

第六次实验报告

计卓 2101 王彬

第六次实验分为 MNIST 手写数据集的学习和 Iris 三分类问题的求解。我们首先介绍 MNIST 手写数据集识别的过程。

MNIST 手写数据集识别

该实验基于华为 MindSpore-1.7.0 库，使用 LeNet-5 网络结构神经网络进行深度学习。我们的实验在华为云上进行了运行，最终我们的准确率可以达到 0.98347，基本符合预期。

LeNet-5 网络结构通过多轮卷积层、降采样层后，对于得到的结果再进行全连接后，再输出。我们尝试在代码中构建 LeNet-5 的网络结构，并且对其进行多次前向的卷积和池化。我们的代码如图 1 LeNet-5 建构代码所示。

图 1 LeNet-5 建构代码

```
79 class LeNet5(nn.Cell):
80     """
81     LeNet网络结构
82     """
83     def __init__(self, num_class=10, num_channel=1):
84         super(LeNet5, self).__init__()
85         self.conv1 = nn.Conv2d(num_channel, 6, 5, pad_mode='valid')
86         self.conv2 = nn.Conv2d(6, 16, 5, pad_mode='valid')
87         self.fc1 = nn.Dense(16 * 5 * 5, 120)
88         self.fc2 = nn.Dense(120, 84)
89         self.fc3 = nn.Dense(84, num_class)
90         self.relu = nn.ReLU()
91         self.max_pool2d = nn.MaxPool2d(kernel_size=2, stride=2)
92         self.flatten = nn.Flatten()
93
94     def construct(self, x):
95         # 使用定义好的运算构建前向网络
96         x = self.conv1(x)
97         x = self.relu(x)
98         x = self.max_pool2d(x)
99         x = self.conv2(x)
100        x = self.relu(x)
101        x = self.max_pool2d(x)
102        x = self.flatten(x)
103        x = self.fc1(x)
104        x = self.relu(x)
105        x = self.fc2(x)
106        x = self.relu(x)
107        x = self.fc3(x)
108        return x
```

之后，我们对于 MindSpore 库中所支持的各类操作进行调用与训练即可。原代码在提交文件夹下“mnist_experiment.py”下，测试结果如下图 2 所示。

图 2 MNIST 测试结果

```
epoch: 1 step: 1875, loss is 2.302959442138672
epoch: 2 step: 1875, loss is 2.3108954429626465
epoch: 3 step: 1875, loss is 0.11261879652738571
epoch: 4 step: 1875, loss is 0.0939168930053711
===== Starting Testing =====
===== Accuracy:{'Accuracy': 0.9834735576923077} =====
```

鸢尾花三分类问题

鸢尾花三分类则是一个传统机器学习的问题。我们将构建一个 Softmax 回归模型，对于鸢尾花的种类进行预测。

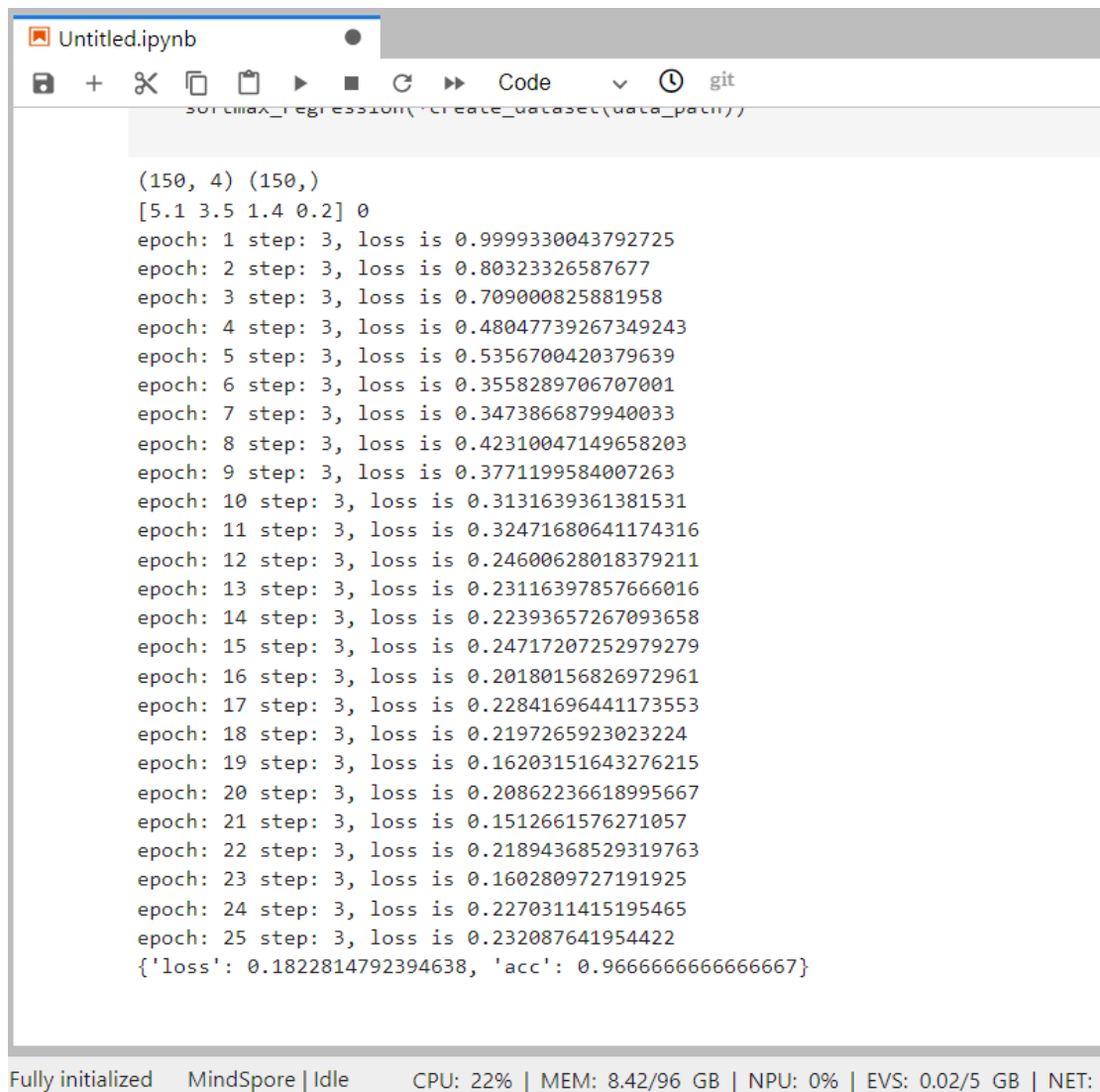
我们首先下载数据集（Iris.data，<http://archive.ics.uci.edu/ml/machine-learning-databases/iris/>）。数据集共有 4 个属性和分类结果，我们将数据集根据 8:2 的比例区分训练集和测试集，并按照题目所给提示进行按部就班的填空。如图 3 所示，我们将题目中空出的 Todo-List 均进行了填写，本题完整代码见压缩包内“yuanweihua_experiment.py”。

图 3 鸢尾花预测实验的填写状况

```
17 def create_dataset(data_path):
18     # Todo 每个类的前五个样本信息
19     with open(data_path) as csv_file:
20         data = list(csv.reader(csv_file, delimiter=','))
21     # Todo 分别将Iris-setosa, Iris-versicolor, Iris-virginica对应为0, 1, 2三类
22     iris = datasets.load_iris()
23     label_map = {}
24     label_map['Iris-setosa'] = 0
25     label_map['Iris-versicolor'] = 1
26     label_map['Iris-virginica'] = 2
27     X_iris, y_iris = iris.data, iris.target
28     X = np.array([[float(x) for x in s[:-1]] for s in data[:150]], np.float32)
29     Y = np.array([label_map[s[-1]] for s in data[:150]], np.int32)
30     print(X_iris.shape, y_iris.shape)
31     print(X_iris[0], y_iris[0])
32
33     # Todo Using random choice and split dataset into train set and validation se
34     # X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, ra
35     train_idx = np.random.choice(150, 120, replace=False)
```

我们依然在华为云上使用 MindSpore-1.7.0 进行训练。训练结果如图 4 所示，我们的预测准确率为 0.96667，最终损失率为 0.1822，基本达成题目所给要求。

图 4 鸢尾花预测实验结果



```
softmax_regression(create_dataset(data_path))

(150, 4) (150,)
[5.1 3.5 1.4 0.2] 0
epoch: 1 step: 3, loss is 0.9999330043792725
epoch: 2 step: 3, loss is 0.80323326587677
epoch: 3 step: 3, loss is 0.709000825881958
epoch: 4 step: 3, loss is 0.48047739267349243
epoch: 5 step: 3, loss is 0.5356700420379639
epoch: 6 step: 3, loss is 0.3558289706707001
epoch: 7 step: 3, loss is 0.3473866879940033
epoch: 8 step: 3, loss is 0.42310047149658203
epoch: 9 step: 3, loss is 0.3771199584007263
epoch: 10 step: 3, loss is 0.3131639361381531
epoch: 11 step: 3, loss is 0.32471680641174316
epoch: 12 step: 3, loss is 0.24600628018379211
epoch: 13 step: 3, loss is 0.23116397857666016
epoch: 14 step: 3, loss is 0.22393657267093658
epoch: 15 step: 3, loss is 0.24717207252979279
epoch: 16 step: 3, loss is 0.20180156826972961
epoch: 17 step: 3, loss is 0.22841696441173553
epoch: 18 step: 3, loss is 0.2197265923023224
epoch: 19 step: 3, loss is 0.16203151643276215
epoch: 20 step: 3, loss is 0.20862236618995667
epoch: 21 step: 3, loss is 0.1512661576271057
epoch: 22 step: 3, loss is 0.21894368529319763
epoch: 23 step: 3, loss is 0.1602809727191925
epoch: 24 step: 3, loss is 0.2270311415195465
epoch: 25 step: 3, loss is 0.232087641954422
{'loss': 0.1822814792394638, 'acc': 0.9666666666666667}
```

Fully initialized MindSpore | Idle CPU: 22% | MEM: 8.42/96 GB | NPU: 0% | EVS: 0.02/5 GB | NET:

综上所述，我们完成了第六次实验的两组代码完成。我们的代码运行正确，且均高于实验预期的 0.95 的准确率。完整代码详见提交文件夹“第六次实验代码”下的“mnist_experiment.py”和“yuanweihua_experiment.py”。