

Presentation

1 CNN (Convolutional Neural Networks) 卷积神经网络

1.1 背景

卷积神经网络 (Convolutional Neural Network, CNN) 是一种在计算机视觉领域取得了巨大成功的深度学习模型。它们的设计灵感来自于生物学中的视觉系统，旨在模拟人类视觉处理的方式。

1.2 卷积

是什么，为什么，怎么卷

1.2.1 卷积定义

百度：卷积、旋积或褶积(英语：Convolution)是通过两个函数f和g生成第三个函数的一种数学运算，其本质是一种特殊的积分变换，表征函数f与g经过翻转和平移的重叠部分函数值乘积对重叠长度的积分。（源于：百度百科）

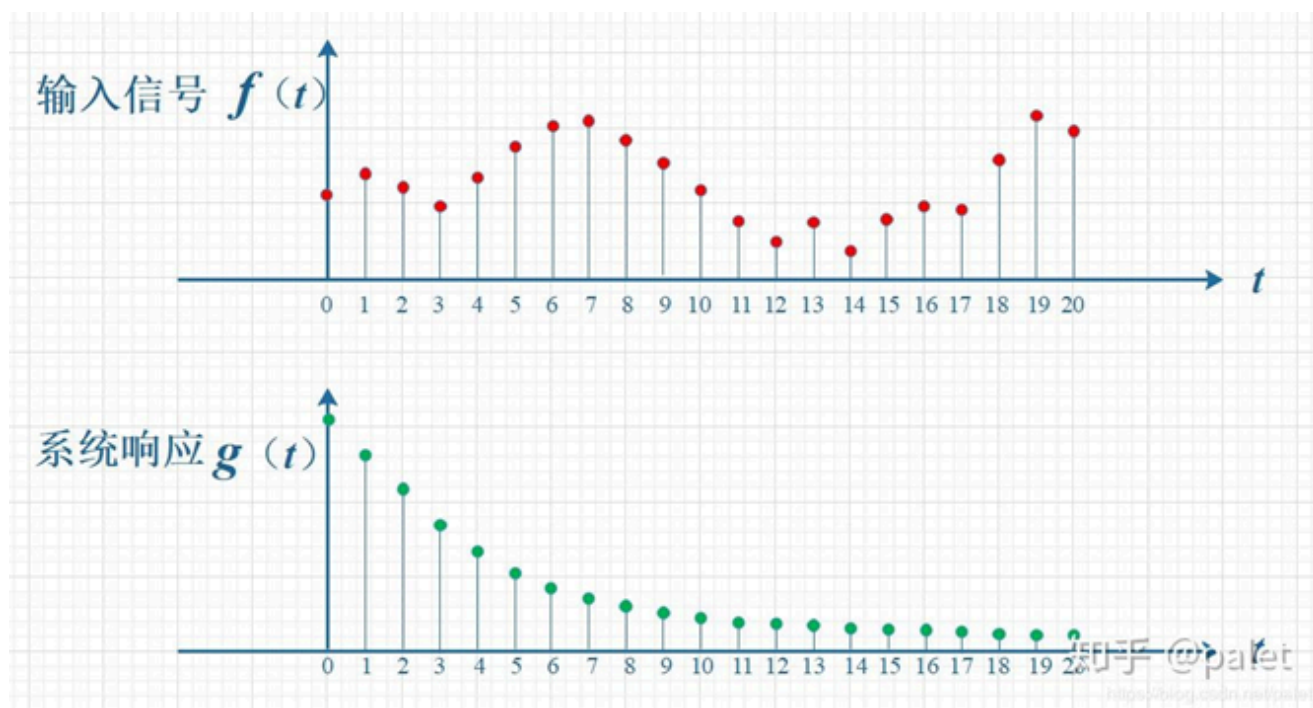
连续LTI系统下的卷积公式：

$$y(t) = \int_{-\infty}^{\infty} x(p)h(t-p) dp = x(t) * h(t)$$

离散LTI系统下的卷积公式：

$$y(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) = x(n) * h(n)$$

1.2.2 Example:



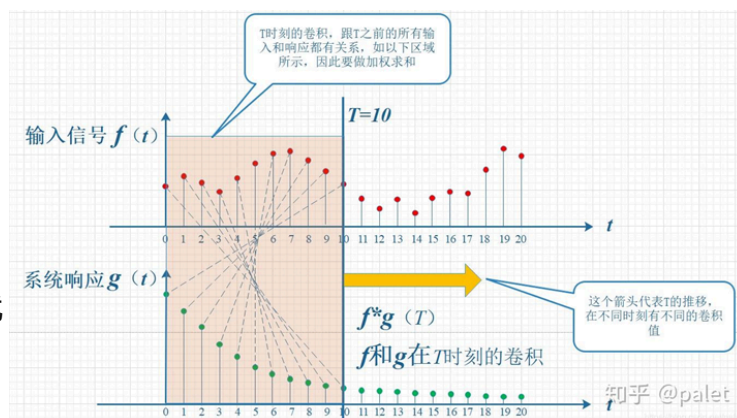
如上图所示，输入信号是 $f(t)$ ，系统响应函数是 $g(t)$ 。

其物理意义是，如果在 $t=0$ 的时刻有一个输入 $f(0)$ ，那么随着时间的增加，不断衰减，或者说到了 $t=T$ 的时刻，原来在 $t=0$ 时刻的输入 $f(0)$ 的值将衰减为 $f(0)g(T)$

$$\begin{aligned} \text{Sum}(10) &= \sum_{i=1}^{10} f(i)g(10-i) \\ &= f(1)g(9) + f(2)g(8) + \cdots + f(9)g(1) + f(10)g(0) \end{aligned}$$

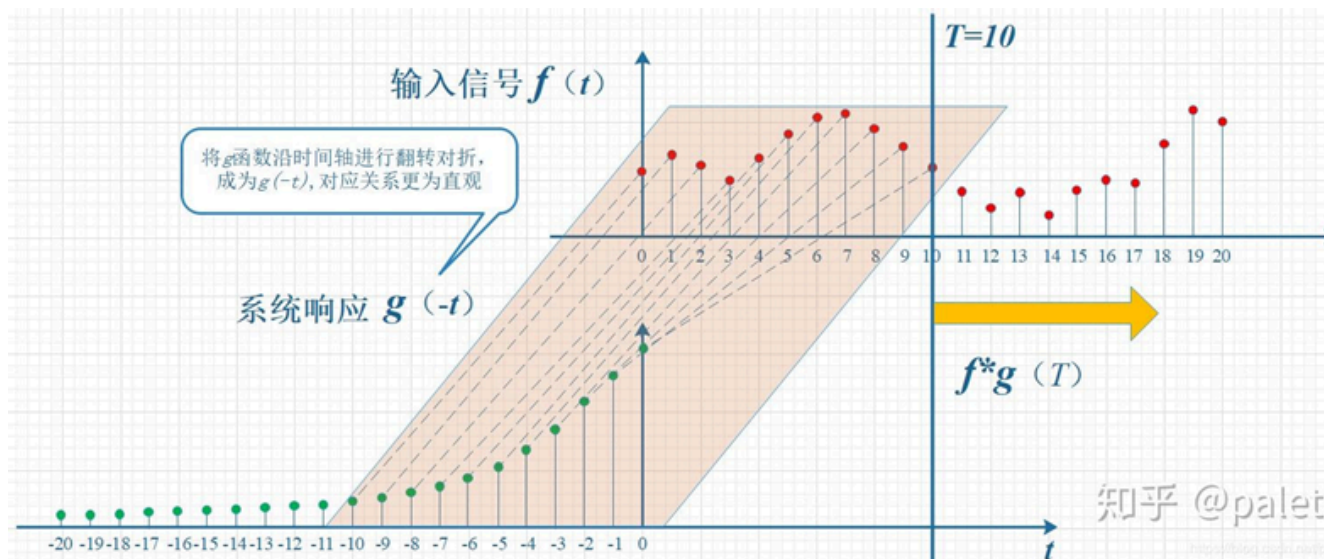
如果信号是连续输入的，在 $T=10$ 的时刻，输出的结果是多少？

$f(10)$ 因为是刚输入的，所以其输出结果应该是 $f(10)g(0)$ ，因此 $t=9$ 的输入 $f(9)$ 因为比 $f(10)$ 提前一个时间单位输入，所以对的系统响应变为 $g(1)$ ，产生的输出就为 $f(9)g(1)$ 。以此类推，这些点对应相乘之后累加，就是 $T=10$ 时刻的输出信号值，最终结果就是 f 和 g 两个函数在 $T=10$ 时刻的卷积值。



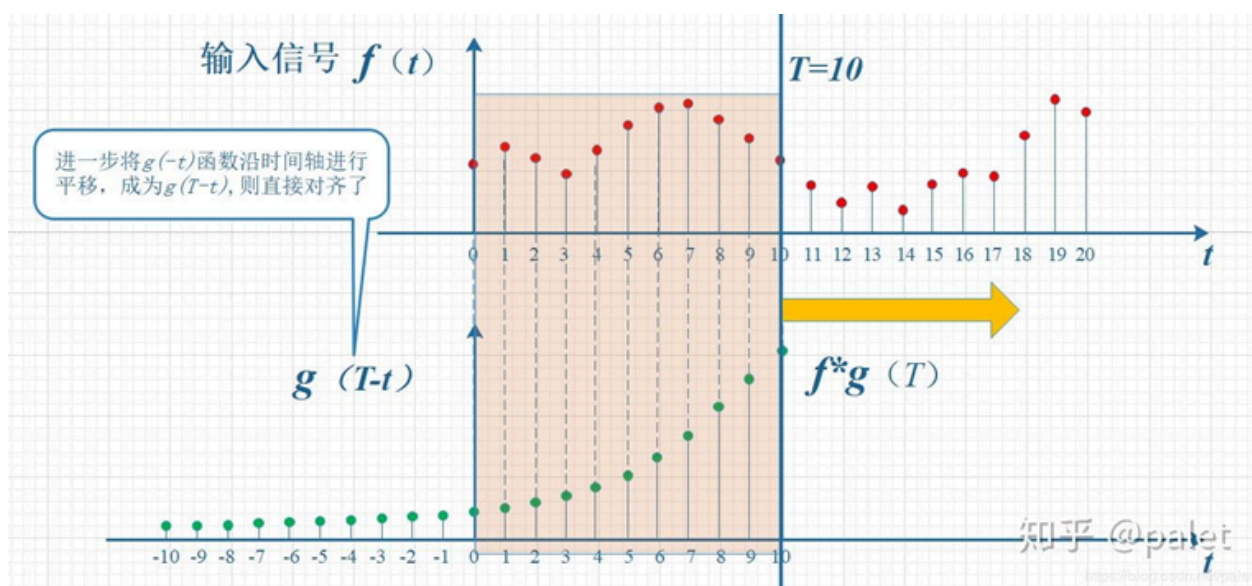
$$f * g(10) = \sum_{i=1}^{10} f(i)g(10-i)$$

- 翻转 - ‘卷’



- 平移 - 平移T个单位后重叠

$$g(T-t) = g[-(t-T)]$$



$$f * g(10) = \sum_{i=1}^{10} f(i)g(10-i)$$

1.2.3 卷神经网络上的卷积

上面介绍了数学上的卷积，本质就是翻转，平移，相乘，叠加的一个过程。那在卷神经网络上的卷积有何不同？

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \quad g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

↓ 翻转

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \quad \begin{bmatrix} \Gamma^T \Gamma q & 0^T \Gamma q & \Gamma^{-T} \Gamma q \\ \Gamma^0 q & 0^0 q & \Gamma^{-0} q \\ \Gamma^T \Gamma^{-q} & 0^T \Gamma^{-q} & \Gamma^{-T} \Gamma^{-q} \end{bmatrix} = \delta$$

$$c_{1,1} = a_{0,0}b_{1,1} + a_{0,1}b_{1,0} + a_{0,2}b_{1,-1} + a_{1,0}b_{0,1} + a_{1,1}b_{0,0} + a_{1,2}b_{0,-1} + a_{2,0}b_{-1,1} + a_{2,1}b_{-1,0} + a_{2,2}b_{-1,-1}$$

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

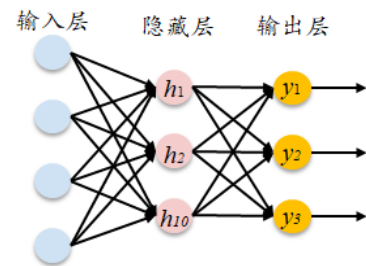
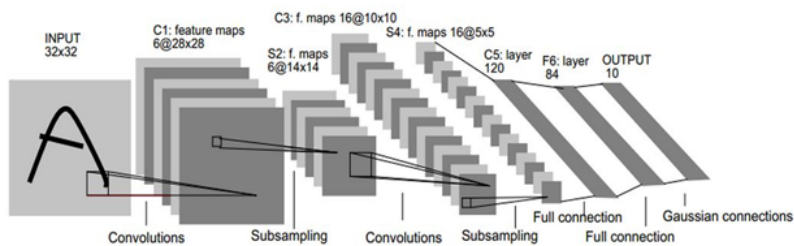
Convolved
Feature

- 在数学中的卷积，像上面的信号系统中，是为了处理信号而定义的一个运算，所以“翻转”是根据问题的需要而进行设定的
- 卷神经网络中的卷积，目的是为了提取图像特征，实际上只是用了加权求和的特点。但是更重要的是卷神经网络中的卷积核，当中的参数不是给定的，是需要根据实际数据训练得出的。

1.3 神经网络

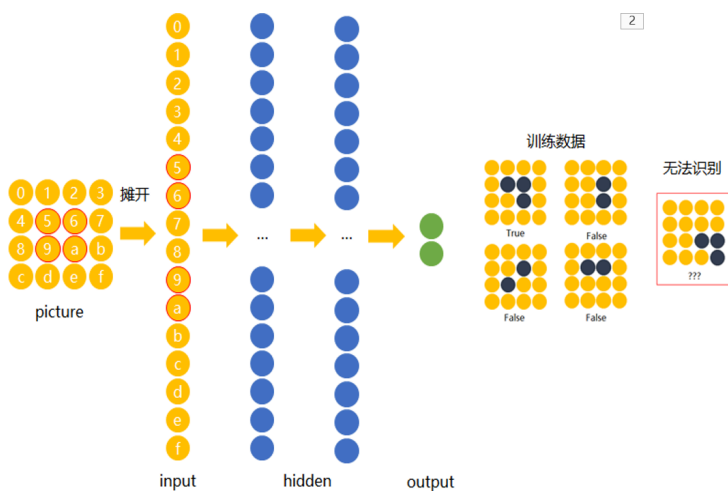
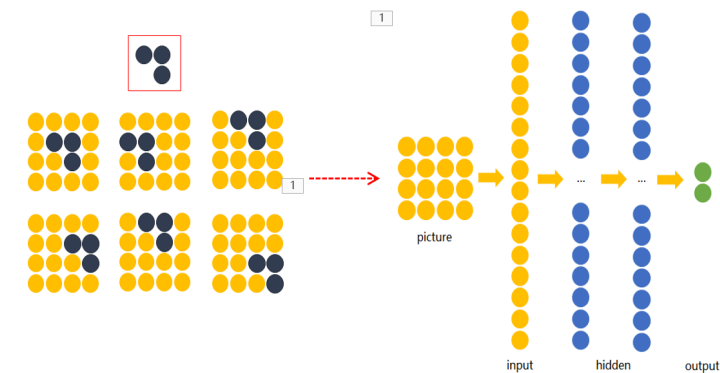
1.3.1 Compare

- Deep neural learning VS Shallow neural learning



1.3.1.1 BP神经网络(浅神经网络)

用BP神经网络识别图中的“横折”



- 为什么无法识别其他位置上的横折？
 - BP神经网络自身的全连接结构导致无法处理平移旋转等变换
- BP神经网络的缺陷
 1. 不能移动
 2. 不能变形
 3. 计算量大

- 解决办法
 1. 大量物体位于不同位置的训练数据(计算量大)
 2. 增加网络的隐藏层个数
 3. 使用卷积神经网络(CNN)

1.3.1.2 CNN(深度神经网络)

卷积神经网络大致流程:

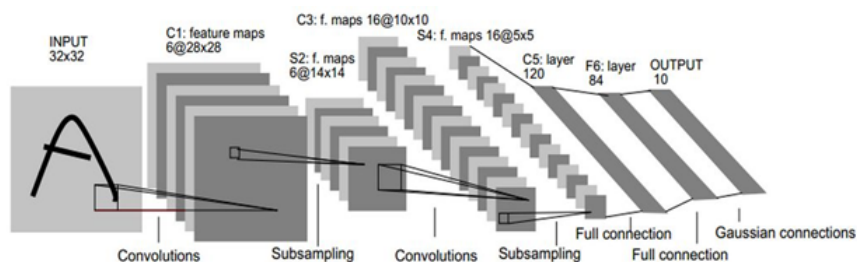
Convolutional layer(卷积)

ReLU layer(非线性映射)

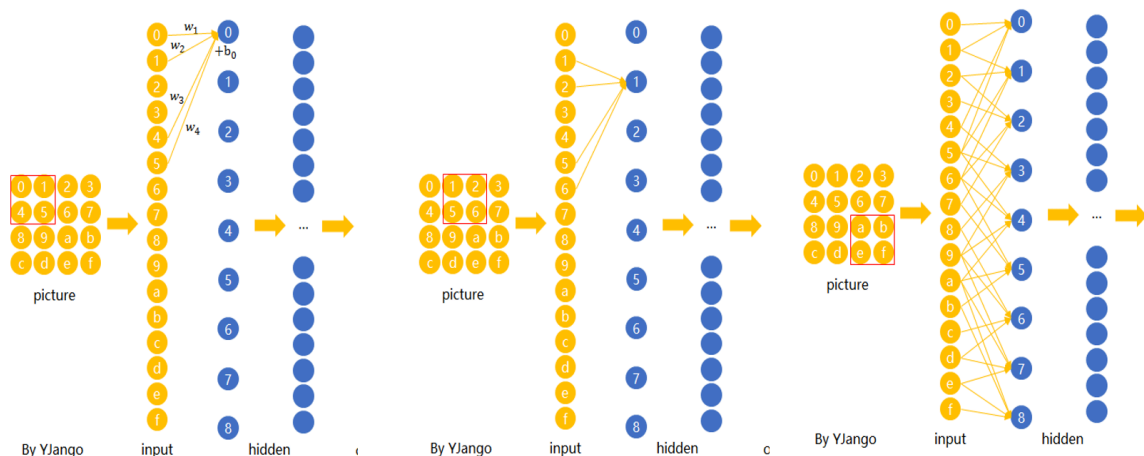
Pooling layer(池化)

Fully connected layer(全连接)

Output(输出)的组合



在CNN中, 先选择一个局部区域 (filter / kernel), 用这个局部区域去扫描整张图片。局部区域所圈起来的所有节点会被连接到下一层的一个节点上。



1.3.1.2.1 权值共享

权值共享是一种减少模型参数的方法, 在卷积神经网络 (CNN) 中, 同一个卷积核 (滤波器) 的参数在整个输入图像的不同区域被重复使用。这种方法可以显著降低计算复杂度

比如,你去一家连锁咖啡厅,点一杯美式,它的味道几乎是一模一样的。这是因为所有的咖啡师 (神经网络的神经元)都使用同一种配方(权重)。在卷积神经网络里,卷积核(滤波器)就像这个配方,它在整个图像上滑动(类似咖啡师去处理不同的订单),使用同一组参数来检测特征。

权值共享通过减少参数数量和提升特征提取能力, 从而有效解决了传统BP (误差反向传播) 神经网络在处理平移、旋转等变换时的局限性

filter:

1	0	1
0	1	0
1	0	1

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

1	1 _{x1}	1 _{x0}	0	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0	1 _{x1}	1 _{x0}	1
0	0	1	1	0
0	1	1	0	0

Image

4	3	

Convolved
Feature

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

4	3	4
2	4	3
2	3	4

Convolved
Feature

1	1	1	0	0
0 _{x1}	1 _{x0}	1 _{x1}	1	0
0 _{x0}	0 _{x1}	1 _{x0}	1	1
0 _{x1}	0 _{x0}	1 _{x1}	1	0
0	1	1	0	0

Image

4	3	4
2		

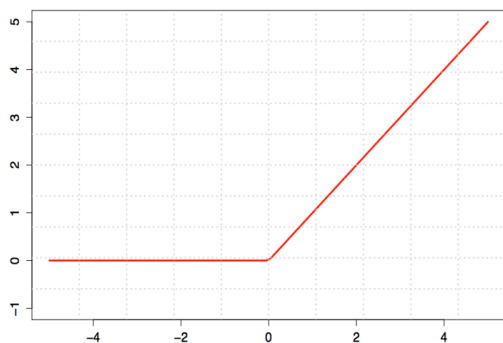
Convolved
Feature

1.3.1.2.2 非线性映射(ReLU)

ReLU的作用是引入非线性,使得网络能够学习复杂的非线性映射关系。如果没有 ReLU, 网络只是一个线性变换的叠加, 本质上仍然是一个线性模型, 无法有效处理复杂问题。

和前馈神经网络一样, 经过线性组合和偏移后, 会加入非线性增强模型的拟合能力。

经过线性组合和偏移后, 会加入非线性增强模型的拟合能力。将卷积所得的Feature Map经过ReLU变换 (element wise)。



- 当 $x > 0$ 时, 输出 x (保持线性)
- 当 $x \leq 0$ 时, 输出 0 (抑制负值)

1.3.1.2.3 example

LeNet-5