

人工智能导论第二次大作业

张蔚桐 2015011493 自55

1 程序编译与运行

程序基于TensorFlow使用Python开发，相关依赖如requirements.txt所示。可以在符合相关要求的平台上运行

受后期文件格式的调整的原因，相关资源文件和输出文件的位置有所不同，如出现此类情况请对源码相关位置进行适当修改。其中根目录下model文件夹下为数据集和固化的模型

2 黑白手写数字的识别

本模块源码在Task1文件夹下，输出的预测值在Task1/predict.txt

本模块的开发参照了TensorFlow教程上的相关代码。并对具体数据集进行了处理。

程序首先完成了对数据集的导入和处理工作，将数据集变形为适合深度学习进行处理的形式。之后程序完成了对深度学习模型的设置。

我们采用了TensorFlow教程上推荐的深度学习模型，模型采用经典的LeNet-5模型。整体为卷积-池化-卷积-池化-全连接-dropout-softmax共7层网络，考虑到了手写数字的平移不变性和旋转不变形。

其中第一层卷积卷积核大小为5*5，将28*28的输入图像卷积为32个特征，并进行池化。第二层卷积核大小不变，最终得到64个特征，并进行池化。两次池化之后全连接层将7*7的具有1024的特征的变换为1024个神经元输出。通过Dropout层来减少某一恒定特征导致的过拟合，并将最后1024个神经元输出利用softmax层输出为10个特征。

比较10个特征大小可以得到预测值。

训练过程在四核i7上（无GPU支持）迭代20,000次共计一到两个小时。由main.py完成，并将模型固化到model/frozen_model.pb中

测试过程已经在Query.py文件中封装好函数GreyQuery函数，函数接受一组查询图片并返回一组预测值，具体的查询样例在example.py中，程序预测测试集并给出测试集正确率超过了98%。这个正确率是可以接受的。

3 彩色手写图像的识别

本模块源码在Task1文件夹下，输出的预测值在Task2/predict.txt

由于题目中没有给出训练集的标签，同时通过一些可视化手段可以观察到给定的数据集质量并不是很好，其中很多数字人眼也很难识别，因此需要对图像进行预处理。

我们观察到大部分的图像数字区域和背景区出现了明显的差异，因此我们希望在一张图像中对色彩进行聚类区分哪些位置是数字的位置，哪些位置是背景的位置。

首先我们采用PCA方法对色彩进行特征提取，PCA方法找到一个正交变换使得整个数据集的方差可以从大到小排布。下面所有操作均在RGB空间上进行。首先我们计算一张图片协方差矩阵，将协方差矩阵对角化找到最大的特征值，将这个特征值对应的特征向量作为变换阵对整个数据集进行变换，得到一个28*28的一维数组，形成一个新的灰度图。根据KL变换等统计学性质，这个灰度图最大限度地保留了原来彩色图片的方差。我们将整张灰度图的阈值变化到0到255的灰度区间。之后，我们需要保证整张图的数字部分为白色，因此我们假定整张图中数字部分占少数，检查灰度图的均值，如果超过了128.则将灰度图取反。

我们注意到，在黑白图像中白色部分出现了可观的数值变化，而黑色部分基本均为0，为了在统计学上尽可能的和黑白图像相似，我们再次对原图像进行kmeans聚类分析。我们使用kmeans方法对一张图像中的色彩分为两类，具体方法是采用欧几里得距离和聚类质心进行迭代等方法，这里略去不提。分为两类之后，我们认为这两类分别是前景（数字）和背景，分别设置为0和255.之后和PCA方法相同，我们检查整幅图片的均值，并将数字部分变成白色。

我们将上述两张图片逐点取最小值，得到背景区为黑色，前景为渐变白色的处理之后的图像。并使用训练好的模型对黑白图片进行聚类分析处理。

测试过程已经在function.py文件中封装好函数RGBQuery函数，函数接受一组查询图片并返回一组预测值，具体的查询样例在example.py中，程序预测测试集并给出测试集正确率约为75%。

后期采取了一些其他的处理方式但是没有将正确率提升至75%以上。

4 其他分析和处理

我们对PCA处理过的图像进行了分析，可以看出有些图片因为本身质量就不是很好导致PCA等聚类方法效果较差，考虑可能是前景和背景没有分开

的原因。我们引入PCA之后的方差进行研究。利用测试集，我们发现仅仅进行PCA处理的图像正确率大约为50%。而测试集中超过一半的数据PCA之后灰度图像的方差均大于5000，对这部分图像的测试正确率为85%，说明方差判据的效果还是比较明显的，五分之一的图像的方差大于7500，而这部分图像的测试正确率为90%，十分之一的图像方差大于10000，而这部分图像的测试正确率甚至达到了95%以上。我们计划采用7500筛选训练集图像，产生大约10000幅图像使用彩色LeNet-5进行测试，这部分的尝试可以参见etc.py 和mainRGB.py 文件，发现训练效果不好，可能是因为网络调参的原因。由于计算资源有限，为保证作业提交，这部分内容只能暂停。

另一方面，在一个小样本数据集上的训练-测试发现，尽管采用了7500方差筛选作为“精英”样本进行训练，保证了训练集的可靠性，但是由于测试集的不可靠性（PCA第一方差过低可能是因为图像本身比较混淆或其他原因导致的），训练效果和测试效果相差较大，测试正确率不能达到70%，我们希望将前面使用的PCA-kmeans方法和这套方法联合，并增大网络深度进行测试，但是受到计算资源的限制（同时因为期末考试，CPU时间占用明显）只能先提交相关文件。