满足要求，需要对上述 1)~6)的过程进行检查和修改，并进行迭代，直到达到项目要求为止。

五、基准模型

本项目一共使用了三个基准模型：InceptionResNetV2、DenseNet201、Xception。其中：

- Xception
- InceptionResNetV2

[4] https://keras.io/zh/applications/

- DenseNet201

基准模型的介绍在“解决方案”中已有描述，故此处不再赘述。

六、评价指标

本项目使用的评价指标有：

- 准确率 Accuracy

  准确率用于定义分类模型的性能，主要描述了预测结果在多大百分比上是准确的，一般来说，同等情况下，准确率越高，分类模型性能越好。

  Accuracy 的定义：

$$accuracy = \frac{n}{N} * 100\%$$

  其中：n 为正确预测猫或狗的图片数量，N 为 test 数据集中所有图片数量。

- 损失函数 LogLoss

  LogLoss 是一种基于交叉熵的损失函数[5]，用于评价错误分类所带来的损失，当预测结果是正确的时候，LogLoss 应该是 0，当预测结果为错误的时候，LogLoss 应该是 1。对分类器来说，LogLoss 越小越好。在本项目中 LogLoss 表示为 Loss 评分，作为最终提交到 Kaggle 的结果。

  LogLoss 定义为：

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

其中：n为测试集图片数量，$\hat{y}_i$为测试集中图片i预测为预测值的可能性，y_i为测试集中图片i的实际结果，即本来应是猫或狗，也即标签类别，若标签为猫，预测为猫，则y_i为1，否则为0。这里涉及log0问题，需要对预测概率进行处理，使得可以对结果进行log计算，比如对最小值和最大值都进行限定： $y_i = y_i$.clip(min = 0.005, max = 0.995)。

七、设计大纲

第一步： 数据归类。从 Kaggle 下载数据集并解压缩，分为三个文件：train.zip, test.zip, sample_submission.csv，解压缩后在项目文件夹中生成 train 文件夹、test 文件夹。然后建立一个 DATASET 文件夹，作为分类完毕后供模型使用的数据文件夹。DATASET 文件夹中建立 train 文件夹和 validation 文件夹，在两个文件夹中又分别建立 cats 和 dogs 文件夹。异常图片也放在 DATASET 下的文件夹 exception 中，另建一个 submission 文件夹，用于存储提交的 csv 数据文件，项目中 DATASET 大致目录结构如下：

```
DATASET/
    ImageNetFullClasses.csv
    exception/
    submission/
        sample_submission.csv
    test/
```

[5] https://en.wikipedia.org/wiki/Loss_functions_for_classification#Cross_entropy_loss_(Log_Loss)

```
        pics/
    train/
        cats/
        dogs/
```

第二步：数据清洗。主要是找出异常图片，判断为异常的图片移动到 DATASET/exception 目录下。拟使用多个模型对图片进行清洗，取较为平衡（尽可能多地包含异常图片，尽可能少地包含误判的图片，可能需要人工核对）的 top 值，对多个模型获取到的异常图片取并集，作为最终需要清洗掉的异常图片集。

第三步：搭建模型。完成第一步和第二步后，可以用来建模的图片应该都位于 DATASET/train 下，并按 cats 和 dogs 分好类。

搭建模型的过程大致如下：

1）选择基准模型：可选模型较多，本项目中选取几个到目前为止得分比较高的基准模型[6]：InceptionResNetV2、DenseNet201、Xception。这些模型都是 Keras 提供的预训练模型（原模型来自于 ImageNet，而 ImageNet 一共有 1000 个分类），便于用来进行预测、特征提取和 fine-tune。

2）训练特征向量：输入为 train 数据集，采用 1）所述的基准模型分别进行训练，并将训练完成后的特征向量保存在 saved_models 文件夹中。

3）载入特征向量：主要目的是将三个模型的特征向量融合为一个完整的向量空间，完成这一步骤后形成新的输入向量和标签，并对输入的特征进行混洗（shuffle）。

4）编译模型：选择优化器 optimizer：'adadelta'，损失函数 loss：'binary_crossentropy'，评价指标 metrics：'accuracy'。

5）训练模型：对融合后的新模型进行训练，并保存最佳的模型参数文件（命名为：weights.final_model.hdf5），位置位于 saved_model 文件夹。

6）调参：可调的参数主要有 Epochs，dropout，learning-rate(即 lr)。其中主要调节 Epochs 和 dropout。lr 使用较为平衡的默认值即可(Optimizer 使用 Adadelta，建议使用优化器的默认参数[7])。Epochs 可使用 Keras 提供的 EarlyStopping 方法[8]。dropout 参数常用范围为 0~1 之间的小数[9]，常用的调节范围为 0.1~0.9，可以考虑使用循环从 0.1 到 0.9 逐个训练最终模型，取最佳的 dropout 值，可以参考的经验值一般在 0.2~0.5 之间。

7）评价模型：使用 Epochs-Loss 曲线及 Epochs-Acc 曲线对模型进行评价，其中 Epochs 为完整训练的轮数，Loss 为评价指标的损失分数，分为 loss（对训练集划分为 train 和 valid 后，train 对应的损失分数）和 val_loss（valid 数据集上的损失分数），Acc 为评价指标的准确率，分为 acc(对应划分后的 train 数据集)和 val_acc（对应划分后的 valid 数据集）。Epochs 为横坐标，各 metrics 指标为纵坐标，观察曲线特征，比如收敛程度、loss 与 val_loss 位置比较、acc 与 val_acc 位置比较，判断模型的结果是否收敛、是否过拟合或欠拟合。

第四步：预测。使用上面搭建好的模型对 test 数据集中的图片进行预测，并套用 Kaggle 提供的模板文件 sample_submission.csv 将预测结果写入一个新的 csv 文件（初步命名为

[6] https://keras.io/zh/applications/
[7] https://keras.io/zh/optimizers/
[8] https://keras.io/callbacks/
[9] https://keras.io/layers/core/#dropout

pred_submission.csv）。

第五步：提交预测文件，获取 Kaggle 评分。将评分结果与项目要求进行对照，如不满足要求，需对上述步骤进行检查和修改，并进行迭代，直到达到项目要求为止。

八、引用

1. https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
2. https://keras.io/zh/applications/
3. https://en.wikipedia.org/wiki/Loss_functions_for_classification#Cross_entropy_loss_(Log_Loss)
4. https://keras.io/zh/optimizers/
5. https://keras.io/callbacks/
6. https://keras.io/layers/core/#dropout