



人工智能与深度学习

4-1 卷积运算

王中雷

经济学院、王亚南经济研究院、邹至庄经济研究院

厦门大学

(第一版)

内容摘要

1. 一维卷积
2. 填充 (Padding)
3. 步幅 (Stride)
4. 二维卷积
5. 例子
6. 多通道卷积

卷积神经网络背景

1. 当输入特征是图片时，为什么不考虑全连接神经网络

- 模型参数将变得非常巨大
 - ▷ 增加计算和存储负担
 - ▷ 容易造成过拟合
- 忽略了图片中的空间结构
- 通常不具有空间平移等变性

2. 针对图片处理，我们需要考虑一个新的模型框架（卷积神经网络）

卷积神经网络

1. 卷积运算的优势：

- 不同空间位置的卷积输出可以独立计算，因此适合并行化
- 在图片不同区域，卷积运算用同一个卷积核（kernel），所以，模型参数更少
- 卷积运算考虑到了图片的局部信息，所以该计算具有一定的平移等变性

2. 以卷积运算为核心的神经网络，称为卷积神经网络（Convolutional Neural Network, CNN）

卷积运算

1. 卷积运算并不陌生

- 在概率论中，卷积运算涉及到两个随机变量加和的密度函数： $Z = X + Y$

$$f_Z(z) = \int f_X(x)f_Y(z-x)\mu(\mathrm{d}x),$$

- ▷ X, Y : 两个独立的随机变量
- ▷ μ : 控制测度 (dominating measure)
- ▷ $f_\star(\cdot)$: 随机变量 $\star$ 对应的概率密度函数

2. 简单起见，记 $f(\cdot)$ 和 $g(\cdot)$ 两个函数的卷积为

$$h(z) = (f * g)(z) = \int f(x)g(z-x)\mu(\mathrm{d}x) = \int f(z-x)g(x)\mu(\mathrm{d}x)$$

- 函数 f 和 g 需使得上述卷积运算有意义

一维卷积

输入

3	6	5	4	8	9	1	7	9	10	10
---	---	---	---	---	---	---	---	---	----	----

卷积核

-1	0	1
----	---	---

计算

结果

2	-2	3	5	-7	-2	8	3	1
---	----	---	---	----	----	---	---	---

1. 我们通过翻转 $g(x)$ ，来得到一个卷积核 (kernel)，我们可用 `np.sum` 进行卷积运算
2. 本质上讲，其是两个向量的内积
3. 索引标签 x 的具体数值不太重要

填充 (Padding)

1. 直观

- 通过卷积运算，输入数据的尺寸在减小
- 图片边缘的信息未被充分利用

2. 填充就是为了解决上面的问题

- 人为地在数据边缘添加 0
- 边缘信息能够被充分利用
- 控制输出尺寸

填充 (Padding)

输入

0	3	6	5	4	8	9	1	7	9	10	10	0
---	---	---	---	---	---	---	---	---	---	----	----	---

卷积核

-1	0	1
----	---	---

计算

结果

6	2	-2	3	5	-7	-2	8	3	1	-10
---	---	----	---	---	----	----	---	---	---	-----

1. 输出与输入具有相同的长度

填充 (Padding)

1. 两种常见的填充方式:

- **Valid**: 不施加填充
- **Same**: 施加填充, 使得输出和输入的尺寸相同

2. 我们可以在数据边缘填充多圈零元素

- 不同的边缘, 可以添加不同数量的 0

3. 我们在进行卷积运算时, 也可以用大于 1 的步幅

- 降低输出特征的规模
- 平衡准确性和计算效率

步幅

输入

3	6	5	4	8	9	1	7	9	10	10
---	---	---	---	---	---	---	---	---	----	----

卷积核

-1	0	1
----	---	---

计算

结果

2	3	-7	8	1
---	---	----	---	---

1. 在上面的例子中，步幅（stride）为 2，也可以是其他正整数
2. 填充侧重于边界信息的利用程度，而步幅控制卷积的采样密度
3. 填充和步幅可以在同一个卷积运算中同时使用

空洞 (Dilated/Atrous) 卷积

输入

3	6	5	4	8	9	1	7	9	10	10
---	---	---	---	---	---	---	---	---	----	----

空洞卷积核

-1	0	1	0	1
----	---	---	---	---

计算

结果

10	7	4	12	2	8	18
----	---	---	----	---	---	----

1. 在上面的例子中，空洞率是 2

- 空洞率为 n 意味着，在卷积核元素之间插入 $(n - 1)$ 个 0

2. 空洞增加了卷积运算的感受野 (receptive field)，但不增加卷积核参数个数

权衡

1. 适当填充可以增加边界像素参与卷积计算的次数，并控制输出尺寸，但计算量和内存占用也越大
2. 步幅越大，计算更省，但输出更稀疏，细节可能丢得更多
3. 空洞率越大，感受野也就越大，但相邻位置的连续局部信息可能没那么紧密

二维卷积

原始序列							
填充 (Padding = 1)							
0	0	0	0	0	0	0	0
0	3	6	5	4	8	0	0
0	9	1	7	9	6	0	0
0	8	0	5	0	9	0	0
0	6	2	0	5	2	0	0
0	6	3	7	0	9	0	0
0	0	0	0	0	0	0	0

卷积核

-1	1
-1	1

结果

步幅 (Stride=2)

3	-1	4
17	11	6
12	2	6

均值核



$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

输出平滑

弱化局部噪声和尖锐信息



高斯核



$$\begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$



输出平滑

权重更加中心化

更少产生生硬模糊

锐化卷积



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



突出局部变化明显的位置
图片中物体的轮廓会更清楚

竖直方向Sobel核



$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

对左右变化敏感

检测物体的竖直边界



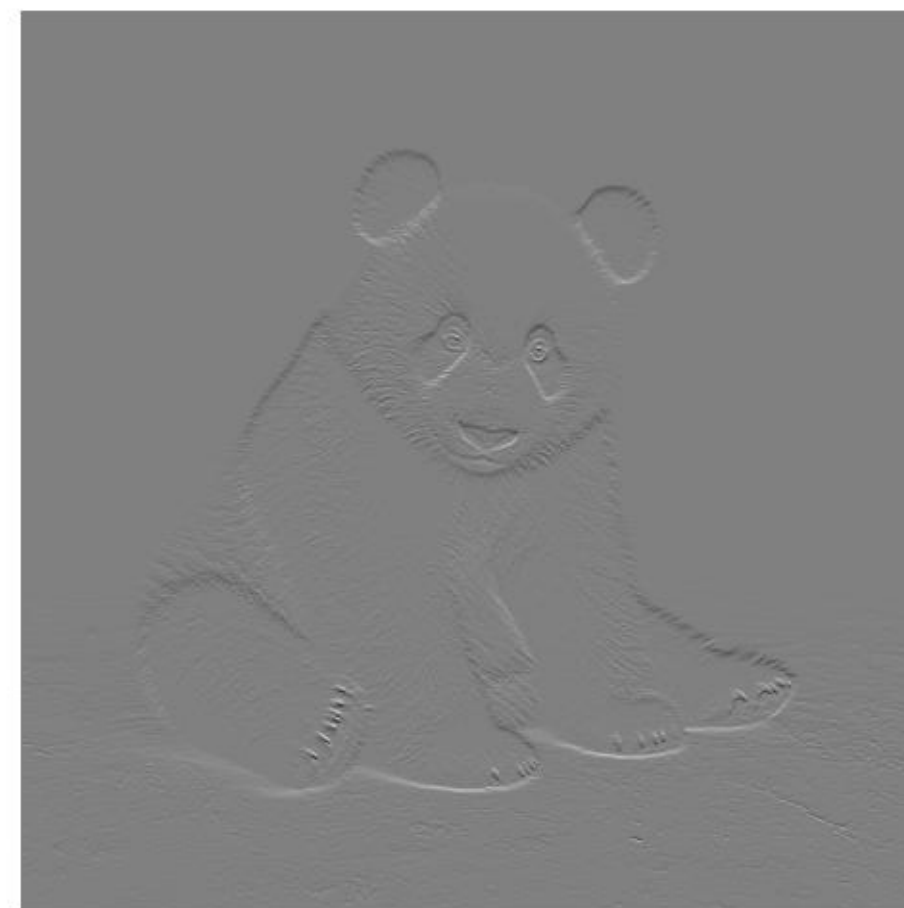
水平方向Sobel核



$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

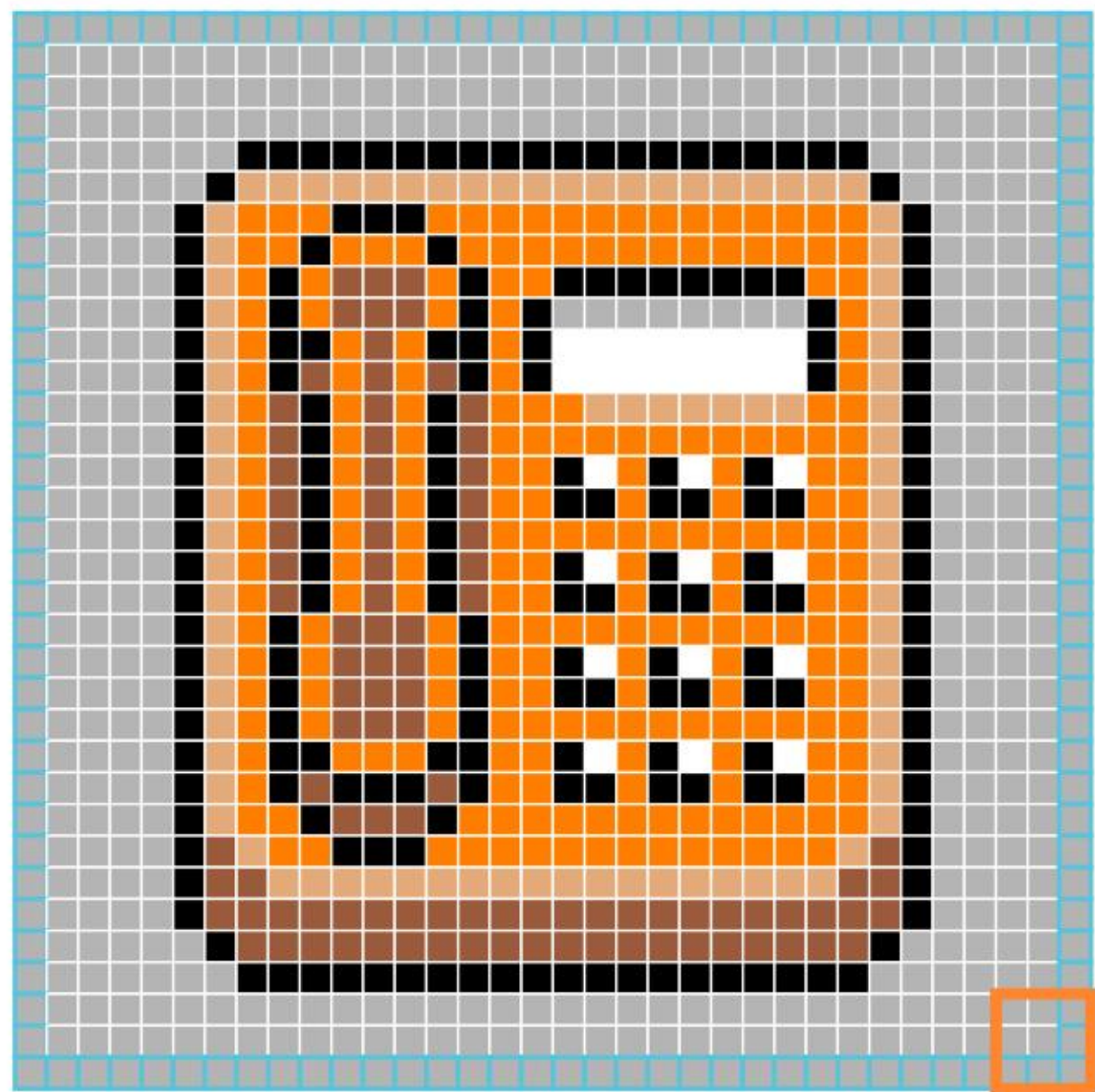
对上下变化敏感

检测物体的水平边界



另一个例子

原始图片



卷积核

-1	-1	-1
-1	9	-1
-1	-1	-1

锐化

-1	0	1
-2	0	2
-1	0	1

Sobel (竖直)

-1	-2	-1
0	0	0
1	2	1

Sobel (水平)



多通道卷积

1. 通常来讲，卷积核的通道数与输入“图片”的通道数相同

2. 输出图片尺寸的计算

- 输入图片尺寸: $d_C \times d_H \times d_W$
- 卷积核尺寸: $d_C \times f_1 \times f_2$
- 填充 (Padding): $p_1 \times p_2$ (默认 $p_1 = p_2 = p$)
- 步幅 (Stride): $s_1 \times s_2$ (默认 $s_1 = s_2 = s$)
- 空洞率: $d_1 \times d_2$ (默认 $d_1 = d_2 = d$)
- 输出图像尺寸: $d'_H \times d'_W$

$$d'_H = \left\lfloor \frac{d_H + 2p_1 - d_1(f_1 - 1) - 1}{s_1} + 1 \right\rfloor, \quad d'_W = \left\lfloor \frac{d_W + 2p_2 - d_2(f_2 - 1) - 1}{s_2} + 1 \right\rfloor$$

备注

1. 对于多通道卷积而言，一个卷积核对应的结果为一个矩阵
 - 在实际任务中，我们通常运用多个同规模的卷积核，进而得到一个多通道的张量结果
 - 不同的卷积核可用于提取不同类型的特征
2. 卷积神经网络训练的主要目标是学习卷积核及偏置等参数，使其能够提取与任务相关的特征
 - 以便针对特定问题，更加高效地提取图片信息

课程总结

1. 参数规模大幅减少：同一个卷积核的参数在不同空间位置之间共享
2. 高计算效率：卷积运算可用 GPU 进行并行
3. 有效利用空间特征：卷积运算专注于输入图片的局部区域
4. 平移等变性：同一个卷积核，被用于图片的不同区域