## 使用卷积神经网络实现手写数字识别

|  |  |
| --- | --- |
| 【实验名称】 | 使用卷积神经网络实现手写数字识别 |
| 【知识点准备】 | |
| Python基础，keras，科学计算包基础（pandas，numpy） | |
| 【实验目的】 | |
| 1. 加强对卷积神经网络的理解  2. 感受卷积神经网络的强大应用  3. 加强对深度学习流程的理解 | |
| 【实验内容】 | |
| 构建网络卷积层  查看模型结构及参数个数  构建网络的全连接层  加载数据并进行归一化处理  训练模型  对模型进行评估 | |
| 【实验要求】 | |
| 模型在测试集上的准确率大于等于97% | |
| 【实验步骤】 | |
| * **单机模式**  1. 构建网络卷积层   from keras import layers  from keras import models model = models.Sequential() model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1))) model.add(layers.MaxPooling2D((2, 2))) model.add(layers.Conv2D(64, (3, 3), activation='relu')) model.add(layers.MaxPooling2D((2, 2))) model.add(layers.Conv2D(64, (3, 3), activation='relu'))  2 . 查看模型结构及参数个数  model.summary()  3 . 构建网络的全连接层  model.add(layers.Flatten())  model.add(layers.Dense(64, activation='relu')) model.add(layers.Dense(10, activation='softmax'))  4 . 查看此时模型结构及参数个数  model.summary()  5 . 加载数据并进行归一化处理  from keras.datasets import mnist  from keras.utils import to\_categorical  import numpy as np  f = np.load('mnist.npz')  train\_images, train\_labels = f['x\_train'], f['y\_train']  test\_images, test\_labels = f['x\_test'], f['y\_test']  train\_images = train\_images.reshape((60000, 28, 28, 1)) train\_images = train\_images.astype('float32') / 255 test\_images = test\_images.reshape((10000, 28, 28, 1)) test\_images = test\_images.astype('float32') / 255 train\_labels = to\_categorical(train\_labels) test\_labels = to\_categorical(test\_labels)  6 . 训练模型  model.compile(optimizer='rmsprop',  loss='categorical\_crossentropy', metrics=['accuracy']) model.fit(train\_images, train\_labels, epochs=5, batch\_size=64)    7 . 对模型进行评估  test\_loss, test\_acc = model.evaluate(test\_images, test\_labels)  print(test\_acc) | |