**远程科研项目报告**

**——手写体数字识别**

贺英豪

2019年10月11日

1. **研究背景**

随着大数据技术的发展，对大数据的相应处理技术就显得十分重要，这次研究，通过对使用docker的容器部署MNIST手写数字识别模型来识别手写数字，并在Cassandra中展示结果这一过程来实现对大数据存储以及应用等方面的学习，进而对大数据的相关知识有相应的了解。

* 1. **Docker**

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中,然后发布到任何流行的Linux机器上。Docker最大的优势在于其效率，当改变其中的镜像时，只需要改变其变化的部分，而不需要重建整个Docker镜像。本次实验实将使用Docker的容器来进行。

可以通过pip install docker命令下载docker，下载完成后通过docker --version参看其版本看是否安装成功。

如果国内网速慢可以在后面加上下列代码使用清华的源（下同）

-i https://pypi.tuna.tsinghua.edu.cn/simple

* 1. **Mnist与TensorFlow**

简单来说MNIST数据集是一个手写体数据集，这个训练集是由60000个用例组成。通过这个训练集我们可以实现对手写体数字的识别。

TensorFlow是一个基于数据流编程（dataflow programming）的符号数学系统，被广泛应用于各类机器学习（machine learning）算法的编程实现。本次将使用TensorFlow实现对mnist模型的保存于调用。

通过pip install tensorflow

conda install -c conda-forge tensorflow

完成TensorFlow的下载。

**1.3 Cassandra**

Cassandra是一套开源分布式NoSQL数据库系统，具有灵活，可扩展性等诸多优点，本项目将使用Cassandra来进行结果的记录与存储。

通过dokcer pull cassandra来讲Cassandra容器拉到本地。这个过程比较慢（国内网络问题），如果不成功也可以使用如下命令：

docker pull [registry.docker-cn.com/library/cassandra:latest](http://registry.docker-cn.com/library/cassandra:latest" \t "https://mail.qq.com/cgi-bin/_blank)

1. **具体实现**

**2.1 虚拟环境配置**

由于docker与Windows的兼容问题，本次项目我将在Ubuntu系统上来完成，虽然虚拟环境的配置与项目本身关系不大，但是通过配置好虚拟环境，更有利于Linux系统上的后续操作。

**1安装：pip install virtualenv**

**2创建虚拟环境：virtualenv hyh（环境名） -p /usr/bin/python3.7**

**3进入虚拟环境：source bin/activate**

创建好后下次进入只需要 workon hyh即可。

**2.2 Mnist模型保存**

通过在mnis\_deep.py或者mnist\_softmax.py中加入如下代码实现文件的保存：

saver = tf.train.Saver()

with tf.Session() as sess:

Init = tf.global\_varianles\_initializer()

sess run init

#在""中填入保存路径

save\_path = saver.save(sess, "./modle.ckpt")

保存成功后会生成checkpoint、modle.ckpt.meta、modle.ckpt.data-00000-of-00001、modle.ckpt.inde、四个文件。需要调用时可通过saver.restore(sess, "./modle.ckpt")调用。

**2.3 Cassandra配置**

通过docker network create some-network创建一个网络

在通过docker run --name hyh(名字)-cassandra --network some-network -d cassandra:latest运行

最后ocker run --name hyh-cassandra -p 9042:9042 -d cassandra:latest将其映射到9042端口。

如果要在python中使用Cassandra还需要通过如下命令下载相应的驱动程序：

pip install cassandra-driver

**2.4 程序运行**

通过docker build --tag=final(名字) .

将写好的app.py生成一个docker镜像，在通过：

docker run -p 4000:7000 final

将其映射到4000端口

最后通过curl -XPORT来上传本地图片进行识别。

通过Cassandra的cqlsh命令查询保存的时间戳等结果。

1. **心得体会**

这次的项目终端来说其实不算难，但是在进行过程中真的遇到了好多好多的小问题导致进展的不顺利，首先最大的问题就是网络，Linux系统使用的各种软件源都来自国外，这导致很多东西的下载速度非常缓慢，甚至会下不下来。其次各种安装包的版本也是个问题，之前下载了TensorFlow最新版本结果导致无法运行还很难找到错误，最后卸载新的版本下载旧的才得以解决问题。这种版本以及虚拟环境的问题报错真的耽误了很多时间。

不过这次的收获也很多，虽然项目不复杂，但是老师的课程确从整个大数据的结构框架入手，让我对大数据的整体框架有了一个系统性的了解，同时也对大数据不同层面所使用的软件，所应当掌握的技术有了初步的认识。这对于我来说确实是一种全新的体验，之从来没接触过大数据的我也开始对这方面产生了兴趣，将来或许可以考虑选择这一个方向。

最后还得感谢老师这么多天的细心指导，很多问题基本是在老师的指导下才能顺利解决。同时老师也为我扩宽了知识面与视野，让我对大数据方向有了进一步的认识。