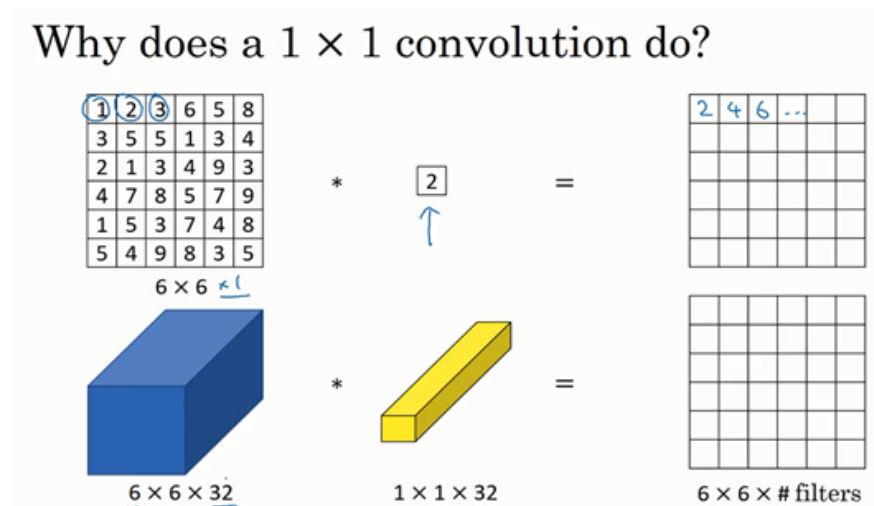


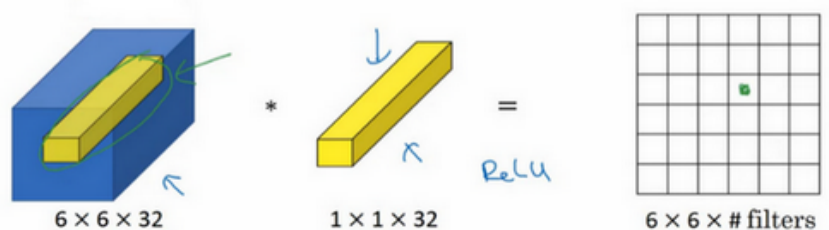
## 2.5 网络中的网络以及 $1 \times 1$ 卷积 (Network in Network and $1 \times 1$ convolutions)

在架构内容设计方面，其中一个比较有帮助的想法是使用  $1 \times 1$  卷积。也许你会好奇， $1 \times 1$  的卷积能做什么呢？不就是乘以数字么？听上去挺好笑的，结果并非如此，我们来具体看看。

过滤器为  $1 \times 1$ ，这里是数字 2，输入一张  $6 \times 6 \times 1$  的图片，然后对它做卷积，起过滤器大小为  $1 \times 1 \times 1$ ，结果相当于把这个图片乘以数字 2，所以前三个单元格分别是 2、4、6 等等。  
用  $1 \times 1$  的过滤器进行卷积，似乎用处不大，只是对输入矩阵乘以某个数字。但这仅仅是对于  $6 \times 6 \times 1$  的一个通道图片来说， $1 \times 1$  卷积效果不佳。



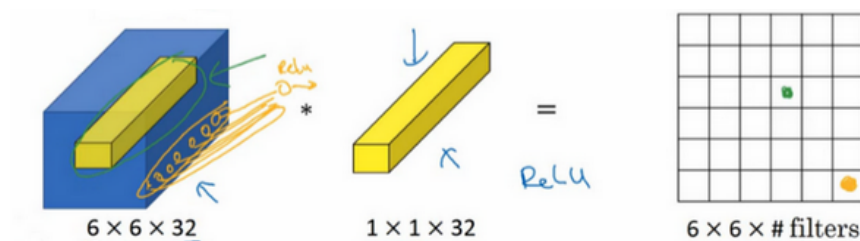
如果是一张  $6 \times 6 \times 32$  的图片，那么使用  $1 \times 1$  过滤器进行卷积效果更好。具体来说， $1 \times 1$  卷积所实现的功能是遍历这 36 个单元格，计算左图中 32 个数字和过滤器中 32 个数字的元素积之和，然后应用 ReLU 非线性函数。



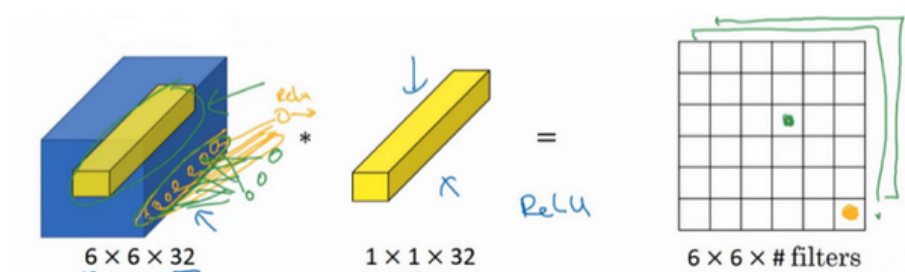
我们以其中一个单元为例，它是这个输入层上的某个切片，用这 36 个数字乘以这个输入层上  $1 \times 1$  切片，得到一个实数，像这样把它画在输出中。

这个  $1 \times 1 \times 32$  过滤器中的 32 个数字可以这样理解，一个神经元的输入是 32 个数字（输

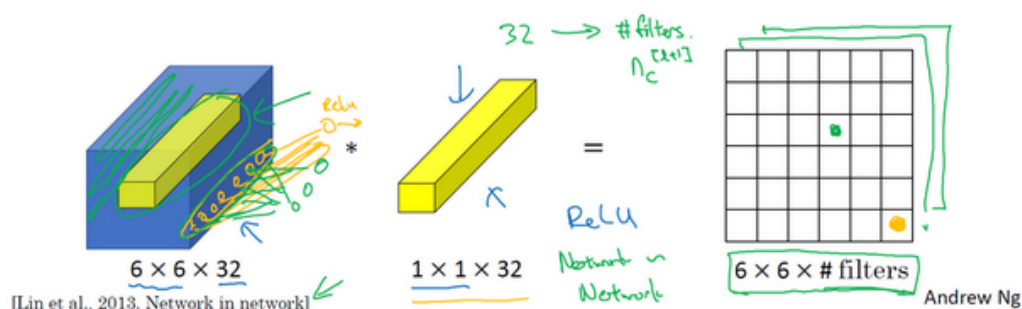
入图片中左下角位置 32 个通道中的数字），即相同高度和宽度上某一切片上的 32 个数字，这 32 个数字具有不同通道，乘以 32 个权重（将过滤器中的 32 个数理解为权重），然后应用 ReLU 非线性函数，在这里输出相应的结果。



一般来说，如果过滤器不止一个，而是多个，就好像有多个输入单元，其输入内容为一个切片上所有数字，输出结果是 6x6 过滤器数量。



所以 1x1 卷积可以从根本上理解为对这 32 个不同的位置都应用一个全连接层，全连接层的作用是输入 32 个数字（过滤器数量标记为  $n_c^{[l+1]}$ ，在这 36 个单元上重复此过程），输出结果是 6x6x#filters（过滤器数量），以便在输入层上实施一个非平凡（non-trivial）计算。

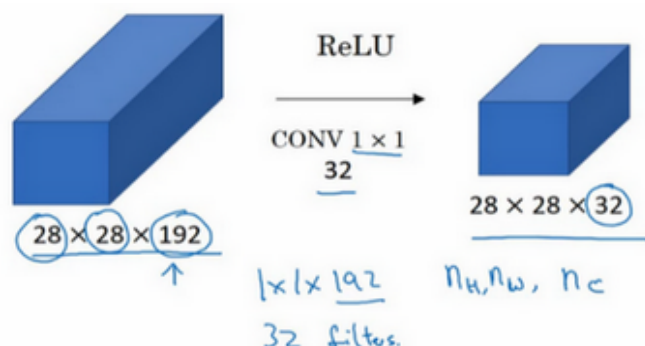


这种方法通常称为 1x1 卷积，有时也被称为 **Network in Network**，在林敏、陈强和颜水成的论文中有详细描述。虽然论文中关于架构的详细内容并没有得到广泛应用，但是 1x1 卷积或 **Network in Network** 这种理念却很有影响力，很多神经网络架构都受到它的影响，包括下节课要讲的 **Inception** 网络。

举个 1x1 卷积的例子，相信对大家有所帮助，这是它的一个应用。

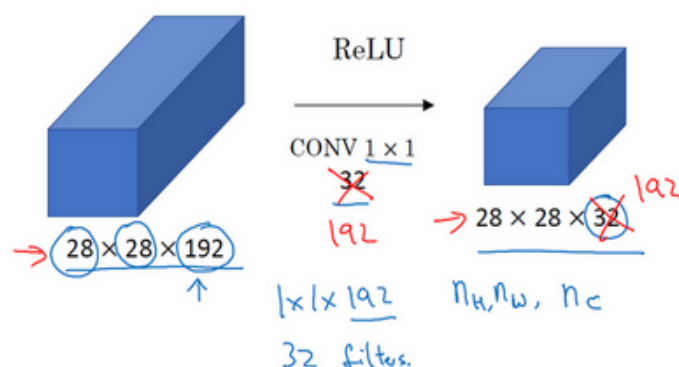
假设这是一个 28x28x192 的输入层，你可以使用池化层压缩它的高度和宽度，这个过程我们很清楚。但如果通道数量很大，该如何把它压缩为 28x28x32 维度的层呢？你可以用 32

个大小为  $1 \times 1$  的过滤器，严格来讲每个过滤器大小都是  $1 \times 1 \times 192$  维，因为过滤器中通道数量必须与输入层中通道的数量保持一致。但是你使用了 32 个过滤器，输出层为  $28 \times 28 \times 32$ ，这就是压缩通道数 ( $n_c$ ) 的方法，对于池化层我只是压缩了这些层的高度和宽度。



在之后我们看到在某些网络中  $1 \times 1$  卷积是如何压缩通道数量并减少计算的。当然如果你想保持通道数 192 不变，这也是可行的， $1 \times 1$  卷积只是添加了非线性函数，当然也可以让网络学习更复杂的函数，比如，我们再添加一层，其输入为  $28 \times 28 \times 192$ ，输出为  $28 \times 28 \times 192$ 。

## Using $1 \times 1$ convolutions



$1 \times 1$  卷积层就是这样实现了一些重要功能的 (doing something pretty non-trivial)，它给神经网络添加了一个非线性函数，从而减少或保持输入层中的通道数量不变，当然如果你愿意，也可以增加通道数量。后面你会发现这对构建 Inception 网络很有帮助，我们放在下节课讲。

这节课我们演示了如何根据自己的意愿通过  $1 \times 1$  卷积的简单操作来压缩或保持输入层中的通道数量，甚至是增加通道数量。下节课，我们再讲讲  $1 \times 1$  卷积是如何帮助我们构建 Inception 网络的，下节课见。

