# Deep Convolutional Neural Networks

IEEE Signal Processing Magazine | November 2018 |

神经网络是人工智能（AI）的下属领域。多维信号处理的神经网络的主要类型是深度卷积神经网络（CNNs）。 术语“深度”通常是指具有从“很少”到几十个或更多个卷积层的网络，而深度学习是指用于训练这些系统使用代表特定问题领域的数据自动学习其功能参数的方法。 CNN目前正在广泛的应用领域中使用，所有这些领域都具有一个共同的目标，即能够从（通常是庞大的）数据库中自动学习功能，并预测其对学习阶段未遇到的情况的响应。 最终，学习到的特征可用于完成某些任务，例如对CNN预期处理的信号类型进行分类。 本文的目的有二：1）介绍CNN的基本体系结构，以及2）通过计算示例说明如何训练和在实践中使用CNN解决特定类别的问题。

相关知识

经过数十年的停滞，人工智能最近经历了爆炸式的发展。其作为行业，政府和企业的战略工具，在全球范围内广受关注。这种兴趣基于这样一个事实，即人工智能使计算机有可能从经验中学习，归纳其行为并执行通常与人类智慧相关的任务。AI的一些应用已为公众所熟知，例如在国际象棋上击败大师，识别指纹并解释口头命令的计算机。其他应用程序还不太为人所知，例如欺诈检测，搜索大量数据中的模式以及控制复杂的工业过程。但是，尽管这些应用程序各有不同，但它们都是基于深度学习中的相同概念。二维（2-D）信号处理中特别令人感兴趣的是使用深度学习对数字图像的内容进行自动识别，目前，该技术在从脸部和视网膜识别以及质量检查，医学诊断和自动驾驶汽车导航等生物识别技术到视觉识别等领域都取得了空前的成功。

## 先决条件

理解本文的唯一先决条件是演算（特别是微分和链式规则），和线性代数，但都在本科知识范畴中。

## 背景以及问题描述

使用计算机执行自动图像识别任务的兴趣可以追溯到半个多世纪之前。 在1950年代中期和1960年代初期，一类所谓的学习机器[1]在机器学习领域引起了极大的兴奋。 原因是开发了数学证明，表明当用线性可分离的数据集训练时，称为感知器的基本计算单元将在有限的迭代步骤中收敛到一个解决方案。 该解决方案采用超平面系数的形式，该系数能够正确分离特征超空间中的这些数据类。 不幸的是，基本感知器不足以完成具有实际意义的任务。 随后通过组装这些设备的多层来扩展感知器功能的尝试缺乏有效的训练算法，例如那些对感知器本身产生了兴趣的算法[2]。这种令人沮丧的技术随着1986年反向传播技术的发展而改变，反向传播技术是一种训练神经网络的方法，该网络由类似感知器的单元组成[3]。 反向传播在1989年我们现在称为深度CNN的背景下首次应用于二维信号[4]。 在接下来的二十年中，类似的努力在相对较低的水平上进行，但是直到2012年，2012年ImageNet挑战赛的结果发布证明了深层CNN的力量之后，这些神经网络才开始广泛用于图像模式 识别和其他成像应用[5]，[6]。 如今，CNN是解决复杂图像识别任务和其他重要领域的首选方法，这将在不久后提及。 机器的模式识别涉及以下四个基本阶段：1）采集2）预处理3）特征提取4）分类。 采集生成原始输入模式（例如，数字图像）； 预处理处理诸如降噪和几何校正之类的任务； 特征提取处理计算属性，这是区分一类模式与另一类模式的基础。 分类是将给定输入模式分配给几个预定义类之一的过程。 特征提取通常是最难解决的问题，通常需要进行大量工程设计才能为给定应用定义和测试一组合适的特征。 CNN提供了一种替代方法，该方法可以通过利用样本的大型数据库（称为训练集）来自动学习特征，这些样本代表了感兴趣的应用领域。 本教程解决的问题是定义一个基于CNN的策略，用于从大型训练数据库中自动提取特征，并使用这些特征从训练数据库以及一组独立的测试图像中准确识别图像。 到目前为止，这类问题是CNN的主要应用，但不是唯一的用途。 CNN目前已成功应用于许多其他领域，包括语音识别，语义图像分割和自然语言处理[8]。 在每种情况下，如何构造CNN的细节可能会有所不同，但是其操作原理与本文中讨论的原理相同。

## 解决方案

我们通过使用由卷积，激活和池化层组成的深度模块化CNN体系结构，来解决上一部分中所述问题的解决方案。 然后，将CNN的输出放入一个深度全连接的神经网络（FCN），该网络的目的是将一组二维特征映射到每个输入图像的类标签中。 这种方法的核心是使用样本训练数据来学习每个网络层的操作参数的能力。 为此，我们使用反向传播且循环使用数据集来迭代地调整网络权重（也称为系数，参数和超参数）。 最后，我们通过使用大型手写数字字符数据库训练CNN / FCN系统，然后使用训练阶段未使用的一组图像对其进行测试，证明了该解决方案的有效性。如我们在“计算示例”部分中所示，系统在两个数据集的图像上实现的识别精度均超过99％。

**深度卷积神经网络（Deep CNNs）**

图1显示了CNN一级的基本结构。 实际上，CNN可以具有数十个这样的结构，它们相互串联。 除了阶段的数量外，CNN架构中元素在每个阶段的定义和使用也有所不同，但是图1中的基本结构对于所有阶段都是必不可少的。

如图所示，CNN的一个阶段通常由三个部分（volumes）组成，分别由input maps, feature maps, and pooled feature maps (or pooled maps)组成。 pooled maps并不总是在每个阶段都使用，在某些应用程序中根本没有使用。 所有map都是二维数组，其大小通常随volumes的变化而变化，但是一个volume内的所有map的大小相同。如果CNN的输入是RGB彩色图像，则输入体积将包含三个映射-RGB图像的红色，绿色和蓝色分量图像或通道。 input maps volume一词来自这样一个事实：输入的高度和宽度和深度，高度和宽度是每张map的空间维度, 深度等于体积中地图的数量。 在我们的讨论中，第一阶段的深度（input volume）通常由多光谱图像的通道组成(深度为3)。所有其他阶段的input volume是上一个阶段的已经池化（pooled）的map（或没有pooled的feature maps）。如果存在，则阶段中pooled map的数量等于feature map的数量。

在CNN的每个阶段执行的基本操作是卷积，这些神经网络从中得出其名称。 尽管卷积在信号处理中是普遍存在的操作，但并非总是明确指出在CNN中执行的卷积类型通常是volume卷积[[1]](#footnote-1)，而是卷积核（也称为滤波器）数量与深度这个维度之间的关系是无可替代的。

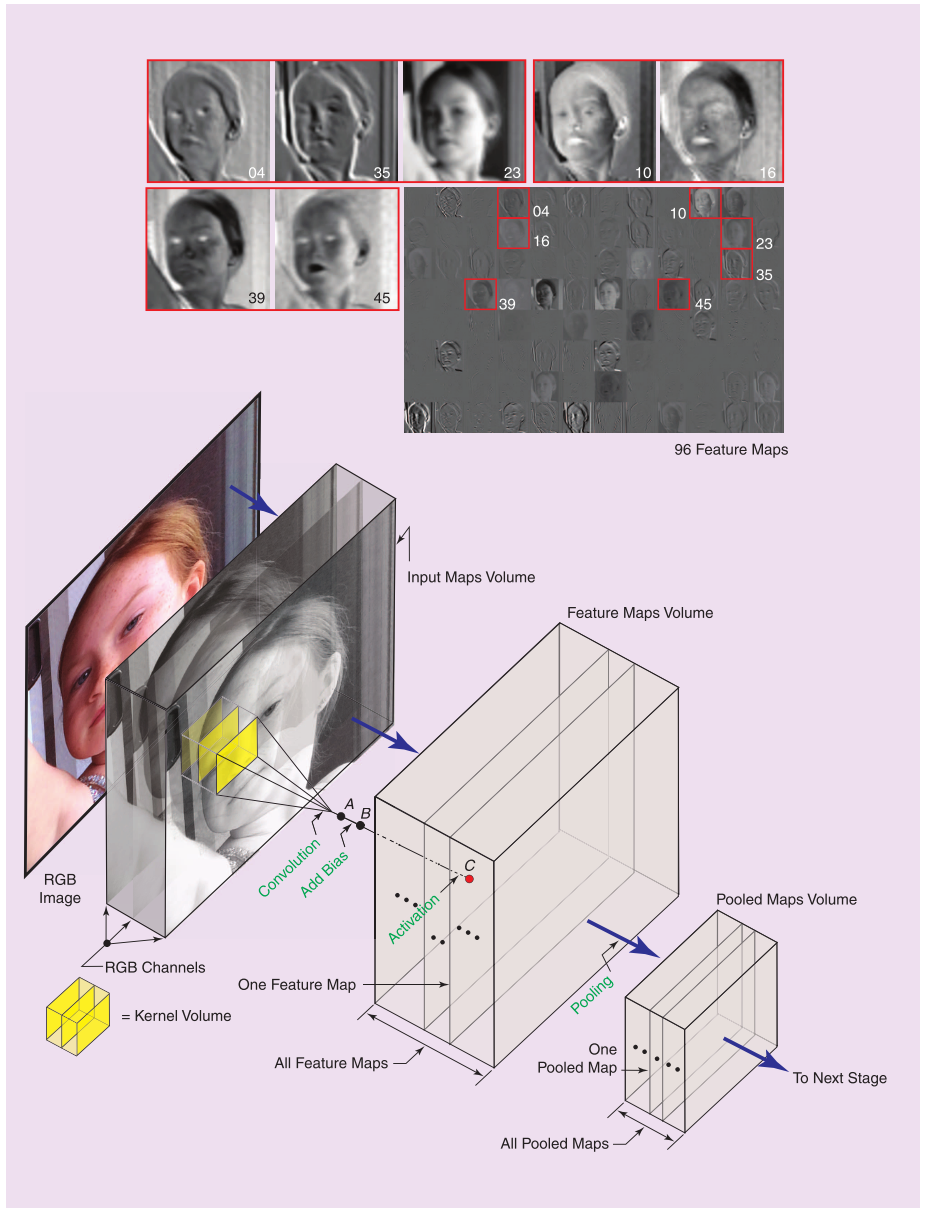
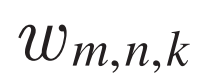
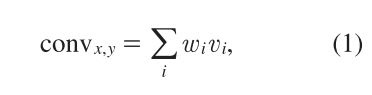


图 1

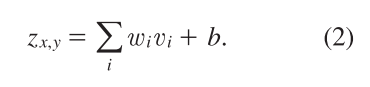
CNN的一个阶段的组成部分，由输入映射卷、特征映射卷和可选的混合映射卷组成。输入卷中的贴图对应于所示RGB图像的三个通道。该阶段有96个特征图和96个集合图。突出显示的特征图，显示为图像和数字标识，说明了CNN能够从输入图像中提取的特征类型。

图1说明了这一概念，其中以黄色显示的内核体积由三个单独的2-D内核组成。从该图可以明显看出，每个阶段中每个内核的数量始终等于该阶段中input volume的深度。在不同的2D内核及其对应的2D输入映射之间执行卷积。因为在深度方向上没有变化，所以这种情况下的卷积数量就是各个二维卷积的总和。 要了解CNN的工作原理，它有助于将注意力集中在一对空间坐标（x，y）的体积卷积结果上。

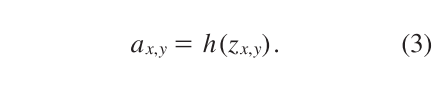
令表示与第k个图相关联的二维核的权重 在输入卷中，m、n表示核的高度和宽度。在任何特定的空间位置，地图的（x，y），此内核与第k个map之间的卷积是内核的权重与map上与该内核在空间上重合的元素的乘积之和。为了获得体积卷积，在每个相应的2-D内核与其在相同空间位置的map之间执行乘积求和运算。 每个乘积之和是一个标量，并且该点的体积卷积是k个所得标量的总和，其中K是input volume的深度。 以等式形式编写此代码将需要K个2-D加和。 但是，由于将在下一部分中说明的原因，我们可以重新定义索引并将K总和写为如下公式：



其中w是核权重，v是input map中空间对应元素的值，是输入体积的所有map在相同空间坐标（x，y），处的体积卷积的结果。公式（1）给出了图1中点A的结果。点B的结果是通过将标量偏差，b添加到（1）而获得的



我们将在下一部分中讨论这种偏见的性质。C点的结果是通过将标量zx,y通过称为激活函数h的非线性传递而获得的



在实践中使用的激活函数包括S形，双曲正切（，,和所谓的整流线性单元（ReLUs）。 生成的ax,y称为激活值，变成在空间（x，y）的特征图的值，如图1中标为C的点所示。完整的特征图，也称为激活map, 通过在input map的所有空间位置执行刚刚提到的三个操作来生成。 每个特征图都有一个内核量(kernel)和一个与之关联的偏差（bias）。目的是使用训练数据来学习内核体积的权重和每个特征图的偏差。我们将在以下两节中解释如何学习这些系数，并给出CNN应用程序的详细计算示例。

图1还说明了卷积能够提取的特征类型。输入的CNN的是一张277\*277的RGB图像，所以input volume的深度为3， 与红绿蓝三通道符合。我们使用一张人物主题的图像作为输入，使得结果特征图可视化。这里特征图有96张，每一张都由不同的核（大小为11\*11\*3）滤波，因为96个核深度为3，总的计算有volumes为3 \* 96 = 288二维的核卷积。我们对细致的特征图编号以及分组以说明体积卷积可能导致的各种互补特征。第一组展示3个特征图，其中两个（4和35）强调边缘内容，第三个（23）是输入的模糊版本。 第二组有两个贴图（10和16），这些贴图捕获互补的灰色阴影（例如，注意头发强度的差异）。在第三组中，特征图39强调对象的眼睛和衣服，在输入的RGB图像中这两者都是蓝色的。特征图45还强调了蓝色，但同时也强调了与RGB图像中红色调相对应的区域，例如对象的嘴唇，头发和皮肤。这两个特征图比其他两组中的图对颜色内容更敏感。 随后的阶段将在这些特征图上进行操作，以从数据中提取更多的抽象，如我们稍后在“计算示例”部分中所述。用于生成96个特征图的卷积核体积的权重来自AlexNet，这是CNN经过训练的CNN，使用超过1,000万个属于1,000个对象类别的图像进行了训练[5]。这个系统从未“看到”我们在图1中使用的图像。图1中所示的pooling或下采样是受研究的启发，这些研究表明，哺乳动物的大脑在视觉认知过程中执行类似的操作。pooled的map只是分辨率较低的要素地图。一种典型的池化方法是用该邻域中值的平均值替换特征图中每个大小邻域的值，例如2\*2. 使用大小为2\*2的邻域会在特征图大小的每个空间维度上生成大小为二分之一的合并map。因此，池化的结果是大量的数据减少，这有助于加快处理速度。 但是，主要缺点是，每次执行合并时，map大小也会显着减小。即使层数相对于输入图像的大小较大，即使使用2\*2大小的邻域，每个空间维度的一半缩小也很快成为问题。这是为什么在大型CNN系统中仅偶尔使用池化的原因之一。与激活功能一样，使用的池的类型在定义CNN的体系结构中也起着作用。除了邻域平均外，实践中还使用了另外两种池化方法：最大池（max pooling）和最大二层池（L2 pooling），其中最大池将邻域中的值替换为其元素的最大值。L2池为所有值的算术平方根。已证明最大池在分类大型图像数据库方面特别有效，并且具有简单性和速度性。如前所述，在图层中使用池化时，每个池化map仅从一个feature map生成，因此feature map和pooled map的数量相同。

CNN每个阶段的基本体系结构是通过指定特征图的数量以及是否在该阶段使用池来定义的。还指定了内核和池大小以及卷积跨度（stride），卷积跨度定义为卷积运算之间内核位移增量的数量。例如，跨度为2表示在input maps中的每个其他空间位置执行卷积。每个阶段所需的二维卷积核的数量等于input volumes的深度乘以特征图的数量。 一个阶段中所有内核的空间尺寸是相同的，并被指定为CNN阶段定义的一部分。 通常，在CNN的所有阶段都使用相同类型的激活。 对于网络的一个或多个阶段定义池化时使用的池化方法的大小和类型也是如此。CNN的构造有两种主要方式：完全卷积网络（不要与完全连接的网络相混淆）仅由图1所示形式的阶段串联而成。 全卷积架构的主要应用是图像分割，其中目标是标记输入图像中的每个像素。 由于图的大小随级数的增加而减小，因此使用了额外的处理（例如上采样），以便输出图的大小与输入图像的大小相同。 实际上，完全卷积网络可以“端对端”连接，这样，由于卷积，地图大小首先可以减小，然后通过相同的网络以反向过程运行，该网络的地图使用“向后”逐级增加 卷积。 最终输出是与输入大小相同的图像，但是其中的像素已标记并分组为区域[8]。

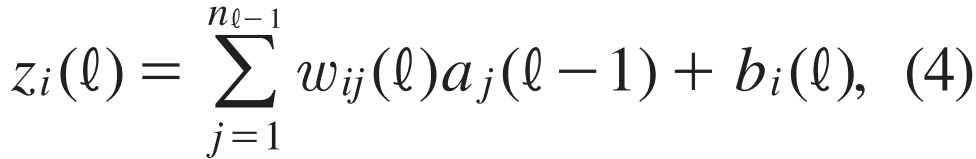
使用CNN的第二种主要方式是图像分类，如前所述，到目前为止，这是CNN的最广泛使用。 在此应用程序中，CNN的最后阶段中的输出映射被馈送到FCN，其功能是将其输入分类为预定数量的类别之一。 因为CNN的输出量由2-D映射组成，并且正如我们将在下一节中显示的那样，FCN的输入是矢量，所以CNN和FCN之间的接口是一个简单的阶段，可以将2-D数组转换为 向量。 “如何计算示例”部分讨论了如何完成所有这些操作并将其应用于解决特定问题。

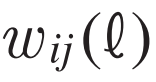
**Deep FCNs**

单个感知器是执行乘积和运算的计算单元，,在一组权重之间, ,以及一组输入标量模式特征，。由这些特征形成的向量称为模式（或特征）向量。假设Z=0，使得等式表示一个超平面，其中系数Wn+1是使超平面从相应的n维欧几里德空间的原点偏移的偏置。在“经典”感知器中，乘积和计算的输出被馈送到硬阈值h以产生激活值，a=h(z)，二进制输出通常用[-1，1]表示。然后，a=-1，或1使得h分配给不同的两类。神经网络由感知器组成，它们的激活函数从硬阈值转变为光滑函数，像sigmoid，hyperbolic tangent，ReLU。由于假定的反应与哺乳动物脑中神经元的运作方式之间存在假定的相似性，因此将所得单位称为人工神经元。图2是由人工神经元层组成的深层FCN的示意图，其中一层中每个神经元的输出都连接到下一层中每个神经元的输入，因此称为完全连接。 输入层由图案矢量的分量形成，，输出层中神经元的数量等于给定应用程序中模式类别的数量。输入和输出层是可见的，因为我们可以观察它们的输出值。神经网络中的所有其他层都是隐藏层。注意，就一个图层中的每个map中的element未连接到下一层中的map的每个元素而言，CNN并不是完全连接的

训练CNN / FCN网络的目的是确定前者中卷积体积的权重和偏差，以及后者中解决给定问题的神经元权重和偏差。 如“背景和问题陈述”部分所述，这些参数是使用反向传播估算的，反向传播是一种基于FCN输出神经元处观察到的误差值来迭代调整系数的方法。

由图2中的缩放神经元执行的计算为



是l层中第i个神经元的权重，它连接着l-1层的第j个神经元的输出

是l-1层中第j个神经元的输出

是l层中第i个神经元的偏置；



是l-1层神经元的个数；

第i个神经元的输出可以通过送入h这个非线性函数而获得



这两个简单的表达式完全表征了FCN任一层中神经元的行为。 基本上，这些等式表明，FCN任一层中神经元的输入是前一层中所有神经元的输出，而该神经元的输出是神经元权重与其输入的乘积之和。 添加一个标量值，然后将总和传递给非线性函数。 在（4）和（5）中需要注意的重要一点是，它们在形式上与（2）和（3）相同，这表明CNN和FCN执行相同类型的神经计算。 这种相似性的最终结果是，训练CNN和FCN遵循相同的计算规则，并考虑到CNN在体积（volumes）上运行而FCN在矢量（vectors）上运行这一事实。

## 参考文献

[1] F. Rosenblatt, “Two theorems of statistical sepa­ra­bility in the perceptron,” in Proc. Symp. No. 10 Mechanisation Thought Processes, London, 1959, vol. 1, pp. 421–456.

[2] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington, D.C.: Spartan, 1962.

[3] D. E. Rumelhart, G. E. Hinton, R. J. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1, D. E. Rumelhart et al., Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[4] Y. LeCun, B. Boser, J. S. Denker D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Back­propagation applied to handwritten zip code recognition,” Neural Comput., vol. 1, no. 4, pp. 541–551, 1989.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton,“ImageNet classification with deep convolutional neural networks,” in Proc. Advances Neural Information Processing Systems 25, 2012, pp. 1097–1105.

[6] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, May, 2015.

[8] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 640–651, 2017.

1. volume卷积意思是卷积的次数不只是一次，而是多个核，每个核都进行一次卷积 [↑](#footnote-ref-1)