手写体识别实验报告

201250070 郁博文

首先，加载 MNIST 数据集，并归一化，将像素值缩放到 0 到 1 之间，

```
# 加载MNIST数据集
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# 将像素值缩放到0到1之间
x_train, x_test = x_train / 255.0, x_test / 255.0
```

然后增加了一维以适应 CNN 的输入要求。

接着，创建了一个包含三个卷积层、两个池化层和两个全连接层的 CNN 模型。模型的第一层是一个 32 个过滤器（filters）大小为 3x3 的卷积层，激活函数使用 ReLU。接下来的池化层将图像的大小减半。然后再加上一个 64 个过滤器大小为 3x3 的卷积层，接着还有一个池化层。最后是一个 64 个过滤器大小为 3x3 的卷积层，然后通过 Flatten()函数将二维特征图变为一维向量，接着连接两个全连接层。

```
# 创建CNN模型
model = tf.keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(28, 28, 1)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10)
])
```

最后，编译模型，使用 adam 优化器和稀疏分类交叉熵损失函数，同时设定评价指标为准确率。对训练数据进行了 5 个 epoch 的训练，使用测试数据评估了模型的性能，输出了测试集的准确率。

```
# 编译模型
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

# 训练模型
model.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))

# 评估模型
test_loss, test_acc = model.evaluate(x_test,  y_test, verbose=2)
print('\nTest accuracy:', test_acc)
```

最终结果如下，accuracy 为 99.22%，可能是因为过拟合了。