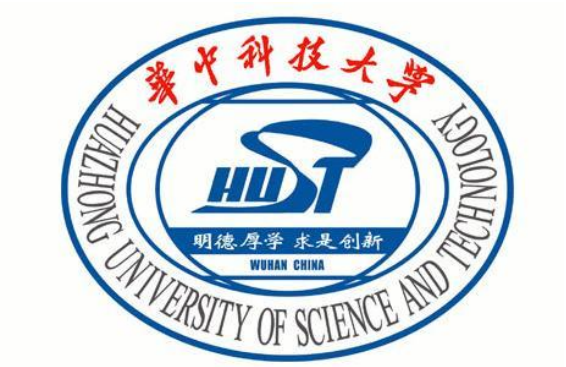
华中科技大学

计算机科学与技术学院

《机器学习》第六次实验报告



专　　业：计算机科学与技术

班　　级：2107 班

学　　号：U202115538

姓　　名：陈侠锟

成　　绩：

指导教师：张腾

完成日期：2023 年 5 月 14 日

目录

鸢尾花三分类问题与 mnist 手写数据集识别

1. 实验要求

a) 总体要求

根据所给代码注释进行填空，实现鸢尾花三分类与 mnist 手写数据集识别的功能，要求鸢尾花三分类的准确率在 95%以上，mnist 手写数据集的测试准确率在 95%以上。

要求在实验中能够使用 MindSpore 框架中的损失函数、优化器，搭建 LeNet5 神经网络等，并在华为云平台上进行测试。

b) 数据来源

鸢尾花数据集：

https://gitee.com/link?target=http%3A%2F%2Farchive.ics.uci.edu%2Fml%2Fdatasets%2FIris

mnist 手写数据集：

http://yann.lecun.com/exdb/mnist/

2. 算法设计与实现

此部分将对两份代码中的填空部分进行解释。

a) 鸢尾花三分类问题

i. 函数 create_dataset

首先对存储路径为 data_path 的数据进行读取，用 readlines 一次性读取全部的行，如下图所示。

```
# Todo 每个类的前五个样本信息
f = open(data_path, 'r')
data = f.readlines()
```

图 2.1 读取数据

之后分别将 Iris-setosa，Iris-versicolor，Iris-virginica 对应为 0，1，2 三类（利用字典），如下图所示。

```
# Todo 分别将Iris-setosa, Iris-versicolor, Iris-virginica对应为0, 1, 2三类
label_map = {'Iris-setosa\n': 0, 'Iris-versicolor\n': 1, 'Iris-virginica\n': 2}
```

图 2.2 分类

利用 random.choice 函数将数据集分成 8:2 的两部分，前者做训练集，后者做测试集，其中，np.random.choice 函数中的两个参数分别代表选取范围和选取大小，如下图所示。

```
# Todo Using random choice and split dataset into train set and validation set by 8:2.
train_idx = np.random.choice(range(len(X)), int(0.8 * len(X))) # 参数为: 选取范围, 选取大小
test_idx = np.array(list(set(range(len(X))) - set(train_idx)))
```

图 2.3 划分训练集和测试集

将测试数据集转换为 MindSpore 使用的形式（这里参考了已给的将训练集转换形式的那部分代码），并将 drop_remainder 设置为 False，防止因为测试集过小而被舍弃。如下图所示。

```
# Convert the test data to MindSpore Dataset.
XY_test = list(zip(X_test, Y_test))
ds_test= dataset.GeneratorDataset(XY_test, ['x', 'y'])
ds_test = ds_test.shuffle(buffer_size=120).batch(32, drop_remainder=False)
```

图 2.4 将测试集转换为 MindSpore 的形式

ii. 函数 softmax_regression

使用 MindSpore 框架中的交叉熵损失计算函数进行计算，函数的参数 sparse 指定目标值是否使用稀疏格式；参数 reduction 指定应用于输出结果的计算方式。经测试，在华为云平台上运行代码时，若设置 sparse = False 则会报错，而令 reduction = mean 可使 loss 值达到最低。代码如下图所示。

```
# Todo 使用交叉熵损失计算
loss = nn.SoftmaxCrossEntropyWithLogits(sparse=True, reduction='mean')
```

图 2.5 使用交叉熵损失计算

交叉熵损失的数学公式如下。

$$\ell(x_i, c) = -\log\left(\frac{\exp(x_i[c])}{\sum_j \exp(x_i[j])}\right) = -x_i[c] + \log\left(\sum_j \exp(x_i[j])\right)$$

图 2.6 交叉熵损失的数学公式

设置优化器，其中参数的值根据提供的注释来设置，如下图所示。

```
# Todo 使用动量优化器优化参数，其中学习率设置为0.05，动量设置为0.9
opt = nn.optim.Momentum(net.trainable_params(), learning_rate=0.05, momentum=0.9)
```

图 2.7 设置优化器

iii. 主函数 main

设置数据集路径，如下图所示。

```
# Todo:设置路径
data_path = ""  # 在华为云平台完成实验时，路径应改为/home/ma-user/work/iris.data
```

图 2.8 设置数据集路径

在华为云上读取数据时，路径应设置为/home/ma-user/work/iris.data

b) mnist 手写数据集识别

i. 函数 create_dataset

设置放缩的大小，经查询知输入层接受的图片大小应是 32*32；设置归一化参数，代码如下图所示。

```
# Todo 设置放缩的大小
resize_height, resize_width = 32,32 # 输入层为尺寸32*32的图片
# Todo 归一化
rescale = 1.0 / 225.0
```

图 2.9 设置放缩大小及设置归一化参数

使用 shuffle 和 batch 操作进行数据处理（这里的用法也参考了鸢尾花代码文件中的用法），如下图所示。

```
# Todo 进行shuffle、batch、repeat操作
buffer_size = 10000
mnist_ds = mnist_ds.shuffle(buffer_size = buffer_size).batch(batch_size, drop_remainder = True)
return mnist_ds
```

图 2.10 使用 shuffle 和 batch 操作进行数据处理

ii. 神经网络 LeNet5 的搭建

通过查询 MindSpore 官方文档实现了 LeNet5 网络的搭建，如下图所示。

```
def __init__(self, num_class=10, num_channel=1):
    super(LeNet5, self).__init__()
    # 以下两个函数参考MindSpore官方文档
    self.conv1 = nn.Conv2d(num_channel, 6, 5, pad_mode='valid')
    self.conv2 = nn.Conv2d(6, 16, 5, pad_mode='valid')
    self.fc1 = nn.Dense(16 * 5 * 5, 120, weight_init=Normal(0.02))
    self.fc2 = nn.Dense(120, 84, weight_init=Normal(0.02))
    self.fc3 = nn.Dense(84, num_class, weight_init=Normal(0.02))
    self.relu = nn.ReLU()
    self.max_pool2d = nn.MaxPool2d(kernel_size=2, stride=2)
    self.flatten = nn.Flatten()

def construct(self, x):
    # 使用定义好的运算构建前向网络
    x = self.conv1(x)
    x = self.relu(x)
    x = self.max_pool2d(x)
    x = self.conv2(x)
    x = self.relu(x)
    x = self.max_pool2d(x)
    x = self.flatten(x)
    x = self.fc1(x)
    x = self.relu(x)
    x = self.fc2(x)
    x = self.relu(x)
    x = self.fc3(x)
    return x
```

图 2.11 LeNet5 网络的搭建

其中，conv1 为卷积层，输入的通道数为 num_channel，输出的通道数为 6，卷积核大小 5*5；conv2 为卷积层，输入的通道数为 6，输出的通道数为 16，卷积核大小为 5*5；fc1 为全连接层，输入个数为 16*5*5，输出个数为 120；fc2 为全连接层，输入个数为 120，输出个数为 84；fc3 为全连接层，输入个数为 84，分类的个数为 num_class；ReLU 为激活函数；max_pool2d 为池化层；flatten 用于多维数组展平为一维数组。

iii. 函数 train_net

加载训练用数据集之后直接利用 MindSpore 的 train 函数进行训练，代码如下。

```
def train_net(model, epoch_size, data_path, repeat_size, ckpoint_cb, sink_mode):
    """定义训练的方法"""
    # 加载训练数据集
    ds_train = create_dataset(os.path.join(data_path, "train"), 128, repeat_size)
    model.train(epoch_size, ds_train, callbacks=[ckpoint_cb, LossMonitor(125)], dataset_sink_mode=sink_mode)
```

图 2.12 定义训练的方法

iv. 函数 test_net

加载测试用数据集后直接利用 MindSpore 的 eval 函数进行验证，并输出正确率，代码如下。

```python
def test_net(model, data_path):
    """定义验证的方法"""
    dataset_eval = create_dataset(os.path.join(data_path, "test"))
    acc = model.eval(dataset_eval, dataset_sink_mode = False)
    print("{}".format(acc))
```

图 2.13 定义验证的方法

3. 实验环境与平台

实验平台：华为 Modelarts
CPU：2 核 8GB
Python：3.7.10
镜像：mindspore1.2.0-openmpi2.1.1-ubuntu18.04
tensorflow1.15-mindspore1.7.0-cann5.1.0-euler2.8-aarch64

4. 结果与分析

a) 鸢尾花三分类问题

得到的分类结果如下图所示。

```
!python yuanweihua_experiment.py

epoch: 1 step: 3, loss is 1.031557
epoch: 2 step: 3, loss is 0.7610287
epoch: 3 step: 3, loss is 0.6462583
epoch: 4 step: 3, loss is 0.4901954
epoch: 5 step: 3, loss is 0.3576305
epoch: 6 step: 3, loss is 0.45708314
epoch: 7 step: 3, loss is 0.4008247
epoch: 8 step: 3, loss is 0.37959045
epoch: 9 step: 3, loss is 0.43809596
epoch: 10 step: 3, loss is 0.38404837
epoch: 11 step: 3, loss is 0.30719244
epoch: 12 step: 3, loss is 0.23967561
epoch: 13 step: 3, loss is 0.30403373
epoch: 14 step: 3, loss is 0.36535662
epoch: 15 step: 3, loss is 0.28054222
epoch: 16 step: 3, loss is 0.2048519
epoch: 17 step: 3, loss is 0.21001716
epoch: 18 step: 3, loss is 0.31740525
epoch: 19 step: 3, loss is 0.21134505
epoch: 20 step: 3, loss is 0.19350979
epoch: 21 step: 3, loss is 0.23076057
epoch: 22 step: 3, loss is 0.15827383
epoch: 23 step: 3, loss is 0.20771278
epoch: 24 step: 3, loss is 0.15324715
epoch: 25 step: 3, loss is 0.18861957
[WARNING] ME(12820:140121180460864,MainProcess):2023-05-
k mode currently.So the evaluating process will be perfo
{'acc': 0.9666666666666667, 'loss': 0.18293747305870056}
```

图 4.1 鸢尾花三分类问题的结果

b) mnist 手写数据集识别

得到的识别结果如下图所示。

```
!python mnist_experiment.py

/home/ma-user/anaconda3/envs/MindSpore/
et (5.0.0)/charset_normalizer (2.0.12) 
  RequestsDependencyWarning)
epoch: 1 step: 125, loss is 2.2983823
epoch: 1 step: 250, loss is 1.226889
epoch: 1 step: 375, loss is 0.3937568
epoch: 2 step: 32, loss is 0.10915408
epoch: 2 step: 157, loss is 0.18906969
epoch: 2 step: 282, loss is 0.061702434
epoch: 2 step: 407, loss is 0.12924601
{'Accuracy': 0.9710536858974359}
Predicted: "6", Actual: "6"
```

图 4.2 mnist 手写数据集识别的结果

c) 分析

上述结果可以看出，两个问题的预测结果均符合精度要求，但同时我发现，在反复运行代码时，正确率会在 0.9~1.0 之间浮动，尚不清楚原因。

5. 个人体会

本实验采用华为云平台下的 MindSpore 框架进行实验，这对我来说无疑是一个陌生的东西，但好在有官方提供的环境搭建手册以及从网上搜索到的 MindSpore 拥有的函数说明、神经网络搭建方法等工具，让我不至于毫无头绪。

本次实验让我对机器学习的两个经典例子——鸢尾花三分类和 mnist 手写数据集识别有了新的认识，同时我也第一次接触到了 LeNet5 神经网络的搭建，LeNet5 网络结构较为简单，也正适合作为学习神经网络的入门课。我学习了神经网络的训练、验证和测试流程，为后续的学习打下基础，对超参数的不断调整来提高正确率的过程，也考验着耐心与细心。今后我也会尝试更加复杂的神经网络与数据集，来提高自己机器学习的实战能力。