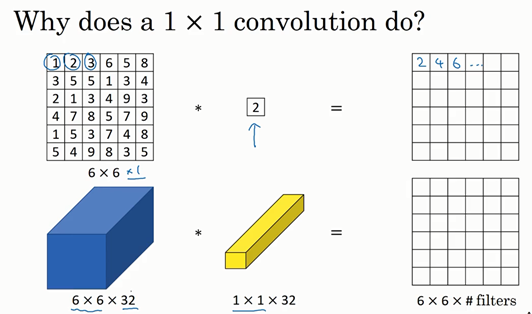
### 2.5 网络中的网络以及 1×1 卷积（Network in Network and 1×1 convolutions）

在架构内容设计方面，其中一个比较有帮助的想法是使用1×1卷积。也许你会好奇，1×1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1×1，这里是数字2，输入一张6×6×1的图片，然后对它做卷积，起过滤器大小为1×1×1，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1×1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6×6×1的一个通道图片来说，1×1卷积效果不佳。



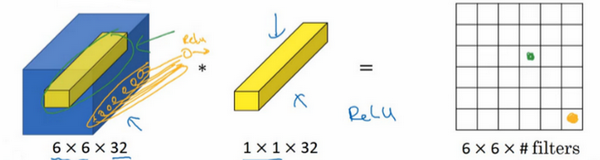
**如果是一张6×6×32的图片，那么使用1×1过滤器进行卷积效果更好。具体来说，1×1卷积所实现的功能是遍历这36个单元格，计算左图中32个数字和过滤器中32个数字的元素积之和，然后应用ReLU非线性函数。**

图片包含 物体, 时钟

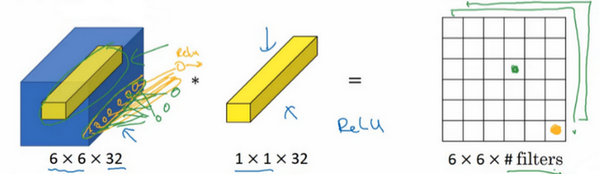
描述已自动生成

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1×1切片，得到一个实数，像这样把它画在输出中。

这个1×1×32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6×6过滤器数量。



**所以1×1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，在这36个单元上重复此过程）,输出结果是6×6×#filters（过滤器数量），以便在输入层上实施一个非平凡（non-trivial）计算。**

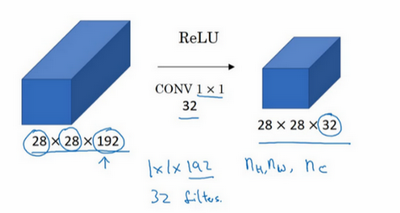
图片包含 文字

描述已自动生成

这种方法通常称为1×1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1×1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1×1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28×28×192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28×28×32维度的层呢？你可以用32个大小为1×1的过滤器，严格来讲每个过滤器大小都是1×1×192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28×28×32，这就是压缩通道数（）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1×1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，1×1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28×28×192，输出为28×28×192。

图片包含 文字

描述已自动生成

**1×1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1×1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1×1卷积是如何帮助我们构建**Inception**网络的，下节课见。