卷积神经网络

卷积神经网络（Convolutional Neural Network， CNN 或 ConvNet）是一种具有局部连接、权重共享等特性的深层前馈神经网络。

主要用来处理图像信息：1）参数太多，导致整个神经网络的训练效率会非常低，也很容易出现过拟合；2）局部不变性特征，全连接前馈网络很难提取局部不变特征，一般需要进行数据增强来提高性能。

卷积神经网络一般是由卷积层、汇聚层和全连接层交叉堆叠而成的前馈神经网络，使用反向传播算法进行训练。局部连接，权重共享以及汇聚使得卷积神经网络具有一定程度上的平移、缩放和旋转不变性。全连接层一般在卷积网络的最顶层。和前馈神经网络相比，卷积神经网络的参数更少。

卷积

一维卷积：用于计算信号的延迟累积，那么在时刻 t 收到的信号 y（t）为当前时刻产生的信息和以前时刻延迟信息的叠加：

$$y_t = \sum_{k=1}^{m} w_k \cdot x_{t-k+1} \qquad \mathbf{y} = \mathbf{w} \otimes \mathbf{x}$$

二维卷积：卷积也经常用在图像处理中。给定一个图像$X \in R^{m\times n}$，和滤波器$W \in R^{m\times n}$，一般$m \ll M$，$n \ll N$，其卷积为

$$y_{ij} = \sum_{u=1}^{m}\sum_{v=1}^{n} w_{uv} \cdot x_{i-u+1,j-v+1}.$$

互相关（Cross-Correlation）

衡量两个序列相关性的函数，通常是用滑动窗口的点积计算来实现。互相关和卷积的区别在于卷积核仅仅是否进行翻转。因此互相关也可以称为不翻转卷积。

$$y_{ij} = \sum_{u=1}^{m}\sum_{v=1}^{n} w_{uv} \cdot x_{i+u-1,j+v-1}.$$

在神经网络中使用卷积是为了进行特征抽取，卷积核是否进行翻转和其特征抽取的能力无关。特别是当卷积核是可学习的参数时，卷积和互相关是等价的。

卷积的变种

可以引入滤波器的滑动步长和零填充来增加卷积的多样性，可以更灵活地进行特征抽取。

滤波器的步长（Stride）是指滤波器在滑动时的时间间隔。

零填充（Zero Padding）是在输入向量两端进行补零

假设卷积层的输入神经元个数为 n，卷积大小为 m，步长为 s，输入神经元两端各填补 p 个零，那么该卷积层的神经元数量为(n-m+2p)/s+1。

窄卷积（Narrow Convolution）、宽卷积（Wide Convolution）、等宽卷积（Equal-Width Convolution）。

卷积的数学性质

1. **交换性**

如果不限制两个卷积信号的长度，卷积是具有交换性的，即 x⊗y=y⊗x。

2. **导数**

假设$Y = W \otimes X$，其中$X \in \mathbb{R}^{M\times N}$，$W \in \mathbb{R}^{m\times n}$，$Y \in \mathbb{R}^{(M-m+1)\times(N-n+1)}$，函数$f(Y) \in \mathbb{R}$为一个标量函数，则

$$
\begin{aligned}
\frac{\partial f(Y)}{\partial w_{uv}} &= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial y_{ij}}{\partial w_{uv}} \frac{\partial f(Y)}{\partial y_{ij}} && (5.11)\\
&= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} x_{i+u-1,j+v-1} \frac{\partial f(Y)}{\partial y_{ij}} && (5.12)\\
&= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial f(Y)}{\partial y_{ij}} x_{u+i-1,v+j-1} && (5.13)
\end{aligned}
$$

从公式(5.13)可以看出，$f(Y)$关于W的偏导数为X和$\frac{\partial f(Y)}{\partial Y}$的卷积

$$
\frac{\partial f(Y)}{\partial W} = \frac{\partial f(Y)}{\partial Y} \otimes X. \qquad (5.14)
$$

同理得到，

$$
\begin{aligned}
\frac{\partial f(Y)}{\partial x_{st}} &= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial y_{ij}}{\partial x_{st}} \frac{\partial f(Y)}{\partial y_{ij}} && (5.15)\\
&= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} w_{s-i+1,t-j+1} \frac{\partial f(Y)}{\partial y_{ij}}, && (5.16)
\end{aligned}
$$

其中当$(s-i+1) < 1$，或$(s-i+1) > m$，或$(t-j+1) < 1$，或$(t-j+1) > n$时，$w_{s-i+1,t-j+1} = 0$。即相当于对W进行了$p = (M-m, N-n)$的零填充。
时，$w_{s-i+1,t-j+1} = 0$。即相当于对W进行了$p = (M-m, N-n)$的零填充。

从公式(5.16)可以看出，$f(Y)$关于X的偏导数为W和$\frac{\partial f(Y)}{\partial Y}$的宽卷积。公式(5.16)中的卷积是真正的卷积而不是互相关，为了一致性，我们用互相关的"卷积"，即

$$
\begin{aligned}
\frac{\partial f(Y)}{\partial X} &= \mathbf{rot180}\left(\frac{\partial f(Y)}{\partial Y}\right) \tilde{\otimes} W && (5.17)\\
&= \mathbf{rot180}(W) \tilde{\otimes} \frac{\partial f(Y)}{\partial Y}, && (5.18)
\end{aligned}
$$

其中$\mathbf{rot180}(\cdot)$表示旋转180度。

卷积神经网络

卷积层、汇聚层和全连接层构成。

用卷积来代替全连接

在全连接前馈神经网络中，如果第 l 层有 n^l 个神经元，第 $l-1$ 层有 $n^{(l-1)}$ 个神经元，连接边有 $n^{(l)} \times n^{(l-1)}$ 个，也就是权重矩阵有 $n^{(l)} \times n^{(l-1)}$ 个参数。当 m 和 n 都很大时，权重矩阵的参数非常多，训练的效率会非常低。

如果采用卷积来代替全连接，第 l 层的净输入 $\mathbf{z}^{(l)}$ 为第 $l-1$ 层活性值 $\mathbf{a}^{(l-1)}$ 和滤波器 $\mathbf{w}^{(l)} \in \mathbb{R}^m$ 的卷积，即

$$\mathbf{z}^{(l)} = \mathbf{w}^{(l)} \otimes \mathbf{a}^{(l-1)} + b^{(l)}, \tag{5.19}$$

其中滤波器 $\mathbf{w}^{(l)}$ 为可学习的权重向量，$b^{(l)} \in \mathbb{R}^{n^{l-1}}$ 为可学习的偏置。

局部连接 在卷积层（假设是第 l 层）中的每一个神经元都只和下一层（第 $l-1$ 层）中某个局部窗口内的神经元相连，构成一个局部连接网络。如图5.5b所示，卷积层和下一层之间的连接数大大减少，有原来的 $n^l \times n^{l-1}$ 个连接变为 $n^l \times m$ 个连接，m 为滤波器大小。

权重共享 作为参数的滤波器$\mathbf{w}^{(l)}$对于第 l 层的所有的神经元都是相同的。

由于局部连接和权重共享，卷积层的参数只有一个 m 维的权重 $\mathbf{w}^{(l)}$ 和 1 维的偏置 $b^{(l)}$，共 $m+1$ 个参数。参数个数和神经元的数量无关。此外，第 l 层的神经元个数不是任意选择的，而是满足 $n^{(l)} = n^{(l-1)} - m + 1$。

卷积层

提取一个局部区域的特征，不同的卷积核相当于不同的特征提取器。为了更充分地利用图像的局部信息，通常将神经元组织为三维结构的神经层，其大小为高度 M×宽度 N×深度 D，有 D 个 M×N 大小的特征映射构成。

特征映射（Feature Map）为一幅图像（或其它特征映射）在经过卷积提取到的特征，每个特征映射可以作为一类抽取的图像特征。为了提高卷积网络的表示能力，可以在每一层使用多个不同的特征映射，以更好地表示图像的特征。

- 输入特征映射组：$\mathbf{X} \in \mathbb{R}^{M \times N \times D}$ 为三维张量（tensor），其中每个切片(slice) 矩阵 $X^d \in \mathbb{R}^{M \times N}$ 为一个输入特征映射，$1 \le d \le D$；
- 输出特征映射组：$\mathbf{Y} \in \mathbb{R}^{M' \times N' \times P}$ 为三维张量，其中每个切片矩阵 $Y^p \in \mathbb{R}^{M' \times N'}$ 为一个输出特征映射，$1 \le p \le P$；
- 卷积核：$\mathbf{W} \in \mathbb{R}^{m \times n \times D \times P}$ 为四维张量，其中每个切片矩阵 $W^{p,d} \in \mathbb{R}^{m \times n}$ 为一个两维卷积核，$1 \le d \le D, 1 \le p \le P$。

为了计算输出特征映射 Y^p，用卷积核 $W^{p,1}, W^{p,2}, \cdots, W^{p,D}$ 分别对输入特征映射 $X^1, X^2, \cdots, X^D$ 进行卷积，然后将卷积结果相加，并加上一个标量偏置 b 得到卷积层的净输入 Z^p，再经过非线性激活函数后得到输出特征映射 Y^p。

$$Z^p = \mathbf{W}^p \otimes \mathbf{X} + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p, \tag{5.20}$$

$$Y^p = f(Z^p). \tag{5.21}$$

在输入为$\mathbf{X} \in \mathbb{R}^{M\times N\times D}$，输出为$\mathbf{Y} \in \mathbb{R}^{M'\times N'\times P}$的卷积层中，每一个输入特征映射都需要$D$个滤波器以及一个偏置。假设每个滤波器的大小为$m \times n$，那么共需要$P \times D \times (m \times n) + P$个参数。

汇聚层

进行特征选择，降低特征数量，并从而减少参数数量(减少特征维数也可以通过增加卷积步长来实现)，但特征映射组中的神经元个数并没有显著减少。如果后面接一个分类器，分类器的输入维数依然很高，很容易出现过拟合。汇聚函数：

1. 最大汇聚（Maximum Pooling）：一般是取一个区域内所有神经元的最大值。

$$Y_{m,n}^d = \max_{i \in R_{m,n}^d} x_i, \tag{5.22}$$

其中x_i为区域R_k^d内每个神经元的激活值。

2. 平均汇聚（Mean Pooling）：一般是取区域内所有神经元的平均值。

$$Y_{m,n}^d = \frac{1}{|R_{m,n}^d|} \sum_{i \in R_{m,n}^d} x_i. \tag{5.23}$$

汇聚层不但可以有效地减少神经元的数量，还可以使得网络对一些小的局部形态改变保持不变性，并拥有更大的感受野。过大的采样区域会急剧减少神经元的数量，会造成过多的信息损失。

典型的卷积网络结构

一个卷积块为连续 M 个卷积层和 b 个汇聚层（M 通常设置为 2~5，b 为 0 或 1）。一个卷积网络中可以堆叠 N 个连续的卷积块，然后在接着 K 个全连接层（N 的取值区间比较大，比如 1~100 或者更大；K 一般为 0~2）。

整个网络结构趋向于使用更小的卷积核（比如 1×1 和 3×3）以及更深的结构(比如层数大于 50)。此外，由于卷积的操作性越来越灵活(比如不同的步长)，汇聚层的作用变得也越来越小，因此目前比较流行的卷积网络中，汇聚层的比例也逐渐降低，趋向于全卷积网络。

参数学习

卷积核中权重以及偏置：卷积网络也可以通过误差反向传播算法来进行参数学习。

不失一般性，对第l层为卷积层，第$l-1$层的输入特征映射为$\mathbf{X}^{(l-1)} \in \mathbb{R}^{M\times N\times D}$，通过卷积计算得到第$l$层的特征映射净输入$\mathbf{Z}^{(l)} \in \mathbb{R}^{M'\times N'\times P}$。第$l$层的第$p(1 \le p \le P)$个特征映射净输入

$$Z^{(l,p)} = \sum_{d=1}^{D} W^{(l,p,d)} \otimes X^{(l-1,d)} + b^{(l,p)}, \tag{5.25}$$

其中$W^{(l,p,d)}$和$b^{(l,p)}$为卷积核以及偏置。第l层中共有$P \times D$个卷积核和P个偏置，可以分别使用链式法则来计算其梯度。

根据公式(5.14)和(5.25)，损失函数关于第l层的卷积核$W^{(l,p,d)}$的偏导数为

$$\frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial W^{(l,p,d)}} = \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l,p)}} \otimes X^{(l-1,d)} \tag{5.26}$$

$$= \delta^{(l,p)} \otimes X^{(l-1,d)}, \tag{5.27}$$

其中$\delta^{(l,p)} = \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l,p)}}$为损失函数关于第$l$层的第$p$个特征映射净输入$Z^{(l,p)}$的偏导数。

同理可得，损失函数关于第l层的第p个偏置$b^{(l,p)}$的偏导数为

$$\frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial b^{(l,p)}} = \sum_{i,j} [\delta^{(l,p)}]_{i,j}. \tag{5.28}$$

卷积网络中，每层参数的梯度依赖其所在层的误差项$\delta^{(l,p)}$。

汇聚层 当第$l+1$层为汇聚层时，因为汇聚层是下采样操作，$l+1$层的每个神经元的误差项δ对应于第l层的相应特征映射的一个区域。l层的第p个特征映射中的每个神经元都有一条边和$l+1$层的第p个特征映射中的一个神经元相连。根据链式法则，第l层的一个特征映射的误差项$\delta^{(l,p)}$，只需要将$l+1$层对应特征映射的误差项$\delta^{(l+1,p)}$进行上采样操作（和第l层的大小一样），再和l层特征映射的激活值偏导数逐元素相乘，就得到了$\delta^{(l,p)}$。

第l层的第p个特征映射的误差项$\delta^{(l,p)}$的具体推导过程如下：

$$\delta^{(l,p)} \triangleq \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l,p)}} \tag{5.29}$$

$$= \frac{\partial X^{(l,p)}}{\partial Z^{(l,p)}} \cdot \frac{\partial Z^{(l+1,p)}}{\partial X^{(l,p)}} \cdot \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l+1,p)}} \tag{5.30}$$

$$= f_l'(Z^{(l,p)}) \odot \mathbf{up}(\delta^{(l+1,p)}), \tag{5.31}$$

其中$f_l'(\cdot)$为第l层使用的激活函数导数，$\mathbf{up}$为上采样函数（upsampling)，与汇聚层中使用的下采样操作刚好相反。如果下采样是最大汇聚（max pooling)，误差项$\delta^{(l+1,p)}$中每个值会直接传递到上一层对应区域中的最大值所对应的神经元，该区域中其它神经元的误差项的都设为0。如果下采样是平均汇聚（mean pooling)，误差项$\delta^{(l+1,p)}$中每个值会被平均分配到上一层对应区域中的所有神经元上。

卷积层 当$l+1$层为卷积层时，假设特征映射净输入$\mathbf{Z}^{(l+1)} \in \mathbb{R}^{M' \times N' \times P}$，其中第$p(1 \le p \le P)$个特征映射净输入

$$Z^{(l+1,p)} = \sum_{d=1}^{D} W^{(l+1,p,d)} \otimes X^{(l,d)} + b^{(l+1,p)}, \tag{5.32}$$

其中$W^{(l+1,p,d)}$和$b^{(l+1,p)}$为第$l+1$层的卷积核以及偏置。第$l+1$层中共有$P \times D$个卷积核和P个偏置。

第 l 层的第 d 个特征映射的误差项 $\delta^{(l,d)}$ 的具体推导过程如下：

$$\delta^{(l,d)} \triangleq \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l,d)}} \tag{5.33}$$

$$= \frac{\partial X^{(l,d)}}{\partial Z^{(l,d)}} \cdot \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial X^{(l,d)}} \tag{5.34}$$

$$= f_l'(Z^{(l)}) \odot \sum_{p=1}^{P} \left(\mathbf{rot180}(W^{(l+1,p,d)}) \tilde{\otimes} \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial Z^{(l+1,p)}} \right) \tag{5.35}$$

$$= f_l'(Z^{(l)}) \odot \sum_{p=1}^{P} \left(\mathbf{rot180}(W^{(l+1,p,d)}) \tilde{\otimes} \delta^{(l+1,p)} \right), \tag{5.36}$$

其中 $\tilde{\otimes}$ 为宽卷积。

几种典型的卷积神经网络

LeNet-5

卷积层的每一个输出特征映射都依赖于所有输入特征映射，相当于卷积层的输入和输出特征映射之间是全连接的关系。如果第 p 个输出特征映射依赖于第 d 个输入特征映射，则$T_{p,d}$=1，否则为 0。

$$Y^p = f\left(\sum_{\substack{d, \\ T_{p,d}=1}} W^{p,d} \otimes X^d + b^p \right), \tag{5.37}$$

其中 T 为 P×D 大小的连接表。假设连接表 T 的非零个数为 K，每个滤波器的大小为 m×n，那么共需要 K×m×n+P 参数。

AlexNet

Inception 网络

一个卷积层包含多个不同大小的卷积操作，称为 Inception 模块。Inception 网络是由有多个 inception 模块和少量的汇聚层堆叠而成。

残差网络

通过给非线性的卷积层增加直连边的方式来提高信息的传播效率。

残差单元由多个级联的（等长）卷积层和一个跨层的直连边组成，再经过 ReLU 激活后得到输出。

其它卷积方式

转置卷积

低维特征映射到高维特征的卷积操作称为转置卷积

我通过增加卷积操作的步长 s>1 来实现对输入特征的降采样操作，大幅降低特征维数。通过减少转置卷积的步长 s<1 来实现上采样操作，大幅提高特征维数。

空洞卷积

增加输出单元的感受野：（1）增加卷积核的大小；（2）增加层数；（3）在卷积之前进行汇聚操作。前两种操作会增加参数数量，而第三种会丢失一些信息。

空洞卷积(Atrous Convolution)，或称为膨胀卷积(Dilated Convolution)，是一种不增加参数数量，同时增加输出单元感受野的一种方法。通过给卷积核插

入“空洞”来变相地增加其大小。

如果在卷积核的每两个元素之间插入 d-1 个空洞，卷积核的有效大小为$m' = m + (\mathrm{m} - 1) * (\mathrm{d} - 1)$，其中 d 称为膨胀率（Dilation Rate）。