|  |  |  |
| --- | --- | --- |
| SRT计划项目个人研究报告 | | |
|  | 项目名称: | 基于 Keras 深度学习框架的多标签图像分类研究 |
|  | 申请者: | 周宽 |
|  | 学 院: | 信息科学技术学院 |
|  | 专 业: | 计算机科学与技术 |
|  | 指导教师: | 顾兴健 职称: 讲师 |
| 2018年 5月 11 日 | | |

1. 概述

我们的题目是：基于Keras深度学习框架的多标签图像分类研究，这是一年前在我们的指导老师的建议下定下的题目。关于多标签图像分类，多标签图像分类研究是当前的机器学习方面的一个热点问题，关于它的理解与研究。我们也是从无到有慢慢的摸索过来的，经过诸多材料的阅读及研究，我们决定用以tensroflow作为后端，用keras作为框架，在Linux系统上来完成我们srt的研究，刚开始环境的配置和Linux系统的安装是我们主要的任务，然而在2017年2月份，Google才正式对外发布Google TensorFlow 1.0版本，并保证本次的发布版本API接口完全满足生产环境稳定性要求，这标志着它可以正式在生产环境放心使用，所以tensorflow是非常新的库，因此我们在安装tensorflow库时并没有太多经验可以参考，这花费了我们不少时间。后来这部分搞定之后，我们着手我们的研究，用一组训练图片，你么让计算机记住他们，并且学会识别它的同类，这是我们开始研究的方面，通过阅读书籍和查找网络上资料，我们逐渐知道了方法，然而我们发现由于我们的电脑机器计算能力和开销的缘故，就算我们训练出了一些类的模型结果，其识别精度并不能很高，那么如何去得到几类精度很高的训练模型成了我们的主要研究方向，所以我们把过程放在了深度学习的迁移学习上面，通过再训练模型，只需要不大的训练集和训练次数，便能得到一个很高的识别精度，这样也提现了我们工作的意义。

1. **研究内容及过程**

环境配置好后，便从最经典的数据集Minist入手，了解Minist之后，我发现　官、方的MNIST例子里面训练数据的下载和导入都是用已经写好的脚本完成的，至于里面实现细节也没高兴去看源码，感觉写得太正式，我这个初学者不好理解，于是我便查阅资料理解其实现细节，最终了解到，关于标签分类：一般过程是这样的：假如有了训练集（用于训练的图片）及测试集（用来测试的图片） ，首先提取训练集中的有标签图片，添加多层神经网络训练这些图片，得到训练模型，最后用训练模型检验测试集，来测试模型识别的精准度。这样的话，通常需要很大的测试集和训练次数，至少要有 10000 张测试图片、10000 次最大训练次数，才能得到一个较好的图片识别精准度。然而学校提供给我们的资源极其有限，如果这样做的话难以达到有意义的训练效果，而且我们的训练集也有限，所以我们采用迁移学习的方法通过再训练模型，从而达到在一个小的训练集上得到一个较好的训练识别准确率的目的。

那么什么是迁移学习，什么是再训练呢？迁移学习 (Transfer learning) 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。考虑到大部分数据或任务是存在相关性的，所以通过迁移学习我们可以将已经学到的模型参数（也可理解为模型学到的知识）通过某种方式来分享给新模型从而加快并优化模型的学习效率不用像大多数网络那样从零学习。迁移学习的基本思路是利用预训练模型，即已经通过现成的数据集训练好的模型（这里预训练的数据集可以对应完全不同的待解问题，例如具有相同的输入，不同的输出） 。开发者需要在预训练模型中找到能够输出可复用特征（feature）的层次（layer） ，然后利用该层次的输出作为输入特征来训练那些需要参数较少的规模更小的神经网络。由于预训练模型此前已经习得了数据的组织模式（patterns） ，此这个较小规模的网络只需要学习数据中针对特定问题的特定联系就可以了。

所以预训练模型就像前人的学习大量知识和书籍后得到的经验，而我们的迁移学习，就是不断地改进完善经验。Google 为在 Imagenet 上训练神经网络，使用了很多 GPU,连续训练了几周，使用了大数据和大量计算资源，这些预训练好的模型可以看做以二进制存放的经验，使用预训练模型将给模型的训练一个很好的起点，最终也往往也取得比直接在目标数据集从 0 开始的训练更好的效果。

预训练模型大体由两部分组成，一是抽取特征的部分，二是用于预测的部分。有了预训练模型，我们的起点就比较高，因此学习速率 (learning rate) 应当调小,否则将很快将之前在大数据训练中得到的合适权重带偏，然后在小数据集上很快陷入过拟合状态，预测效果反而下降。因此训练过程中应当把预训练模型的前面很多层的权重都冻结，我们可以训练预训练模型的最后几层，或者自己再在预训练模型除预测部分的特征提取的最后加上自己建立的卷积层、池化层等，在我们的小数据集训练。通过反向传播调整特征提取部分最后几层的权重，这样就达到了既不扰乱预训练模型训练得到的大部分合适权重，又能有针对性地在我们的小数据集上训练和强化的效果。这样我们就能通过小型数据集，利用极其有限的计算资源取得理想的训练效果。

1. **技术问题及解决**

在研究中，我们也遇到了诸多技术问题和困惑：例如，最开始的开发环境配置，我的电脑安装好Linux系统后进入不了该系统，网上查了很多相关资料，又重新装了好几遍，也没能解决，最后我发现有一位同学有过相似的经历，通过他的帮助解决了我的问题。还有关于python的学习，在研究中，也遇到了一些python的问题，错综复杂，不过通过查阅资料和请教别都解决了。再有就是关于多标签深度学习模型训练中出现的过拟合问题：如何区分出全局特征和局部特征，避免出现过拟合，也是我们值得注意的一点。

1. **体会心得**

在这一年的srt项目的研究中，我们小组从开发环境的配置及工具安装，到最后的研究成果出现，其间经历了很多问题，查阅了很多资料，学习到了很多知识，最后的到了我们较满意的结果。感谢我们的指导老师顾兴建老师，在我们的研究过程中给了许多宝贵的建议和资料，给我们指明了方向，让我们少走了许多弯路。也感谢邱日组长的付出，在他的带领下，我们小组更加团结有效率的进行研究！