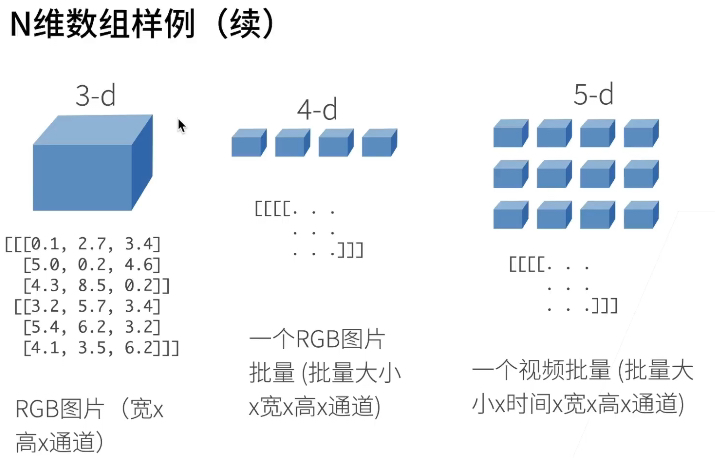
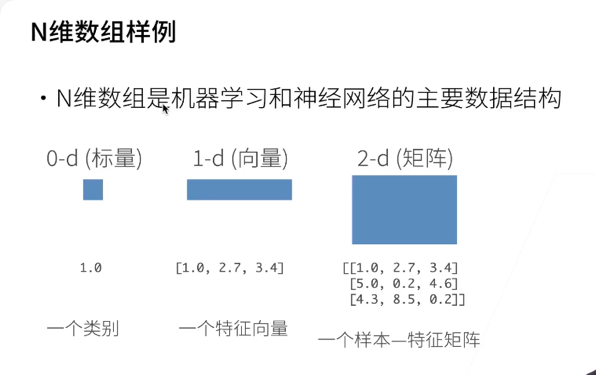
动手学深度学习笔记一

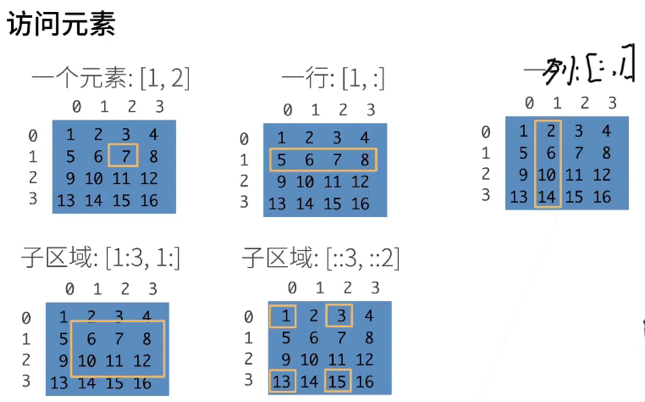
1.1数组（张量）的基本操作

当我们安装好基础的框架后就可以开始进行初步的深度学习了。首先我们要了解的是数组，N维数组是机器学习和神经网络的主要数据结构。



零维的数组是一个标量，基本可以理解为简单的一个数；一维的数组就是一个向量，在空间中有长度有方向；而二维数组就是一个矩阵，包含行列；三维的数组可以类比一张RGB图片的宽、高、通道；四维和五维以此类推。

接下来我们试着来创建数组，创建数组需要的是设置数组的形状（例如i\*j的矩阵）、元素的数据类型（如32位的浮点数）、每个元素的值。



当我们创建了一个数组后，我们就可以访问其中的元素，如上图中去查找一个元素（访问第i行第j列使用[i,j]）、访问一行或一列时使用[i,:]或[:,j]，访问子区域[ix:iy,j:]等。

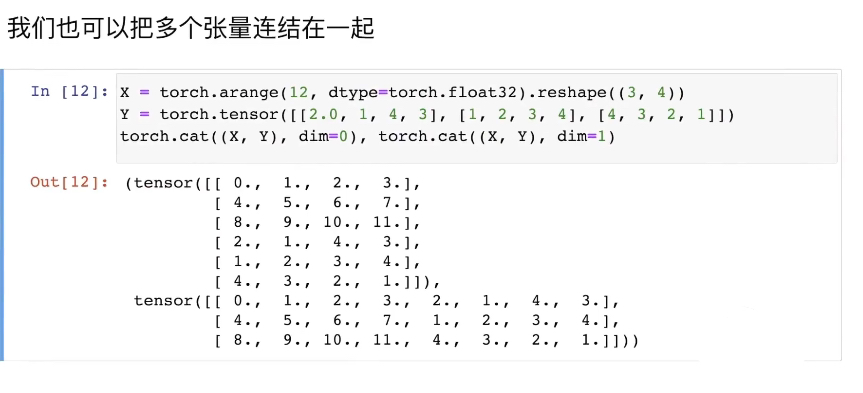
之后我们开始实战，首先引用torch模块，然后使用arange（）函数创建张量，默认是一维的张量，shape（）可以得到张量的形状（包括几维和长度），numel（）得到张量里元素的个数。

我们想改变一个张量的形状，我们可以使用reshape（x，y）函数来操作，其中x代表行，y代表列。zeros（）创建全0张量，ones（）创建全1张量，同样可以自定义形状。

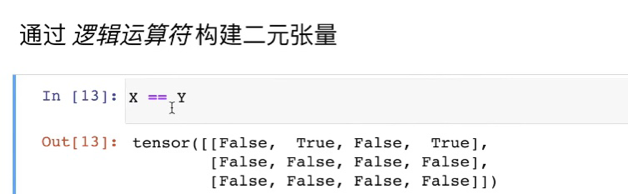
如果我们自定义张量中的元素，那么可以对列表进行嵌套赋值。

既然创建的张量中都是元素，那么也一定可以进行基本运算，+、-、\*、/等。

cat（）可以将两个行或列数相同的张量堆砌起来，需要在后面设定dim的值，dim=0时按行，dim=1时按列（均是在二维的情况下进行的，有三位张量的话可以使用dim=2的操作，更高维度以此类推）。

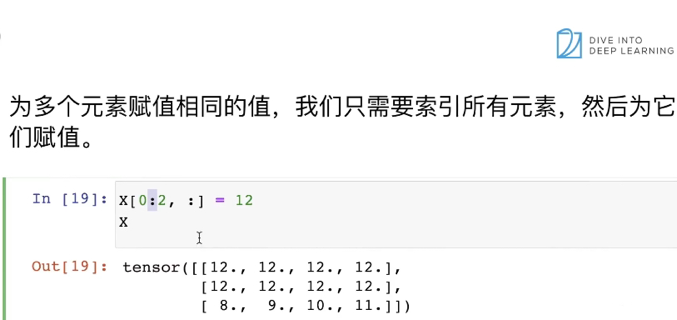
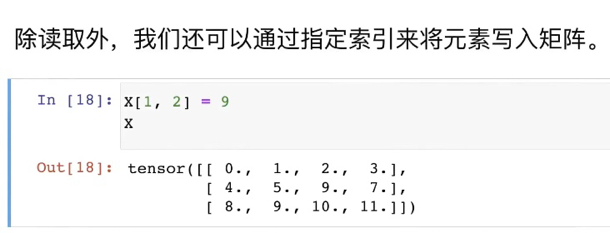
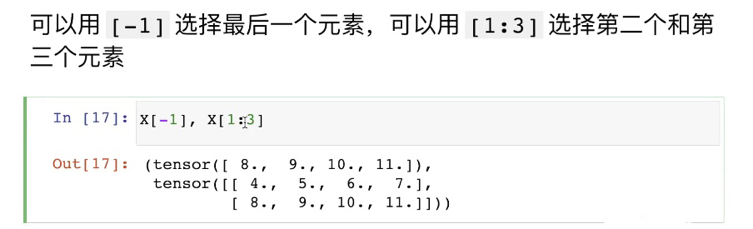


两个维度和形状相同的张量也可以通过逻辑运算符构建新的张量，里面的元素全为bool值。



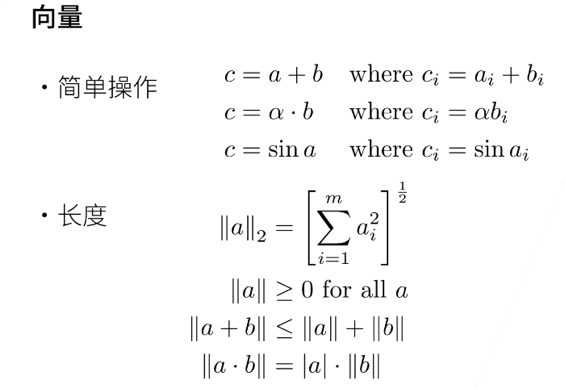
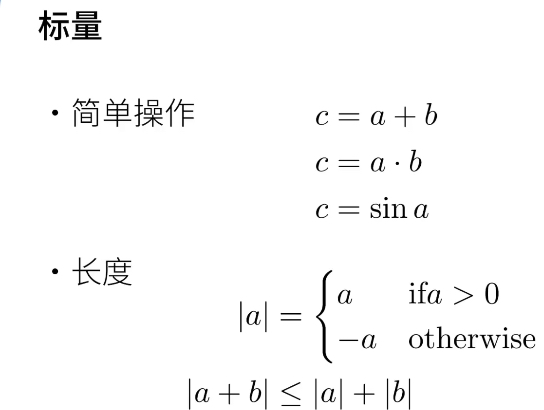
x.sum（）会对张量中的所有元素求和，最终得到一个零维且长度为一的标量。

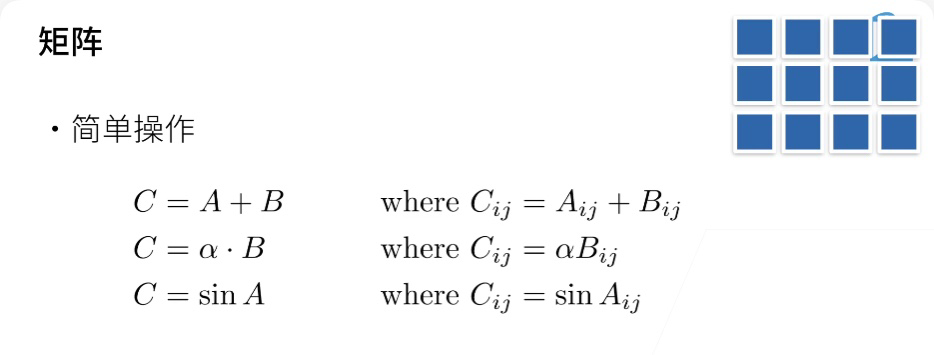
在执行张量运算中，即使形状不同，只要维度相同，那么我们仍然可以通过广播机制（broadcasting mechanism）来执行按元素操作，具体机制的执行命令是（以二维的张量为例）：当我们拥有两个维度不同的张量时，相比较少的那一个维度会自动复制自己以达到与较多的维度相同，例如一个形状为（3，1）的张量和一个形状为（1，2）的张量相加，他们都会自动复制变成形状为（3，2）的张量。



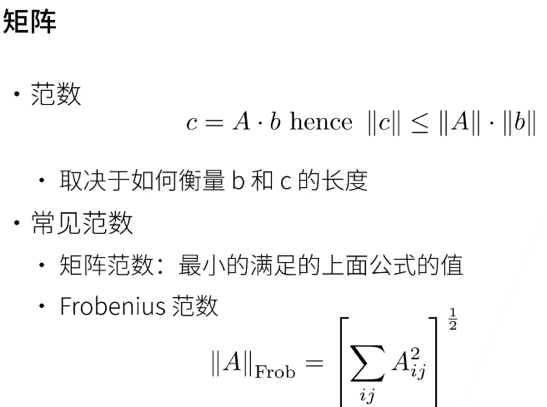
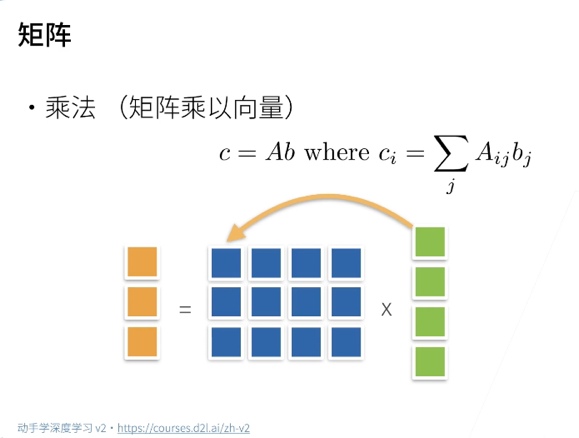
PS：涉及到一些内存的分配问题，以后再细探讨。

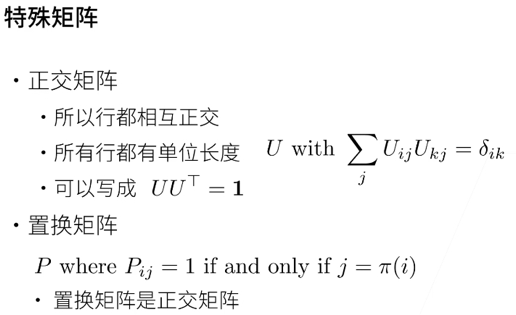
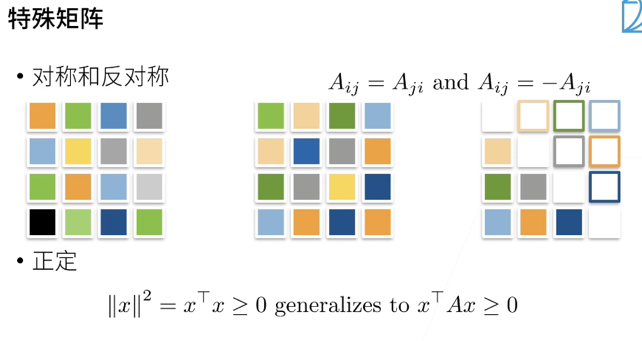
1.2张量的基本运算





以上是一些基本的张量运算，我们主要着重的是矩阵的乘法运算。





我们也会遇到很多特殊的矩阵，这些以后再进行详解。