探索“卷积神经网络”

3.1 深入浅出话“卷积”

本章主要介绍卷积神经网络(Convolutional Neural Networks,CNN)。在此之前,需要先对“二维卷积”(以下简称“卷积”)进行深入的了解,它是研究卷积神经网络的前提和基础。

本节主要讲解的问题如下:

- 卷积是如何实现的;
- 为什么要进行卷积运算。

3.1.1 卷积的运算过程

从系统工程的角度看,卷积是为研究系统对输入信号的响应而提出的,卷积有很多种,本节着重介绍二维滑动卷积。

滑动卷积涉及 3 个矩阵:第一个矩阵通常尺寸较大且固定不动,本书称之为“输入矩阵”(或“待处理矩阵”);第二个矩阵尺寸较小,在输入矩阵上从左到右、从上到下进行滑动,本书称之为“卷积核”;卷积核在输入矩阵上面滑动的过程中,将对应的两个小矩阵的相应元素相乘并求和,结果依次作为第三个矩阵元素,本书称该矩阵为“特征矩阵”。上述 3 个矩阵及卷积运算符如图 3.1.1 所示。

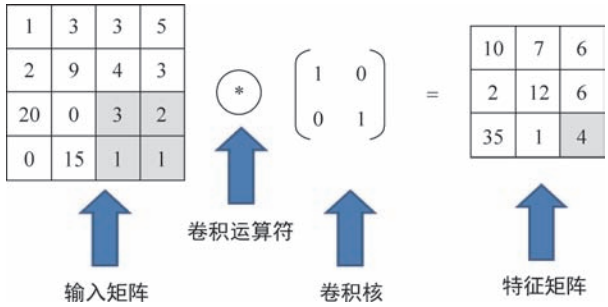

图 3.1.1　输入矩阵、卷积核、特征矩阵及卷积运算符

下面详细介绍滑动卷积的运算过程。

将图 3.1.1 中所示的两个矩阵进行卷积运算。

第 1 步(见图 3.1.2)：

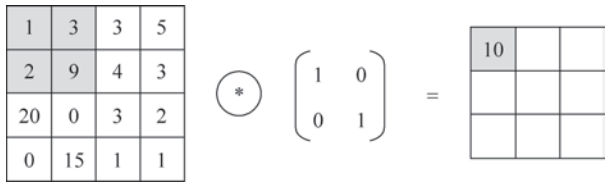

图 3.1.2　滑动卷积运算第 1 步

计算过程如下：

$$(1\times 1)+(3\times 0)+(2\times 0)+(9\times 1)=10$$

第 2 步(见图 3.1.3)：

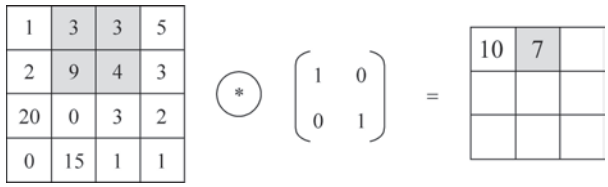

图 3.1.3　滑动卷积运算第 2 步

第 3 步(见图 3.1.4)：

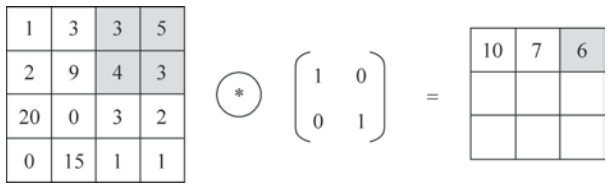

图 3.1.4　滑动卷积运算第 3 步

完成第一行的运算之后，上述运算过程就从下一行开始从左到右继续进行，如图 3.1.5 所示。

第 4 步(见图 3.1.5)：

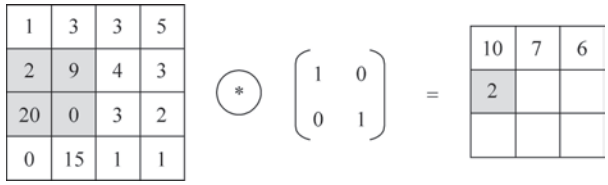

图 3.1.5　滑动卷积运算第 4 步

重复相同的步骤，直到全部完成(见图 3.1.6)。

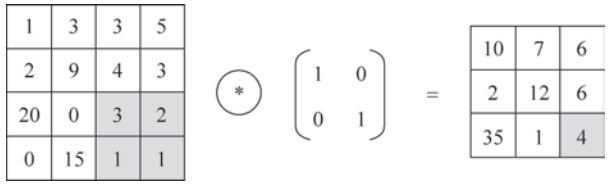

图 3.1.6　滑动卷积运算最后一步

在上述卷积运算的过程中，每次都是滑动 1 个像素。当然，也可以每次滑动 2 个或多个像素。每次滑动的像素个数称为步长(stride)。

请读者仔细观察如图 3.1.6 所示的输入矩阵与特征矩阵的元素的个数，不难发现，特征矩阵元素的个数少于输入矩阵的元素个数(请读者思考其中的原因)。如果需要得到与输入矩阵元素个数相等的特征矩阵该如何处理呢？方法很简单，需要对输入矩阵的边缘添加 0 元素，这个过程称为零填充(zero padding)。通过零填充，实现滑动卷积的过程如图 3.1.7 所示。

■ 温馨提示

在二维滑动卷积运算过程中，卷积核在滑动过程中始终都在输入矩阵内部，所得到的特征矩阵的元素个数会比输入矩阵的元素个数少，在程序中称这种滑动卷积方式为 valid；如果采用零填充的方式，使特征矩阵元素的个数与输入矩阵元素个数相同，在程序中称这种滑动卷积方式为 same。

3.1.2　卷积核对输出结果的影响

如图 3.1.8 所示，特征矩阵的(3,1)元素的值最大。那么，为什么该元素的值最大呢？通过观察输入矩阵和卷积核元素的特征可知：(3,1)元素所对应的子矩阵与卷积核在形态上类似，二者都是对角矩阵，而且相同位置上的数值都较大。由此可见，子矩阵与卷积核在形态上类似时，卷积运算就会生成一个较大的值。

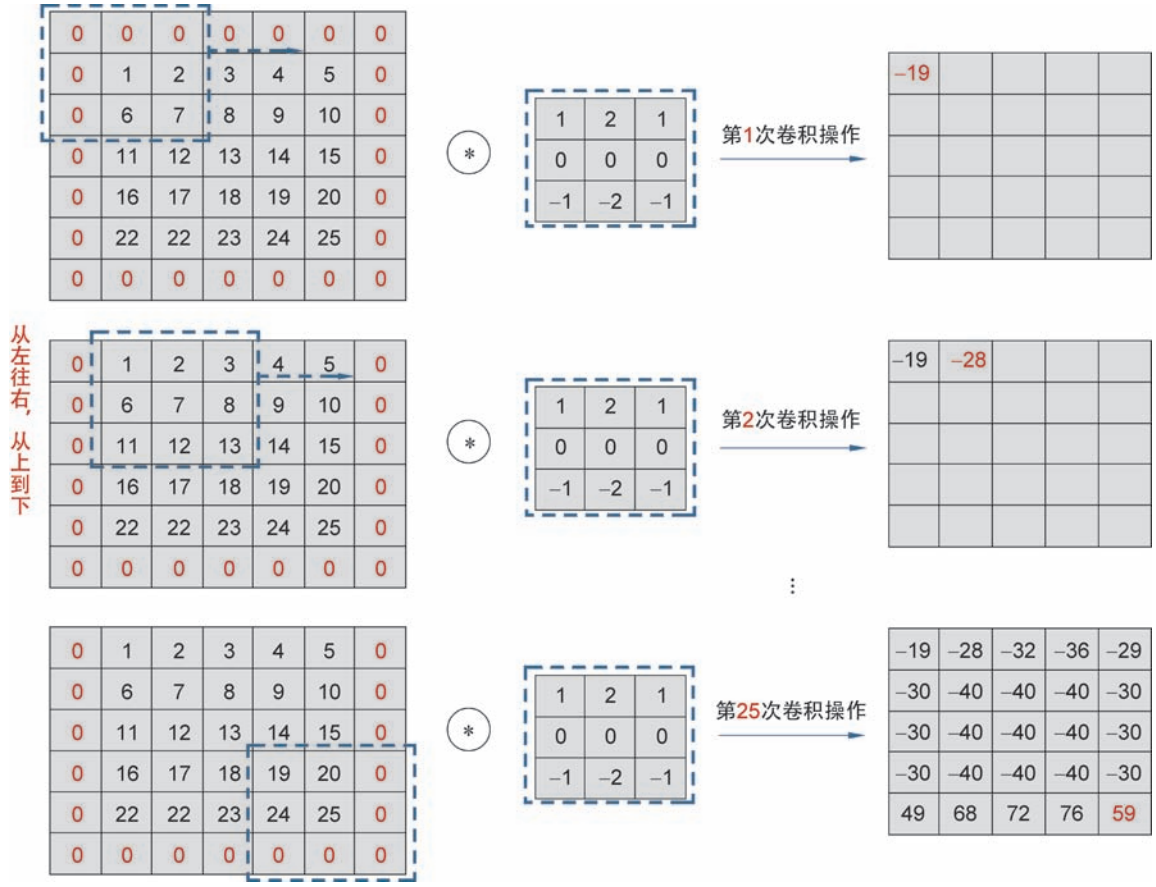

图 3.1.7　通过零填充实现滑动卷积的过程示意图

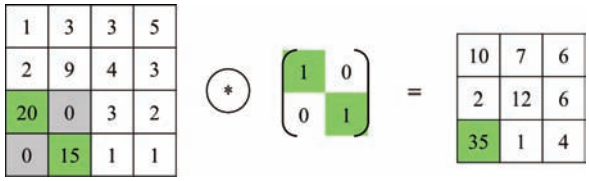

图 3.1.8　特征矩阵中(3,1)元素的计算过程

如图 3.1.9 所示，输入矩阵中(3,1)元素的值为 20，在输入矩阵中的值是最大的，但通过卷积运算后结果为 2，原因是子矩阵与卷积核的形态差异很大。

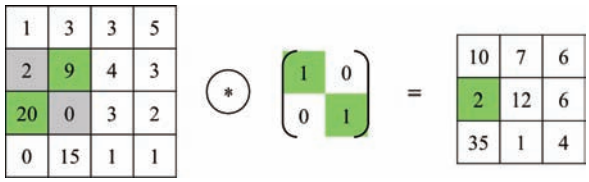

图 3.1.9　特征矩阵中(2,1)元素的计算过程

如果要使特征矩阵中(2,1)元素的值变大，可以将卷积核更换为和对应的子矩阵与卷积核在形态上类似的，如图 3.1.10 所示。

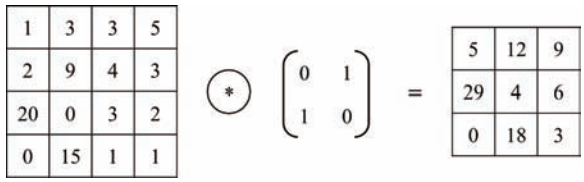

图 3.1.10　更换卷积核后特征矩阵中(2,1)元素的计算过程

■ 一语中的

由上面的分析可知，对二维数字图像进行卷积运算，可以判断图像的像素与卷积核的相似程度，相似程度越高，得到的响应值越大，因此可以通过滑动卷积运算来提取图像的特征。

3.1.3　卷积运算在图像特征提取中的应用

当给定一个“A”的图像，计算机怎么识别这个图像就是“A”呢？一个可能的办法就是计算机存储一张标准的“A”图像，然后把需要识别的未知图像跟标准图像进行比对，如果二者一致，则判定未知图像即是一个“A”图像。对于计算机来说，只要图像稍稍有一点变化，便会造成识别的困难，如图 3.1.11 所示。

这是因为在计算机的“眼”中，一幅图像看起来就像是一个二维的像素数组(如同棋盘或马赛克)，每一个位置对应一个数字。在这个例子当中，像素值 1 代表白

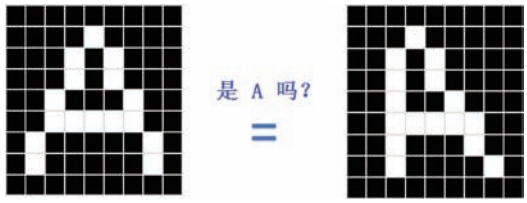

图 3.1.11　图像变化造成的识别困难

色,像素值－1 代表黑色,如图 3.1.12 所示。

■ 温馨提示

本段中所涉及的数字图像处理的相关知识，请读者参阅本节后面的扩展阅读——数字图像处理的基础知识。

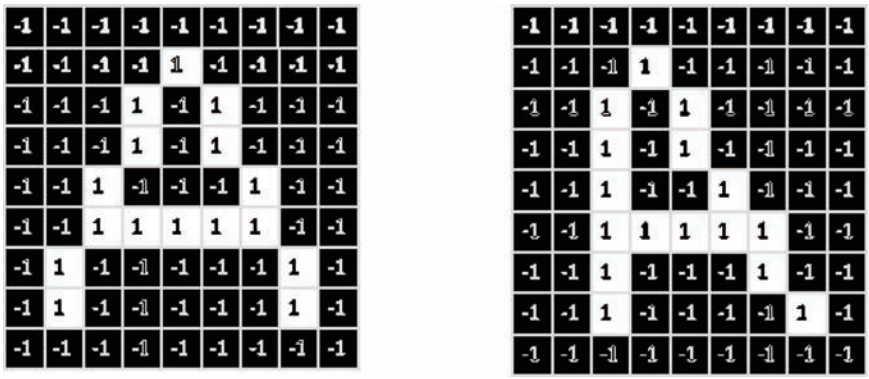

图 3.1.12　计算机中数字图像的存储形式

对于这个例子,计算机认为上述两幅图中的白色像素除了中间的 6 个小方格里面是相同的,其他 4 个角上都不同。因此,计算机判别右边那幅图不是“A”,两幅图不同(见图 3.1.13),这显然不合理。

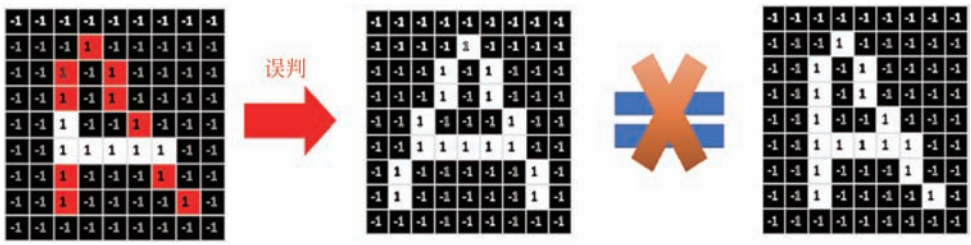

图 3.1.13　计算机出现了不合理的误判

针对上述问题,可以通过二维卷积运算来提取图像的特征,从而提高识别的准确率。具体思路如下：可以提取一些局部特征,通过这些特征来进行匹配,从而实

现识别(见图 3.1.14)。

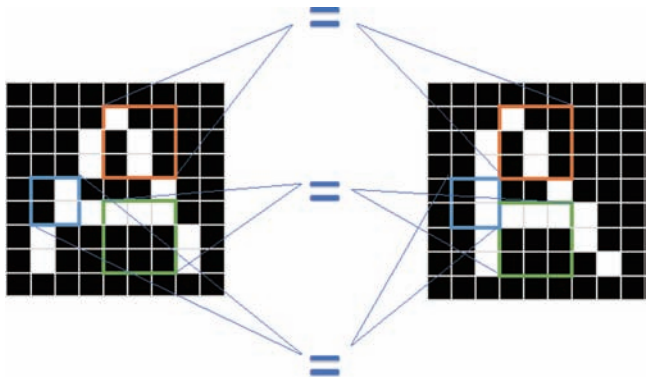

图 3.1.14　通过提取局部特征来进行匹配识别

因此,可以来设计一些具有某种特征的卷积核,通过图像与卷积核进行卷积运算,来提取特征。

再回到上面这个例子,可以设计如图 3.1.15 所示的卷积核,来提取输入图像的相应特征,如图 3.1.16～图 3.1.18 所示。

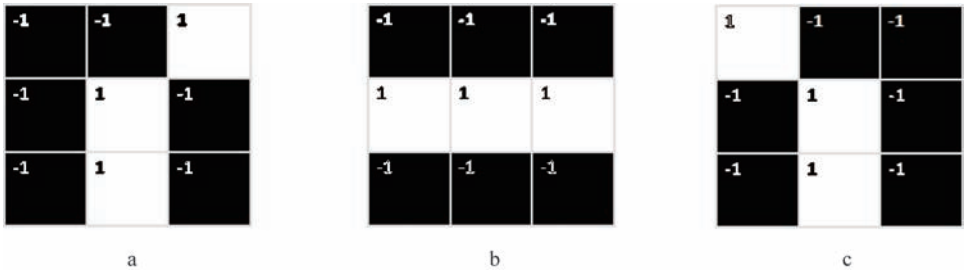

图 3.1.15　卷积核

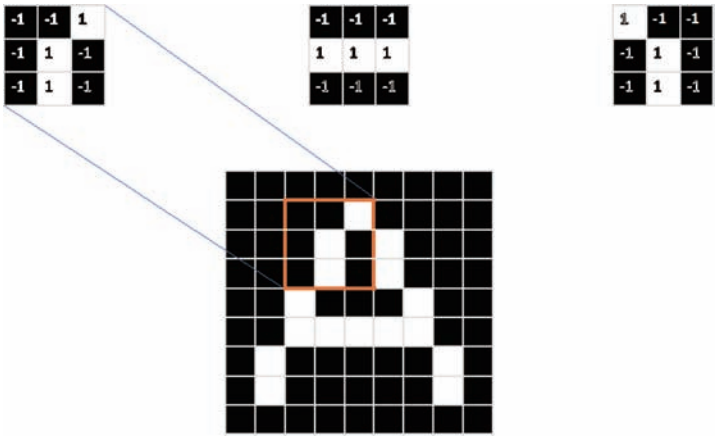

图 3.1.16　通过卷积核 a 提取“A”的左上边缘特征

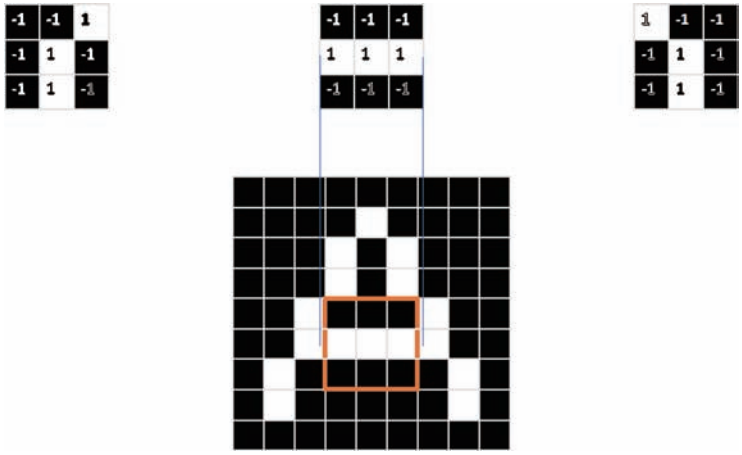

图 3.1.17　通过卷积核 b 提取“A”的中心线特征

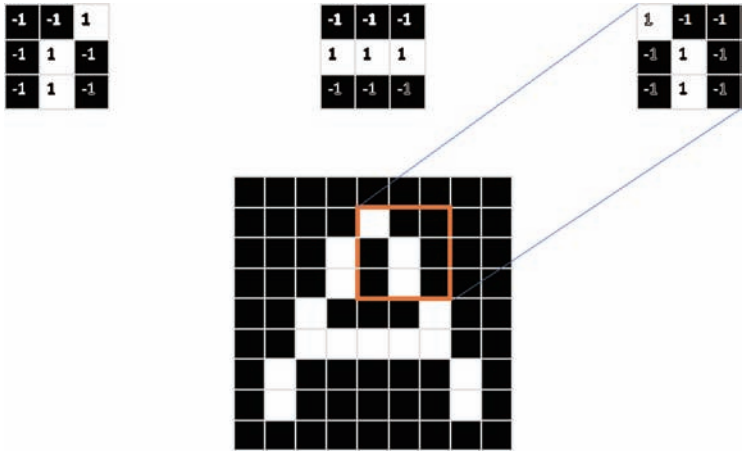

图 3.1.18　通过卷积核 c 提取“A”的右上边缘特征

| 扩展阅读 |

数字图像处理的基础知识

数字图像是由一个一个的“小点”组成，把这样的小点称为“像素”。“像素”的英文为 pixel，它是 picture 和 element 的合成词，表示图像元素的意思。可以对“像素”进行如下理解：像素是一个面积概念，是构成数字图像的最小单位。像素的大小与图像的分辨率有关，分辨率越高，像素就越小，图像就越清晰。图 3.1.19 所示的是不同像素图像之间的比较。

在数字图像中，每个像素点亮度的大小称为“灰度”。

了解了“像素”和“灰度”的概念之后，一幅二维的像素为 $M\times N$ 的数字图像可以表示为一个 $M\times N$ 矩阵，矩阵中每个元素的值为其所对应的像素的灰度。

(a) 像素为320×240的图像　　(b) 像素为80×60的图像

图 3.1.19　像素不同的图像比较

下面介绍几种常见的数字图像类型。

- 黑白图像：图像的每个像素只能是黑或白，没有中间的过渡，故又称为二值图像。二值图像的像素值为 0、1。黑白图像及其矩阵表示如图 3.1.20 所示。
- 灰度图像：灰度图像是指每个像素的信息由一个量化的灰度级来描述的图像，没有彩色信息。灰度图像及其矩阵表示如图 3.1.21 所示。

图 3.1.20　黑白图像及其矩阵表示

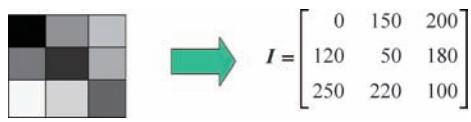

图 3.1.21　灰度图像及其矩阵表示

- 彩色图像：彩色图像是指每个像素的信息由 RGB 三原色构成的图像，其中 RGB 是由不同的灰度级来描述的。彩色图像及其矩阵表示如图 3.1.22 所示。

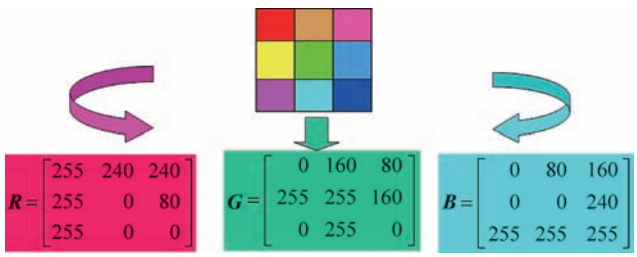

图 3.1.22　彩色图像及其矩阵表示

编程体验 1

读入一幅数字图像并显示

将本书配套资料中的图片 study.jpg 复制到 C:\我的文档\ MATLAB 文件夹下。（注：由于版本及安装路径不同，MATLAB 文件夹的路径也不相同，请读者按

照自己计算机上安装 MATLAB 的实际情况进行操作,作者计算机中的路径为 C:\Users\zhao\Documents\MATLAB)。

在 MATLAB 的命令窗口输入如图 3.1.23 所示的命令。该程序实现了读入一幅 RGB 图像、查看图像的大小和维度、显示图像的功能。该程序的运行效果如图 3.1.24 所示。

```
命令行窗口
>> % 读入图像
I=imread('study.jpg');
% 查看所读入图像的大小和维度
size(I)
%显示图像
imshow(I)
```

图 3.1.23　基于 MATLAB 实现读入图像并显示的指令

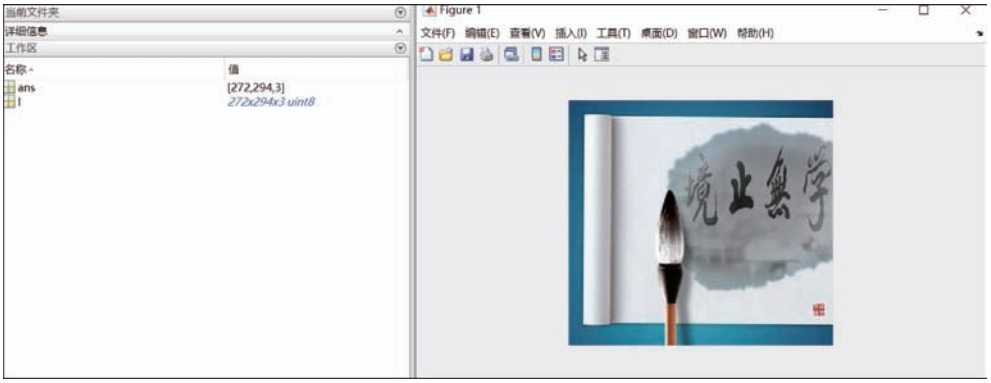

图 3.1.24　编程体验 1 的运行效果

编程体验 2

基于 MATLAB 实现二维图像的滑动卷积

在 MATLAB 的命令窗口输入如图 3.1.25 所示的命令。该程序构造了一个卷积核 L,对输入的图像 I 进行卷积运算,并生成特征矩阵 C。仔细观察卷积核 L,其中心元素的值和周围元素的值差别很大,因此用 L 对 I 进行卷积运算,图像 I 中的发生像素值突变的边缘被提取了出来,如图 3.1.26 所示。

```
Command Window
>> %读入图像
I = imread('coins.png');
%显示输入图像
imshow(I)
%构造卷积核
L= [ 0 1 0
     1 -4 1
     0 1 0];
%进行数据转换
I=double(I);
L=double(L);
%进行卷积运算(本例中调用MATLAB中的conv2函数)
C=conv2(I,L,'same');
%进行数据转换
C=uint8(C);
figure
%显示卷积后的图像
imshow(C)
```

图 3.1.25　基于 MATLAB 实现二维图像的滑动卷积指令

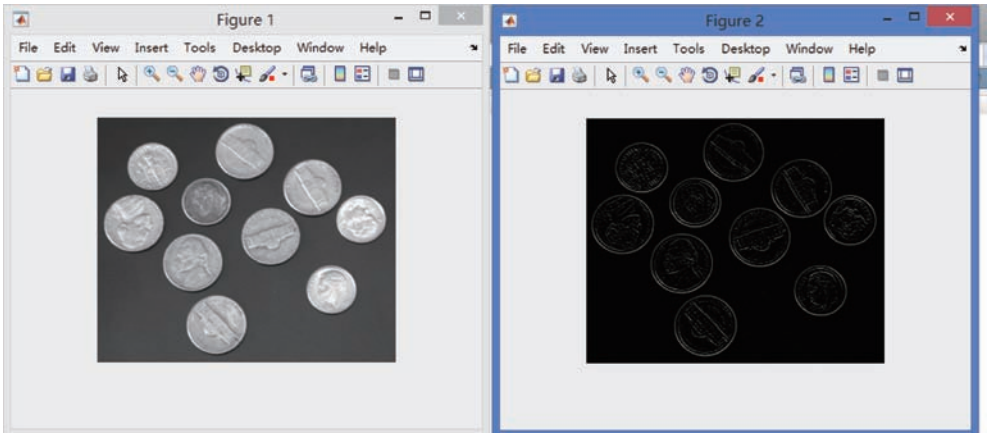

图 3.1.26　程序运行结果(注：左侧为输入图像，右侧为卷积后的结果)

3.2　解析“卷积神经网络”

3.1 节详细讲解了“卷积”的运算过程及其在图像特征提取方面的作用。在此基础之上，从 3.2 节开始将详细讲解本章的核心内容——“卷积神经网络”。

本节主要讲解的问题如下：

- 卷积神经网络是如何构成的？每一部分的功能是什么？
- 与传统的全连接深度神经网络相比，卷积神经网络有什么不同？
- 从仿生学的角度，如何理解卷积神经网络？

3.2.1　从 ImageNet 挑战赛说起

通过计算机的“眼睛”识人辨物是科研工作者追求的梦想,并一直为之努力。传统的方法是对输入的图像进行预处理,将其通过某种或某几种算法提取特征,再通过机器学习的算法进行分类。

ImageNet 挑战赛是由斯坦福大学计算机科学家李飞飞组织的年度机器学习竞赛(设立 ImageNet 挑战赛的目的详见本节的“扩展阅读”)。在比赛中,参赛队伍会得到超过一百万张图像的训练数据集,每张图像都被手工标记一个标签,大约有 1000 种类别。参赛队伍开发的图像分类程序对未包含在训练集内的其他图像进行分类,程序可以进行多次猜测,如果前 5 次猜测中有一次与人类选择的标签相匹配,则被判为成功。

在 2010 年首届 ImageNet 挑战赛上,冠军团队采用“特征提取+支持向量机”的方法,分类错误率为 28.2%。2011 年,冠军参赛队伍对特征提取方法进行了优化和改进,将分类错误率降低到了 25.7%。如果将 ImageNet 挑战赛所用的数据集交给“人眼”去分类,那错误率又是多少呢?答案是 5.1%。

传统的方法产生“瓶颈”的原因是:无法提取用于图像分类的“有效特征”。例如,如何识别图像中的物体是苹果,根据形状还是根据颜色?如何表达“物体有没有腿”这样抽象的概念?

2012 年的 ImageNet 挑战赛具有里程碑意义。Alex Krizhevsky 和他的多伦多大学的同事在该项比赛中首次使用深度卷积网络,将图片分类的错误率一举降低了 10 个百分点,正确率达到 84.7%。自此以后,ImageNet 挑战赛变成为卷积神经网络比拼的舞台,各种改进型的卷积神经网络如雨后春笋,层出不穷。2015 年,微软研究院的团队将错误率降低到了 4.9%,首次超过了人类。到了 2017 年,ImageNet 挑战赛的冠军团队将图像分类错误率降低到了 2.3%,这也是 ImageNet 挑战赛举办的最后一年,因为卷积神经网络已经将图像分类问题解决得很好了。

近年来,卷积神经网络之所以展现出良好的发展势头,是因为卷积神经网络可以自主地提取输入信息的“有效特征”,并且进行层层递进抽象;卷积神经网络之所以能够提取输入信息的“有效特征”,是因为其包括多个卷积层。图 3.2.1 展示了获得 2012 年 ImageNet 挑战赛冠军的 AlexNet,这个神经网络的主体部分由 5 个卷积层和 3 个全连接层组成,该网络的第一层以图像为输入,通过卷积及其他特定形式的运算从图像中提取特征,接下来每一层以前一层提取出的特征作为输入并进行卷积及其他特定形式的运算,便可以得到更高级的特征。经过多层的变换之后,深度网络就可以将原始图像转换成高层次的抽象特征。

上述由低级到高级的抽象过程和人类的认知过程相似。举个例子来说,在汉语的学习和理解的过程中,通过笔画的组合,可以得到汉字;通过汉字的组合,可

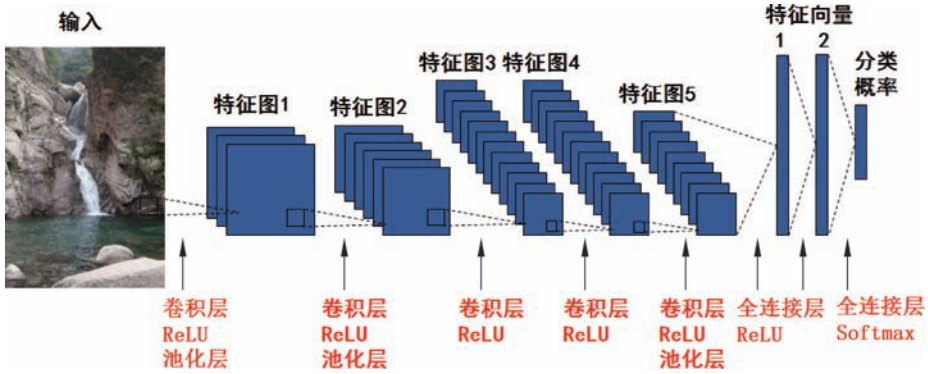

图 3.2.1　AlexNet 网络示意图

以得到词汇；通过词汇的分析，可以了解语义。从笔画到语义，实现了由低级到高级的抽象。这个特征提取与抽象的过程，可用如图 3.2.2 所示的过程来体现。

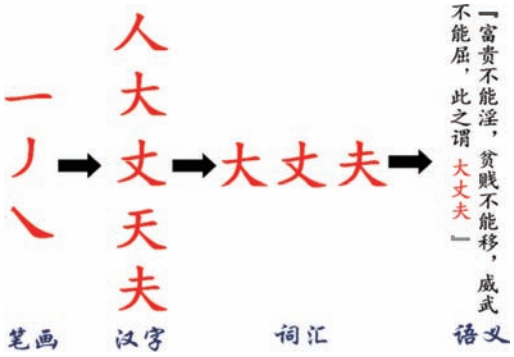

图 3.2.2　汉语言的抽象过程

然而，卷积神经网络并非是在 2012 年才被提出的。20 世纪 60 年代加拿大科学家 David Hubel 和瑞典科学家 Torsten Wiesel 提出了感受野(receptive field)的概念，当时科学家通过对猫的视觉皮层细胞研究发现，每一个视觉神经只会处理一小块区域的视觉图像，即感受野。到了 20 世纪 80 年代，日本科学家 Kunihiko Fukushima 提出神经认知机(neocognitron)的概念，神经认知机中包含两大类神经元——用来提取特征的 S-cells 和用来抗形变的 C-cells。S-cells 的功能与卷积层提取特征类似，C-cells 的功能与激活函数、池化层的功能类似。LeCun 等提出了 LeNet，用于手写字体识别。

3.2.2　卷积神经网络的结构

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络。典型的卷积神经网络结构如图 3.2.3 所示，可以分为卷积层、激活函数、池化层和全连接层。

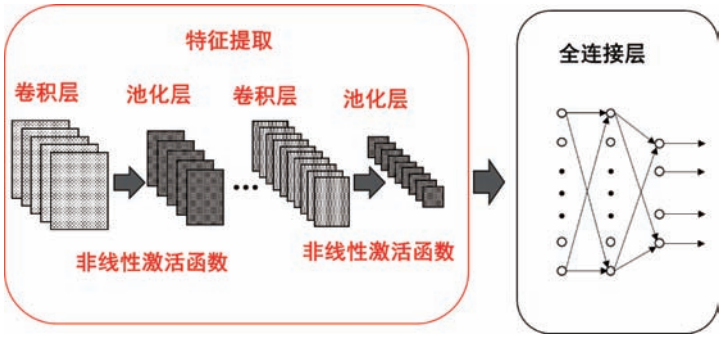

图 3.2.3　典型的卷积神经网络的结构

由于卷积神经网络中的全连接层与本书第 2 章讲解的深度网络的全连接层原理相同,故在本节中不再赘述。下面主要介绍卷积层、非线性激活函数、池化层的工作原理。

3.2.3　卷积层的工作原理

卷积层是通过卷积核对输入信息进行卷积运算(卷积运算的原理详见 3.1 节),从而提取特征的。

一个卷积神经网络往往有多个卷积层,如图 3.2.1 所示的 AlexNet 就含有 5 个卷积层。在基于卷积神经网络的数字图像识别过程中,第一个卷积层会直接接收图像像素级的输入,来提取其与卷积核相匹配的特征,并传递给下一层;接下来每一层都以前一层提取出的特征作为输入,与本层的卷积核进行卷积运算,提取更加抽象的特征。

同一个卷积层中可以有多个不同的卷积核,该卷积层的输入分别和这多个卷积核进行卷积运算,形成新的特征图。由此可见,特征图的个数与该卷积层卷积核的个数相关,该过程如图 3.2.4 所示。

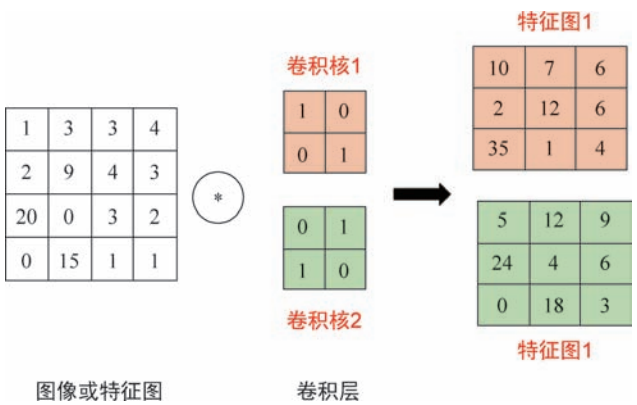

图 3.2.4　一个卷积层含有多个卷积核的计算示意图

在卷积神经网络工作的过程中，无论是输入还是中间过程产生的特征图，通常都不是单一的二维图，可能是多个二维图，每一张二维图称为一个通道(channel)。比如一幅 RGB 图像就是由 R 通道、G 通道、B 通道 3 个通道组成(见图 3.1.22)。对于多通道的输入，每个通道采用不同的卷积核做卷积，然后将对应特征矩阵的元素进行累加即可，其过程如图 3.2.5 所示。

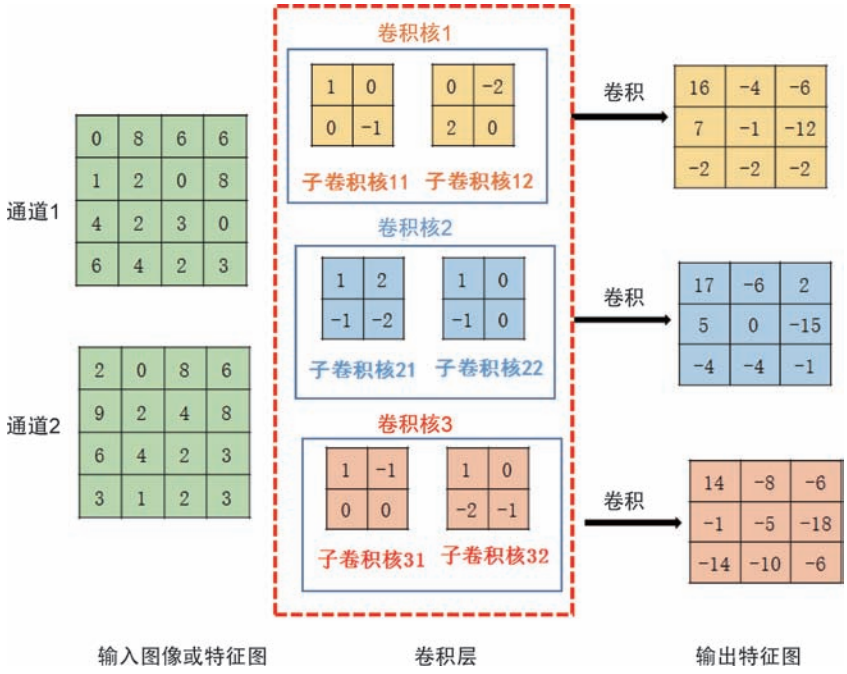

图 3.2.5　多通道多卷积核的计算过程示意图(注：卷积过程中步长为 1)

在如图 3.2.5 所示的多通道多卷积核的计算过程中，输入为两个通道，卷积层中有 3 个卷积核，每个卷积核又分为两个子卷积核，其中第一个子卷积核与通道 1 的输入进行卷积，第二个子卷积核与通道 2 的输入进行卷积，然后将两个卷积结果的对应元素进行相加。例如，在第一个输出特征矩阵中，(1,1)元素的计算过程如下：

$$[0\times1+8\times0+1\times0+2\times(-1)]+[2\times0+0\times(-2)+9\times2+2\times0]=16$$

在卷积神经网络中，卷积核中的元素是需要通过训练确定的，称之为参数。这里要讲解一个重要的概念——“参数共享”(parameter sharing)。所谓“参数共享”，是对于一幅输入图像或特征图，在进行卷积的过程中，其每个位置都是用同一个卷积核去进行运算的，即每个位置和同一组参数进行相乘，然后相加。

对于卷积神经网络来说，“参数共享”有什么意义呢？通过以下这个例子来说明。如果输入一幅像素为 1000×1000 的灰度图像，其输入为 1 000 000 个点，在输入层之后如果是相同大小的一个全连接层，那么将产生 1 000 000×1 000 000 个连接，也就是说，这一层就有 1 000 000×1 000 000 个权重参数需要去训练；如果输入

层之后连接是卷积层,该卷积层有 6 个卷积核,卷积核的尺寸为 5×5,那么总共有(5×5+1)×6=156 个参数需要去训练(注:括号中的 1 代表同一个卷积核的偏置,将在 3.3 节中介绍如何训练卷积核的参数)。由此可见,与全连接层相比,卷积层需要训练的参数要减少很多,从而降低了网络的复杂度,提高了训练效率,避免了过多连接导致的过拟合现象。

■ 经验分享

卷积层的作用主要体现在两个方面:一是提取特征;二是减少需要训练的参数,降低深度网络的复杂度。

3.2.4　非线性激活函数的工作原理

我们需要在每个卷积层之后加入非线性激活函数。之所以要加入非线性激活函数,原因如下:卷积运算是一种线性运算,线性运算有一个性质——若干个线性运算的叠加可以用一个线性运算来表示;如果将多个卷积运算直接堆叠起来,虽然进行了很多层卷积运算,但多层卷积运算可以被合并到一起并用一个卷积运算来代替,这与用多个卷积核设置多个卷积层来提取图像的不同特征并进行高级抽象的初衷是违背的。因此,在每个卷积层后面加一个非线性激活函数,那么每个卷积层的效果就可以得到“保留”。

非线性激活函数有很多种,ReLU 函数是卷积神经网络中常用的一种,它的表达式为 $f(x)=\max(0,x)$,对于输入的特征向量或特征图,它会将小于零的元素变为零,保持其他元素不变。由于 ReLU 函数的计算非常简单,所以它的计算速度往往比其他非线性函数快,加之其在实际应用中的效果很好,因此在很多深度网络中被广泛使用。

■ 经验分享

为了加深对非线性激活函数的理解,我们可以拿中医中的“针灸”作为类比。当针与皮肤有一段距离时,人不会感到疼痛,针与皮肤的远近和大脑中的“痛感”没有关系;当针接触到皮肤并且扎进皮肤时,人就会感到疼痛,也就是大脑中的“痛感”被激活了,针扎进皮肤的距离与大脑中的“痛感”具有相关性。非线性激活函数就是这个原理,神经网络训练出来的信息,如果没有达到阈值,说明是无用信息;如果超过阈值,特征就会通过非线性激活函数传递下去。

3.2.5　池化层的工作原理

池化(pooling)操作实质上是一种对统计信息提取的过程。在卷积神经网络中,池化运算是对特征图上的一个给定区域求出一个能代表这个区域特殊点的值,

常见的两种池化方法是最大池化(max-pooling)和平均池化(average-pooling)。

图 3.2.6 是最大池化示意图,将整个矩阵分为多个子区域,取每个子区域的最大值作为新矩阵中的对应元素。

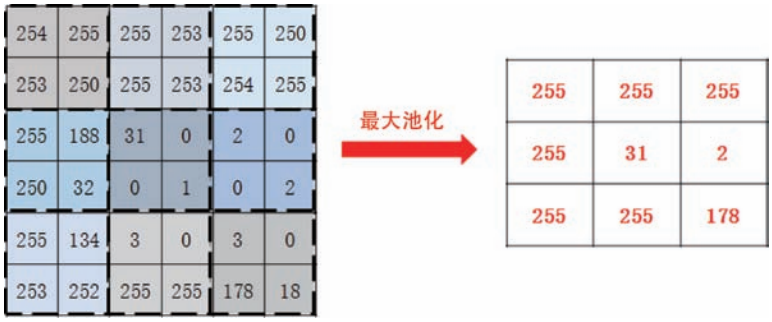

图 3.2.6　最大池化示意图

图 3.2.7 是平均池化示意图,与最大池化不同的是,它是取每个子区域的平均值作为新矩阵中的对应元素。

图 3.2.7　平均池化示意图

■ 温馨提示

池化操作也可以按照一定的步长（stride）来进行。图 3.2.6 和图 3.2.7 中池化操作的步长为 2。在实际的卷积网络结构中，池化操作的步长要小于池化区域的边长，这样能使相邻池化区域有一定的重叠，常见的情况是池化步长等于池化区域的边长减 1，比如,池化区域为 2×2，步长可以设为 1。

池化层的主要作用表现在两个方面：

(1) 减少特征图的尺寸。从上面的分析可知,特征图在经过池化后,尺寸减小了,这对于减少计算量和防止过拟合是非常有利的。

(2) 引入不变性。比如最常用的最大池化是选取特征图子区域中最大的那个值,所以这个最大值无论在子区域的哪个位置,通过最大池化运算总会选到它；所以这个最大值在这个子区域内的任何位移对运算结果都不会产生影响,相当于对

微小位移的不变性。

3.2.6　卷积神经网络与全连接神经网络的区别

通过上面的介绍,大家已经对卷积神经网络的结构和各部分的功能有了一定的了解。那么,本节讲到的卷积神经网络与第 2 章讲的全连接神经网络有哪些区别呢?

区别 1:架构上的区别。

全连接神经网络为“平面网络”,主要由输入层、激活函数、全连接层组成;卷积神经网络为“立体网络”,其组成包括输入层、卷积层(可能有多个)、激活函数(可能有多个)、池化层(可能有多个)、全连接层,两者的区别如图 3.2.8 所示。

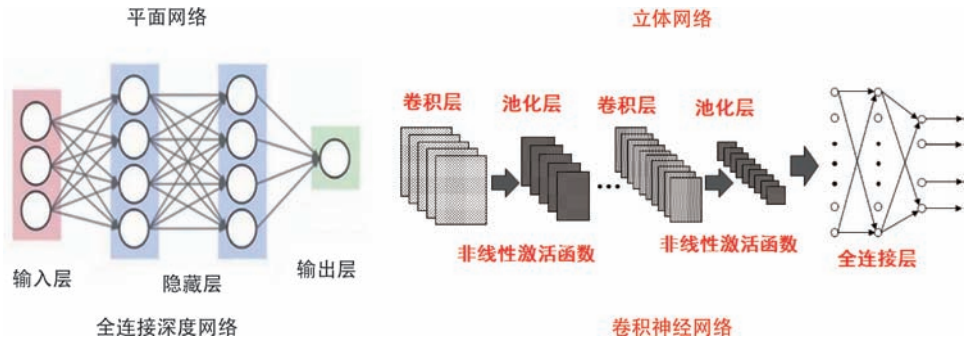

图 3.2.8　架构上的区别

区别 2:功能上的区别。

全连接深度神经网络:无法对输入量进行特征提取;卷积神经网络:可以实现特征自动提取功能。

3.2.7　从仿生学角度看卷积神经网络

1981 年,诺贝尔医学生理学奖颁发给了 David Hubel,他发现了视觉系统信息处理机制,证明大脑的可视皮层是分级的。David Hubel 认为人的视觉功能主要有两个:一个是抽象,一个是迭代。抽象就是把非常具体的形象的元素抽象出来形成有意义的概念;这些有意义的概念又会往上迭代,变成更加抽象,从而使人可以感知到的抽象概念。

如果要模拟人脑,就要模拟抽象和递归迭代的过程,把信息从最细微的像素级别抽象到“属性”的概念,让人能够接受。卷积神经网络的工作原理便体现了这一点,如图 3.2.9 所示。因此,从仿生学的角度来看,卷积神经网络是一种模仿大脑的可视皮层工作原理的深度神经网络。

卷积神经网络在图像分类、目标检测、图像分割等方面应用效果显著,极大地推动了计算机视觉技术的发展及应用。相关的应用实例将在本书第 5 章结合 MATLAB 的 Deep learning Toolbox 的程序代码进行讲解。

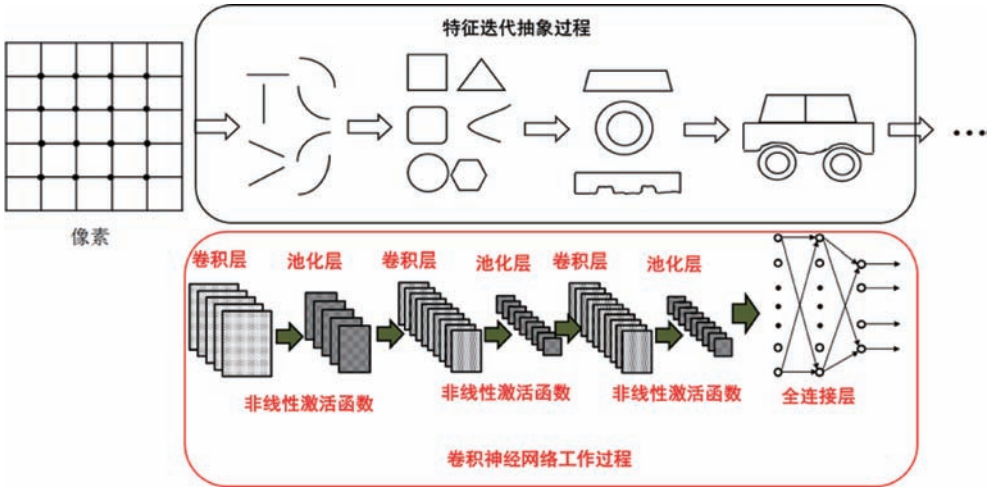

图 3.2.9　卷积神经网络对特征迭代抽象过程示意图

| 扩展阅读 |

创建 ImageNet 挑战赛初衷*

以下是《医学与机器》栏目的主持人、《深度医学》的作者 Eric J. Topol 博士与李飞飞教授的访谈摘录。

李飞飞教授：“我在 AI 领域的专业方向主要是研究计算机视觉与机器学习之间的交集。早在 2006 年，我就试图解决计算机视觉领域的一个核心难题——用直白的话来说，就是如何实现物体识别。

“人类是一种非常聪明的动物，会以非常丰富的方式观察这个世界。但是，这种视觉智能的基础在于准确识别出周遭环境中多达几十万种不同的物体，包括小猫、树木、椅子、微波炉、汽车、行人等。从这个角度出发探索机器智能的实现，无疑是实现人工智能的第一步，而且直到现在也仍然是重要的一步。

“我们一直在为此努力。当时我还年轻，在学校担任副教授。在评上副教授的第一年，我就开始研究这个问题。但我突然间意识到，那个时代下的所有机器学习算法，在本质上只能处理含有几十种对象的一组极小数据类别。这些数据集中的每个类别只包含 100 张或者最多几百张图片，这样的素材量远远无法与人类及其他动物的实际成长经历相契合。

“受到人类成长过程的启发，我们意识到大数据对于推动机器学习发展的重要意义。充足的数据量不仅能够改善模式的多样性，同时也在数学层面有着关键的积极意义，能够帮助一切学习系统更好地实现泛化，而非被束缚在总量远低于真实

* 本文节选自公众号“AI 前线”2020 年 2 月 15 日的文章，略有删减。

世界的数据集内,经历一次又一次的过度拟合。

“以这一观念为基础,我们认为接下来不妨做点疯狂的事情,那就是把周遭环境中的所有物体都整理出来。具体是怎么做的?我们受到了英语词汇分类法 WordNet 的启发,这种方法由语言学家 George Miller 于 20 世纪 80 年代提出。在 WordNet 中,能够找到超过 8 万个用于描述客观对象的名词。

“最终收集到 22 000 个对象类,这些对象类通过不同的搜索引擎从互联网上下载而来。此外,还通过 Amazon Mechanical Turk 发动了规模可观的众包工程项目,这一干就是两年。我们吸引到来自 160 多个国家和地区的超过 5 万名参与者,他们帮助我们清理并标记了近 10 亿张图像,并最终得到一套经过精心规划的数据集。数据集中包含 22 000 个对象类以及下辖的 15 000 万张图像,这就是今天大家所熟悉的 ImageNet。

“我们立即把成果向研究社区开源。从 2010 年开始,我们每年举办一届 ImageNet 挑战赛,诚邀全球各地的研究人员参与解决这一代表计算机视觉领域终极难题的挑战。

“几年之后,来自加拿大的机器学习研究人员们利用名为‘卷积神经网络’这一颇具传统特色的模型获得了 2012 年 ImageNet 挑战赛的冠军。

“我知道,很多人都把 ImageNet 挑战赛视为开启深度学习新时代的里程碑式事件。”

3.3　从数学的角度看卷积神经网络

通过 3.2 节的学习,我们对卷积神经网络的结构和工作机理有了定性的了解;本节从数学角度对卷积神经网络的实现过程、参数确定方法进行详细的探讨。

本节的重点内容主要包括:

- 卷积神经网络哪些参数需要训练确定;
- 采用误差反向传播法确定卷积神经网络参数的原理及步骤。

3.3.1　本书中采用的符号及含义

本书所涉及的符号及含义如表 3.3.1 所示。由于在对神经网络的研究过程中涉及的符号较多,很容易混淆,请对照表 3.3.1 理解、记忆。

表 3.3.1　符号及含义

位　置	符　号	含　义
输入层	x_{ij}	神经元中输入的图像像素(i 行 j 列)的值
卷积核	w_{ij}^{Fk}	第 k 个卷积核的 i 行 j 列的值
卷积层	z_{ij}^{Fk}	卷积层第 k 个子层的 i 行 j 列的加权输入
	b^{Fk}	卷积层第 k 个子层的 i 行 j 列的神经元的偏置
	a_{ij}^{Fk}	卷积层第 k 个子层的 i 行 j 列的神经元的输出

续表

位　置	符　号	含　义
池化层	z_{ij}^{Pk}	池化层第 k 个子层的 i 行 j 列的神经元的输入
	a_{ij}^{Pk}	池化层第 k 个子层的 i 行 j 列的神经元的输出
输出层	w_{k-ij}^{On}	从池化层第 k 个子层的 i 行 j 列的神经元指向输出层第 n 个神经元的箭头的权重
	z_n^O	输出层第 n 个神经元的加权输入
	b_n^O	输出层第 n 个神经元的偏置
	a_n^O	输出层第 n 个神经元的输出

3.3.2　从数学角度看卷积神经网络的工作过程

例 3.3.1　构建一个卷积神经网络，用于识别输入的二值数字图像。

针对 3.3.1 的需求，设计具有一个卷积层、一个池化层、一个输出层的卷积神经网络，如图 3.3.1 所示，每个卷积层有 3 个卷积核，输出层有 3 个神经元。

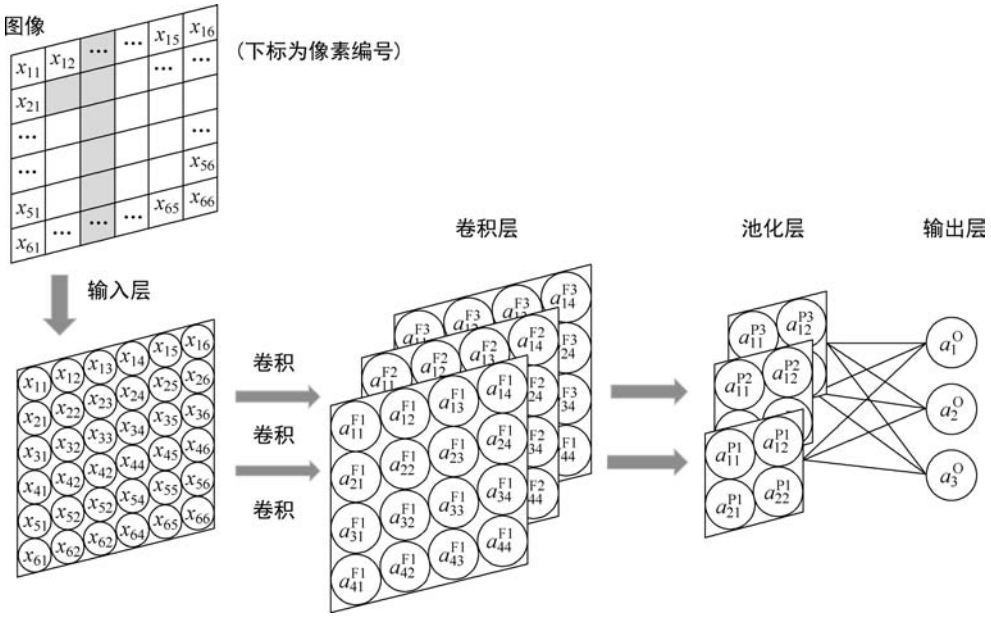

图 3.3.1　用于识别二值数字图像的卷积神经网络

将如图 3.3.1 所示的卷积神经网络用表 3.3.1 中的符号进行表示，如图 3.3.2 所示。

下面对这个网络的详细工作过程进行分析。

输入层：如图 3.3.3 所示，输入数据是 6×6 像素的图像，这些像素值是直接输入到输入层的神经元中的，用 x_{ij} 表示所输入的图像的 i 行 j 列位置的像素数据。

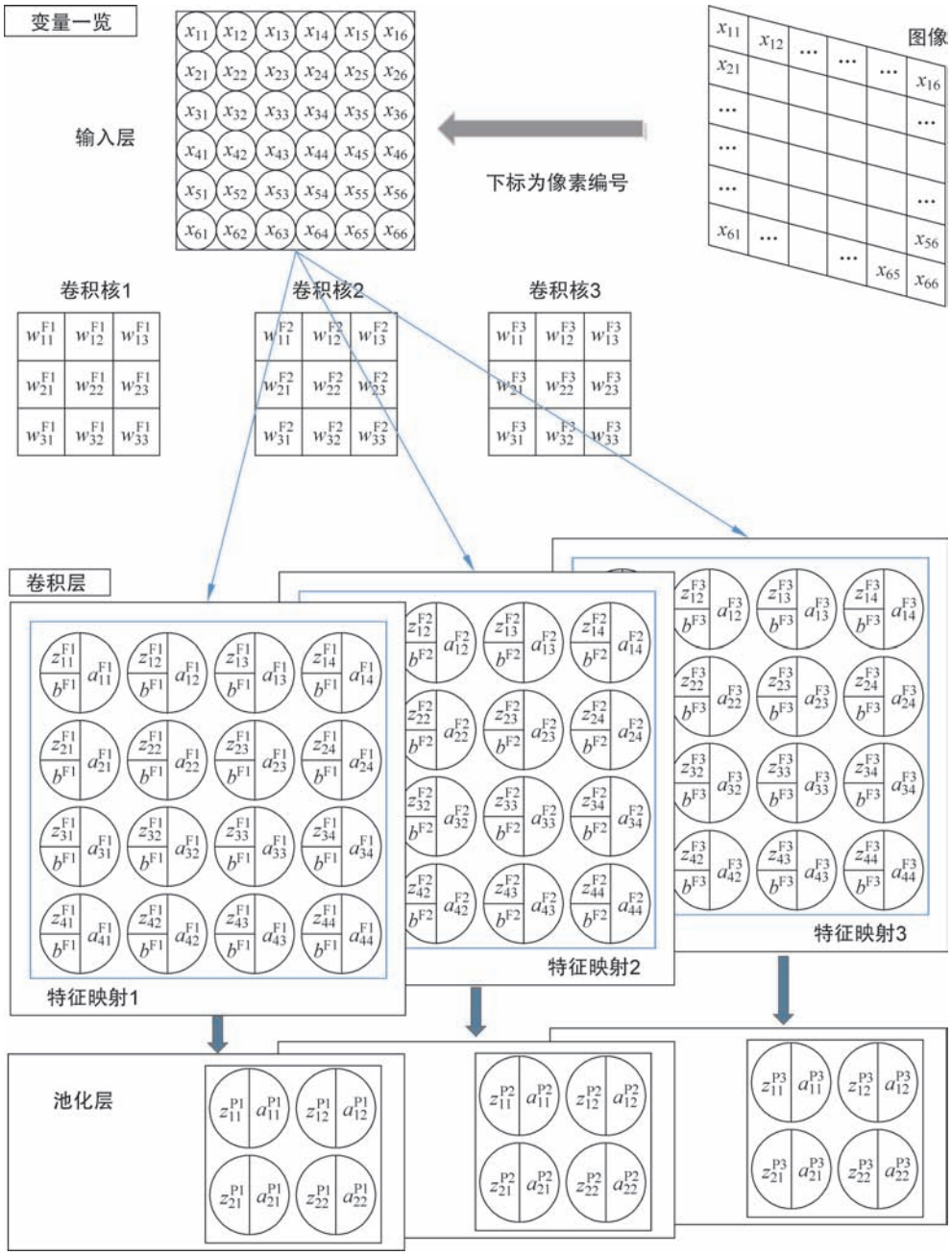

图 3.3.2　用符号详细表示图 3.3.1 所示的神经网络

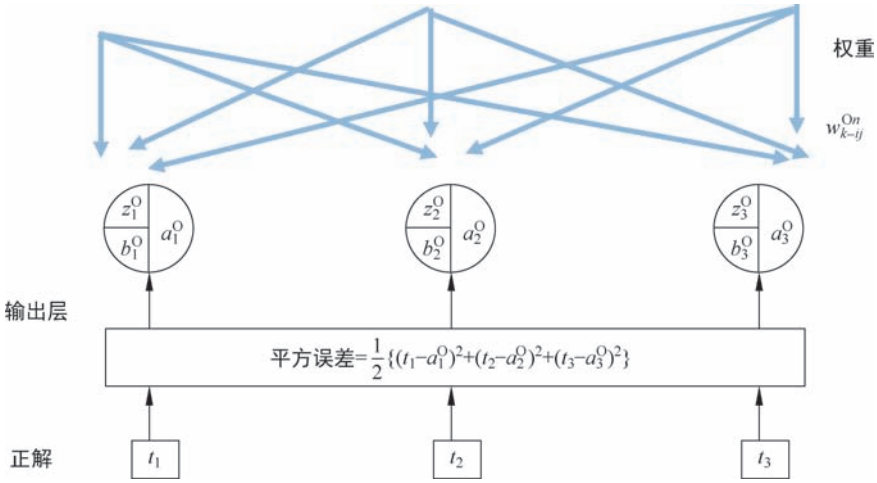

图 3.3.2　（续）

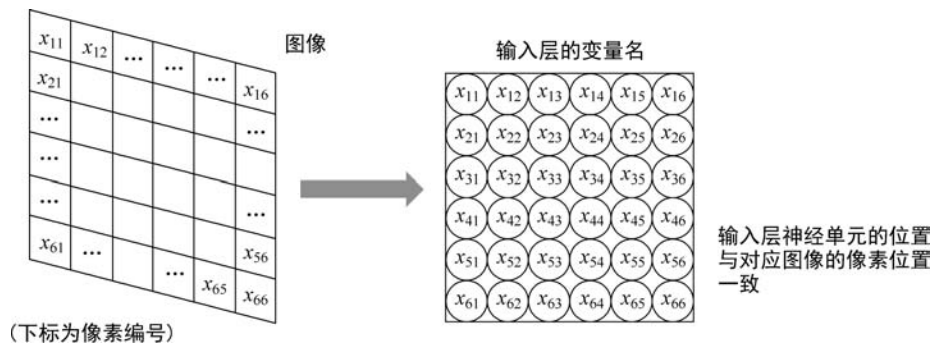

图 3.3.3　用符号表示输入层

在输入层的神经元中，输入值和输出值相同。如果将输入层 i 行 j 列的神经元的输出表示为 a^{I}_{ij}（a 的上标 I 为 Input 的首字母），那么以下关系式成立：

$$a^{\mathrm{I}}_{ij}=x_{ij}$$

卷积层：如图 3.3.4 所示，有 3 个卷积核，由于每个卷积核中元素的数值是通过训练而确定的，所以它们是模型的参数，表示为 w^{Fk}_{11}，w^{Fk}_{12}，…（$k=1,2,3$）。

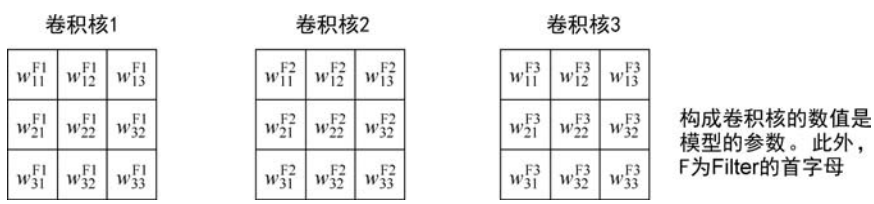

图 3.3.4　用符号表示卷积核

采用卷积核对输入的图像进行卷积运算,如图 3.3.5 所示。

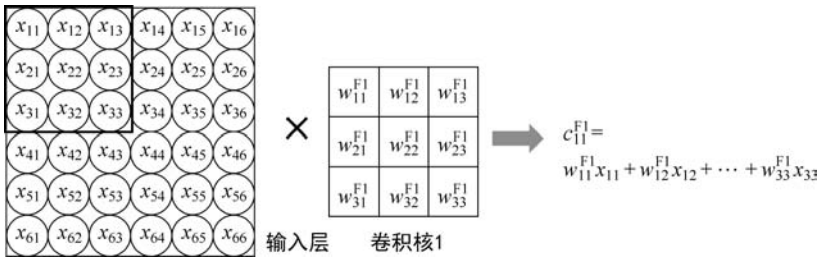

图 3.3.5　采用卷积核进行卷积运算的过程

依次滑动卷积核,用同样的方式计算求得卷积值 c_{12}^{F1},c_{13}^{F1},…,c_{44}^{F1},这样就得到了使用卷积核 1 的卷积的结果。

使用卷积核 k 的卷积结果可用式(3.3.1)表示。

$$c_{ij}^{Fk}=w_{11}^{Fk}x_{ij}+w_{12}^{Fk}x_{i(j+1)}+w_{13}^{Fk}x_{i(j+2)}+\cdots+w_{33}^{Fk}x_{(i+2)(j+2)} \tag{3.3.1}$$

再给卷积后的数值加上一个偏置 b^{Fk},如式(3.1.2)所示,每个卷积核对应同一个偏置,如图 3.3.6 所示。

$$\begin{aligned} z_{ij}^{Fk}=&w_{11}^{Fk}x_{ij}+w_{12}^{Fk}x_{i(j+1)}+w_{13}^{Fk}x_{i(j+2)}+\cdots+\\ &w_{33}^{Fk}x_{(i+2)(j+2)}+b^{Fk} \end{aligned} \tag{3.3.2}$$

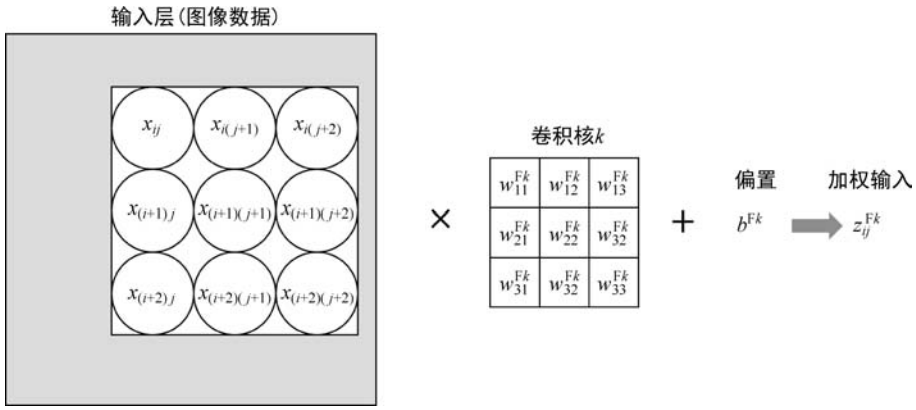

图 3.3.6　每个卷积核对应同一个偏置

若卷积层的激活函数为 $a(z)$,对于加权输入 z_{ij}^{Fk},神经元的输出 a_{ij}^{Fk} 为

$$a_{ij}^{Fk}=a(z_{ij}^{Fk}) \tag{3.3.3}$$

卷积层每个神经元的输出如图 3.3.7 所示。

池化层:最大池化层的工作原理如图 3.3.8 所示,其数学表达式为式(3.1.4)。

$$\begin{cases} z_{ij}^{Pk}=\mathrm{Max}(a_{(2i-1)(2j-1)}^{Pk},a_{(2i-1)(2j)}^{Pk},a_{(2i)(2j-1)}^{Pk},a_{(2i)(2j)}^{Pk}) \\ a_{ij}^{Pk}=z_{ij}^{Pk} \end{cases} \tag{3.3.4}$$

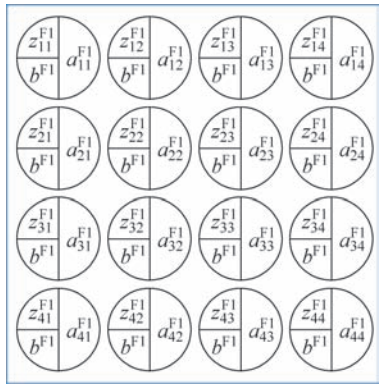

图 3.3.7　卷积层每个神经元的输出示意图

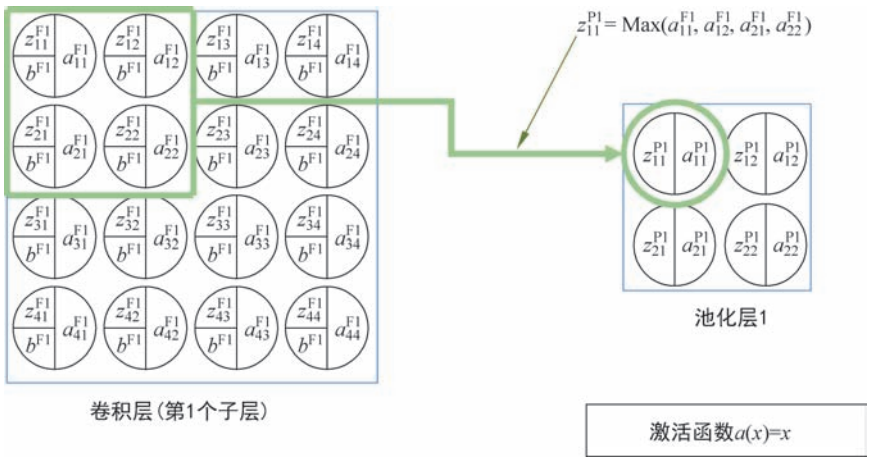

图 3.3.8　最大池化层原理示意图

池化层的神经元没有权重和偏置的概念。激活函数可以认为是 $a(x)=x$，例如，$a_{11}^{P1}=z_{11}^{P1}$。

输出层：输出层有 3 个神经元。如图 3.3.9 所示。输出层第 n 个神经元($n=1,2,3$)的加权输入可以用式(3.3.5)表示，其中，w_{k-ij}^{On} 为输出层第 n 个神经元给池化层神经元的输出 a_{ij}^{Pk}($k=1,2,3$；$i=1,2$；$j=1,2$)分配的权重，b_n^{O} 为输出层第 n 个神经元的偏置。

$$z_n^{O}=w_{1-11}^{On}a_{11}^{P1}+w_{1-12}^{On}a_{12}^{P1}+\cdots+w_{2-11}^{On}a_{11}^{P2}+w_{2-12}^{On}a_{12}^{P2}+\cdots+\cdots+w_{3-11}^{On}a_{11}^{P3}+\cdots+w_{3-12}^{On}a_{12}^{P3}+\cdots+b_n^{O} \tag{3.3.5}$$

输出层第 n 个神经元的输出值为 a_n^{O}，激活函数为 $a(z)$，则

$$a_n^{O}=a(z_n^{O})$$

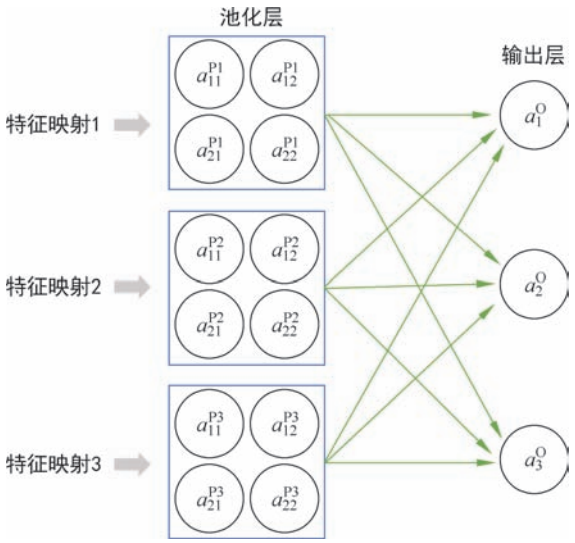

图 3.3.9　池化层与输出层的连接示意图

3.3.3　如何求代价函数

对于如图 3.3.2 所示的卷积神经网络中，输出层神经元的 3 个输出为 a_1^O、a_2^O、a_3^O，对应的学习数据的正解分别记为 t_1、t_2、t_3，平方误差 C 可以用式(3.3.6)表示。平方误差计算示意图如图 3.3.10 所示。

$$C=\frac{1}{2}\{(t_1-a_1^O)^2+(t_2-a_2^O)^2+(t_3-a_3^O)^2\} \tag{3.3.6}$$

注：系数$\frac{1}{2}$是为了后续进行导数计算方便。

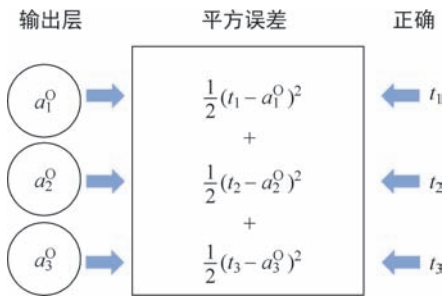

图 3.3.10　平方误差计算示意图

将输入第 k 个训练样本图像时的平方误差的值记为 C_k，如式(3.3.7)所示。

$$C_k=\frac{1}{2}\{(t_1[k]-a_1^O[k])^2+(t_2[k]-a_2^O[k])^2+(t_3[k]-a_3^O[k])^2\} \tag{3.3.7}$$

全体训练样本的平方误差的总和就是代价函数 C_T，如式(3.3.8)所示。

$$C_T = C_1 + C_2 + \cdots + C_{1000} \tag{3.3.8}$$

注意：1000 为训练样本的数量。

对卷积神经网络进行训练的目标就是求出使代价函数 C_T 达到最小的参数。

3.3.4　采用误差反向传播法确定卷积神经网络的参数

在确定卷积神经网络的参数时，梯度下降法也是基础。梯度的方向是函数上升最快的方向，要使代价函数下降最快，应要沿着梯度的反方向逐步下降，这就是梯度下降法的核心思想。

以 C_T 为代价函数，梯度下降法的数学表示如式(3.3.9)所示。

$$(\Delta w_{11}^{\mathrm{Fl}},\cdots,\Delta w_{1-11}^{\mathrm{Ol}},\cdots,\Delta b_1^2,\cdots,\Delta b_1^{\mathrm{O}},\cdots) = -\eta\left(\frac{\partial C_T}{\partial w_{11}^{\mathrm{Fl}}},\cdots,\frac{\partial C_T}{\partial w_{1-11}^{\mathrm{Ol}}},\cdots,\frac{\partial C_T}{\partial b^{\mathrm{Fl}}},\cdots,\frac{\partial C_T}{\partial b_1^{\mathrm{O}}},\cdots\right) \tag{3.3.9}$$

式(3.3.9)右边的括号中为代价函数 C_T 的梯度，其含义如图 3.3.11 所示。

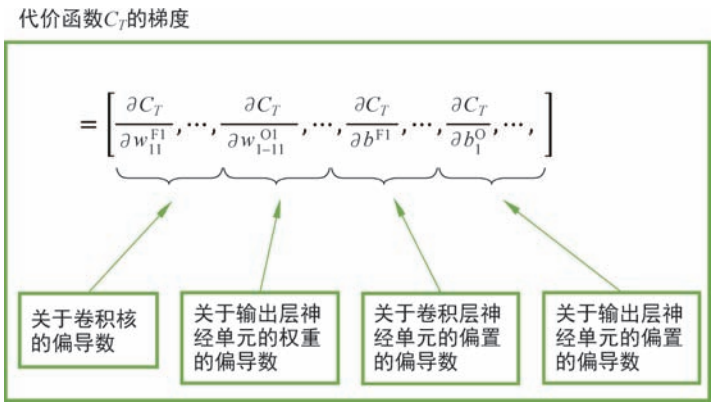

图 3.3.11　代价函数梯度的含义

求代价函数 C_T 的偏导数时，先对式(3.3.6)求偏导数，然后逐一代入样本图像数据，并求和即可。误差反向传播法中引入神经元误差 δ 的概念，如式(3.3.10)所示，$\delta_{ij}^{\mathrm{Fk}}$ 表示卷积层第 k 个子层的第 i 行第 j 列的神经元误差；δ_n^{O} 表示输出层第 n 个神经元的误差。

$$\delta_{ij}^{\mathrm{Fk}} = \frac{\partial C}{\partial z_{ij}^{\mathrm{Fk}}},\quad \delta_n^{\mathrm{O}} = \frac{\partial C}{\partial z_n^{\mathrm{O}}} \tag{3.3.10}$$

卷积层第 1 个子层的第 1 行第 1 列的神经元的误差 $\delta_{11}^{\mathrm{Fl}}$ 以及输出层第 1 个神经元的误差 δ_1^{O} 如图 3.3.12 所示。

根据偏导数的链式求导法则，图 3.3.2 所示的卷积神经网络可以得出：

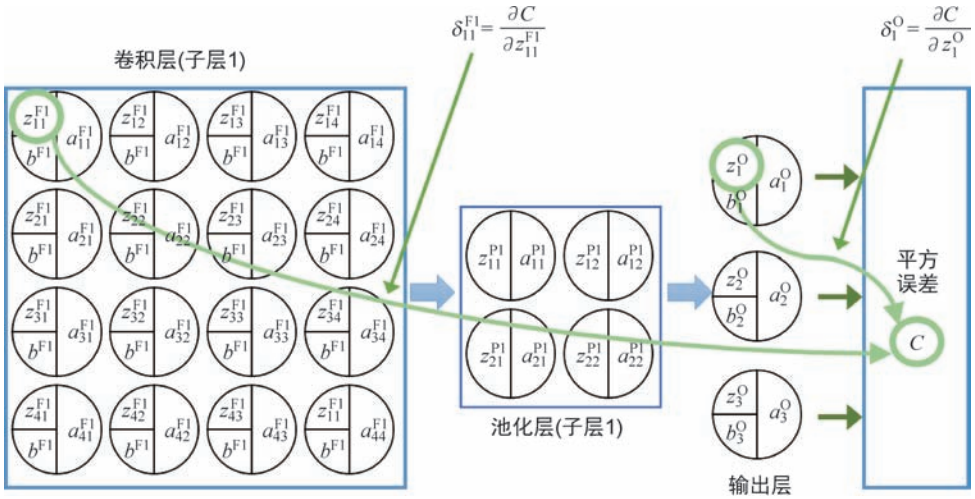

图 3.3.12　神经元误差示意图

$$\begin{cases}\dfrac{\partial C}{\partial w_{k-ij}^{\mathrm{O}n}}=\delta_n^{\mathrm{O}}a_{ij}^{\mathrm{P}k},\dfrac{\partial C}{\partial b_n^{\mathrm{O}}}=\delta_n^{\mathrm{O}}\\ \dfrac{\partial C}{\partial w_{ij}^{\mathrm{F}k}}=\delta_{11}^{\mathrm{F}k}x_{ij}+\delta_{12}^{\mathrm{F}k}+\cdots+\delta_{44}^{\mathrm{F}k}x_{i+3j+3}\\ \dfrac{\partial C}{\partial b^{\mathrm{F}k}}=\delta_{11}^{\mathrm{F}k}+\delta_{12}^{\mathrm{F}k}+\cdots+\delta_{33}^{\mathrm{F}k}+\cdots+\delta_{44}^{\mathrm{F}k}\end{cases}\tag{3.3.11}$$

因此,求出 $\delta_{ij}^{\mathrm{F}k}$ 与 δ_n^{O},便可以得出代价函数 C_T 的梯度。解决思路是先求 δ_n^{O},再根据两者之间的关系便可求出 $\delta_{ij}^{\mathrm{F}k}$,对于图 3.3.2 所示的卷积神经网络,求解 $\delta_{ij}^{\mathrm{F}k}$ 与 δ_n^{O} 的方法如式(3.3.12)所示,其具体的过程本书不进行详细推导,感兴趣的读者可自行进行推导。

$$\begin{cases}\delta_n^{\mathrm{O}}=(a_n^{\mathrm{O}}-t_n)a'(z_n^{\mathrm{O}})\\ \delta_{ij}^{\mathrm{F}k}=\{\delta_1^{\mathrm{O}}w_{k-i'j'}^{\mathrm{O}1}+\delta_2^{\mathrm{O}}w_{k-i'j'}^{\mathrm{O}2}+\delta_3^{\mathrm{O}}w_{k-i'j'}^{\mathrm{O}3}\}\times\\ \qquad(\text{当 }a_{ij}^{\mathrm{F}k}\text{ 在区块中最大时为 1,否则为 0})\times a'(z_{ij}^{\mathrm{F}k})\end{cases}\tag{3.3.12}$$

在以上分析的基础上,可以得出采用误差反向传播法确定卷积神经网络参数的主要步骤如下:

(1) 读入训练样本图像数据;

(2) 设置卷积核参数的初始值、设置网络权重和偏置的初始值、设置学习率等参数的初始值;

(3) 计算出神经元的输出值及损失函数的值;

(4) 根据误差反向传播法计算各层神经元误差;

(5) 根据神经元误差计算损失函数的偏导数;

(6) 根据损失函数的偏导数计算其梯度;

(7) 根据所求出的梯度值,更新网络模型中参数的值;

(8) 反复进行步骤(3)～步骤(7),使损失函数的值最小或降低到某一阈值之内;将此时的参数值作为网络模型中的参数值。

上述步骤的流程图如图 3.3.13 所示。

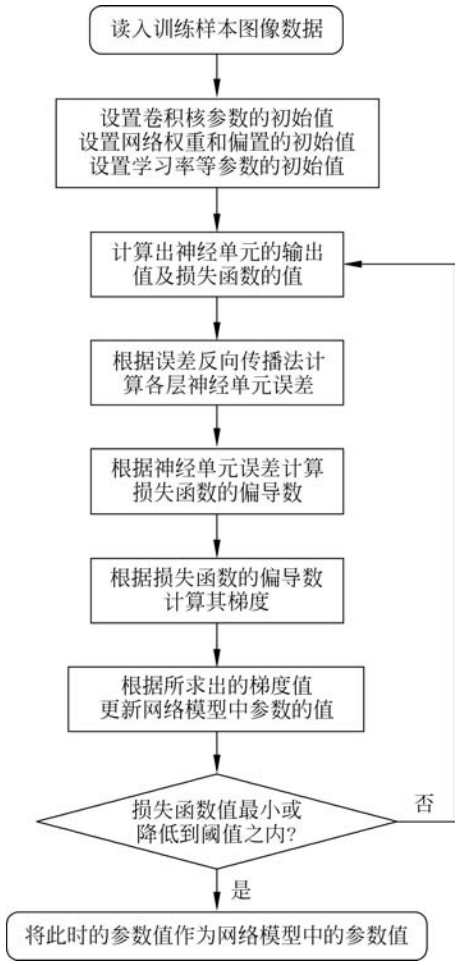

图 3.3.13　采用误差反向传播法确定卷积神经网络参数的流程图

3.4　认识经典的“卷积神经网络”

在前面几节,学习了卷积神经网络的结构和如何调节卷积神经网络的参数,本节将向各位读者展示已经在实际应用中取得良好效果的典型卷积神经网络。

本节主要讲解的问题如下:

- LeNet5、AlexNet、VGG-16 等典型的卷积网络结构的特点;

- 卷积神经网络发展如此迅速的原因。

3.4.1　解析 LeNet5 卷积神经网络

LeNet-5 卷积神经网络出自论文 *Gradient-Based Learning Applied to Document Recognition*,解决的是手写数字识别问题,输入的图像为 28×28 像素的灰度图像,运用的是 MNIST 数据集(对于各类典型数据集的介绍,详见本节的扩展阅读)。LeNet-5 卷积神经网络的结构如图 3.4.1 所示。

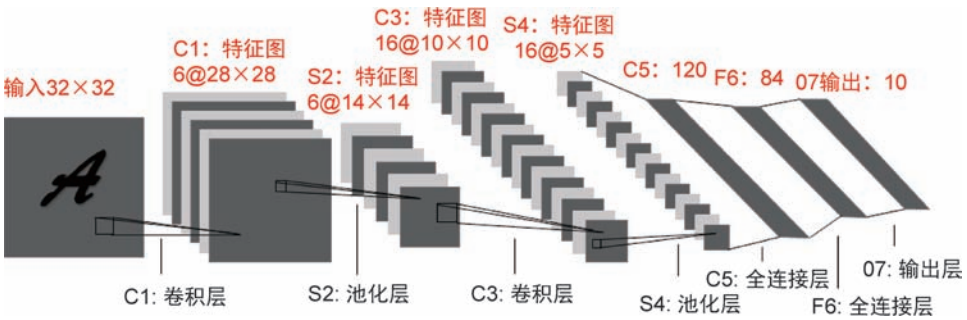

图 3.4.1　LeNet-5 卷积神经网络的结构

下面就对这个网络进行详细分析。

1. C1: 卷积层

- 输入图片大小:32×32 像素。
- 输入图片通道数:1 个。
- 卷积核大小:5×5。
- 卷积核个数:6 个(注:每个卷积核中的参数都不相同)。
- 卷积的步长为 1,卷积的步长为 valid。
- 输出特征图的个数:6 个(注:与卷积核的个数相等)。
- 输出特征图的大小:28×28 像素。
- 本层需要训练的参数:156 个(注:每个卷积核要训练的参数为 5×5 个,再加 1 个公共偏置参数,所以每个卷积核需要训练的参数为 26 个,一共有 6 个卷积核)。

2. S2: 池化层

- 输入特征图:28×28 像素。
- 输入特征图通道数:6 个(注:与上一层输出的特征图的个数相等)。
- 池化的方法:平均池化。
- 每个池化区域大小:2×2 像素。
- 输出特征图的个数:6 个(注:与输入特征图的个数相等)。
- 输出特征图的大小:14×14 像素。

- 本层需要训练的参数：无。

3. C3：卷积层

- 输入特征图大小：14×14 像素。
- 输入特征图通道数：6 个(注：与上一层输出的特征图的个数相等)。
- 卷积核大小：5×5。
- 卷积核个数：16 个(注：每个卷积核中的参数都不相同)。
- 卷积的步长为 1，卷积的步长为 valid。
- 输出特征图的个数：16 个(注：与卷积核的个数相等)。
- 输出特征图的大小：10×10 像素。
- 本层需要训练的参数：1516 个。

在这里，需要强调的是 S2 层产生的 6 个特征图与 C3 层的 16 个卷积核之间的连接关系如图 3.4.2 所示，它们只是部分连接(图 3.4.2 中的 X 表示连接)，并不是全连接，这种连接关系能将连接的数量控制在一个比较合理的范围内。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	X				X	X	X			X	X	X	X		X	X
2	X	X				X	X	X			X	X	X	X		X
3	X	X	X				X	X	X			X		X	X	X
4		X	X	X			X	X	X	X			X		X	X
5			X	X	X			X	X	X	X		X	X		X
6				X	X	X			X	X	X	X		X	X	X

图 3.4.2　S2 层产生的 6 个特征图与 C3 层的 16 个卷积核之间的连接关系

4. S4：池化层

- 输入特征图大小：10×10 像素。
- 池化的方法：平均池化。
- 每个池化区域大小：2×2 像素。
- 输出特征图的个数：16 个(注：与输入特征图的个数相等)。
- 输出特征图的大小：5×5 像素。
- 本层需要训练的参数：无。

5. C5：卷积层

- 输入特征图大小：5×5 像素。
- 输入特征图通道数：16 个(注：与上一层输出的特征图的个数相等)。
- 卷积核大小：5×5。
- 卷积核个数：120 个(注：每个卷积核中的参数都不相同，每个卷积核中有 16 个子卷积核，与输入特征图的通道数相等)。

- 输出特征图的个数：120 个(注：与卷积核的个数相等)。
- 输出特征图的大小：1×1 像素。
- 本层需要训练的参数：48 120 个。

本层需要训练的参数计算过程如下：

由于输入特征图通道有 16 个，故每个卷积核中有 16 个子卷积核(多通道多卷积核的运算过程详见图 3.2.5)，每个子卷积核的大小为 5×5，所以每个卷积核需要确定的参数为 5×5×16+1，其中：1 为卷积核的公共偏置参数；而卷积核的个数为 120 个，故本层需要训练的参数为(5×5×16+1)×120=48 120。

■ 温馨提示

由于 C5 层卷积核的大小与输入的特征图大小相同，故本层也可以看作是全连接层。

6. F6：全连接层

- 输入：120 维向量。
- 节点数：84。
- 非线性激活函数：Sigmoid 函数。
- 本层需要训练的参数：10 164 个(注：84×(120+1)=10 164)。

7. O7：输出层

O7 输出层也是全连接层，共有 10 个节点，分别代表数字 0～9，且如果节点 i 的值为 0，则网络识别的结果是数字 i。采用的是径向基函数(RBF)的网络连接方式。

3.4.2　具有里程碑意义的 AlexNet

AlexNet 是一个引起轰动的卷积深度神经网络，Alex Krizhevsky 用 GPU 训练得到，并以自己的名字命名，在 2012 年的 ImageNet ILSVRC 竞赛中夺冠，在当时的计算机视觉界引起了轰动，也引爆了新一轮的基于深度学习的人工智能研究和应用热潮。

关于 AlexNet 的详细介绍可以参看论文 *ImageNet Classification with Deep Convolutional Neural Networks*[①]。该模型主要由卷积层、池化层(下采样层)和全连接层组成，并引入了一些被后来广泛应用的特性和技巧，比如：使用卷积层和池化层的组合来提取图像的特征；使用 ReLU 作为激活函数；使用 Dropout 抑制过拟合；使用数据扩充(Data Augmentation)抑制过拟合等。

AlexNet 网络主要包含 1 个输入层、1 个输出层、5 个卷积层、3 个下采样层、

① Krizhevsky A，Sutskever I，Hinton G．ImageNet Classification with Deep Convolutional Neural Networks[J]．Advances in neural information processing systems，2012，25(2)．

2 个全连接层。各层的结构和输入输出如表 3.4.1 所示。其中，从输入层到卷积层 1 开始，之后的每一层都被分为 2 个相同的结构进行计算，这是因为 AlexNet 中将计算平均分配到了 2 块 GPU 卡上进行。

表 3.4.1　AlexNet 网络结构及参数

名　称	输　入	卷　积　核	步　长	输　出
输入层	227×227×3	—	—	227×227×3
卷积层 1	227×227×3	3×11×11×48×2	4	55×55×96
池化层 1	55×55×96	—	2	27×27×96
卷积层 2	27×27×96	96×5×5×128×2	1	27×27×256
池化层 2	27×27×256	—	2	13×13×256
卷积层 3	13×13×256	256×3×3×384	1	27×27×128
卷积层 4	13×13×384	3×3×192×2	1	13×13×384
卷积层 5	13×13×384	3×3×192×2	1	13×13×384
池化层 5	13×13×384	—	2	6×6×256
全连接层 6	9216 (6×6×256)	—	—	4096
全连接层 7	4096 (2048×2)	—	—	4096
全连接层 8	4096 (2048×2)	—	—	1000
输出层	1000	—	—	1000

3.4.3　VGG-16 卷积神经网络的结构和参数

VGG 是由牛津大学的 Visual Geometry Group 团队提出，详细内容可以参看论文 *Very Deep Convolutional Networks for Large-Scale Image Recognition*[①]。VGG 继承了 AlexNet 的一些结构，VGG-16 模型深度为 16 层，只使用 3×3 大小的卷积核(极少用了 1×1 卷积核)和 2×2 的池化核，这种小尺寸核有利于减少计算量。VGG 层数更深，特征图更宽，最后 3 个全连接层在形式上完全迁移 AlexNet 的最后 3 层。此外，在测试阶段把网络中原本的 3 个全连接层依次变为 3 个卷积层，所以网络可以处理任意大小的输入。VGG 参数量是 AlexNet 的大约 3 倍。

VGG-16 卷积神经网络的结构和参数如表 3.4.2 所示，随着层数的加深，网络宽高变小，而通道数增大。网络主要包含 1 个输入层、1 个输出层、13 个卷积层、5 个下采样层、3 个全连接层。

---

① Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer ence, 2014.

表 3.4.2　VGG-16 网络结构及参数

名　称	输　入	卷　积　核	步　长	输　出
输入层	224×224×3	—	—	224×224×3
卷积层 1	224×224×3	3×3×3×64	1	224×224×64
卷积层 2	224×224×64	64×3×3×64	—	224×224×64
池化层 2	224×224×64	—	2	112×112×128
卷积层 3	112×112×128	128×3×3×128	1	112×112×128
卷积层 4	112×112×128	128×3×3×128	1	112×112×128
池化层 4	112×112×128	—	2	56×56×256
卷积层 5	56×56×256	128×3×3×256	1	56×56×256
卷积层 6	56×56×256	256×3×3×256	1	56×56×256
卷积层 7	56×56×256	256×3×3×256	1	56×56×256
池化层 7	56×56×256	—	2	28×28×512
卷积层 8	28×28×512	256×3×3×512	1	28×28×512
卷积层 9	28×28×512	512×3×3×512	1	28×28×512
卷积层 10	28×28×512	512×3×3×512	1	28×28×512
池化层 10	28×28×512	—	2	14×14×512
卷积层 11	14×14×512	512×3×3×512	1	14×14×512
卷积层 12	14×14×512	512×3×3×512	1	14×14×512
卷积层 13	14×14×512	512×3×3×512	1	14×14×512
池化层 13	14×14×512	—	2	7×7×512
全连接层 14	4096	—	—	4096
全连接层 15	4096	—	—	4096
全连接层 16	4096	—	—	1000
输出层	1000	—	—	1000

3.4.4　卷积神经网络为何会迅猛发展

深度学习作为一门数据驱动的科学，卷积神经网络本身的性能就和训练数据的总量、多样性有着密不可分的联系。一个“见多识广”的卷积神经网络，对于问题的处理往往更加优秀。如今，由于互联网技术的飞速发展，网络、设备、系统互联互通，产生了大量数据，再加上分布式存储的发展，使数据以指数爆炸性的增长，“大数据”的理念已渗透到社会和生活的方方面面，做一个形象的比喻，数据工程的迅猛发展就像燃料一样，推动着卷积神经网络这枚火箭不断发展。

卷积神经网络的发展与硬件的支持也是分不开的。卷积神经网络的训练过程需要大量的计算资源，而越深层、越复杂的卷积神经网络对硬件资源的需求就越大。这种繁重的计算任务是普通 CPU 难以胜任的，更强大的 GPU(图形处理器)的出现和广泛应用也极大地促进了卷积神经网络的发展。以大家熟知的 AlexNet 为例，为了完成 ImageNet 分类模型的训练，使用一个 16 核的 CPU 需要一个多月

才能训练完成，而使用一块 GPU 则只需两三天，训练效率极大提高；Google 公司研发了人工智能专用芯片 TPU 来进行并行计算，它是为深度学习特定用途特殊设计的逻辑芯片，使得深度学习的训练速度更快。

因此，可以说大数据和高性能硬件是推动卷积神经网络发展的两个重要的助推器。

3.5　思考与练习

1. 典型的卷积神经网络包括哪几部分组成？

2. 在卷积神经网络中，卷积层的作用是什么？

3. 对于卷积神经网络来说，什么是“参数共享”？“参数共享”的意义是什么？

4. 在卷积神经网络中，池化层的作用是什么？常用的池化方法有哪两种？

5. 在卷积神经网络中，哪些参数需要训练来确定？

6. 对卷积神经网络进行训练的目标是什么？

7. 请绘制基于误差反向传播法确定卷积神经网络参数的流程图。

8. 在 LeNet 的第一个卷积层 C1 中，输入图像的大小为 32×32 像素，输入图片通道数为 1 个，卷积核大小为 5×5，卷积的步长为 1，请问本层输出的特征图的个数是多少？本层有多少个参数需要训练？

9. LeNet 的 S4 池化层中输出的特征图的个数为 16 个，在其之后的卷积层 C5 的卷积核个数为 120 个，每个卷积核中有多少个子卷积核？卷积层 C5 输出的特征图的个数是多少？

10. 与 LeNet 相比，AlexNet 采用了哪些方法来抑制过拟合？

11. 在卷积层之后，加入非线性激活函数的目的是什么？

12. 简述多通道卷积的计算过程。

13. 卷积神经网络与全连接神经网络的主要区别是什么？