

理解深度学习

2023 年 9 月 13 日

目录

- 1 卷积网络 2
 - 1.1 卷积神经网络数据集特征 2
 - 1.2 卷积神经网络的由来 4
 - 1.2.1 大脑如何处理数据 4
 - 1.2.2 卷积的由来 5
 - 1.2.3 卷积运算 5
 - 1.2.4 卷积定理 6
 - 1.2.5 卷积核 7
 - 1.3 卷积神经网络的结构 8
 - 1.3.1 填充和跳步 Padding and Stride 9
 - 1.3.2 多通道 Multiple channel 10
 - 1.3.3 池化 Pooling 10
 - 1.3.4 softmax 12
 - 1.4 Understanding CNN 14

Chapter 1

卷积网络

卷积神经网络 (Convolutional Neural Network, CNN) 是一种用于图像处理的深度神经网络。它利用局部连接和权值共享的思想,大大减少了网络的参数量,使其尤其适合处理高维数据。它通过卷积层提取输入图像中的空间关联性,然后通过池化层减少参数量和避免过拟合,最后通过全连接层完成分类或回归任务。

卷积神经网络在图像分类、对象检测、语义分割等视觉任务上表现优异。例如,在图像分类任务中,卷积神经网络可以自动学习提取图像的特征,进行图片中的物体识别。在自然语言处理任务中,卷积神经网络也用于文本分类等任务。

在本节中,我们将从卷积神经网络数据集特征,卷积神经网络的由来,卷积神经网络的结构以及对于卷积神经网络的理解这些方面来介绍卷积神经网络。

1.1 卷积神经网络数据集特征

卷积神经网络适用于处理具有网格结构的数据,如图像、视频和音频等。这些数据集大多拥有以下特性:

不变性 不变性具有多种形式,如平移不变性,旋转不变性,视点不变性,大小(伸缩)不变性和光源不变性。下图中展示了各种不变性。

局部关联 很多具有空间或者时间结构的数据都有局部关联的特点,也就是随着距离增长,两个空间点上的数据的关联强度不断下降。为了刻画衰减,我们首先介绍两种常见的衰减,指数衰减和幂级数衰减。

指数衰减的特征为具有一定的衰减周期,在经过一个衰减周期之后会变为原始数据的一半,我们称之为半衰期。例如,对于一个形如 $y = e^{-\frac{x}{\tau}}$ 的指数衰减模型,它的半衰期为 $\tau \ln(2)$ 。

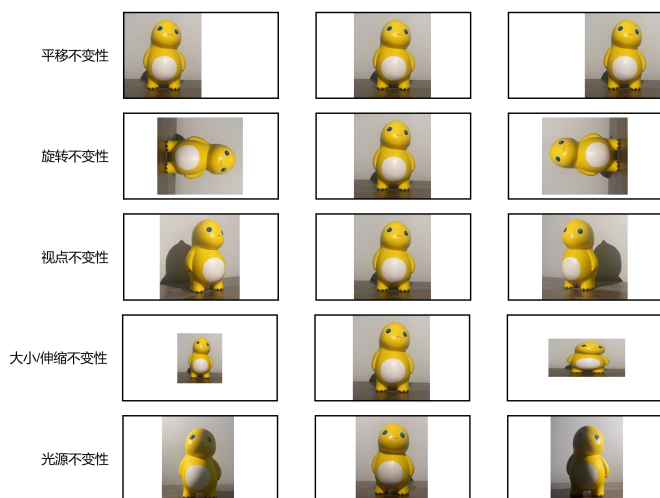
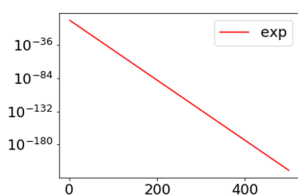
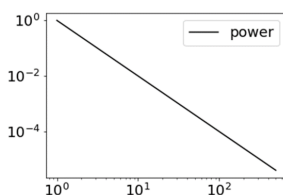


图 1.1: 图像的各种不变性

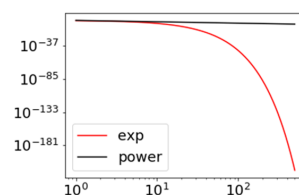
幂级数衰减没有衰减周期，如图??，它比指数衰减要慢地多，例如 $y = 1/x$ ，当 x 越大的时候，其衰减就越慢。另一种理解幂级数衰减的办法是通过指数衰减，幂级数衰减可以理解为有无穷多个半衰期的指数衰减的和，例如，考察 $\int_0^\infty \exp(-\alpha x) d\alpha = \frac{1}{x}$ ，其左式为周期各不相同的指数衰减的积分，右侧为幂级数衰减。



(a) 指数衰减



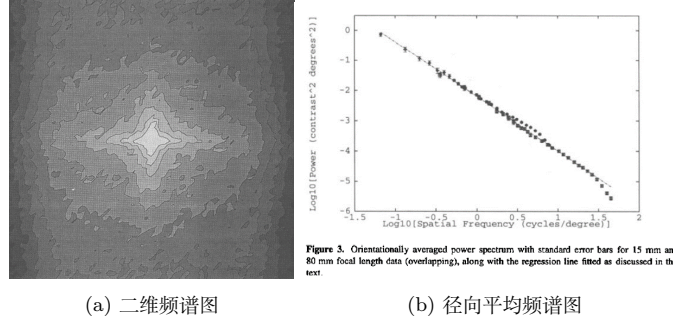
(b) 指数衰减



(c) 指数衰减和幂级数衰减速度对比

为了验证自然图像的局部关联特性，我们计算两个常用的量，功率谱和关联函数，事实上，这两个量可以通过傅里叶变换相互转换。

功率谱 频谱图为将原函数做傅里叶变换后求模长的平方得到的图像。在频谱图中，越浅表示其频率越高。研究图像的功率谱后，我们可以发现幅度关于频率基本为幂指数衰减，即低频主导，存在高频成分。



(a) 二维频谱图

(b) 径向平均频谱图

图 1.2: 频谱图。图片来源Ruderman (1994)。

关联 对于关联 I 我们的定义如下，

$$I = \int d\phi_1 d\phi_2 p_2(\phi_1, \phi_2) \log_2 \frac{p_2(\phi_1, \phi_2)}{p_1(\phi_1) p_1(\phi_2)} \quad (1.1)$$

有关时 $I > 0$ ；无关时 $p_2(\phi_1, \phi_2) = p_1(\phi_1) p_1(\phi_2)$ ，因此有 $I = 0$ 。

分析上述的定义，我们可以看到关联为局部关系主导的，同时随着距离的增大，关联会慢慢变小。

接下来我们要研究的问题就是如果两点毫无关联，我们要多少样本来学会无关。假设 x, y 无关，且均值为 1，方差为 1，则我们有如下式子：

$$\text{cov}(x, y) = \frac{1}{n} \sum_{i=1}^n x_i y_i$$

$$E(\text{cov}(x, y)) = 0$$

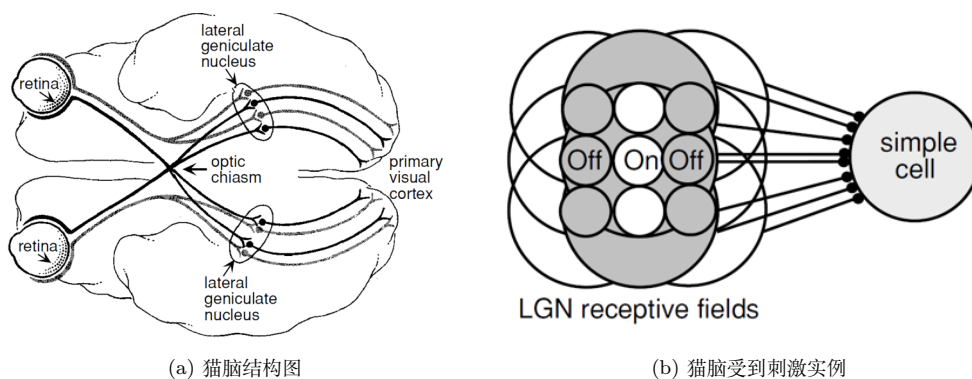
$$\text{Var}(\text{cov}(x, y)) \sim \frac{1}{n}$$

那么其标准差 $\sim \frac{1}{\sqrt{n}}$ ，也就是说，对于关联在 10^{-3} 的点，在无先验知识的情况下，我们需要 10^6 个样本才能学习到两点无关。

1.2 卷积神经网络的由来

1.2.1 大脑如何处理数据

在文献Dayan & Abbott (2005) 介绍的实验中以猫脑为例，探究光刺激变化对猫脑神经信号的影响。实验发现，猫脑对不同位置 and 不同角度的光信号会产生不同的反应。在受到刺激时，猫脑会在不同方向释放不同信号，以达到促进或者抑制的作用。



1.2.2 卷积的由来

受到猫脑对不同位置和方向的感知方式的启发，卷积应运而生，通过神经网络中卷积的平移不变性与其对方向的敏感性，我们可以在某种意义上得到人工神经网络与生物神经网络的共性。事实上，以图像为例，我们可以将图像看作一个二维的信号，或者说刺激。如图1.3所示，在黑色位置，我们可以将其理解为有刺激的区域。

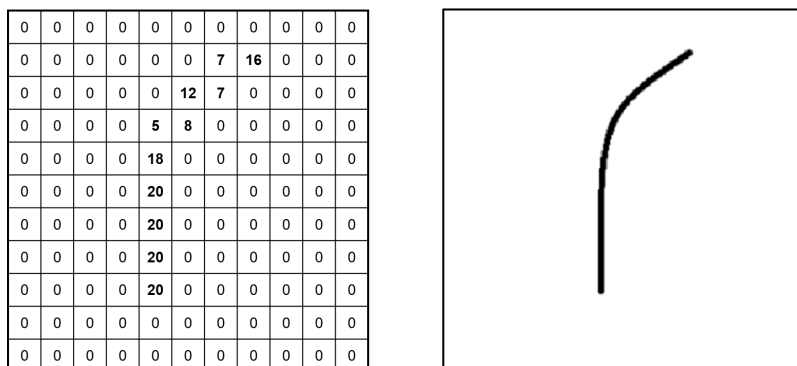


图 1.3: 图像与刺激

1.2.3 卷积运算

在具体神经网络中的卷积运算，在后文中会提到，这里我们只是在理论层面给出卷积运算的理论公式。此处考虑二维卷积运算， $f(x,y)$ 为输入特征图， $g(x,y)$ 为卷积核。

我们有：

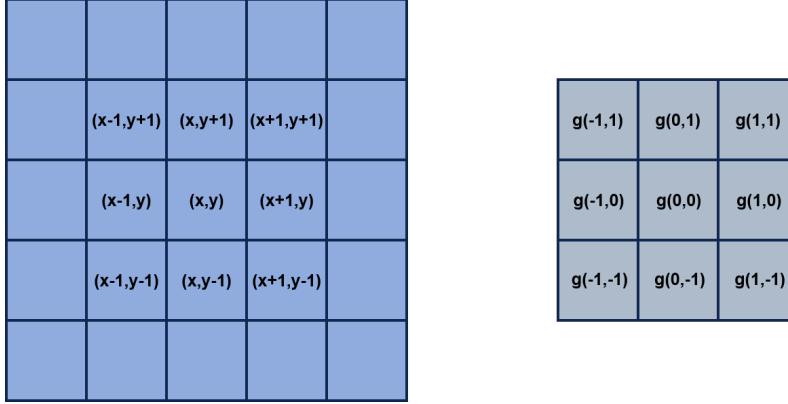


图 1.4: 卷积运算。左: 输入特征图 $f(x, y)$; 右: 卷积核 $g(x, y)$ 。

$$\begin{aligned}
 F(x, y) = & f(x-1, y+1)g(-1, 1) + f(x, y+1)g(0, 1) + f(x+1, y+1)g(1, 1) \\
 & + f(x-1, y)g(-1, 0) + f(x, y)g(0, 0) + f(x+1, y)g(1, 0) \\
 & + f(x-1, y-1)g(-1, -1) + f(x, y-1)g(0, -1) + f(x+1, y-1)g(1, -1)
 \end{aligned}$$

进一步, 有下列离散和连续形式的卷积方程:

$$\begin{aligned}
 F(x, y) &= \sum_{u, v} f(x-u, y-v)g(u, v) \\
 F(x, y) &= \int f(x-u, y-v)g(u, v) du dv
 \end{aligned}$$

1.2.4 卷积定理

Theorem 1 (卷积定理). 函数卷积的傅里叶变换是函数傅里叶变换的乘积。即一个域中的卷积对应于另一个域中的乘积, 例如时域中的卷积对应于频域中的乘积。具体而言, 有

$$\mathcal{F}[f] \cdot \mathcal{F}[g] = \mathcal{F}[f * g] \quad (1.2)$$

接下来我们给出卷积定理的一个简短的证明。事实上, 只需要证明函数卷积的傅立叶变换是函数傅立叶变换的乘积即可。

证明. 令 f, g 属于 L^1 空间, 令 F, G 分别为 f, g 的傅里叶变换, h 为 f, g 的卷积, 即

$$\begin{aligned}
F(\boldsymbol{\xi}) &= \mathcal{F}[f](\boldsymbol{\xi}) = \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x} \\
G(\boldsymbol{\xi}) &= \mathcal{F}[g](\boldsymbol{\xi}) = \int_{\mathbb{R}^n} g(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x} \\
h(\mathbf{z}) &= \int_{\mathbb{R}^n} f(\mathbf{x}) g(\mathbf{z} - \mathbf{x}) d\mathbf{x}
\end{aligned}$$

注意到

$$\iint |f(\mathbf{x}) g(\mathbf{z} - \mathbf{x})| d\mathbf{z} d\mathbf{x} = \int \left(|f(\mathbf{x})| \int |g(\mathbf{z} - \mathbf{x})| d\mathbf{z} \right) d\mathbf{x} = \int |f(\mathbf{x})| \|g\|_1 d\mathbf{x} = \|f\|_1 \|g\|_1$$

因此由 Fubini 定理, 我们得出 $h \in L^1(\mathbb{R}^n)$, 于是它的傅里叶变换 H 定义为

$$H(\boldsymbol{\xi}) = \mathcal{F}[h] = \int_{\mathbb{R}^n} h(\mathbf{z}) e^{-2\pi i \mathbf{z} \cdot \boldsymbol{\xi}} d\mathbf{z} = \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} f(\mathbf{x}) g(\mathbf{z} - \mathbf{x}) d\mathbf{x} e^{-2\pi i \mathbf{z} \cdot \boldsymbol{\xi}} d\mathbf{z}$$

注意到 $|f(\mathbf{x}) g(\mathbf{z} - \mathbf{x}) e^{-2\pi i \mathbf{z} \cdot \boldsymbol{\xi}}| = |f(\mathbf{x}) g(\mathbf{z} - \mathbf{x})|$ 因此对以上变量我们可以再次应用 Fubini 定理 (即交换积分顺序):

$$H(\boldsymbol{\xi}) = \int_{\mathbb{R}^n} f(\mathbf{x}) \left(\int_{\mathbb{R}^n} g(\mathbf{z} - \mathbf{x}) e^{-2\pi i \mathbf{z} \cdot \boldsymbol{\xi}} d\mathbf{z} \right) d\mathbf{x}$$

做变量替换 $\mathbf{y} = \mathbf{z} - \mathbf{x}$, $d\mathbf{y} = d\mathbf{z}$, 得到

$$\begin{aligned}
H(\boldsymbol{\xi}) &= \int_{\mathbb{R}^n} f(\mathbf{x}) \left(\int_{\mathbb{R}^n} g(\mathbf{y}) e^{-2\pi i (\mathbf{y} + \mathbf{x}) \cdot \boldsymbol{\xi}} d\mathbf{y} \right) d\mathbf{x} \\
&= \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \boldsymbol{\xi}} \left(\int_{\mathbb{R}^n} g(\mathbf{y}) e^{-2\pi i \mathbf{y} \cdot \boldsymbol{\xi}} d\mathbf{y} \right) d\mathbf{x} \\
&= \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x} \int_{\mathbb{R}^n} g(\mathbf{y}) e^{-2\pi i \mathbf{y} \cdot \boldsymbol{\xi}} d\mathbf{y} \\
&= F(\boldsymbol{\xi}) \cdot G(\boldsymbol{\xi})
\end{aligned}$$

□

1.2.5 卷积核

我们来考虑一下图片过卷积核所得到的特征图能给予网络的信息。事实上, 我们可以通过卷积核获得模糊、锐化等处理, 还可以获得某些具有方向性的特征, 这是符合生物大脑的特征构造的。具体特征图如图:

| | | |
|---|--|---|
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| Box blur (normalized) | $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian blur (approximation) | $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |  |

图 1.5: 由卷积获得的特征图。图片来源Karn (2016)。

1.3 卷积神经网络的结构

卷积神经网络 (CNN) 的典型结构包括以下几个模块:

- 卷积层 (Convolutional Layer) 这个层的主要运算是卷积操作。卷积层的参数包括卷积核的大小、数量, 卷积步长等。卷积层能够提取输入的本地相关特征。
- 池化层 (Pooling Layer) 池化层一般在卷积层之后, 通过 DOWN 采样操作减小数据维度, 并能够提取轮廓信息。常见的池化操作有最大池化、平均池化等。
- 全连接层 (Fully Connected Layer) 卷积层和池化层之后一般会接一到两个全连接层, 这就是典型的多层感知器结构。全连接层会将前层输出变成一个向量, 然后连接到最后的输出层。
- 输出层 (Output Layer) 输出层的神经元数量取决于问题的类别数量。比如分类问题中, 输出层就是分类的类别数。

一个典型的卷积神经网络结构往往由几个卷积-池化层排列组合, 而最后接上 1-2 个全连接层, 最后是一个输出层。这种结构能够逐层提取图像的特征, 并最终输出分类或回归结果。以下我们将逐步拆解这些模块中的重要组件。

1.3.1 填充和跳步 Padding and Stride

我们观察如下卷积操作 (图 1.6)，如果只是用 3×3 的卷积核与 5×5 的图像做卷积，我们最终得到一个 3×3 的输出，我们发现输出结果相对于原始图像变小了。

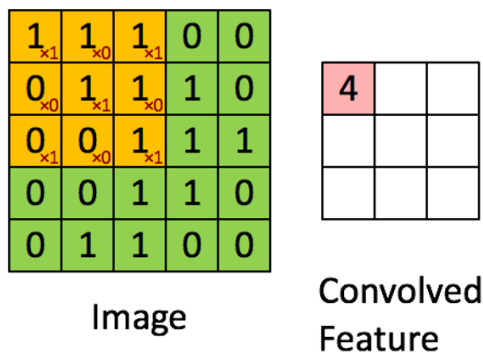


图 1.6: Convolution。原始图像为 5×5 的图像，与 3×3 的卷积核做卷积运算最终得到一个 3×3 的输出，大小比原图像小。图片来源Karn (2016)。

有时我们并不希望图像在经过卷积后“缩水”。为了解决这个问题，我们引入 Zero Padding 操作，即在做卷积之前在原始图像周围补一些 0，来确保通过卷积以后输出大小尽量与原图像大小保持一致。(图 1.7)

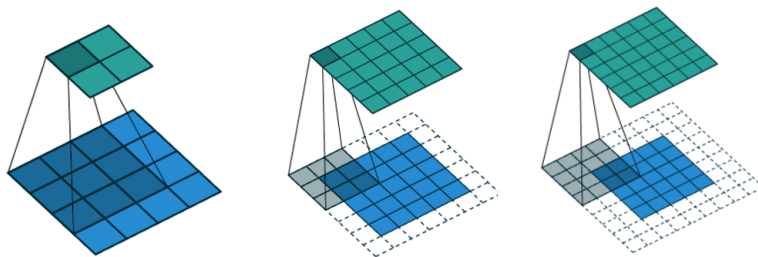


图 1.7: Zero padding。原始图像为 4×4 的图像，与 3×3 的卷积核做卷积运算最终得到一个 2×2 的输出。在加入一层 zero padding 之后，图像由 5×5 变为 7×7 ，卷积运算的输出结果为 5×5 。在加入两层 zero padding 之后，图像变为 9×9 ，与 5×5 卷积核做卷积，输出结果为 6×6 。

观察图 1.8 中第一行所示的卷积操作，做卷积时卷积核一步一步在原图像上移动来得到输

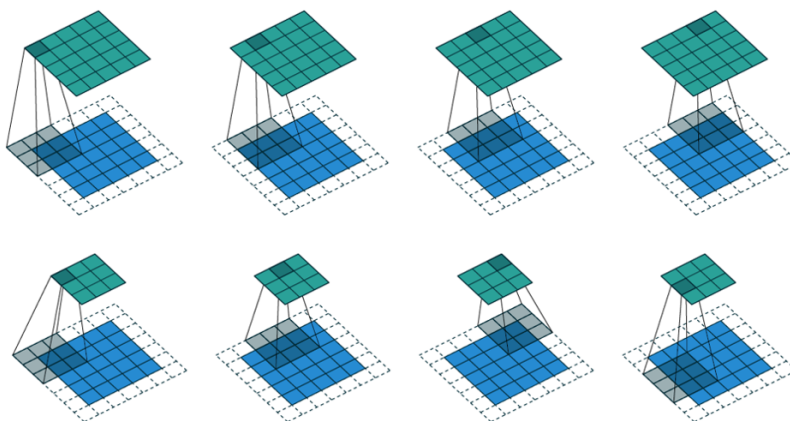


图 1.8: Stride。图中所示的是 $\text{stride}(2,2)$ 的卷积运算， 5×5 的原始图像通过 zero padding 操作变为 7×7 的图像，用一个 3×3 的卷积核与它做卷积，横向纵向步长均为 2，最终得到一个 3×3 的输出。

出，我们称这种运算为步长为 1 的卷积运算；如第二行所示，我们也可以令卷积核一次移动两个单位，我们称这种运算为步长为 2 的卷积运算；一般的，如果卷积核横向步长为 i ，纵向步长为 j ，我们就得到了一个 $\text{stride}(i,j)$ 的卷积运算。这样做的好处是避免了卷积过程中图像中单个像素点参与了多次卷积运算，造成较大的信息重叠；并且有助于减小计算量。

最后我们通过 1 个例子 (图 1.9) 来理解卷积和 Padding、 $\text{stride}(i,j)$ 的运算机制。

1.3.2 多通道 Multiple channel

接下来我们来看一个多通道 (Multiple channel) 的模型，首先我们理解一下什么叫多通道，例如一张彩色照片，我们知道彩色照片是由红、绿、蓝三种颜色混合而成的，因此图片上红色、绿色、蓝色的值分别构成了一个输入的“片”，我们称之为一个通道。

图 1.10 中所示的是一个三通道模型，我们分别用三个 3×3 的卷积核与它们分别做卷积，在每个点上获得的数值相加再加上一个共同的 bias 项，就得到了一个输出。注意因为我们每个神经元共享参数，所以我们的偏置项 (bias) 是不变的。

1.3.3 池化 Pooling

在完成卷积运算后，得到的输出对于计算机而言可能还是很大，为了提升计算的效率，我们可以做适当的下采样，这里我们引入一个新的操作 Pooling，Pooling 分为 max pooling 和 average pooling 两种，max pooling 将选中区域中最大值提取出来作为输出，average pooling

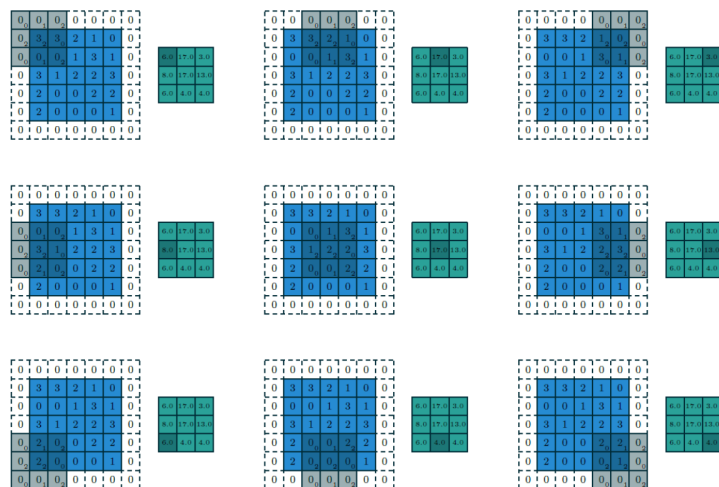


图 1.9: 计算离散卷积运算的输出。图中所示的是 stride(2,2) 的卷积运算, 5×5 的原始图像通过 zero padding 操作变为 7×7 的图像, 用一个 3×3 的卷积核与它做卷积, 横向纵向步长均为 2, 最终得到一个 3×3 的输出。

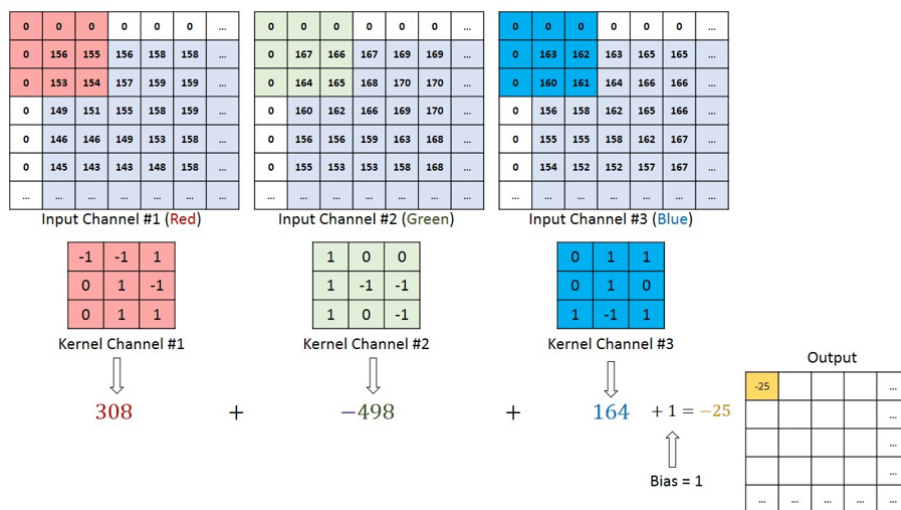


图 1.10: Multiple channel. 分别用三个 3×3 的卷积核与它们分别做卷积, 在每个点上获得的数值相加再加上一个共同的 bias 项, 就得到了一个输出。因为我们每个神经元共享参数, 所以我们的偏置项 (bias) 是不变的, 该例中 bias 恒定为 1。在该卷积运算中, 3 个卷积核提供了 $3 \times 3 \times 3 = 27$ 个参数, 加上 bias 项, 一共是 28 个参数。图片来源Karn (2016)。

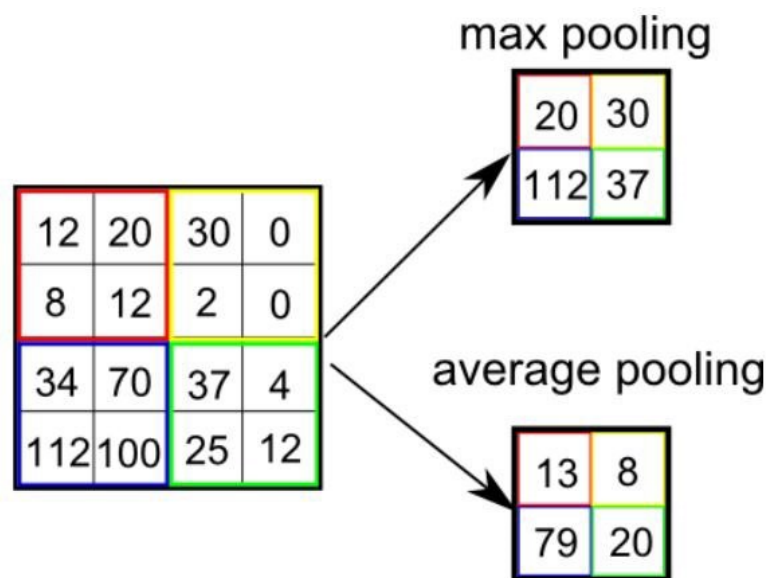


图 1.11: Pooling. 输入为 4×4 图像，将其划分为 4 个 2×2 的小块，max pooling 将每一块中的最大值作为输出，average pooling 将每一块中的平均值作为输出，两种 pooling 方式均得到 2×2 的输出。图片来源Karn (2016)。

将选中区域中所有值的平均作为输出。(图 1.11)

Pooling 怎么做梯度下降？

因为 Pooling 不是一个单射函数，所以理论上我们无法直接用数学的方法去求它的梯度，为了计算需要，我们人为定义 Pooling 按照如下方式进行梯度计算：

对于 max pooling，我们记录最大值点的位置，在梯度反向传播的过程中，我们把梯度全部传递到该位置，即仅对该位置做梯度下降，其他点保持不变。

对于 average pooling，在梯度反向传播的过程中，梯度会平均分配到每一个单元来进行梯度下降。

图 1.12详细描述了该过程。

1.3.4 softmax

在图片分类任务中，我们最终应得到的输出是图片属于某类的概率，但显然神经网络的输出不是概率的形式，因此我们需要通过适当操作将输出转化为概率的形式，这就是我们本节考虑的 softmax 操作。(图 1.13)

softmax 运用指数函数有两个好处，首先指数函数的输出均为正值，概率也为正值，具有

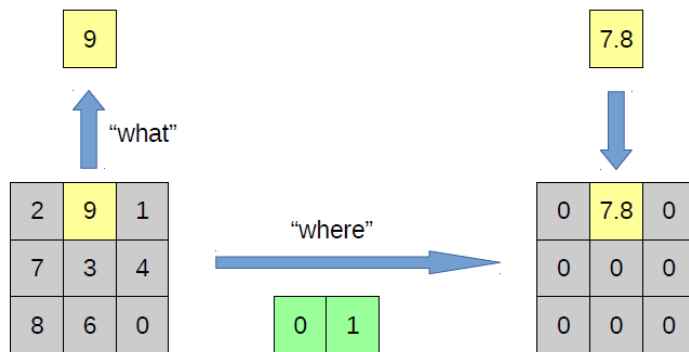


图 1.12: Gradient for pooling. 对于 max pooling, 图中最大值点位置在 (0,1) 处, 因此反向传播的梯度将全部用来更新 (0,1) 处的参数; 对于 average pooling, 每个单元将分配到反向传播梯度的 1/9。

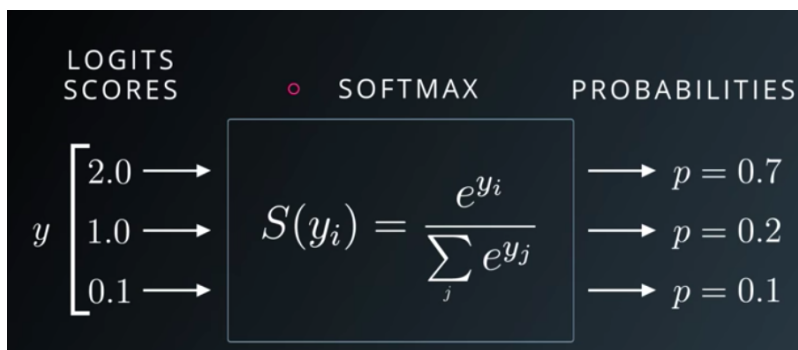


图 1.13: Softmax. 将输出转化为概率的形式。图片来源 <https://medium.com/data-science-bootcamp>。

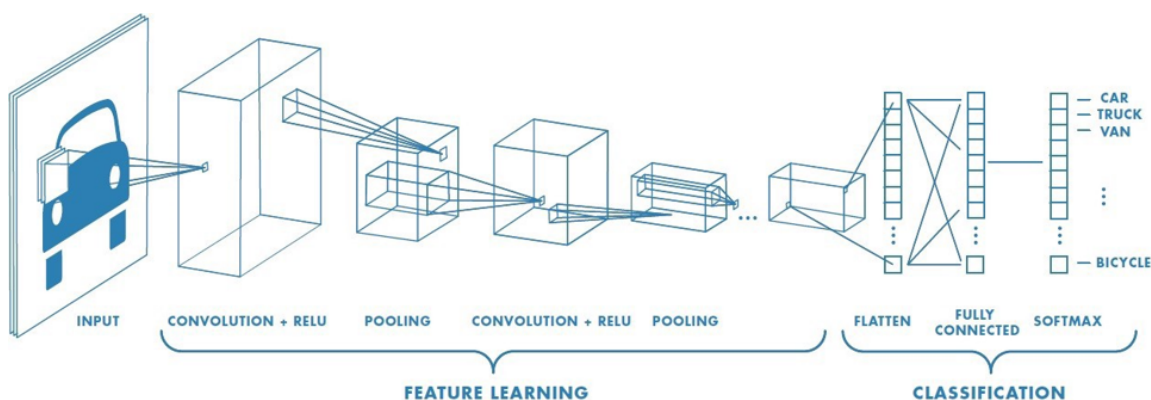


图 1.14: 卷积神经网络结构。图片来源Karn (2016)。

一致性；其次，指数函数斜率大，能够将不同的输入进一步拉开差距。

有了上边的这些运算，我们可以正式定义卷积神经网络 (CNN) 了，如图 1.14所示，假设我们输入是一张 RGB 图像，即有红、绿、蓝三个通道，我们用三个卷积核分别对三个通道做卷积，加上偏置项后得到了一层输出，我们再用另外的三个卷积核与图像做卷积，加上另一个偏置项后获得另一层输出，依此类推，我们最终得到了一个多通道的输出，再通过一个非线性函数 (ReLU) 输入到 Pooling，pooling 后通道数不变，每一片的大小变小，假设现在通道数为 n ，那么我们用 m 组，每组 n 个卷积核与之做卷积，获得一个 m 个通道的输出，再通过 ReLU 输入到新的 Pooling 层。重复以上操作数次后，最终得到的输出维数较小，我们把输出的数据展平 (flatten)，输入到一个全连接层，最后通过一个 softmax 获得分类的概率输出。

1.4 Understanding CNN

在本章的最后我们重新来理解一下 CNN 的机制。

首先我们要考虑的是 CNN 是否真的抓住了某些平移不变性，我们注意到，在 pooling 运算中，无论最大值点在所考虑区块的哪个位置，经过 pooling 运算后我们都能够将其输出，因此 pooling 中存在某种意义上的平移不变性。

其次我们考虑局部关联性，我们知道图片中的内容都具有一些局部关联，比如图片中大多数像素点附近的颜色都是相同的。而长存关联是比较少的，大多数情况下，随着距离提升，像素点之间所传达的信息关联性会变小。这与我们的卷积运算有一定的相似性，因为相比全连接而言，卷积也是一个局部的运算，相邻项之间共享参数，起到了一些局部关联的作用。

我们可以通过一张图片来进一步理解 CNN 的特性。(图 1.15)

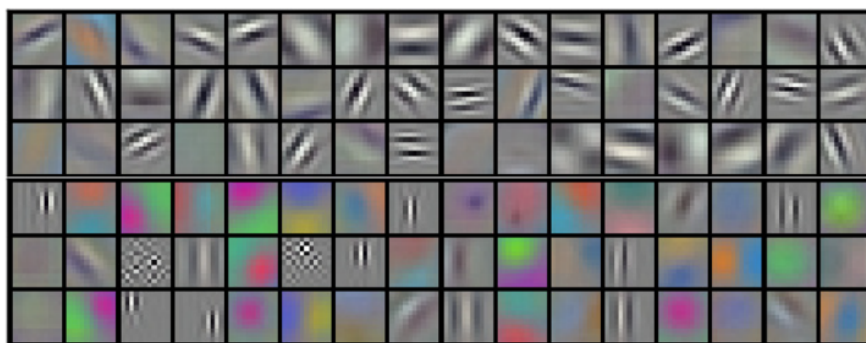


图 1.15: Understand CNN. 图中所示的是 96 组 $11 \times 11 \times 3$ 的卷积核与 $224 \times 224 \times 3$ 的图像做卷积得到的输出，从图中来看卷积运算确实学到了一些局部的方向性。图片来源Krizhevsky et al. (2012)

参考文献

- Dayan, P. & Abbott, L. F. (2005), *Theoretical neuroscience: computational and mathematical modeling of neural systems*, MIT press.
- Karn, U. (2016), ‘An intuitive explanation of convolutional neural networks’, *The data science blog* .
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* ‘Advances in neural information processing systems’, pp. 1097–1105.
- Ruderman, D. L. (1994), ‘The statistics of natural images’, *Network: computation in neural systems* **5**(4), 517.