第一章　卷积神经网络-手写体数字识别

1.1　实验内容与任务

MNIST数据集是一个手写体数字的图像数据集，训练集包括60000张图片，测试集包括10000张图片，每张图片是一个8位的灰度图片，尺寸为28×28,训练集的前20张图片如图1.1所示。现要求训练一个卷积神经网络，用于识别数字图片。

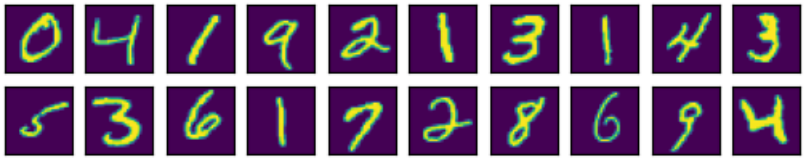

图 1.1: 数字图片

1.2　实验过程及要求

1. 实验环境要求：Windows/Linux操作系统，Python编译环境，numpy、keras、matplotlib等程序库。
2. 学习理解神经网络、卷积网络层、图像处理等知识。
3. 下载数据集，构建卷积神经网络，进行网络训练与评估。
4. 调整网络超参数，记录网络训练的过程

5. 撰写实验报告。

1.3　教学目标

1. 能够理解和掌握神经网络模型。

2. 能够应用keras平台搭建卷积神经网络，实现图像分类。

3. 能够调整神经网络超参数，提高神经网络的性能。

4. 提高对复杂工程问题建模和分析的能力。

1.4　相关知识及背景

神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的激活函数，每两个节点间的连接都代表一个对于通过该连接信号的加权值，网络的输出是连接方式、权重值和激活函数的综合作用。通过调整神经网络的结构、规模、参数，它可以逼近任何函数，因此神经网络目前成为机器学习的重要工具。

图像处理中可以通过卷积计算获得图像特征，卷积神经网络可以进行图像特征提取，以实现图像分类、目标检测等任务。2012年，在大规模机器视觉识别竞赛(ILSVRC)上，卷积神经网络AlexNet超出其他学习方法，取得了最好结果。从此，卷积神经网络技术在图像处理领域得到广泛的应用。

1.5　实验教学与指导

1.5.1　神经网络与层

各种信息处理过程可以看作是一个函数处理$y = f(x)$，然而f的形式往往是未知并且复杂的。神经网络使用简单的线性处理或非线性处理进行连续复合的方式，来逼近或者模拟f。用x^0表示输入数据，用x^{i-1}, f^i, x^i表示

第i次复合处理的输入、简单处理函数和输出，则神经网络处理过程为

$$\begin{aligned} x^0 &= x \\ x^1 &= f^1(x^0) \\ &\dots \\ x^k &= f^k(x^{k-1}) \\ y &= f^{k+1}(x^k) \end{aligned} \tag{1.1}$$

其中，$x^0 = x$是输入层，$x^i = f^i(x^{i-1}), i = 1,\dots,k$是$k$个隐含层，$y = f^{k+1}(x^k)$称为输出层。在神经网络中，$f$ 通常定义为x的线性函数或非线性函数的组合，例如常见的全连接层的定义为

$$f(x) = \sigma(wx + b) \tag{1.2}$$

这些简单函数的复合可以逼近复杂函数，而每个网络层的w, b等参数一起构成了神经网络的参数，x的每个元素是一个神经元。神经网络学习的任务是通过大量的训练数据来找到各网络层的参数。

1.5.2　激活函数

网络层要加入非线性成分，因为多个线性处理的复合还是线性的，只有线性处理的神经网络的函数模拟能力有限。可以如同公式1.2一样，在线性处理后加入一个非线性的成分，称为激活函数。当然也可以把激活函数定义为一个单独的层。常用的激活函数有sigmoid、tanh、relu函数等。因为神经网络是使用梯度下降法完成参数训练，sigmoid函数在多数区域梯度为0，会影响训练的速度，因此常常使用relu函数作为激活函数。

$$\text{relu}(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \tag{1.3}$$

1.5.3　损失函数与优化计算

给定一批训练数据$x_1, y_1, x_2, y_2, \dots, x_n, y_n$。为了计算网络参数，定义一个损失函数来衡量网络预测与真实数据之间的差异，通过在训练集上最小化损失函数来获得对参数的估计。假设神经网络用函数$f_w(x)$ 表示，用神

经网络进行回归处理时，常用均方误差(MSE)作为损失函数，

$$loss = \sum_{i=1}^{n}(f_w(x_i) - y_i)^2 \tag{1.4}$$

当用神经网络进行分类处理时，输出一个概率向量，用负的对数似然作为损失函数，也称为交叉熵损失函数，

$$loss = -\sum_{i=1}^{n} y_i \log(f_w(x_i)) \tag{1.5}$$

其中y_i用one-hot形式表示。

由于神经网络$f(x)$的线性和非线性函数都是基本光滑的，因此损失函数也是光滑的。神经网络中用梯度下降法来最小化损失函数，从而获得最佳的参数。虽然主要应用梯度下降，但在具体的计算中，嵌入了不同的处理手段，目前常用的优化算子有随机梯度下降(SGD)，自适应学习率优化算子AdaGrad, RMSProp等。

1.5.4　图像处理与卷积层

用一个2值图片1.2(a)为例，其中有一个“7”字。图1.2(b)是一个尺寸较小的模板，称为卷积核。将模板在图片1.2(a)上滑动，并与覆盖的区域作乘法，

$$F(x,y) = \sum_{i=0}^{a-1}\sum_{j=0}^{b-1} I(x+i, y+i)w(x+i, y+i)$$

其中F是输出特征，I是输入的图片，w是卷积核，a和b是卷积核的尺寸。这个操作称为卷积(跟一般信号处理的严格定义有差别)。卷积的结果按位置列在图1.2(c)中。可以发现，在位置(2,3)处出现最大的卷积值5。这个示意图说明通过卷积操作，能大致判断在图片1.2(a) 的(2,3) 处，可能有一个“7”字存在。

卷积核可以有不同的尺寸，我们还可以用两个小一点的模板组合起来处理，如模板1.2(d)通过卷积判断有一个水平线，模板1.2(e)通过卷积判断有一个垂直线，则很可能有一个“7”存在。组合的方法可以避免使用复杂的卷积核。通常情况下，图像处理中卷积核是方形的。

在进行卷积操作时，除了卷积核尺寸可以不同外，在图片上滑动的步幅也可以不同。卷积核的尺寸和卷积的步幅，可以影响到输出的尺寸。

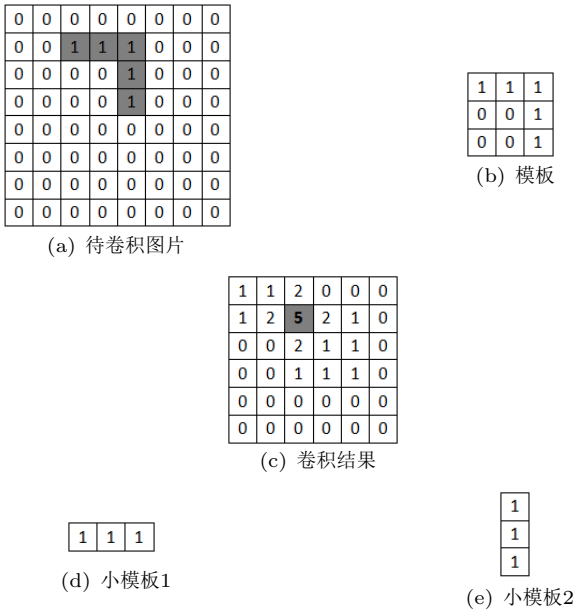

图 1.2: 卷积操作示意图

卷积处理能获取空间关系的特征，从而特别适用于图像处理。神经网络中的卷积层，由一批不同值的卷积核构成，它们进行卷积操作后的输出，相当于提取了一次图片特征。下一个网路层还可以继续是卷积层，提取更高级的特征。

1.5.5　池化层

神经网络的一个网络层输入的数据量越多，需要配置的参数就越多，计算耗费的时间就越大，训练也就越困难。而图片作为一个空间信息的载体，大部分邻近的信息是相同的，因此通过采样的方式，扔掉一部分数据，但图片基本特征还能保持。池化层完成的就是这种采样来缩小输入的工作。跟卷积操作类似，用一个池化窗口在图片上滑动，池化结果是窗口内数据取平均值或者最大值，称为平均池化或者最大池化。数据缩小的程度与窗口尺寸和步幅相关。

1.5.6　丢弃处理Dropout

神经网络具有大量的参数，有很强的数据拟合能力。但是拟合能力过强，在用来预测时并不一定有好的效果，因为训练数据本身可能会有误差，实际模型也可能并没有那么复杂。避免过拟合的方法之一是训练过程中，随机丢弃一些神经元，使得网络结构变小，从而能部分抑制过拟合。另外，丢弃一些神经元，可以迫使网络其他的神经元能学到一些更一般的特征。

1.5.7　基于keras的卷积神经网络处理框架

准备数据

```
(x_Train,y_Train),(x_Test,y_Test)=mnist.load_data()
x_Train=x_Train.reshape(
    x_Train.shape[0],28,28,1).astype('float32')/256
x_Test=x_Test.reshape(
    x_Test.shape[0],28,28,1).astype('float32')/256
y_Train= to_categorical(y_Train)
y_Test = to_categorical(y_Test)
```

构建神经网络模型

```
model = Sequential()
model.add(Conv2D(filters=16,  #2维卷积层，16个卷积核
          kernel_size=(5,5),   #卷积核的尺寸
          padding='same',
          input_shape=(28,28,1),
          activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(filters=36,
          kernel_size=(5,5),
          padding='same',
          activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())            #展开成一维向量
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
```

```
model.add(Dense(10,activation='softmax'))
```

训练和评估神经网络模型

```
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['accuracy'])
train_history=model.fit(x=x_Train,
                        y=y_Train,
                        validation_split=0.2,
                        epochs=15,
                        batch_size=300,
                        verbose=2)
scores = model.evaluate(x_Test,y_Test,batch_size=512)
```

1.6　实验报告要求

实验报告需包含实验任务、实验平台、实验原理、实验步骤、实验数据记录、实验结果分析和实验结论等部分，特别是以下重点内容：

1. Keras中关于神经网络的开发步骤。
2. 卷积神经网络超参数的调整。
3. 卷积神经网络进行图像分类的性能分析与研究。

1.7　考核要求与方法

实验总分100分，通过实验报告进行考核，标准如下：

1. 报告的规范性10分。报告中的术语、格式、图表、数据、公式、标注及参考文献是否符合规范要求。
2. 报告的严谨性40分。结构是否严谨，论述的层次是否清晰，逻辑是否合理，语言是否准确。
3. 实验的充分性50分。实验是否包含“实验报告要求”部分的3个重点内容，数据是否合理，是否有创新性成果或独立见解。

1.8　案例特色或创新

本实验的特色在于：培养学生应用Keras平台搭建神经网络，能够理解并应用卷积神经网络实现手写数字识别，能够调整神经网络参数，提高神经网络性能，对实验结果进行有效的可视化展示，培养学生对复杂工程问题建模和分析的能力。