Reading Notes – Convolutional Neural Network

2023 年 9 月 26 日

目录

1 卷积神经网络 2

1.1 引入 . . . 2
1.2 卷积 (Convolution) . . . 2
1.2.1 定义&公式 . . . 2
1.2.2 互相关（Cross-Correlation） . . . 6
1.2.3 卷积的变种 . . . 6
1.2.4 卷积的数学性质 . . . 6
1.3 卷积神经网络 . . . 8
1.3.1 卷积层性质 . . . 8
1.3.2 卷积层 (Convolutional Layer) 介绍 . . . 9
1.3.3 汇聚层/池化层 (Pooling Layer Layer) 介绍 . . . 11
1.3.4 整体结构 . . . 12
1.4 参数学习 . . . 12
1.4.1 卷积神经网络的反向传播算法 . . . 13
1.4.2 卷积层 . . . 13
1.4.3 汇聚层 . . . 13
1.5 几种典型的卷积神经网络 . . . 13
1.5.1 LeNet-5 网络 . . . 13
1.5.2 AlexNet 网络 . . . 14
1.5.3 Inception 网络 . . . 15
1.5.4 残差网络 (Residual Network,ResNet) . . . 16
1.6 参考资料 . . . 17

1 卷积神经网络

1.1 引入

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.

全连接前馈网络处理图像存在问题：

1. 参数太多：隐藏层的每个神经元到输入层都有多相互独立的链接，每个连接都对应一个权重参数。随着隐藏层神经元数量增多，参数的规模增加，神经网络的训练效率低，易出现过拟合。

2. 局部不变性特征：自然图像中的物体都具有局部不变性特征，比如在尺度缩放、平移、旋转等操作不影响其语义信息。而全连接前馈网络很难获取。这些局部不变特征，一般需要进行数据增强来提高性能。

感受野 (Receptive Field)：指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。

卷积神经网络有结构上的特性：局部连接，权重共享以及汇聚 $\Rightarrow$ 使得卷积神经网络具有一定程度上有平移、缩放和旋转不变性。和前馈神经网络相比，卷积神经网络的参数更少。

1.2 卷积 (Convolution)

在泛函分析中，卷积、旋积或褶积 (英语：Convolution) 是通过两个函数 f 和 g 生成第三个函数的一种数学运算，其本质是一种特殊的积分变换，表征函数 f 与 g 经过翻转和平移的重叠部分函数值乘积对重叠长度的积分。

卷积功能：在一个图像（或某种特征）上滑动一个卷积核（即滤波器），通过卷积操作得到一组新的特征。

1.2.1 定义＆公式

- 一维卷积

1. 卷积定义：如果函数是连续的，设 $f(x), g(x)$ 是 $R1$ 上的两个可积函数，则卷积可以被表示为

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{1}$$

2. 从离散入手, 卷积是两个变量在某范围内相乘后求和的结果:

   一维卷积经常用在信号处理中，用于计算信号的延迟累积。假设一个信号发生器每个时刻 t 产生一个信号 x_t，其信息的衰减率为 w_k，即在 $k-1$ 个时间步长后，信息为原来的

w_k 倍。假设 $w_1 = 1, w_2 = 1/2, w_3 = 1/4$ 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，

$$y_t = 1 \times x_t + 1/2 \times x_{t-1} + 1/4 \times x_{t-2} \tag{2}$$

$$= w_1 \times x_t + w_2 \times x_{t-1} + w_3 \times x_{t-2} \tag{3}$$

$$= \sum_{k=1}^{3} w_k \cdot x_{t-k+1} \tag{4}$$

及我们可以将卷积过程看作翻转 $\Rightarrow$ 滑动 $\Rightarrow$ 叠加 $\Rightarrow$ 滑动 $\Rightarrow$ 叠加。

ω_1,ω_2... 称为滤波器 (Filter) 卷积核 (Convolution Kernel)。假设滤波器长度为 K，它和一个信号序列 $x_1, x_2, ...$ 的卷积为（假设 y_t 下标从 K 开始）

$$y_t = \sum_{k=1}^{K} \omega_k x_{t-k+1} \tag{5}$$

信号序列 x 和滤波器 ω 的卷积定义为

$$y = \omega * x \tag{6}$$

一般情况下滤波器长度 K 远小于信号序列 x 长度。不同情况下可以设计不同滤波器来提取信号序列不同特征。

当 $\omega = [\frac{1}{K}, ..., \frac{1}{K}]$ 时，卷积相当于信号序列的简单移动平均 (Moving Average)。

当 $\omega = [1, -2, 1]$ 时，可以近似实现对信号序列的二阶微分。

$$x''(t) = x(t+1) + x(t-1) - 2x(t) \tag{7}$$

3. 翻转原因: 一维卷积翻转原因在于“时间的相对性”，二维卷积的翻转某种程度是对一维时间序列卷积的定义延续与同意。

- 二维卷积

图像为一个两维结构，卷积也经常用在图像处理中，将一维卷积进行扩展。

设定给定一个图像 $X \in R^{M \times N}$，和滤波器 $W \in R^{U \times V}$，一般 $m << M, n << N$，其卷积为（假设 y_{ij} 的下标 (i, j) 从 (U, V) 开始。

$$y_{ij} = \sum_{u=1}^{U} \sum_{v=1}^{V} \omega_{uv} x_{i-k+1, j-v+1} \tag{8}$$

输入信息 X 和滤波器 W 的二维卷积定义为:

$$Y = W * X \tag{9}$$

特征映射：一幅图像在经过卷积操作后得到的结果。

滤波目的：过滤是信号和图像处理中基本的任务，其目的是根据应用环境的不同，选择性的提取图像中某些认为是重要的信息。

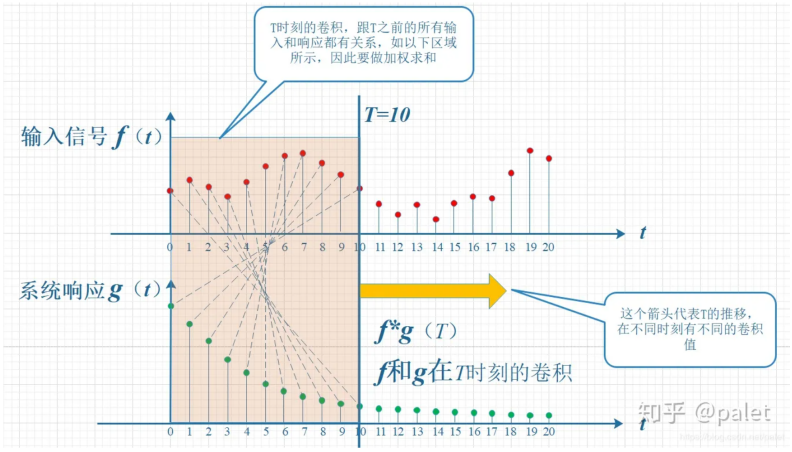


图 1: 翻转过程（1）

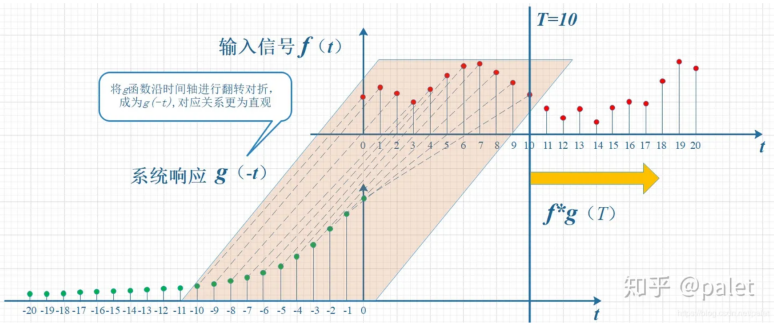


图 2: 翻转过程（2）

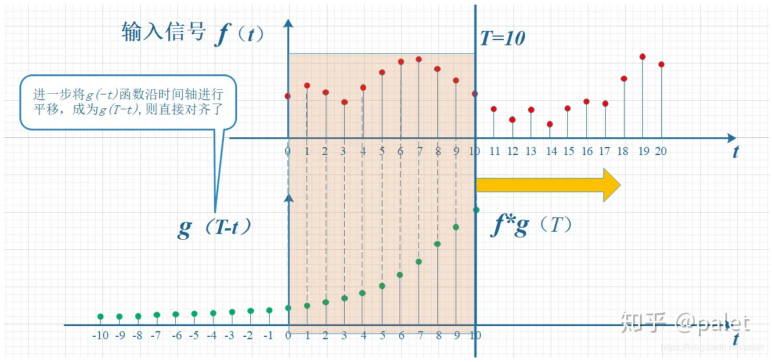


图 3: 翻转过程（3）

1. 均值滤波 (Mean Filter)：

   对目标像素及周边像素取平均值后再填回目标像素来实现滤波目的的方法, 根据公式8对 U, V 定义

$$\omega_{uv} = \frac{1}{UV} \tag{10}$$

   , 是 ω 元素值相同的一种特殊邻域滤波。

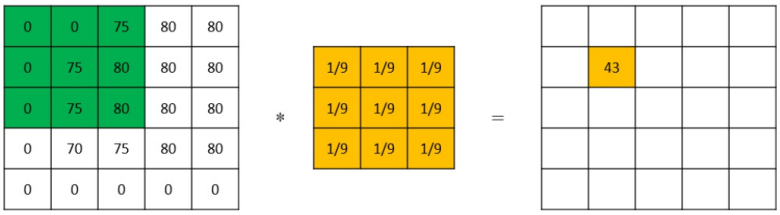


图 4: 均值滤波示意图

2. 高斯滤波器：

一种线性滤波器，能够有效的抑制噪声，平滑图像。其作用原理和均值滤波器类似，都是取滤波器窗口内的像素的均值作为输出。其窗口模板的系数和均值滤波器不同，均值滤波器的模板系数都是相同的; 而高斯滤波器的模板系数，则随着距离模板中心的增大而系数减小。高斯滤波器相比于均值滤波器对图像的模糊程度较小。

二维高斯函数举例：

$$h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{11}$$

对高斯函数进行离散化，以模板的中心位置为坐标原点进行取样, 带入坐标值，得到的高斯函数值作为模板的系数。对于上式，各元素计算公式如下：

$$H_{ij} = \frac{1}{2\pi\sigma} e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}} \tag{12}$$

计算得到形式为小数和整数的模板。

对于小数形式的模板，不需要任何处理。

对于整数形式的模板，则需要进行归一化处理，将模板左上角的值归一化为 1，并在模板的前面加一个系数 $\frac{1}{\sum_{(i,j)\in\omega}\omega_{i,j}}$, 即使用模板左上角的系数的倒数作为归一化的系数 (左上角的系数值被归一化为 1)，模板中的每个系数都乘以此值，然后将得到的值取整，就得到了整数型的高斯滤波器模板。

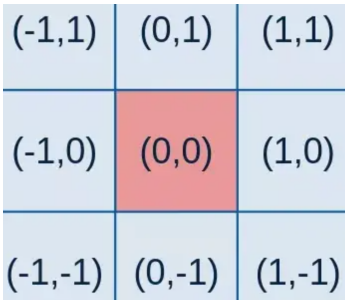


图 5: 高斯滤波器模版示意图

σ 越大，分布越分散，各部分比重差别不大，于是生成的模板各元素值差别不大，类似

于平均模板；σ 越小，分布越集中，中间部分所占比重远远高于其他部分，反映到高斯模板上就是中心元素值远远大于其他元素值，于是自然而然就相当于中间值的点运算。

1.2.2 互相关（Cross-Correlation）

互相关是一个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。

在计算卷积的过程中，理论上需要进行卷积核翻转。在具体实现上，一般会以互相关操作来代替卷积, 因此互相关也可以称作不翻转卷积。设定给定一个图像 $X \in R^{M\times N}$，和滤波器 $W \in R^{U\times V}$，它们的互相关为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

$$y_{ij} = \sum_{u=1}^{m}\sum_{v=1}^{n} \omega_{uv} \cdot x_{i+k-1,j+v-1} \tag{13}$$

另一种表述：

$$Y = W \otimes X \tag{14}$$

$$= rot180(W) * X \tag{15}$$

1.2.3 卷积的变种

在卷积的标准定义基础上，还可以引入卷积核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。

步长（Stride）是指卷积核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 M，卷积大小为 K，步长为 S，在输入两端各填补 P 个 0，那么该卷积层的神经元数量为 $\frac{(M-K+2P)}{S}+1$

1. 窄卷积（Narrow Convolution）：步长 $S=1$，两端不补零 $P=0$，卷积后输出长度为 $(M-K+1)$
2. 宽卷积（Wide Convolution）：步长 $S=1$，两端补零 $P=K-1$，卷积后输出长度为 $(M+K-1)$
3. 等宽卷积（Equal-Width Convolution）：步长 $S=1$，两端补零 $P=\frac{K-1}{2}$，卷积后输出长度为 M

1.2.4 卷积的数学性质

二维卷积性质推导

- 交换性

如果不限制两个卷积信号的长度，真正的翻转卷积是具有交换性的，即 $x*y=y*x$

1. 对于连续可导函数, 卷积定义如公式1所示，令 $t-\tau=a$，则 $\tau=t-a, d\tau=-da$

$$(f*g)(t)=\int_{-\infty}^{\infty}f(\tau)g(t-\tau)d\tau \tag{16}$$

$$=\int_{\infty}^{-\infty}f(t-a)g(a)(-da) \tag{17}$$

$$=\int_{-\infty}^{\infty}f(t-a)g(a)da \tag{18}$$

$$=(f*g)(t) \tag{19}$$

即

$$(f*g)(t)=(f*g)(t) \tag{20}$$

2. 对于卷积神经网络

互相关的“卷积”，也具有一定“交换性”。对于宽卷积之前介绍：给定一个二维图像 $X\in R^{M\times N}$ 和一个二维卷积核 $W\in R^{U\times V}$, 对图像 X 进行零填充，两端各补 $U-1$ 和 $V-1$ 个零，得到全填充 (Full Padding) 的图像 $\tilde{X}\in R^{(M+2U-2)\times(N+2V-2)}$。图像 X 和卷积核 W 的宽卷积定义为

$$W\tilde{\otimes}X\triangleq W\otimes\tilde{X} \tag{21}$$

当输入信息和卷积核有固定长度时，它们的宽卷积依然具有交换性，即

$$rot180(W)\tilde{\otimes}X\triangleq rot180(X)\tilde{\otimes}W \tag{22}$$

- 导数

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

$$\frac{\partial f(Y)}{\partial\omega_{uv}}=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}\frac{\partial y_{ij}}{\partial\omega_{uv}}\frac{\partial f(Y)}{\partial y_{ij}} \tag{23}$$

结合公式13,

$$\frac{\partial f(Y)}{\partial\omega_{uv}}=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}x_{i+u-1,j+v-1}\frac{\partial f(Y)}{\partial y_{ij}} \tag{24}$$

$$=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}\frac{\partial f(Y)}{\partial y_{ij}}x_{i+u-1,j+v-1} \tag{25}$$

$$=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}\frac{\partial f(Y)}{\partial y_{ij}}x_{u+i-1,v+j-1} \tag{26}$$

证得 $f(Y)$ 关于 W 的偏导数为 X 和 $\frac{\partial f(Y)}{\partial Y}$ 的卷积

$$\frac{\partial f(Y)}{\partial W}=\frac{\partial f(Y)}{\partial Y}\otimes X \tag{27}$$

令 $s=u+i-1, t=v+j-1$，得到 $u=s+1-i, v=t+1-j$。同理，

$$\frac{\partial f(Y)}{\partial x_{st}} = \sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}\frac{\partial y_{ij}}{\partial x_{st}}\frac{\partial f(Y)}{\partial y_{ij}} \tag{28}$$

$$= \sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V-1}\omega_{s-i+1,t-j+1}\frac{\partial f(Y)}{\partial y_{ij}} \tag{29}$$

需要区分一维卷积交换性为互相关，而二维卷积交换性需要翻转卷积核，依据公式15,

$$\frac{\partial f(Y)}{\partial X} = rot180(\frac{\partial f(Y)}{\partial Y})\tilde{\otimes}W \tag{30}$$

$$= rot180(W)\tilde{\otimes}\frac{\partial f(Y)}{\partial Y} \tag{31}$$

1.3 卷积神经网络

在全连接前馈神经网络中，如果第 l 层有 M_l 个神经元，第 $l-1$ 层有 M_{l-1} 个神经元，连接边有 $M_l \times M_{l-1}$ 个，也就是权重矩阵有 $M_l \times M_{l-1}$ 个参数．当 M_l 和 M_{l-1} 都很大时，权重矩阵的参数非常多，训绿的效率会非常低, 因此用卷积代替全连接，即：

$$z^{(l)} = \omega^{(l)} \otimes a^{(l-1)} + b^{(l)} \tag{32}$$

以上参数中：

1. $z^{(l)}$：l 层净输入
2. $a^{(l-1)}$：第 $l-1$ 层活性值
3. $\omega^{(l)} \in R^K$：卷积核（可学习权重向量）
4. K：卷积核大小
5. $b^{(l)} \in R$：可学习偏置

1.3.1 卷积层性质

1. 局部连接

   在卷积层第 l 层中的每一个神经元都只和前一层（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, 假设卷积核大小为 K．卷积层和前一层之间的连接数大大减少，由原来全连接神经网络中 $M_l \times M_{l-1}$ 个连接变为 $M_l \times K$ 个连接。

2. 权重共享

   作为参数的卷积核 $\omega_{(l)}$ 对于第 l 层的所有的神经元都是相同的。权重共享可以理解为一个卷积核只捕捉输入数据中的一种特定的局部特征。因此，如果要提取多种特征就需要使用多个不同的卷积核。

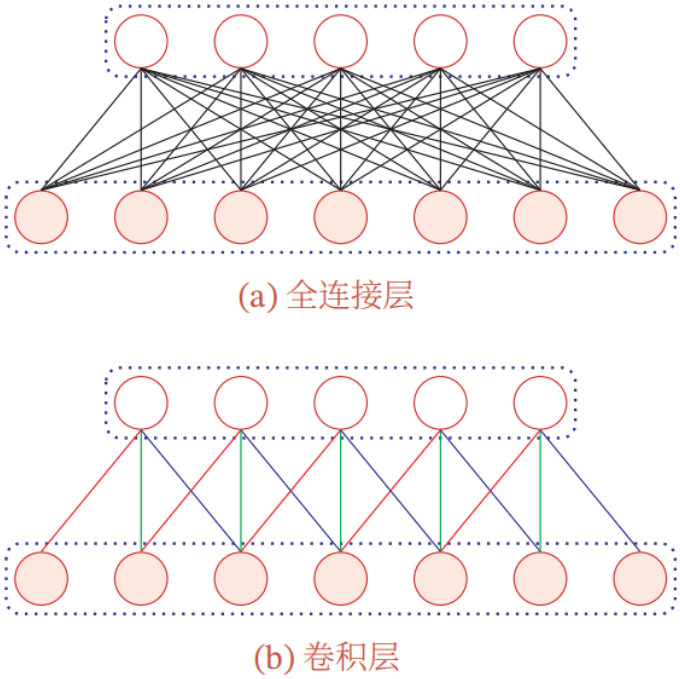


图 6: 全连接层和卷积层对比

由于局部连接和权重共享，卷积层的参数只有一个 K 维的权重 $\omega^{(l)}$ 和 l 维的偏置 $b^{(l)}$。共 $K+1$ 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 $M_{(l)} = M_{(l-1)} - K + 1$。

当卷积有深度时卷积层中参数个数 $= (K+1)\times$ 层数

1.3.2 卷积层 (Convolutional Layer) 介绍

卷积层的作用是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．上一节中描述的卷积层的神经元和全连接网络一样都是一维结构．由于卷积网络主要应用在图像处理上，而图像为二维结构，因此为了更充分地利用图像的局部信息，通常将神经元缀织为三维结构的神经层，其大小为 M(高/长)$\times N$(宽)$\times D$(深), 由 D 个 $M \times N$ 大小的特征映射构成 (默认一个特征映射的厚度为 1)。

特征映射 (Feature Map): 为一幅图像（或其他特征映射）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．

假设一个卷积层的结构如下：

1. 输入特征映射组：$\chi \in R^{M\times N\times D}$ 为三维张量（Tensor），其中每个切片（Slice）矩阵 $X^d \in R^{M\times N}$ 为一个输入特征映射，$1 \leqslant d \leqslant D$

2. 输出特征映射组：$\boldsymbol{y} \in R^{M'\times N'\times P}$ 为三维张量（Tensor），其中每个切片（Slice）矩阵

$Y^p \in R^{M' \times N'}$ 为一个输入特征映射，$1 <= p <= P$

3. 卷积核：$W \in R^{U \times V \times P \times D}$

计算输出特征 Y^p:

用卷积核 $W^{p,1}, W^{p,1}, ...W^{p,D}$ 分别对输入特征映射 $X^1, X^2, ..., X^D$ 进行卷积，然后将卷积结果相加，并加上一个标量偏置 b^p 得到卷积层的净输入 Z^p，（是指没有经过非线性激活函数的净活性值（Net Activation））, 再经过非线性激活函数后得到输出特征映射 Y_p.

$$Z^p = W^p \odot X + b^p = \sum_{d=1}^{D} W^{p,d} \odot X^d + b^p \tag{33}$$

$$Y^p = f(Z^p) \tag{34}$$

四维向量理解: 在之前步骤中，假设输入图片只有一张图片，而在实际情况中，图片存在“通道”的概念，输入图片包含通道，每张输入图片都需要一个单独的二维卷积核进行二维卷积运算, 因此需要乘输入特征映射个数 D, 卷积核从二维向量变为三维向量 $M \times N \times D$。输

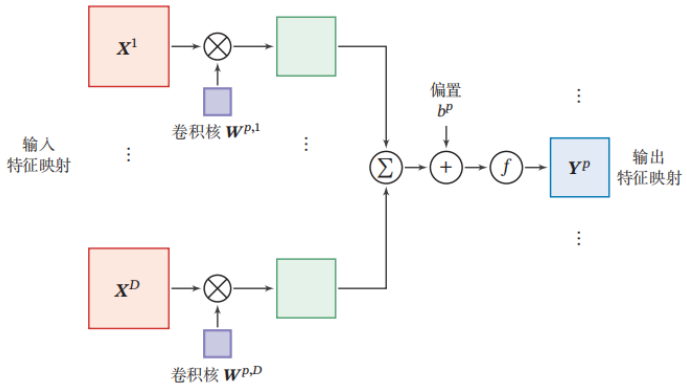


图 7: 卷积层中从输入特征映射组 X 到输出特征映射 Y^p 的计算示例

入图片里的每个 $M \times N$ 的二维向量和卷积核里的每个 $m \times n$ 的二维向量做二维卷积运算，可以得到 D 个 $M' \times N'$ 的二维向量 D 个 $M' \times N'$ 的二维向量求和，加上偏置，构成一个 $M' \times N'$ 的二维向量，也就是一张输出图片。要得到 P 张输出图片，需要将上述操作重复 P 次，因此得到了四维向量 $W \in R^{U \times V \times P \times D}$。

在此式上推导，当第 $(l+1)$ 层为卷积层时，此时特征映射净输入 $\mathcal{Z}^{(l+1)} \in R^{M' \times N' \times P}$, 其中第 p 个特征映射净输入：

$$Z^{l+1,p} = \sum_{d=1}^{D} W^{(l+1,p,d)} \otimes X^{(l,d)} + b^{(l+1,p)} \tag{35}$$

以图像颜色通道 (RGB) 为例，RGB 共有三通道及三个不同的卷积核，即 $D = 3$。从三个通道分别计算此位置特征值，最终特征值为三通道特征值之和。

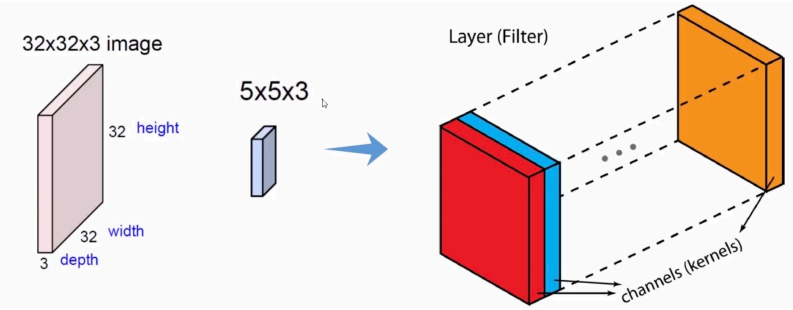


图 8: RGB 特征提取示意图

1.3.3 汇聚层/池化层 (Pooling Layer Layer) 介绍

为什么能进行汇聚：

图像中的相邻像素倾向于具有相似的值，卷积层相邻的输出像素也具有相同的值，意味着卷积层输出中包含的很多信息是冗余的。

汇聚层也叫子采样层 (Subsampling Layer)，其作用是进行特征选择，降低特征数量，从而减少参数数量．卷积操作减少了参数数量但并未显著减少神经元数量，因此需要用汇聚层减少特征（神经元）数量，避免过拟合。

假设汇聚层的输入特征映射组为 $\chi \in R^{M\times N\times D}$，对于其中每一个特征映射 $X^d \in R^{M\times N}$，$1 \leqslant d \leqslant D$，将其划分为很多区域 $R^d_{m,n}, 1 \leqslant m \leqslant M', 1 \leqslant n \leqslant N'$。这些区域可以重叠，也可以不重叠．汇聚 (Pooling) 是指对每个区域进行下采样得到一个值，作为这个区域的概括。

下采样 (Down Sampling)：即从多数集中选出一部分数据与少数集重新组合成一个新的数据集,，是一个数据损失的过程。在实际运用中，下采样可以：

1. 使得图像符合显示区域的大小
2. 生成对应图像的缩略图

以下为两种常用汇聚：

1. 最大汇聚（Maxinum Pooling or Max Pooling）：对于某区域 $R^d_{m,n}$,x_i 为区域 R^d_k 内每个神经元的活性值，选择这个区域内所有神经元的最大活性值作为这个区域的表示，即

$$y^d_{m,n} = \underset{i\in R^d_{m,n}}{max}\ x_i \tag{36}$$

2. 平均汇聚（Wide Convolution）：取区域内所有神经元活性值的平均值，即

$$y^d_{m,n} = \frac{1}{R^d_{m,n}} \sum_{i\in R^d_{m,n}} x_i \tag{37}$$

目前主流的卷积网络中，汇聚层仅包含下采样操作．但在早期的一些卷积网络（比如 LeNet-5）中，有时也会在汇聚层使用非线性激活函数，比如

$$y^{'d} = f(\omega^d Y^d + b^d) \tag{38}$$

为什么进行非线性映射：是为了使多层神经网络具有实际意义。若进行多层线性映射，不管层数的多少，最终会与单层感知机等效。

为什么引用 Relu 函数：后续反向传播，求误差梯度时，求导涉及除法。对于深层网络 Sigmoid 函数反向传播时，易出现梯度消失状况，Relu 使一部分参数为 0，减少了参数间相互依存关系。

1.3.4 整体结构

一个典型的卷积网络是由卷积层、汇聚层、全连接层交叉堆叠而成。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，然后在后面接 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 $0-2$）．

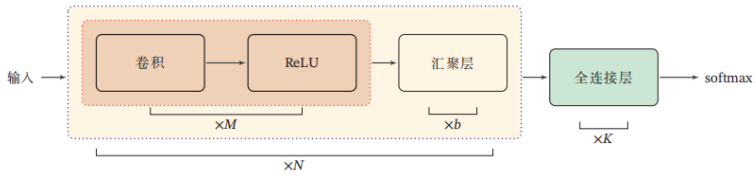


图 9: 卷积网络结构

1.4 参数学习

在卷积网络中，参数为卷积核中权重以及偏置．在全连接前馈神经网络中，梯度主要通过每一层的误差项 δ 进行反向传播，并进一步计算每层参数的梯度。

根据定义，只计算卷积层中参数梯度。令对第 l 层为卷积层，第 $l-1$ 层的输入特征映射为 $\chi \in R^{M\times N\times D}$, 通过卷积计算得到第 l 层的特征映射净输入 $Z^{(l)} \in R^{M'\times N'\times P}$, 第 l 层的第 $p(1 \leqslant p \leqslant P)$ 个特征映射净输入为

$$Z^{(l,p)} = \sum_{d=1}^{D} W^{(l,p,d)} \otimes X(l-1,d) + b^{(l,p)} \tag{39}$$

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．

令 $\delta^{(l,p)} = \frac{\partial \mathcal{L}}{\partial Z^{(l,p)}}$ 根据公式39和公式39，损失函数 $\mathcal{L}$ 关于第 l 层的卷积核 $W^{(l,p,d)}$ 的偏导为：

$$\frac{\partial \mathcal{L}}{\partial W^{l,p,d}} = \frac{\partial \mathcal{L}}{\partial Z^{(l,p)}} \otimes X^{(l-1,d)} \tag{40}$$

$$= \delta^{(l,p)} \otimes X^{(l-1,d)} \tag{41}$$

损失函数 $\mathcal{L}$ 关于第 l 层的第 p 个偏置 $b^{(l,p)}$ 的偏导为：

$$\frac{\partial \mathcal{L}}{\partial b^{(l,p)}} = \sum_{i,j} [\delta(l,p)]_{i,j} \tag{42}$$

1.4.1 卷积神经网络的反向传播算法

在卷积神经网络中，主要有两种不同功能的神经层：卷积层和汇聚层，他们的误差项的计算有所不同，需要分开计算。

1.4.2 卷积层

当第 $(l+1)$ 层为卷积层时，依据公式35，第 l 层第 d 个特征映射的误差项 $\delta^{(l,d)}$ 为：

$$\delta^{(l,d)} = \frac{\partial \mathcal{L}}{\partial Z^{(l,d)}} \tag{43}$$

$$= \frac{\partial X^{(l,d)}}{\partial Z^{(l,d)}} \frac{\partial \mathcal{L}}{\partial X^{(l,d)}} \tag{44}$$

$$= f_l'(Z^{(l,d)}) \odot \sum_{P=1}^{P} = 1(rot180(W^{(l+1,p,d)} \tilde{\otimes} \frac{\partial \mathcal{L}}{\partial Z^{(l+1,p)}}) \tag{45}$$

$$= f_l'(Z^{(l,d)}) \odot \sum_{P=1}^{P} = 1(rot180(W^{(l+1,p,d)} \tilde{\otimes} \delta^{(l+1,p)}) \tag{46}$$

1.4.3 汇聚层

当第 $(l+1)$ 层为汇聚层时，第 l 层第 d 个特征映射的误差项 $\delta^{(l,d)}$ 为 (结合汇聚层为下采样操作)，$f_l'(\cdot)$ 为第 l 层使用的激活函数导数,up 为上采样函数。

如果下采样是最大汇聚，误差项 $\delta^{(l+1,p)}$ 中每个值会直接传递到前一层对应区域中的最大值所对应的神经元，该区域中其他神经元的误差项都设为 0。

如果下采样是平均汇聚，误差项 $\delta^{(l+1,p)}$ 中每个值会被平均分配到前一层对应区域中的所有神经元上。

依据公式35，第 l 层第 d 个特征映射的误差项 $\delta^{(l,d)}$ 为：

$$\delta^{(l,d)} = \frac{\partial \mathcal{L}}{\partial Z^{l,p}} \tag{47}$$

$$= \frac{\partial X^{(l,p)}}{\partial Z^{(l,p)}} \frac{\partial Z^{(l+1,p)}}{\partial X^{(l,p)}} \frac{\partial \mathcal{L}}{\partial Z^{(l+1,p)}} \tag{48}$$

$$= f_l'(Z^{(l,p)}) \odot up(\delta^{(l+1,p)}) \tag{49}$$

1.5 几种典型的卷积神经网络

1.5.1 LeNet-5 网络

此网络中所有激活函数采用 Sigmoid(F_6 结果通过其输出) 在从 S_2 汇聚层到 C_3 卷积层的过程中，运用了特殊手段连接表，得到了 16 组 10×10 的输出。连接表如图11所示：

径向基函数 (Radial Basis Function, RBF): 径向基函数是某种沿径向对称的标量函数，通常定义为样本到数据中心之间径向距离（通常是欧氏距离）的单调函数（由于距离是径向同性的）。

层数	输入	卷积核/池化尺寸	参数个数	连接数	输出
输入层	32×32				
C_1	1024	6 个 5×5 $(S=1)$	$6 \times 25 + 6$	156×784	6 组 $(32-5+1)^2$
S_2	4704	2×2 平均汇聚	$6 \times (1+1)$	$6 \times 196 \times (4+1)$	$6 \times 14 \times 14$
C_3	1176	60 个 5×5	$60 \times 25 + 16$	100×1516	16 组 10×10
S_4	1600	下采样	16×2	$16 \times 25 \times (4+1)$	16 组 5×5
C_5	400	1920 个 5×5	$1920 \times 25 + 120$	$120 \times (16 \times 25 + 1)$	120 组 1×1
F_6	120	–	$84 \times (120 + 1)$	10164	
输出层					对应类别得分

表 1: LeNet-5 结构梳理

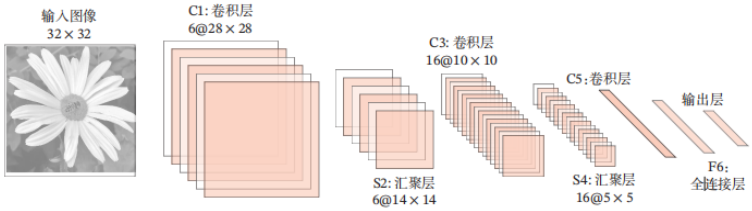


图 10: LeNet-5 结构

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

图 11: LeNet-5 中 C3 层连接表

1.5.2 AlexNet 网络

AlexNet 的结构如图12所示，包括 5 个卷积层、3 个汇聚层和 3 个全连接层（其中最后一层是使用 Softmax 函数的输出层）. 因为网络规模超出了当时的单个 GPU 的内存限制，将网络拆为两半，分别放在两个 GPU 上，GPU 间只在某些层（比如第 3 层）进行通信

局部响应归一化优势：

1. 为后面数据处理的方便，归一可以避免一些不必要的数值问题。

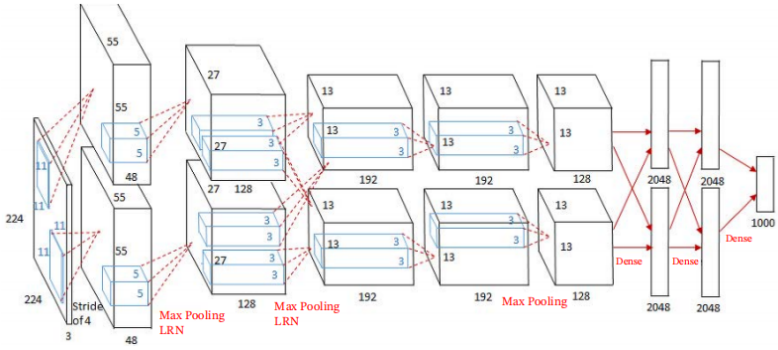


图 12: AlexNet 结构

层数	输入	卷积核/池化尺寸	步长	零填充	输出
输入层	$224 \times 224 \times 3$				
1^{st} 卷积层	$224 \times 224 \times 3$	2 个 $11 \times 11 \times 3 \times 3 \times 48$ 卷积核	4	3	2 个 $55 \times 55 \times 48$
1^{st} 汇聚层	2 个 $55 \times 55 \times 48$	3×3 最大汇聚	2	–	2 个 $27 \times 27 \times 48$
2^{nd} 卷积层	2 个 $27 \times 27 \times 48$	2 个 $5 \times 5 \times 48 \times 128$ 卷积核	1	2	2 个 $27 \times 27 \times 128$
2^{nd} 汇聚层	2 个 $27 \times 27 \times 128$	3×3	2	–	2 个 $13 \times 13 \times 128$
3^{rd} 卷积层	2 个 $13 \times 13 \times 128$	1 个 $3 \times 3 \times 256 \times 384$	1	1	2 个 $13 \times 13 \times 192$
4^{th} 卷积层	2 个 $13 \times 13 \times 192$	2 个 $3 \times 3 \times 192 \times 128$	1	1	2 个 $13 \times 13 \times 128$
5^{th} 卷积层	2 个 $13 \times 13 \times 192$	2 个 $3 \times 3 \times 192 \times 192$	1	1	2 个 $13 \times 13 \times 192$
3^{th} 汇聚层	2 个 $13 \times 13 \times 192$	3×3 最大汇聚	2	–	2 个 $6 \times 6 \times 128$
全连接 1	2 个 $6 \times 6 \times 128$				4096
全连接 2	4096				4096
全连接 3	1000				1000
输出层					1000 个类别条件概率

表 2: AlexNet-5 结构梳理

2. 同一量纲。样本数据的评价标准不一样，需要对其量纲化，统一评价标准。这算是应用层面的需求。

3. 避免神经元饱和。

4. 保证输出数据中数值小的不被吞食。

1.5.3 Inception 网络

- Inception 网络: 在 Inception 网络中，一个卷积层包含多个不同大小的卷积操作。

Inception 网络同时使用不同大小的卷积核，并将得到的特征映射在深度上拼接（堆叠）起来作为输出特征映射。Inception 网络有多种版本，最早版本 (GoogLenet) 如图14:

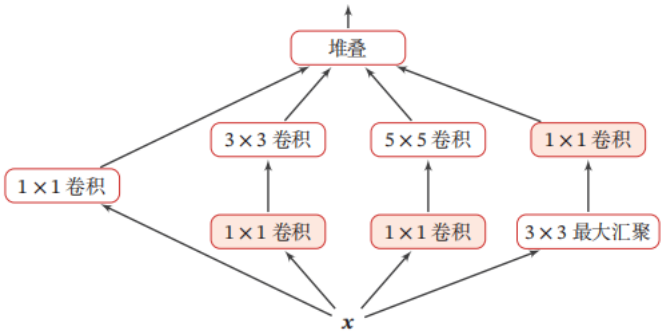


图 13: Inception v1 的模块结构

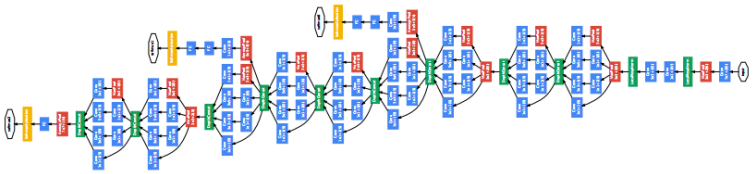
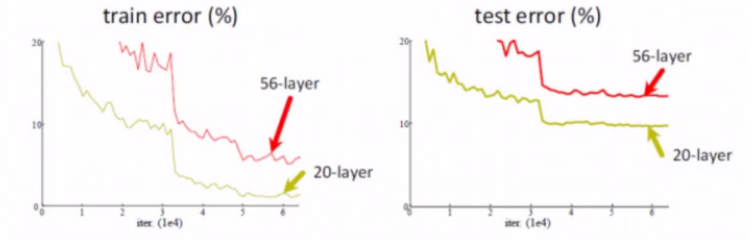


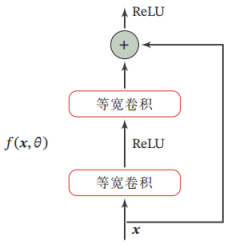
图 14: GoogLeNet 网络结构

1.5.4 残差网络 (Residual Network,ResNet)

通过给非线性的卷积层增加直连边 (Shortcut Connection) (也称为残差连接 (Residual Connection)) 的方式来提高信息传播效率。



(a) 深层网络遇到的问题



(b) 一个简单残差单元

图 15: 残差网络

如图15a，卷积神经网络目标是随着层数增加训练集误差和测试集误差都减小，但由于随层数增多，会随之出现表现较差部分，因此提出了同等映射 (Equivalent Mapping) 和残差网络。

在一个深度网络中，用一个非线性单元 $f(x:\theta)$ 逼近一个目标函数 $h(x)$。将目标函数拆分成两部分：恒等函数 (Identity Function) 和残差函数 (Residue Function)$h(x)-x$。

$$h(x) = x + (h(x) - x) \tag{50}$$

残差网络就是将很多个残差单元串联起来构成的一个非常深的网络。

- 高速网络 (Highway Network) 对比传统神经网络，高速网络引入 transform gate T 和 carry gate C。

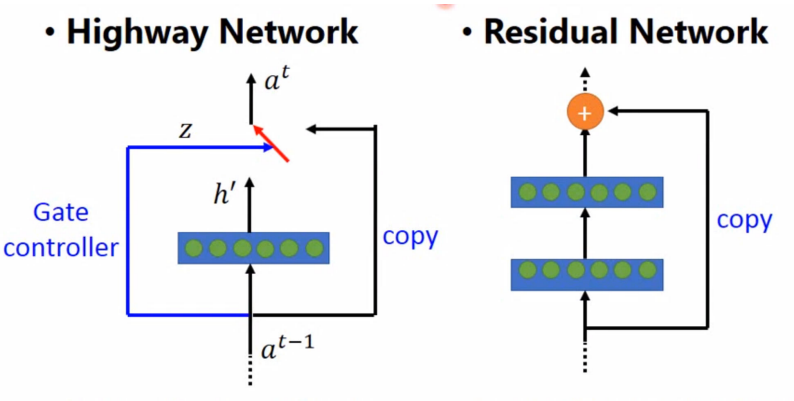


图 16: 残差网络与高速网络对比

$$y = H(x, W^H) \cdot T(x, W_T) + x \cdot C(x, W_c) \tag{51}$$

1.6 参考资料

@misc 高斯滤波器 note = https://zhuanlan.zhihu.com/p/165146141,

@misc 卷积神经网络动态图示 note = https://www.bilibili.com/video/BV1x44y1P7s2/?buvid=YD45E5D98FE2F9BB4C11AA086B57ED660107&is_story_h5=false&mid=tyOV3JrnXjiUl1kKsH7Jbw%3D%3D&p=1&plat_id=114&share_from=ugc&share_medium=ipad&share_plat=ios&share_source=WEIXIN&share_tag=s_i&timestamp=1695534145&unique_k=kc8aiMZ&up_id=223755925,

@misc 卷积积分 note = https://blog.csdn.net/SUKHOI27SMK/article/details/102577433,

@miscLeNet-5 可视化 note = https://www.bilibili.com/video/BV1Z54y187WW/?buvid=YD45E5D98FE2F9BB4C11AA086B57ED660107&is_story_h5=false&mid=tyOV3JrnXjiUl1kKsH7Jbw%3D%3D&p=1&plat_id=122&share_from=ugc&share_medium=ipad&share_plat=ios&share_session_id=11A6F158-B13C-40BD-ACED-0398E9E47766&share_source=WEIXIN&share_tag=s_i&timestamp=1695574535&unique_k=xURS3wc&up_id=137653110,

@misc 径向基函数 note = https://https://blog.csdn.net/qq_42732229/article/details/109514793,ι