Numpy教程

NumPy是Python中用于数据分析、机器学习、科学计算的重要软件包。它极大地简化了向量和矩阵的操作及处理。python的不少数据处理软件包依赖于NumPy作为其基础架构的核心部分（例如scikit-learn、SciPy、pandas和tensorflow）。除了数据切片和数据切块的功能之外，掌握numpy也使得开发者在使用各数据处理库调试和处理复杂用例时更具优势。



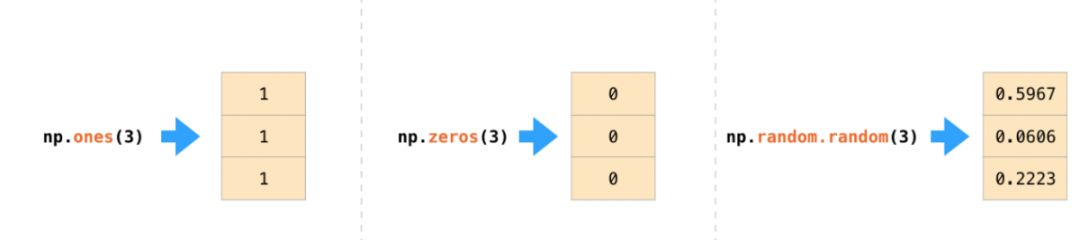
在本文中，将介绍NumPy的主要用法，以及它如何呈现不同类型的数据（表格，图像，文本等），这些经Numpy处理后的数据将成为机器学习模型的输入。

import numpy as np

**NumPy中的数组操作**

**创建数组**

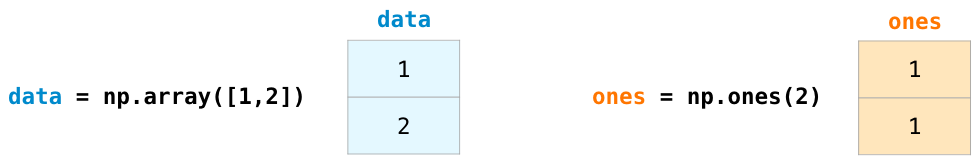
我们可以通过将python列表传入np.array()来创建一个NumPy数组（也就是强大的ndarray）。在下面的例子里，创建出的数组如右边所示，通常情况下，我们希望NumPy为我们初始化数组的值，为此NumPy提供了诸如ones()，zeros()和random.random()之类的方法。我们只需传入元素个数即可：



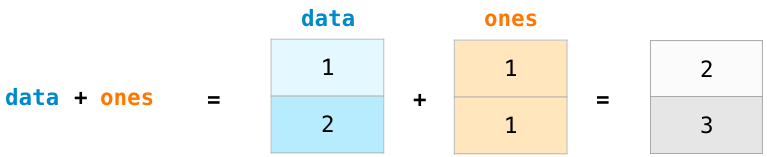
一旦我们创建了数组，我们就可以用其做点有趣的应用了，文摘菌将在下文展开说明。

**数组的算术运算**

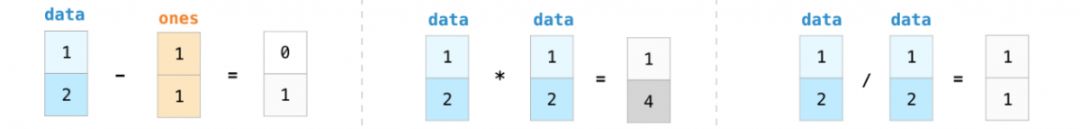
让我们创建两个NumPy数组，分别称作data和ones：



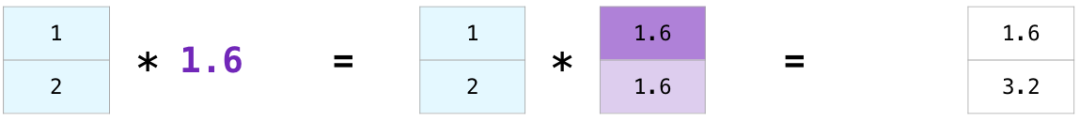
若要计算两个数组的加法，只需简单地敲入data + ones，就可以实现对应位置上的数据相加的操作（即每行数据进行相加），这种操作比循环读取数组的方法代码实现更加简洁。



当然，在此基础上举一反三，也可以实现减法、乘法和除法等操作：



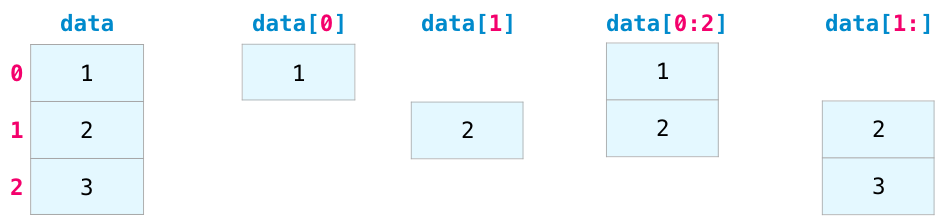
许多情况下，我们希望进行数组和单个数值的操作（也称作向量和标量之间的操作）。比如：如果数组表示的是以英里为单位的距离，我们的目标是将其转换为公里数。可以简单的写作data \* 1.6：



NumPy通过数组广播（broadcasting）知道这种操作需要和数组的每个元素相乘。

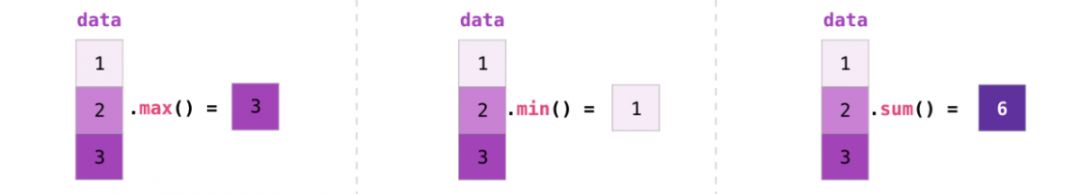
**数组的切片操作**

我们可以像python列表操作那样对NumPy数组进行索引和切片，如下图所示：



**聚合函数**

NumPy为我们带来的便利还有聚合函数，聚合函数可以将数据进行压缩，统计数组中的一些特征值：



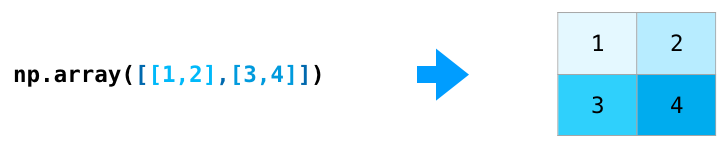
除了min，max和sum等函数，还有mean（均值），prod（数据乘法）计算所有元素的乘积，std（标准差），等等。上面的所有例子都在一个维度上处理向量。除此之外，NumPy之美的一个关键之处是它能够将之前所看到的所有函数应用到任意维度上。

**NumPy中的矩阵操作**

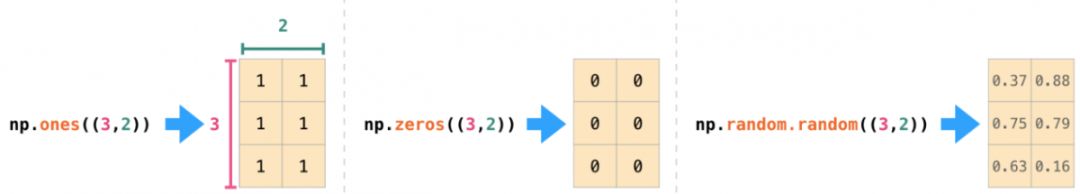
**创建矩阵**

我们可以通过将二维列表传给Numpy来创建矩阵。

np.array([[1,2],[3,4]])

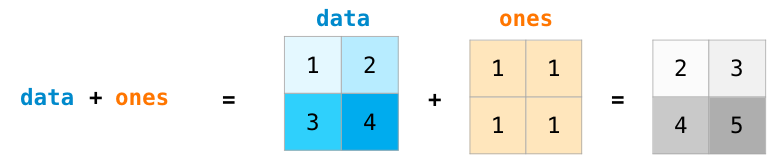


除此外，也可以使用上文提到的ones()、zeros()和random.random()来创建矩阵，只需传入一个元组来描述矩阵的维度：

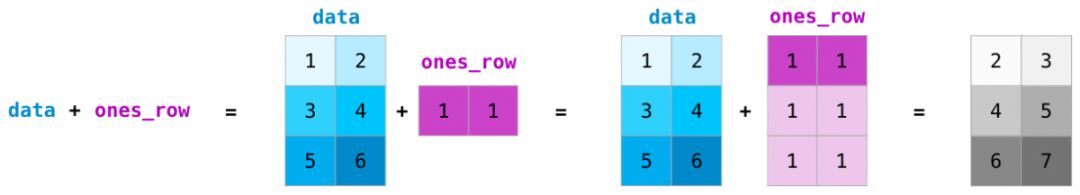


**矩阵的算术运算**

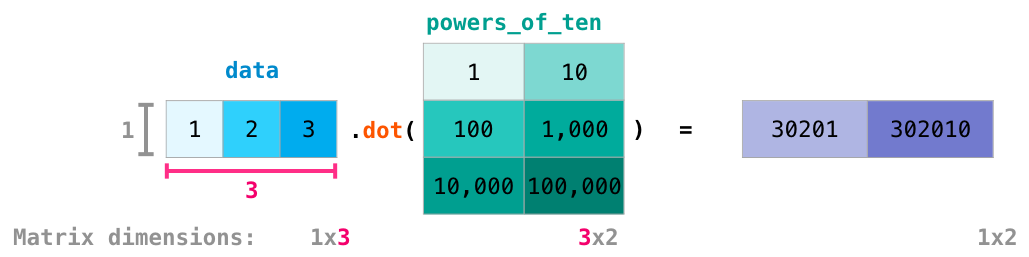
对于大小相同的两个矩阵，我们可以使用算术运算符（+-\*/）将其相加或者相乘。NumPy对这类运算采用对应位置（position-wise）操作处理：



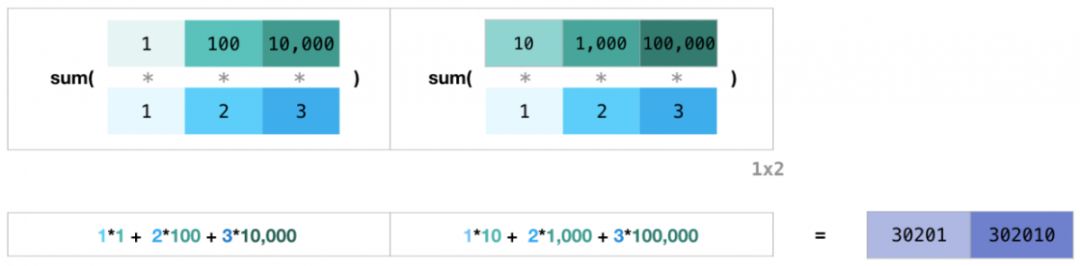
对于不同大小的矩阵，只有两个矩阵的维度同为1时（例如矩阵只有一列或一行），我们才能进行这些算术运算，在这种情况下，NumPy使用广播规则（broadcast）进行操作处理：



与算术运算有很大区别是使用点积的矩阵乘法。NumPy提供了dot()方法，可用于矩阵之间进行点积运算：

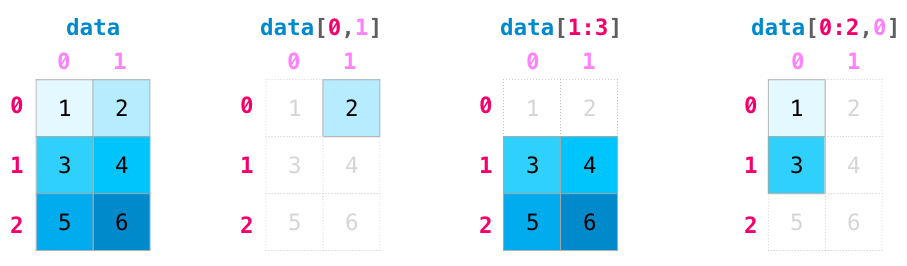


上图的底部添加了矩阵尺寸，以强调运算的两个矩阵在列和行必须相等。可以将此操作图解为如下所示：

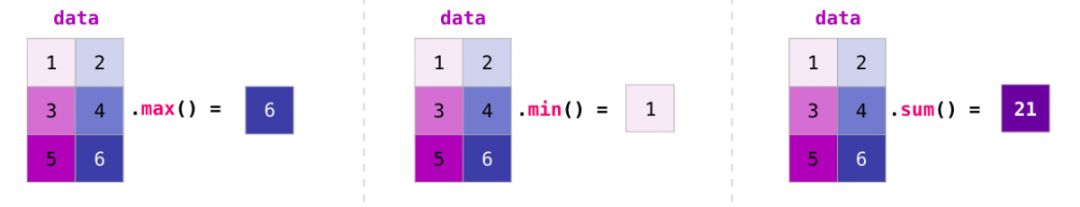


**矩阵的切片和聚合**

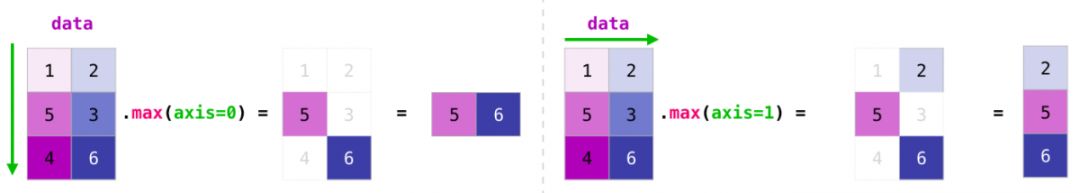
索引和切片功能在操作矩阵时变得更加有用。可以在不同维度上使用索引操作来对数据进行切片。



我们可以像聚合向量一样聚合矩阵：

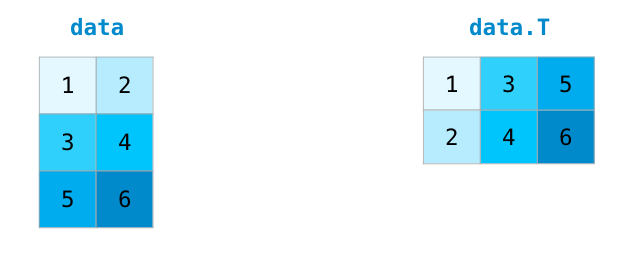


不仅可以聚合矩阵中的所有值，还可以使用axis参数指定行和列的聚合：

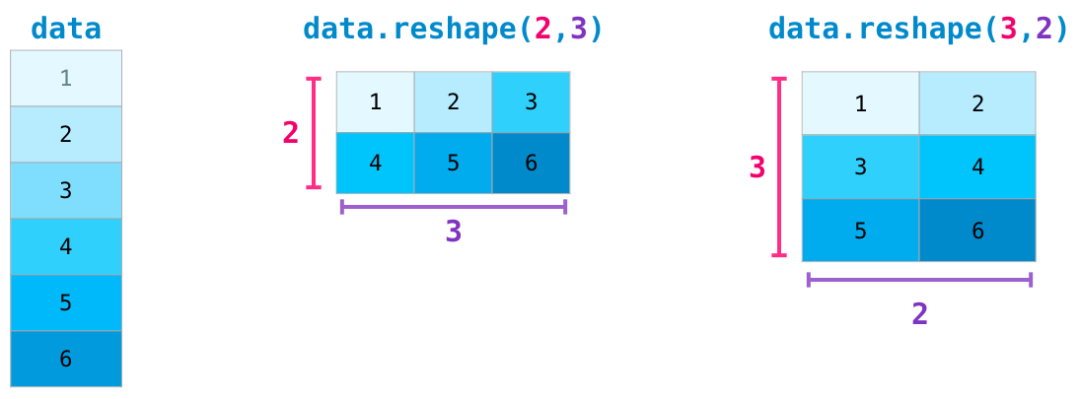


**矩阵的转置和重构**

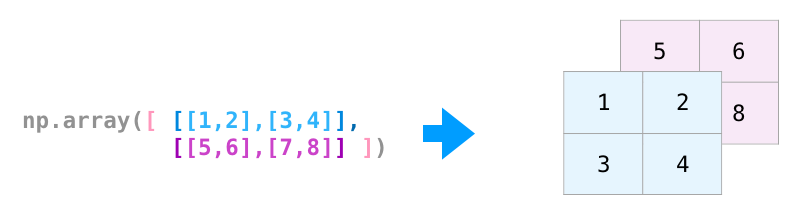
处理矩阵时经常需要对矩阵进行转置操作，常见的情况如计算两个矩阵的点积。NumPy数组的属性T可用于获取矩阵的转置。



在较为复杂的用例中，你可能会发现自己需要改变某个矩阵的维度。这在机器学习应用中很常见，例如模型的输入矩阵形状与数据集不同，可以使用NumPy的reshape()方法。只需将矩阵所需的新维度传入即可。也可以传入-1，NumPy可以根据你的矩阵推断出正确的维度：



上文中的所有功能都适用于多维数据，其中心数据结构称为ndarray（N维数组）。



很多时候，改变维度只需在NumPy函数的参数中添加一个逗号，如下图所示：

