山东大学 计算机科学与技术 学院

人工智能导论 课程实验报告

|  |  |  |
| --- | --- | --- |
| 学号：202300130183 | 姓名： 宋浩宇 | 邮箱：202300130183＠mail.sdu.edu.cn |
| 实验题目：⼋、⼿写数字识别与垃圾分类 | | |
| 实验过程：  （记录实验过程、遇到的问题和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）  首先完成数据集的下载，在http://yann.lecun.com/exdb/mnist/网站下载标准MNIST数据集及标签。   1. 将Ubyte形式的数据转化为可视化的图片，此处使用了pytorch框架和matplotlib来实现图像的转化     此处因为matplotlib库中的savefig使用了HTTP协议通过本地回环网络ip127.0.0.1来把图片输出到本地，因此使用了time库的sleep函数来防止单位时间请求发送过多导致报错。   1. 为了实现神经网络实现MNIST手写数字识别，此处构建多层感知机的模型。     4.声明全局变量作为模型的超参数    5.对数据进行transform操作（由于使用的是Windows系统的Mindspore，不支持CUDA，所以此处没有将数据转成tensor对象）     1. 定义损失函数     损失函数使用了目标值与预测值的差的平方的和作为标准，此处还涉及到概率的放缩，即将预测向量的和放缩到1   1. 构建网络     该网络一共有四层，输入层接受一个784维的向量，即图像的点阵，隐含层有两层，分别由lay1\_node\_cnt和lay2\_node\_cnt设置，输出层输出一个十维的向量，代表网络的预测概率   1. 定义梯度下降函数     该函数返回一个numpy数组，里边存储的是损失函数对于网络中每一个参数的偏导数，function参数接受损失函数，x，y分别是训练数据和网络的输出结果，此处利用到了mindspore的自动微分功能。   1. 训练模型     此处用到了mindspore官方网站帮助档里给的样例代码，简要来说就是对该数据集进行n\_epochs次学习，每次挑出batch\_size\_train个数据来进行学习，参数的调整由梯度下降函数返回的numpy数组和学习率决定，在学习完一次后在按照batch\_size\_test设置的一次测试的数据来计算损失函数的平均值。完成学习循环之后模型即为训练完毕。   1. 储存模型      1. 在另一段程序中加载模型 | | |
| 结果分析与体会：  华为的mindspore框架为人工智能应用的开发提供了许多方便的工具，而且还提供了许多云计算服务。本身昇思框架也有着开源的开发者社区，有着丰富的简体中文学习资源。但是框架的性能相对于tensorflow和pytorch等存在更久的框架还有些距离，而且对于windows系统的cuda开发支持不够，导致windows上跑模型会用不上nvidia显卡的加速，只能靠CPU，希望华为能早日将该框架的windows系统的cuda版发布出来。顺便一提这个网络只是一个多层感知机模型，它的中间层是完全的黑箱，另外mindspore也提供了卷积层较为简单的实现方式，也可以使用卷积神经网络来优化这个学习过程。 | | |