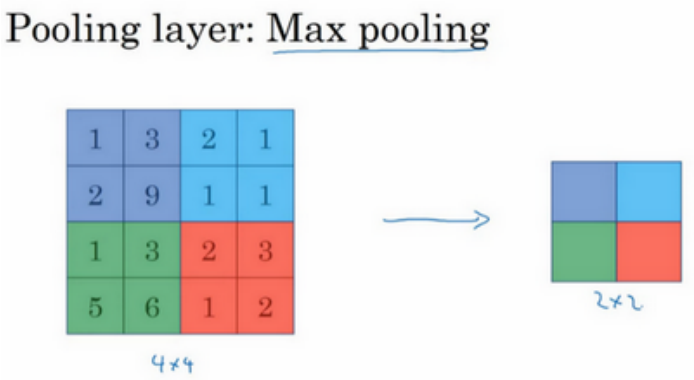
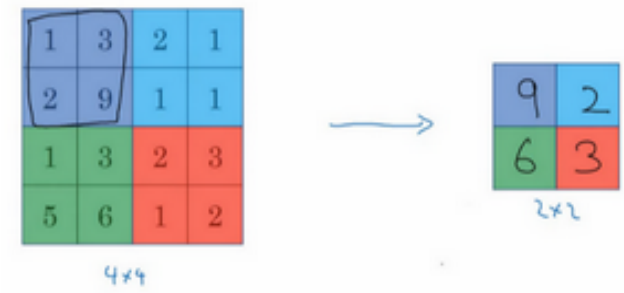
1.9 池化层（Pooling layers）

除了卷积层，卷积网络也经常使用池化层来缩减模型的大小，提高计算速度，同时提高所提取特征的鲁棒性，我们来看一下。

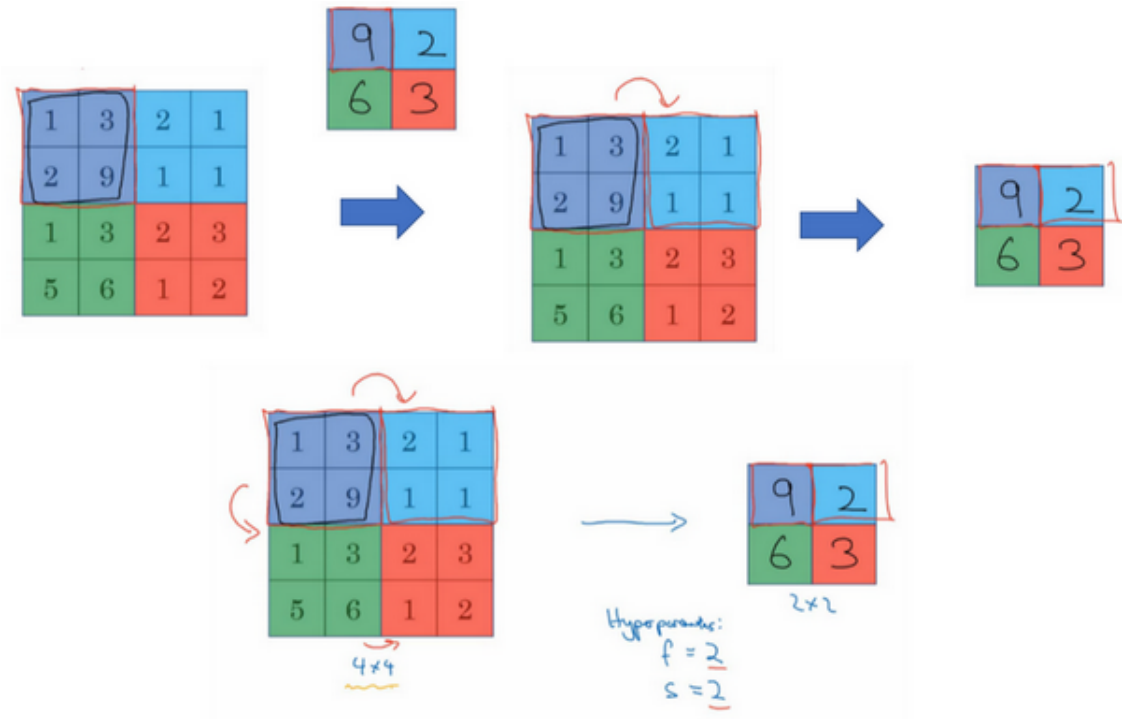


先举一个池化层的例子，然后我们再讨论池化层的必要性。假如输入是一个 4×4 矩阵，用到的池化类型是**最大池化（max pooling）**。执行最大池化的树池是一个 2×2 矩阵。执行过程非常简单，把 4×4 的输入拆分成不同的区域，我把这个区域用不同颜色来标记。对于 2×2 的输出，输出的每个元素都是其对应颜色区域中的最大元素值。



左上区域的最大值是 9，右上区域的最大元素值是 2，左下区域的最大值是 6，右下区域的最大值是 3。为了计算出右侧这 4 个元素值，我们需要对输入矩阵的 2×2 区域做最大值运算。**这就像是应用了一个规模为 2 的过滤器，因为我们选用的是 2×2 区域，步幅是 2，这些就是最大池化的超参数。**

因为我们使用的过滤器为 2×2，最后输出是 9。然后向右移动 2 个步幅，计算出最大值 2。然后是第二行，向下移动 2 步得到最大值 6。最后向右移动 3 步，得到最大值 3。这是一个 2×2 矩阵，即$f = 2$，步幅是 2，即$s = 2$。



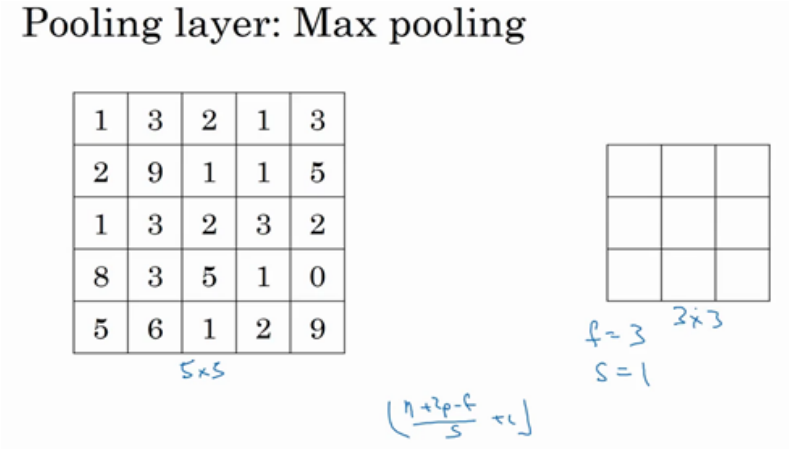
这是对最大池化功能的直观理解，你可以把这个 4×4 输入看作是某些特征的集合，也许不是。你可以把这个 4×4 区域看作是某些特征的集合，也就是神经网络中某一层的非激活值集合。**数字大意味着可能探测到了某些特定的特征，左上象限具有的特征可能是一个垂直边缘，一只眼睛，或是大家害怕遇到的 CAP 特征。**显然左上象限中存在这个特征，这个特征可能是一只猫眼探测器。然而，右上象限并不存在这个特征。最大化操作的功能就是只要在任何一个象限内提取到某个特征，它都会保留在最大化的池化输出里。**所以最大化运算的实际作用就是，如果在过滤器中提取到某个特征，那么保留其最大值。如果没有提取到这个特征，可能在右上象限中不存在这个特征，那么其中的最大值也还是很小，这就是最大池化的直观理解。**

必须承认，人们使用最大池化的主要原因是此方法在很多实验中效果都很好。尽管刚刚描述的直观理解经常被引用，不知大家是否完全理解它的真正原因，不知大家是否理解最大池化效率很高的真正原因。

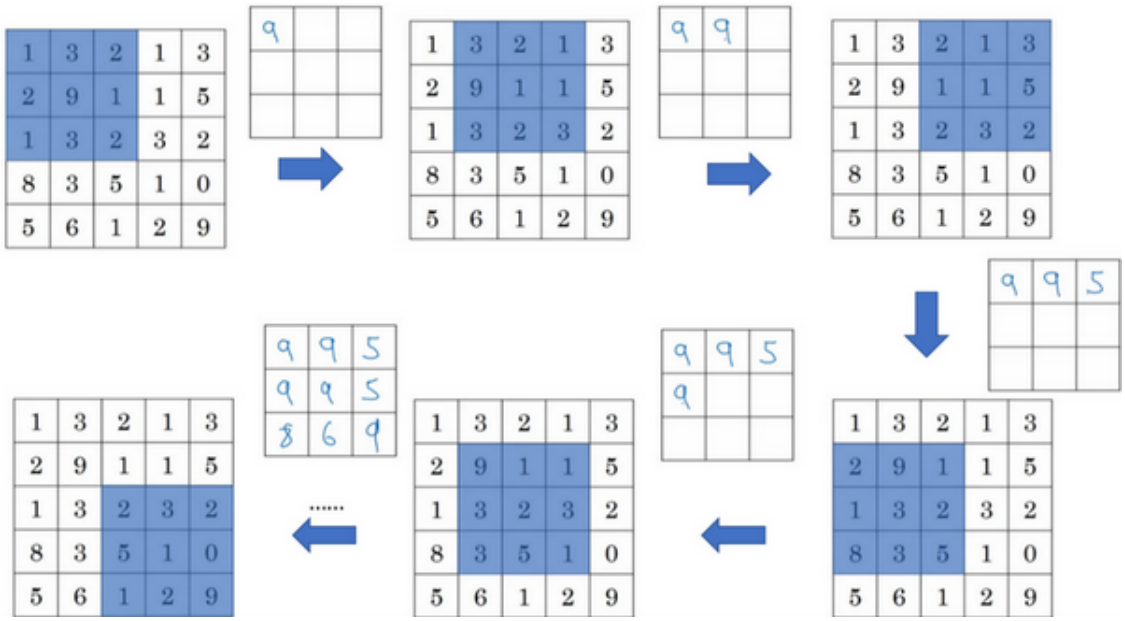
其中一个有意思的特点就是，它有一组超参数，但并没有参数需要学习。实际上，梯度下降没有什么可学的，一旦确定了f和s，它就是一个固定运算，梯度下降无需改变任何值。

我们来看一个有若干个超级参数的示例，输入是一个 5×5 的矩阵。我们采用最大池化法，它的过滤器参数为 3×3，即$f=3$，步幅为 1，$s=1$，输出矩阵是 3×3。之前讲的计算卷积层输出大小的公式同样适用于最大池化，即$\frac{n+2p-f}{s}+1$，这个公式也可以计算最大池化

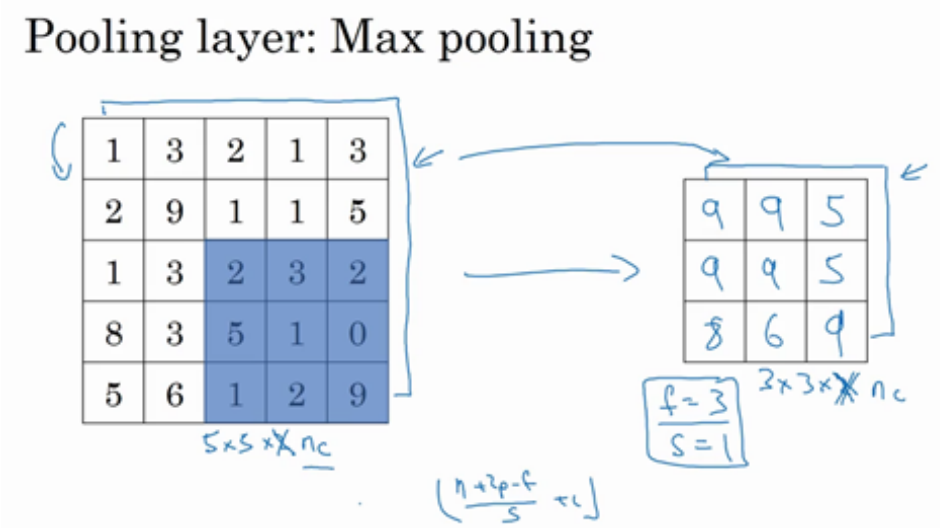
的输出大小。



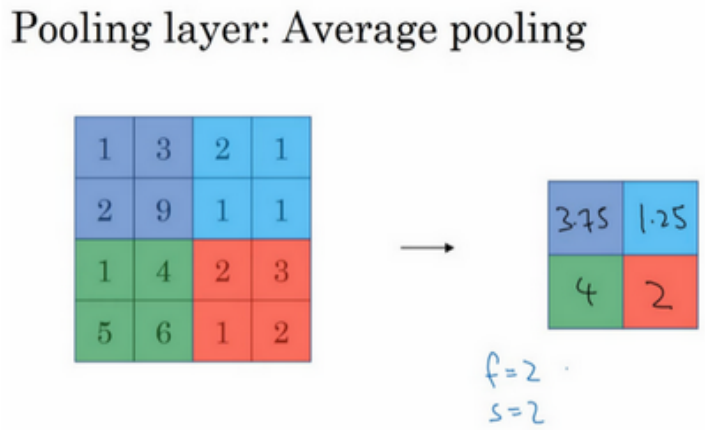
此例是计算 3×3 输出的每个元素，我们看左上角这些元素，注意这是一个 3×3 区域，因为有 3 个过滤器，取最大值 9。然后移动一个元素，因为步幅是 1，蓝色区域的最大值是 9. 继续向右移动，蓝色区域的最大值是 5。然后移到下一行，因为步幅是 1，我们只向下移动一个格，所以该区域的最大值是 9。这个区域也是 9。这两个区域的最大值都是 5。最后这三个区域的最大值分别为 8，6 和 9。超参数$f = 3$，$s = 1$，最终输出如图所示。



以上就是一个二维输入的最大池化的演示，如果输入是三维的，那么输出也是三维的。例如，输入是 5×5×2，那么输出是 3×3×2。计算最大池化的方法就是分别对每个通道执行刚刚的计算过程。如上图所示，第一个通道依然保持不变。对于第二个通道，我刚才画在下面的，在这个层做同样的计算，得到第二个通道的输出。一般来说，如果输入是 5×5×n_c，输出就是 3×3×n_c，n_c个通道中每个通道都单独执行最大池化计算，以上就是最大池化算法。



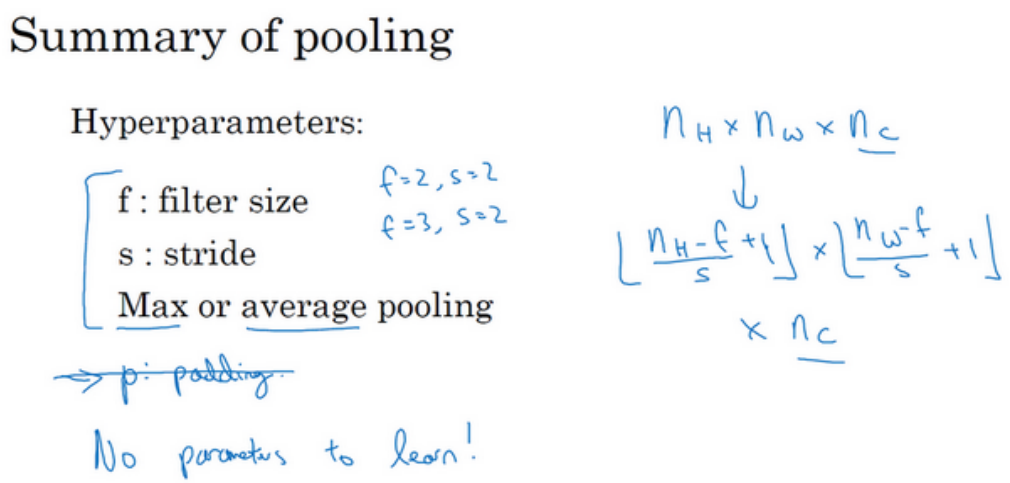
另外还有一种类型的池化，**平均池化**，它不太常用。我简单介绍一下，这种运算顾名思义，选取的不是每个过滤器的最大值，而是平均值。示例中，紫色区域的平均值是 3.75，后面依次是 1.25、4 和 2。这个平均池化的超级参数$f=2$，$s=2$，我们也可以选择其它超级参数。



目前来说，最大池化比平均池化更常用。但也有例外，就是深度很深的神经网络，你可以用平均池化来分解规模为 7×7×1000 的网络的表示层，在整个空间内求平均值，得到 1×1×1000，一会我们看个例子。但在神经网络中，最大池化要比平均池化用得更多。

总结一下，池化的超级参数包括过滤器大小f和步幅s，常用的参数值为$f=2$，$s=2$，应用频率非常高，其效果相当于高度和宽度缩减一半。也有使用$f=3$，$s=2$的情况。至于其它超级参数就要看你用的是最大池化还是平均池化了。你也可以根据自己意愿增加表示 **padding** 的其他超级参数，虽然很少这么用。最大池化时，往往很少用到超参数 **padding**，当然也有例外的情况，我们下周会讲。大部分情况下，最大池化很少用 **padding**。目前p最常用的值是 0，即$p=0$。最大池化的输入就是$n_H \times n_W \times n_c$，假设没有 **padding**，则输出$\lfloor \frac{n_H-f}{s} +$

$1] \times \lfloor \frac{n_w - f}{s} + 1 \rfloor \times n_c$。输入通道与输出通道个数相同，因为我们对每个通道都做了池化。需要注意的一点是，池化过程中没有需要学习的参数。执行反向传播时，反向传播没有参数适用于最大池化。只有这些设置过的超参数，可能是手动设置的，也可能是通过交叉验证设置的。



除了这些，池化的内容就全部讲完了。最大池化只是计算神经网络某一层的静态属性，没有什么需要学习的，它只是一个静态属性。

关于池化我们就讲到这儿，现在我们已经知道如何构建卷积层和池化层了。下节课，我们会分析一个更复杂的可以引进全连接层的卷积网络示例。