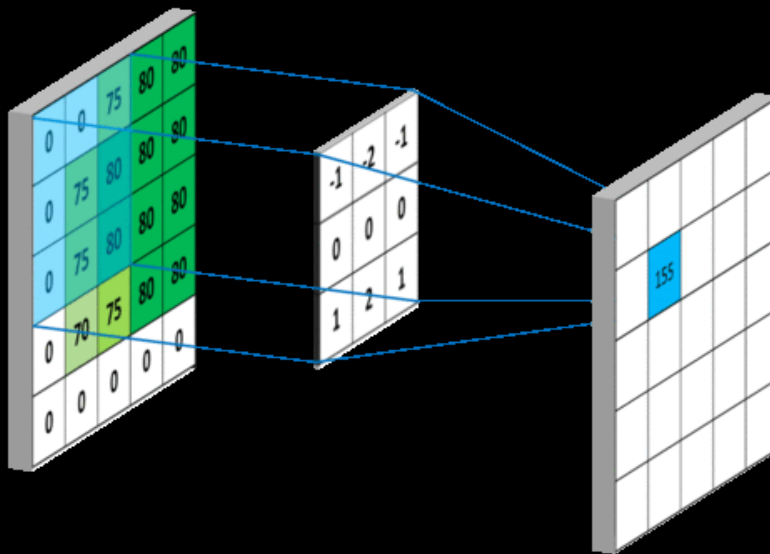


1.一次卷积的计算量(时间复杂度)

2.卷积参数量 (空间复杂度)

3.举个例子 vgg-16

1.一次卷积的计算量(时间复杂度)



<https://blog.csdn.net/mtj66>

一次卷积的计算量,如何计算呢,参考上图,只是其中一个channel,

$$((K * K + 1) * C_{in} + (C_{in} - 1)) * M * M * C_{out}$$

其中(这需要对卷积过程有较深的理解)

$(K * K + 1)$ 表示在一个卷积核计算时在叠加bias,

乘以 C_{in} 表示在输入的channel方向进行乘积

$C_{in} - 1$ 表示一次卷积之后进行 C_{in} 方向的叠加

$((K * K + 1) * C_{in} + (C_{in} - 1))$ 表示最终汇聚成特征图上的一个点

乘以 $M * M * C_{out}$ 表示在最终的输出特征图,所有点的个数

当然简化之后一次卷积的时间复杂度: $O(K * K * C_{in} * M * M * C_{out})$

忽略bias参数,简化之后参数: $K * K * C_{in} * C_{out}$

M 每个卷积核输出特征图 (*Feature Map*) 的边长

K 每个卷积核 (*Kernel*) 的边长

C_{in} 每个卷积核的通道数,也即输入通道数,也即上一层的输出通道数。

C_{out} 本卷积层具有的卷积核个数,也即输出通道数。

可见,每个卷积层的时间复杂度由输出特征图面积 M^2 、卷积核面积 K^2 、输入 C_{in} 和输出通道数 C_{out} 完全决定。

其中,输出特征图尺寸本身又由输入矩阵尺寸 X 、卷积核尺寸 K 、*Padding*、*Stride* 这四个参数所决定,表示如下:

$$M = (X - K + 2 * Padding) / Stride + 1$$

注1: 为了简化表达式中的变量个数,这里统一假设输入和卷积核的形状都是正方形。

注2: 严格来讲每层应该还包含 1 个 *Bias* 参数,这里为了简洁就省略了。

2.卷积参数量 (空间复杂度)

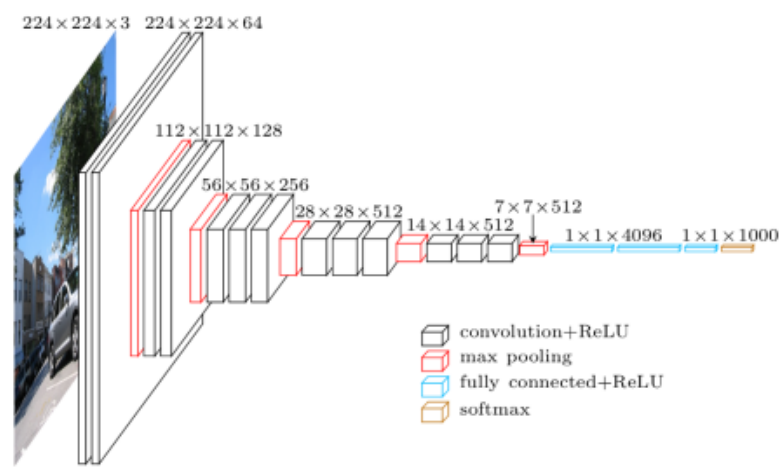
$K * K * C_{in} * C_{out}$

3.举个例子 vgg-16

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as “conv(receptive field size)-(number of channels)”. The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64	conv3-64	conv3-64	conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128	conv3-128	conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

http://blog.csdn.net/qq_25737169



2. 卷积操作的计算量和参数量

对于卷积操作的计算量（时间复杂度）和参数量（空间复杂度）可以看这篇文章：[卷积神经网络的复杂度分析-Michael Yuan的文章](#)

注意，这些复杂度计算都是估算，并非精确值。

我们以VGG-16的第一层卷积为例：输入图像 $224 \times 224 \times 3$ ，输出 $224 \times 224 \times 64$ ，卷积核大小 3×3 。

计算量：

$$Times \approx 224 \times 224 \times 3 \times 3 \times 3 \times 64 = 8.7 \times 10^7$$

参数量：

$$Space \approx 3 \times 3 \times 3 \times 64 = 1728$$

再举一个例子，VGG-16的最后一个卷积层：输入 $14 \times 14 \times 512$ ，输出 $14 \times 14 \times 512$ ，卷积核大小 3×3 。

计算量：

$$Times \approx 14 \times 14 \times 3 \times 3 \times 512 \times 512 = 4.6 \times 10^8$$

参数量：

$$Space \approx 3 \times 3 \times 512 \times 512 = 2.4 \times 10^6$$

3. 全连接层的计算量和参数量

考虑VGG-16的最后一个全连接层：上层神经元数为4096，下层神经元数为1000。这样的全连接层复杂度应该如何计算？

其实全连接层可以视为一种特殊的卷积层：上层为 $1 \times 1 \times 4096$ ，下层为 $1 \times 1 \times 1000$ ，使用的 1×1 的卷积核进行卷积。

那么，计算量：

$$Times \approx 1 \times 1 \times 1 \times 1 \times 4096 \times 1000 = 4 \times 10^6$$

参数量：

$$Space \approx 1 \times 1 \times 4096 \times 1000 = 4 \times 10^6$$

4. VGG-16复杂度分析

从上述计算中，相信大家对深度网络的复杂度已经有了一些体会，比如VGG-16中：

- 1、卷积层的时间复杂度大致是同一数量级的
- 2、随着网络深度加深，卷积层的空间复杂度快速上升（每层的空间复杂度是上层的两倍）
- 3、全连接层的空间复杂度比卷积层的最后一层还大

当然，深度网络的复杂度是和网络结构紧密相关的，**上述3个结论仅对VGG这种网络结构有效。**

还可以看这个：<https://zhuanlan.zhihu.com/p/31575074>
