

深圳技术大学考试答题纸

(以论文、报告等形式考核专用)

二〇二二 ~ 二〇二四 学年度第 二 学期

课程编号 IB00108 课程名称 Python 程序设计 主讲教师 柯笑 评分
2022002

学 号 02017 姓名 陈佳煜 专业年级 计算机一班

教师评语：

题目：**fashion-mnist-master 数据分类**

1. 研究动机及背景介绍

动机： Fashion-MNIST 数据集是为了替代原始的 MNIST 数据集，用于基准测试机器学习算法。MNIST 数据集对于现代算法来说已经过于简单，无法挑战现有模型的极限，因此需要一个更具挑战性和多样性的数据集。

重要性： Fashion-MNIST 作为一个更具挑战性和多样性的数据集，反映了更实际的场景。这对于推动计算机视觉研究和提高分类算法的准确性非常重要，尤其是在自动化零售系统、库存管理和个性化时尚推荐等应用中。

背景： Fashion-MNIST 包含 70,000 张灰度图像，涵盖 10 个不同的时尚类别，每个类别包含 7,000 张图像。每张图像的大小为 28x28 像素。数据集分为 60,000 张训练图像和 10,000 张测试图像，格式与原始 MNIST 数据集相同。

2. 数据来源

数据集描述：Fashion-MNIST 数据集是公开可用的，可以从 Zalando Research 网站下载。数据集包含：

训练集： 60,000 张图像

测试集： 10,000 张图像

图像大小： 28x28 像素

类别数量： 10

数据特征： 数据集包括各种时尚物品，如 T 恤、裤子和鞋子，标签从 0 到 9，每个标签对应一个特定的时尚物品。

统计分析：

均值和方差： 计算像素值的均值和方差以了解分布情况。

箱线图： 可视化像素值分布以检测异常值。

t 检验和正态性检验： 确认数据是否符合正态分布，这可能影响算法的选择。

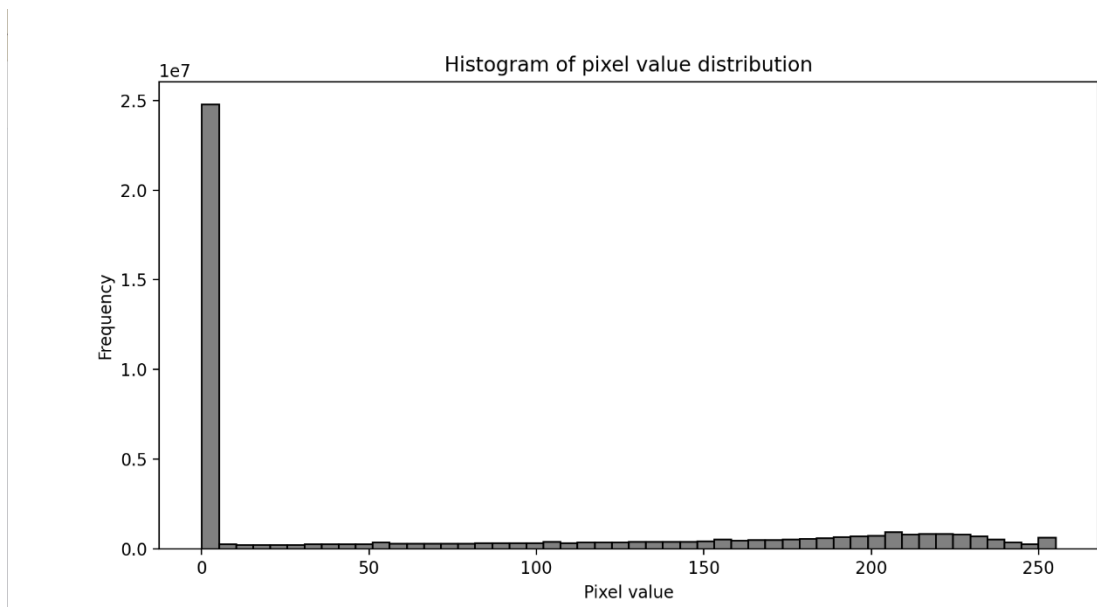
```
# 计算像素值的均值和方差
train_images = np.array(fashion_mnist_train_dataset.images)
mean_pixel_value = np.mean(train_images)
std_pixel_value = np.std(train_images)
print(f'像素值均值: {mean_pixel_value}, 像素值标准差: {std_pixel_value}')
```

```
像素值均值: 72.94042205810547, 像素值标准差: 90.02120971679688
正态性检验p值: 0.0
```

```
# 绘制像素值分布箱线图（取样100张图像）
sample_images = train_images[:100]

plt.figure(figsize=(10, 5))
sns.boxplot(data=sample_images.reshape(100, -1))
plt.title('Boxplot of pixel value distribution (sample of 100 images)')
plt.show()
```

像素值分布直方图



3. 数据预处理

标准化： 通过除以 255 将像素值标准化到 $[0, 1]$ 范围内。

缺失值处理： 确保数据集中没有缺失值，因为所有图像的大小和标签是一致的。

异常值检测和处理： 使用箱线图和 z 分数检测并处理像素值中的异常值。

数据增强：应用旋转、缩放和翻转等变换，增加训练集的多样性，提高模型的鲁棒性。

```
# 数据标准化和增强
trans = Compose([Normalize(mean=[127.5], std=[127.5], data_format='CHW'), ToTensor()])
fashion_mnist_train_dataset = FashionData(dir='data/fashion', type="train", trans=trans)
fashion_mnist_test_dataset = FashionData(dir='data/fashion', type='t10k', trans=trans)
```

```
# 绘制像素值分布箱线图（取样100张图像）
sample_images = train_images[:100]

plt.figure(figsize=(10, 5))
sns.boxplot(data=sample_images.reshape(100, -1))
plt.title('Boxplot of pixel value distribution (sample of 100 images)')
plt.show()
```

```
/fashion-mnist-master/data/train.py
像素值均值：72.94042205810547，像素值标准差：90.02120971679688
正态性检验p值：0.0
```

4. 降维技术

主成分分析（PCA）：

降低数据维度，同时保留大部分方差。

应用于加速训练过程并降低计算复杂度。

t-SNE：另一种将高维数据可视化的技术，通过降维到 2 或 3 维进行可视化。

```
# PCA降维
pca = PCA(n_components=50)
train_images_pca = pca.fit_transform(train_images)
test_images_pca = pca.transform(np.array(fashion_mnist_test_dataset.images))
print(f'PCA后训练数据形状：{train_images_pca.shape}')
print(f'PCA后测试数据形状：{test_images_pca.shape}')
```

PCA后训练数据形状: (60000, 50)

PCA后测试数据形状: (10000, 50)

Layer (type)	Input Shape	Output Shape	Param #
Flatten-1	[[1, 28, 28]]	[1, 784]	0
Linear-1	[[1, 784]]	[1, 100]	78,500
ReLU-1	[[1, 100]]	[1, 100]	0
Linear-2	[[1, 100]]	[1, 100]	10,100
ReLU-2	[[1, 100]]	[1, 100]	0
Linear-3	[[1, 100]]	[1, 10]	1,010
Softmax-1	[[1, 10]]	[1, 10]	0

Total params: 89,610

Trainable params: 89,610

Non-trainable params: 0

Input size (MB): 0.00

Forward/backward pass size (MB): 0.01

Params size (MB): 0.34

Estimated Total Size (MB): 0.35

5. AI 模型构建

模型选择:

多层感知机 (MLP): 作为基线模型的基本神经网络模型。

卷积神经网络 (CNN): 用于图像分类的高级模型。

模型详情:

输入层: 28x28 像素展平成 784 个节点。

隐藏层: 两个隐藏层, 每层 100 个神经元。

输出层: 10 个神经元, 对应 10 个类别, 使用 softmax 激活函数。

参数设置:

学习率: 0.001

批量大小: 256

训练轮数: 5

优化器: Adam

```
# 定义多层感知机模型
class MultilayerPerceptron(nn.Layer):
    def __init__(self):
        super(MultilayerPerceptron, self).__init__()
        self.linear1 = nn.Linear(1*28*28, 100)
        self.linear2 = nn.Linear(100, 100)
        self.linear3 = nn.Linear(100, 10)

    def forward(self, inputs):
        x = paddle.flatten(inputs, start_axis=1, stop_axis=-1)
        x = self.linear1(x)
        x = F.relu(x)
        x = self.linear2(x)
        x = F.relu(x)
        x = self.linear3(x)
        y = F.softmax(x, axis=1)
        return y
```

```
# 定义网络结构
network = nn.Sequential(
    nn.Flatten(),
    nn.Linear(784, 100),
    nn.ReLU(),
    nn.Linear(100, 100),
    nn.ReLU(),
    nn.Linear(100, 10),
    nn.Softmax(axis=1)
)
```

```
# 配置模型
model = paddle.Model(network)
model.prepare(optimizer=paddle.optimizer.Adam(learning_rate=0.001, parameters=model.parameters()),
              loss=paddle.nn.CrossEntropyLoss(),
              metrics=paddle.metric.Accuracy())

model.summary((1, 28, 28))

# 训练模型
model.fit(fashion_mnist_train_dataset, epochs=5, batch_size=256, verbose=1)
```

```
Epoch 1/5
step 235/235 [=====] - loss: 1.6199 - acc: 0.7702 - 28ms/step
Epoch 2/5
step 235/235 [=====] - loss: 1.5841 - acc: 0.8431 - 28ms/step
Epoch 3/5
step 235/235 [=====] - loss: 1.6208 - acc: 0.8533 - 27ms/step
Epoch 4/5
step 235/235 [=====] - loss: 1.6602 - acc: 0.8625 - 28ms/step
Epoch 5/5
step 235/235 [=====] - loss: 1.5538 - acc: 0.8674 - 30ms/step
Eval begin...
step 10000/10000 [=====] - loss: 1.4613 - acc: 0.8468 - 888us/step
```

6. 模型评估与比较

评估指标：

准确率： 正确分类图像的比例。

```
Eval samples: 10000
{'loss': [1.4613158702850342], 'acc': 0.8468}
Predict begin...
step 10000/10000 [=====] - 607us/step
Predict samples: 10000
第2条记录 真实值: 1  预测值: 1
第15条记录 真实值: 1  预测值: 1
第38条记录 真实值: 7  预测值: 7
第211条记录 真实值: 1  预测值: 1
第222条记录 真实值: 2  预测值: 3
第323条记录 真实值: 5  预测值: 5
```

精确率、召回率、F1 分数： 用于评估分类性能的指标。

混淆矩阵： 用于可视化不同类别的表现。

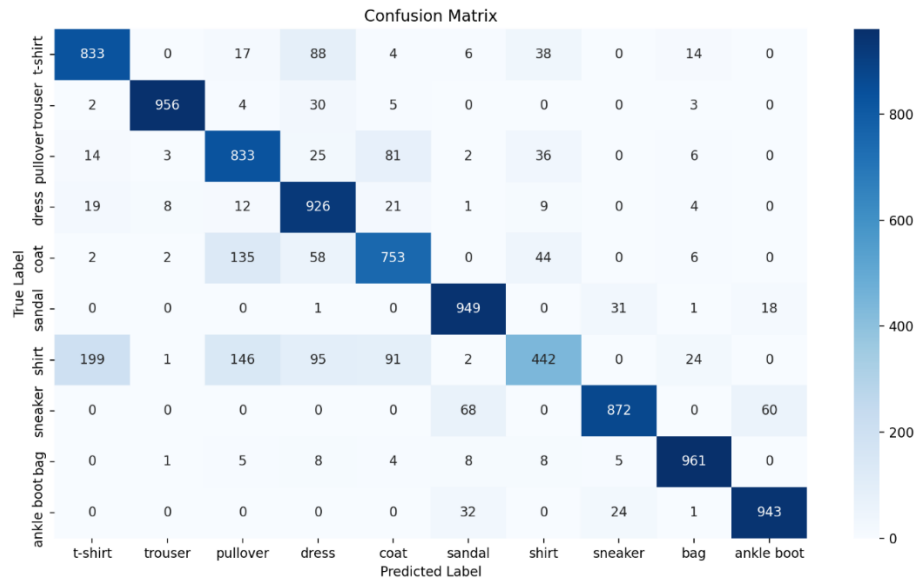
```
# 计算混淆矩阵
true_labels = [label[1][0] for label in fashion_mnist_test_dataset]
predicted_labels = [np.argmax(pred) for pred in predict_result[0]]
conf_matrix = confusion_matrix(true_labels, predicted_labels)

# 打印分类报告
class_report = classification_report(true_labels, predicted_labels, target_names=label_list)
print('分类报告:\n', class_report)

# 绘制混淆矩阵
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_list, yticklabels=label_list)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

混淆矩阵

fashion-mnist-master 数据分类



比较：

单一模型： 基于所选模型的评估。

多个模型： 如果应用，比较 MLP 和 CNN 以确定性能更好的模型。

```
# 评估模型
eval_result = model.evaluate(fashion_mnist_test_dataset, verbose=1)
print(eval_result)

# 预测测试集
predict_result = model.predict(fashion_mnist_test_dataset)
```

分类报告：	precision	recall	f1-score	support
t-shirt	0.78	0.83	0.81	1000
trouser	0.98	0.96	0.97	1000
pullover	0.72	0.83	0.77	1000
dress	0.75	0.93	0.83	1000
coat	0.79	0.75	0.77	1000
sandal	0.89	0.95	0.92	1000
shirt	0.77	0.44	0.56	1000
sneaker	0.94	0.87	0.90	1000
bag	0.94	0.96	0.95	1000
ankle boot	0.92	0.94	0.93	1000
accuracy			0.85	10000
macro avg	0.85	0.85	0.84	10000
weighted avg	0.85	0.85	0.84	10000

7. 参考文献

Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms [Internet]. ar5iv.org; 2017 [cited 2024-01-05].

Papers With Code. Fashion-MNIST Dataset [Internet]. Papers With Code. [cited 2024-01-05].

Papers With Code. Fashion-MNIST Benchmark (Image Classification) [Internet]. Papers With Code. [cited 2024-01-05].

姓名	学号	分工	总体贡献 (%)
陈汉翀	202200202041	AI 模型训练	30
陈佳煜	202200202017	统计分析	50
魏子葆	202200202016	数据收集	20

