机器学习　神经网络　深度学习（Deep Learning）

CNN(卷积神经网络)是什么？有入门简介或文章吗?

想要了解一下CNN的原理及作用，还有实现方法。

21 个回答　默认排序

机器之心
欢迎关注我们的微信公众号：机器之心（almosthuman2014）

1304 人赞同了该回答

Part 1：图像识别任务

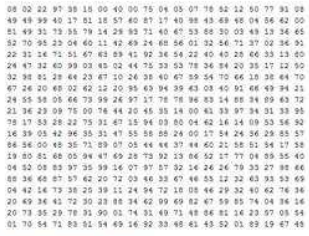
卷积神经网络，听起来像是计算机科学、生物学和数学的诡异组合，但它们已经成为计算机视觉领域中最具影响力的革新的一部分。神经网络在 2012 年崭露头角，Alex Krizhevsky 凭借它们赢得了那一年的 ImageNet 挑战赛（大体上相当于计算机视觉的年度奥林匹克），他把分类误差记录从 26% 降到了 15%，在当时震惊了世界。自那之后，大量公司开始将深度学习用作服务的核心。Facebook 将神经网络用于自动标注算法、谷歌将它用于图片搜索、亚马逊将它用于商品推荐、Pinterest 将它用于个性化主页推送、Instagram 将它用于搜索架构。

然而，应用这些网络最经典最流行的案例是进行图像处理。在图像处理任务中，让我们看一下如何使用卷积神经网络进行图像分类。

问题空间

图像分类是对输入图像的操作，最终输出一组最好地描述了图像内容的分类（如猫、狗等）或分类的概率。对人类来说，识别是打出生便开始学习的技能之一，对成人来说更是信手拈来，毫不费力。我们只需一眼便能快速识别我们所处的环境以及环绕在我们身边的物体。当我们看到一张图片或是环看四周的时候，无需刻意观察，多数时候也能立即描述出场景特征并标记出每一个对象。快速识别不同模式、根据早前知识进行归纳、以及适应不同的图像环境一直都是人类的专属技能，机器尚未享有。


What We See


What Computers See

输入与输出

当计算机看到一张图像（输入一张图像）时，它看的是一大堆像素值。根据图片的分辨率和尺寸，它将看到一个 32 x 32 x 3 的数组（3 指代的是 RGB 值）。为了讲清楚这一点，假设我们有一张

CNN(卷积神经网络)是什么？有入门简介或文章吗？

JPG 格式的 480 x 480 大小的彩色图片，那么它对应的数组就有 480 x 480 x 3 个元素，其中每个数字的值从 0 到 255 不等，其描述了对应那一点的像素灰度。当我们人类对图像进行分类时，这些数字毫无用处，可它们却是计算机可获得的唯一输入。其中的思想是：当你提供给计算机这一数组后，它将输出描述该图像属于某一特定分类的概率的数字（比如：80% 是猫、15% 是狗、5% 是鸟）。

我们想要计算机做什么

现在我们知道了问题所在以及输入与输出，就该考虑如何处理了。我们想要计算机能够区分开所有提供给它的图片，以及搞清楚猫猫狗狗各自的特有特征。这也是我们人类的大脑中不自觉进行着的过程。当我们看到一幅狗的图片时，如果有诸如爪子或四条腿之类的明显特征，我们便能将它归类为狗。同样地，计算机也可以通过寻找诸如边缘和曲线之类的低级特点来分类图片，继而通过一系列卷积层级建构出更为抽象的概念。这是 CNN（卷积神经网络）工作方式的大体概述，下面是具体细节。

生物学连接

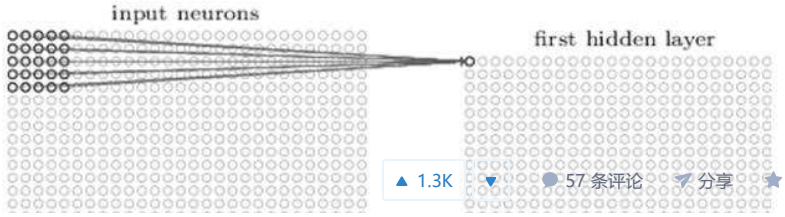
首先介绍些背景。当你第一次听到卷积神经网络这一术语，可能会联想到神经科学或生物学，那就对了。可以这样说。CNN 的确是从视觉皮层的生物学上获得启发的。视觉皮层有小部分细胞对特定部分的视觉区域敏感。Hubel 和 Wiesel 于 1962 年进行的一项有趣的试验详细说明了这一观点，他们验证出大脑中的一些个体神经细胞只有在特定方向的边缘存在时才能做出反应（即放电）。例如，一些神经元只对垂直边缘兴奋，另一些对水平或对角边缘兴奋。Hubel 和 Wisesl 发现所有这些神经元都以柱状结构的形式进行排列，而且一起工作才能产生视觉感知。这种一个系统中的特定组件有特定任务的观点（视觉皮层的神经元细胞寻找特定特征）在机器中同样适用，这就是 CNN 的基础。

结构

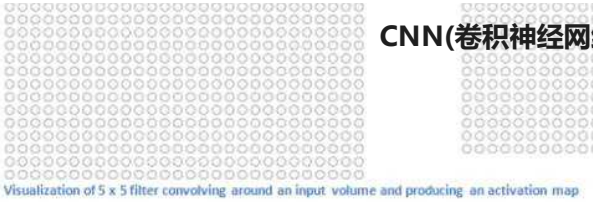
回到细节上来。更为详细的 CNN 工作概述指的是你挑一张图像，让它历经一系列卷积层、非线性层、池化（下采样（downsampling））层和完全连接层，最终得到输出。正如之前所说，输出可以是最好地描述了图像内容的一个单独分类或一组分类的概率。如今，难点在于理解其中每一层的工作方法。我们先来看最重要的部分。

第一层——数学部分

CNN 的第一层通常是卷积层（Convolutional Layer）。首先需要了解卷积层的输入内容是什么。如上所述，输入内容为一个 32 x 32 x 3 的像素值数组。现在，解释卷积层的最佳方法是想象有一束手电筒光正从图像的左上角照过。假设手电筒光可以覆盖 5 x 5 的区域，想象一下手电筒光照过输入图像的所有区域。在机器学习术语中，这束手电筒被叫做过滤器（filter，有时候也被称为神经元（neuron）或核（kernel）），被照过的区域被称为感受野（receptive field）。过滤器同样也是一个数组（其中的数字被称作权重或参数）。重点在于过滤器的深度必须与输入内容的深度相同（这样才能确保可以进行数学运算），因此过滤器大小为 5 x 5 x 3。现在，以过滤器所处在的第一个位置为例，即图像的左上角。当筛选值在图像上滑动（卷积运算）时，过滤器中的值会与图像中的原始像素值相乘（又称为计算点积）。这些乘积被加在一起（从数学上来说，一共会有 75 个乘积）。现在你得到了一个数字。切记，该数字只是表示过滤器位于图片左上角的情况。我们在输入内容上的每一位置重复该过程。（下一步将是将过滤器右移 1 单元，接着再右移 1 单元，以此类推。）输入内容上的每一特定位置都会产生一个数字。过滤器滑过所有位置后将得到一个 28 x 28 x 1 的数组，我们称之为激活映射（activation map）或特征映射（feature map）。之所以得到一个 28 x 28 的数组的原因在于，在一张 32 x 32 的输入图像上，5 x 5 的过滤器能够覆盖到 784 个不同的位置。这 784 个位置可映射为一个 28 x 28 的数组。



CNN(卷积神经网络)是什么？有入门简介或文章吗？



（注意：包括上图在内的一些图片来自于 Micheal Nielsen 的「神经网络与深度学习（ Neural Networks and Deep Learning）」一书。我强烈推荐这本书。这本书可免费在线浏览：Neural networks and deep learning）

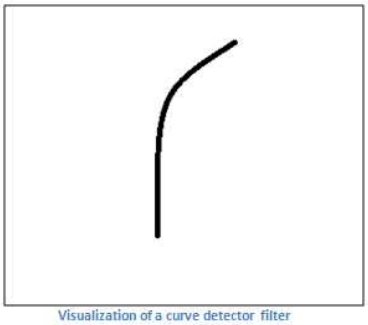
当我们使用两个而不是一个 5 x 5 x 3 的过滤器时，输出总量将会变成 28 x 28 x 2。采用的过滤器越多，空间维度（ spatial dimensions）保留得也就越好。数学上而言，这就是卷积层上发生的事情。

第一层——高层次角度

不过，从高层次角度而言卷积是如何工作的？每个过滤器可以被看成是特征标识符（ feature identifiers）。这里的特征指的是例如直边缘、原色、曲线之类的东西。想一想所有图像都共有的一些最简单的特征。假设第一组过滤器是 7 x 7 x 3 的曲线检测器。（在这一节，为了易于分析，暂且忽略该过滤器的深度为 3 个单元，只考虑过滤器和图像的顶层层面。）作为曲线过滤器，它将有一个像素结构，在曲线形状旁时会产生更高的数值（切记，我们所讨论的过滤器不过是一组数值！）

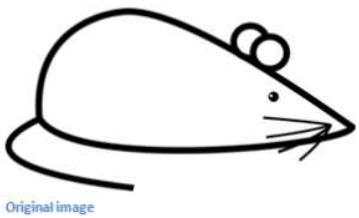
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

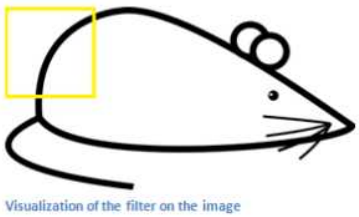
Pixel representation of filter



左图：过滤器的像素表示；右图：曲线检测器过滤器的可视化；对比两图可以看到数值和形状的对应

回到数学角度来看这一过程。当我们将过滤器置于输入内容的左上角时，它将计算过滤器和这一区域像素值之间的点积。拿一张需要分类的照片为例，将过滤器放在它的左上角。





切记，我们要做的是将过滤器与图像的原始像素值相乘。

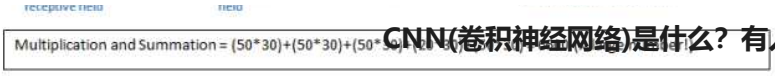
0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the

*

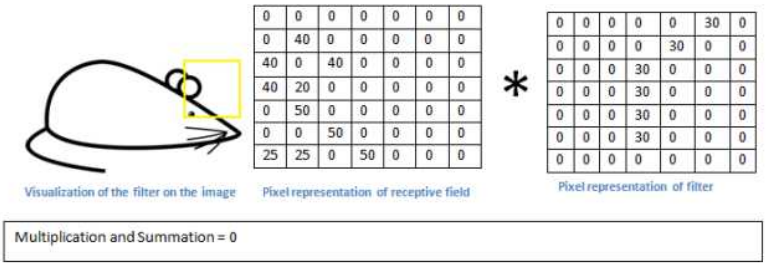
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



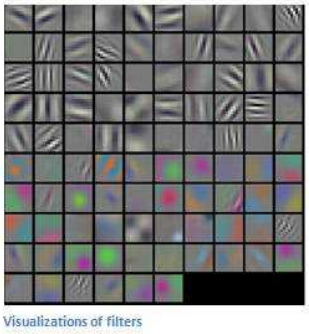
*左图：感受野的可视化；右图：感受野的像素表示 * 过滤器的像素表示*

简单来说，如果输入图像上某个形状看起来很像过滤器表示的曲线，那么所有点积加在一起将会得出一个很大的值！让我们看看移动过滤器时会发生什么。



这个值小了很多！这是因为图像的这一部分和曲线检测器过滤器不存在对应。记住，这个卷积层的输出是一个激活映射（activation map）。因此，在这个带有一个过滤器卷积的例子里（当筛选值为曲线检测器），激活映射将会显示出图像里最像曲线的区域。在该例子中，28 x 28 x 1 的激活映射的左上角的值为 6600。高数值意味着很有可能是输入内容中的曲线激活了过滤器。激活地图右上角的值将会是 0，因为输入内容中没有任何东西能激活过滤器（更简单地说，原始图片中的这一区域没有任何曲线）。这仅仅是一组检测右弯曲线的过滤器。还有其它检测左弯曲线或直线边缘的过滤器。过滤器越多，激活映射的深度越大，我们对输入内容的了解也就越多。

声明：我在本小节中描绘的过滤器（filter）只是为了描述卷积中的数学过程。在下图中你可以看到训练后的网络中第一个卷积层的过滤器的实际可视化。尽管如此，主要观点仍旧不变。当在输入内容中寻找特定特征时，第一层上的过滤器在输入图像上进行卷积运算和「激活」（即计算高数值）。



上图来自于斯坦福大学由 Andrej Karpathy 和 Justin Johnson 授课的 CS 231N 课程，推荐给渴望更深层理解 CNN 的人们：CS231n: Convolutional Neural Networks for Visual Recognition

网络中的更深处

在传统卷积神经网络架构中，卷积层之间还有其它类型的层。我强烈建议有兴趣的人阅读和它们有关的材料，并理解相应的功能和作用；但总的来说，它们提供的非线性和维度保留有助于提高网络的稳健性（robustness）并控制过拟合。一个典型的 CNN 结构看起来是这样的：

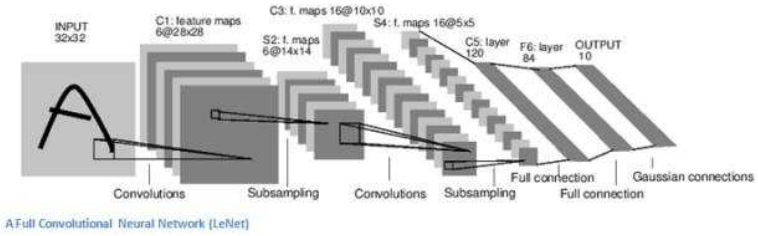
输入→卷积→ReLU→卷积→ReLU→池化→ReLU→卷积→ReLU→池化→全连接

CNN(卷积神经网络)是什么？有入门简介或文章吗？

我们稍后再来讨论关键的最后一层，先回顾一下学到了哪些。我们讨论了过滤器是如何在第一个卷积层检测特征的。它们检测边缘和曲线一类的低级特征。正如想象的那样，为了预测出图片内容的分类，网络需要识别更高级的特征，例如手、爪子与耳朵的区别。第一个卷积层的输出将会是一个 28 x 28 x 3 的数组（假设我们采用三个 5 x 5 x 3 的过滤器）。当我们进入另一卷积层时，第一个卷积层的输出便是第二个卷积层的输入。解释这一点有些困难。第一层的输入是原始图像，而第二卷积层的输入正是第一层输出的激活映射。也就是说，这一层的输入大体描绘了低级特征在原始图片中的位置。在此基础上再采用一组过滤器（让它通过第 2 个卷积层），输出将是表示了更高级的特征的激活映射。这类特征可以是半圆（曲线和直线的组合）或四边形（几条直线的组合）。随着进入网络越深和经过更多卷积层后，你将得到更为复杂特征的激活映射。在网络的最后，可能会有一些过滤器会在看到手写笔迹或粉红物体等时激活。如果你想知道更多关于可视化卷积网络中过滤器的内容，可以查看 Matt Zeiler 和 Rob Fergus 的一篇讨论该问题的颇为杰出的研究论文。在 YouTube 上，Jason Yosinski 有一段视频十分视觉化地呈现了这一过程（如下）。有趣的是，越深入网络，过滤器的感受野越大，意味着它们能够处理更大范围的原始输入内容（或者说它们可以对更大区域的像素空间产生反应）。

完全连接层

检测高级特征之后，网络最后的完全连接层就更是锦上添花了。简单地说，这一层处理输入内容（该输入可能是卷积层、ReLU 层或是池化层的输出）后会输出一个 N 维向量，N 是该程序必须选择的分类数量。例如，如果你想得到一个数字分类程序，如果有 10 个数字，N 就等于 10。这个 N 维向量中的每一数字都代表某一特定类别的概率。例如，如果某一数字分类程序的结果矢量是 [0 .1 .1 .75 0 0 0 0 0 .05]，则代表该图片有 10% 的概率是 1、10% 的概率是 2、75% 的概率是 3、还有 5% 的概率是 9（注：还有其他表现输出的方式，这里只展示了 softmax 的方法）。完全连接层观察上一层的输出（其表示了更高级特征的激活映射）并确定这些特征与哪一分类最为吻合。例如，如果该程序预测某一图像的内容为狗，那么激活映射中的高数值便会代表一些爪子或四条腿之类的高级特征。同样地，如果程序测定某一图片的内容为鸟，激活映射中的高数值便会代表诸如翅膀或鸟喙之类的高级特征。大体上来说，完全连接层观察高级特征和哪一分类最为吻合和拥有怎样的特定权重，因此当计算出权重与先前层之间的点积后，你将得到不同分类的正确概率。



A Full Convolutional Neural Network (LeNet)

训练（也就是：什么能让其有效）

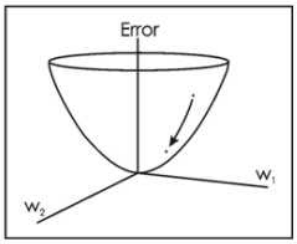
下面是神经网络中的一个我尚未提及但却最为重要的部分。阅读过程中你可能会提出许多问题。第一卷积层中的滤波器是如何知道寻找边缘与曲线的？完全连接层怎么知道观察哪些激活图？每一层级的滤波器如何知道需要哪些值？计算机通过一个名为反向传播的训练过程来调整过滤器值（或权重）。

在探讨反向传播之前，我们首先必须回顾一下神经网络工作起来需要什么。在我们刚出生的时候，大脑一无所知。我们不晓得猫啊狗啊鸟啊都是些什么东西。与之类似的是 CNN 刚开始的时候，权重或过滤器值都是随机的。滤波器不知道要去寻找边缘和曲线。更高层的过滤器值也不知道要去寻找爪子和鸟喙。不过随着年岁的增长，父母和老师向我们介绍各式各样的图片并且一一作出标记。CNN 经历的便是一个介绍图片与分类标记的训练过程。在深入探讨之前，先设定一个训练集，在这里有上千张狗、猫、鸟的图片，每一张都依照内容被标记。下面回到反向传播的问题上来。

反向传播可分为四部分，分别是前向传导、损失函数、后向传导，以及权重更新。在前向传导中，选择一张 32×32×3 的数组训练图像并让它通过整个网络。在第一个训练样例上，由于所有的权重或者过滤器值都是随机初始化的，输出可能会是 [.1 .1 .1 .1 .1 .1 .1 .1 .1 .1]，即一个不偏向任何数字的输出。一个有着这样权重的网络无法寻找低级特征，或者说是不能做出任何合理的分类。接下来是反向传播的损失函数部分。切记我们现在使用的是既有图像又有标记的训练数据。假设输入的第一张训练图片为 3，标签将会是 [0 0 0 1 0 0 0 0 0 0]。损失函数有许多种定义方法，常见的一种是 MSE （均方误差）。

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

假设变量 L 等同该数值。正如所料，前两张训练图片的损失将会极高。现在，我们直观地想一下。我们想要预测标记（卷积网络的输出）与训练标记相同（意味着网络预测正确）。为了做到这一点，我们想要将损失数量最小化。将其视为微积分优化问题的话，也就是说我们想要找出是哪部分输入（例子中的权重）直接导致了网络的损失（或错误）。



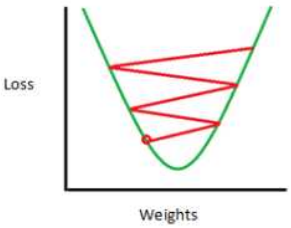
One way of visualizing this idea of minimizing the loss is to consider a 3-D graph where the weights of the neural net (there are obviously more than 2 weights, but let's go for simplicity) are the independent variables and the dependent variable is the loss. The task of minimizing the loss involves trying to adjust the weights so that the loss decreases. In visual terms, we want to get to the lowest point in our bowl shaped object. To do this, we have to take a derivative of the loss (visual terms: calculate the slope in every direction) with respect to the weights.

这是一个 dL/dW 的数学等式，W 是特定层级的权重。我们接下来要做的是在网络中进行后向传导，测定出是哪部分权重导致了最大的损失，寻找调整方法并减少损失。一旦计算出该导数，将进行最后一步也就是权重更新。所有的过滤器的权重将会更新，以便它们顺着梯度方向改变。

$$w = w_i - \eta \frac{dL}{dW}$$

w = Weight
w_i = Initial Weight
η = Learning Rate

学习速率是一个由程序员决定的参数。高学习速率意味着权重更新的动作更大，因此可能该模式将花费更少的时间收敛到最优权重。然而，学习速率过高会导致跳动过大，不够准确以致于达不到最优点。



Consequence of a high learning rate where the jumps are too large and we are not able to minimize the loss.

总的来说，前向传导、损失函数、后向传导、以及参数更新被称为一个学习周期。对每一训练图片，程序将重复固定数目的周期过程。一旦完成了最后训练样本上的参数更新，网络有望得到足够好的训练，以便层级中的权重得到正确调整。

测试

最后，为了检验 CNN 能否工作，我们准备不同的另一组图片与标记集（不能在训练和测试中使用相同的！）并让它们通过这个 CNN。我们将输出与实际情况（ground truth ）相比较，看看网络是否有效！

企业如何使用 CNN

数据、数据、数据。数据越多的企业在竞争中越发彰显优势。你提供给网络的训练数据越多，你能进行的训练迭代也越多，紧接着权重更新也多，那么当用于产品时调整出的网络自然就好。Facebook （和 Instagram）可以使用它如今拥有的十几亿用户的图片，Pinterest 可以使用它站点上 500 亿花瓣的信息，谷歌可以使用搜索数据，亚马逊可以使用每天销售的数以百万计的商品数据。而你现在也知道它们使用数据背后的神奇之处了。

Part 2：卷积神经网络中的部分问题

CNN(卷积神经网络)是什么？有入门简介或文章吗?

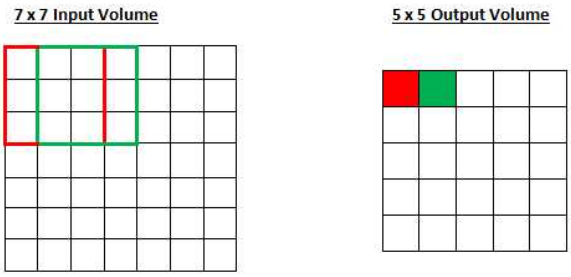
引言

在这篇文章中，我们将更深入地介绍有关卷积神经网络（ConvNet）的详细情况。声明：我确实知道本文中一部分内容相当复杂，可以用一整篇文章进行介绍。但为了在保持全面性的同时保证简洁，我会在文章中相关位置提供一些更详细解释该相关主题的论文链接。

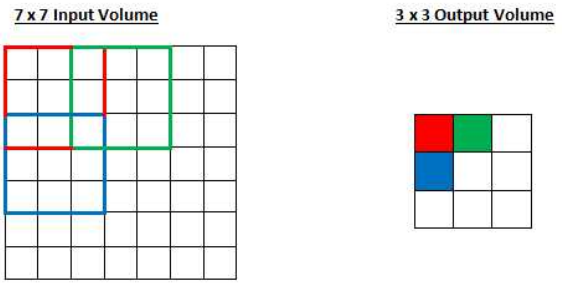
步幅和填充

好了，现在来看一下我们的卷积神经网络。还记得过滤器、感受野和卷积吗？很好。现在，要改变每一层的行为，有两个主要参数是我们可以调整的。选择了过滤器的尺寸以后，我们还需要选择步幅（stride）和填充（padding）。

步幅控制着过滤器围绕输入内容进行卷积计算的方式。在第一部分我们举的例子中，过滤器通过每次移动一个单元的方式对输入内容进行卷积。过滤器移动的距离就是步幅。在那个例子中，步幅被默认设置为1。步幅的设置通常要确保输出内容是一个整数而非分数。让我们看一个例子。想象一个 7 x 7 的输入图像，一个 3 x 3 过滤器（简单起见不考虑第三个维度），步幅为 1。这是一种惯常的情况。

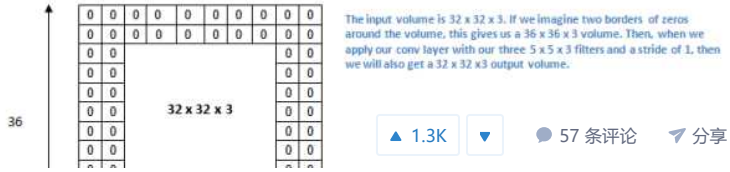


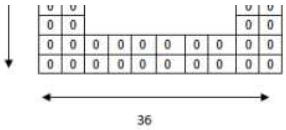
还是老一套，对吧？看你能不能试着猜出如果步幅增加到 2，输出内容会怎么样。



所以，正如你能想到的，感受野移动了两个单元，输出内容同样也会减小。注意，如果试图把我们的步幅设置成 3，那我们就会难以调节间距并确保感受野与输入图像匹配。正常情况下，程序员如果想让接受域重叠得更少并且想要更小的空间维度（spatial dimensions）时，他们会增加步幅。

现在让我们看一下填充（padding）。在此之前，想象一个场景：当你把 5 x 5 x 3 的过滤器用在 32 x 32 x 3 的输入上时，会发生什么？输出的大小会是 28 x 28 x 3。注意，这里空间维度减小了。如果我们继续用卷积层，尺寸减小的速度就会超过我们的期望。在网络的早期层中，我们想要尽可能多地保留原始输入内容的信息，这样我们就能提取出那些低层的特征。比如说我们想要应用同样的卷积层，但又想让输出量维持为 32 x 32 x 3 。为做到这点，我们可以对这个层应用大小为 2 的零填充（zero padding）。零填充在输入内容的边界周围补充零。如果我们用两个零填充，就会得到一个 36 x 36 x 3 的输入卷。





CNN(卷积神经网络)是什么？有入门简介或文章吗？

如果我们在输入内容的周围应用两次零填充，那么输入量就为 32×32×3。然后，当我们应用带有 3 个 5×5×3 的过滤器，以 1 的步幅进行处理时，我们也可以得到一个 32×32×3 的输出

如果你的步幅为 1，而且把零填充设置为

$$Zero\ Padding = \frac{(K-1)}{2}$$

K 是过滤器尺寸，那么输入和输出内容就总能保持一致的空间维度。

计算任意给定卷积层的输出的大小的公式是

$$O = \frac{(W-K+2P)}{S} + 1$$

其中 O 是输出尺寸，K 是过滤器尺寸，P 是填充，S 是步幅。

选择超参数

我们怎么知道要用多少层、多少卷积层、过滤器尺寸是多少、以及步幅和填充值多大呢？这些问题很重要，但又没有一个所有研究人员都在使用的固定标准。这是因为神经网络很大程度上取决于你的数据类型。图像的大小、复杂度、图像处理任务的类型以及其他更多特征的不同都会造成数据的不同。对于你的数据集，想出如何选择超参数的一个方法是找到能创造出图像在合适尺度上抽象的正确组合。

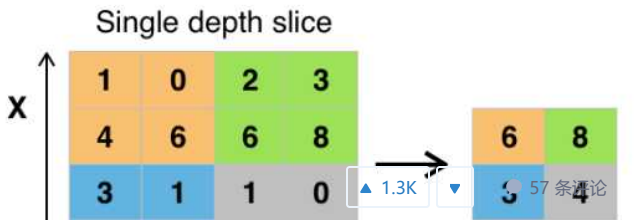
ReLU（修正线性单元）层

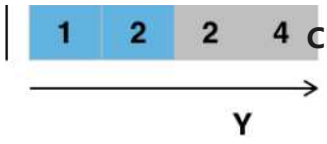
在每个卷积层之后，通常会立即应用一个非线性层（或激活层）。其目的是给一个在卷积层中刚经过线性计算操作（只是数组元素依次（element wise）相乘与求和）的系统引入非线性特征。过去，人们用的是像双曲正切和 S 型函数这样的非线性方程，但研究者发现 ReLU 层效果好得多，因为神经网络能够在准确度不发生明显改变的情况下把训练速度提高很多（由于计算效率增加）。它同样能帮助减轻梯度消失的问题——由于梯度以指数方式在层中消失，导致网络较底层的训练速度非常慢。ReLU 层对输入内容的所有值都应用了函数 f(x) = max(0, x)。用基本术语来说，这一层把所有的负激活（negative activation）都变为零。这一层会增加模型乃至整个神经网络的非线性特征，而且不会影响卷积层的感受野。

- 参见 Geoffrey Hinton（即深度学习之父）的论文：Rectified Linear Units Improve Restricted Boltzmann Machines

池化层

在几个 ReLU 层之后，程序员也许会选择用一个池化层（pooling layer）。它同时也被叫做下采样（downsampling）层。在这个类别中，也有几种可供选择的层，最受欢迎的就是最大池化（max-pooling）。它基本上采用了一个过滤器（通常是 2x2 的）和一个同样长度的步幅。然后把它应用到输入内容上，输出过滤器卷积计算的每个子区域中的最大数字。





Example of Maxpool with a 2x2 filter and a stride of 2

带有 2×2 和过滤器的旦步幅为 2 的最大池化的例子

池化层还有其他选择，比如平均池化（average pooling）和 L2-norm 池化 。这一层背后的直观推理是：一旦我们知道了原始输入（这里会有一个高激活值）中一个特定的特征，它与其它特征的相对位置就比它的绝对位置更重要。可想而知，这一层大幅减小了输入卷的空间维度（长度和宽度改变了，但深度没变）。这到达了两个主要目的。第一个是权重参数的数目减少到了75%，因此降低了计算成本。第二是它可以控制过拟合（overfitting）。这个术语是指一个模型与训练样本太过匹配了，以至于用于验证和检测组时无法产生出好的结果。出现过拟合的表现是一个模型在训练集能达到 100% 或 99% 的准确度，而在测试数据上却只有50%。

Dropout 层

如今，Dropout 层在神经网络有了非常明确的功能。上一节，我们讨论了经过训练后的过拟合问题：训练之后，神经网络的权重与训练样本太过匹配以至于在处理新样本的时候表现平平。Dropout 的概念在本质上非常简单。Dropout 层将「丢弃（drop out）」该层中一个随机的激活参数集，即在前向通过（forward pass）中将这些激活参数集设置为 0。简单如斯。既然如此，这些简单而且似乎不必要且有些反常的过程的好处是什么？在某种程度上，这种机制强制网络变得更加冗余。这里的意思是：该网络将能够为特定的样本提供合适的分类或输出，即使一些激活参数被丢弃。此机制将保证神经网络不会对训练样本「过于匹配」，这将帮助缓解过拟合问题。另外，Dropout 层只能在训练中使用，而不能用于测试过程，这是很重要的一点。

- 参考 Geoffrey Hinton 的论文：Dropout: A Simple Way to Prevent Neural Networks from Overfitting

网络层中的网络

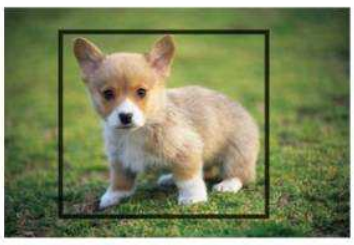
网络层中的网络指的是一个使用了 1 x 1 尺寸的过滤器的卷积层。现在，匆匆一瞥，你或许会好奇为何这种感受野大于它们所映射空间的网络层竟然会有帮助。然而，我们必须谨记 1x1 的卷积层跨越了特定深度，所以我们可以设想一个1 x 1 x N 的卷积层，此处 N 代表该层应用的过滤器数量。该层有效地使用 N 维数组元素依次相乘的乘法，此时 N 代表的是该层的输入的深度。

- 参阅 Min Lin 的论文：Network In Network

分类、定位、检测、分割

本系列第一部分使用的案例中，我们观察了图像分类任务。这个过程是：获取输入图片，输出一套分类的类数（class number）。然而当我们执行类似目标定位的任务时，我们要做的不只是生成一个类标签,而是生成一个描述图片中物体suo所在位置的边界框。

Object Classification is the task of identifying that picture is a dog

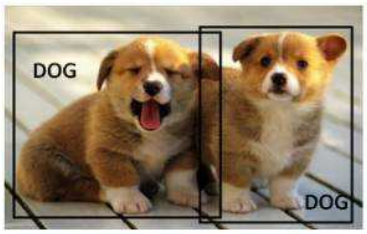


Object Localization involves the class label as well as a bounding box to show where the object is located.

我们也有目标检测的任务，这需要图片上所有目标的定位任务都已完成。

因此，你将获得多个边界框和多个类标签。

最终，我们将执行目标分割的任务：我们需要输出类标签的同时输出图片中每个目标的轮廓。



Object Detection involves localization of multiple objects (doesn't have to be the same class).

Object Segmentation involves the class label as well as an outline of the object in interest.

关于目标检测、定位、分割的论文有很多，这里就不一一列出了。

迁移学习

如今，深度学习领域一个常见的误解在于没有谷歌那样的巨量数据，你将没有希望创建一个有效的深度学习模型。尽管数据是创建网络中至关重要的部分，迁移学习的思路将帮助我们降低数据需求。迁移学习指的是利用预训练模型（神经网络的权重和参数都已经被其他人利用更大规模的数据集训练好了）并用自己的数据集将模型「微调」的过程。这种思路中预训练模型扮演着特征提取器的角色。你将移除网络的最后一层并用你自有的分类器置换（取决于你的问题空间）。然后冻结其他所有层的权重并正常训练该网络（冻结这些层意味着在梯度下降/最优化过程中保持权值不变）。

让我们探讨一下为什么做这项工作。比如说我们正在讨论的这个预训练模型是在 ImageNet （一个包含一千多个分类，一千四百万张图像的数据集）上训练的 。当我们思考神经网络的较低层时，我们知道它们将检测类似曲线和边缘这样的特征。现在，除非你有一个极为独特的问题空间和数据集，你的神经网络也会检测曲线和边缘这些特征。相比通过随机初始化权重训练整个网络，我们可以利用预训练模型的权重（并冻结）聚焦于更重要的层（更高层）进行训练。如果你的数据集不同于 ImageNet 这样的数据集，你必须训练更多的层级而只冻结一些低层的网络。

- Yoshua Bengio （另外一个深度学习先驱 ）论文：How transferable are features in deep neural networks?
- Ali Sharif Razavian 论文：CNN Features off-the-shelf: an Astounding Baseline for Recognition
- Jeff Donahue 论文：DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition

数据增强技术

现在我们对卷积网络中数据的重要性可能已经感到有些麻木了，所以我们来谈下如何利用一些简单的转换方法将你现有的数据集变得更大。正如我们之前所提及的，当计算机将图片当作输入时，它将用一个包含一列像素值的数组描述（这幅图）。若是图片左移一个像素。对你和我来说，这种变化是微不可察的。然而对计算机而已，这种变化非常显著：这幅图的类别和标签保持不变，数组却变化了。这种改变训练数据的数组表征而保持标签不变的方法被称作数据增强技术。这是一种人工扩展数据集的方法。人们经常使用的增强方法包括灰度变化、水平翻转、垂直翻转、随机编组、色值跳变、翻译、旋转等其他多种方法。通过利用这些训练数据的转换方法，你将获得两倍甚至三倍于原数据的训练样本。

深度 | 从入门到精通：卷积神经网络初学者指南（附论文）

CNN(卷积神经网络)是什么？有入门简介或文章吗？

YJango
日本会津大学 人机界面实验室博士在读

该文是**卷积神经网络--介绍**，并假设你理解前馈神经网络。

如果不是，强烈建议你读完**如何简单形象又有趣地讲解神经网络是什么？** 后再来读该篇。

目录

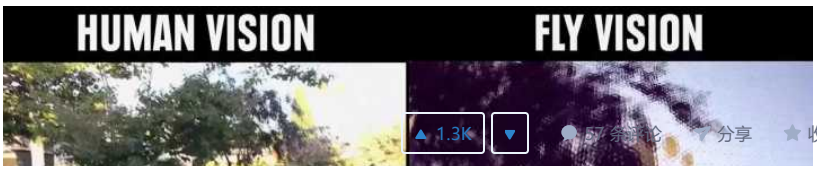
视觉感知

一、画面识别是什么任务？

学习知识的第一步就是**明确任务**，清楚该知识的输入输出。卷积神经网络最初是服务于画面识别的，所以我们先来看看画面识别的实质是什么。

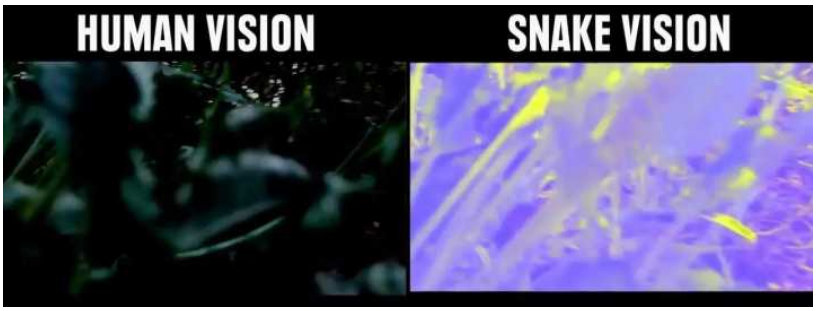
先观看几组动物与人类视觉的差异对比图。

1. 苍蝇的视觉和人的视觉的差异



CNN(卷积神经网络)是什么？有入门简介或文章吗？

2. 蛇的视觉和人的视觉的差异



（更多对比图请参考链接）

通过上面的两组对比图可以知道，即便是相同的图片经过不同的视觉系统，也会得到不同的感知。

这里引出一条知识：生物所看到的景象并非世界的原貌，而是长期进化出来的**适合自己生存环境的一种感知方式。** 蛇的猎物一般是夜间行动，所以它就进化出了一种可以在夜间也能很好观察的感知系统，感热。

任何视觉系统都是将图像反光与脑中所看到的概念进行关联。

所以画面识别实际上并非识别这个东西客观上是什么，而是寻找人类的视觉关联方式，并再次应用。 如果我们不是人类，而是蛇类，那么画面识别所寻找的 就和现在的不一样。

> **画面识别实际上是寻找（学习）人类的视觉关联方式，并再次应用。**

二、图片被识别成什么取决于哪些因素?

下面用两张图片来体会识别结果取决于哪些因素。

1. 老妇与少女

CNN(卷积神经网络)是什么？有入门简介或文章吗?

请观察上面这张图片，你看到的是老妇还是少女？ 以不同的方式去观察这张图片会得出不同的答案。 图片可以观察成有大鼻子、大眼睛的老妇。也可以被观察成少女，但这时老妇的嘴会被识别成少女脖子上的项链，而老妇的眼睛则被识别为少女的耳朵。

2. 海豚与男女

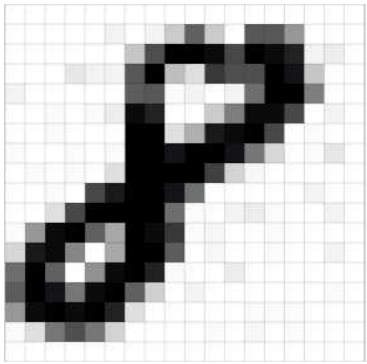
上面这张图片如果是成人观察，多半看到的会是一对亲热的男女。倘若儿童看到这张图片，看到的则会是一群海豚（男女的轮廓是由海豚构造出的）。所以，识别结果受年龄，文化等因素的影响，换句话说：

> **图片被识别成什么不仅仅取决于图片本身，还取决于图片是如何被观察的。**

图像表达

我们知道了“画面识别是从大量的 (x, y) 数据中寻找人类的视觉关联方式 ，并再次应用。 其 x -是输入，表示所看到的东西 y -输出，表示该东西是什么。

在自然界中， x 是物体的反光，那么在计算机中，图像又是如何被表达和存储的呢?



[from]

图像在计算机中是一堆按顺序排列的数字，数值为0到255。0表示最暗，255表示最亮。 你可以把这堆数字用一个长长的向量来表示，也就是tensorflow的mnist教程中784维向量的表示方式。 然而这样会失去平面结构的信息，为保留该结构信息，通常选择矩阵的表示方式：28x28的矩阵。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

上图是只有黑白颜色的灰度图，而更普遍的图片表达方式是RGB颜色模型，即红（Red）、绿（Green）、蓝（Blue）三原色的色光以不同的比例相加，以产生多种多样的色光。

这样，RGB颜色模型中，单个矩阵就扩展成了有序排列的三个矩阵，也可以用三维张量去理解，其中的每一个矩阵又叫这个图片的一个channel。

在电脑中，一张图片是数字构成的“长方体”。可用 宽width, 高height, 深depth 来描述，如图。

> 画面识别的输入 x 是shape为(width, height, depth)的三维张量。

接下来要考虑的就是该如何处理这样的“数字长方体”。

画面不变性

在决定如何处理“数字长方体”之前，需要清楚所建立的网络拥有什么样的特点。 我们知道一个物体不管在画面左侧还是右侧，都会被识别为同一物体，这一特点就是不变性（invariance），如下图所示。

我们希望所建立的网络可以尽可能的满足这些不变性特点。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

为了理解卷积神经网络对这些不变性特点的贡献，我们将用不具备这些不变性特点的前馈神经网络来进行比较。

图片识别--前馈神经网络

方便起见，我们用depth只有1的灰度图来举例。 想要完成的任务是：在宽长为4x4的图片中识别是否有下图所示的“横折”。 图中，黄色圆点表示值为0的像素，深色圆点表示值为1的像素。 我们知道不管这个横折在图片中的什么位置，都会被认为是相同的横折。

若训练前馈神经网络来完成该任务，那么表达图像的三维张量将会被摊平成一个向量，作为网络的输入，即(width, height, depth)为(4, 4, 1)的图片会被展成维度为16的向量作为网络的输入层。再经过几层不同节点个数的隐藏层，最终输出两个节点，分别表示“有横折的概率”和“没有横折的概率”，如下图所示。

下面我们用数字（16进制）对图片中的每一个像素点（pixel）进行编号。 当使用右侧那种物体位于中间的训练数据来训练网络时，网络就只会对编号为5,6,9,a的节点的权重进行调节。 若让该网络识别位于右下角的“横折”时，则无法识别。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

解决办法是用大量物体位于不同位置的数据训练，同时增加网络的隐藏层个数从而扩大网络学习这些变体的能力。

然而这样做十分不效率，因为我们知道在左侧的“横折”也好，还是在右侧的“横折”也罢，大家都是“横折”。 为什么相同的东西在位置变了之后要重新学习？有没有什么方法可以将中间所学到的规律也运用在其他的位置？ 换句话说，也就是**让不同位置用相同的权重**。

图片识别--卷积神经网络

卷积神经网络就是让权重在不同位置共享的神经网络。

局部连接

在卷积神经网络中，我们先选择一个局部区域，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。

为了更好的和前馈神经网络做比较，我将这些以矩阵排列的节点展成了向量。 下图展示了被红色方框所圈中编号为0,1,4,5的节点是如何通过 w_1, w_2, w_3, w_4 连接到下一层的节点0上的。

这个带有连接强弱的红色方框就叫做 **filter** 或 **kernel** 或 **feature detector**。 而filter的范围叫做 **filter size**，这里所展示的是2x2的filter size。

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \quad (1)$$

第二层的节点0的数值就是局部区域的线性组合，即被圈中节点的数值乘以对应的权重后相加。 用 x 表示输入值， y 表示输出值，用图中标注数字表示角标，则下面列出了两种计算编号为0的输出值 y_0 的表达式。

注：在局部区域的线性组合后，也会和前馈神经网络一样，加上一个偏移量 b_0。

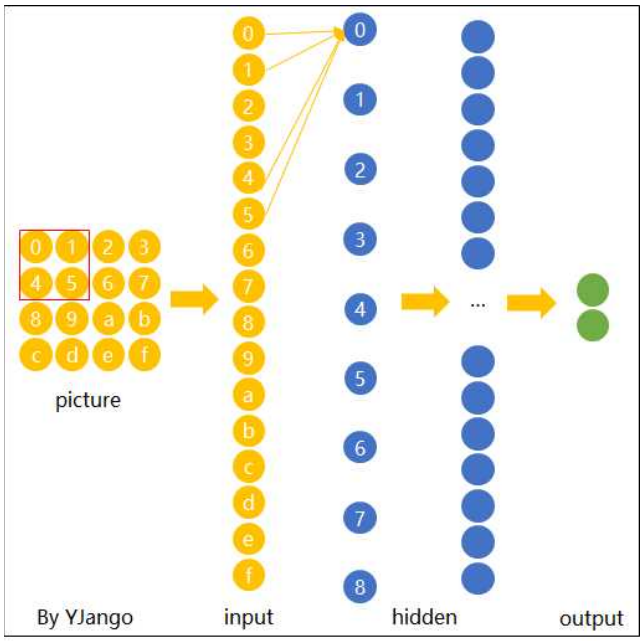
$$y_0 = x_0 * w_1 + x_1 * w_2 + x_4 * w_3 + x_5 * w_4 + b_0$$

$$y_0 = [\, w_1 \quad w_2 \quad w_3 \quad w_4 \,] \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_4 \\ x_5 \end{bmatrix} + b_0$$

空间共享

当filter扫到其他位置计算输出节点 y_i 时， w_1, w_2, w_3, w_4 ，**包括** b_0 是共用的。

下面这张动态图展示了当filter扫过不同区域时，节点的链接方式。 动态图的最后一帧则显示了所有连接。 可以注意到，每个输出节点并非像前馈神经网络中那样与全部的输入节点连接，而是部分连接。 这也就是为什么大家也叫前馈神经网络（feedforward neural network）为fully-connected neural network。 图中显示的是一步一步的移动filter来扫描全图，一次移动多少叫做stride。



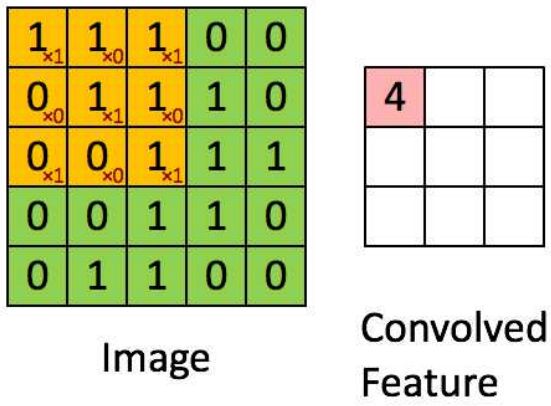
> **空间共享也就是卷积神经网络所引入的先验知识。**

输出表达

如先前在图像表达中提到的，图片不用向量去表示是为了保留图片平面结构的信息。 同样的，卷积后的输出若用上图的排列方式则丢失了平面结构信息。 所以我们依然用矩阵的方式排列它们，就得到了下图所展示的连接。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

这也就是你们在网上所看到的下面这张图。在看这张图的时候请结合上图的连接一起理解，即输入（绿色）的每九个节点连接到输出（粉红色）的一个节点上的。



经过一个feature detector计算后得到的粉红色区域也叫做一个“**Convolved Feature**” 或 “**Activation Map**” 或 “**Feature Map**”。

Depth维的处理

现在我们已经知道了depth维度只有1的灰度图是如何处理的。 但前文提过，图片的普遍表达方式是下图这样有3个channels的RGB颜色模型。 当depth为复数的时候，每个feature detector是如何卷积的?

现象：2x2所表达的filter size中，一个2表示width维上的局部连接数，另一个2表示height维上的局部连接数，并却没有depth维上的局部连接数，是因为depth维上并非局部，而是全部连接的。

在2D卷积中，filter在张量的width维, height维上是局部连接，在depth维上是贯串全部channels的。

类比：想象在切蛋糕的时候，不管这个蛋糕有多少层，通常大家都会一刀切到底，但是在长和宽这两个维上是局部切割。

下面这张图展示了，在depth为复数时，filter是如何连接输入节点到输出节点的。 图中红、绿、蓝颜色的节点表示3个channels。 黄色节点表示一个feature detector卷积后得到的Feature Map。 其中被透明黑框圈中的12个节点会被连接到黄黑色的节点上。

- 在输入depth为1时：被filter size为2x2所圈中的4个输入节点连接到1个输出节点上。
- 在输入depth为3时：被filter size为2x2，但是贯串3个channels后，所圈中的12个输入节点连接到1个输出节点上。
- 在输入depth为 n 时：2x2x n 个输入节点连接到1个输出节点上。

(可从vectary在3D编辑下查看)

注意：三个channels的权重并不共享。 即当深度变为3后，权重也跟着扩增到了三组，如式子(3)所示，不同channels用的是自己的权重。 式子中增加的角标r,g,b分别表示red channel, green channel, blue channel的权重。

$$\begin{bmatrix} w_{r1} & w_{r2} \\ w_{r3} & w_{r4} \end{bmatrix}, \begin{bmatrix} w_{g1} & w_{g2} \\ w_{g3} & w_{g4} \end{bmatrix}, \begin{bmatrix} w_{b1} & w_{b2} \\ w_{b3} & w_{b4} \end{bmatrix} \quad (3)$$

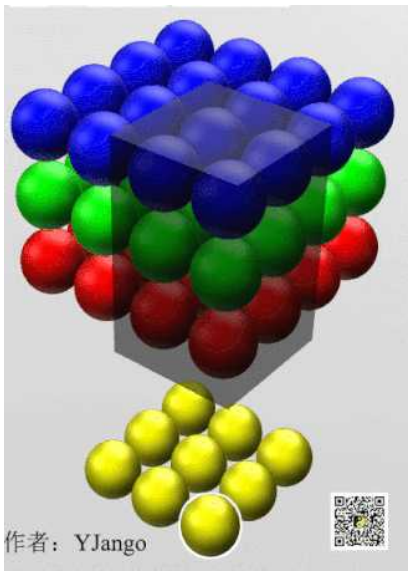
计算例子：用 x_{r0} 表示red channel的编号为0的输入节点， x_{g5} 表示green channel编号为5个输入节点。 x_{b1} 表示blue channel。如式子(4)所表达，这时的一个输出节点实际上是12个输入节点的线性组合。

$$y_0 = x_{r0} * w_{r1} + x_{r1} * w_{r2} + x_{r4} * w_{r3} + x_{r5} * w_{r4} + x_{g0} * w_{g1} + x_{g1} * w_{g2} + x_{g4} * w_{g3} + x_{g5} * w_{g4} + x_{b0} * w_{b1} + x_{b1} * w_{b2} + x_{b4} * w_{b3} + x_{b5} * w_{b4} + b_0$$

$$y_0 = \begin{bmatrix} w_{r1} & w_{r2} & w_{r3} & w_{r4} \end{bmatrix} \cdot \begin{bmatrix} x_{r0} \\ x_{r1} \\ x_{r4} \\ x_{r5} \end{bmatrix} + \begin{bmatrix} w_{g1} & w_{g2} & w_{g3} & w_{g4} \end{bmatrix} \cdot \begin{bmatrix} x_{g0} \\ x_{g1} \\ x_{g4} \\ x_{g5} \end{bmatrix} + \begin{bmatrix} w_{b1} & w_{b2} & w_{b3} & w_{b4} \end{bmatrix} \cdot \begin{bmatrix} x_{b0} \\ x_{b1} \\ x_{b4} \\ x_{b5} \end{bmatrix} + b_0$$

(4)

当filter扫到其他位置计算输出节点 y_i 时，那12个权重在不同位置是共用的，如下面的动态图所展示。 透明黑框圈中的12个节点会连接到被白色边框选中的黄色节点上。



> **每个filter会在width维, height维上，以局部连接和空间共享，并贯串整个depth维的方式得到一个Feature Map。**

Zero padding

细心的读者应该早就注意到了，4x4的图片被2x2的filter卷积后变成了3x3的图片，每次卷积后都会小一圈的话，经过若干层后岂不是变的越来越小？ Zero padding就可以在这时帮助控制Feature Map的输出尺寸，同时避免了边缘信息被一步步舍弃的问题。

例如：下面4x4的图片在边缘Zero padding一圈后，再用3x3的filter卷积后，得到的Feature Map尺寸依然是4x4不变。

通常大家都想要在卷积时保持图片的原始尺寸。 选择3x3的filter和1的zero padding，或5x5的filter和2的zero padding可以保持图片的原始尺寸。 这也是为什么大家多选择3x3和5x5的filter的原因。 另一个原因是3x3的filter考虑到了像素与其距离为1以内的所有其他像素的关系，而5x5则是考虑像素与其距离为2以内的所有其他像素的关系。

尺寸：Feature Map的尺寸等于(input_size + 2 * padding_size − filter_size)/stride+1。

注意：上面的式子是计算width或height一维的。padding_size也表示的是单边补零的个数。例如(4+2-3)/1+1 = 4，保持原尺寸。

不用去背这个式子。其中(input_size + 2 * padding_size)是经过Zero padding扩充后真正要卷积的尺寸。 减去 filter_size后表示可以滑动的范围。 再除以可以一次滑动（stride）多少后得到滑动了多少次，也就意味着得到了多少个输出节点。 再加上第一个不需要滑动也存在的输出节点后就是最后的尺寸。

形状、概念抓取

知道了每个filter在做什么之后，我们再来思考这样的一个filter会抓取到什么样的信息。

我们知道不同的形状都可由细小的“零件”组合而成的。比如下图中，用2x2的范围所形成的16种形状可以组合成格式各样的“更大”形状。

卷积的每个filter可以探测特定的形状。又由于Feature Map保持了抓取后的空间结构。若将探测到细小图形的Feature Map作为新的输入再次卷积后，则可以由此探测到“更大”的形状概念。 比如下图的第一个“大”形状可由2,3,4,5基础形状拼成。第二个可由2,4,5,6组成。第三个可由6,1组成。

除了基础形状之外，颜色、对比度等概念对画面的识别结果也有影响。卷积层也会根据需要去探测特定的概念。

可以从下面这张图中感受到不同数值的filters所卷积过后的Feature Map可以探测边缘，棱角，模糊，突出等概念。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

[from]

如我们先前所提，图片被识别成什么不仅仅取决于图片本身，还取决于图片是如何被观察的。

而filter内的权重矩阵W是网络根据数据学习得到的，也就是说，我们让神经网络自己学习以什么样的方式去观察图片。

拿老妇与少女的那幅图片举例，当标签是少女时，卷积网络就会学习抓取可以成少女的形状、概念。 当标签是老妇时，卷积网络就会学习抓取可以成老妇的形状、概念。

下图展现了在人脸识别中经过层层的卷积后，所能够探测的形状、概念也变得越来越抽象和复杂。

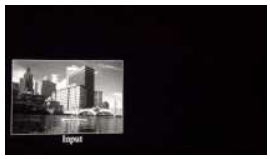
> **卷积神经网络会尽可能寻找最能解释训练数据的抓取方式。**

多filters

每个filter可以抓取探测特定的形状的存在。 假如我们要探测下图的长方框形状时，可以用4个filters去探测4个基础“零件”。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

因此我们自然而然的会选择用多个不同的filters对同一个图片进行多次抓取。 如下图（动态图过大，如果显示不出，请看到该链接观看），同一个图片，经过两个（红色、绿色）不同的filters扫描过后可得到不同特点的Feature Maps。 每增加一个filter，就意味着你想让网络多抓取一个特征。



[from]

这样卷积层的输出也不再是depth为1的一个平面，而是和输入一样是depth为复数的长方体。

如下图所示，当我们增加一个filter（紫色表示）后，就又可以得到一个Feature Map。 将不同filters所卷积得到的Feature Maps按顺序堆叠后，就得到了一个卷积层的最终输出。

> **卷积层的输入是长方体，输出也是长方体。**

这样卷积后输出的长方体可以作为新的输入送入另一个卷积层中处理。

加入非线性

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。

将卷积所得的Feature Map经过ReLU变换（elementwise）后所得到的output就如下图所展示。

[from]

输出长方体

现在我们知道了一个卷积层的输出也是一个长方体。 那么这个输出长方体的(width, height, depth)由哪些因素决定和控制。

这里直接用CS231n的Summary：

计算例子：请体会CS231n的Convolution Demo部分的演示。

矩阵乘法执行卷积

如果按常规以扫描的方式一步步计算局部节点和filter的权重的点乘，则不能高效的利用GPU的并行能力。 所以更普遍的方法是用两个大矩阵的乘法来一次性囊括所有计算。

因为卷积层的每个输出节点都是由若干个输入节点的线性组合所计算。 因为输出的节点个数是 $W_2 \times H_2 \times D_2$ ，所以就有 $W_2 \times H_2 \times D_2$ 个线性组合。

读过我写的线性代数教程的读者请回忆，矩阵乘矩阵的意义可以理解为批量的线性组合按顺序排列。 其中一个矩阵所表示的信息是多组权重，另一个矩阵所表示的信息是需要进行组合的向量。 大家习惯性的把组成成分放在矩阵乘法的右边，而把权重放在矩阵乘法的左边。 所以这个大型矩阵乘法可以用 $W_{row} \cdot X_{col}$ 表示，其中 W_{row} 和 X_{col} 都是矩阵。

卷积的每个输出是由局部的输入节点和对应的filter权重展成向量后所计算的，如式子(2)。 那么 W_{row} 中的每一行则是每个filter的权重，有 $F \cdot F \cdot D_1$ 个； 而 X_{col} 的每一列是所有需要进行组合的节点（上面的动态图中被黑色透明框圈中的节点），也有 $F \cdot F \cdot D_1$ 个。 X_{col} 的列的个数则表示每个filter要滑动多少次才可以把整个图片扫描完，有 $W_2 \cdot H_2$ 次。 因为我们有多个filters， W_{row} 的行的个数则是filter的个数 K 。

最后我们得到：

$W_{row} \in R^{K \times F \cdot F \cdot D_1}$

$X_{col} \in R^{F \cdot F \cdot D_1 \times W_2 \cdot H_2}$

$W_{row} \cdot X_{col} \in R^{K \times W_2 \cdot H_2}$

当然矩阵乘法后需要将 $W_{row} \cdot X_{col}$ 整理成形状为 $W_2 \times H_2 \times D_2$ 的三维张量以供后续处理（如再送入另一个卷积层）。 X_{col} 则也需要逐步的局部滑动图片，最后堆叠构成用于计算矩阵乘法的形式。

Max pooling

在卷积后还会有一个pooling的操作，尽管有其他的比如average pooling等，这里只提max pooling。

max pooling的操作如下图所示：整个图片被不重叠的分割成若干个同样大小的小块（pooling size）。每个小块内只取最大的数字，再舍弃其他节点后，保持原有的平面结构得出output。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

[from]

max pooling在不同的depth上是分开执行的，且不需要参数控制。 那么问题就max pooling有什么作用？部分信息被舍弃后难道没有影响吗？

[from]

Max pooling的主要功能是downsampling，却不会损坏识别结果。 这意味着卷积后的Feature Map中有对于识别物体不必要的冗余信息。 那么我们就反过来思考，这些“冗余”信息是如何产生的。

直觉上，我们为了探测到某个特定形状的存在，用一个filter对整个图片进行逐步扫描。但只有出现了该特定形状的区域所卷积获得的输出才是真正有用的，用该filter卷积其他区域得出的数值就可能对该形状是否存在的判定影响较小。 比如下图中，我们还是考虑探测“横折”这个形状。 卷积后得到3x3的Feature Map中，真正有用的就是数字为3的那个节点，其余数值对于这个任务而言都是无关的。 所以用3x3的Max pooling后，并没有对“横折”的探测产生影响。 试想在这里例子中如果不使用Max pooling，而让网络自己去学习。 网络也会去学习与Max pooling近似效果的权重。因为是近似效果，增加了更多的parameters的代价，却还不如直接进行Max pooling。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

Max pooling还有类似“选择句”的功能。假如有两个节点，其中第一个节点会在某些输入情况下最大，那么网络就只在这个节点上流通信息；而另一些输入又会让第二个节点的值最大，那么网络就转而走这个节点的分支。

但是Max pooling也有不好的地方。因为并非所有的抓取都像上图的极端例子。有些周边信息对某个概念是否存在的判定也有影响。 并且Max pooling是对所有的Feature Maps进行等价的操作。就好比用相同网孔的渔网打鱼，一定会有漏网之鱼。

全连接层

当抓取到足以用来识别图片的特征后，接下来的就是如何进行分类。 全连接层（也叫前馈层）就可以用来将最后的输出映射到线性可分的空间。 通常卷积网络的最后会将末端得到的长方体平摊(flatten)成一个长长的向量，并送入全连接层配合输出层进行分类。

卷积神经网络大致就是covolutional layer, pooling layer, ReLu layer, fully-connected layer的组合，例如下图所示的结构。

[from]

这里也体现了深层神经网络或deep learning之所以称deep的一个原因：模型将特征抓取层和分类层合在了一起。 负责特征抓取的卷积层主要是用来学习“如何观察”。

下图简述了机器学习的发展，从最初的人工定义特征再放入分类器的方法，到让机器自己学习特征，再到如今尽量减少人为干涉的deep learning。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

[from]

结构发展

以上介绍了卷积神经网络的基本概念。 以下是几个比较有名的卷积神经网络结构，详细的请看CS231n。

- **LeNet**：第一个成功的卷积神经网络应用
- **AlexNet**：类似LeNet，但更深更大。使用了层叠的卷积层来抓取特征（通常是一个卷积层马上一个max pooling层）
- **ZF Net**：增加了中间卷积层的尺寸，让第一层的stride和filter size更小。
- **GoogLeNet**：减少parameters数量，最后一层用max pooling层代替了全连接层，更重要的是Inception-v4模块的使用。
- **VGGNet**：只使用3x3 卷积层和2x2 pooling层从头到尾堆叠。
- **ResNet**：引入了跨层连接和batch normalization。
- **DenseNet**：将跨层连接从头进行到尾。

总结一下：这些结构的发展趋势有：

- 使用small filter size的卷积层和pooling
- 去掉parameters过多的全连接层
- Inception（稍后会对其中的细节进行说明）
- 跳层连接

不变性的满足

接下来会谈谈我个人的，对于画面不变性是如何被卷积神经网络满足的想法。 同时结合不变性，对上面提到的结构发展的重要变动进行直觉上的解读。

需要明白的是为什么加入不变性可以提高网络表现。 并不是因为我们用了更炫酷的处理方式，而是加入了先验知识，无需从零开始用数据学习，节省了训练所需数据量。 思考表现提高的原因一定要从训练所需要的数据量切入。 提出满足新的不变性特点的神经网络是计算机视觉的一个主要研究方向。

平移不变性

可以说卷积神经网络最初引入局部连接和空间共享，就是为了满足平移不变性。

因为空间共享，在不同位置的同一形状就可以被等价识别，所以不需要对每个位置都进行学习。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

旋转和视角不变性

个人觉得卷积神经网络克服这一不变性的主要手段还是靠大量的数据。 并没有明确加入“旋转和视角不变性”的先验特性。

Deformable Convolutional Networks似乎是对此变性进行了进行增强。

尺寸不变性

与平移不变性不同，最初的卷积网络并没有明确照顾尺寸不变性这一特点。

我们知道filter的size是事先选择的，而不同的尺寸所寻找的形状（概念）范围不同。

从直观上思考，如果选择小范围，再一步步通过组合，仍然是可以得到大范围的形状。 如3x3尺寸的形状都是可以由2x2形状的图形组合而成。所以形状的尺寸不变性对卷积神经网络而言并不算问题。 这恐怕ZF Net让第一层的stride和filter size更小，VGGNet将所有filter size都设置成3x3仍可以得到优秀结果的一个原因。

但是，除了形状之外，很多概念的抓取通常需要考虑一个像素与周边更多像素之间的关系后得出。也就是说5x5的filter也是有它的优点。 同时，小尺寸的堆叠需要很多个filters来共同完成，如果需要抓取的形状恰巧在5x5的范围，那么5x5会比3x3来的更有效率。 所以一次性使用多个不同filter size来抓取多个范围不同的概念是一种顺理成章的想法，而这个也就是Inception。 可以说Inception是为了尺寸不变性而引入的一个先验知识。

Inception

下图是Inception的结构，尽管也有不同的版本，但是其动机都是一样的：消除尺寸对于识别结果的影响，一次性使用多个不同filter size来抓取多个范围不同的概念，并让网络自己选择需要的特征。

你也一定注意到了蓝色的1x1卷积，撇开它，先看左边的这个结构。

输入（可以是被卷积完的长方体输出作为该层的输入）进来后，通常我们可以选择直接使用像素信息(1x1卷积)传递到下一层，可以选择3x3卷积，可以选择5x5卷积，还可以选择max pooling的方式downsample刚被卷积后的feature maps。 但在实际的网络设计中，究竟该如何选择需要大量的实验和经验的。 Inception就不用我们来选择，而是将4个选项给神经网络，让网络自己去选择最合适的解决方案。

接下来我们再看右边的这个结构，多了很多蓝色的1x1卷积。 这些1x1卷积的作用是为了让网络根据需要能够更灵活的控制数据的depth的。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

1x1卷积核

如果卷积的输出输入都只是一个平面，那么1x1卷积核并没有什么意义，它是完全不考虑像素与周边其他像素关系。 但卷积的输出输入是长方体，所以1x1卷积实际上是对每个像素点，在不同的channels上进行线性组合（信息整合），且保留了图片的原有平面结构，调控depth，从而完成升维或降维的功能。

如下图所示，如果选择2个filters的1x1卷积层，那么数据就从原本的depth 3 降到了2。若用4个filters，则起到了升维的作用。

这就是为什么上面Inception的4个选择中都混合一个1x1卷积，如右侧所展示的那样。 其中，绿色的1x1卷积本身就1x1卷积，所以不需要再用另一个1x1卷积。 而max pooling用来去掉卷积得到的Feature Map中的冗余信息，所以出现在1x1卷积之前，紧随刚被卷积后的feature maps。（由于没做过实验，不清楚调换顺序会有什么影响。）

跳层连接

前馈神经网络也好，卷积神经网络也好，都是一层一层逐步变换的，不允许跳层组合。 但现实中是否有跳层组合的现象?

比如说我们在判断一个人的时候，很多时候我们并不是观察它的全部，或者给你的图片本身就是残缺的。 这时我们会靠单个五官，外加这个人的着装，再加他的身形来综合判断这个人，如下图所示。 这样，即便图片本身是残缺的也可以很好的判断它是什么。 这和前馈神经网络的先验知识不同，它允许不同层级之间的因素进行信息交互、综合判断。

残差网络就是拥有这种特点的神经网络。大家喜欢用identity mappings去解释为什么残差网络更优秀。 这里我只是提供了一个以先验知识的角度去理解的方式。 需要注意的是每一层并不会像我这里所展示的那样，会形成明确的五官层。 只是有这样的组合趋势，实际无法保证神经网络到底学到了什么内容。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

用下图举一个更易思考的例子。 图形1,2,3,4,5,6是第一层卷积层抓取到的概念。 图形7,8,9是第二层卷积层抓取到的概念。 图形7,8,9是由1,2,3,4,5,6的基础上组合而成的。

但当我们想要探测的图形10并不是单纯的靠图形7,8,9组成，而是第一个卷积层的图形6和第二个卷积层的8,9组成的话，不允许跨层连接的卷积网络不得不用更多的filter来保持第一层已经抓取到的图形信息。并且每次传递到下一层都需要学习那个用于保留前一层图形概念的filter的权重。 当层数变深后，会越来越难以保持，还需要max pooling将冗余信息去掉。

一个合理的做法就是直接将上一层所抓取的概念也跳层传递给下下一层，不用让其每次都重新学习。 就好比在编程时构建了不同规模的functions。 每个function我们都是保留，而不是重新再写一遍。提高了重用性。

同时，因为ResNet使用了跳层连接的方式。也不需要max pooling对保留低层信息时所产生的冗余信息进行去除。

Inception中的第一个1x1的卷积通道也有类似的作用，但是1x1的卷积仍有权重需要学习。 并且Inception所使用的结合方式是concatenate的合并成一个更大的向量的方式，而ResNet的结合方式是sum。 两个结合方式各有优点。 concatenate当需要用不同的维度去组合成新观念的时候更有益。 而sum则更适用于并存的判断。比如既有油头发，又有胖身躯，同时穿着常年不洗的牛仔裤，三个不同层面的概念并存时，该人会被判定为程序员的情况。 又比如双向LSTM中正向和逆向序列抓取的结合常用相加的方式结合。在语音识别中，这表示既可以正向抓取某种特征，又可以反向抓取另一种特征。当两种特征同时存在时才会被识别成某个特定声音。

在下图的ResNet中，前一层的输入会跳过部分卷积层，将底层信息传递到高层。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

在下图的DenseNet中，底层信息会被传递到所有的后续高层。

后续

随着时间推移，各个ResNet,GoogLeNet等框架也都在原有的基础上进行了发展和改进。 但基本都是上文描述的概念的组合使用加上其他的tricks。

如下图所展示的，加入跳层连接的Inception-ResNet。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

但对我而言，

> **真正重要的是这些技巧对于各种不变性的满足。**

深度学习通俗易懂教程专栏超智能体 - 知乎专栏

阅读列表：

1. **深层神经网络：**深层学习为何要“Deep”（上）（由于下篇写的并不通俗，不推荐阅读，用公开课代替）
2. **反向传播算法实例：**未编写
3. **深度学习总览：**公开课：深层神经网络设计理念
4. **深度学习入门误区：**知乎Live（公开课涵盖了Live的内容，若觉得作者辛苦也可参加。算了，还是不要参加了！）
5. **Tensorflow：**TensorFlow整体把握
6. **前馈神经网络（1）：**前馈神经网络--代码LV1
7. **前馈神经网络（2）：**前馈神经网络--代码LV2
8. **前馈神经网络（3）：**前馈神经网络--代码LV3
9. **循环神经网络（1）：**循环神经网络--介绍
10. **循环神经网络（2）：**循环神经网络--实现LSTM
11. **循环神经网络（3）：**循环神经网络--scan实现LSTM
12. **循环神经网络（4）：**循环神经网络--双向GRU
13. **卷积神经网络（1）：**卷积神经网络--介绍

编辑于 2017-10-19

许铁-巡洋舰科技
微信公众号请关注chaoscruiser ,铁哥个人微信号ironcruiser

87 人赞同了该回答

应该说， CNN是这两年深度学习风暴的罪魁祸首， 自2012年， 正是它让打入冷宫的神经网络重见天日并且建立起自己在人工智能王国的霸主地位。

如过你认为深度学习是只能用来理解图像的，你就大错特错了， 因为它的用途太广了，上至文字，中有图像， 下至音频， 从手写数字识别到大名鼎鼎的GAN对抗学习， 都离不开它。

不过要了解CNN，还是拿图像做例子比较恰当。一句话来说CNN图像处理的本质，就是信息抽取，巨大的网络可以抽取一步步得到最关键的图像特征， 我们有时也叫自动的特征工程。

CNN的建造灵感来自于人类对视觉信息的识别过程。 人脑对物体的识别的第一个问题是： 对应某一类对象的图像千千万， 比如一个苹果， 就有各种状态的成千上万状态， 我们识别物体的类别，事实上是给这成千上万不同的图片都打上同一个标签。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

物理里管这种一个事物的结果与一些列的变化都没有关的特性，叫不变性，比如如果你转动一个苹果任何一个角度它都是苹果， 这就是苹果有旋转不变性，但是数字6就不行，如果你给它旋转特定角度它就变成9了， 它就没有旋转不变性。

我们人通常可以无视这些变化认出事物来，也就是把和这种变化有关的信息忽略。如果我们对图像进行识别， 事实上我们的算法就要有人的这种本领， 首先让它学会什么东西与真实的物体信息是无关的。

就拿数字识别举个例子吧， 一个数字是什么，虽然与旋转的角度有关系， 但与它在图片中的上下左右没关系， 我们管这种不变性叫平移不变性。

解决这个问题，最粗暴的一个方法是制造很多的样本，比如把“1” 放在很多不同的位置，然后让机器在错误中学习。 然后穷尽所有的位置， 不过我相信没有人是这么完成对物体的识别的。

那怎么办？CNN中的卷积正是这一问题的答案，因为卷积操作本身具有平移不变性（我知道听起来不明觉厉 ，请看下文）。

卷积，顾名思义， “卷”有席卷的意思， “积 “有乘积的意思。 卷积实质上是用一个叫kernel的矩阵，从图像的小块上一一贴过去，一次和图像块的每一个像素乘积得到一个output值， 扫过之后就得到了一个新的图像。我们用一个3*3的卷积卷过一个4*4的图像， 看看取得的效果。

卷积的数学过程

一个卷积核就像一个小小的探测器， 它的DNA是被刻录在卷积核的数字里的， 告诉我们它要干什么， 而卷积核扫过图片，只要它的DNA是不变的，那么它在图片上下左右的哪个位置看到的结果都相同， 这变是卷积本身具有平移不变性的原理。 由于这种不变性， 一个能够识别1的卷积在图片的哪个位置都可以识别1，一次训练成本，即可以对任何图片进行操作。

图像处理领域，卷积早已有另一个名字 ， 叫做滤镜，滤波器， 我们把图像放进去，它就出来一个新图像，可以是图像的边缘，可以是锐化过的图像，也可以是模糊过的图像。

如果大家玩过photoshop， 大家都会发现里面有一些滤镜，比如说锐化，模糊， 高反差识别这一类，都是用着类似的技术，这样的技术所作的事情是图像的每个小片用一个矩阵进行处理，得到一个画面的转换 。 我们有时候会说低通和高通滤镜 ，低通滤镜通常可以用来降噪， 而高通则可以得到图像的细微纹理。 你玩photoshop，玩的就是卷积，卷积核里面的数字定了， 它的功能也就定了。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

为什么这样做有效果了？因为图像的特征往往存在于相邻像素之间， kernel就是通过计算小区域内像素的关系来提取局部特征，可以理解为一个局部信息的传感器， 或物理里的算子。

比如提到的边缘提取滤镜， 它所做的物理操作又称为拉普拉斯， 只有像素在由明亮到变暗的过程里它才得1， 其他均得0，因此它所提取的图像特征就是边缘。 事实上我们知道图像中的信息往往包含在其边缘，你给以一个人画素描， 一定能够完全识别这个人 。 我们通过寻找到信息的关键载体-边缘， 而把其他多余的信息过滤掉，得到了比第一层更好处理的图像， 大大减少了需要搜索图像的可能性。

卷积的边缘抽取过程

常用于卷积的Kernel本质是两个： 第一， kernel具有局域性， 即只对图像中的局部区域敏感， 第二， 权重共享。 也就是说我们是用一个kernel来扫描整个图像， 其中过程kernel的值是不变的。 这点就可以保证刚刚说的平移不变形。 比如说你识别一个物体， 显然你的识别不应该依赖物体的位置。 和位置无关， 及平移不变。

那卷积如何帮你从不同的图形中识别数字1了？数字的尖锐的线条会让卷积的值很高（响起警报）。 无论你1出现在图像中的哪一个位置， 我的局部扫描+统一权重算法都给你搞出来， 你用同一个识别1的卷积核来扫过图片，voila，任何一个位置我都给你找出来。

那卷积和神经网络有什么关系了？答案是卷积扫过图像，每一个卷积核与图像块相乘的过程，都可以看作是一个独立的神经元用它的神经突触去探测图像的一个小局部，然后再做一个决策，就是我看到了什么或没看到什么。整个卷积过程， 不就对应一层神经网络吗？啊哈， 整个卷积过程相当于一层神经网络！

CNN(卷积神经网络)是什么？有入门简介或文章吗?

一个个小探测器一般的神经元

刚刚说了卷积是一个能够对图片中任何位置的同一类信息进行抽取的工具， 那么我们还讲到我们除了抽取， 还要做的一个工作是，取出重要信息，扔掉不重要的，实现这一个的操作，叫做pooling

但是大家注意，这个时候如果原图像是28*28， 那么从kernel里出来的图形依然是28*28， 而事实上， 事实是上， 大部分时候一个图像的局部特征的变化都不会是像素级。我们可以把局部特征不变形看做一个假设， 把这个假设作为一个数学公式加入到卷积层里帮我们过滤冗余信息， 这就是pooling所做的事情 -也就是扔掉你周边得和你长得差不多得那些像素。

Max Pooling的数学过程

Pooling的本质即降采样，以提升统计效率，用一个比较冠冕的话说是利用局部特征不变性降维，pooling的方法很多，常见的叫做max pooling，就是找到相邻几个像素里值最大的那个作为代表其它扔掉。

这样经过从卷积到pooling的过程， 在识别1的任务里，我们可以验明在每个小区域里有没有存在边缘， 从而找到可能存在1的区域。 在pooling的终结点， 我们得到的是一个降低维度了的图像，这个图像的含义是告诉你在原有的图像的每个区域里是含有1还是不含有1， 又叫做特征图。

好了，我们可以从一堆图片中识别出1了， 那么我们怎么搞定2呢？ 我们把2写成一个Z型， 你有没有思路我们如何做到这点？ 我们不能只识别竖着的线条，还需要识别横向的线条，记住，一个卷积层只搞定一个特征，如果你既要找竖线也要找横线， 我们需要两个不同的卷积层，并且把他们并联在一起，

CNN(卷积神经网络)是什么？有入门简介或文章吗?

手写数字识别

然后呢？ 横线对应一张特征图， 竖线对应另一个张特征图， 如果要识别2， 你无非需要比较这两张特征图，看是否有哪个位置两个特征图同时发生了警报（既有横线又有竖线）。

这个比较的过程，我们还是可以用一个卷积搞定（理由依然是平移不变性）！

这个时候， 新的卷积层对之前并连的两个卷积的结果做了一个综合， 或者说形成了一个特征之特征， 即横向和竖线交叉的特征。

这里把我们的理论可以更上一层路。 深度意味着什么？ 我们想一下， 要正确的识别一个图像，你不可能只看变，也不可能只看边角， 你要对图像的整体有认识才知道张三李四。 也就是说我们要从局部关联进化到全局关联， 真实的图像一定是有一个全局的，比如手我的脸， 只有我的眼镜，鼻子耳朵都被一起观察时候才称得上我的脸，一个只要局部，就什么都不是了。如何提取全局特征?

从一个层次到另一个层次的递进， 通常是对上一层次做横向以及纵向的整合（图层间的组合或图层之内的组合或两者），我们的特征组合是基于之前已经通过pooling降低维度的图层，因此事实上每一个神经元决策的信息相对上一层都更多，我们用一个学术名词 – 感受野来表述一个神经元决策所涵盖的像素多少， 上一层次看到更多的输入神经元， 因此感受野看更多了 。 越靠近顶层的神经元， 所要做的事情就越接近全局关联。

越深，感受野越大， 表示越抽象

这和物理学的一个基本方法--尺度变换有着异曲同工之妙（我们后面讲）， 也是提取全局信息的一个非常核心的办法，我管它叫级级递进法。 你一级一级的进行对画面进行降采样， 把图像里的四个小格子合成一个， 再把新的图像里四个小格子合成一个， 直到一个很大的图像被缩小成一个小样。每一层的卷积，都不是一个卷积，而是一组执行不同特征提取的卷积网络，比如我刚刚说的 不同方向的边缘沟成的一组卷积， 你可以想象后面有不同大小的角度组成的一组网络， 他体现了在一个空间尺度上我们所能够达到的特征工程。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

如此级级互联， 越靠上层感受野就越大。 整个CNN网络如同一封建等级社会，最上层的，就是君王，它是整个集团唯一具有全局视野的人，下一级别， 是各大领主，然后是领主下的风尘，骑士，知道农民（底层神经元）。

我们把刚刚的全局换一个词叫抽象。深度卷积赋予了神经网络以抽象能力。 这样的一级级向上卷积做基变换的过程，有人说叫搞基（深度学习就是搞基），深一点想叫表征， 和人的思维做个比喻就是抽象。 抽象是我在很深的层次把不同的东西联系起来，CNN教会了我们事先抽象的一种物理方法。

到目前为止， 我所描述的是都是一些人工的特征工程，即使网络在深，顶多说的上是深度网络，而与学习无关。我们说这样一个系统（mxnxpxz）， 我们要人工设计，几乎穷经皓首也可能做的都是错的。我们说， 这样的一个结构， 只能靠机器自己学，这就是深度学习的本质了， 我们通过几条basic假设（正则）和一个优化函数，让优化（进化）来寻找这样一个结构。 Basic假设无非图像的几个基本结构， 体现在几个不变形上，物理真是好伟大啊。

深度学习的训练，就是计算机帮助人完成了机器学习最难的一步特征工程（特征工程本质就是基变换啊）。以前人类穷尽脑汁思考如何做图像识别， 是寻找人是如何识别图像的， 希望把人能用来识别物体的特征输入给计算机， 但是现在通过深度卷积，计算机自己完成了这个过程。

卷积网络在2012 年的发展趋势， 大家可以关注几个方向：

1， 更深的模型： 从AlexNet到VCG19， High way network 再到残差网络， 一个主要的发展趋势是更深的模型。 当你采用更深的模型，经常你会发现一些神奇的事情发生了。 当然网络的宽度（通道数量）也在增加。

这只是最初级的CNN

CNN(卷积神经网络)是什么？有入门简介或文章吗?

这也只是小菜一碟

2， 更通畅的信息交换：深，带来的第一个问题是训练困难， 反向传播难以传递。 从残差网络，到目前开始流行的Dense Network， 一个主要的发展趋势是不同层级间的信息的交换越来越通畅。 我们逐步在不同层之间加入信息的直连通道。

Dense Network

3， 与监督学习之外的学习方法的结合， 如迁移学习， 半监督学习， 对抗学习， 和强化学习。 后两者的有趣程度远超监督学习。

4， 轻量化， CNN网络越来越深， 使得网络的文件动辄装不下， 这点使得CNN网络的轻量化部署成为重点， 我们希望在性能和能耗中取中。 一个很好的办法是对网络权重进行减枝，去掉不重要的权重， 另外一个是把每个权重的数据位数本身缩减，甚至是使用0和1表示， 虽然看上去我们丢失了很多信息， 但是由于巨大网络中的信息是统计表达的，我们到底损失多大还真不一定。

酷似生物过程的剪枝处理

以上是CNN的小结， 不要以为图像处理与你无关，我刚刚说的其实一篇文章如果你把它转化为一个矩阵无非一个图像， 一段音频你给它转换成一个矩阵无非一个图像， 你看， 都可以和CNN挂钩。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

我想说，无论你是做什么的， 无论是苦逼的计算机工程师， 游戏设计师， 还是外表高大上的金融分析师，甚至作为一个普通消费者， 你的生活以后都和CNN脱不开干系了， 预知更多情报还请关注：

巡洋舰的深度学习实战课程， 手把手带你进行深度学习实战， 课程涵盖机器学习，深度学习， 深度视觉， 深度自然语言处理， 以及极具特色的深度强化学习，看你能不能学完在你的领域跨学科的应用深度学习惊艳你的小伙伴，成为身边人眼中的大牛。刚刚讲的方法都将在课程里详细展开。

目前课程线下版本已经基本报名完毕（特殊申请可加一到两个名额）， 为了缓解众多异地学员的需求， 我们提出一个线上加线下的课程简版， 课程包括全部课程视频， notebook作业， 和一个课程模块的来京线下实践机会， 名额限5名，预报从速，详情请联系陈欣（cx13951038115）。

发布于 2017-12-16

▲ 87 ▼ 1 条评论 分享 收藏 感谢 ··· 收起

Eric Leo

17 人赞同了该回答

推荐一个英文的文档，名字叫：
A guide to convolution arithmetic for deep learning

此外，Li FeiFei的一个关于开放课程里面，也有相关的介绍
CS231n Convolutional Neural Networks for Visual Recognition

编辑于 2016-12-01

▲ 17 ▼ 1 条评论 分享 收藏 感谢 ···

Nick-Atom
炼金术士一枚， 有问题咨询请先报上八字。

5 人赞同了该回答

po一组我的博客， 从全连接网络开始， 到CNN, 详细介绍了每种模型的原理， 所有模型均使用Python/Numpy实现， 不需要任何深度学习框架。

欢迎关注转发。

CNN卷积网络的Python实现(一):FCN全连接网络 - Nick's Tech Blog

CNN卷积网络的Python实现(二):Regularization正则化实现 - Nick's Tech Blog

CNN卷积网络的Python实现(三):卷积网络实现 - Nick's Tech Blog

CNN卷积网络的Python实现(四):池化和BN层的实现 - Nick's Tech Blog

CNN卷积网络的Python实现(五):卷积网络实现 - Nick's Tech Blog

附赠一篇RNN/LSTM哦～～

RNN, LSTM与ImageCaptioning原理及Python实现

编辑于 00:53

▲ 5 ▼ 2 条评论 分享 收藏 感谢 ···

机智的大群主
雷锋公开课，用技术洞见未来，公众号【AI研习社】

16 人赞同了该回答

卷积神经网络（Convolutional Neural Network,CNN）新手指南

▲ 1.3K ▼ 57 条评论 分享 收藏 感谢 ··· 收起

CNN(卷积神经网络)是什么？有入门简介或文章吗?

引言

卷积神经网络：听起来像是生物与数学还有少量计算机科学的奇怪结合，但是这些网络在计算机视觉领域已经造就了一些最有影响力的创新。2012年神经网络开始崭露头角，那一年Alex Krizhevskyj在ImageNet竞赛上（ImageNet可以算是竞赛计算机视觉领域一年一度的“奥运会”竞赛）将分类错误记录从26%降低到15%，这在当时是一个相当惊人的进步。从那时起许多公司开始将深度学习应用在他们的核心服务上，如Facebook将神经网络应用到他们的自动标注算法中，Google（谷歌）将其应用到图片搜索里，Amazon（亚马逊）将其应用到产品推荐服务，Pinterest将其应用到主页个性化信息流中，Instagram也将深度学习应用到它们的图像搜索架构中。

然而最经典的，或者说最流行的神经网络使用范例是将其用于图像处理领域。提到图像处理，本文主要介绍的是如何使用卷积神经网络来进行图像分类。

更多人工智能：知乎专栏--硬创公开课

问题空间

图像分类是将输入图像（猫、狗等）进行分类输出或者将其分类成最能描述图像特征的类别的任务。对于人类来说，认知是我们出生之后就学会的第一个技能，也是作为成年人来说非常自然和轻松的技能。我们可以毫不犹豫迅速识别出我们周围的环境以及物体，当我们看到一张图片或者观察周遭环境时，大部分时间我们都能马上对场景做出判断并且给每个物体都打上标识，这些甚至都不需要刻意去观察。这些技能能够迅速识别其模式，从我们以前的经验中做出推论，然后将其运用至不同的图片或者环境中——这些都是我们与机器不同的地方。

输入与输出

当计算机看到一张图片时（即输入一张图片），它所看到的是一系列的像素值。根据图片的分辨率与大小，计算机将看到的是一个32×32×3的数字阵列（3指代的是RGB—颜色值）。我们稍微将一下这个，假设我们有一张480 x 480的JPG格式图片，它的表达阵列即为480 x 480 x 3。这些数字中的每一个值都可以从0取到255，它描述了在这一点上的像素强度。这些数字虽然对于我们进行图像分类时没有任何意义，但其却是计算机在图像输入时唯一获取的数据。这个理念就是你给电脑指定相关数据排列，它将图像是一个特定的类别的可能性进行输出（如80—猫，15—狗，05—鸟等）。

我们希望电脑做什么

CNN(卷积神经网络)是什么？有入门简介或文章吗？

现在我们了解到问题是在输入和输出上，让我们来考虑如何解决这个问题。我们希望电脑能做到的是在所有的给定图像中分辨出不同的类别，它能找到那些“狗之所以是狗”或者“猫之所以是猫”的特性。这个就是在我们的头脑中潜意识里进行认知识别的过程，当我们看到一张狗的图像时，我们能够将其分类因为图像上有爪子或者四条腿等明显的特征。以类似的方式计算机能够进行图像分类任务，通过寻找低层次的特征如边缘和曲线，然后运用一系列的卷积层建立一个更抽象的概念。这是卷积神经网络应用的一个总体概述，接下来我们来探讨下细节。

生物联系

首先要稍微普及下一点背景知识，当你第一次听到卷积神经网络这个词时，你也许会想这是是不与神经科学或者生物学有关？恭喜你，猜对了一部分。卷积神经网络的确从生物学上的视觉皮层得到启发，视觉皮层有微小区域的细胞对于特定区域的视野是十分敏感的。

1962年，Hubel和 Wiesel发现大脑中的部分神经元只对一定的方向的边缘做出回应。例如，当暴露在垂直边缘或者一些当水平或对角线边缘时，一些神经元才会做出回应。Hubel和 Wiesel发现，所有这些神经元都被架构在一个柱状结构中，这样的架构使它们能够产生视觉感知。系统中的特定成员可以完成特定任务这种理念（神经细胞在视觉皮层中寻找特定的特征）也能很好地应用在机器学习上，这也是卷积神经网络的基础。

架构

对于卷曲神经网络更详细的介绍是将图片通过一系列的卷积、非线性、池（采样）、全连接层，然后得到一个输出。正如我们前面所说的，输出是一个类或者一个图像类别的可能性概率。现在，困难的部分是了解每一层的任务。

第一层—数学

卷积神经网络的第一层是卷积层，第一件事是你要记住卷曲层的输入时什么。像我们之前提到的，输入的是一个32×32×3的系列像素值。解释卷积层的最好方式是想象一个手电筒正在图像的左上方进行照射，假设手电照射的区域是5 x 5的范围。再想象下这个手电筒在输入图像的各个区域进行滑动。在机器学习术语中，这个手电筒叫做过滤器（有时候也称为神经元或者核心），它照射着的区域被称为接受场。这个过滤器也是一系列的数据（这些数据被称为权重或者参数）。必须提到的是这个过滤器的深度必须是和输入的深度相同（这样才能保证数学正常工作），所以这个过滤器的尺寸是5×5×3。现在，让我们先拿第一个位置的过滤器为例。由于过滤器在输入图像上是滑动或卷积的，它是相乘的值在滤波器的原始图像的像素值（又名计算元素的乘法），这些乘法全部相加（从数学上讲，这将是75次乘法总和）。所以现在你有一个数字。请记住，这个数字只是当过滤器在图像的左上角时才有代表性，现在我们在每一个位置上重复这个过程。（下一步将过滤器移动到右边的1个单位，然后再向右移动1个单位，等等），每一个输入层上独特的位置都会产生一个数字。将过滤器滑动完所有位置的，你会发现剩下的是一个28×28×1的系列数字，我们称之为激活图或者特征图。你得到一个28×28阵列的原因是有784个不同的位置，一个5×5的过滤器可以适配一个32×32的输入图像，这组784个数字可以被映射到一个28×28阵列。

目前我们使用两个5 x 5 x 3的过滤器，我们的输出量将是28×28×2。通过使用更多的过滤器，我们能够更好地维持空间尺寸。在数学层面上来说，这些是在一个卷积层中进行的任务。

第一层—高阶视角

让我们从高阶角度来谈谈这个卷积层的任务，这些过滤器中每个都可以被认为是特征标识符。当我说特征时，我说的是如直边、简单的颜色和曲线等。思考一下，所有的图像都有同样的最简单的特征。我们的第一个过滤器是7×7×3，而且是一个曲线探测器。（在这一部分让我们忽略一个事实，过滤器是3个单位深的，只考虑顶部过滤器的深度和图像。）作为一个曲线检测器，过滤器将有一个更高的数值且有曲线的形状的像素结构（记住关于这些过滤器，我们考虑的只是数字）。

CNN(卷积神经网络)是什么？有入门简介或文章吗?

现在，让我们回到数学可视化部分。当我们在输入的左上角有了这种滤波器后，它会在哪个区域的过滤器和像素值之间计算乘积。现在让我们以一个我们要分类的图像为例，把我们的过滤器放在左上角。

记住，我们需要做的是使用图像中的原始像素值在过滤器中进行乘积。

基本上在输入图像中，如果有一个形状是类似于这种滤波器的代表曲线，那么所有的乘积累加在一起会导致较大的值！现在让我们看看当我们移动我们的过滤器时会发生什么。

检测值竟然要低得多！这是因为在图像中没有任何部分响应曲线检测过滤器。记住，这个卷积层的输出是一个激活图。因此，在简单的情况下一个过滤器的卷积（如果该过滤器是一个曲线检测器），激活图将显示其中大部分可能是在图片中的曲线区域。在这个例子中，我们的28×28×1激活图左上方的值将是6600，这种高值意味着很可能是在输入中有某种曲线导致了过滤器的激活。因为没有任何东西在输入使过滤器激活（或更简单地说，在该地区的原始图像没有一个曲线），其在我们的激活图右上方的值将是0。记住，这仅仅只是一个过滤器。这个过滤器将检测线向外和右边的曲线，我们可以有其他的曲线向左或直接到边缘的过滤器线条。过滤器越多，激活图越深，我们从于输入中获取的信息也就越多。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

声明：在这一节中描述的过滤器是简化的，其主要目的是描述在一个卷积过程中的数学过程。在下图中你会看到一些对训练过的网络中第一个卷积层的过滤器的实际显示示例，尽管如此，主要的论据仍然是相同的。

进一步深入网络

现在展示一个传统的卷积神经网络结构，还有其他层在这些层之间穿插转换。强烈建议那些有兴趣的读者去了解他们的功能和作用，但一般来说他们提供的非线性和尺寸留存有助于提高网络的鲁棒性，同时还能控制过度拟合。一个经典的卷积神经网络架构看起来是这样的：

然而，最后一层是非常重要的内容，不过我们将在后面提到。让我们退后一步，回顾一下我们目前提到的东西。我们谈到了第一个卷积层的过滤器被设计用来探测。他们检测到低阶的特征如边缘和曲线。正如想象的那样，为了预测图像的类型，我们需要神经网络能够识别更高阶的特征，如手、爪子、耳朵。让我们考虑经过第一层卷积层后网络的输出是什么，这将是一个28×28×3的体量（假设我们使用三个5×5×3过滤器）。当穿过另一个卷积层时，卷积层的第一输出成为第二卷积层的输入，这有难以视觉化想象。当我们谈论第一层时，输入的只是原始图像。然而，当我们谈论第二个卷积层时，输入是第一层的结果激活图（S）。因此，每一层的输入基本上是描述某些低阶特征在原始图像中的位置。现在当你应用一组过滤器（通过第二卷积层），输出将被激活且代表更高阶的特征。这些特征的类型可能是半圆（曲线和直线边缘的组合）或方形（几个直边的组合）。当通过网络、更多的卷积层，可以激活地图，代表更多和更复杂的特征。在神经网络的结束，可能有一些激活的过滤器，表示其在图像中看到手写字迹或者粉红色的物体时等等。另一个有趣的事情是当你在网络往更深的地方探索时，过滤器开始有越来越大的接受场，这意味着他们能够从一个更大的区域或者更多的原始输入量接收信息。

全连接层

现在我们可以检测到这些高阶特征，锦上添花的是在神经网络的末端连接一个全连接层。这层基本上将一个输入量（无论输出是卷积或ReLU或池层）和输出一个N是程序选择类别的N维向量，具体过程如下图所示。这个全连接层的工作方式是，它着眼于前一层的输出（代表高阶特征的激活图），并确定哪些功能是最相关特定的类。例如如果该程序预测，一些图像是一只狗，它在激活图中会有高的值，代表高阶特征如一个爪子或4条腿等。类似地，如果该程序是预测一些图像是鸟的功能，它在激活图中会有高阶值，代表高阶特征如如翅膀或喙等。

训练过程

训练工程作为神经网络的一个部分，我之前故意没有提到，因为它有可能是最重要的一部分。阅读时你可能会遇到有很多问题，例如第一个卷积层中过滤器如何知道寻找边缘和曲线？全连接层如何知道激活图在哪里？每一层的过滤器如何知道有什么样的值？计算机能够调整其过滤值（或权重）的方式是通过一个称为反向传播的训练过程。

CNN(卷积神经网络)是什么？有入门简介或文章吗？

在我们介绍反向传播之前，我们必须先回顾下谈谈神经网络运行所需要的是什么。在我们出生的那一刻，我们的思想是全新的，我们不知道什么是猫，什么是鸟。类似地，在卷积神经网络开始之前，权重或过滤器的值是随机的，过滤器并不知道去寻找边缘和曲线，在更高阶的层过滤器不知道去寻找爪子和喙。然而当我们稍微大了一点之后，我们的父母和老师给我们展示了不同的图片和图像，并给了我们一个相应的标签。给图像以标签这个想法既是卷积神经网络（CNNs）的训练过程。在讲到它之前，让我们稍微介绍下我们有一个训练集，其中有成千上万的狗，猫和鸟类的图像，每一个图像有一个标签对应它是什么动物的图片。

反向传播可以分为4个不同的部分：前向传播、损失计算、反向传播、权重更新。在前向传播的过程中，你需要一个数字阵列为32×32×3的训练图像，并将其传递通过整个网络。在我们的第一个训练例子中，所有的权重或过滤器的值被随机初始化，输出可能是类似[.1 .1 .1 .1 .1 .1 .1 .1 .1 .1]的东西，基本上是一个不能优先考虑任何数字的输出。目前权重的网络是无法寻找那些低阶的功能，因此也无法对分类可能性作出任何合理的结论。这就到了反向传播中的损失计算部分。我们现在使用的是训练数据，此数据有一个图像和一个标签。比方说，第一个输入的训练图像是一个3，则该图像的标签将是[0 0 1 0 0 0 0 0]。损失计算可以按许多不同的方式定义，但常见的是MSE（均方差）—½倍（实际预测）的平方。

假设变量L等于这个值，正如你想象的，对于第一组训练图像来说损失将是非常高的。现在，让我们更直观地来进行思考。我们想得到一个点的预测标签（ConvNet的输出）作为训练相同的训练标签（这意味着我们的网络预测正确）。为了实现则这个，我们要尽量减少我损失量。可视化在微积分上只是一个优化问题，我们需要找出哪些输入是（在我们的例子中的权重）最直接导致网络的损失（或错误）所在。

这是一个dL / DW的数学等价，其中W是在一个特定层的权重。现在我们要做的通过网络是执行一个反向传播过程，检测哪个权重损失最大并且寻找方法来调整它们使损失降低。一旦我们将这个计算过程进行完，就可以到最后一步—权重更新。把所有的过滤器的权重进行更新，使它们在梯度方向上进行改变。

学习速率是由程序员选择的一个参数。一个高的学习率意味着更多步骤是在权重更新部分，因此它可能需要更少的时间为最佳的权重在模型上进行收敛。然而学习率太高，可能会导致跨越太大而不够精准达到最佳点。

前向传播、损失计算、反向传播、参数更新的过程也称为一个epoch。程序会对于每一个固定数量的epoch、每个训练图像重复这一过程。在最后一个训练例子中完成了参数更新后，网络应该都训练的足够好了，各层的权重也应该调参正确了。

测试

CNN(卷积神经网络)是什么？有入门简介或文章吗?

最后要来测试我们的卷积神经网络是否工作，将不同的图片和标签集通过卷积神经网络，将输出结果与真实值进行对比，即可测试出其是否正常运行。

业界如何使用卷积神经网络

数据，数据，数据。给一个卷积神经网络的更多的训练数据，可以做的更多的训练迭代，也就能实现更多的权重更新，对神经网络进行更好的调参。Facebook（还有Instagram）可以使用数以亿计的用户目前的所有照片，Pinterest可以使用在其网站上的500亿的信息，谷歌可以使用搜索数据，亚马逊可以使用每天数以百万计的产品购买数据。

现在你知道他们是如何使用这些的魔法了，有兴趣的话可以自己尝试一下。

PS : 本文由雷锋网独家编译，未经许可拒绝转载！

via Adit Deshpande

编辑于 2017-01-03

Leon
公众号：工业大数据与PHM研究

5 人赞同了该回答

Convolution Neural Networks卷积神经网络

卷积神经网络是人工神经网络的一种，已成为当前语音分析和图像识别领域的研究热点。它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。该优点在网络的输入是多维图像时表现的更为明显，使图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。卷积网络是为识别二维形状而特殊设计的一个多层感知器，这种网络结构对平移、比例缩放、倾斜或者共他形式的变形具有高度不变性。

2.2.1 网络结构

卷积神经网络是一个多层的神经网络，其基本运算单元包括：卷积运算、池化运算、全连接运算和识别运算。

图4 卷积神经网络结构

l 卷积运算：前一层的特征图与一个可学习的卷积核进行卷积运算，卷积的结果经过激活函数后的输出形成这一层的神经元，从而构成该层特征图，也称特征提取层，每个神经元的输入与前一层的局部感受野相连接，并提取该局部的特征，一旦该局部特征被提取，它与其它特征之间的位置关系就被确定。

l 池化运算：它把输入信号分割成不重叠的区域，对于每个区域通过池化（下采样）运算来降低网络的空间分辨率，比如最大值池化是选择区域内的最大值，均值池化是计算区域内的平均值。通过该运算来消除信号的偏移和扭曲。

l 全连接运算：输入信号经过多次卷积核池化运算后，输出为多组信号，经过全连接运算，将多组信号依次组合为一组信号。

l 识别运算：上述运算过程为特征学习运算，需在上述运算基础上根据业务需求（分类或回归问题）增加一层网络用于分类或回归计算。

2.2.2 训练过程

CNN(卷积神经网络)是什么？有入门简介或文章吗？

卷积网络在本质上是一种输入到输出的映射，它能够学习大量的输入与输出之间的映射关系，而不需要任何输入和输出之间的精确的数学表达式，只要用已知的模式对卷积网络加以训练，网络就具有输入输出对之间的映射能力。卷积网络执行的是有监督训练，所以其样本集是由形如：（输入信号，标签值）的向量对构成的。

2.2.3 典型改进

卷积神经网络因为其在各个领域中取得了好的效果，是近几年来研究和应用最为广泛的深度神经网络。比较有名的卷积神经网络模型主要包括1986年Lenet，2012年的Alexnet，2014年的GoogleNet，2014年的VGG，2015年的Deep Residual Learning。这些卷积神经网络的改进版本或者模型的深度，或者模型的组织结构有一定的差异，但是组成模型的机构构建是相同的，基本都包含了卷积运算、池化运算、全连接运算和识别运算。

2.2.4 模型优缺点

(1)优点：

（1）、权重共享策略减少了需要训练的参数，相同的权重可以让滤波器不受信号位置的影响来检测信号的特性，使得训练出来的模型的泛化能力更强；

（2）、池化运算可以降低网络的空间分辨率，从而消除信号的微小偏移和扭曲，从而对输入数据的平移不变性要求不高。

(2)缺点：

（1）、深度模型容易出现梯度消散问题。

发布于 2017-04-11

▲ 5 ▼ 添加评论 分享 收藏 感谢 ··· 收起

郭婷婷
设计师、插画师、鲁美动画系

4 人赞同了该回答

CNN的入门可以参考今年斯坦福刚开的一门课，slides和一些资料可以从课程主页下载到：
Stanford University CS231n: Convolutional Neural Networks for Visual Recognition

发布于 2016-11-18

▲ 4 ▼ 添加评论 分享 收藏 感谢 ···

David 9
机器学习 编程 思想开放

3 人赞同了该回答

这篇博客不错, 可以参考:
Pycon 2016 tensorflow 研讨会总结 -- tensorflow 手把手入门 #第一讲
还有第二讲:
Pycon 2016 tensorflow 研讨会总结 -- tensorflow 手把手入门, 用"人话"解释CNN #第三讲 CNN

发布于 2017-01-15

▲ 3 ▼ 添加评论 分享 收藏 感谢 ···

梁雨

2 人赞同了该回答

这是高票答案翻译的那篇文章的原作者博客，有些地方翻译的不是太准确，可以中英文对照来看，现在已经出了第三部分了。

CS Undergrad at UCLA ('19)

编辑于 2017-06-11

▲ 2 ▼ 添加评论 分享 收藏 感谢 ···

我爱吃桃
想向天再借五百年

2 人赞同了该回答

最高票答案很好啦，我补充一下。引入卷积操作，相比较于全连接操作的区别。

如果采用全连接:那么会造成神经网络参数极为庞大。导致训练困难。所以现在神经网络结构的设计开始慢慢尝试丢弃全连接层，只要卷积，池化

▲ 1.3K ▼ 57 条评论 分享 收藏 感谢 ··· 收起

CNN(卷积神经网络)是什么？有入门简介或文章吗？

如果采用卷积:CNN是有两个关键点，那就是局部链接以及权重共享，这两点可以大大减少参数的数量。题主不妨自己比较一下。当然，这两个关键点也是有神经生物学原理支撑的。此外，CNN里面有pooling(池化)操作。池化操作目的是为了减少神经元的数量。如果对图像一直卷积卷积，不用池化，然后接全链接层，这样会导致神经元个数极为庞大。

最后推荐关于CNN的资料。

有神经网络基础:看UFLDL(百度搜一下就知道了)，或者CS231n

没有神经网络基础:看Coursera上Andrew ng的机器学习第四，第五单元有关神经网络的内容。然后再看上面提及的资料。

如果只是想了解一下这个东西的话:简单理解为一个机器学习算法(黑盒子)，不过这个算法模拟人类神经元之间的信息传递，学习过程。

编辑于 2016-12-10

▲ 2 ▼ 4 条评论 分享 收藏 感谢 ···

酥沐陌
阿里巴巴最菜前端

2 人赞同了该回答

1. 卷积神经网络本身就是神经网络的一种 只不过权重采用了卷积核的形式。细说的话还有许多但对于cnn是什么这个问题没必要展开。
2. 建议先了解基本的线性回归、logistic回归后，然后阅读UFLDL及学习CS231n。如果英文有压力，可以搜搜中文博客的介绍。

发布于 2016-12-09

▲ 2 ▼ 添加评论 分享 收藏 感谢 ···

蒋竺波
新加坡研究院 AI Engineer 公众号：follow_bobo

2 人赞同了该回答

原创文章，一家之言。

个人公众号：follow_bobo

转载请通知本人。

大家好，我是波波，欢迎再次来到CNN入门讲解。

上次我们讲什么卷积以及卷积在卷积神经网络里的作用，那我们这一期来理一理卷积神经网络大致的结构

▲ 1.3K ▼ 57 条评论 分享 收藏 感谢 ··· 收起

CNN(卷积神经网络)是什么？有入门简介或文章吗？

吗？

我不！！

惊不惊喜？！

抱歉，我就是这样一个水性杨花，做事看心情的美男子。

不服，请点赞。

哈哈哈哈哈哈哈哈哈哈asdo8uh2ojkladojadb183789@&^EU#(kjt

（一阵尴尬的沉默）

上一期我们讲了，卷积实际上可以充当一个对原图像进行二次转化，提取feature 的作用，

相当于信号处理的滤波器，**大家可以再去了解一下高斯滤波，拉普拉斯滤波等，这些都可以写成卷积的形式**

比如这样：

CNN(卷积神经网络)是什么？有入门简介或文章吗?

网上找的，侵删

神经网络想必大家都有了解，评价神经网络好坏的一个重要的依据就是：

以最少的代价，使神经网络获得最好的准确率

关键词：**代价，准确率**

那我们来看看下面这张图：

我们先来看一个手写体的‘2’ （大小是32*32）

正常情况下，我们人眼看这张图的全局，立马就能识别出这是个数字‘2’

机器如何去看全局呢

实际上，就是把所有pixels 一股脑全放进一个神经网络

让神经网络去处理这些pixels

这个神经网络，我们先叫他全连接层吧

就像下图一样：

CNN(卷积神经网络)是什么？有入门简介或文章吗?

感觉就是暴力解决嘛

抛开准确率不说

我们大概会得到3*32*32*1024*512 = 1610612736 个参数，这些参数量是非常大的了

而这仅仅是一个32*32 的图像，**运算量非常大了**

同时，由于**参数过多**，极其容易产生训练结果好，检测结果差的现象（overfitting）

但我们仔细看这个图，我们肉眼其实也是有选择性的，我们并不太关心上图中灰色的区域，以及数字’2‘中的黑色区域

我们更关心’2‘与灰色区域的相交的边缘，因为这才是我们判断的主要依据

那我们会不会也可以在计算机里也这么做，**主要提取边缘特征，对于灰色和黑色这种冗余或者不重要的的区域特征，我们尽量丢弃或者少保留，那么这样可能会减少参数或者减少提参数的过程**

既然这样，**那我们干脆在全连接层前面，对输入图像进行一个预处理吧**

把那些没用的

烂糟的

统统扔掉

于是我们加了个采集模块

我们只保留那些我们想要的，或者相对比较重要的pixels

我们就叫它采样层吧

所以，我们得到下面这个网络：

然后，我们再来看一个问题：

我们来把这个’2‘分成块，一块一块看：

CNN(卷积神经网络)是什么？有入门简介或文章吗?

你一块块看

是不是仍然会发现这是个数字’2‘?

也是说，你大脑会自动把这些分散的图片组合起来进行识别

同时如果我们这么看：

是不是你就识别不出来这是‘2’

也就是说，我们发现了两个现象：

- **如果我们只知道局部的图片，以及局部的相对位置，只要我们能将它正确组合起来，我们也可以对物体进行识别**
- **同时局部与局部之间关联性不大，也就是局部的变化，很少影响到另外一个局部**

我们还要解决两个问题：

- **输入的只是原始图片，我们还没有提取图片的特征**
- **我们目前要处理的参数仍然非常多，我们需要对原始输入进行降维或者减少参数**

我们现在回到上次讲的卷积：

CNN(卷积神经网络)是什么？有入门简介或文章吗？

一个3x3 source pixels 经过一个3x3的卷积核后，source pixels 的特征映射成一个1x1 destination pixel

然后我们再加上以上我们提到一些人眼识别图像的性质

那么

我们就会发现

来着早不如来得巧，卷积加上刚刚好

好了，我们得到了如下神经网络：

（大家请忽略图上的数字，以后会讲的）

实际上，我们还会遇到两个问题：

- **一张图片特征这么多，一个卷积层提取的特征数量有限的，提取不过来啊！**
- **我怎么知道最后采样层选出来的特征是不是重要的呢？**

烦人！

CNN(卷积神经网络)是什么？有入门简介或文章吗？

啊哈哈

我这里又要提一个新概念----**级联分类器（cascade of classifiers）**

这个概念我以后会在机器学习入门上会细说，是个很常用的方法

我来大概介绍一下级联分类器

大概意思就是我从一堆弱分类器里面，挑出一个最符合要求的弱分类器，用着这个弱分类器把不想要的数据剔除，保留想要的数据

然后再从剩下的弱分类器里，再挑出一个最符合要求的弱分类器，对上一级保留的数据，把不想要的数据剔除，保留想要的数据。

最后，通过不断串联几个弱分类器，进过数据层层筛选，最后得到我们想要的数据

（你们也可以去搜搜Adaboost）

那么，针对刚才的问题，我们是否也可以级联一个卷积层和采样层？

是的，效果还不错

于是，如下图所示：

灯，等灯等灯

最简单的一个卷积神经网络，就诞生了

我想要申明的是，这是一个最简答的CNN结构，往后我会慢慢把结构加深

好了，我们来总结一下

CNN(卷积神经网络)是什么？有入门简介或文章吗？

CNN主要由3钟模块构成：

- 卷积层
- 采样层
- 全连接层

大致上可以理解为：

- 通过第一个卷积层提取最初特征，输出特征图（feature map）
- 通过第一个采样层对最初的特征图（feature map ）进行特征选择，去除多余特征,重构新的特征图
- 第二个卷积层是对上一层的采样层的输出特征图（feature map）进行二次特征提取
- 第二个采样层也对上层输出进行二次特征选择
- 全连接层就是根据得到的特征进行分类

初学者是不是有点懵逼

其实整个框架很好理解，我举个生动形象的例子：

输入图像好比甘蔗。

卷积层好比A君要吃甘蔗，吃完之后（卷积），他得出一系列结论，这个甘蔗真好吃，我还想再吃！

啊不是，说错了

他得出结论，这个甘蔗是圆柱形，长条，甜的，白的，多汁的等等（提取特征）

采样层就好比第一个吃甘蔗的人告诉B君，吃甘蔗，重要的是吃个开心，为什么开心，因为它又甜又多汁，还嘎嘣脆（特征选取）

第二个卷积层就好比，B君并没有去吃剩下的甘蔗，而是

头也不回。

拦也拦不住的

去吃A君吐出的甘蔗渣

然后得出结论，嗯~~，

咦~~？

哦~~！

‘原来这甘蔗渣是涩的，是苦的，不易嚼，不好咽’ B君这么说道（二次提取特征）

第二个采样层就好比，B君对别人说，这个甘蔗渣啊，吃的我不开心，因为它很涩，不好咽（二次特征选取）

如果你们要吃，注意点！

注意点!

意点！

点！

CNN(卷积神经网络)是什么？有入门简介或文章吗?

全连接层的作用，就好比是一个决策者，他听取了A,B君的描述

这样，如果有人吃很多东西，其中就有甘蔗

他们吃的时候，有一样东西吃起来，感觉和A,B君描述的非常接近，那么决策者就认为

这个很大概率是甘蔗了

好了，我讲了这么多，是不是大概有点理解CNN了？

什么

没有

你那按我说方法去吃甘蔗试试？

编辑于 2017-11-25

liyanjiebeijing
算法工程师

1 人赞同了该回答

看到很多回答都是从头开始阐述CNN的，这里推荐一篇文章，它假设读者已经了解传统的神经网络结构，并且讲解了CNN与传神经网络的区别：CS231n Convolutional Neural Networks for Visual Recognition。

如果读者想了解传统的神经网络，请参见下面的电子书：

Neural networks and deep learning

发布于 2017-09-19

gegey
在深务工人员！

1 人赞同了该回答

可以参考下面的文章，应该是简单易懂吧~

Deep Learning： Convolutional Neural Networks

发布于 2017-09-16

CNN(卷积神经网络)是什么？有入门简介或文章吗？

Louis
机器学习算法工程师

1 人赞同了该回答

A Beginner's Guide To Understanding Convolutional Neural Networks

发布于 2017-02-25

▲ 1 ▼ 添加评论 分享 收藏 感谢 ···

赵鑫
工科博士生

coursera或者网易云课堂上吴恩达老师的深度学习工程师 专项课，里面有一个课程就是讲CNN的，入门很不错的课程，coursera上还有课后编程实践课，配套使用效果更佳。

发布于 昨天 10:48

▲ 0 ▼ 添加评论 分享 收藏 感谢 ···

金燕
多智时代 duozhishidai.com

卷积神经网络是一种多阶段全局可训练的人工神经网络模型，可以从经过少量预处理，甚至原始数据中学习到抽象的、本质的和高阶的特征。

卷积神经网络在车牌检测、人脸检测、手写体识别和目标识别等领域已经得到了广泛的应用，卷积神经网络在二维模式问题上，通常表现得比多层感知器好，原因在于卷积神经网络在结构中加入了二维码模式的拓扑结构，并使用3种重要的结构特征：局部接受域、权值共享和子采样来保证输入信号的目标平移、放缩和扭曲一定程度上的不变性。

卷积神经网络主要由特征提取和分类器组成，特征提取包含多个卷积层和子采样层，分类器一般使用一层和两层的全连接神经网络。卷积层具有局部接受域结构特征，子采样层具有子采样结构特征，这两层都具有权值共享结构特征。

神经网络大大优化了机器学习的速度，使人工智能技术获得了突破性进展，在此基础上，图像识别、语音识别、机器翻译都取得了长足进步，所以说，我们更要知道，什么是神经网络，深度神经网络怎么分类的，主要是做什么的？ - 人工智能 多智时代

发布于 2017-12-10

▲ 0 ▼ 添加评论 分享 收藏 感谢 ···

杨安锋
互联网/机器学习/程序猿/数据挖掘爱好者

高票答案写的真是太好了

发布于 2017-05-11

▲ 0 ▼ 添加评论 分享 收藏 感谢 ···

崔永明
算法工程师一枚

学习了很多。。。。

发布于 2017-04-06

▲ 0 ▼ 添加评论 分享 收藏 感谢 ···

随风之鱼
科学家

有没有高手讲讲如何针对需要设计一个算法？或者如何设计一个学习方法。

发布于 2017-01-28

▲ 0 ▼ 添加评论 分享 收藏 感谢 ···

写回答

1 个回答被折叠（为什么？）

▲ 1.3K ▼ 57 条评论 分享 收藏 感谢 ··· 收起

CNN(卷积神经网络)是什么？有入门简介或文章吗?