

**人工智能实验报告**

题 目 基于Mindspore框架与ModelArts平台 的MNIST手写体识别实验

专 业 计算机类

学　　 号 120L021905

姓 名 郎朗

同 组 人 员 无

1. **背景简介/问题描述**

当今人工智能火热，相应的有不少专业领域持续火热，例如机器学习、深度学习、强化学习等。本实验为验证性实验，最终为两部分内容：基于Mindspore框架的模型本地训练及预测、基于Modelarts平台和Tensorflow框架的模型训练及部署。通过两个内容，对人工智能有一个感性的认识。

上述两部分内容都是面向同一个问题，即MNIST手写体识别实验。利用不同的框架，在本地/云端进行模型训练及结果预测。

MNIST数据集是一个手写体数字的图片数据集，由美国NIST发起整理。其中有10类28\*28的灰度图片组成，训练数据集包含60000张图片，测试数据集包含10000张图片。1998年，Yan LeCun等人发表论文《Gradient-Based Learning Applied to Document Recognition》，首次提出了LeNet-5网络，利用MNIST数据集实现手写字体识别。本实验就是选用的LeNet-5网络进行验证。

1. **算法介绍**

2.1 LeNet-5网络简介

LeNet-5是一个比较简单的卷积神经网络（卷积神经网络主要用于图像处理问题），下图中显示了LeNet-5网络的结构：

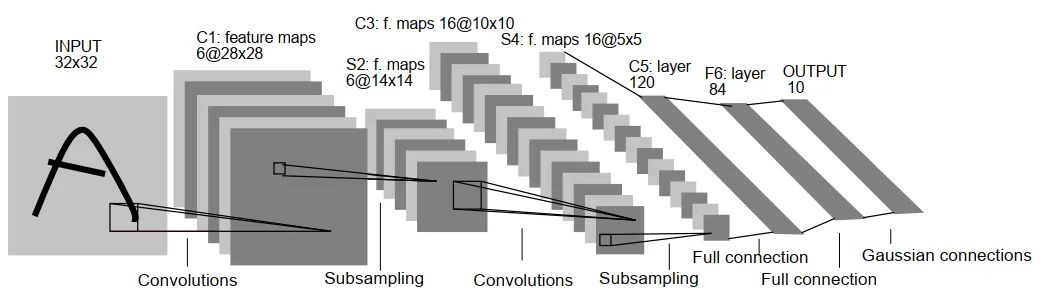


图1.LeNet-5网络结构

不算输入层LeNet-5网络包含7层，分别是C1,S2,C3,S4,C5,F6以及OUTPUT，其中C代表卷积层，S代表池化层，F代表全连接层，后面的数字代表当前是第几层。

输入层是接收32\*32像素的图像；C1层使用6个5\*5大小的卷积核，填充padding=0，步长stride=1进行卷积，得到6个28\*28大小的特征图；S2层使用6个2\*2大小的卷积核进行池化，padding=0，stride=2，得到6个14\*14大小的特征图（相当于先降采样，再用sigmod激活函数）；C3层使用16个5\*5\*n大小的卷积核，padding=0，stride=1进行卷积，得到16个10\*10大小的特征图；S4层使用16个2\*2大小的卷积层池化，padding=0，stride=2，得到16个5\*5大小的特征图；C5层使用120个5\*5\*16大小的卷积核，padding=0，stride=1进行卷积，得到120个1\*1大小的特征图（相当于120个神经元的全连接层）；F6共有84个神经元，与C5进行全连接，计算输入向量和权重向量之间的点积，加上偏置，使用sigmod激活函数输出；最后Output层使用RBF函数（欧式距离函数），计算输入向量和参数向量之间的欧式距离。

下面是以手写体数字3为例的简要识别过程，可以发现最终的3是比较规范的。

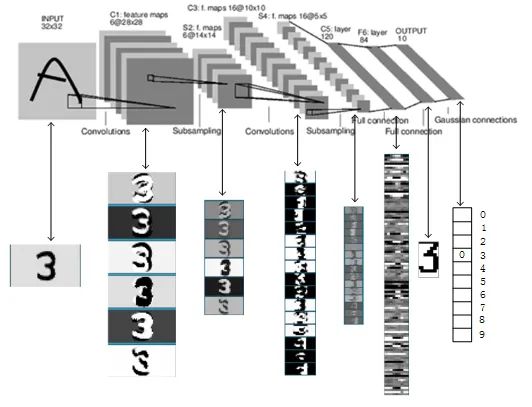


图2.手写体数字3的识别过程

2.2 基于Mindspore框架的模型本地训练及预测

本地训练及预测大体可以分为前期准备、算法代码实现（这里代码由实验指导给出，只需要进行相应的填充和填写即可）。

前期准备需要在MindSpore官网安装对应版本的MindSpore，选择的配置如下图所示：

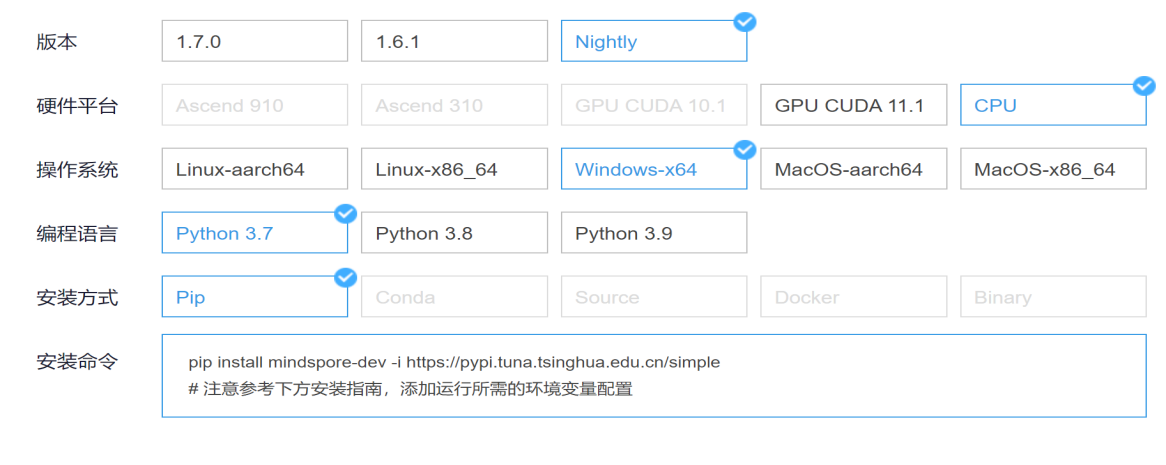


图3.MindSpore安装

在下述链接中下载MNIST数据集并放到项目指定位置， <http://yann.lecun.com/exdb/mnist/。>

剩下来的就是算法代码实现（根据实验指导跑通代码）。首先导入相 关Python库，并导入context模块，配置运行时需要的信息（此部分已经在实验包中给出）；其次是数据处理部分，根据数据集存储地址，生成数据集，构建训练、验证函数，通过给定的create\_dataset函数对传入的数据集进行处理，大致流程是定义数据集、定义进行数据增强和处理所需要的一些参数、根据参数生成对应的数据增强操作、使用map映射函数将数据操作应用到数据集、对生成的数据集进行处理；紧接着是定义网络，选择相对简单的LeNet-5网络（简介见2.1），通过继承mindspore.nn.cell.Cell定义LeNet网络，预先在\_\_init\_\_方法中定义，再通过定义construct方法完成前向构造（定义网络各层代码由实验指导给出）；然后定义损失函数及优化器，其中损失函数为目标函数，作为衡量预测值和实际值差异的大小，通过减小损失函数的值，不断优化模型，而优化器用于最小化损失函数。损失函数和优化器、编译形成模型、训练网络都由实验指导提供；最后进行训练（运行train.py）和验证过程（运行eavl.py）。

2.3 基于Modelarts平台和Tensorflow框架的模型训练及部署

此部分根据下述链接中的指导进行即可，链接如下：<https://support.huaweicloud.com/bestpractice-modelarts/modelarts_10_0080.html>

准备工作阶段，需要注册华为云账号、设置访问授权、在OBS中创建相应的桶和文件夹（如“test-modelarts-cz”的桶），获取MNIST数据集并上传至新建的dataset-mnist文件夹中。

适配算法代码部分，需要将本地代码进行简单的代码适配（此部分由gitee提供适配好的算法），在gitee下载后，找到对应的train\_mnist\_tf.py进行上传。

创建算法，在ModelArts管理控制台处，进行“我的算法”创建，选择“显示旧版镜像”+Tensorflow+TF-1.8.0-python3.6，选择创建的代码目录路径“test-modelarts-cz”，找到训练脚本“train\_mnist\_tf.py”，设置data\_url以及train\_url参数。

训练模型，在训练管理中创建训练作业，根据上一步创建的算法，选择对应的训练输入“dataset-mnist”以及训练输出“mnist-model”，资源类型根据需求选取即可，此处选取GPU：1\*NVIDIA-V100(32GB)|CPU:8核64GB，提交训练作业后，在训练管理中查看，状态为已完成即训练结束。



图4.训练作业完成状态

在完成的训练作业中找到对应的输出路径，在输出中可以找到训练生成的模型，生成的模型如下：



图5.训练生成的模型

部署AI应用，在此之前需要将推理代码“customize\_service.py”和配置文件“config.json”上传至OBS中，结果如上图图5所示。随后在AI应用管理中创建AI应用，选择训练作业作为元模型来源，进行在线服务。当AI应用列表中模型为“正常”时，即为创建成功。在AI应用中部署在线服务，根据需求填写相应的参数，部署完成后等待服务为运行中状态即可使用，如下图所示：



图6.AI应用创建成功、部署成功

服务部署完成后，在在线服务页面进入服务详情，在预测页签上传预测的图片进行预测即可。

1. **算法实现**

3.1 基于Mindspore框架的模型本地训练及预测的实验结果

本地训练共进行10个epoch，耗时25686.345ms，每步时间约13.699ms。

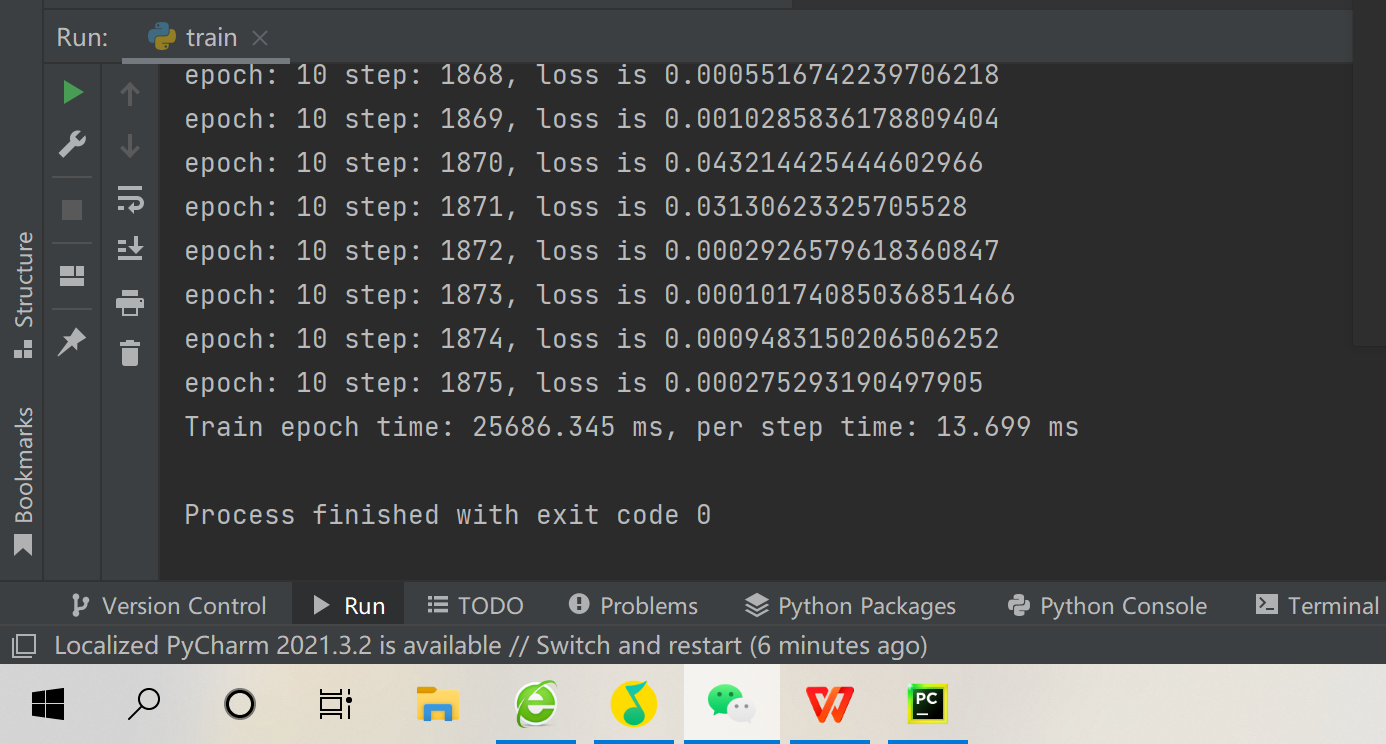


图7.本地训练结果

本地预测精度约为0.9760，实际结果如下图所示：

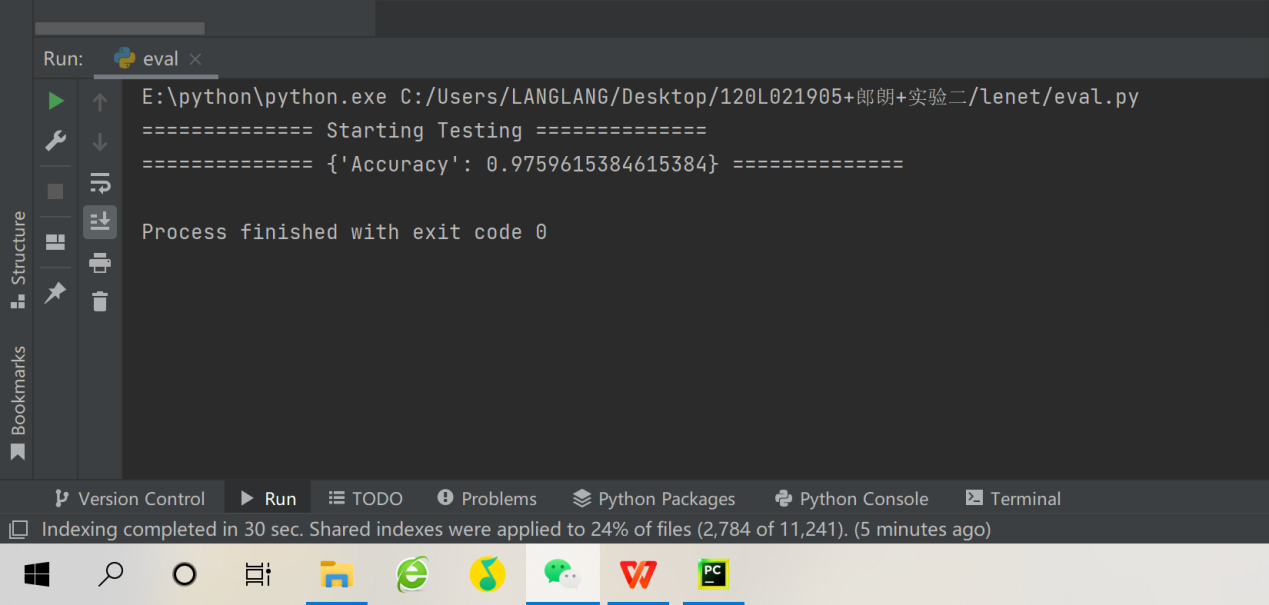


图8.本地预测结果

3.2 基于Modelarts平台和Tensorflow框架的模型训练及部署的实验结果

预测图片分为本人写的白底黑字、本人写的黑底白字、训练数据中的数据，结果如下图所示，可以看出白底黑字的都是置信概率为1和0的预测，而黑底白字则存在置信概率在0~1之间的情况，训练数据中的数据则置信概率为1且预测正确。

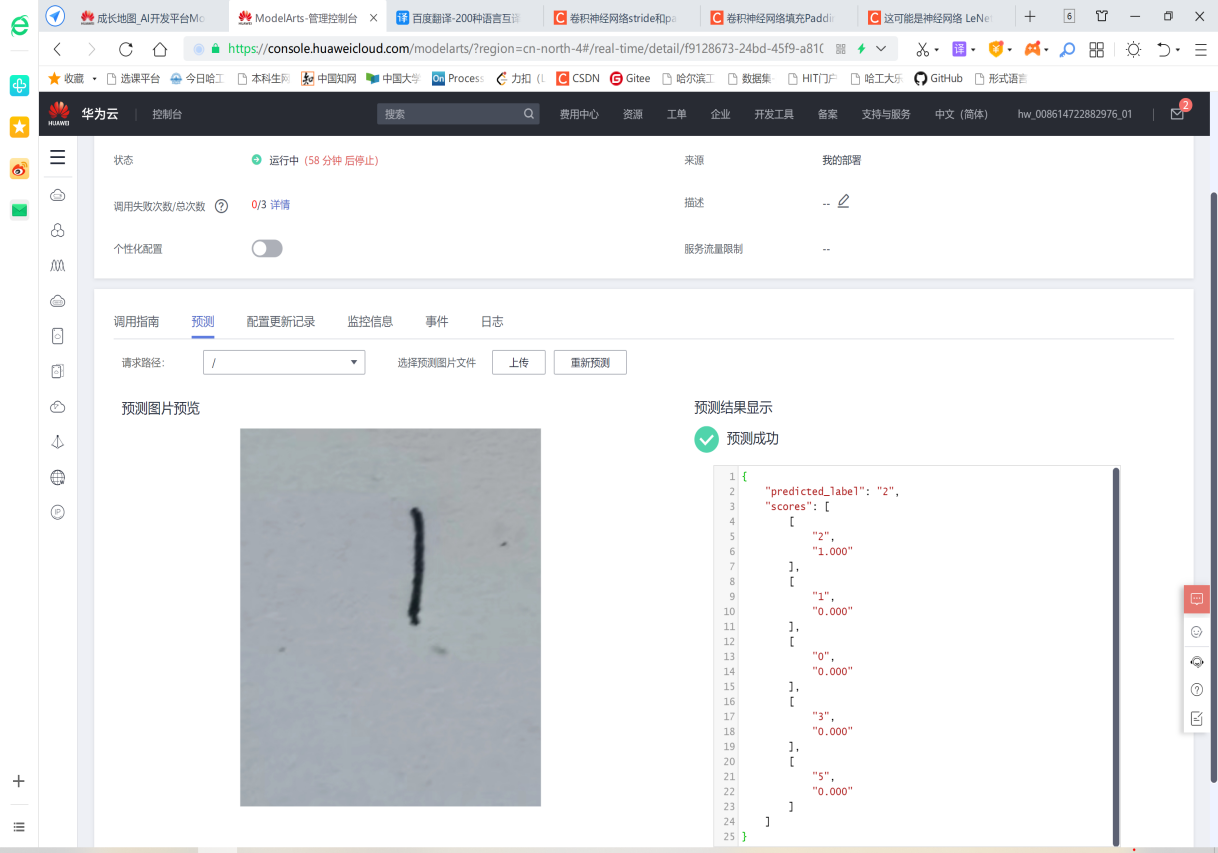


图9.白底黑字预测1

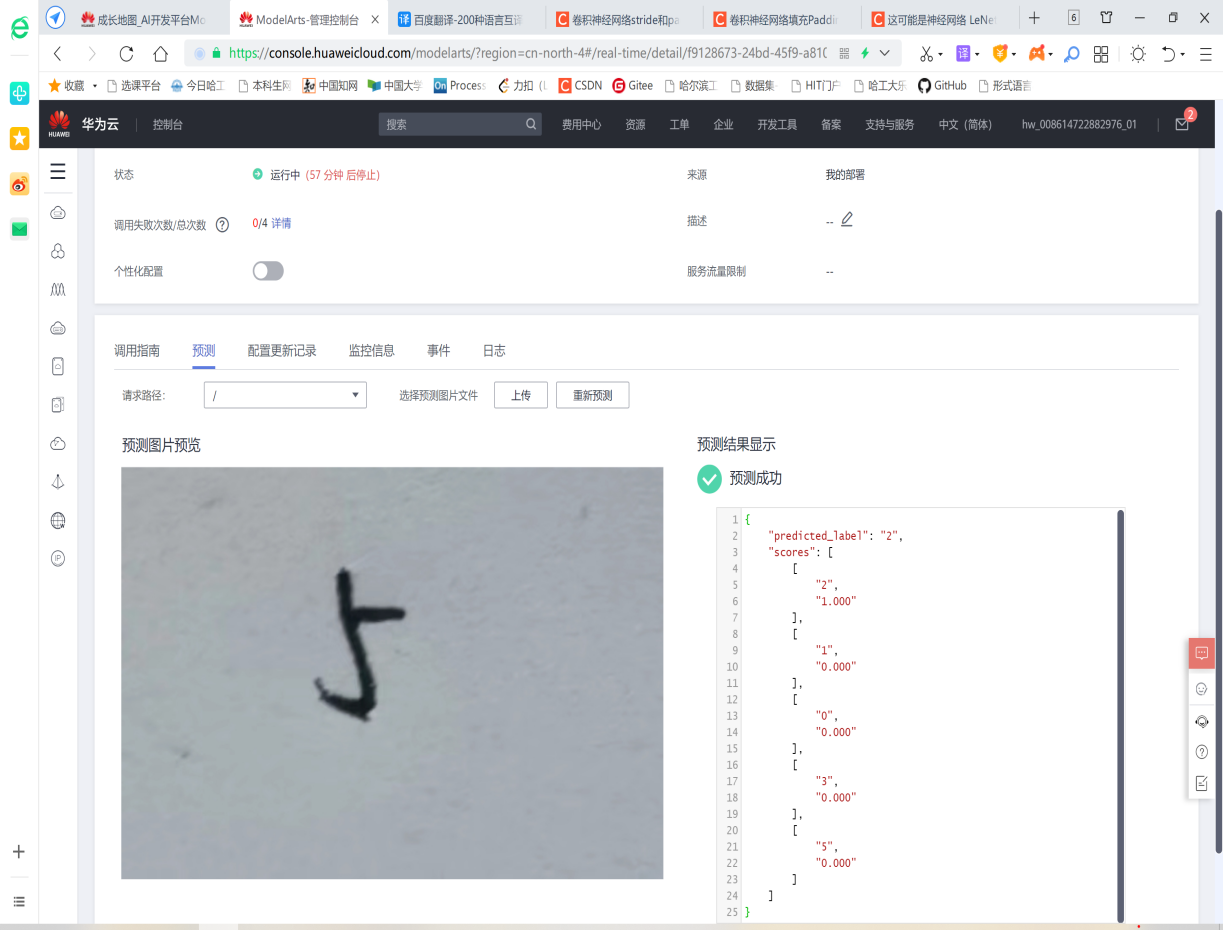


图10.白底黑字预测2

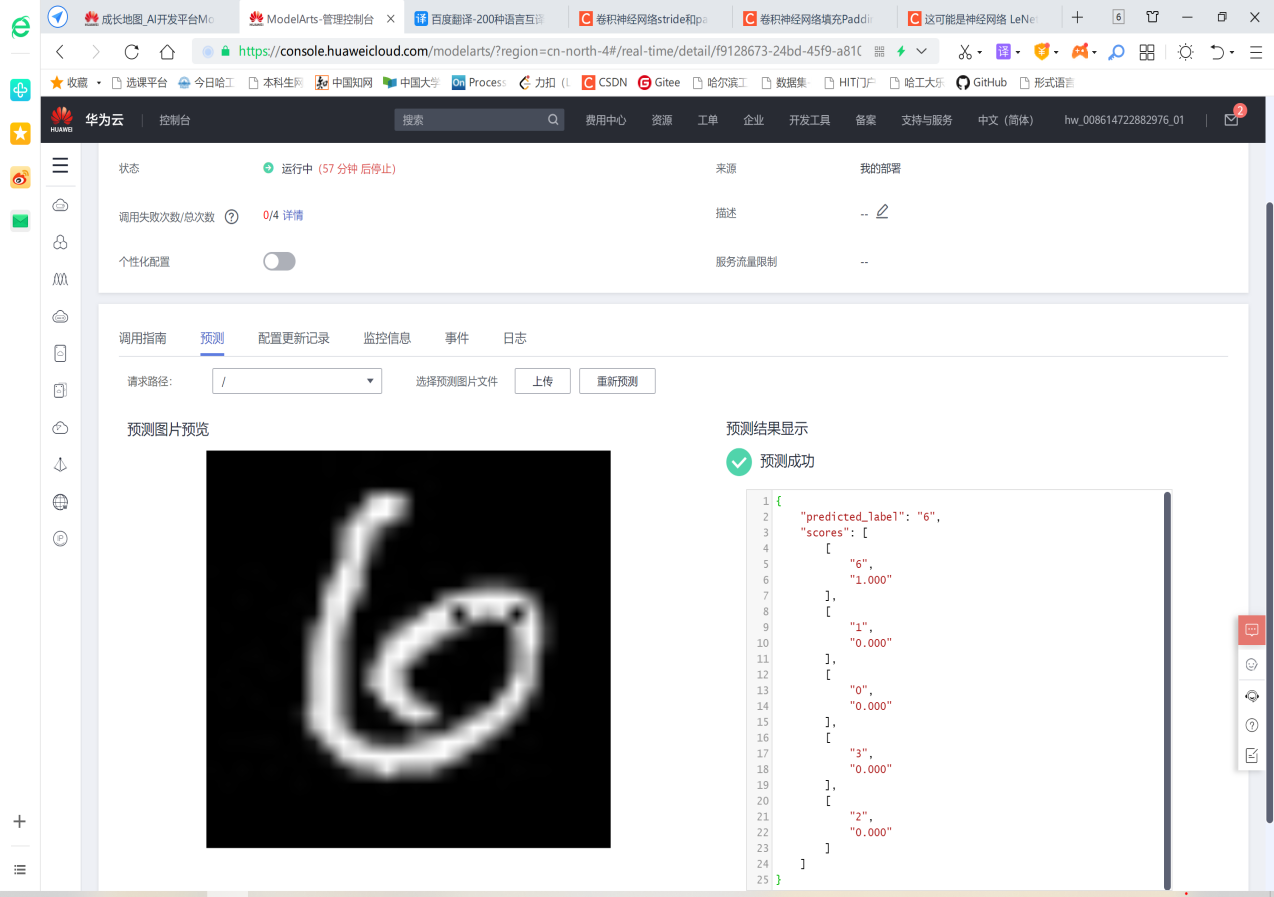


图11.训练数据中的数据预测

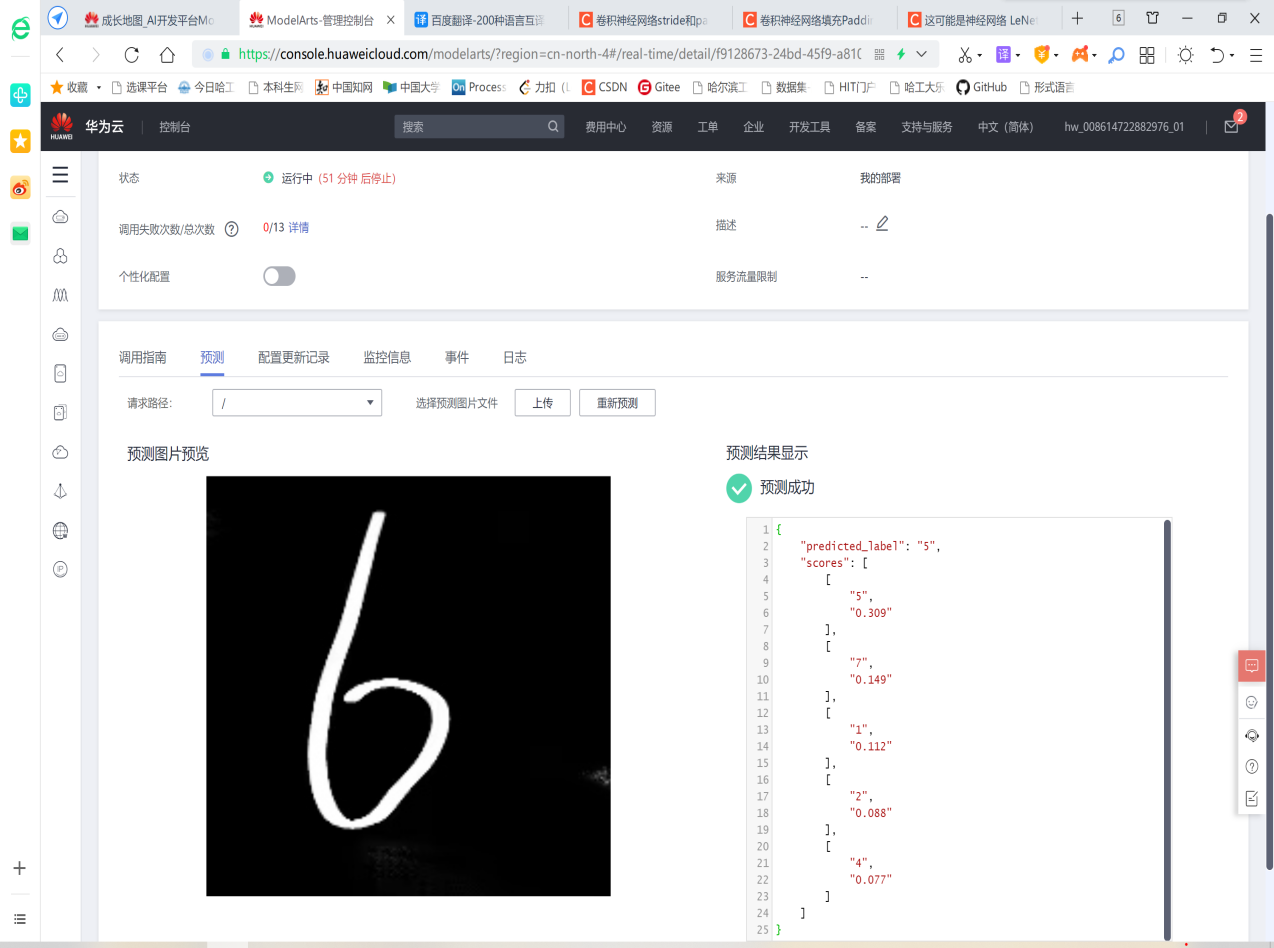


图12.黑底白字+本人手写预测1

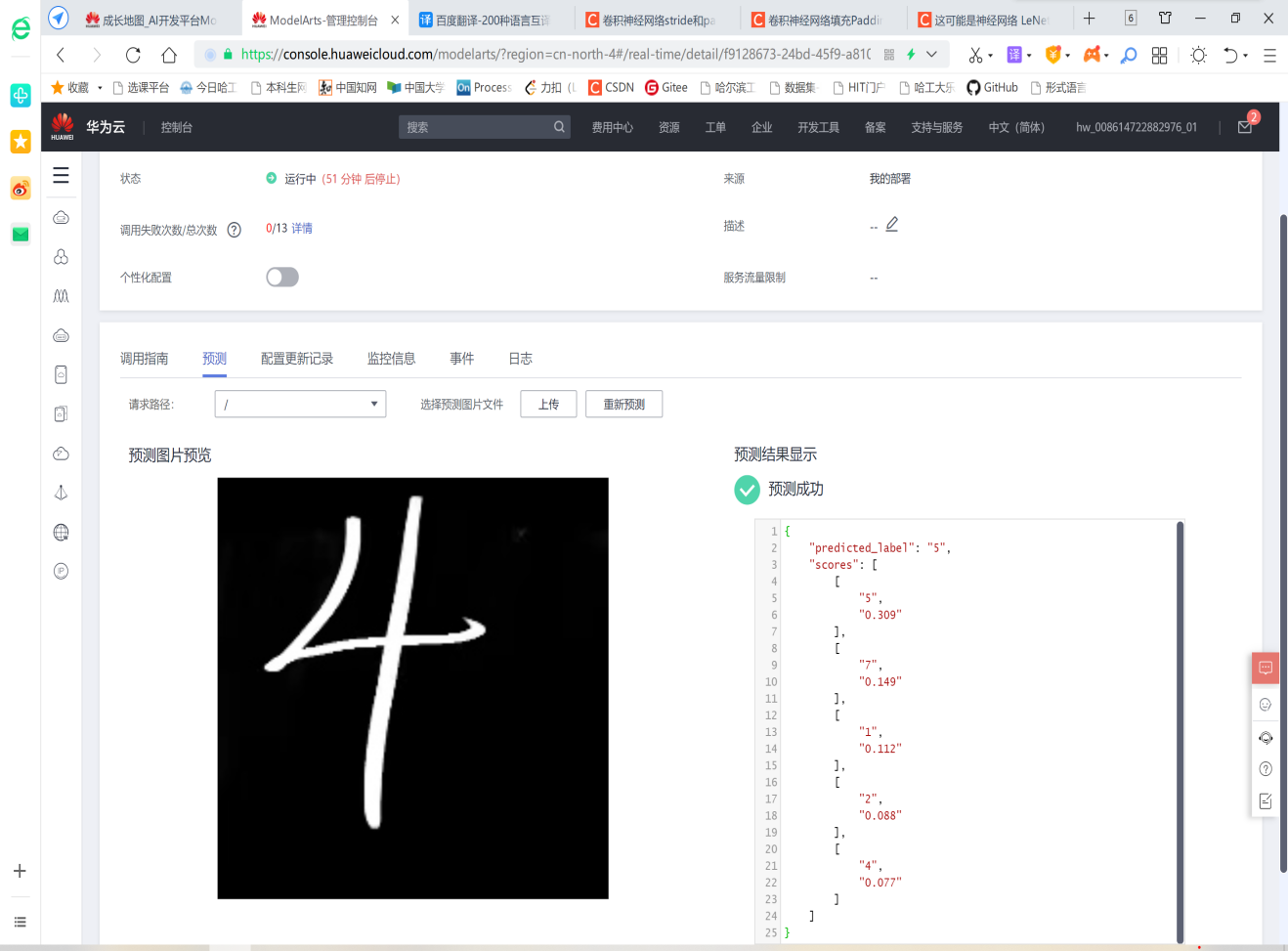


图13.黑底白字+本人手写预测2

1. **讨论及结论**

4.1 讨论

在本地运行时，可以看到每轮、每步的损失大体趋势上看是在下降的，符合最小化损失函数的要求。如下图所示：

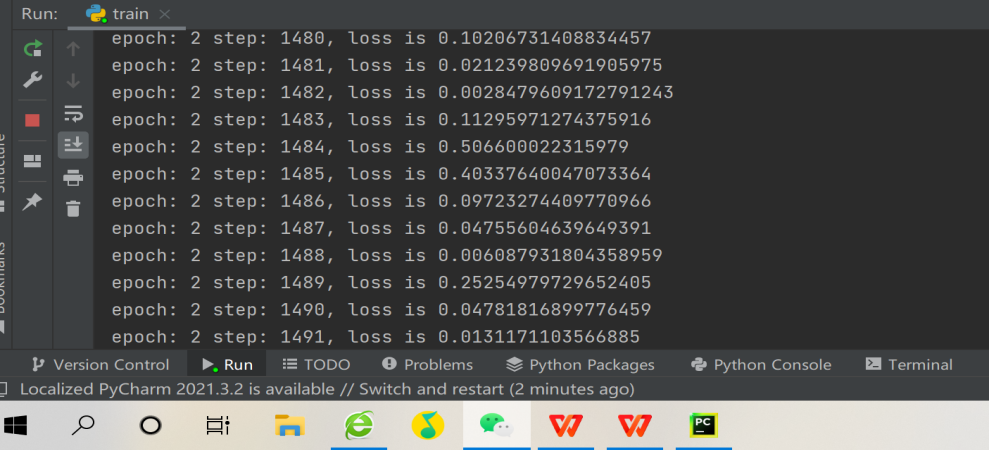


图14.本地训练过程1

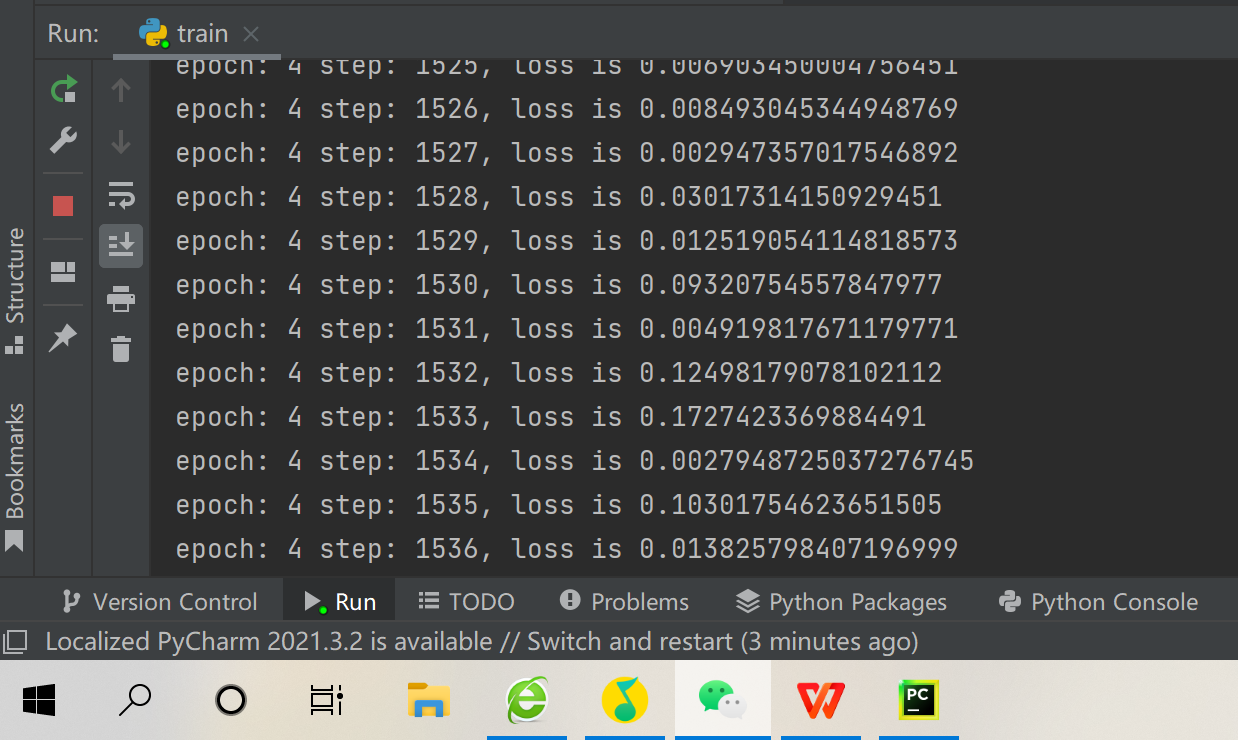


图15.本地训练过程2

最终本地预测的精度也较高，这可能与数据集的复杂度相关，换 用其他数据集可能在相同的参数下得不到类似的效果。

在云端运行Tensorflow框架的代码时，对于白底黑字的数据预测置信均为若干0和一个1（图9、图10所示），结果基本为错误，而黑底白字的数据预测有若干差别，分布在0~1之间（图11、图12、图13），说明此模型对数据输入有要求（黑底白字才能正确进行预测）。其次，当用模型训练数据进行预测时，也得到了正确的置信概率为1的预测，数据集可能过简单，也可能说明了在预测阶段不能使用训练数据（用其他方法除外，例如交叉验证法，但是不是本次训练数据作为本次的预测数据）。另外，根据“符合”规则的输入预测结果来看，以图12为例，数据输入为手写体6，但是预测结果，5的置信概率为0.309,7的置信概率为0.149,1的置信概率为0.112,2的置信概率为0.088,4的置信概率为0.077，其他的过小未进行输出，预测的结果还是存在一定错误的，可能是模型复杂度过低，训练不够等造成的，可以选用更复杂的模型、再进行一定的训练进行预测。

4.2 结论

1.训练过程是一个最小化损失函数的过程，大体上损失函数的值是下降的；

2.数据集的复杂度、选取的模型复杂度对生成的模型效果有一定的影响，数据集过复杂，模型过简单，可能会欠拟合；数据集过简单，模型过复杂可能会过拟合；

3.模型对于输入有一定要求，不符合要求的输入难以得到期待的正确结果；

4.在预测过程中，不要将本次训练数据作为本次预测数据，否则预测结果可能会过好，对本次模型评估效果有一定的影响。

**参考文献**

1. https://blog.csdn.net/red\_stone1/article/details/121804658?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165460953116782391888541%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=165460953116782391888541&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~top\_click~default-2-121804658-null-null.142^v11^pc\_search\_result\_control\_group,157^v13^control&utm\_term=LeNet-5&spm=1018.2226.3001.4187
2. <https://support.huaweicloud.com/bestpractice-modelarts/modelarts_10_0080.html>

注：报告中图的下方要有图题（如图1. XXXX），表格需要用三线表，表头需列在表的上方（如表1. XXXX）。图表居中排列。