

卷积神经网络基础



目 录

- 一. 计算机视觉
- 二. 卷积基础操作
- 三. 简单的卷积网络
- 四. 池化层
- 五. 卷积网络实例



一. 计算机视觉

计算机视觉问题

计算机视觉（Computer Vision）是深度学习应用的主要方向之一。

一般的CV问题包括以下三类：

图像识别；目标检测；神经风格转换。

Computer Vision Problems

Image Classification



Neural Style Transfer



Object detection



Andrew Ng



一. 计算机视觉

计算机视觉存在的问题

- (1) 使用传统神经网络处理计算机视觉的一个主要问题是输入层维度很大。例如一张 $64 \times 64 \times 3$ 的图片，神经网络输入层的维度为12288。如果图片尺寸较大，例如一张 $1000 \times 1000 \times 3$ 的图片，神经网络输入层的维度将达到3百万，使得网络权重 W 非常庞大。
- (2) 这样会造成两个后果，一是神经网络结构复杂，数据量相对不够，容易出现过拟合；二是所需内存、计算量较大。
- (3) 解决这一问题的方法就是使用卷积神经网络（CNN）。

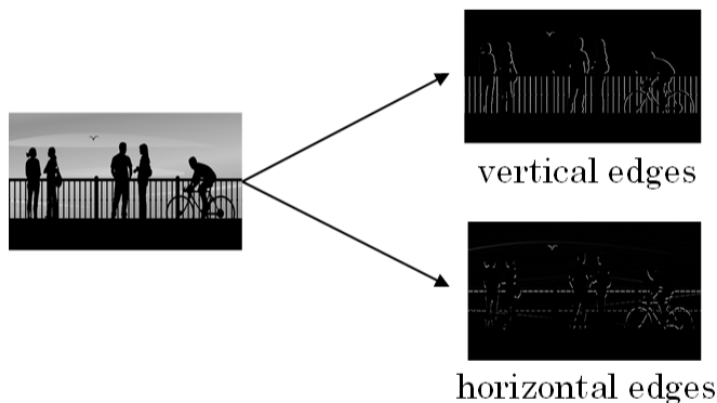


二. 卷积基础操作

边缘检测

最常检测的图片边缘有两类：一是垂直边缘（**vertical edges**），二是水平边缘（**horizontal edges**），如左下图。

图片的边缘检测可以通过与相应滤波器进行**卷积**来实现。以垂直边缘检测为例，原始图片尺寸为**6x6**，滤波器**filter**尺寸为**3x3**，卷积后的图片尺寸为**4x4**，得到结果如右下图所示：



Vertical edge detection

3 ¹	0 ⁰	1 ⁻¹	2	7	4
1 ¹	5 ⁰	8 ⁻¹	9	3	1
2 ¹	7 ⁰	2 ⁻¹	5	1	3
0	1	3	1 ¹	7 ⁰	8 ⁻¹
4	2	1	6 ¹	2 ⁰	8 ⁻¹
2	4	5	2 ¹	3 ⁰	9 ⁻¹

*

1	0	-1
1	0	-1
1	0	-1

filter

=

-5			
			-16



二. 卷积基础操作

边缘检测

图片边缘有两种渐变方式，一种是由明变暗，另一种是由暗变明。
以垂直边缘检测为例，下图展示了两两种方式的区別。实际应用中，这两种渐变方式并不影响边缘检测结果，可以对输出图片取绝对值操作，得到同样的结果。

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

 =

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

*

1	0	-1
1	0	-1
1	0	-1

 =

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0



二. 卷积基础操作

边缘检测的滤波器算子

垂直边缘检测和水平边缘检测的滤波器算子如下所示：

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

下图展示一个水平边缘检测的例子：

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

*

1	1	1
0	0	0
-1	-1	-1

=

0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0



二. 卷积基础操作

滤波器与深度学习的关系

(1) 在深度学习中，如果想检测图片的各种边缘特征，而不仅限于垂直边缘和水平边缘，那么滤波器（filter）的数值一般需要通过模型训练得到，类似于标准神经网络中的权重 W 一样由梯度下降算法反复迭代求得。

(2) 卷积神经网络（Convolutional neural network, CNN）的主要目的就是计算出这些filter的数值。确定得到了这些filter后，CNN浅层网络也就实现了对图片所有边缘特征的检测。

3 x 3大小的filter



3 x 3



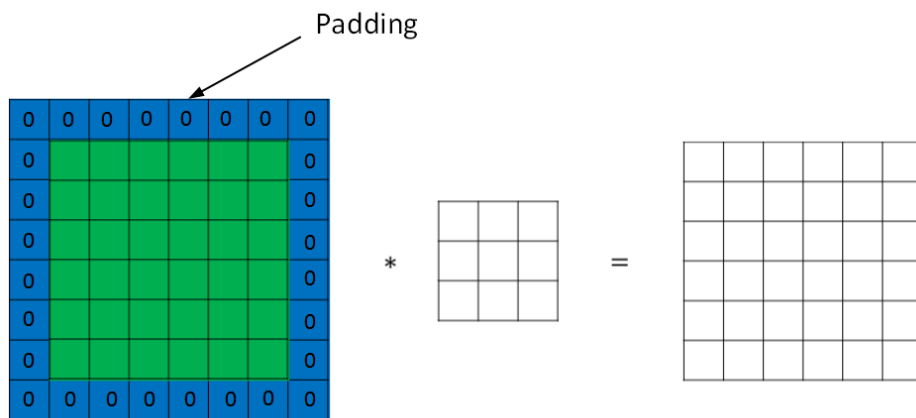
二. 卷积基础操作

填充 (Padding)

如果原始图片尺寸为 $n \times n$ ，filter尺寸为 $f \times f$ ，则卷积后的图片尺寸为 $(n-f+1) \times (n-f+1)$ ，注意 f 一般为奇数。这样会带来两个问题：

- (1) 卷积运算后，输出图片尺寸**缩小**
- (2) 原始图片边缘信息对输出贡献得少，输出图片**丢失**边缘信息

为了解决图片缩小的问题，可以使用padding方法，即把原始图片尺寸进行扩展，扩展区域补零，用 p 来表示每个方向扩展的宽度。



二. 卷积基础操作

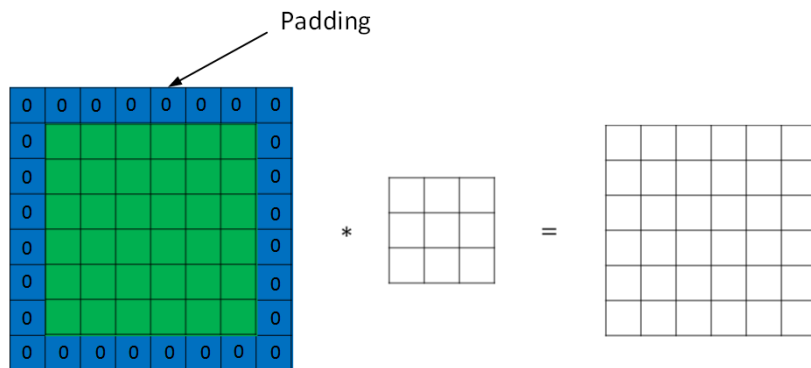
填充 (Padding)

经过padding之后, 原始图片尺寸为 $(n+2p) \times (n+2p)$, filter尺寸为 $f \times f$, 则卷积后的图片尺寸为 $(n+2p-f+1) \times (n+2p-f+1)$ 。若要保证卷积前后图片尺寸不变, 则 p 应满足:

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

没有padding操作, $p=0$, 称之为“Valid convolutions”;

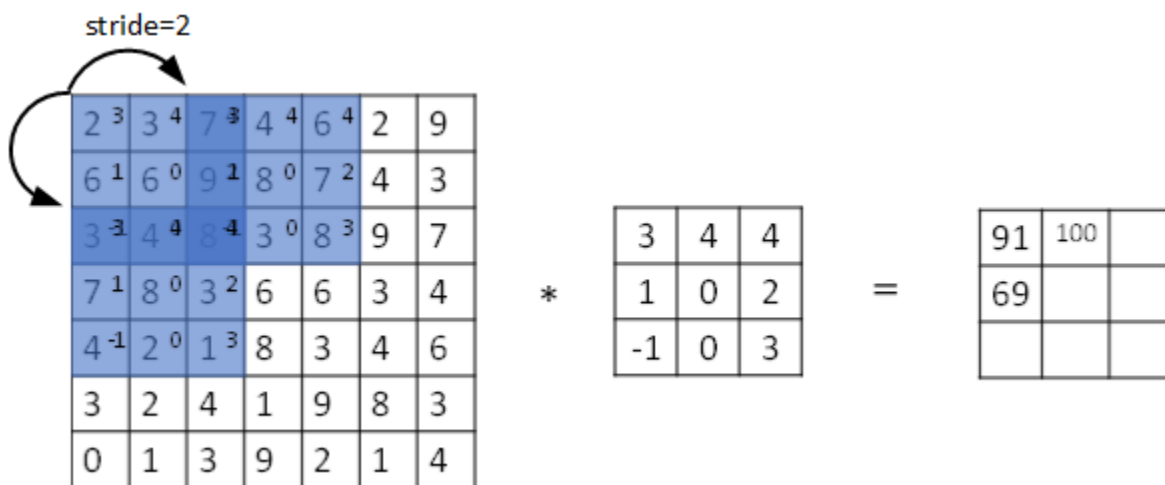
有padding操作, $p = \frac{f-1}{2}$, 称之为“Same convolutions”。



二. 卷积基础操作

卷积步长 (Strided Convolutions)

步长 (Stride) 表示filter在原图片中水平方向和垂直方向每次的步进长度。之前默认stride=1。若stride=2，则表示filter每次步进长度为2，即隔一点移动一次。



二. 卷积基础操作

卷积步长 (Strided Convolutions)

用s表示stride长度，p表示padding长度，如果原始图片尺寸为n x n，filter尺寸为f x f，则卷积后的图片尺寸为：

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

上式中， $\lfloor \dots \rfloor$ 表示向下取整。



二. 卷积基础操作

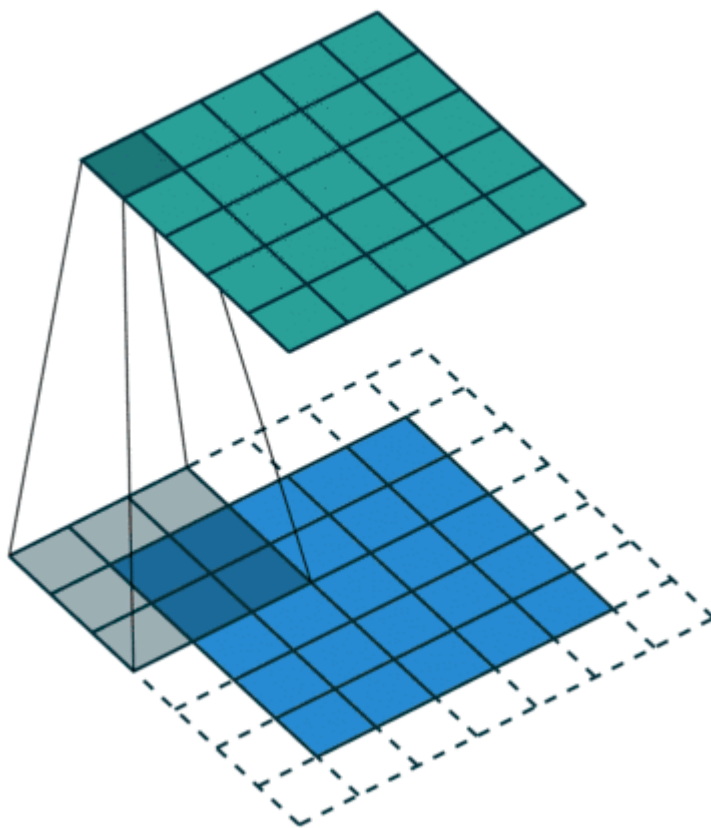
卷积的过程

卷积过程的一个动态图：

$f = 3$

$p = 3$

$s = 1$

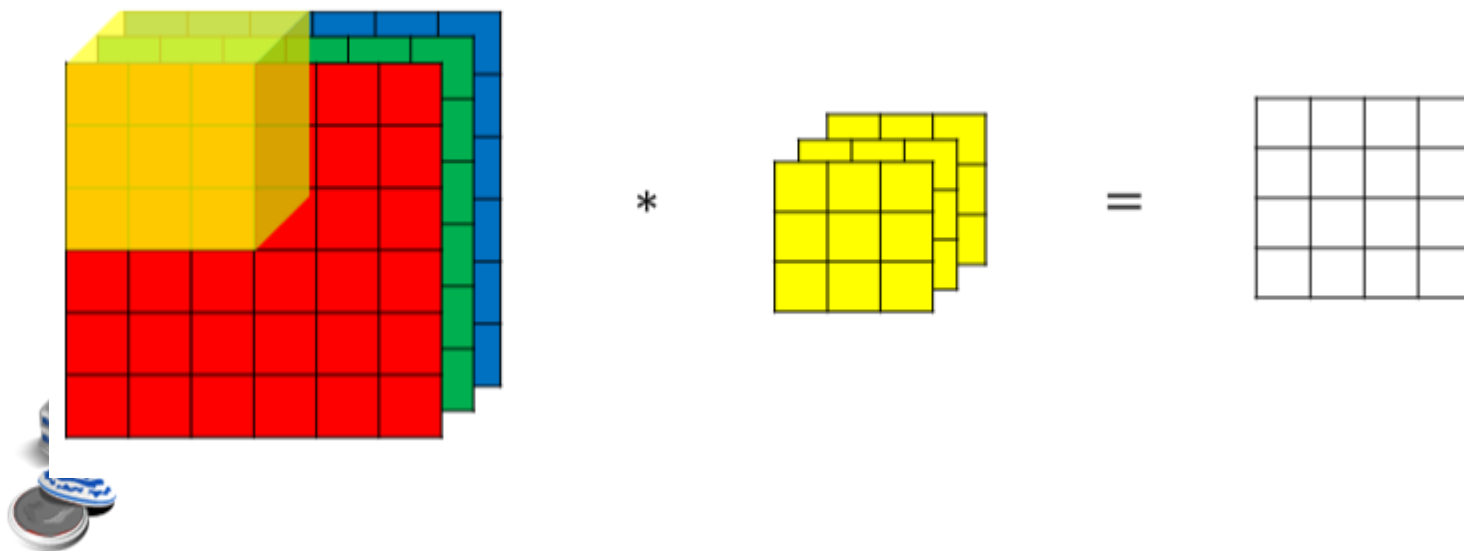


二. 卷积基础操作

filters卷积

(1) 对于3通道的RGB图片，其对应的滤波器算子同样也是3通道的。例如一个图片是 $6 \times 6 \times 3$ ，分别表示图片的高度（height）、宽度（weight）和通道（channel）。

(2) 3通道图片的卷积运算与单通道图片的卷积运算基本一致。过程是将每个单通道（R，G，B）与对应的filter进行卷积运算求和，然后再将3通道的和相加，得到输出图片的一个像素值。

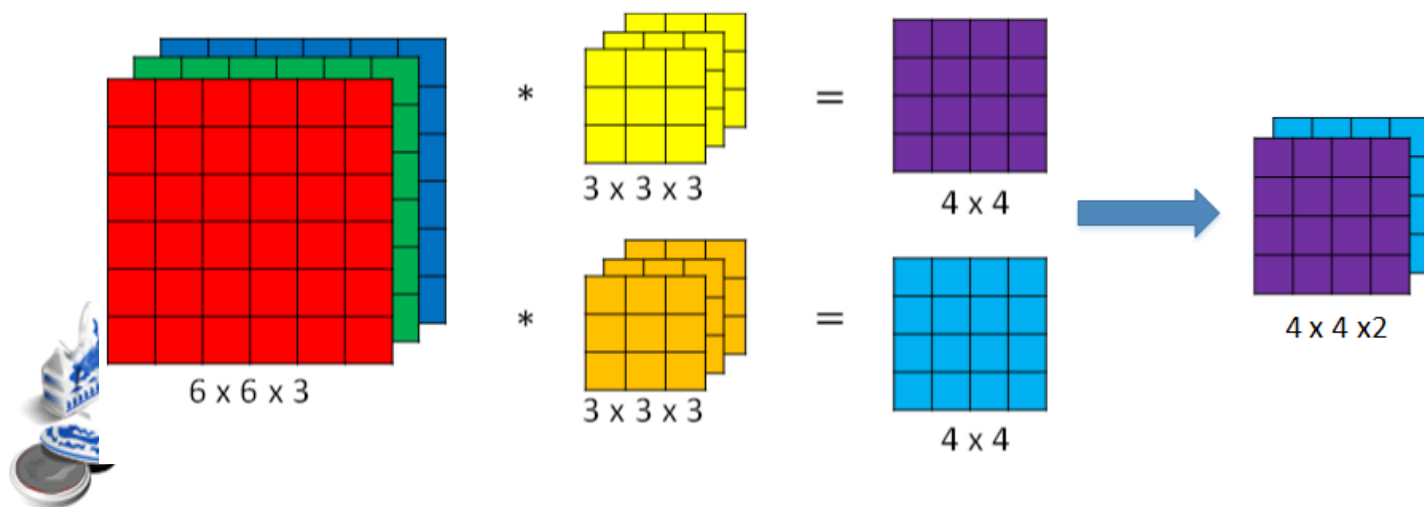


二. 卷积基础操作

filters卷积

(1) 为了进行多个卷积运算，实现更多边缘检测，可以增加更多的滤波器组。例如设置第一个滤波器组实现垂直边缘检测，第二个滤波器组实现水平边缘检测。这样，不同滤波器组卷积得到不同的输出，个数由滤波器组决定。

(2) 若输入图片的尺寸为 $n \times n \times n_c$ ，filter尺寸为 $f \times f \times n_c$ ，则卷积后的图片尺寸为 $(n-f+1) \times (n-f+1) \times n'_c$ 。其中， n_c 为图片通道数目， n'_c 为滤波器组个数。



三. 简单的卷积网络

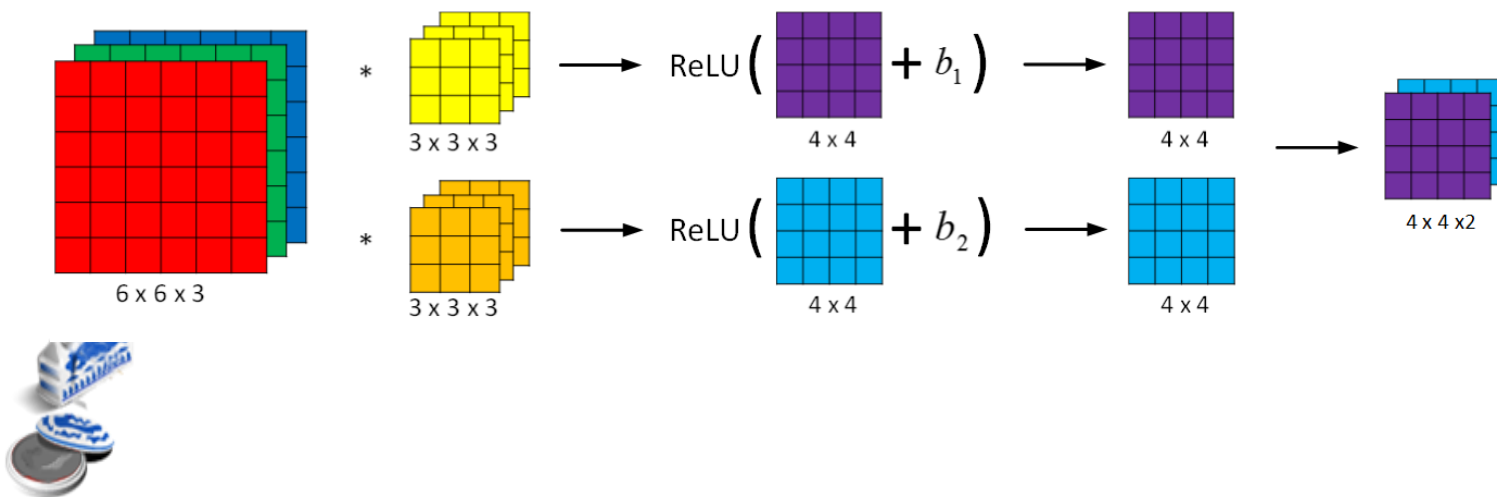
CNN的单层结构

相比之前的卷积过程，CNN的单层结构多了激活函数ReLU和偏移量**b**。整个过程与标准的神经网络单层结构非常类似：

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

卷积运算对应着上式中的乘积运算，滤波器组数值对应着权重 $W^{[l]}$ ，所选的激活函数为ReLU。



三. 简单的卷积网络

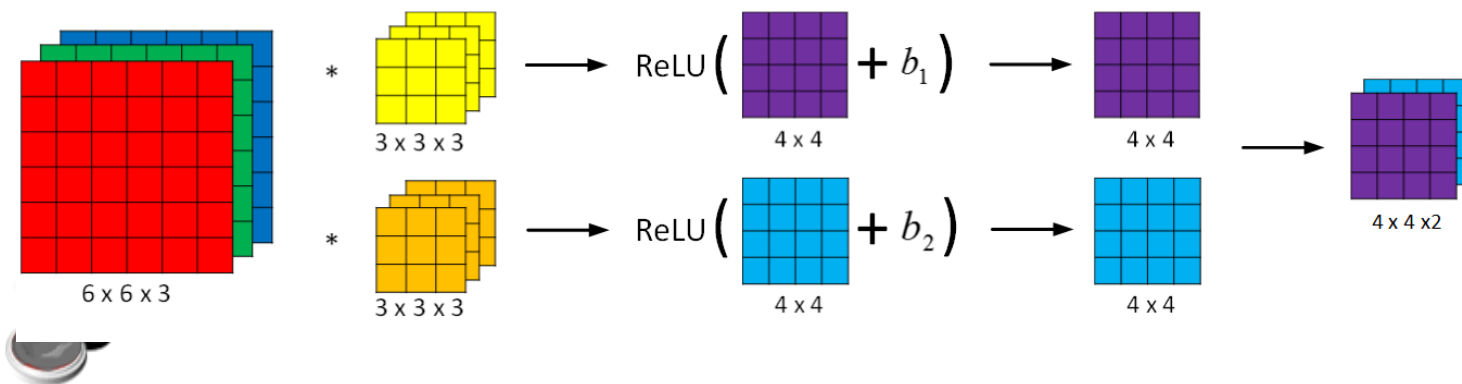
CNN的单层结构参数

计算下图中参数的数目：

(1) 每个滤波器组有 $3 \times 3 \times 3 = 27$ 个参数，还有1个偏移量 b ，则每个滤波器组有 $27 + 1 = 28$ 个参数，两个滤波器组总共包含 $28 \times 2 = 56$ 个参数。

(2) 选定滤波器组后，**参数数目与输入图片尺寸无关**。所以，就不存在由于图片尺寸过大，造成参数过多的情况。

例如一张 $1000 \times 1000 \times 3$ 的图片，标准神经网络输入层的维度将达到3百万，而在CNN中，参数数目只由滤波器组决定，数目相对来说要少得多，这是CNN的优势之一。



三. 简单的卷积网络

CNN单层结构的标记符号

总结一下CNN单层结构的所有标记符号，设层数为 l 。

- $f^{[l]} = \text{filter size}$
- $p^{[l]} = \text{padding}$
- $s^{[l]} = \text{stride}$
- $n_c^{[l]} = \text{number of filters}$

输入维度为： $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$

每个滤波器组维度为： $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

权重维度为： $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

偏置维度为： $1 \times 1 \times 1 \times n_c^{[l]}$

输出维度为： $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

其中：

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

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

如果有 m 个样本，进行向量化运算，相应的输出维度为：

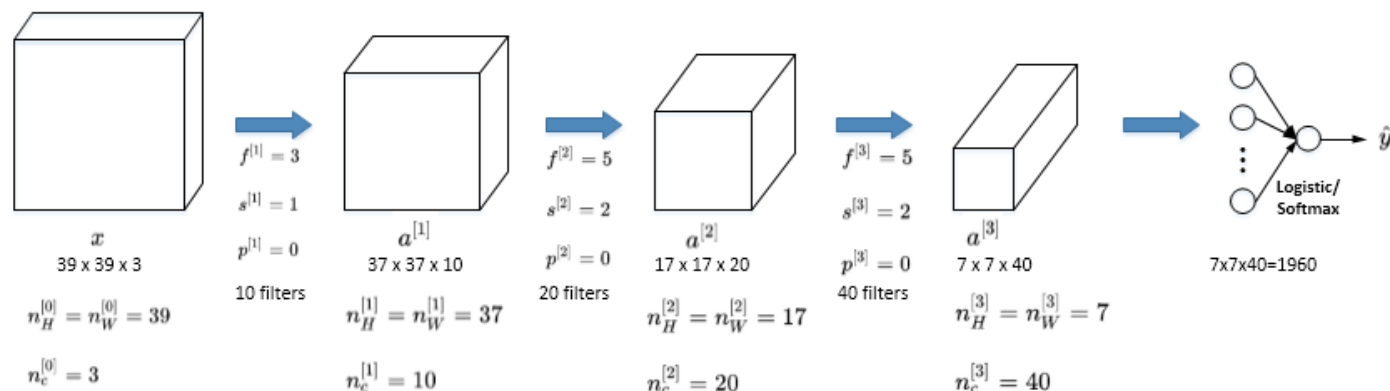
$$m \times n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$$



三. 简单的卷积网络

简单的CNN示例

下面介绍一个简单的CNN网络模型：



该CNN模型各层结构如上图所示。需要注意的是， $a^{[3]}$ 的维度是 $7 \times 7 \times 40$ ，将 $a^{[3]}$ 排列成1列，维度为 1960×1 ，然后连接最后一级输出层。输出层可以是一个神经元，即二元分类（logistic）；也可以是多个神经元，即多元分类（softmax）。最后得到预测输出 \hat{y} 。

值得一提的是，随着CNN层数增加， $n_H^{[l]}$ 和 $n_W^{[l]}$ 一般逐渐减小，而 $n_c^{[l]}$ 一般逐渐增大。



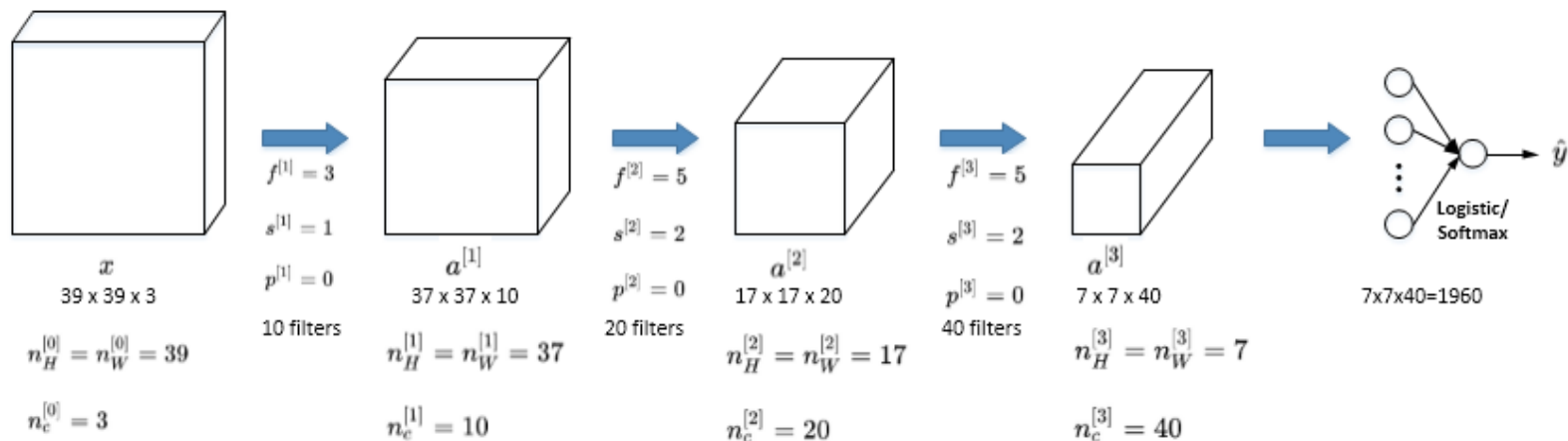
三. 简单的卷积网络

CNN layer的类型

CNN有三种类型的layer:

- (1) Convolution层 (CONV)
- (2) Pooling层 (POOL)
- (3) Fully connected层 (FC)

CONV最为常见也最重要，关于POOL和FC之后再介绍。

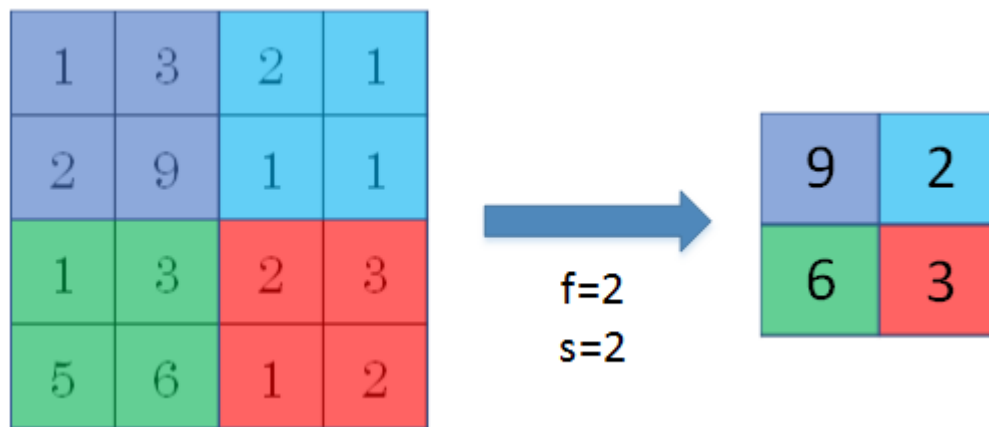


四. 池化层

池化层（Pooling Layers）

池化层（Pooling Layers）是CNN中用来减小尺寸，提高运算速度的，同样能减小noise影响，让各特征更具有健壮性。

Pooling layers的做法比convolution layers简单许多，没有卷积运算，仅仅是在滤波器算子滑动区域内取最大值，即max pooling，这是最常用的做法。注意，超参数p（填充）很少在pooling layers中使用。



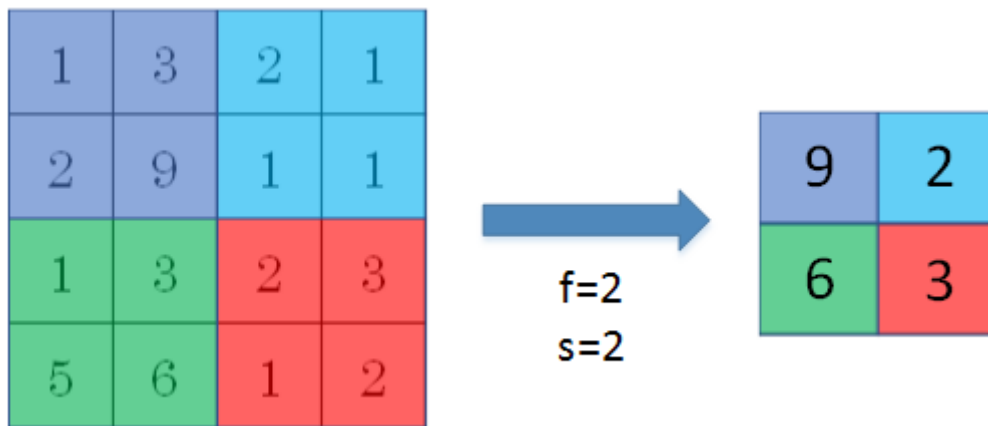
Max Pooling



四. 池化层

池化层的优点

- (1) Max pooling的好处是只保留区域内的最大值（特征），忽略其它值，降低noise影响，提高模型健壮性。
- (2) Max pooling需要的超参数仅为滤波器尺寸 f 和滤波器步进长度 s ，没有其他参数需要模型训练得到，计算量很小。
- (3) 如果是多个通道，那么就每个通道单独进行max pooling操作。



Max Pooling

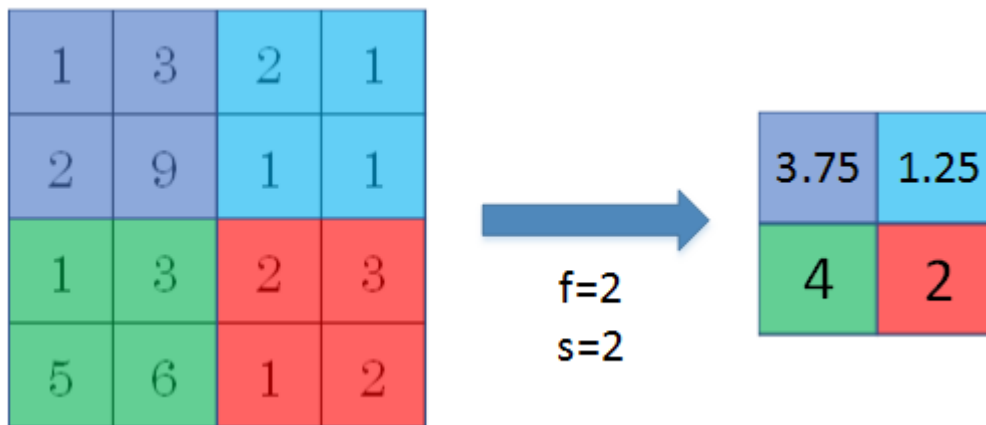


四. 池化层

均值池化层

除了max pooling之外，还有一种做法：**average pooling**。顾名思义，**average pooling**就是在滤波器算子滑动区域计算平均值。

实际应用中，**max pooling**比**average pooling**更为常用。



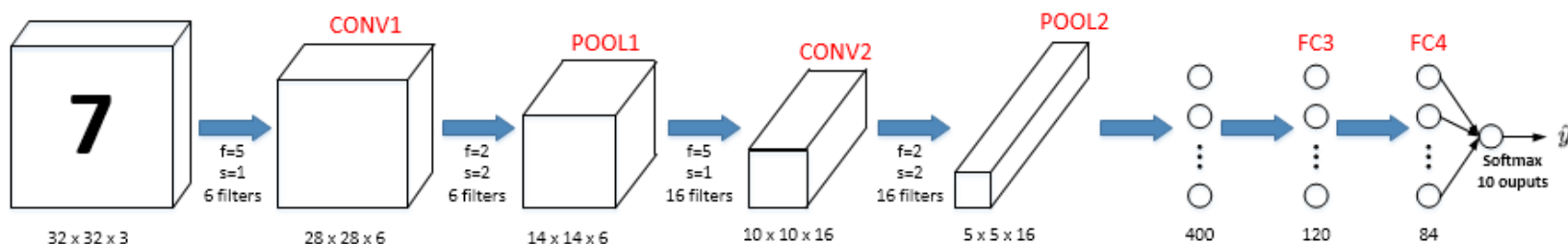
Average Pooling



五. 卷积网络实例

常用的CNN结构的例子

下面介绍一个简单的数字识别的CNN例子：



图中，CON层后面紧接一个POOL层，CONV1和POOL1构成**第一层**，CONV2和POOL2构成**第二层**。特别注意的是FC3和FC4为**全连接层FC**，它跟标准的神经网络结构一致。最后的输出层（softmax）由10个神经元构成。



五. 卷积网络实例

网络各层的尺寸和参数

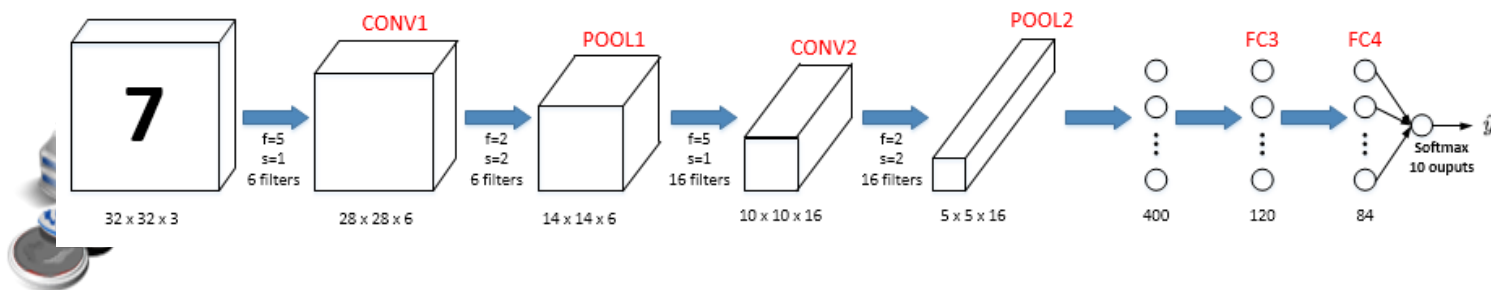
整个CNN各层的尺寸和参数如下表格所示：

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3072	0
CONV1(f=5,s=1)	(28,28,6)	4704	156
POOL1	(14,14,6)	1176	0
CONV2(f=5,s=1)	(10,10,16)	1600	416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48120
FC4	(84,1)	84	10164
Softmax	(10,1)	10	850

32*32*3

5*5*6+6

84*120+84



五. 卷积网络实例

为什么使用卷积？

相比标准神经网络，CNN的优势之一就是参数数目要少得多。参数数目少的原因有两个：

（1）**参数共享**：一个特征检测器（例如垂直边缘检测）对图片某块区域有用，同时也可能作用在图片其它区域。

（2）**连接的稀疏性**：因为滤波器算子尺寸限制，每一层的每个输出只与输入部分区域内有关。

除此之外，由于CNN参数数目较小，所需的训练样本就相对较少，从而一定程度上**不容易发生过拟合现象**。而且，CNN比较擅长捕捉区域位置偏移。也就是说CNN进行物体检测时，不太受物体所处图片位置的影响，增加检测的准确性和系统的健壮性。



总结

- 一. 简述计算机视觉的任务；
- 二. 从边缘检测开始，检测**filter**如何得到经过卷积得到图像的信息。进一步简述卷积的基础操作，例如填充、卷积步长等。然后简述了多个**filters**如何进行卷积；
- 三. 简述了简单的卷积网络，卷积网络的符号表示，介绍网络层的类型；
- 四. 介绍池化层的原理，简述其带来的好处；
- 五. 介绍了一个卷积网络实例，总结它的网络参数，以及为什么需要进行卷积。



谢谢聆听



参考网页:

CNN基础: https://blog.csdn.net/red_stone1/article/details/78651063

