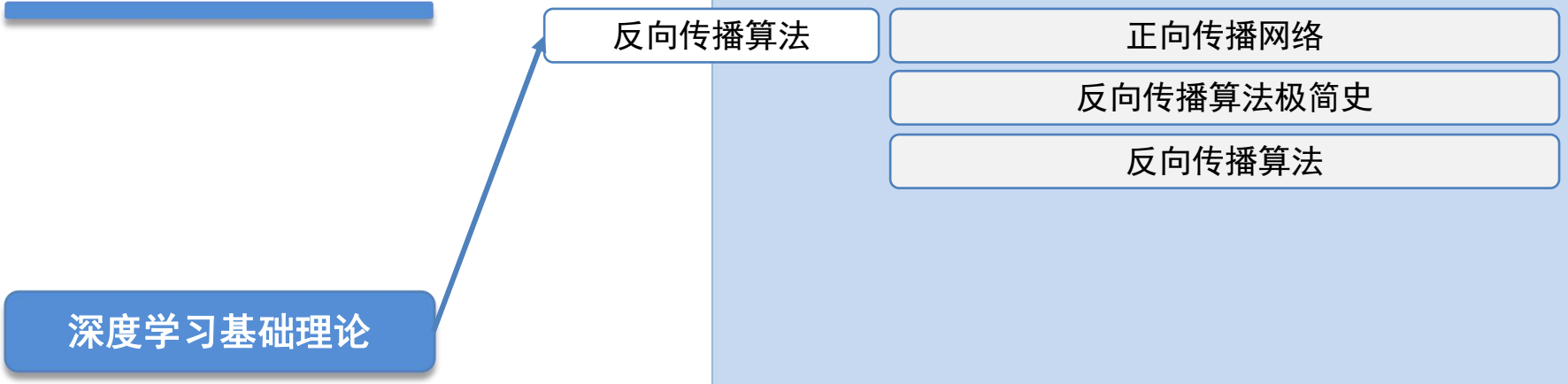
Tedu.cn
达内教育

深度学习

DEEP LEARNING

DAY02

深度学习基础理论



反向传播算法

正向传播网络

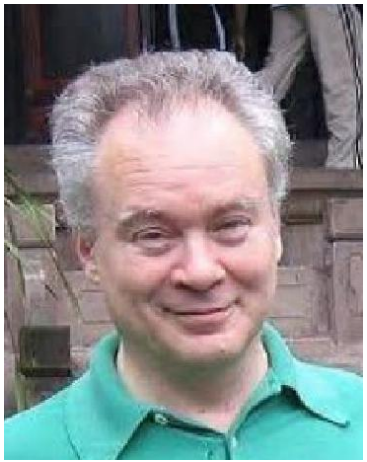
- 前一层的输出作为后一层的输入的逻辑结构，每一层神经元仅与下一层的神经元全连接，通过增加神经网络的层数虽然可为其提供更大的灵活性，让网络具有更强的表征能力，也就是说，能解决的问题更多，但随之而来的数量庞大的网络参数的训练，一直是制约多层神经网络发展的一个重要瓶颈。

反向传播算法极简史（一）

- 1974年，哈佛大学沃伯斯博士在他的博士论文中，首次提出了通过误差的反向传播来训练人工神经网络，以解决神经网络数量庞大的参数训练问题。但是，沃伯斯的工作并没有得到足够的重视，因为当时神经网络正陷入低潮，可谓“生不逢时”。
- 1986年，由杰弗里·辛顿（Geoffrey Hinton）和大卫·鲁姆哈特（David Rumelhart）等人在著名学术期刊Nature（自然）上发表了论文“借助误差反向传播算法的学习表征（Learning Representations by Back-propagating errors）”，系统而简洁地阐述了反向传播算法在神经网络模型上的应用。反向传播算法非常好使，它直接把纠错的运算量降低到只和神经元数目本身成正比的程度。

反向传播算法极简史（二）

- 后来，沃伯斯得到了IEEE（电气电子工程师学会）神经网络分会的先驱奖；杰弗里·辛顿与Yoshua Bengio、Yann LeCun（合称“深度学习三巨头”）共同获得了2018年的图灵奖

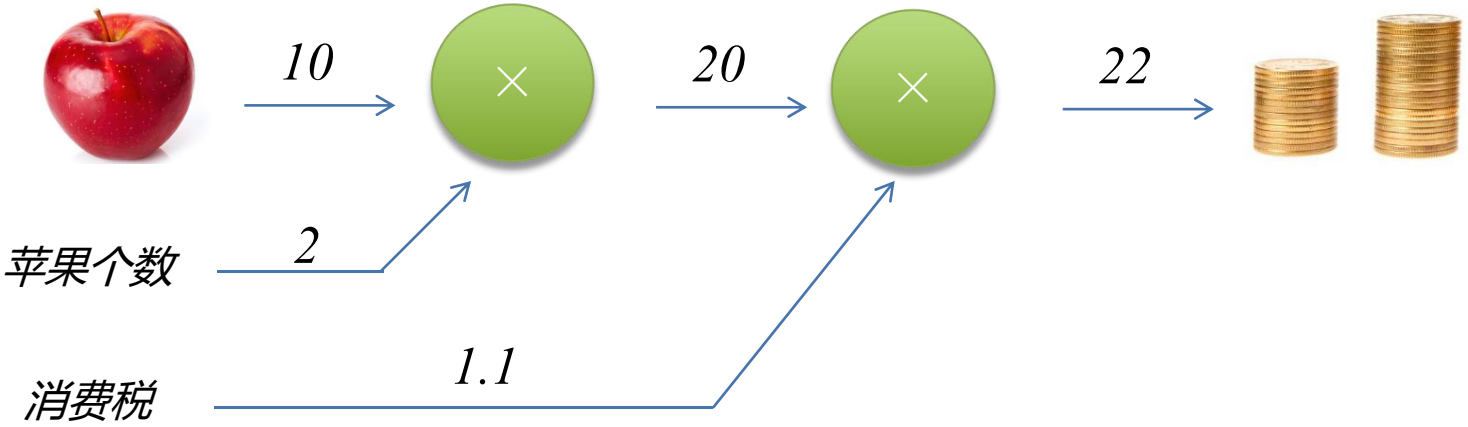


沃伯斯

杰弗里·辛顿

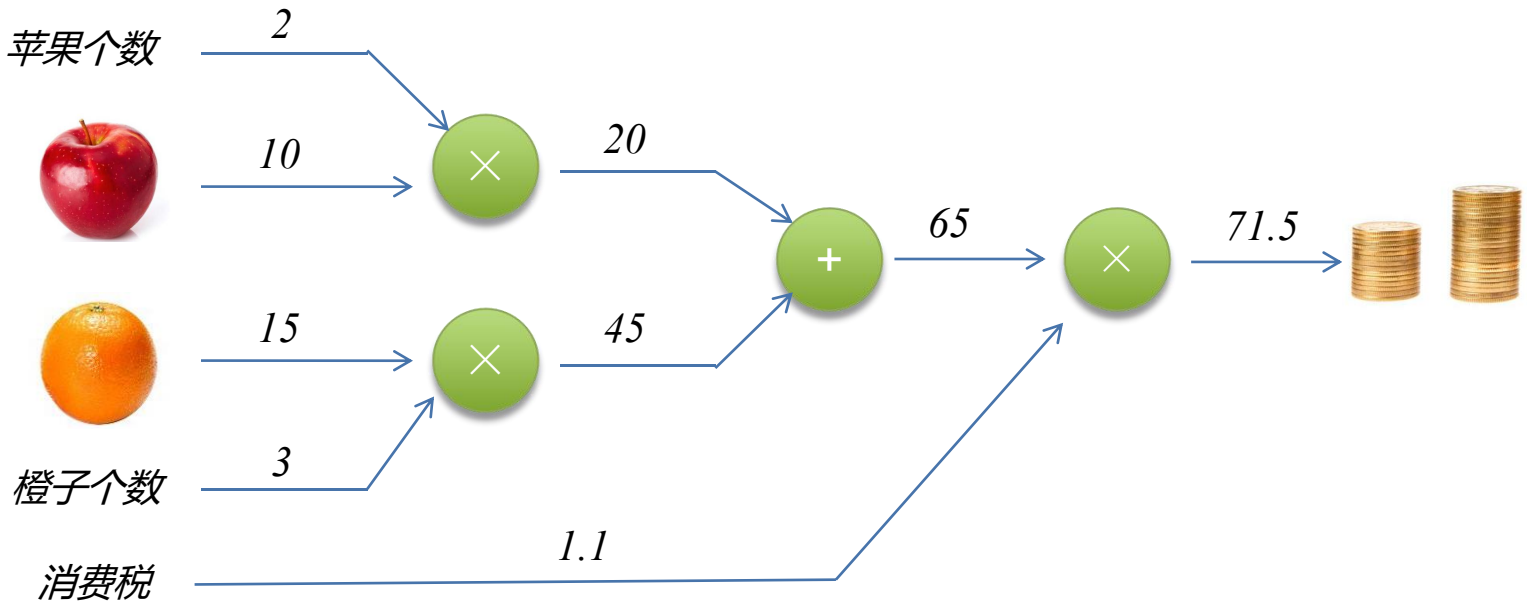
图解反向传播（一）

- 问题：Tom在超市买了2个苹果，每个10元，消费税10%，请计算应该支付的金额



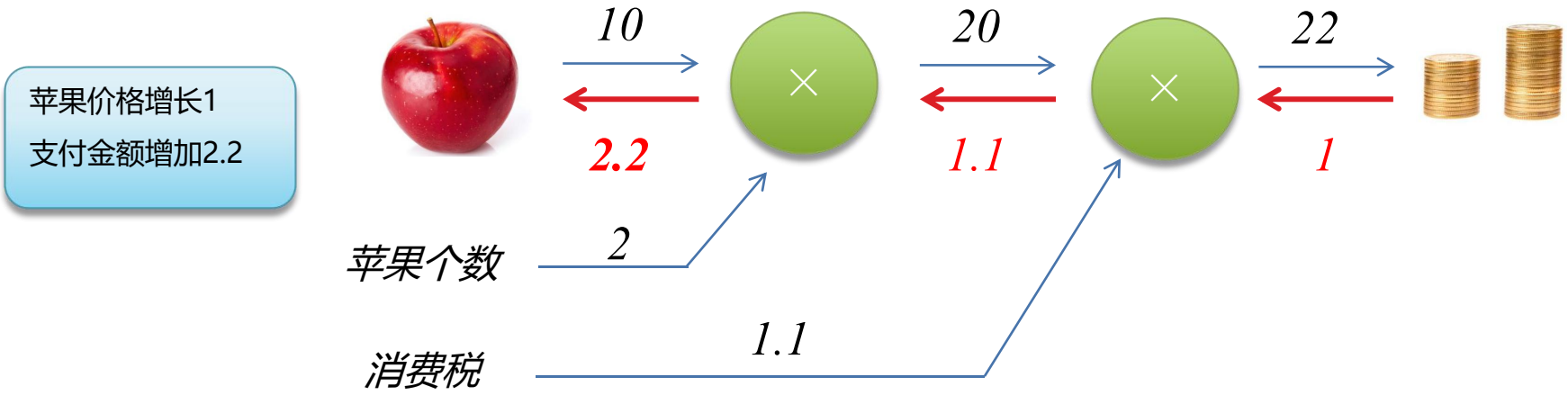
图解反向传播（二）

- 问题：Tom在超市买了2个苹果，3个橙子，其中苹果每个10元，橙子每个15元，消费税10%，请计算应该支付的金额



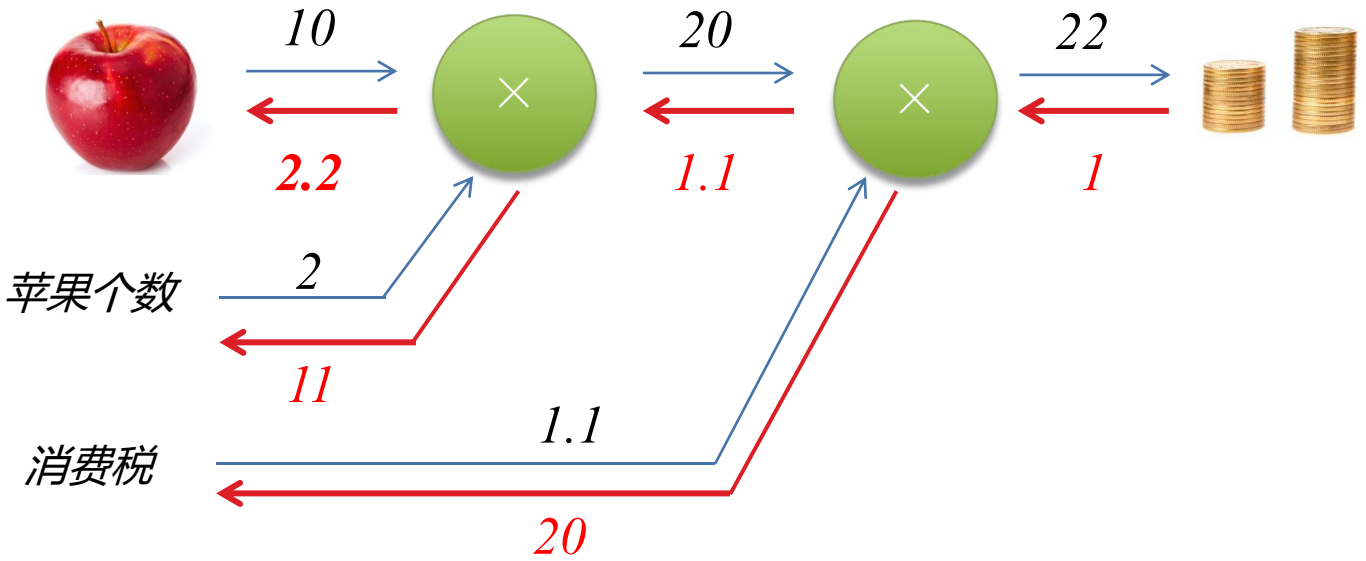
图解反向传播（三）

- 问题：Tom在超市买了2个苹果，每个10元，消费税10%，请计算苹果价格上涨会在多大程度上影响支付金额。设苹果的价格为x，支付金额为L，则相当于求 $\frac{\partial L}{\partial x}$ 。这个导数的值表示当苹果的价格稍微上涨时，支付金额会增加多少。



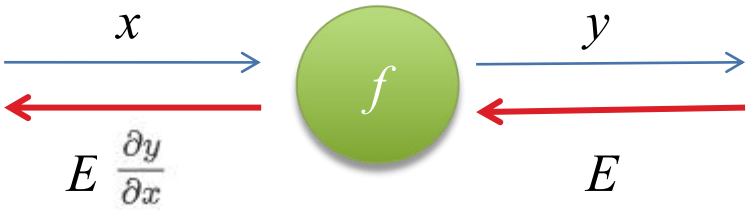
图解反向传播（四）

苹果价格的导数为2.2，苹果个数导数为11，消费税导数为20，可以解释为：苹果价格、苹果个数或消费税增加相同的值，分别对支付金额产生2.2倍、11倍、20倍的影响



反向传播计算

- 考虑函数 $y = f(x)$，输出为E，反向传播的计算顺序是，将信号E乘以节点的局部导数（偏导数），传递给前面的节点。



链式求导法则

- 考虑如下复合函数

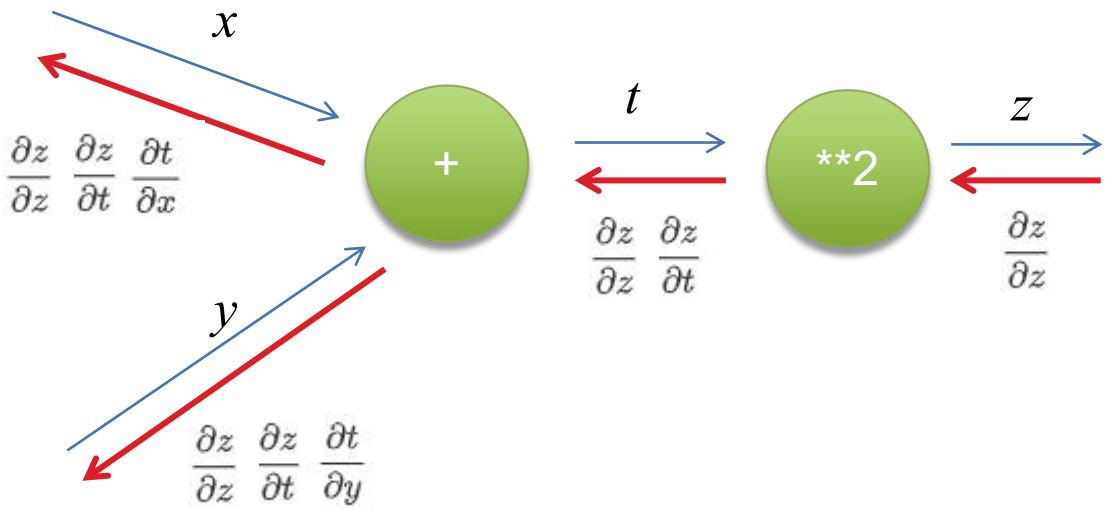
$$z = t^2$$
$$t = x + y$$

z 关于 x 的导数(x 变化对 z 的影响率)可以表示为：

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x}$$

这称之为“链式求导法则”

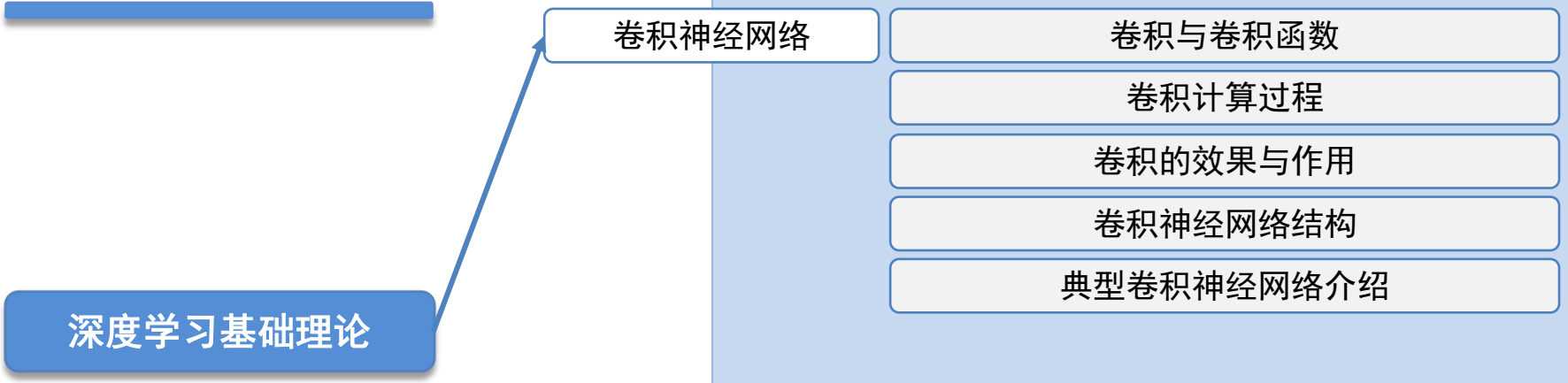
链式求导法则（续）


x
y
+
t
**2
z
$\frac{\partial z}{\partial z}$ $\frac{\partial z}{\partial t}$ $\frac{\partial t}{\partial x}$
$\frac{\partial z}{\partial z}$ $\frac{\partial z}{\partial t}$ $\frac{\partial t}{\partial y}$
$\frac{\partial z}{\partial z}$ $\frac{\partial z}{\partial t}$
$\frac{\partial z}{\partial z}$

小结

- 本章节主要介绍了反向传播算法，其目的是根据预测输出，调整权重参数，使得模型更快收敛。

深度学习基础理论


深度学习基础理论
卷积神经网络
卷积与卷积函数
卷积计算过程
卷积的效果与作用
卷积神经网络结构
典型卷积神经网络介绍

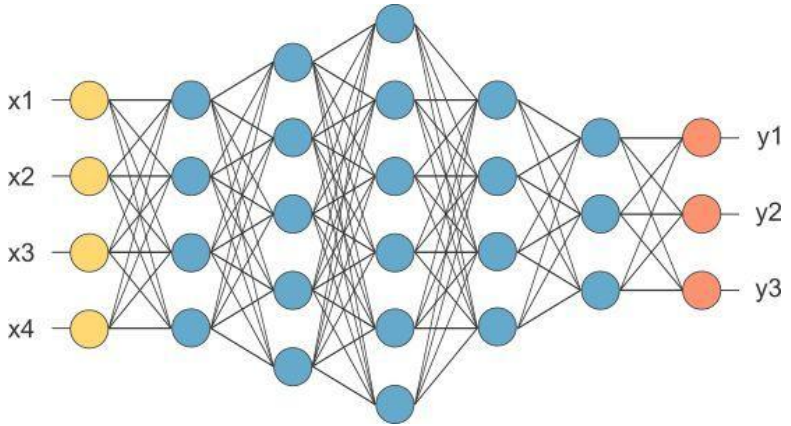
卷积神经网络

全连接神经网络的局限

- 之前介绍的神经网络，相邻层所有神经元之间都有连接，这称为全连接（fully-connected）。全连接神经网络有以下几个问题：

1）未考虑数据的“形状”，会破坏数据空间结构。例如，输入数据是图像时，图像通常是高长通道方向上的3维形状。但是，向全连接层输入时，需要将3维数据拉平为1维数据。

2）全连接网络层次深度受限，一般不超过七层。



卷积神经网络

- 卷积神经网络（Convolutional Neural Network，CNN）针对全连接网络的局限做出了修正，加入了卷积层（Convolution层）和池化层（Pooling层）。CNN被广泛应用于图像识别、语音识别等各种场合，在图像识别的比赛中，基于深度学习的方法几乎都以CNN为基础（比如，AlexNet、VGGNet、Google Inception Net及微软的ResNet等）上。近几年深度学习大放异彩，CNN功不可没。

什么是卷积

- “卷积”其实是一个数学概念，它描述一个函数和另一个函数在某个维度上的加权“叠加”作用。函数定义如下：

$$s(t)=\int_{-\infty}^{\infty} f(a)*g(t-a)\mathrm{d}a$$

其中，函数 f 和函数 g 是卷积对象，a 为积分变量，星号“*”表示卷积。公式所示的操作，被称为连续域上的卷积操作。这种操作通常也被简记为如下公式：

$$s(t)=f(t)*g(t)$$

离散卷积与多维卷积

- 一般情况下，我们并不需要记录任意时刻的数据，而是以一定的时间间隔（也即频率）进行采样即可。对于离散信号，卷积操作可用如下表示：

$$s(t) = f(t) \times g(t) = \sum_{a=-\infty}^{\infty} f(a)g(t-a)$$

当然，对于离散卷积的定义可推广到更高维度的空间上。例如，二维的公式可表示为公式：

$$s(i,j) = f(i,j) \times g(i,j) = \sum_{m} \sum_{n} f(m,n)g(i-m, j-n)$$

生活中的卷积

- 在一根铁丝某处不停地弯曲，假设发热函数是f(t)，散热函数是g(t)，此时此刻的温度就是f(t) 跟g(t) 的卷积
- 在一个特定环境下，发声体的声源函数是f(t)，该环境下对声源的反射效应函数是g(t)，那么在这个环境下感受到的声音就是f(t) 的和g(t) 的卷积
- 记忆也是一种卷积

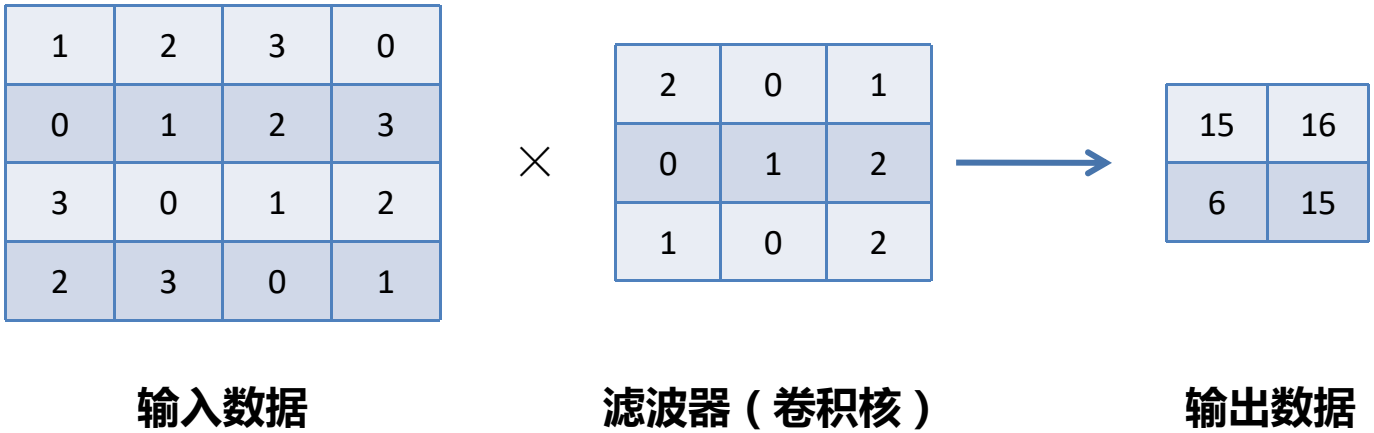
$$h_{记忆}(t)$$

$$=f_{认知}(t)*g_{遗忘}(t)$$

$$=\int_0^{+\infty} f_{认知}(\tau)\, g_{遗忘}(t-\tau)\, \mathrm{d}\tau$$

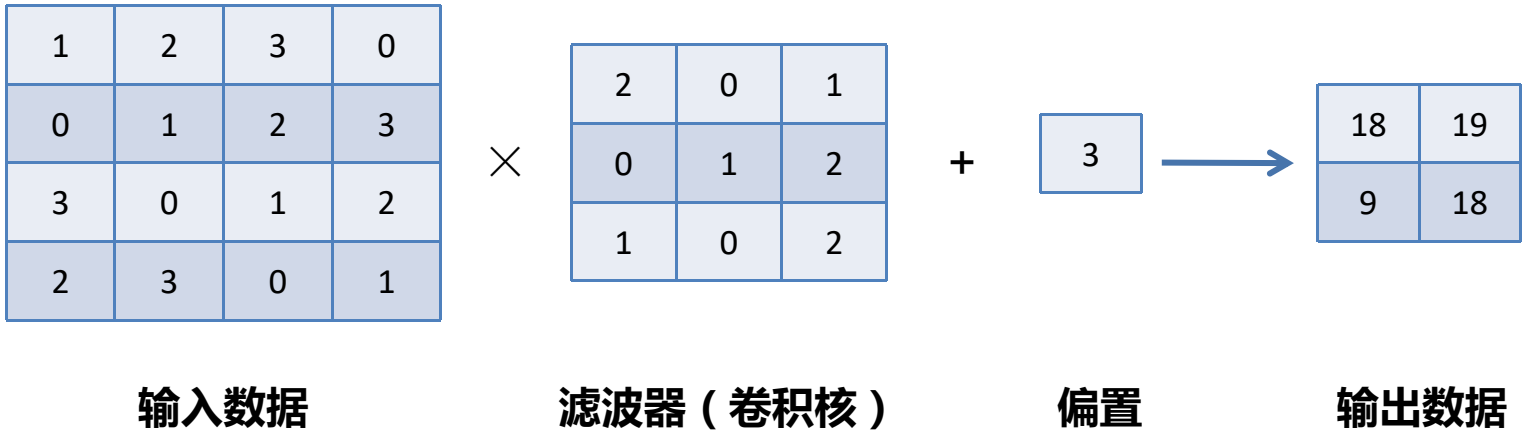
卷积运算（一）

- 以下是一个单通道、二维卷积运算示例



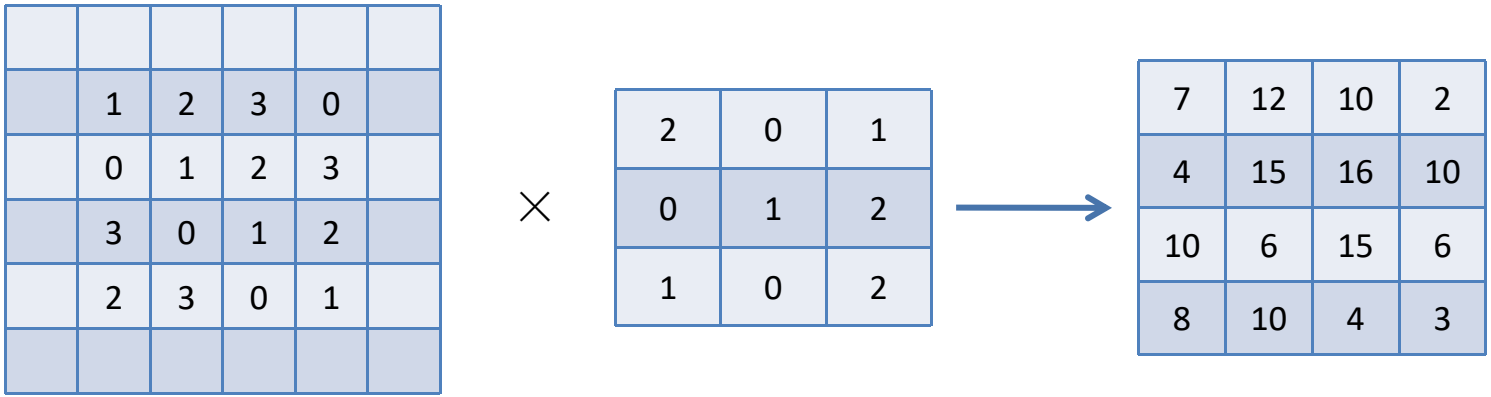
卷积运算（二）

- 以下是一个单通道、二维卷积加偏置值的运算示例



卷积运算（三）

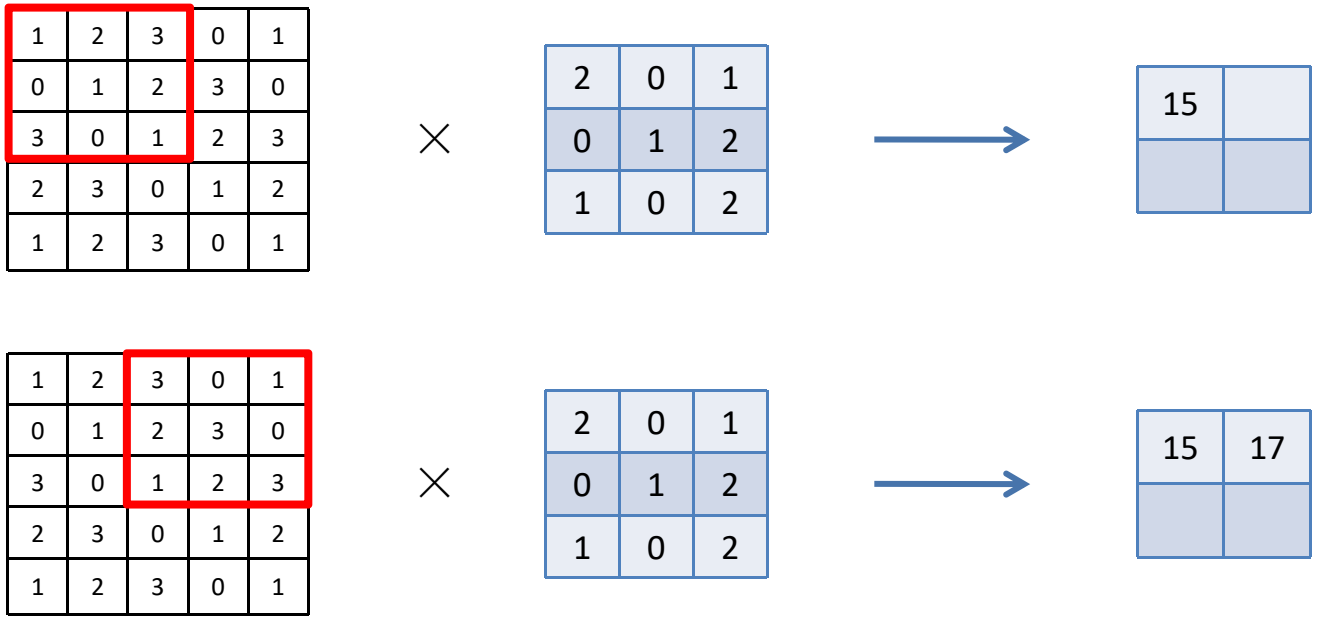
- 以下是一个带填充（padding）的单通道、二维卷积运算示例



输入数据（padding:1）　　**滤波器（卷积核）**　　**输出数据**

卷积运算（四）

- 以下是一个步幅（stride）为2的卷积运算示例



卷积运算（五）

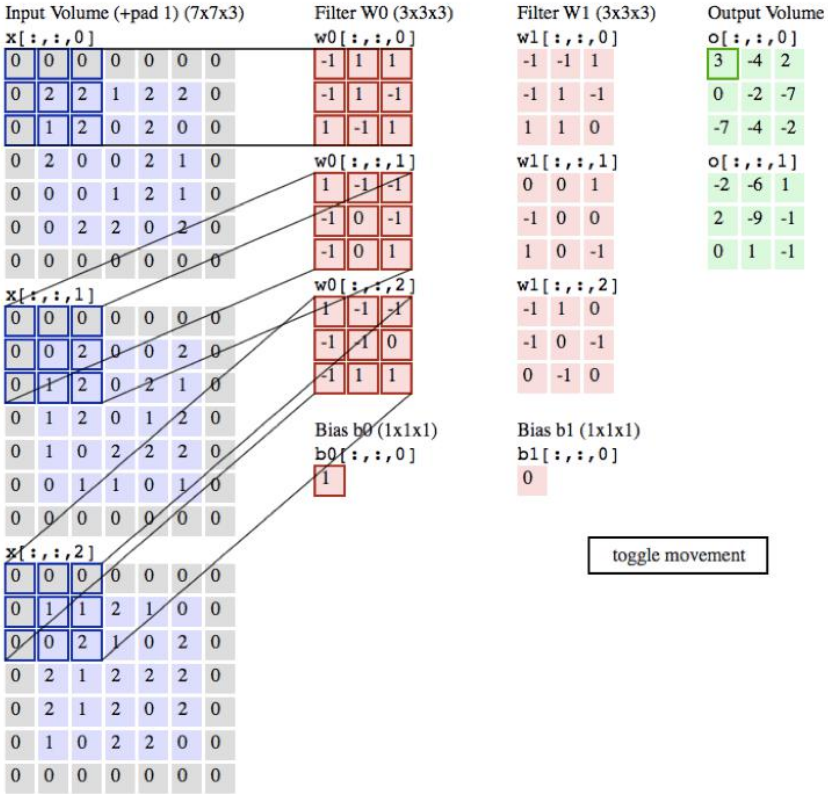
- 输出矩阵大小计算公式

$$OH = \frac{H + 2P - FH}{S} + 1 \qquad OW = \frac{W + 2P - FW}{S} + 1$$

其中，输入大小为(H, W)，滤波器大小为(FH, FW)，输出大小为(OH, OW)，填充为P，步幅为S。例如：输入大小（28,31）；填充2；步幅3；滤波器大小(5,5)，则输出矩阵大小为：

$$OH = \frac{28 + 2 \cdot 2 - 5}{3} + 1 = 10 \qquad OW = \frac{31 + 2 \cdot 2 - 5}{3} + 1 = 11$$

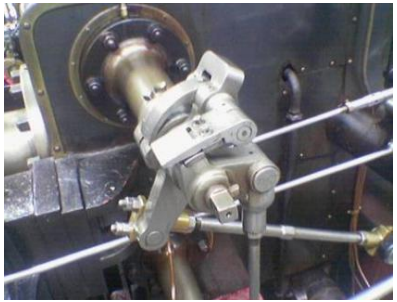
多通道卷积运算

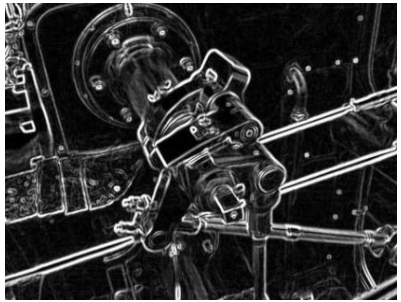

Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 2 2 1 2 2 0
0 1 2 0 2 0 0
0 2 0 0 2 1 0
0 0 0 1 2 1 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 0 2 0 0 2 0
0 1 2 0 2 1 0
0 1 2 0 1 2 0
0 1 0 2 2 2 0
0 0 1 1 0 1 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 1 1 2 1 0 0
0 0 2 1 0 2 0
0 2 1 2 2 2 0
0 2 1 2 0 2 0
0 1 0 2 2 0 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
-1 1 1
-1 1 -1
1 -1 1
w0[:,:,1]
1 -1 -1
-1 0 -1
-1 0 1
w0[:,:,2]
1 -1 -1
-1 1 0
-1 1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
-1 -1 1
-1 1 -1
1 1 0
w1[:,:,1]
0 0 1
-1 0 0
1 0 -1
w1[:,:,2]
-1 1 0
-1 0 -1
0 -1 0
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume
o[:,:,0]
3 -4 2
0 -2 -7
-7 -4 -2
o[:,:,1]
-2 -6 1
2 -9 -1
0 1 -1
toggle movement

卷积运算的效果（一）

- 通过卷积运算，能对输入数据起到加强或平滑效果。在图像处理中，通过选取合适的卷积核（或称算子），可以对图像进行锐化、去噪、模糊、加强边沿。

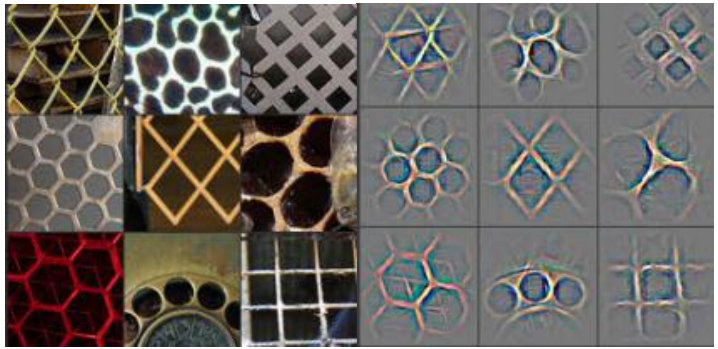
$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$





卷积运算的效果（二）

- 卷积运算能提取深层次复杂特征

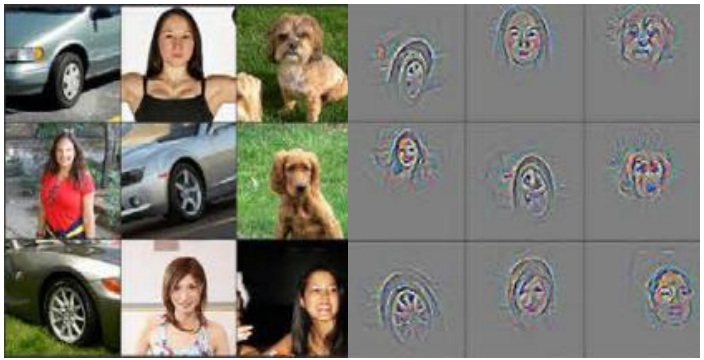


纹理相似性

复杂环境下提取主体

卷积运算的效果（三）

- 卷积运算能提取深层次复杂特征

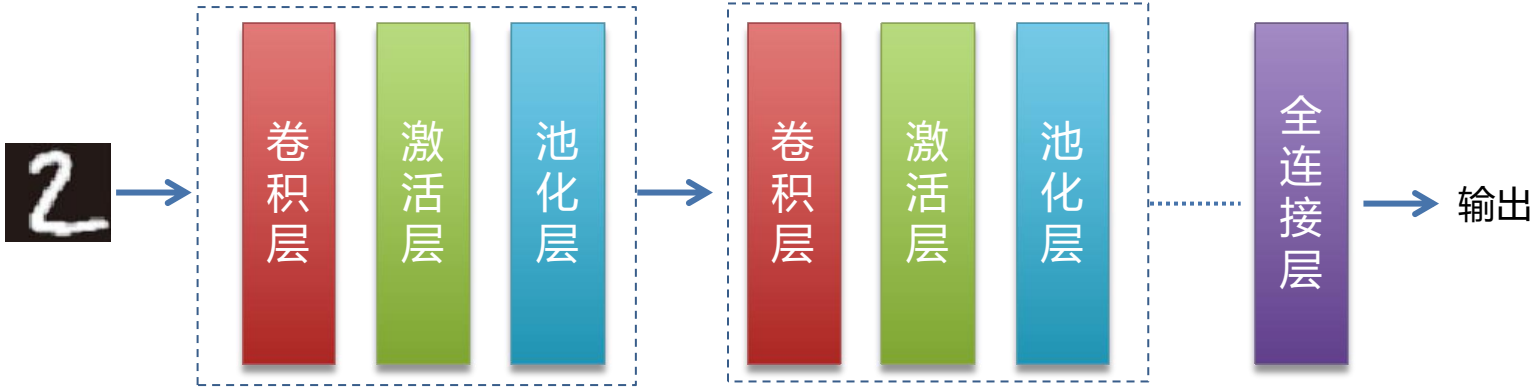


形态变化

无关场景，卷积效果集中于草地背景

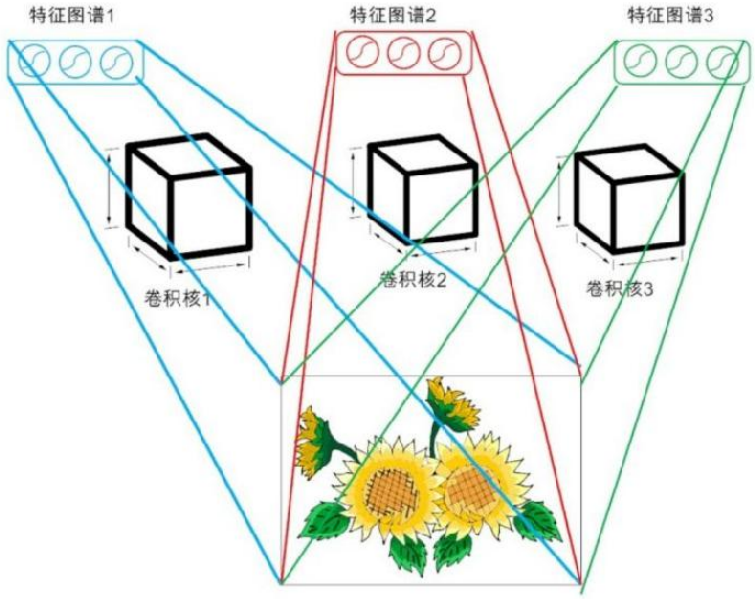
卷积神经网络结构

- 通常情况下，卷积神经网络由若干个卷积层（Convolutional Layer）、激活层（Activation Layer）、池化层（Pooling Layer）及全连接层（Fully Connected Layer）组成。



卷积层

- 它是卷积神经网络的核心所在，通过卷积运算，达到降维处理和提取特征两个重要目的

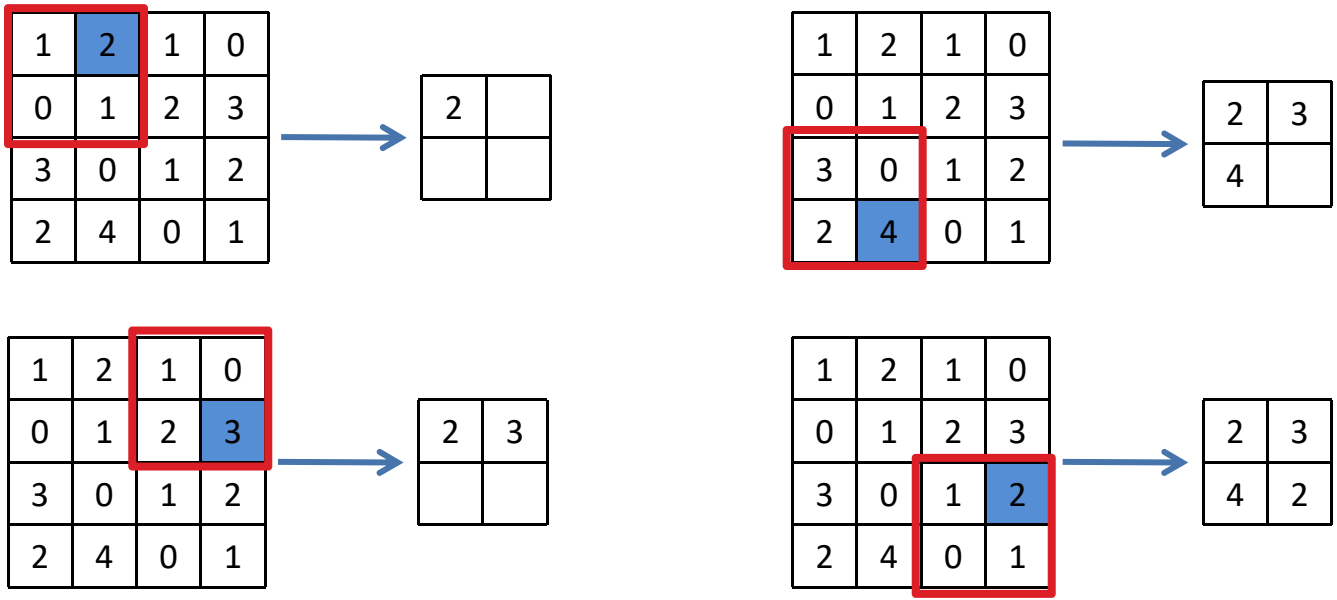


激活层

- 其作用在于将前一层的线性输出，通过非线性的激活函数进行处理，这样用以模拟任意函数，从而增强网络的表征能力。前面章节中介绍的激活函数，如挤压函数Sigmoid也是可用的，但效果并不好。在深度学习领域，ReLU（Rectified-Linear Unit，修正线性单元）是目前使用较多的激活函数，主要原因是它收敛更快，次要原因在于它部分解决了梯度消失问题。

池化层

- 也称子采样层或下采样层（Subsampling Layer），目的是**缩小高、长方向上的空间的运算，以降低计算量，提高泛化能力**。如下的示例，将4*4的矩阵缩小成2*2的矩阵输出

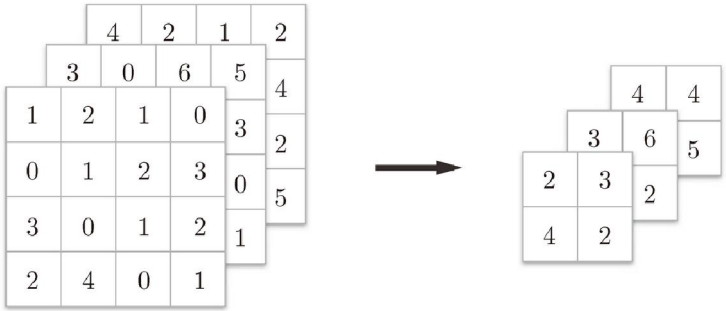


池化层的计算

- 对于每个输入矩阵，我们将其切割成若干大小相等的正方形小块，对每一个区块取最大值或者平均值，并将结果组成一个新的矩阵
- Max池化：对各个参与池化计算的区域取最大值，形成的新矩阵。在图像识别领域，主要使用Max池化
- Average池化：对各个参与池化计算的区域计算平均值

池化层的特征

- **没有要学习的参数。**池化层和卷积层不同，没有要学习的参数。池化只是从目标区域中取最大值（或者平均值），所以不存在要学习的参数
- **通道数不发生变化。**经过池化运算，输入数据和输出数据的通道数不会发生变化



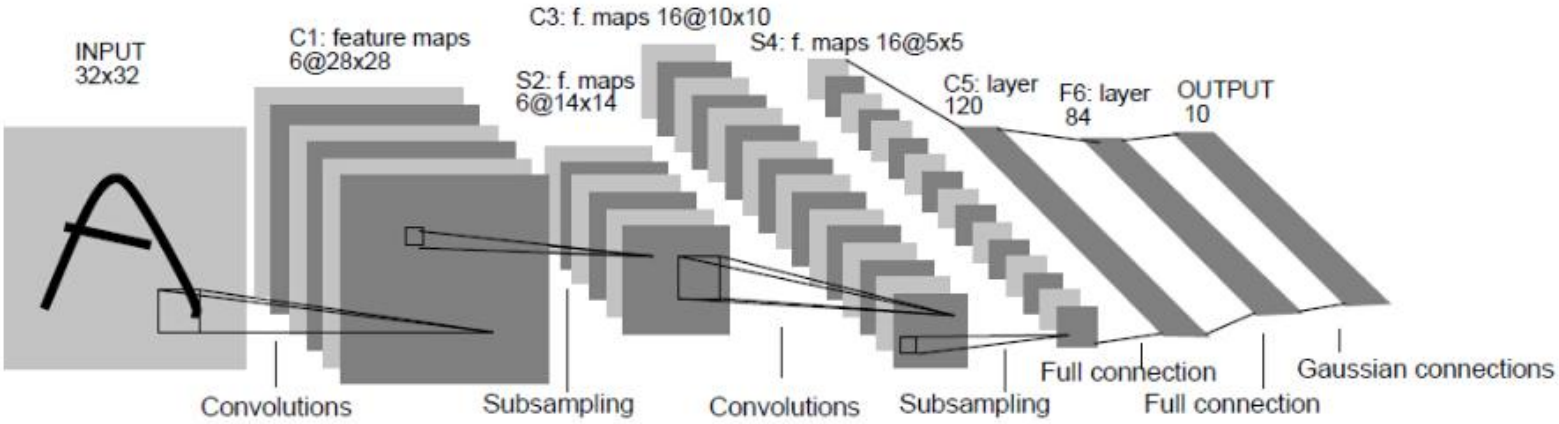
- **对微小的位置变化具有鲁棒性（健壮）。**输入数据发生微小偏差时，池化仍会返回相同的结果

全连接层

- 这个网络层相当于多层感知机（Multi-Layer Perceptron，简称MLP），其**在整个卷积神经网络中起到分类器的作用**
- 通过前面多个“卷积-激活-池化”层的反复处理，待处理的数据特性已有了显著提高：一方面，输入数据的维度已下降到可用传统的前馈全连接网络来处理了；另一方面，此时的全连接层输入的数据已不再是“泥沙俱下、鱼龙混杂”，而是经过反复提纯过的结果，因此输出的分类品质要高得多。

典型CNN示例（一）：LeNet

- LeNet是 Yann LeCun在1998年提出，用于解决手写数字识别的视觉任务。自那时起，CNN的最基本的架构就定下来了：卷积层、池化层、全连接层。



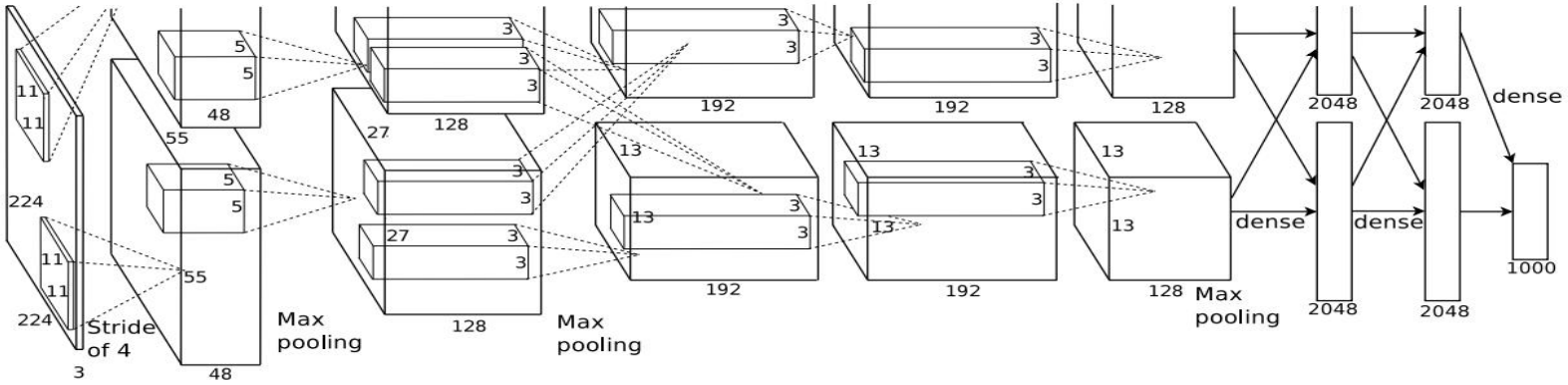
典型CNN示例（一）：LeNet（续）

- 输入：输入32*32大小单通道图像
- 两个“卷积-池化层”
- 第一个全连接层神经元数目为500，再接激活函数
- 第二个全连接层神经元数目为10，得到10维的特征向量，用于10个数字的分类训练，送入softmaxt分类，得到分类结果的概率

典型CNN示例（二）：AlexNet

- AlexNet是2012年ImageNet竞赛冠军获得者Hinton和他的学生Alex Krizhevsky设计的。AlexNet将LeNet的思想发扬光大，把CNN的基本原理应用到了很深很宽的网络中。其特点有：
  - ✓ 使用ReLU作为CNN的激活函数，并验证其效果在较深的网络超过了Sigmoid，成功解决了Sigmoid在网络较深时的梯度弥散问题
  - ✓ 使用Dropout（丢弃学习）随机忽略一部分神经元防止过拟合。在AlexNet中主要是最后几个全连接层使用了Dropout
  - ✓ 在CNN中使用重叠的最大池化。此前CNN中普遍使用平均池化，AlexNet全部使用最大池化，避免平均池化的模糊化效果
  - ✓ 提出了LRN（Local Response Normalization，局部正规化）层，对局部神经元的活动创建竞争机制，使得其中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，增强了模型的泛化能力
  - ✓ 使用CUDA加速深度卷积网络的训练，利用GPU强大的并行计算能力，处理神经网络训练时大量的矩阵运算

典型CNN示例（二）：AlexNet（续）



- AlexNet网络包含8层，其中前5层为卷积-池化层，后3层为全连接层；输入224×224×3的图像，第一卷积层用96个11×11×3的卷积核对进行滤波，步幅4像素；全连接的每层有4096个神经元，最后一个完全连接的层的输出被馈送到1000路SoftMax，它产生超过1000个类别标签的分布；整个网络共650000个神经元

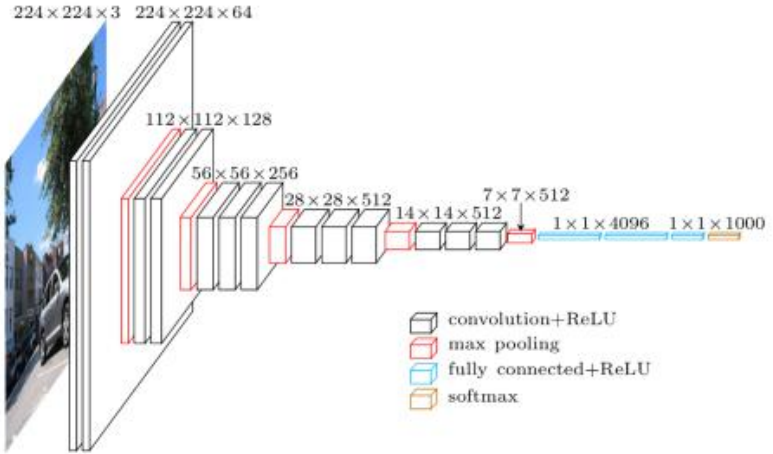
参考论文：http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

典型CNN示例（三）：VGGNet

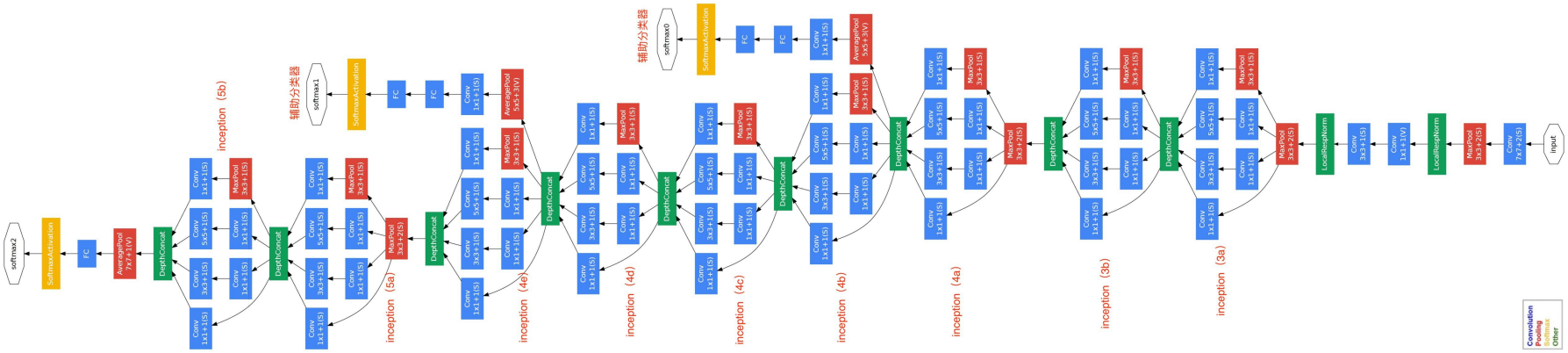
- VGG是Visual Geometry Group, Department of Engineering Science, University of Oxford（牛津大学工程科学系视觉几何组）的缩写，2014年参加ILSVRC（ImageNet Large Scale Visual Recognition Challenge）2014大赛获得亚军（当年冠军为GoogLeNet，但因为VGG结构简单，应用性强，所以很多技术人员都喜欢使用基于VGG的网络）

典型CNN示例（三）：VGGNet（续）

- 主要参数：
  - ✓ 网络深度：16~19层
  - ✓ 5组卷积-池化层，3个全连接层
  - ✓ 三个全连接层，前两层都有4096通道，第三层共1000路及代表1000个标签类别；最后一层为softmax层
  - ✓ 所有卷积层有相同的配置，即卷积核大小为3x3，步长为1，填充为1



典型CNN示例（四）：GoogLeNet



请参考论文：https://arxiv.org/pdf/1409.4842.pdf

小结

- 本章节介绍了卷积神经网络（CNN），CNN是深度学习的主要模型，在解决复杂工程问题中表现出了良好的性能。卷积神经网络主要由以下几层构成：
  - ✓ **卷积层。**执行卷积运算
  - ✓ **激活层。**对卷积结果执行激活函数运算
  - ✓ **池化层。**降低数据规模，防止过拟合
  - ✓ **全连接层。**执行输出计算

今日总结

- 反向传播算法
- 卷积神经网络