卷积神经网络

3.1 卷积神经网络简介

卷积神经网络（Convolutional Neural Network,CNN）是一种常见的深度学习架构，其初期主要是用来解决图像识别的问题，但早期由于缺乏训练数据和计算能力，要在不产生过拟合的情况下训练高性能卷积神经网络是很困难的。近来GPU的发展，使得卷积神经网络研究涌现并取得一流结果，其表现的应用已经不仅仅应用在图像方面了，可运用在音频、自然语言处理等方面。

卷积神经网络受生物自然视觉认知机制启发而来，20世纪 90 年代，LeCun et al. 等人发表论文，确立了CNN的现代结构，后来又对其进行完善。他们设计了一种多层的人工神经网络，取名叫做LeNet-5，可以对手写数字做分类。2006年起，人们设计了很多方法，想要克服难以训练深度CNN的困难。其中，最著名的是 Krizhevsky et al.提出了一个经典的CNN 结构，并在图像识别任务上取得了重大突破。其方法的整体框架叫做 AlexNet，与 LeNet-5 类似，但要更加深一些。AlexNet 取得成功后，研究人员又提出了其他的完善方法，其中最著名的要数 VGGNet, GoogleNet和 ResNet这四种。从结构看，CNN 发展的一个方向就是层数变得更多，ILSVRC 2015 冠军 ResNet 是 AlexNet 的20 多倍，是 VGGNet 的8 倍多。通过增加深度，网络便能够利用增加的非线性得出目标函数的近似结构，同时得出更好的特性表征。但是，这样做同时也增加了网络的整体复杂程度，使网络变得难以优化，很容易过拟合。

研究人员提出了很多方法来解决这一问题。在下面的章节中，我们会先详细介绍CNN的原理、结构，并一起探讨一下LeNet、AlexNet、VGGNet，并给大家看一个有趣的小例子。

3.2卷积神经网络的原理

卷积神经网络通过卷积来模拟特征区分，并且通过卷积的权值共享及池化，来降低网络参数的数量级，最后通过传统神经网络完成分类等任务。

那么为什么不用传统的神经网络来做图像识别呢？从下面的例子就可以很容易的理解。如果我们采用传统的神经网络来处理一张1000*1000像素的黑白图片，即只有一个颜色通道，那么一张图片就有100万个像素点，如果我们连接一个相同大小的隐藏层，那么将产生100万×100万=1万亿个连接，这还仅仅是一层全连接层，计算量就已经无法接受了。我们必须减少需要训练的权重数量，一是降低训练的复杂度，二十过多的连接很容易造成过拟合，减少连接可以降低模型的泛华能力。

图像在空间上是有组织结构的，每一个像素点在空间上和周围的像素点是紧密联系的，但是和太远的点就没有太大的关系了，因此，每一个神经元并不需要接受所有的像素点的信息，只需要接

受局部的像素点作为输入，而后将这些局部信息综合起来就可以得到全局的信息。这样就把之前的全连接改变成了局部连接，如果我们取的局部信息大小是10×10，那么现在就只有10×10×100万=1亿个连接，相比之前的1万亿缩小了1万倍。

虽然我们从1万亿降低到了1亿，但是数量还是很多，因此，引入了卷积的操作，即让每一个隐藏层的节点参数一样，所有我们的参数最终只有10×10=100个即卷积核的大小，并且无论图像有多大都是100个，这就是卷积核的作用。我们不需要担心有多少个隐藏节点，图像有多大，参数量只和卷积核的大小有关系，这就是权值共享。但是如果我们只采用一个卷积核显然是不够的，每一个卷积核只能提取出图像中的一种特征，如果我们引入了多个卷积核，即可提取图像中多种特征，好在图像中的特征并不多，每一张图像都是有最基础的点、线组成，当神经元接收到这些点线特征后，传到下一层在组合成更高级的特征，比如三角形，正方形，再继续抽象出眼睛、鼻子，最后五官组合成了一张脸，从而完成了图像识别。因此我们的问题就很好解决了，只需要提供更多的卷积核，提取出更多的特征，一般来说，我们把100个卷积核放在第一个卷积层就很充足了，这样的话，我们的参数就是100×100=1万，相比之前的1亿我们又降低了10000倍。因此依靠卷积，我们就可以高效的训练局部连接神经网络了。

3.2卷积神经网络的结构

卷积神经网络一般由卷积层、池化层、全连接层组成。其中卷积层与池化层配合，组成多个卷积组，逐层提取特征，最终通过若干个全连接层完成分类。池化层主要是为了降低数据维度。

3.2.1 卷积层（Convolution）

对图像进行卷积操作实际的操作过程是一个滑动的窗口对原图像像素做乘积然后求和。如图所示，我们有一个5x5的图像，我们用一个3x3的过滤器做卷积，如果我们的滑动步长是1，可得到：

假设输入图像大小为n×n过滤器大小为f×f步长为s，则输出图像大小为：

$$\frac{(n-f+1)}{s} * \frac{(n-f+1)}{s}$$

但是这样做卷积运算是有一个缺点的，卷积图像的大小会不断缩小，另外图像的角落的元素只被一个输出所使用，所以在图像边缘的像素在输出中采用较少，也就意味着你丢掉了很多图像边缘的信息，为了解决这两个问题，就引入了padding操作，也就是在图像卷积操作之前，沿着图像边缘用0进行图像填充,就可以保证输出图像和输入图像一样大。

假设输入图像大小为n×n过滤器大小为f×f步长为引入的padding为p，则输出图像大小为：

$$[\frac{(n+2p-f)}{s}+1] * [\frac{(n+2p-f)}{s}+1]$$

在实际训练过程中，卷积核的值是在学习过程中学到的。在具体应用中，往往有多个卷积核，例如一张三原色的图片，我们需要用到三个卷积核。

3.2.2池化层 （Pooling）

池化层是CNN的重要组成部分，通过减少卷积层之间的连接，降低运算复杂程度。池化层一般有两种形式，最大池化层和平均池化层。

最大池化思想很简单，以下图为例，把4×4的图像分割成4个不同的区域，然后输出每个区域的最大值，这就是最大池化所做的事情。其实这里我们选择了2*2的过滤器，步长为2。在一幅真正的图像中提取最大值可能意味着提取了某些特定特征，比如垂直边缘、一只眼睛等等。

平均池化和最大池化唯一的不同是，它计算的是区域内的平均值而最大池化计算的是最大值。在日常应用使用最多的还是最大池化。

3.2.3 全连接层 （Fully Connect）

全连接层其实就是普通的神经网络，在经过全连接层后我们一般会再添加一个sigmoid层或softmax层来做最后的分类。

3.3 Keras实现一个卷积神经网络

本节将会带领大家实现一个简单的卷积神经网络，并在CIFAR-10数据集上进行验证。CIFAR-10是一个经典的数据集，其中包含了60000张32×32像素的彩色图像，其中训练集有50000张，测试集有10000张，CIFAR-10如同其名字，一共标注了10类数据，也就是说这将是一个10分类的图像问题，目前现有的神经网络已经对CIARF-10数据做了非常好的学习，但其复杂度也较高，本节将会使用一个简单的CNN来对数据进行训练，最终的效果在70%左右。

3.4 Keras实现经典的卷积神经网络

本章将介绍三种经典的卷积神经网络，分别是LeNet、AlexNet、VGGNet，这四种网络依照出现的先后顺序排列，深度和复杂度也一次递增。这三个卷积神经网络都在各自的年代使用了先进的网络结构，对于深度学习来说都有很大的推进作用，也象征着这几年神经网络的快速发展

3.4.1 Keras实现LeNet

3.4.2 Keras实现AlexNet

3.4.3 Keras实现VGGNet

3.5 用Keras搭建一个猫狗识别系统