第九章 卷积网络

1. 卷积网络基本概念

**卷积网络**是一种专门用来处理具有类似网络结构的数据的神经网络。例如时间序列数据和图像数据。卷积网络指至少在网络的一层中使用卷积运算来替代一般的矩阵乘法运算的神经网络。

1. 卷积运算

在通常形式中，卷积是对两个实变函数的一种数学运算。

**卷积**（convolution）运算：

设与是R上的两个可积函数，对于积分：

称该运算为卷积。卷积运算常用星号表示：

在卷积网络的术语中，卷积的第一个参数（函数）通常叫做**输入**（input），第二个参数（函数）叫做**核函数**（kernel function）。输出有时被称为**特征映射**（feature map）。

定义离散形式的卷积运算：

在机器学习的应用中，输入通常是多维数组的数据，而核通常是由学习算法优化得到的多维数组的参数。我们把这些多维数组叫做张量。因为在输入与核中的每一个元素都必须明确地分开存储，我们通常假设在存储量数值的有限点集以外，这些函数的值都为零，意味着在实际操作中，我们可以通过对有限个数组元素的求和来实现无限求和。

可以在多个维度上进行卷积运算：

例如在二维空间中，设输入为，二维核函数为，二维卷积运算：

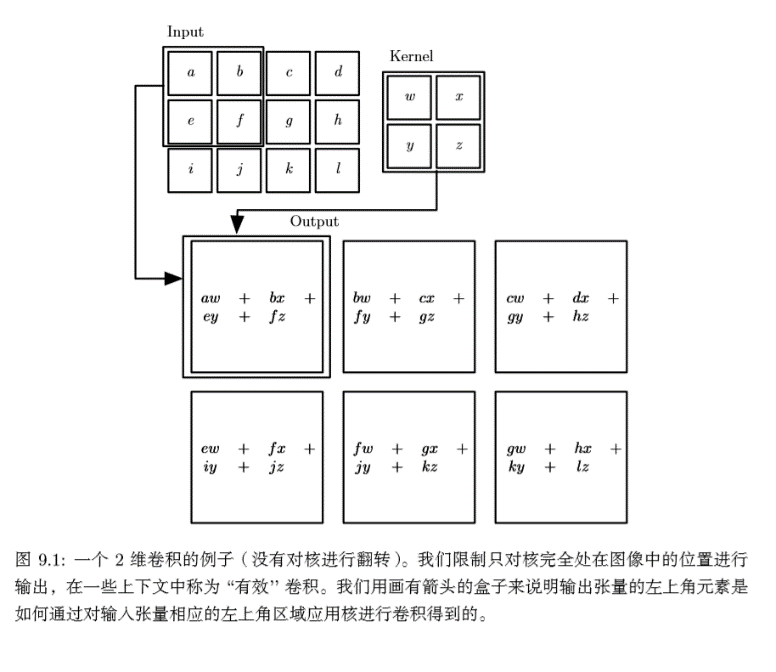
卷积是可以交换（commutative）的，上式可等价写作：

卷积运算可交换性出现是因为将核相对输入进行了翻转（flip）。我们将核翻转的唯一目的是实现可交换性。许多神经网络库会实现一个相关的函数，称为**互相关函数**（cross-correlation），和卷积运算几乎一样但并没有对核进行翻转：

在机器学习中，学习算法会在核合适的位置学得恰当的值，所以一个基于核翻转的卷积运算的学习算法所学习到的核，是对未进行翻转的算法学得的核的翻转。

在机器学习中，卷积经常和其他函数一起使用，无论卷积运算是否对它的核进行翻转，这些函数的组合通常是不可交换的。

例：一个在2维张量上的卷积运算（没有对核进行翻转）：



离散卷积可以看作矩阵的乘法，然而这个矩阵的一些元素被限制为必须和另外一些元素相等。例如单变量离散卷积中，矩阵每一行元素与上一行对应平移一个单位的元素相同。这种矩阵叫做**Toeplitz矩阵**（Toeplitz matrix）。对于二维情况，卷积对应着一个**双重分块循环矩阵**（doubly block circulant matrix）除了这些元素相等的限制以外，卷积通常对应着一个非常稀疏的矩阵（一个几乎所有元素都为0的矩阵），因为核的大小通常要远小于输入图像的大小。

1. 动机

卷积运算通过三个重要的思想来帮助改进机器学习系统：

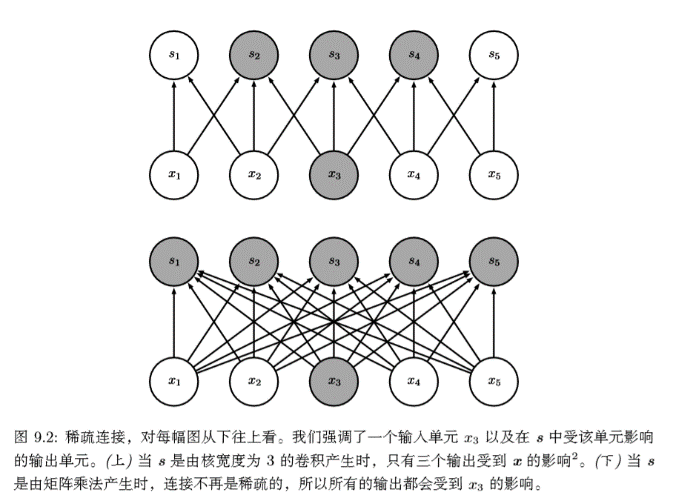
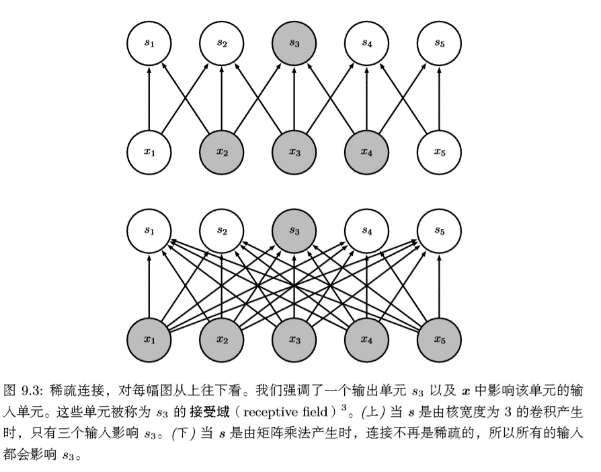
**稀疏交互（sparse interactions）**

**参数共享（parameter sharing）**

**等变表示（equivariant representations）**

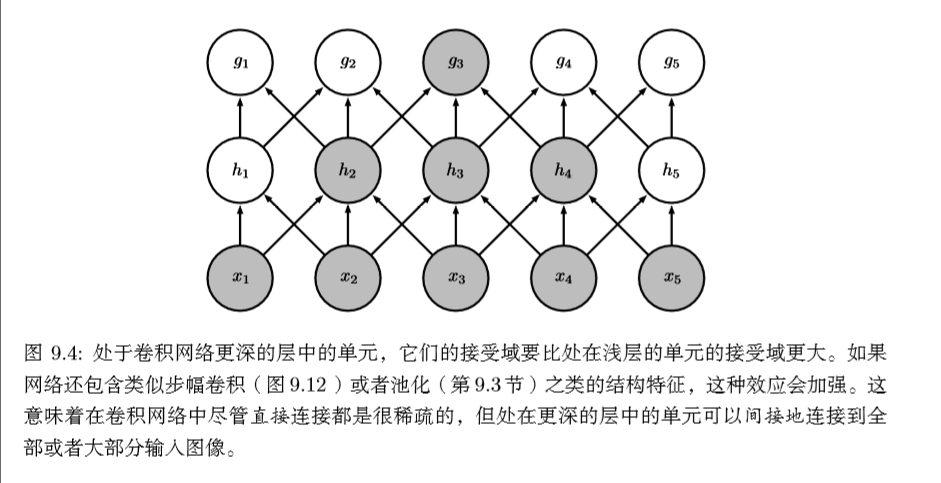
**稀疏交互**（sparse interactions）（也叫做稀疏连接（sparse connectivity）或者稀疏权重（sparse weights））的特征。

这是使核的大小远小于输入的大小来达到的。这意味着我们需要存储的参数更少，不仅减少了模型的存储需求，而且提高了它的统计效率。

例：稀疏连接：

在很多实际应用中著需要保持k比m小几个数量级就能在机器学习中取得好的表现。在深度卷积网络中，处在网络深层的单元坑内与绝大部分输入是间接交互的。

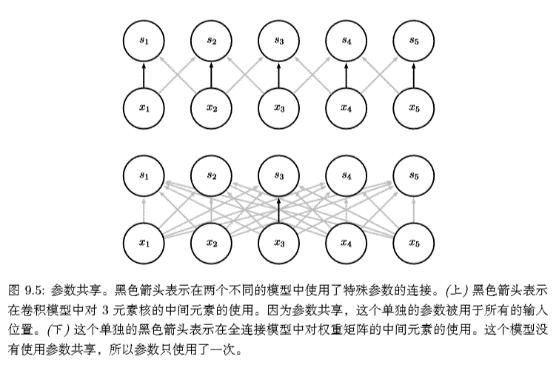
例：网络可以通过只描述稀疏交互的基石来高效地描述多个变量的复杂交互。



**参数共享**（parameter sharing）是指一个模型的多个函数中使用相同的参数。

即一个网络含有绑定的权重（tied weights）。在卷积神经网络中核的每一个元素都作用在输入的每一位置上。卷积运算中的参数共享保证了我们只需要学习一个参数集合，而不是对于每一位置都需要学习一个单独的参数集合。

例：参数共享的实现、提高边缘检测的效率：

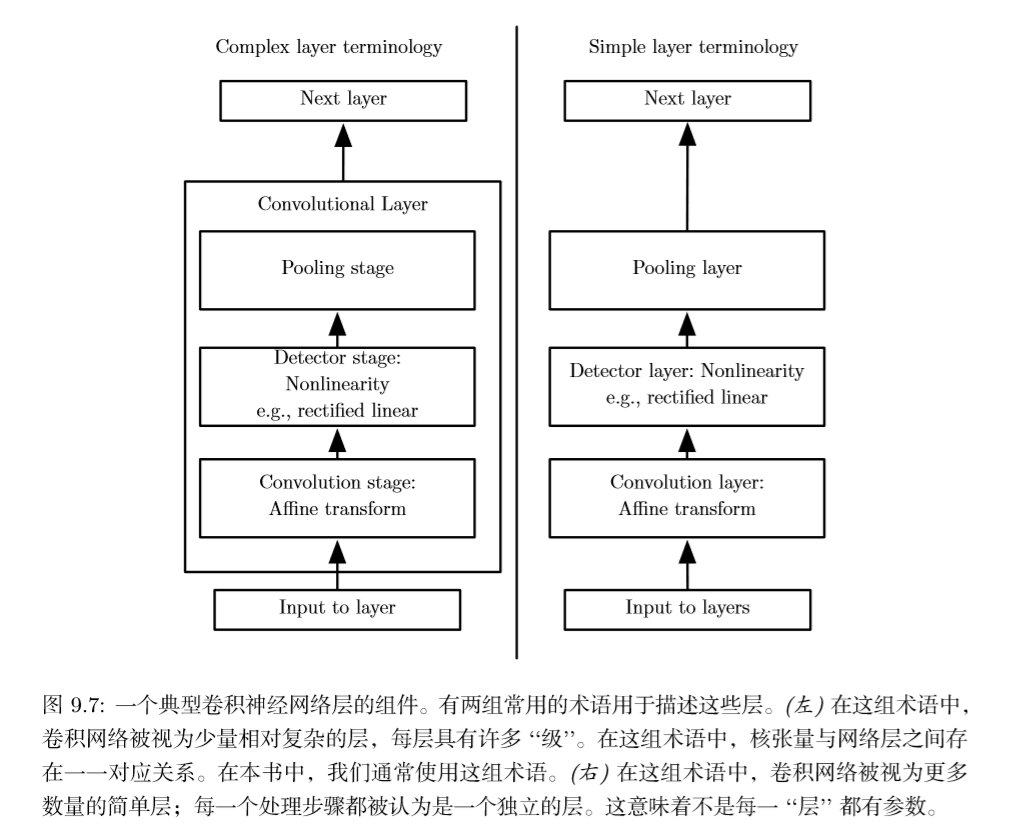


对于卷积，参数共享的特殊形式使得神经网络层具有对平移**等变**（equivariance）的性质。如果一个函数满足输入改变，输出也以同样的方式改变这一性质，就说它是等变的。特别地，若函数与满足就说对具有等变性。对于卷积来说，如果令g是输入的任意平移函数，则卷积函数对g具有等变性。

卷积对其他的一些变换并不是天然等变的，例如对于图像的放缩或旋转变换，需要其他的机制来处理这些变换。

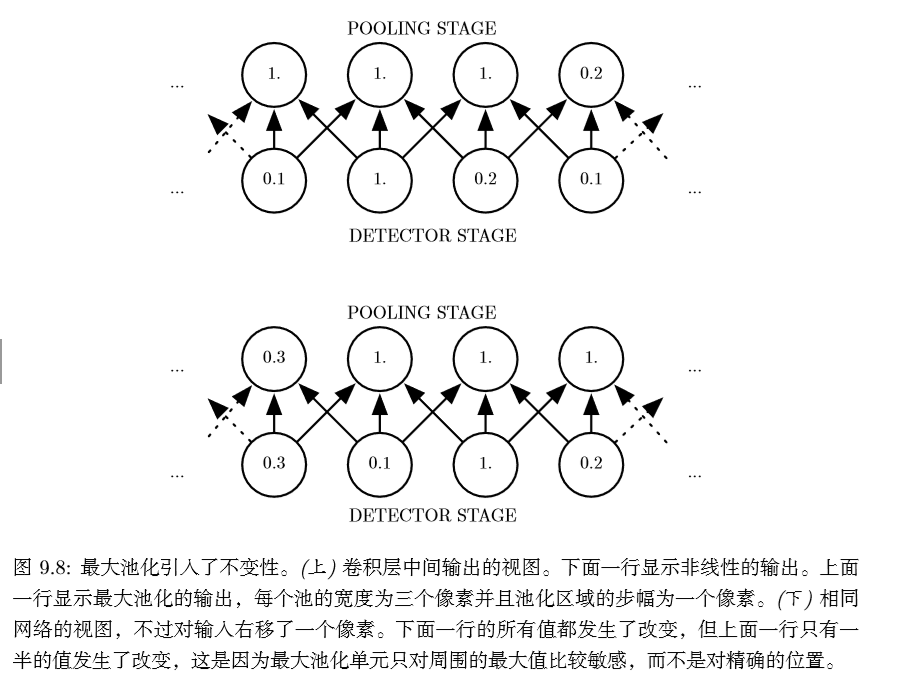
1. 池化

卷积网络一个典型层包含三级。第一级，并行地计算多个卷积产生一组线性激活响应。第二级，每一个线性激活响应将会通过一个非线性的激活函数，例如整流线性激活函数，这级有时也被称为**探测级**（detector stage）。第三级中，使用**池化函数**（pooling function）来进一步调整这一层的输出。



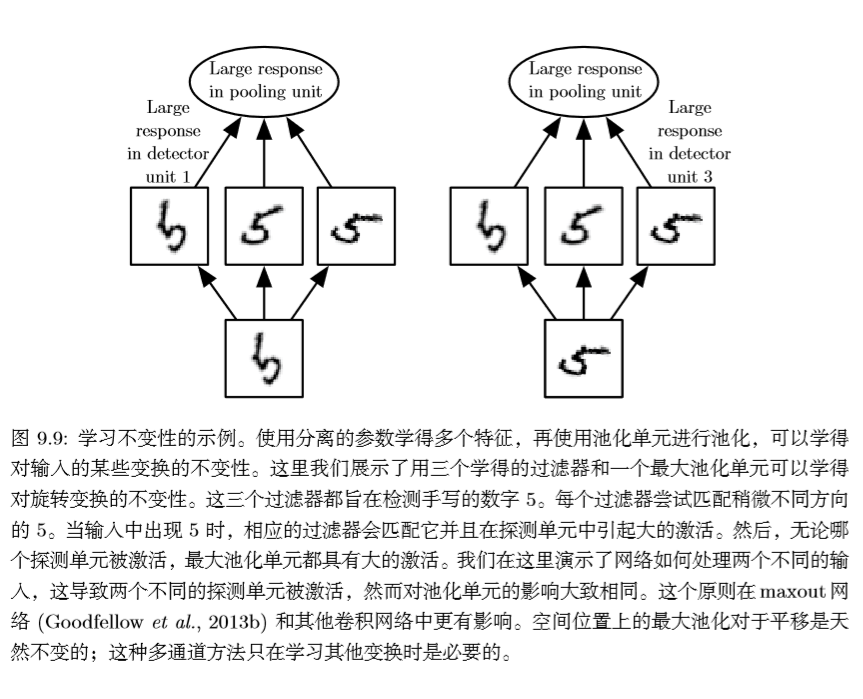
池化函数使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出。**最大池化**（max pooling）函数给出相邻矩形区域内的最大值。其他常用的池化函数包括相邻矩形区域内的平均值、范数以及基于据中心像素距离的加权平均函数。

无论怎样的池化函数，当输入做少量平移时，池化能够帮助输入的表示近似**不变**（invariant）。（即平移不变性，当输入进行少量平移时，经过池化函数后大多数输出并不会发生改变）。局部平移不变性是一个很有用的性质，尤其是当我们关心某个特征是否出现而不关心它出现的具体位置时。

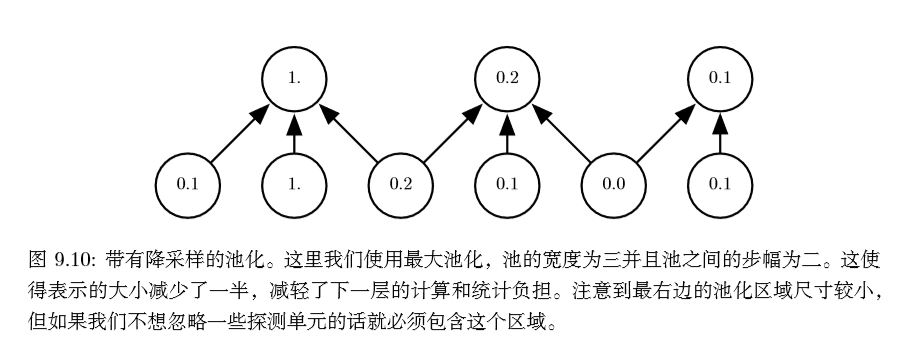


使用池化可以看作是增加了一个无限强的先验：这一层学得的函数必须具有对少量平移的不变性。当这个假设成立时，池化可以极大地提高网络的统计效率。

对空间区域进行池化产生了平移不变性，但当我们对分离参数的卷积的输出进行池化时，特征能够学得应该对于那种变换具有不变性。

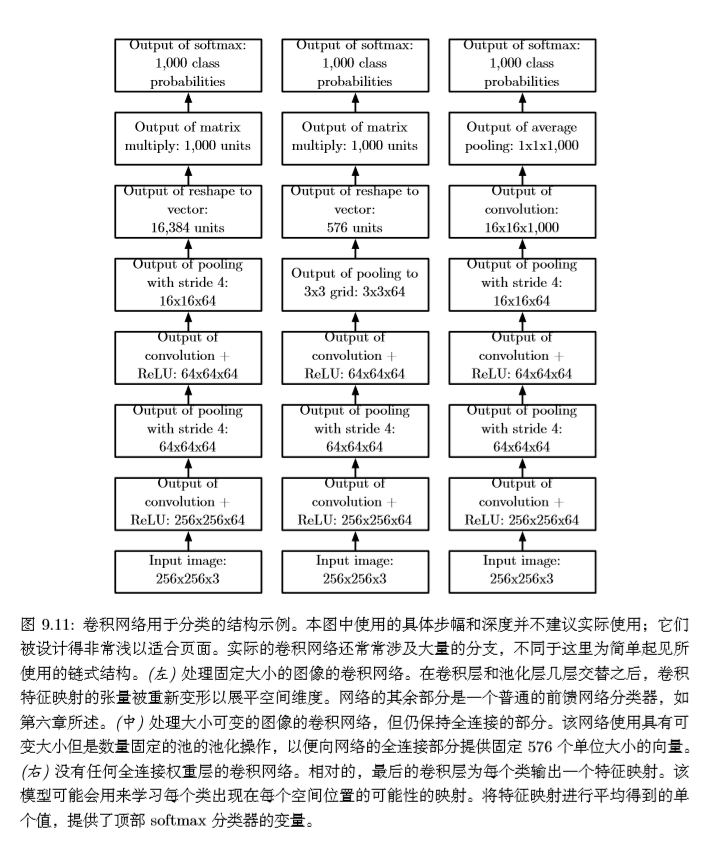


池化综合了全部邻居的反馈，使得池化单元可以小于探测单元，我们可以通过综合池化区域的k个像素的统计特征，而不是单个像素来实现。



在很多任务中，池化对于处理不同大小的输入具有重要作用。将特征一起动态地池化也是可行的。

例：使用卷积和池化操作的用于分类的完整卷积网络结构：



1. 卷积与池化作为一种无限强的先验

先验被认为是强或者弱取决于先验中概率密度的集中程度。弱先验具有较高的熵值，。这样的先验允许数据对于参数的改变具有或多或少的自由性。强先验具有较低的熵值，这样的先验在决定参数最终取值时起着更加积极的作用。一个无限强的先验需要对一些参数的概率置零并且完全禁止对这些参数赋值，无论数据对于这些参数的值给出了多大的支持。

我们可以把卷积的使用当作是对网络中一层的参数引入了一个无限强的先验概率分布。这个先验说明了该层应该学得的函数只包含局部连接关系并且对平移具有等变性。类似的，使用池化也是一个无限强的先验：每一个单元都具有对少量平移的不变性。

把CNN网络当作一个具有无限强先验的全连接网络来实现会导致极大的计算浪费，但可以了解运行原理。

卷积和池化可能导致欠拟合。例如如果任务依赖于保存精确的空间信息，则在所有的特征上使用池化将会增大训练误差。

当比较卷积模型的统计学习表现时，只能以基准中的其他卷积模型作为比较的对象。

1. 基本卷积函数的变体

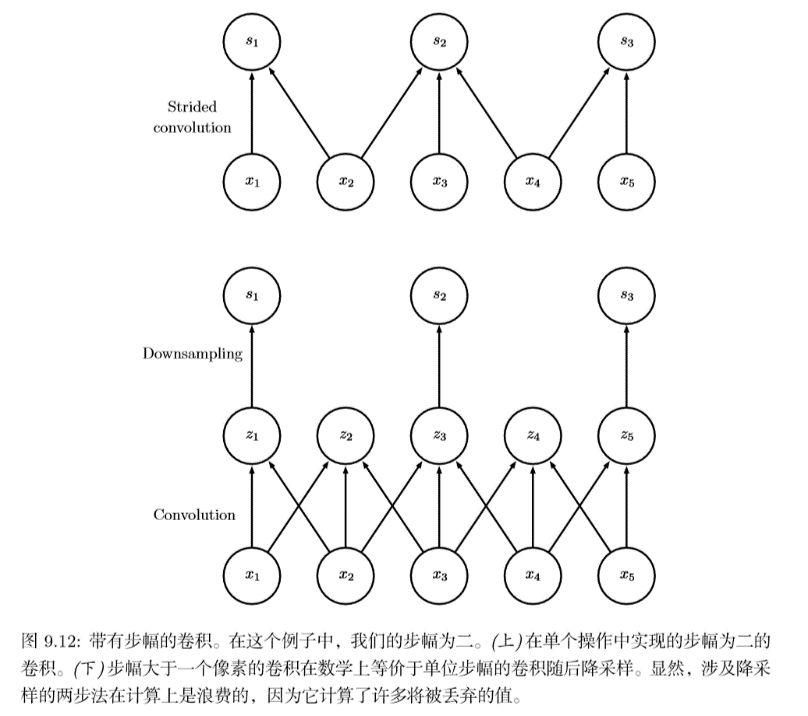
神经网络中的卷积通常指由多个并行卷积组成的运算，因为具有单个核的卷积只能提取一种类型的特征，尽管它作用在多个空间位置上。我们通常希望网络的每一次能够在多个位置上提取多种类型的特征。

输入通常不仅仅是实值的网格，而是由一系列观测数据的向量构成的网络。

因为卷积网络通常使用多通道的卷积，所以即使使用了核翻转，也不一定保证网络的线性运算是可交换的。只有当其中的每个运算的输出和输入具有相同的通道数时，这些多通道的运算才是可交换的。

假设有一个4为核张量K，它的每个元素是，表示输出中处于通道的一个单元和输入中通道中的一个单元的连接强度，并且在输入单元之间有k行l列的偏置。V为输入的观测数据，每个元素为。设输入Z和输出V有相同的形式且不涉及翻转卷积核张量。则输出Z：

我们可以跳过核中的位置来降低计算开销，即看作对全卷积函数输出的下采样（downsampling）。在输出的每个方向上每间隔s个像素进行采样，可定义一个**下采样卷积函数**：

将s称为下采样卷积的**步幅**（stirde）

在任何卷积网络的实现中都有一个重要性质，即对输入V用0进行填充（pad）使之加宽。

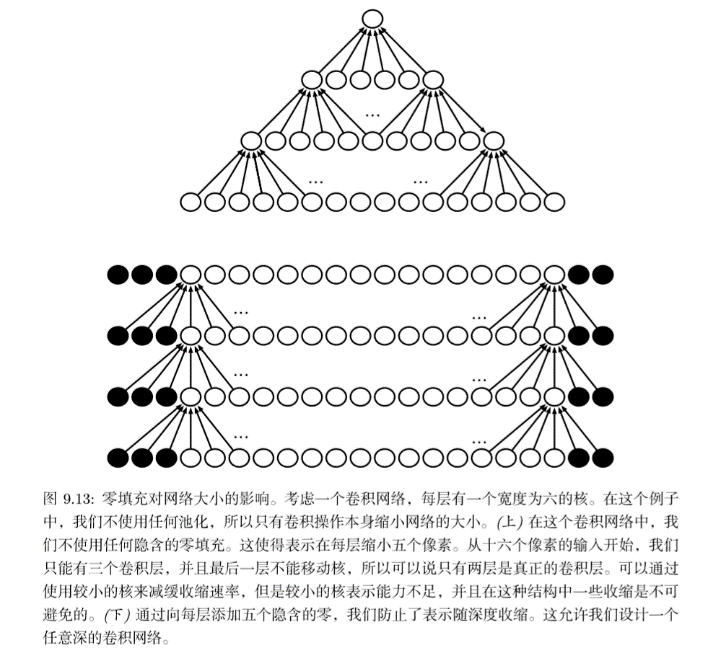
有三种零填充设定的情况值得注意：

**有效**（valid）**卷积**：无论怎样都不使用零填充的情况，并且卷积核只允许访问那些图像中能够完全包含整个核的位置。输出的所有像素都是输入中相同数量像素的函数，使得输出像素的表示更加规范，而输出的大小在每一层都会缩减。

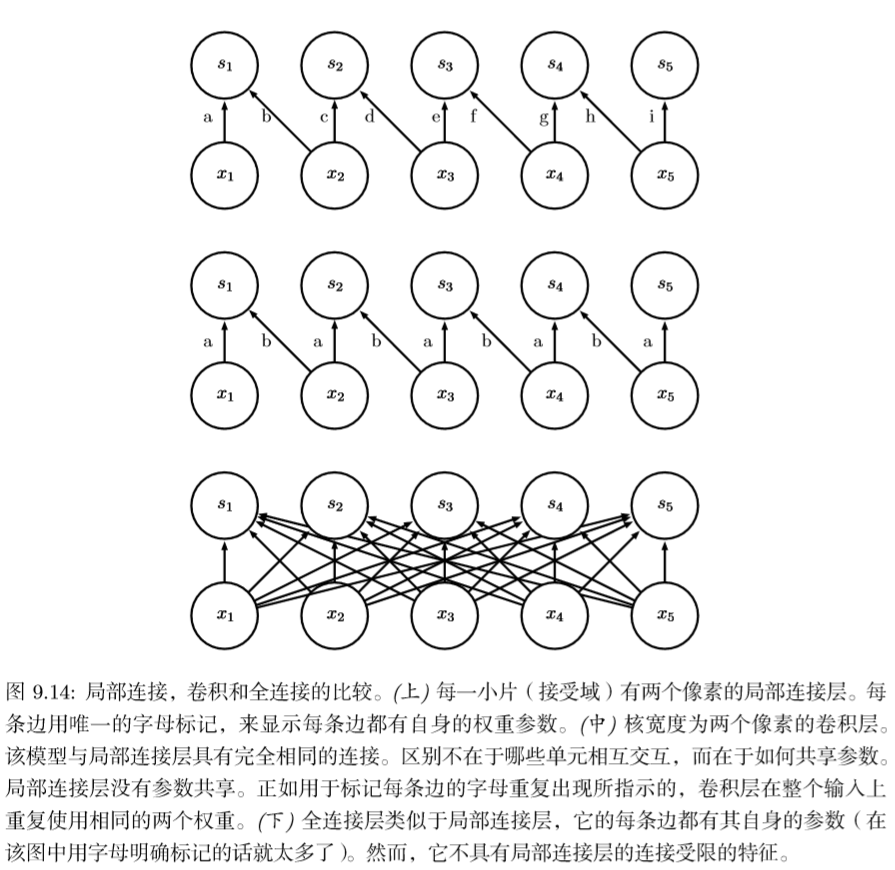
**相同**（same）**卷积**：只进行足够的零填充来保持输入和输出具有相同的大小。在这种情况下卷积运算不改变下一层的结构，而输入像素靠近边界的部分相比较于中间部分对于输出像素的影响更小，可能会导致边界像素存在一定的欠表示

**全**（full）**卷积**：进行足够多的零填充使得每个像素在每个方向上下好被访问了k次，最终输出图像的宽度为m+k-1。这种情况下，输出像素中靠近边界的部分相比于中间的部分是更少像素的函数。导致学得一个在卷积特征映射的所有位置都表现不错的单核更为困难。

通常零填充的最优数量处于“有效卷积”与“相同卷积”之间。



**非共享卷积**（unshared convolution）：具有一个小核的离散卷积运算很像，但并不横跨位置来共享参数，是一些局部连接的网路层，此时多层感知机对应的邻接矩阵是相同的，但每个连接都有它自己的权重。局部连接层的线性部分：

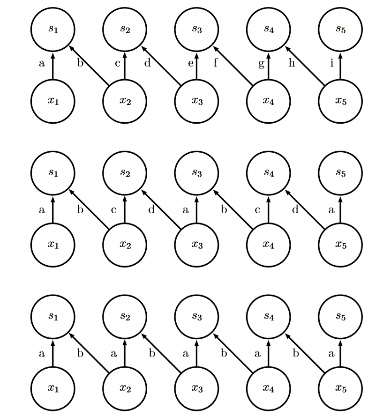


**平铺卷积**（tiled convolution）：学习一组核使得核在空间移动时可以循环利用，意味着在近邻的位置上拥有不同的过滤器，但对于这些参数的存储需求仅仅会增长常数倍，这个常数为核的集合的大小。注意输出的位置在每个方向上在t个不同的核组成的集合中进行循环。

这里的%代表取模运算，性质包括t%t=0，(t + 1)%t = 1等。

局部连接层与平铺卷积层与最大池化存在关联：这些层的探测单元都是由不同的过滤器驱动的。卷积层对于平移具有内置的不变性。

例：局部连接层、平铺卷积、标准卷积比较：

****

**关于反向传播**：

需要训练一个卷积网络，其中包含步幅为s的步幅卷积，该卷积的核为k，作用于多通道的图像V，定义为c（K，V，s）。假设需要最小化某个损失函数J(V, K)。在前向传播过程中，需要用c本身来输出Z，然后Z传递到网络的其余部分并被用来计算损失函数J。在反向传播过程中，会得到一个张量G满足。

对核中的权重求导：

若此层不是网络的底层，需要对V求梯度来对误差进一步反向传播。可以使用：

关于如何在偏置项中共享参数：

对于局部连接层，对每个单元都给定它特有的偏置，对于平铺卷积，用与核一样的平铺模式来共享参数。对于卷积层来说，通常在输出的每个通道上都设置一个偏置，这个派年至在每个卷积映射的所有位置上共享，若输入是已知的固定大小，亦可以在输出映射的每个位置学习一个单独的偏置。

1. 结构化输出

卷积神经网络可以用于输出高维的结构化对象，不仅是预测分类任务的类标签或回归任务的实数值。通常这个对象是一个tensor，由标准卷积层产生。

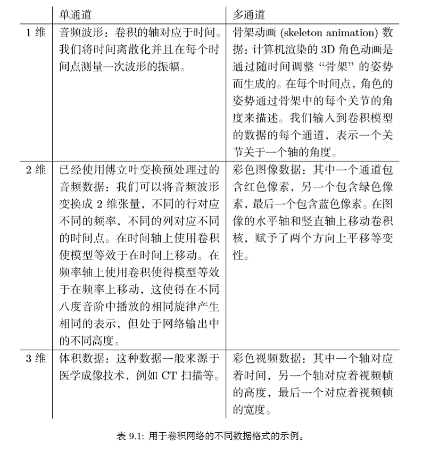
经常出现的一个问题是输出平面可能比输入平面要小。用于对图像中单个对象分类的常用结构中，网络空间维数的最大减少来源于使用大步幅的池化层。

为了产生与输入大小相似的输出，可以避免把池化放在一起，或单纯地产生一张低分辨率的标签网格，原则上可以使用具有单位步幅的池化操作。

对图像逐个像素标记的一种策略是先产生图像标签的原始猜测，然后使用相邻像素之间的交互来修正该原始猜测。（形成一种特殊的循环神经网络）

1. 数据类型

卷积网络使用的数据通常包含多个通道，每个通道是时间上或空间中某一点的不同观测量。



卷积网络一个有点是可以处理具有可变空间尺度的输入。

注意：使用卷积处理可变尺寸的输入，仅对输入是因为包含对同种事物的不同量的观察而导致的尺寸变化这种情况才有意义。若输入是因为它可以选择性地包括不同种类的观察而具有可变尺寸，使用卷积是不合理的。

1. 高效的卷积算法

卷积等效于使用傅里叶变换将输入与核都转换到频域、执行两个信号的住店相乘，再使用傅里叶逆变换转换回时域。

当一个d维的核可以表成d个向量（每一维一个向量）的外积时，该核称为**可分离**（separable）的。当核可分离，它等价于组合d个一维卷积，每个卷积使用这些向量中的一个。