

简介

卷积神经网络(Convolutional Neural Network)是一种前馈神经网络，卷积神经网络是由一个或多个卷积层和顶端的全连通层组成，同时也包括关联权重层和池化层(Pooling Layer)，这一模型可以使用反向传播算法进行训练。

卷积层

卷积层是卷积神经网络的核心层，而卷积运算又是卷积层的核心。对卷积直观的理解，就是两个函数的一种运算。

池化层

池化又称下采样，通过卷积层获得图像的特征后，理论上可以直接使用这些特征训练分类器（如softmax）。但是，这样做将面临巨大的计算量挑战，而且容易产生过拟合的现象。为了进一步降低网络训练参数及模型的过拟合程度，就要对卷积层进行池化处理。

常用的池化方法有两种：1.最大化池化MaxPooling。2.均值池化Average Pooling。更多使用的是MaxPooling

激活函数（非线性函数）

卷积神经网络与标准的神经网络类似，为保证其非线性，也需要使用激活函数，即在卷积运算后，把输出值另加偏移量，输入到激活函数，然后作为下一层的输入

若使用线性函数则加深网络就没有必要了。

Sigmoid函数

$$f(x) = \frac{1}{1 + e^{-x}}$$

Softmax 函数

$$\sigma_i(z) = \frac{e^{z_i}}{\sum_{j=1}^m e^{z_j}}$$

ReLU函数

$$f(x) = \max(0, x)$$

损失函数

损失函数Loss Function在机器学习中非常重要，因为训练模型的过程实际就是优化损失函数的过程。损失函数对每个参数的偏导数就是梯度下降中提到的梯度，防止过拟合时添加的正则化项也是加在损失函数后面。

一般有两种方法

均方误差

$$E = \frac{1}{2} \sum_k (y_k - t_k)^2$$

其中 y_k 表示神经网络的输出， t_k 表示监督数据， k 表示数据的维数

交叉熵误差

$$E = - \sum_k t_k \ln(y_k)$$

而寻找最优参数（权重和偏置），就是指损失函数取最小值时的参数。即采用梯度下降法找到最小值

反向传播

当前节点的输出对该节点的输入求偏导再乘以反向传播回的参数，此处求偏导会涉及链式求导法则。

卷积填充Padding

$n \times n$ 的图像 卷积核为 $f \times f$ ，则得到的图像为

$$(n - f + 1) \times (n - f + 1)$$

通过填充让图像不至于越来越小，于是输出变成了 $(n+2p-f+1)^{**}$

$$(n + 2p - f + 1) \times (n + 2p - f + 1)$$

选择合适的填充像素有两种方式

1.Valid卷积

也即不填充，no padding

得到的图像大小为

$$(n - f + 1) \times (n - f + 1)$$

2.Same卷积

即Pad so that output size is the same as the same input size

输入和输出的大小相同，也即

$$p = \frac{f - 1}{2}$$

且f通常是奇数

卷积步长Strided convolution

于是卷积运算后的图像大小为

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor$$

三维卷积Convolution on RGB images

与二维卷积所做的运算相同，对应的维数中的矩阵元素相乘再讲所有元素加起来得到一个数，在向后挪动

单层卷积网络

Summary of notation

If layer l is a convolution layer:

$f^{[l]} = \text{filter size}$

$p^{[l]} = \text{padding}$

$s^{[l]} = \text{stride}$

$n_c^{[l]} = \text{number of filters}$

Input : $n_H^{[l-1]} \times n_H^{[l-1]} \times n_C^{[l-1]}$

Output : $n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$

$$n_H^{[l]} = \lfloor \frac{n_H^{[l]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \rfloor$$

$$n_W^{[l]} = \lfloor \frac{n_W^{[l]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \rfloor$$

Each filter is : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Activations : $a^{[l]} = n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$

Weights : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

bias : $n_c^{[l]}$

池化层

Max pooling

最大池化层：取出核所覆盖的最大值得到池化数据，且不需要学习

Average pooling

顾名思义取出的值是核所覆盖区域的均值，但并不常用，不如max pooling好使

Summary of pooling

Hyperparameters:

f: filter size

s: stride

Max or average pooling

池化层中padding通常为0

输入为 $n_H \times n_W \times n_c$

Output :

$$\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor \times n_c$$

Fully Connect

全连接层

Neural Network example

假设一张RGB图片为 $32 \times 32 \times 3$ 卷积层1的 $f=5$, $s=1$ 则得到 $28 \times 28 \times 6$,在经过最大池化层 $f=2$, $s=2$, 得到 $14 \times 14 \times 6$ 的大小, 则卷积层1和池化层1构成第一层网络

随着网络的深度的增加每层的宽度和高度都会减少, 而信道数会增加

为何使用卷积Why convolutions?

卷积层的优势在于参数共享和稀疏连接

参数共享Parameter sharing: A feature detector(such as a vertical edge detector)that's useful in one part of the image is probably useful in another part of the image

稀疏连接Sparisty of connections:In each layer,each output value depends only on a small number of inputs.

训练神经网络，使用梯度下降算法优化神经网络中的所有参数来减少损失函数Cost J的值

卷积神经网络实例

LeNet-5

此网络是针对图像灰度进行训练，conv1为 $f=5$ ， $s=1$ ，使用avg pool $f=2$ ， $s=2$ ，

AlexNet

VGG-16

迁移学习

通过社区下载他人训练好的参数来简化自己的训练时间

数据扩充

利用镜像mirroring图片来扩充数据，或随机裁剪random cropping来扩充数据

对象检测Classification with localization

即将一张图片的某些物品检测出来

特征点检测

目标检测

Sliding windows detection 滑动窗口检测，以不同的窗口大小固定的步幅来便利图片，检测到物体则设1，否则0

Convolution implementation of sliding windows

将一个滑动窗口内的信息作为一张图片输入给滑动窗口卷积网络，滑动窗口不断滑动，此算法效率高，但不能输出精准的边界框

Output accurate bounding box

使用yolo算法，you only look once

intersection over union交并比

交并比一般大于0.5认为成功，也即交集比上并集，0.5是人为约定

非最大值抑制

可以保证算法对每个对象只检测一次，只输出概率最大的结果

卷积神经网络的特殊应用

人脸识别

One-shot learning

Learning a "similarity" function

$d(\text{img1}, \text{img2}) = \text{degree of difference between images}$

If $d(\text{img1}, \text{img2}) \leq \tau$ then the same

or $> \tau$ then the different

Siamese network

使用这一功能的方法Siamese network

输入一张图片经过网络输出一个向量标记为 $f(x^{(1)})$, 输入另外一张图片得到另外一组向量记为 $f(x^{(2)})$

然后比较 $d(x^{(1)}, x^{(2)}) = |(f(x^{(1)}) - f(x^{(2)}))|^2$

如果两张图片是同一个人，则得到的d值很小，反之很大

Triplet 损失

一张正确的图像称为anchor，一张同样是一个人的图片称为positive，另一张错误的图片称为negative，故需要同时看三张图片，称为triplet loss

$$L(A, P, N) = \max(|(f(A) - f(P))^2 - (f(A) - f(N))^2| + \alpha, 0)$$

\max 函数中第一个参数 ≤ 0

$$J = \sum_{i=1}^M L(A^{(i)}, P^{(i)}, N^{(i)})$$

Triplet是一个学习人脸识别卷积网络参数的好方法，还有其他的方法

面部验证与二分类

神经风格转移

什么是深度卷积网络