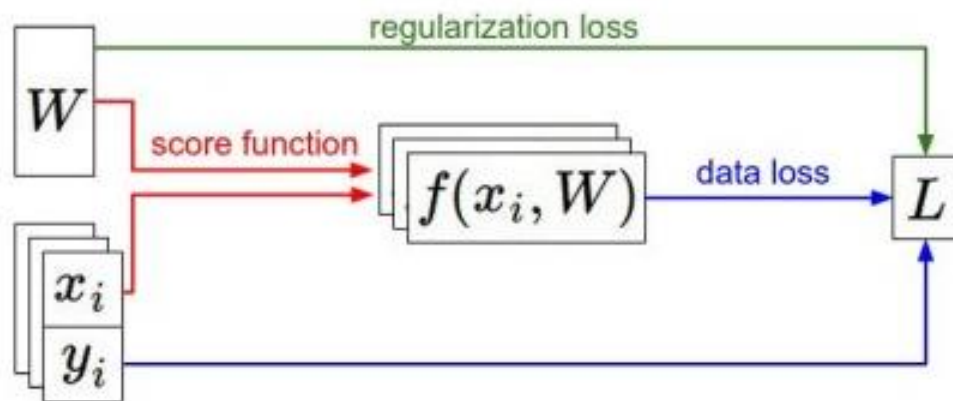




卷积神经网络(CNN)

神经网络信息流动



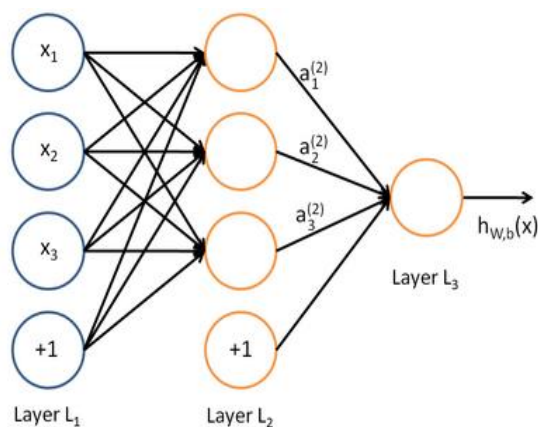
解析：(x,y) 是给定的带标签数据，通常，x为特征，y为标签，固定不变。W为权重(网络参数)，随机初始化(正态分布)，且随时变化。一次前向计算过程，通过score function，得到预测结果，与标签数据对比，计算loss值，通常，loss是规则化项(regularization loss)和均方差项(data loss)的加权和(见式 1),其目的是减小权重的幅度，防止过度拟合。在反向计算时，使用均方差项更新权重。

Notes:

$$\begin{aligned} \text{式1: } J(W, b) &= \left[\frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\ &= \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \end{aligned}$$

神经网络的结点计算

前向计算:



$$\begin{aligned}a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})\end{aligned}$$

反向传播:

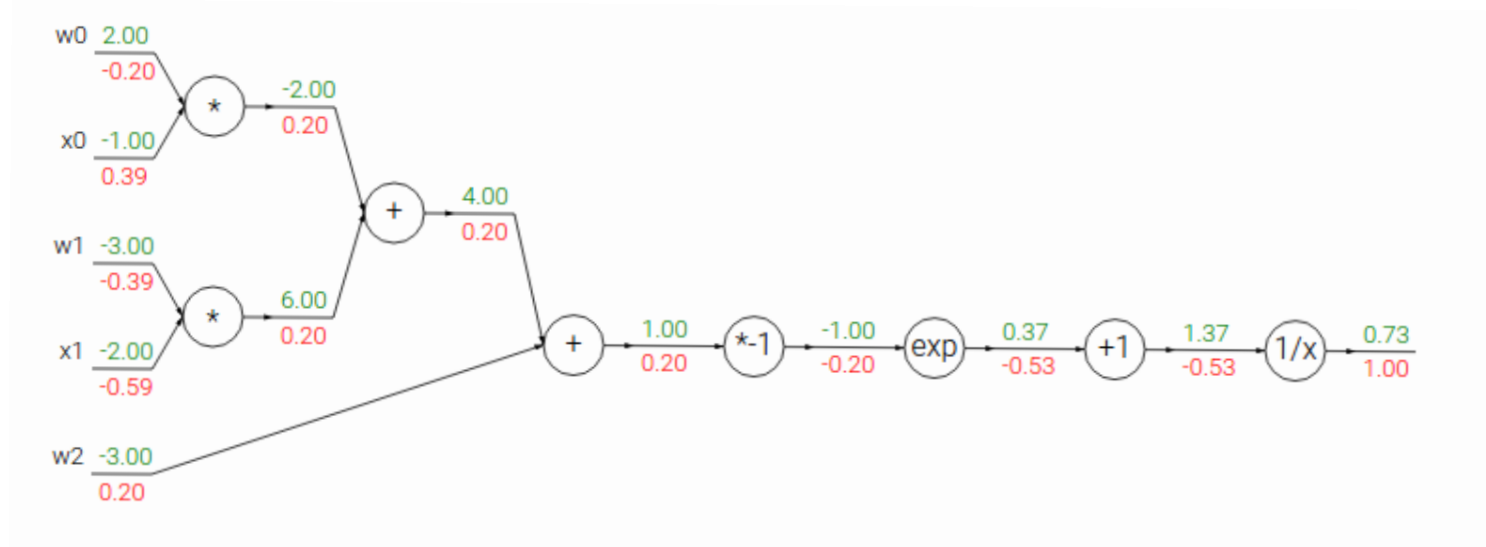
$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b)$$

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)}$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b) = \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b_i^{(l)}} J(W, b; x^{(i)}, y^{(i)})$$

神经网络梯度传播(链式法则)



Notes:

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

卷积神经网络和深度学习的历史

卷积神经网络在深度学习的历史中发挥了重要作用。它们是将研究**大脑获得的深刻理解**成功应用于机器学习应用的**关键例子**，也是**第一个**表现良好的深度模型之一。是**第一个**解决重要商业应用的神经网络，并且仍然是当今深度学习应用的前沿。

在20世纪90年代，A T & T的神经网络研究小组开发了一个用于读取支票的卷积神经网络，到90年代末，NEC部署的这个系统用于读取美国所有支票的10%。后来，微软部署了若干个基于卷积神经网络的**OCR和手写识别系统(MNIST)**。

卷积神经网络也被用来赢得许多比赛。当前对深度学习的商业热潮始于2012年，当时**Alex Krizhevsky**使用新型卷积神经网络(**AlexNet**)赢得了当年的**ImageNet大赛**第一名，TOP-5分类错误率比第二名小约**10%**，引起轰动。

深度学习以及卷积神经网络的适用需要大量的有效训练数据，过去的互联网时代为深度学习提供了大量的**训练数据**，同时随着几十年来**硬件技术**的发展，为利用和计算大量数据提供了条件。所以，近年来，每一次**模型算法**的更新，都取得了良好的效果，为深度学习这把火炬增添了燃料。

卷积神经网络和深度学习的历史

卷积神经网络提供了一种方法来**专业化**神经网络，以处理具有清楚的网络结构的数据，以及将这样的模型放大到非常大的尺寸(**加深层数**)。这种方法在**二维图像拓扑**上的应用是最成功的。同时，卷积神经网络比全连接网络**计算效率**更高，使用它们运行多个实验并调整它们的实现和超参数更容易，更大的网络也更容易训练。

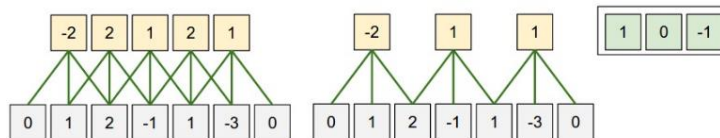
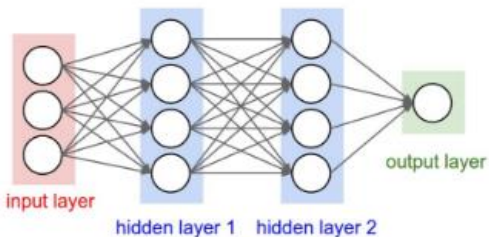
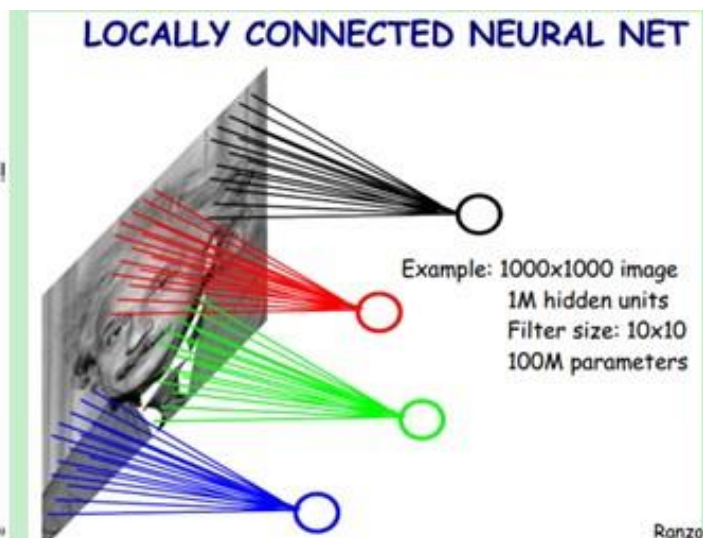
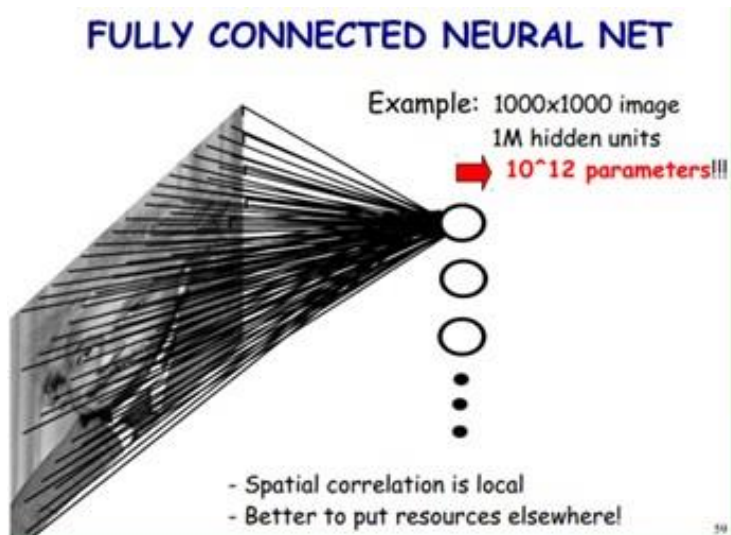
为了处理一维序列数据，便有了循环神经网络，以及基于循环神经网络优化而来的lstm,attention机制等。

利用卷积运算的三个特点：**稀疏交互、参数共享、等变表示**

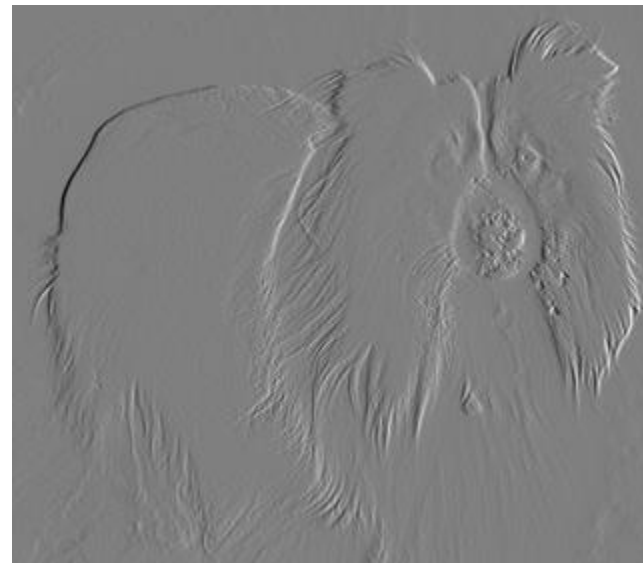
稀疏交互即日常所说的局部链接；**参数共享**即神经元间共享参数；**等变表示**即平移不变性

CNN处理图像

卷积神经网络的计算效率提升，参数量： 10^{12} - $>$ 10^6



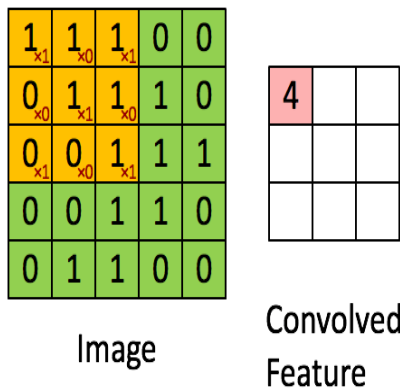
CNN处理图像



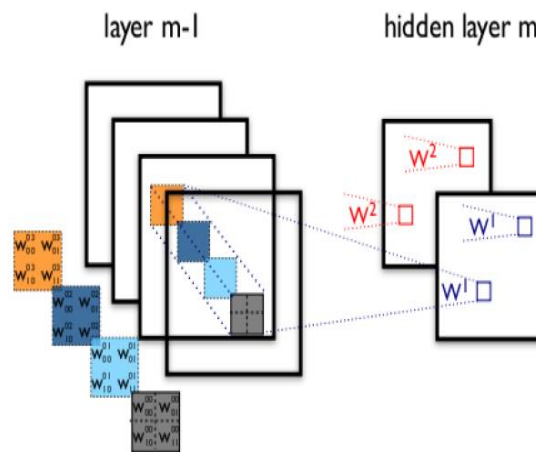
边缘检测的效率。右边的图像是通过获得原始图像中的每个像素并减去左边相邻像素的值而形成的。这对目标检测是有用的操作。两个图像都是 280 像素的高度。输入图像宽 320 像素,而输出图像宽 319 像素。这个变换可以通过包含两个元素的卷积核来描述,并且需要 $319 \times 280 \times 3 = 267,960$ 个浮点运算(每个输出像素需要两次乘法和一次加法)。不使用卷积,需要 $320 \times 280 \times 319 \times 280$ 个或者说超过 80 亿个元素的矩阵,这使得卷积对于表示这种变换更有效 40 亿倍。直接运行矩阵乘法的算法将执行超过 160 亿个浮点运算,这使得卷积在计算上大约有 60,000 倍的效率。将小的局部区域上的相同线性变换应用到整个输入上,卷积是描述这种变换的极其有效的方法。照片来源:Paula Goodfellow。

CNN特性 - 权值共享和多卷积核

卷积神经网络之所以计算效率高，对特征提取的效果好，主要是由于卷积神经网络具有以下三个特性：权值共享，多卷积核，池化。

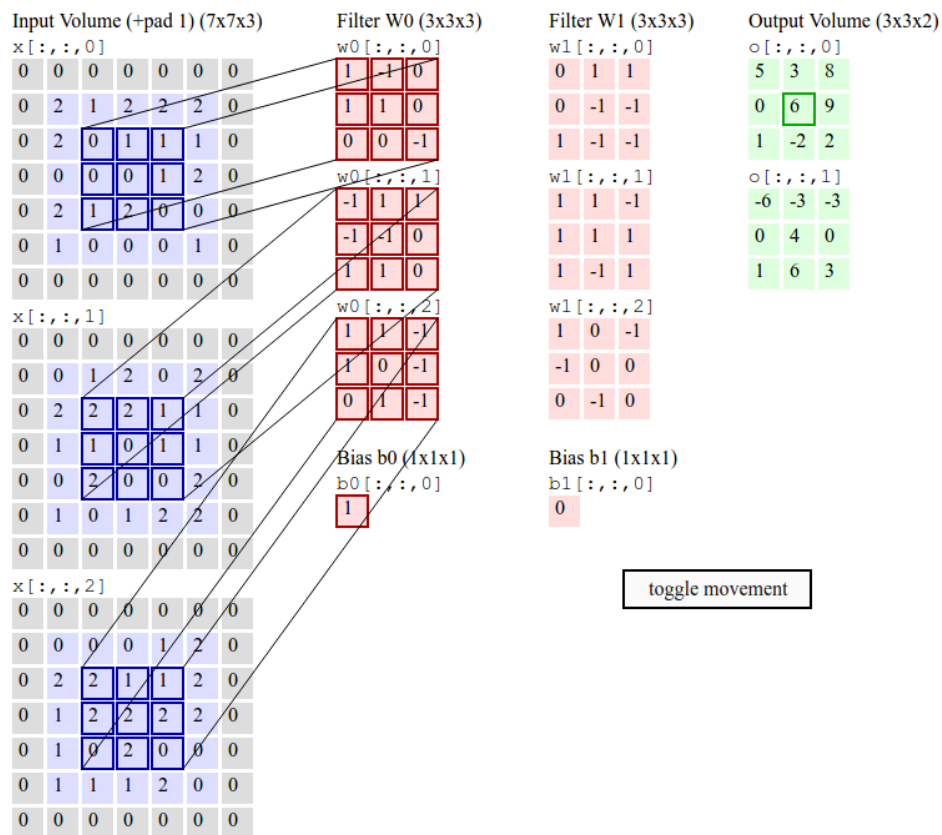


权值共享



多通道多卷积核

CNN多通道和多卷积核

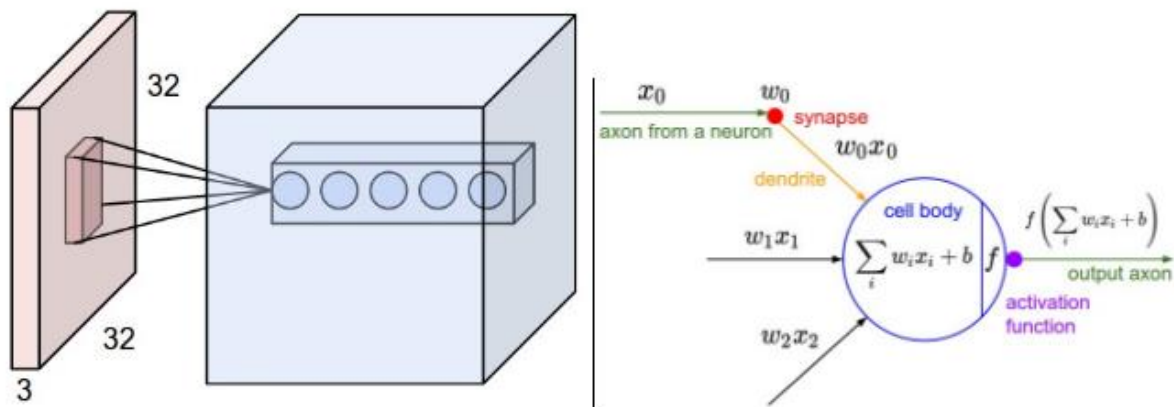


CS231N

<http://cs231n.github.io/convolutional-networks/>

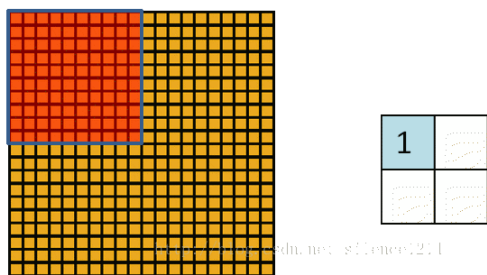
卷积的矩阵转换

感受野和卷积核是卷积运算的一种特殊设定和直观表示，卷积核和感受野之间的卷积运算使用向量矩阵的形式实现，提高了计算效率。



CNN 特性 - 池化

卷积神经网络池化有最大池化(max_pool)和平均池化(avg_pool)，顾名思义，最大池化取区域内最大值，平均池化取区域内平均值。其它池化包括L2范数以及依靠据中心像素距离的加权平均池化。



Convolved
feature

Pooled
feature

CNN池化过程

CNN 特性 - 池化

为什么要池化？

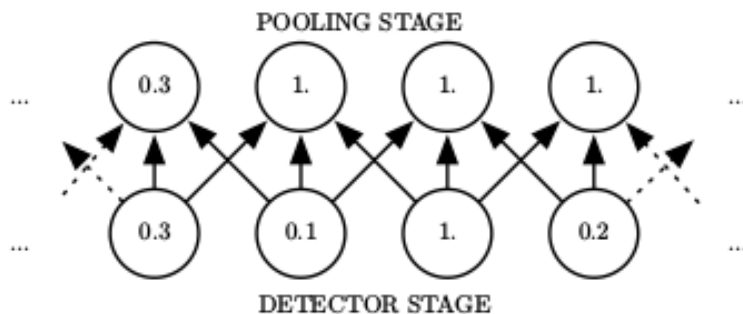
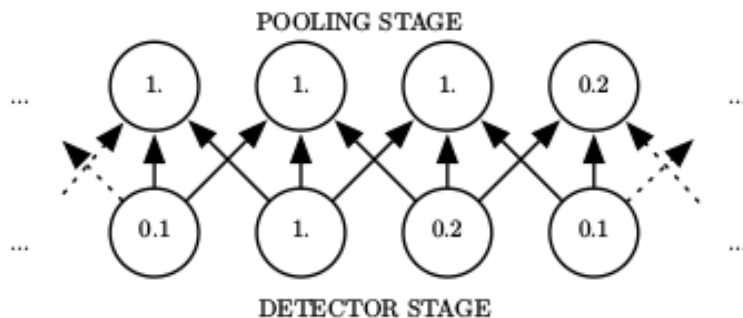
1. 减少参数的量，提高计算效率.
2. 最大池化能显著增强局部特征，平均池化可减少噪声.
(最大池化提取轮廓特征，平均池化可模糊图像)
3. 提高局部平移不变性. (不考虑空间，时间位置. - 张民)

局部平移不变性是一个很重要的性质,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，这对于图像中的目标检测至关重要，同时，在图像识别中，同一类别的图像往往会有细微的差别，局部平移不变性大大提高了图像分类的准确度.

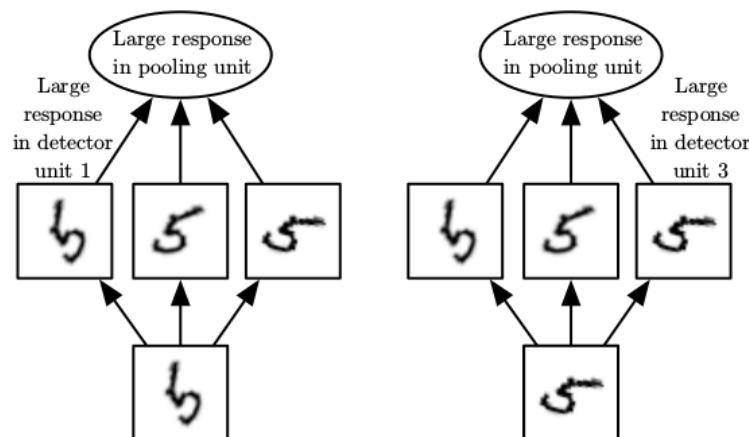
CNN 特性 - 局部平移不变性

为什么最大池化能做到？

上图下半部分的输入像素向右平移一个单位，下图对应位置像素值全部改变，若对输入像素做池化处理，当原始像素平移一个单位后，只有一半的像素值改变。



CNN 特性 - 局部平移不变性



学习不变性的示例。使用分离的参数学得多个特征,再使用池化单元进行池化,可以学得对输入的某些变换的不变性。这里我们展示了用三个学得的过滤器(卷积核)和一个最大池化单元可以学得对旋转变换的不变性。这三个过滤器都旨在检测手写的数字 5。每个卷积核尝试匹配稍微不同方向的 5。当输入中出现 5 时,相应的卷积核会匹配它并且在探测单元(激活函数)中引起大的激活。然后,无论哪个探测单元被激活,最大池化单元都具有大的激活。我们在这里展示网络如何处理两个不同的输入,导致两个不同的探测单元被激活。然而对池化单元的影响大致相同。空间位置上的最大池化对于平移是天然不变的;这种多通道方法在学习其他变换时是必要的。

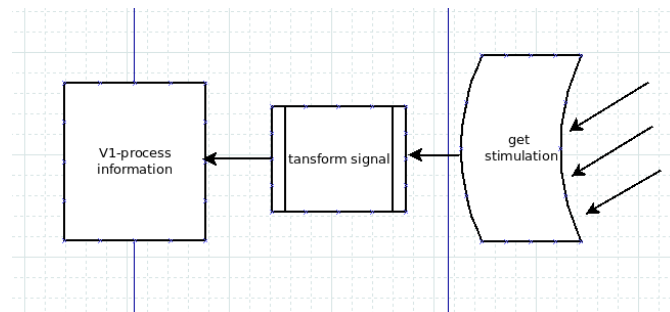
卷积神经网络的神经科学基础

卷积网络也许是生物学启发人工智能的最为成功的故事。虽然卷积网络已经被其它领域指导，但是神经网络的一些关键设计原则来自于神经科学。

卷积网络的历史始于David Hubel 和Torsten Wiesel的神经科学实验，为了确定哺乳动物视觉系统如何工作的基本事实，他们观察了猫的脑内神经元如何响应投影在猫前面屏幕上精确位置的图像。

他们发现：处于视觉系统**较为前面**的神经元对非常**特定**的光模式(例如精确定向的条纹)反应最强烈，但对其他模式几乎完全没有反应。

卷积神经网络的神经科学基础



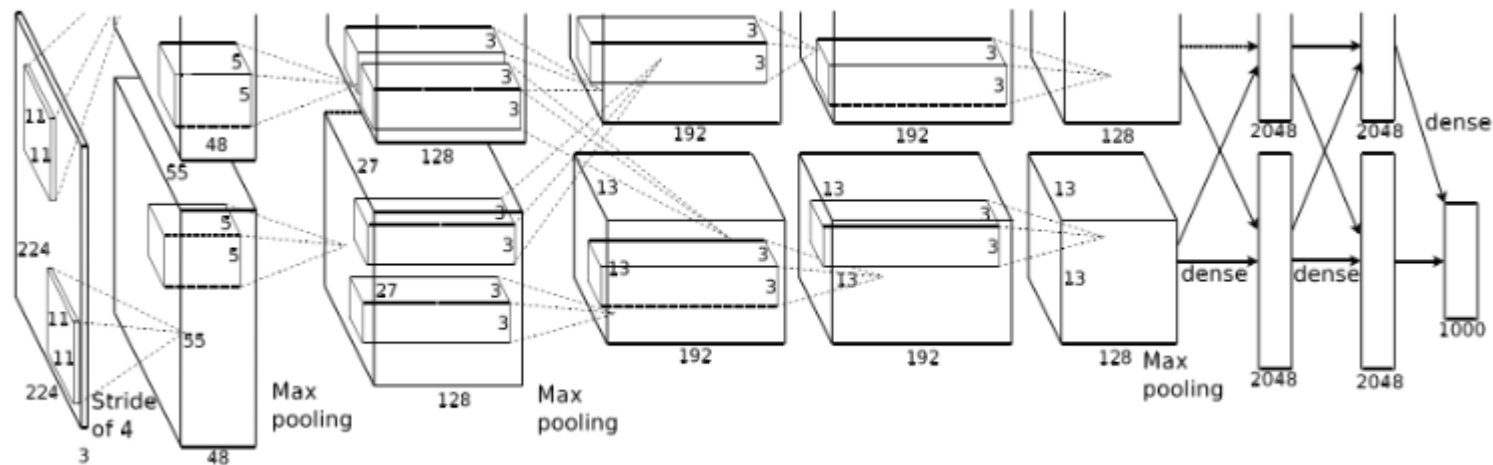
图像是由光到达眼睛并刺激视网膜形成的，视网膜的神经元对图像执行一些简单的预处理，但是基本改变它被表示的方式，然后图像通过视神经和脑部区域，这些区域负责将信号从眼睛传递到位于脑后部的主要视觉皮层，它是大脑对视觉输入开始执行**显著高级处理的第一个区域** (以下简称V1)，卷积网络的设计参考了其对视觉信号处理的三个主要性质：

1. V1具有二维结构来反映视网膜中的图像结构，例如，到达视网膜下半部的光仅影响V1相应的一半。卷积网络通过用二维映射定义特征的方式来实现该特性。(CNN特性：权值共享和局部池化操作)

2. V1包含许多简单细胞，简单细胞的活动在某种程度上可以概括为在一个**小的空间位置接受域内**的图像的**线性函数**，卷积神经网络不同卷积核的权重与输入的线性求和，模拟了简单细胞对输入信号的处理。

3. V1还包括许多的复杂细胞，这些细胞响应由**简单细胞检测的那些特征**，但是复杂细胞对于特征位置的微小偏移具有不变性，这启发了网络的池化单元和激活函数。

ImageNet Classification with Deep Convolutional Neural Networks



ImageNet LSVRC-2012冠军,1000类,120万高清图像,Top5Error:26.2% → 15.3%.

结构:

由6000万个参数和650,000个神经元, 由五个卷积层和其后的max-pooling层以及三个全连接层,1000-way的softmax层组成.以及开创性的使用“**dropout**”技术, 避免了**过拟合**.

计算开销问题:

采取将网络分布在两个GPU上,在每个GPU中放置一半核（或神经元）, 还有一个额外的技巧: GPU间的通讯只在某些层进行。

Rethinking the Inception Architecture for Computer Vision

这也是获得ImageNet LSVRC-2014冠军的模型，共22层的网络。

文章提出获得高质量模型最保险的做法就是**增加模型的深度（层数）或者是其宽度（层核或者神经元数）**，但是这里一般设计思路的情况下会出现两个缺陷（1.若训练数据集有限，参数太多，容易过拟合；2.网络越大计算复杂度越大，难以应用；3.网络越深，梯度越往后穿越容易消失，难以优化模型）。

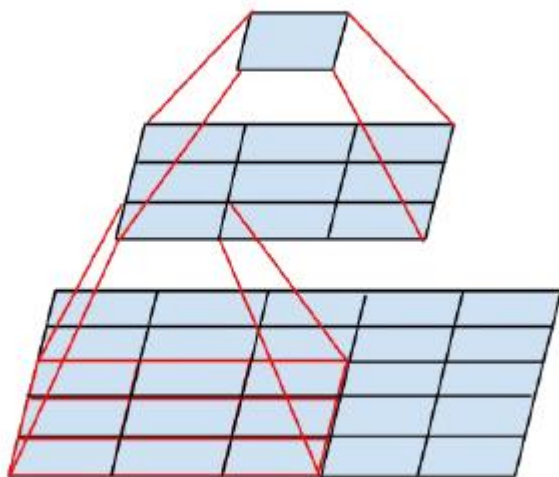
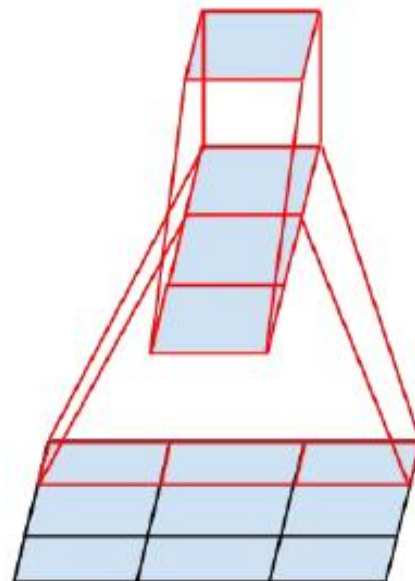


Figure 1. Mini-network replacing the 5×5 convolutions.



Rethinking the Inception Architecture for Computer Vision

googlenet的主要思想就是围绕这两个思路去做的：

1.深度，层数更深，文章采用了22层，googlenet巧妙的在不同深度处增加了两个loss来避免上述提到的梯度消失问题，。

2.宽度，增加了多种核1x1，3x3，5x5，在3x3前，5x5前，max pooling后分别加上了1x1的卷积核起到了降低feature map厚度的作用。以下是googlenet用的inception可以称之为inception v1，如下图所示：

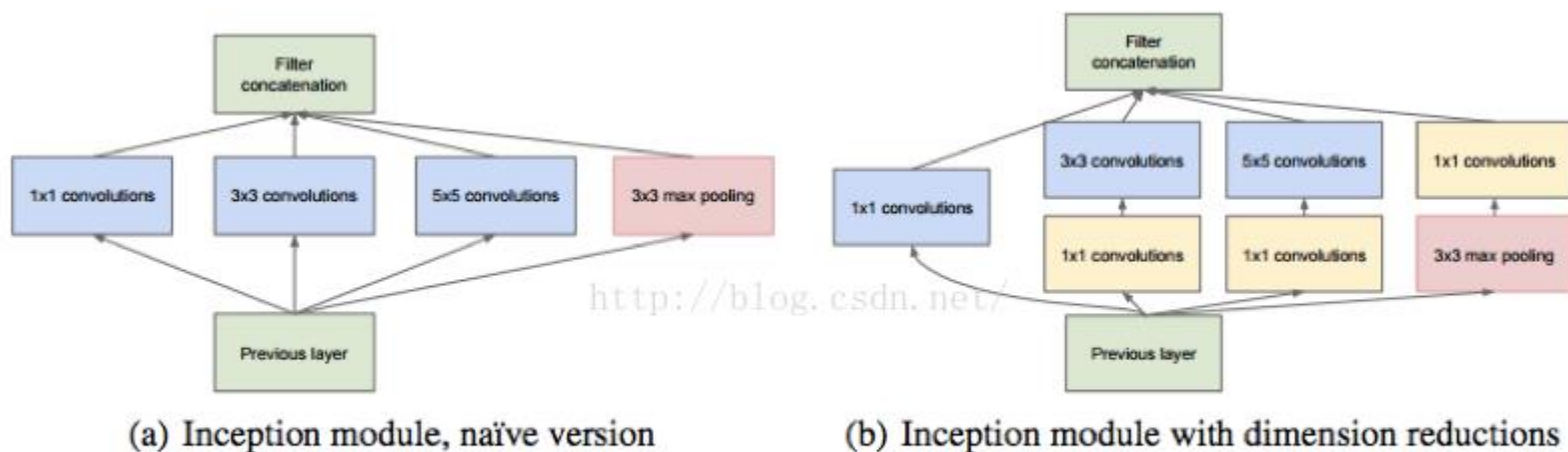


Figure 2: Inception module