

智能物联网实验：

智能数据处理与分析——MNIST 数据集图片分类

张昊迪

实验背景介绍：

近年来神经网络与深度学习的发展在社会生产生活各个领域正在引起深刻变革。图像识别是人工智能与机器学习的典型任务之一。2019 年图灵奖获得者 Yann LeCun 在 30 余年前为美国邮政局开发了一个基于卷积神经网络的手写数字识别系统，用于自动识别信封上的手写邮政编码。

具体内容实施：

MNIST 是一个手写数字图像数据集，由于其简单易用，是深度学习领域的经典数据集之一。通过使用 Python 编程和深度学习技术，我们可以训练神经网络来识别手写数字图像。本实验旨在让学生掌握如何使用 Python 编写深度学习代码，并使用 MNIST 数据集训练和测试神经网络模型。

实验目标：

熟悉 Python 编程语言
掌握使用 PyTorch 构建神经网络模型
理解深度学习中的前向传播和反向传播算法
能够训练神经网络模型并进行准确性测试
了解神经网络模型的训练过程。

实验步骤：

步骤 1: 导入必要的库和数据集

首先，我们需要导入必要的库和 MNIST 数据集。我们将使用 TensorFlow 中的 keras 模块来构建神经网络模型，以及 numpy 和 matplotlib 库来处理数据和可视化结果。

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt

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

步骤 2: 数据预处理

在这一步骤中，我们需要对数据进行预处理。由于 MNIST 数据集中的每个图像都是 28x28 像素的灰度图像，因此我们需要将像素值从 0-255 缩放到 0-1 之间，以便更好地训练我们的神经网络模型。

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

步骤 3: 构建模型

我们将使用一个简单的神经网络模型来处理 MNIST 数据集。该模型包括 3 个层，第一层是输入层，第二层是全连接层，第三层是输出层。我们使用 ReLU 激活函数来激活隐藏层，使用 Softmax 激活函数来激活输出层，以便进行分类。

```
# 构建模型
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

步骤 4: 编译模型

在训练模型之前，我们需要编译模型。在编译模型时，我们需要指定损失函数、优化器和评估指标。在这里，我们将使用交叉熵损失函数、Adam 优化器和准确率作为评估指标。

```
# 编译模型
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

步骤 5: 训练模型

现在，我们已经完成了模型的构建和编译，我们可以使用训练数据来训练模型了。我们将使用批量梯度下降法 (batch gradient descent) 来更新模型的参数。我们设置了 10 个 epochs，即模型将迭代 10 次。

```
# 训练模型
model.fit(x_train, y_train, epochs=10, validation_split=0.1)
```

步骤 6: 评估模型

在训练模型后，我们需要评估模型的性能。我们将使用测试数据来评估模型的性能，并计算模型的准确率。

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

步骤 7: 预测新数据

最后，我们可以使用训练好的模型来预测新的手写数字图像。我们将从测试数据中选择一些图像，并使用模型来预测它们的标签。

```
# 预测新数据
predictions = model.predict(x_test[:5])
print('Predicted labels:', np.argmax(predictions, axis=1))
print('True labels      : ', y_test[:5])
```

你可能遇到的问题：

1. 环境搭建：安装 Python、pip、TensorFlow 出错
2. 理论问题：全连接层、反向传播梯度下降法等概念不懂
3. 实验步骤：上述实验步骤在你的计算机上出问题

你需要做的事情：

Google，并参考官网、技术帖子等：

PyTorch 官方文档：<https://pytorch.org/get-started/locally/>

TensorFlow 官方文档：<https://www.tensorflow.org/install/pip?hl=zh-cn>

MNIST 代码参考示例：<https://zhuanlan.zhihu.com/p/137571225>