**神经网络模型**

08

多层（深度）神经网络是机器学习当前最活跃的分支之一，在计算机视觉、语音以及自然语言处理等方面有成功应用。

本章我们主要讨论常用的神经网络结构：全连接前馈神经网络、卷积神经网络、循环神经网络、残差网络。

神经网络模型在历史上很早就已出现，但在大型应用中真正大展威力在2010年之后。现实中问题通常比较复杂，我们需要用复杂的模型来建模，多层（深度）神经网络模型是一个自然的选择。但深度模型训练困难，需要更多训练数据和计算资源。随着大数据时代的到来和计算能力的提高，以及各种优化技术的发展，深度学习在机器学习中的地位更加重要，在视觉、语音以及自然语言处理等非结构化数据分析方面取得了重要进展。

8.1节介绍神经元的基本结构，8.2节到8.5节分别介绍几种常用的神经网络结构：全连接前馈神经网络、卷积神经网络、循环神经网络、残差网络。下一章我们将讨论深度模型的训练技巧。

## **标题2**  8.1 神经元的基本结构

人脑可能有1000多亿个神经元，每个神经元都可能在几个方向上互相连接，这些大量的神经元及其连接形成一个超大型网络。一个典型的神经元结构如图8-1（a）所示。一个神经元通常具有多个**树突**，主要用来接受传入信息；而**轴突**只有一条，轴突尾端有许多轴突突触可以给其他多个神经元传递信息。

|  |  |
| --- | --- |
|  |  |
| （a）典型神经元结构[9] | （b）M-P神经元模型 |

图8-1 神经元和M-P神经元模型

受人类大脑的启发，神经生理学家沃伦·麦克洛克（McCulloch）和数学家沃尔特·皮茨（Pitts）于1943年提出M-P神经元模型（如图8-1（b）所示）。神经元模型模拟大脑神经元的运行过程，其包含输入、输出与计算功能，其中输入可以类比为神经元的树突，而输出可以类比为神经元的轴突，计算则可以类比为细胞核。

神经元的功能为：

|  |  |
| --- | --- |
|  | （8-1） |

其中，每一个输入对应一个权值，神经元对输入与权重做乘法后求和，求和的结果与偏置/激活阈值比较，最终将结果经过激活函数，输出。其中非线性的激活函数使得神经网络具有非线性建模能力。如果没有激活函数，即使多个神经元连接也只能进行线性映射。常用的激活函数有sigmoid函数和整流线性单元（Rectified Linear Unit, ReLU）函数，更多激活函数请见9.3节。

## **标题2**  8.2 前馈全连接神经网络

多个神经元连接构成神经网络，最常见的一种网络结构为前馈全联接神经网络，亦被称为多层感知机（Multi-Layer Perception，MLP）或深度神经网络（Deep Neural Network，DNN）。

一个典型的前馈全联接神经网络的网络结构如图8-2所示。在输入层和输出层之间，可能有一个或多个隐含层。前馈神经网络的信息只向前单向移动，从输入层开始，通过隐藏层，再到输出层，网络中没有循环或回路。

用表示第层第个神经元的输出，向量表示该层所有神经元的输出，表示第层第个神经元到第层第个神经元之间的连接权重，矩阵向量表示该层所有第层神经元到第层神经元之间的连接权重，表示第层第个神经元的偏置，向量表示该层所有神经元的偏置，表示第层第个神经元激活函数的输入，向量表示该层所有神经元激活函数的输入，则相邻两层神经元之间的关系为：

|  |  |
| --- | --- |
|  | （8-2） |

写成向量/矩阵形式为：

|  |  |
| --- | --- |
|  | （8-3） |

为了书写简洁，记。因此神经网络内部实际上就是矩阵计算。

|  |  |
| --- | --- |
|  |  |
| （a）全连接神经网络层间连接 | （b）相邻层结点之间的连接详情 |

图8-2 全连接神经网络

## **标题2**  8.3 卷积神经网络

卷积神经网络（Convolutional Neural Network，CNN）也是一种前馈神经网络，至少包含一个卷积层。卷积层的输入和输出之间不是全连接，输出结点只与一部分输入结点连接，并且不同位置的输出结点与输入结点的连接权重相等（连接权值共享）。卷积神经网络经常用来处理具有网格结构的数据（如时间序列数据或图像数据），最近研究者们也开始研究对非网格的图结构数据进行卷积的图卷积神经网络。

一个典型的CNN通常由三部分构成：

• 卷积层：提取信号中的局部特征；

• 池化层：降低分辨率/扩大感受野，同时大幅降低参数量（降维）；

• 全连接层：同传统全连接神经网络，用于输出结果。

### **8.3.1** 卷积层

卷积是一种特殊的线性运算，是信号处理中的常用操作，用于提取信号中的局部特征。在信号处理中，为了使得卷积操作满足可交换性，通常对卷积核进行翻转，即输入与卷积核进行一维离散卷积的结果为：

|  |  |
| --- | --- |
|  | （8-4） |

在卷积神经网络中，卷积核是通过学习得到的，因此卷积核是否翻转不重要。为了书写和计算简单，我们不翻转卷积核，这样得到的其实是互相关函数（cross-correlation）：

|  |  |
| --- | --- |
|  | （8-5） |

许多机器学习库实现的是互相关函数，但称之为卷积。在神经网络中，我们对卷积的输出通过激活函数再施加一个非线性变换：

|  |  |
| --- | --- |
|  | （8-6） |

其中为激活函数，为卷积核的权重，为卷积的偏置项。

如果输入是二维图像，卷积核也为二维，得到二维卷积：

|  |  |
| --- | --- |
|  | （8-7） |

从网络连接的角度，在全连接神经网络中，第层的神经元与每一个层的神经元相连，连接权重为。而卷积神经网络中卷积核的大小通常小于输入元素的维度，因此卷积层的神经元与前一层神经元之间是局部连接而非全连接。如图8-3所示，卷积层输出神经元1只与输入层神经元1、2、3等3个神经元相连，与4、5不相连，我们称该卷积的感受野（Receptive Field）大小为3。感受野原指听觉、视觉等神经系统中一些神经元的特性，即神经元只接受其所支配的刺激区域内的信号，这里表示输出神经元只受感受野内输入的影响。

卷积层的另一个特点是权值共享。如在图8-3中，输出神经元1和3的权重相同、2和4的权重相同。我们可以理解为输出神经元1和3提取低层输入中相邻位置的相同特征（如水平边缘），而神经元2和4提取的是图像的另一种特征（如竖直边缘）。我们称神经元1和3为一个滤波器（Filter）、一个卷积核（Kernel）或一个通道（Channel）。图8-3中共有两个滤波器。由于输出神经元1的输入为1、2、3，输出神经元3的输入为3、4、5，这样两个相邻神经元的输入有2个单位位置的差异，我们称该卷积的步幅（Stride）为2。

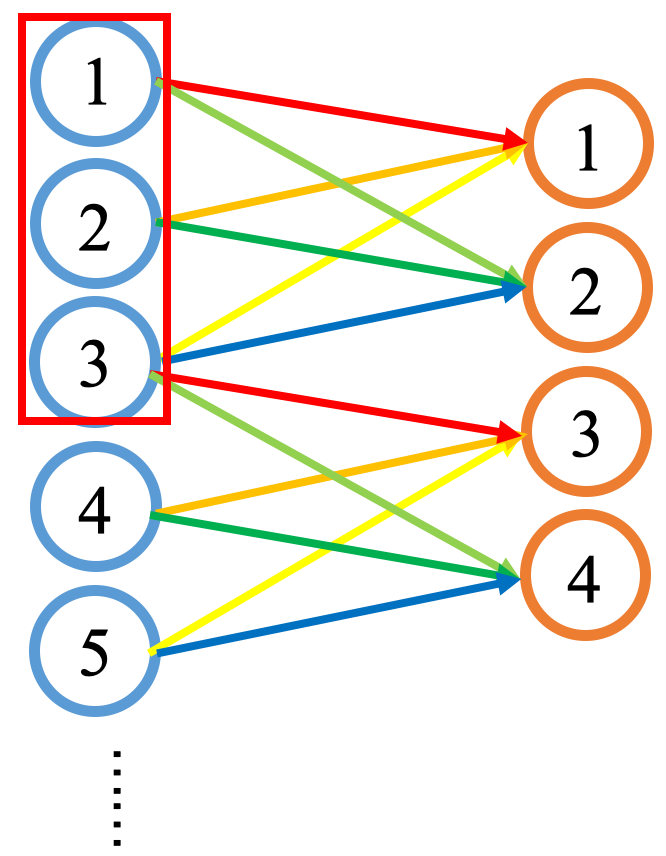


图8-3 卷积层的局部连接和权值共享，相同颜色的连接表示连接权值相等

和全连接相比，卷积层通过采用局部连接和权重共享使得参数大大减少，从而使得模型更简单，预防模型过拟合。如5个输入结点和4个输出结点之间的全联接参数数目为，而如图8-3所示的卷积层需要的参数数目为，其中是偏置项。当输入层神经元数目更多时，全连接神经网络的参数更多，而卷积神经网络的参数不变。

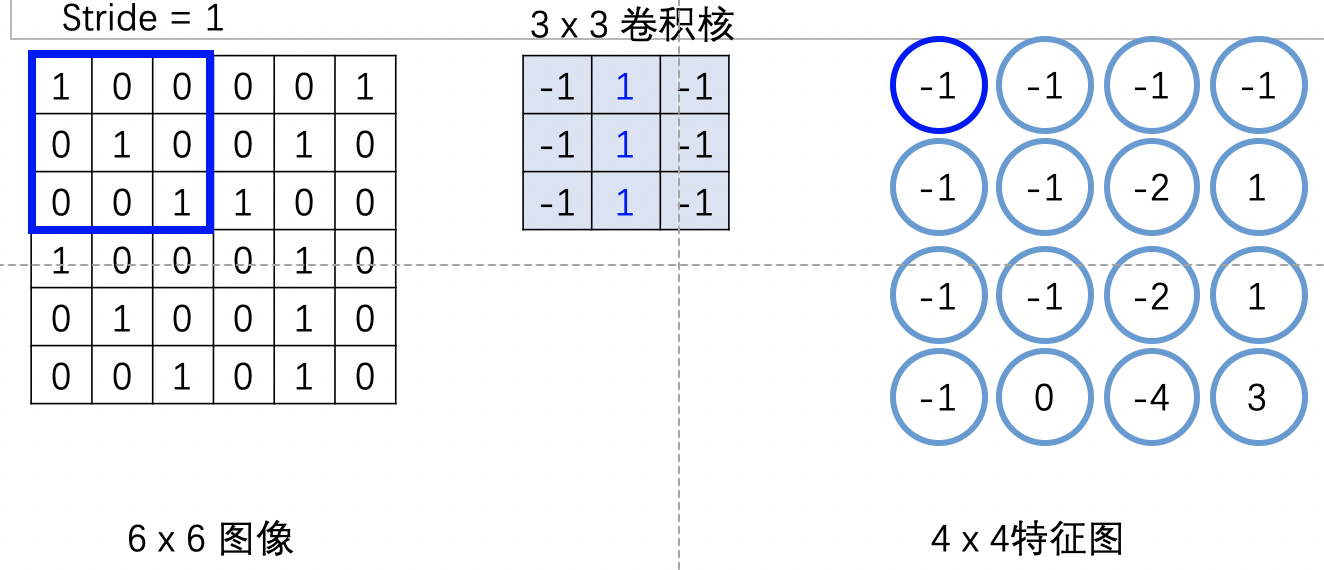
一个二维图像的卷积操作如图8-4所示。为了考虑图像中像素间的空间关系，卷积核也是二维，如。给定一个的图像，经过步幅为1的卷积核卷积后，得到大小为的图像，输出图像的大小比输入变小了。在上述卷积中，中间的像素点有可能参与多次计算，但是边界像素点可能只参与一次。所以结果可能会丢失边界信息。那么为了弥补这个问题，我们引入边界填充（Padding），即通过在图片外围补充一些像素点，扩充输入图像。这些扩充的像素点通常初始化为0。如图8-4（b）所示，虚线部分为填充区域。填充区域越大，输出特征图越大。

另外也可以通过增大步幅缩小输出特征图大小。如图 8-4（c）中，我们将步幅设置为 2，则输出特征图大小为。

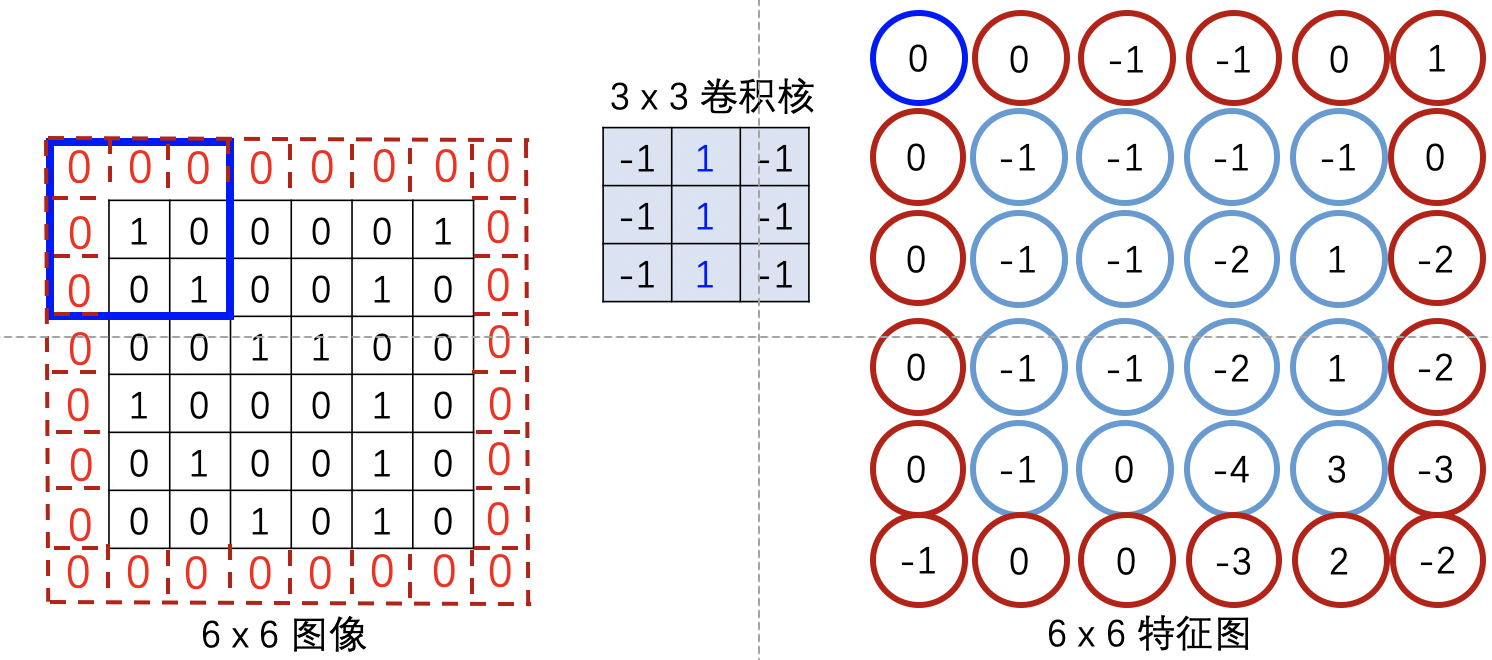
输出特征图宽度与输入图像宽度、卷积核宽度、单边填充区域宽度、以及步幅之间的关系为：

|  |  |
| --- | --- |
|  | （8-8） |

输出特征图高度计算方式类似。

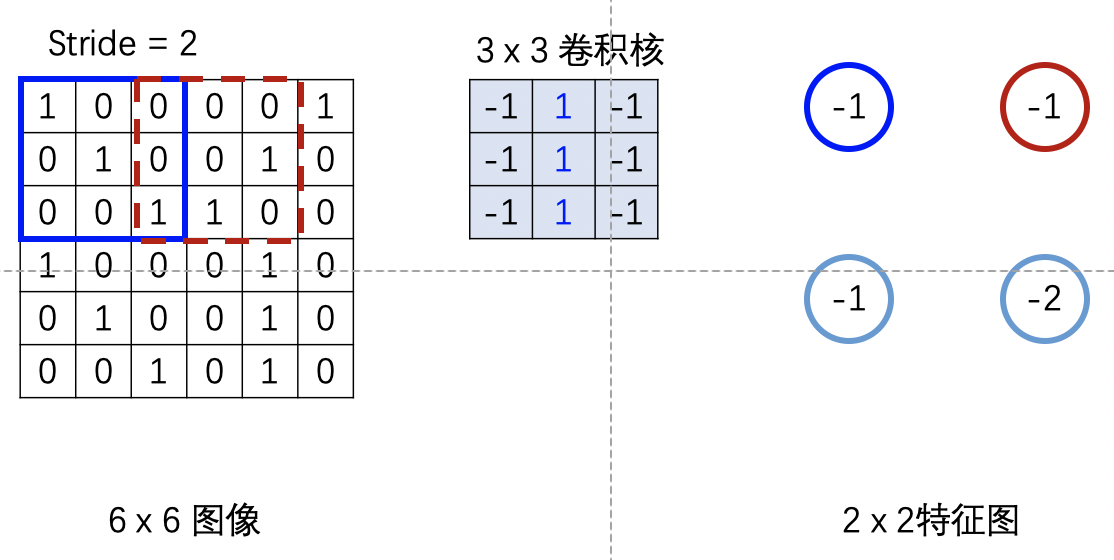


（a）步幅为1的二维卷积



（b）边界填充为1的二维卷积

图8-4 二维卷积



（c）步幅为2的二维卷积

图8-4 二维卷积（续）

### **8.3.2** 池化层

池化（Pooling）是卷积神经网络中的一个重要的概念。池化实际上是一种降采样，通常采用最大池化（Max pooling）或平均池化（Mean Pooling）。池化过程如图8-5所示，将输入信号划分为若干个矩形区域，对每个子区域输出最大值（最大池化）或平均值（平均池化）。

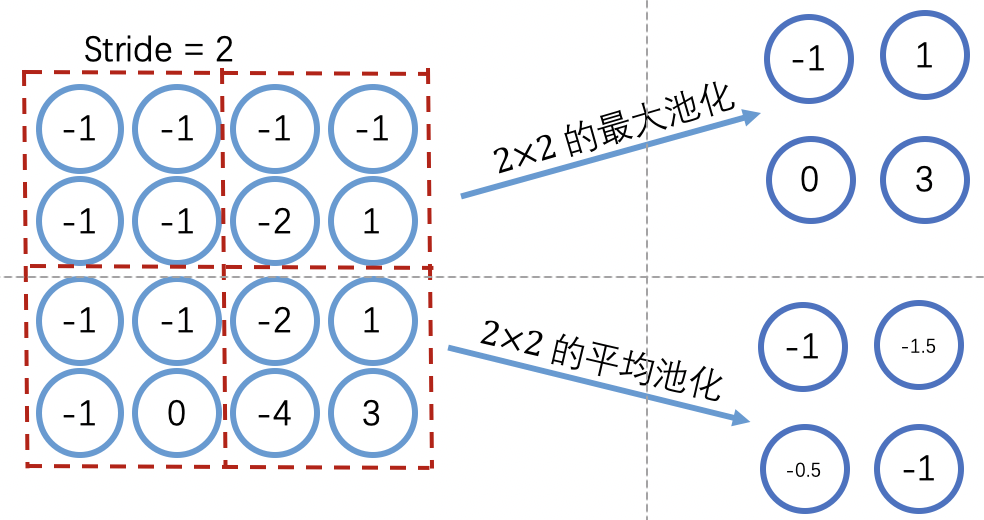


图8-5 池化

池化的用途之一是使得特征有一定的平移不变性，即特征的精确位置不重要。池化层的另一个作用是迅速减少特征维度，通常我们采用步幅为2且大小为的池化，特征的维度经过一轮池化后变成原来的。这样模型的参数数量和计算量下降，也在一定程度上也控制了过拟合。

池化层一般没有参数。过去人们广泛使用平均池化，由于最大池化在实践中表现更好，现在人们更多地使用最大池化。由于池化层过快地减少了数据的维度，因此通常CNN不需要每个卷积层后都接入池化层，可以在多个卷积之间周期性地插入池化层，甚至不再使用池化层，而是通过其他方式适当减少数据大小（如在卷积中增大步幅）。

### **8.3.3** **CNN**示例——**AlexNet**

我们以2012年Imagenet比赛冠军的模AlexNet[10]为例讲解卷积神经网络的组成。AlexNet验证了CNN在复杂任务上的有效性，同时GPU实现使得训练在可接受的时间范围内可以得到结果，使得CNN和GPU都大火了一把。

AlexNet网络结构如8-6所示，用于对ImageNet图像数据集的1000类图像进行分类。AlexNet共有8层（不包括输入层），前面5层是卷积层，后面3层是全连接层，最后一个全连接层的输出传递给一个1000路的Softmax层，对应1000个类标签的分布。由于AlexNet采用两个GPU进行训练，因此该网络结构图由上下两部分组成，一个GPU运行图上方的层，另一个运行图下方的层，两个GPU只在特定层（第二个卷积层、第5个卷积层、全连接层）通信。例如第二、四、五层卷积层的核只和同一个GPU上的前一层的核特征图相连，第三层卷积层和第二层所有的核特征图相连接，全连接层中的神经元和前一层中的所有神经元相连接。

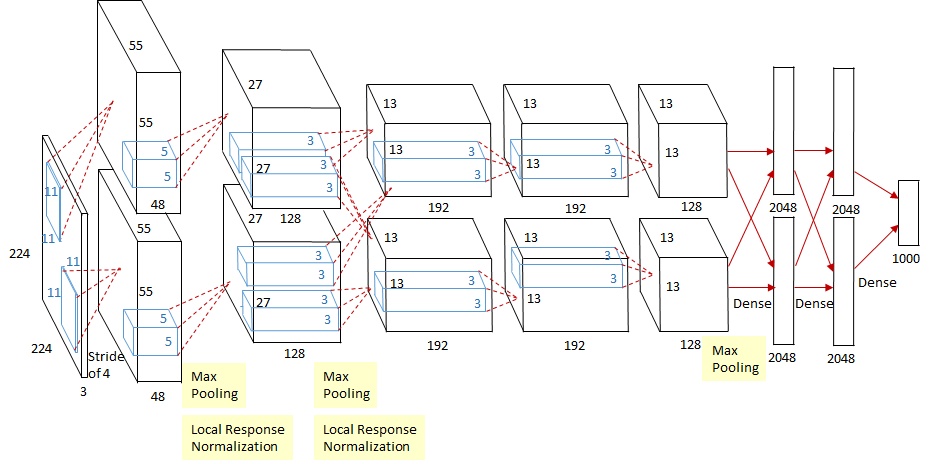


图8-6 AlexNet网络结构[10]

第一个卷积层包含四个步骤：卷积→ReLU→池化→归一化。

（1）卷积

输入的原始图像大小为（RGB图像，注意图8-5中写的是），在本层使用96个的卷积核进行卷积计算，生成新的特征。由于采用了两个GPU并行运算，因此，网络结构图中上下两部分分别承担了48个卷积核的运算。AlexNet中本层的卷积移动步幅，因此卷积后生成的特征图大小为：

|  |
| --- |
|  |

即。

（2）激活层：ReLU

卷积后的特征经过激活函数，输出仍然是特征。AlexNet采用的激活函数为ReLU。

（3）池化

CNN中的池化一般是不重叠的，但AlexNet中的池化为重叠池化，目的是使得信号的维度缩减不致于太过。重叠池化和卷积操作类似，可以定义步幅参数，和卷积的不同在于：卷积操作将窗口元素和卷积核求内积，而池化操作求最大值/平均值等，窗口的滑动等原理完全相同。

AlexNet中池化运算的尺寸为，步幅为2，池化后图像的尺寸为。AlexNet中的池化采用的是最大池化。

（4）归一化

池化后的特征层再进行归一化处理，归一化后的特征大小不变。AlexNet中归一化采用的是局部归一化（Local Response Normalization，LRN），借鉴了生物学上“侧抑制”的做法，利用邻域数据做归一化，使得归一化后响应比较大的值变得相对更大，并抑制其他反馈较小的神经元。因为前一个步骤使用ReLU，ReLU的响应结果是无界的（可能非常大），所以需要归一化。

LRN的处理公式为：

|  |  |
| --- | --- |
|  | （8-9） |

其中分别表示归一化的输入和输出，是一个三维数组，分别表示输入信号的高度、宽度和通道数，表示输入的空间位置索引，为通道索引。注意求和（）是着通道方向进行，表示总的通道数，和分别表示求和的通道半径和偏置。论文中。

不过后来的CNN采用批量归一化（Batch Normalization, BN）取代LRN。BN详细描述请见9.4节。

第二个卷积层和第一个卷积层类似，也包含四个步骤：卷积→ReLU→池化→归一化。

第三个卷积层和第四个卷积层均只包含两个步骤：卷积→ReLU。

第五个卷积层包含三个步骤：卷积→ReLU→池化。

第六个层为全连接层，包含三个步骤：fc→ReLU→Dropout。

fc表示全连接（Fully Connected）。Dropout层在Alexnet中首次被提出，随机丢弃 结点，即在训练时，随机挑选的神经元，使得其输出为0，反向传播更新参数时也不更新这些结点。关于Dropout我们在8.6节再详细讨论。

第七个层为全连接层，也包含三个步骤：fc→ReLU→Dropout。

第八个层为全连接层，也是最后的输出层，包含三个步骤：fc→Softmax。Softmax完成图像分类功能。

## **标题2**  8.4 循环神经网络

在一般的机器学习任务中，我们假设多个输入（和多个输出）是独立，但该假设对序列信号不成立。如一句话中前后单词是有关系的，如果我们要预测下一个单词，最好是能知道前面都已经有过哪些单词。循环神经网络（Recurrent Neural Network，RNN）通过增加一个存储单元，存储序列中信息，从而达到利用序列信息的目的。

### **8.4.1** 简单循环神经网络

一个典型的RNN结构如图 8-7 所示。循环神经网络存储先前时刻输入对应的隐含状态（），并与当前输入（）结合，从而保持当前输入与先前输入的某些关系：

|  |  |
| --- | --- |
| ，  ， | （8-10） |

其中通常为激活函数，通常为S函数或S函数，是输入层到隐藏层的权重矩阵，为输出层的值，是隐藏层到输出层的权重矩阵，是相邻隐藏层之间的权重矩阵。

循环神经网络之所以称为“循环”，是因为一个序列当前的输出与前面的输出有关。我们对时刻的输出进行展开（为了表达简洁，这里忽略了偏置项）：

|  |
| --- |
|  |

从上面可以看出，循环神经网络的输出值受前面历次输入值的影响。

按时间步展开后，我们会发现RNN在所有时间步循环神经元的权重相同，不同时刻只是当前时刻的输入不同、以及前面已经有过的输入不同。注意图8-7中我们用一个长方形表示网络的一层，表示一个向量，而不是在8.2节中用一个神经元表示一个标量。如果将图中的向量展开，得到的表示如图8-8所示。

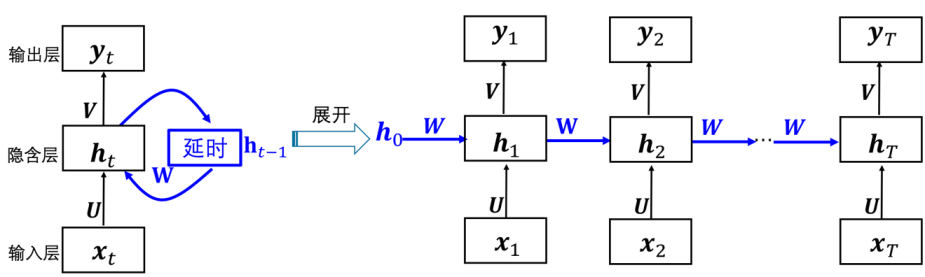


图8-7 循环神经网络

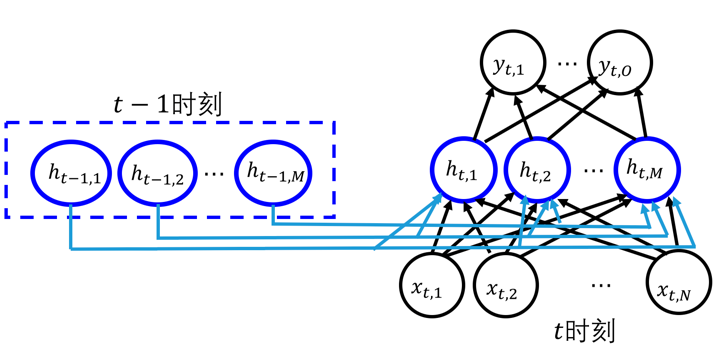


图8-8 按神经元展开的RNN结构

RNN的结构非常灵活，可以是如图8-7所示的多个输入多个输出，如在机器翻译中为序列输入，序列输出。RNN还可以是序列输入，单个输出。如我们根据一个句子判断该语句表达的感情（图8-9（a））；或者是单个输入，序列输出，如根据一幅图像产生图像的语言描述（图8-9（b））。

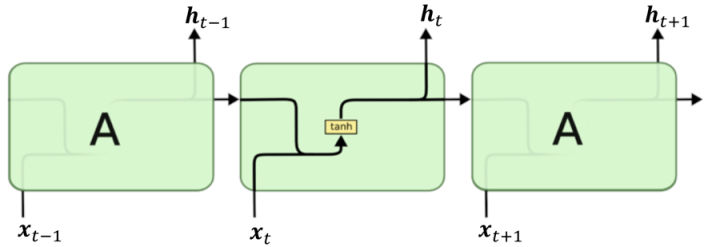
|  |  |
| --- | --- |
|  |  |
| （a）序列输入，单值输出 | （b）单值输入，序列输出 |

图8-9 各种形式的RNN

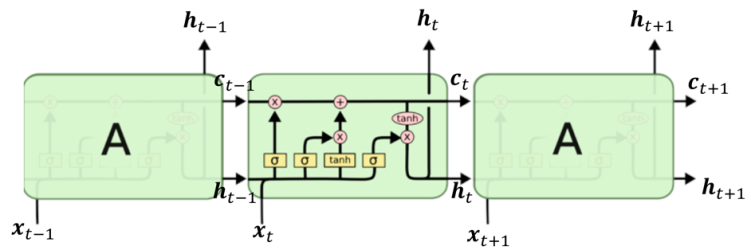
### **8.4.2** 长短时记忆网络

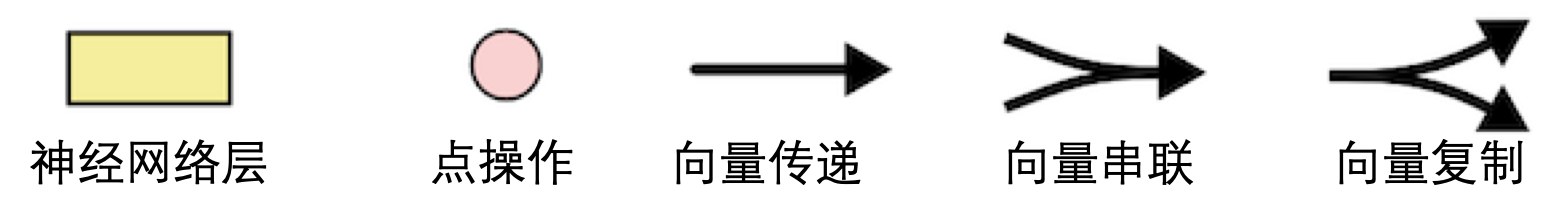
由于RNN在反向传播进行参数更新时，会产生梯度消失或梯度爆炸问题（详见9.1节）。长短时记忆网络（Long Short-Term Memory, LSTM）就是用来解决RNN中梯度消失问题的，从而可以处理长序列。LSTM在各种各样的问题上都工作得很好，被广泛使用。

LSTM的结构如图8-10所示，增加3个**控制门**来记忆长的短时信息：输入门、遗忘门和输出门。门是一种可选择地让信息通过的方式，由一个S神经网络层和点乘运算组成。LSTM 单元一般会输出两种状态到下一个单元，即单元状态和隐藏状态。



（a）简单RNN结构，只包含单一层





（b）LSTM结构，包含四个交互的层

图8-10 LSTM结构

• 遗忘门层（Forget Gate Layer）

LSTM的第一步就是决定我们要从单元状态中丢弃什么信息。这个决定由一个层做出，称为遗忘门层。遗忘门层有和两个输入，其中为前一个单元的隐藏状态，为当前时间步的输入。遗忘门层的计算过程为：

|  |  |
| --- | --- |
| 。 | （8-11） |

函数将会输出一个向量，每个元素取值的范围为0到1，对应单元状态中的每个数值。若单元状态的值为0，则遗忘门就要求单元状态完全忘记该信息，1表示完全保持。

• 输入门层（Input Gate Layer）

下一步是决定要在单元状态下存储的新信息。首先，输入门层的层决定我们将更新哪些值：

|  |  |
| --- | --- |
| 。 | （8-12） |

然后，一个层创建一个可以被添加到状态中的新候选值向量：

|  |  |
| --- | --- |
| 。 | （8-13） |

接下来结合这两个门来创建对单元状态进行更新，将旧的单元状态更新为新的单元状态：

|  |  |
| --- | --- |
|  | （8-14） |

我们将旧的单元状态乘以，去忘记那些我们之前决定忘记的东西；然后我们向状态中添加。这是新的候选值，并且按照我们决定更新每个状态值的程度来缩放。

• 输出门层（Output Gate Layer）

最后我们确定要输出的信息。输出将基于单元状态，但是是一个过滤后的版本。首先我们运行一个层，它决定了我们要输出单元状态的哪些部分：

|  |  |
| --- | --- |
| 。 | （8-15） |

然后我们让单元状态通过（将值变为−1和1之间），并将其乘以门的输出，以便我们只输出我们决定输出的部分：

|  |  |
| --- | --- |
| 。 | （8-16） |

至此，新时刻的单元状态和隐含状态都已更新。

上述为基本的LSTM。人们后来又提出了各种LSTM的变形。文献[12]比较了很多流行的变形，文献[13]测试了一万多种RNN结构，发现在某些任务上，某些特定的结构比LSTM更好。

门控循环单元（Gated Recurrent Unit，GRU）是一种最流行的LSTM的变形。GRU将遗忘门和输入门组合成一个单一的更新门，合并了单元状态和隐藏状态，并进行了一些其他更改。GRU所得到的模型比标准 LSTM 模型更简单，但性能相当。

GRU的计算过程为：

|  |  |
| --- | --- |
| ，  ，  ，  ， | （8-17） |

其中表示重置门，表示更新门。重置门决定是否将之前的状态忘记（作用相当于合并了LSTM中的遗忘门和输入门）。趋于0的时候，前一个时刻的状态信息会被忘掉，隐藏状态会被重置为当前输入的信息。更新门决定是否要将隐藏状态更新为新的状态（作用相当于LSTM中的输出门）。

## **标题2**  8.5 残差神经网络

随着网络深度的不断增加，人们发现深度CNN网络达到一定深度后，再一味地增加层数并不能带来进一步的分类性能提高。图8-11给出了20层网络和56层网络在CIFAR-10训练集和测试集上的性能。我们发现不管在训练集还是测试集上，56层的深层网络性能反而比20层网络的性能更差。这里56层网络在测试集上性能不好并不是由于过拟合引起的，因为该网络在训练集上深层网络的性能就不如浅层网络，我们称之为“退化问题”。



图8-11 20层网络和56层网络在CIFAR-10上的训练误差（左）和测试误差（右）[11]

为了解决退化问题，何恺明等人提出了残差网络（ResNet） [11]。它在2015年的ImageNet图像识别挑战赛夺魁，并深刻影响了后来的深度神经网络的设计。ResNet的结构可以极快地加速超深神经网络的训练，模型的准确率也有非常大的提升。

ResNet的基本思想是在网络中增加直连通道，直接把恒等映射作为网络的一部分，以保证在堆叠网络的过程中，网络至少不会因为继续堆叠而产生退化。但已有的神经网络很难拟合恒等映射函数，因此将网络设计为学习恒等映射的残差，拟合残差比拟合恒等映射更容易。

ResNet的基本模块为如图8-12所示的残差模块。图中右侧的曲线叫做跳接（Shortcut或Skip Connection），通过跳接在激活函数前，将上一层（或几层）之前的输出与本层计算的输出相加，将求和的结果输入到激活函数中做为本层的输出。在残差模块中，输入可通过跨层的数据线路更快地向前传播。

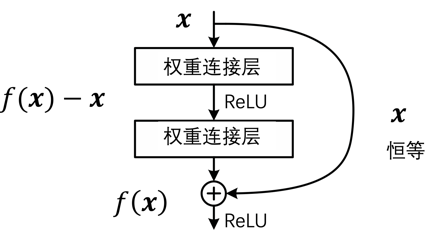


图8-12 残差模块结构。

## **标题2**  8.5 丢弃法（Dropout）

深度学习模型常使用丢弃法来防止过拟合。丢弃是指在深度学习网络的训练过程中，对于每层的神经元，按照一定的概率将其暂时从网络中丢弃。丢弃有不同的实现方法，常用的方法是倒置丢弃法（Inverted Dropout）。也就是说，每次训练时，隐含层每一层都有部分神经元的输出为0，起到简化复杂网络模型的效果，从而避免发生过拟合。一个丢弃率为0.5的网络如8-13（a）所示。

|  |  |
| --- | --- |
|  |  |
| （a）网络丢弃示意 | （b）丢弃可视为集成学习Bagging |

图8-13 丢弃法

假设对于第层神经元，设定保留神经元的概率，则该层有的神经的输出为0。令随机变量为0和1的概率分别为和，对该层神经元的输出元按下述规则进行缩放：

|  |  |
| --- | --- |
|  | （8-18） |

由于，因此

|  |  |
| --- | --- |
|  | （8-19） |

即丢弃法不改变其输入的期望值。

在神经网络迭代训练时，每次迭代随机删除掉隐藏层一定数量的神经元，然后在剩下的神经元上正向和反向更新权重和偏置项；下一次迭代中，恢复之前删除的神经元，重新随机删除一定数量的神经元，进行正向和反向更新和。不断重复上述过程，直至迭代训练完成。

丢弃通过每次迭代训练时，随机选择不同的神经元，相当于每次都在不同的神经网络上进行训练，类似集成学习中Bagging的方法，能够防止过拟合，一个例子如图8-13（b）所示。

还可以从权重的角度来解释为什么丢弃能够有效防止过拟合。对于某个神经元来说，某次训练时，它的某些输入在丢弃的作用被过滤了。而在下一次训练时，又有不同的某些输入被过滤。经过多次训练后，某些输入被过滤，某些输入被保留。这样，该神经元就不会受某个输入非常大的影响，影响被均匀化了。也就是说，对应的权重不会很大。这样从效果上来说，与L2正则类似，都是对权重进行惩罚，减小了的值。

在测试模型时，为了得到更加确定性的结果，一般不使用丢弃法。

## **标题2**  8.7 小结

在设计神经网络时，输入层的结点数目与特征的维度匹配，输出层的结点数目与目标的维度匹配。而中间隐含层的层数、每个隐含层的结点数目，层间的连接方式等等由设计者指定的。这些参数为模型的超参数，不同超参数对应模型的性能差异非常大，通常根据经验来设置几个候选值，选择效果最好的值作为最终选择。最近，自动机器学习（AutoML）或神经网络架构搜索（Neural Architecture Search，NAS）也成为机器学习的研究热点之一。

对DNN，超参数只有隐含层的层数及每个隐含层的结点数目。隐含层越多、每个隐含层的结点数目越多，模型越复杂。DNN中每一结点都能看到全体输入，信息全，但网络参数也多，训练模型需要更多样本和计算量。

CNN的层次结构特别适合图像处理。对于图像数据来说，数据的信息与结构在语义层面上都是组合性的，整体图像的语义是由局部抽象特征组合而成。因此深度CNN这种层级表征结构能依次从简单特征组合成复杂的抽象特征，如我们可以用线段等简单特征组合成简单形状，再进一步组合成物体图像各部位的特征。随着网络的层数增加，每一层对于前一层次的抽象表示更深入。通过抽取更抽象的特征来对事物进行区分，从而获得更好的区分与分类能力。因此一般我们认为CNN网络的层数越多，意能够提取到的特征越丰富越抽象，越具有语义信息。

CNN的超参数除了网络层数外，卷积核的超参数也很重要。早期的在AlexNet中有有11x11的卷积核与5x5的卷积核，但在后来的VGG网络[14]中层数增加，卷积核都变成3x3与1x1，这样可以减少训练时候的计算量，有利于降低总的参数数目。将大卷积核替换为小卷积核可有两种方法实现：

• 将高维卷积差分为多个连续低维卷积

三维卷积可拆分成二维卷积，二维卷积都可拆分为两个一维的卷积。如11x11的卷积可以转换为1x11与11x1两个连续的卷积核计算，总的运算次数由次，变成次。

• 将大的卷积用多个连续小的卷积替代，大的二维卷积核可通过几个小的二维卷积核替代。如的卷积，可以通过两个连续的的卷积替代，计算次数由次，变为次。

对RNN，其梯度较容易出现衰减或爆炸，使得循环神经网络较难捕捉时间序列中时间步距离较大的依赖关系。裁剪梯度（限制最大梯度）可以应对梯度爆炸，但无法解决梯度衰减的问题。因此，在实际应用中，通常采用门控循环神经网络或长短时记忆网络。

## **标题2**  8.8 练习

1. 下列哪一项使神经网络具有非线性？
2. 随机梯度下降
3. 整流线性单元
4. 卷积函数
5. 关于神经网络的模型容量（神经网络逼近复杂函数的能力），下述说法正确的是哪些？
6. 随着隐藏层数量的增加，模型容量增加；
7. 神经网络不能对函数建模；
8. 带有线性激活函数的单隐含层的全联接神经网络可以对函数建模。
9. 下述神经网络中，哪些网络中存在权值共享？
10. 卷积神经网络
11. 循环神经网络
12. 全连接神经网络
13. 假设分别采用DNN和CNN对二维图像进行分类。当输入图像尺寸变大时，DNN和CNN的结构需要做什么变化，参数的数目会怎样变化。
14. 使用丢弃法，在测试时：
15. 随机丢弃去除一些神经元，但无需保留训练中使用的缩放因子
16. 无需随机丢弃去除一些神经元，也无需保留训练中使用的缩放因子
17. 随机丢弃去除一些神经元，但要保留训练中使用的缩放因子
18. 将丢弃法中的参数从0.5增加到0.6可能会导致下列哪些情况？
19. 增强正则化效果
20. 减少正则化效果
21. 神经网络的训练误差更高
22. 神经网络的训练误差更低