一、数据操作

（1）获取数据（2）读入计算机

用tensor，支持GPU。深度学习存储和操作数据的主要接口是张量（𝑛维数组）

1.torch的基本用法，类似numpy

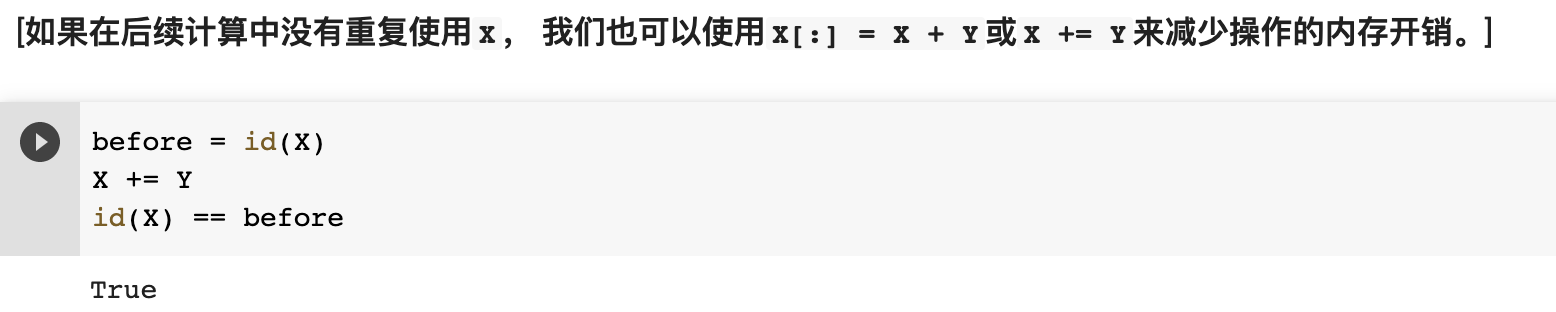
2.运算符

3.广播机制

机制：沿着数组中长度为1的轴进行广播。所以只有每个张量的长度为1的轴彼此可以不同，其他所有轴必须相同。因为维度按照右对齐后，需要复制的维度只有一行/列或者为空，其余行/列不需要复制。

4.索引和切片

5.节省内存：原地操作



cf.X+=Y 没有调用 X !

6.转换为其他python对象

二、数据预处理

1.读取数据

2.处理缺失值：插值/删除

3.转换为张量格式

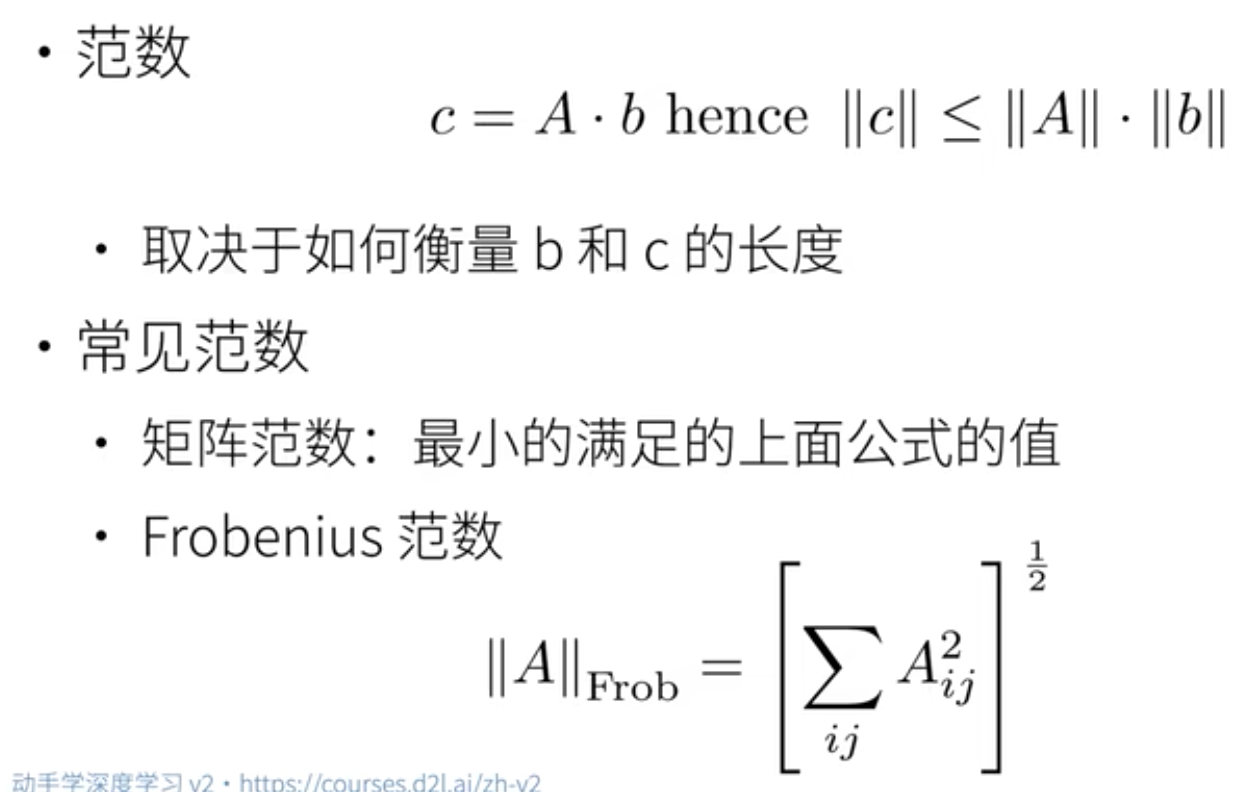
三、线性代数

“矩阵可以分解为因子，这些分解可以显示真实世界数据集中的低维结构。 机器学习的整个子领域都侧重于使用矩阵分解及其向高阶张量的泛化，来发现数据集中的结构并解决预测问题。”

知识点

（1）矩阵乘法：向量的空间扭曲

（2）一般用F范数，简单



（3）特征向量

实现

降维：求和是沿指定的维度降维。

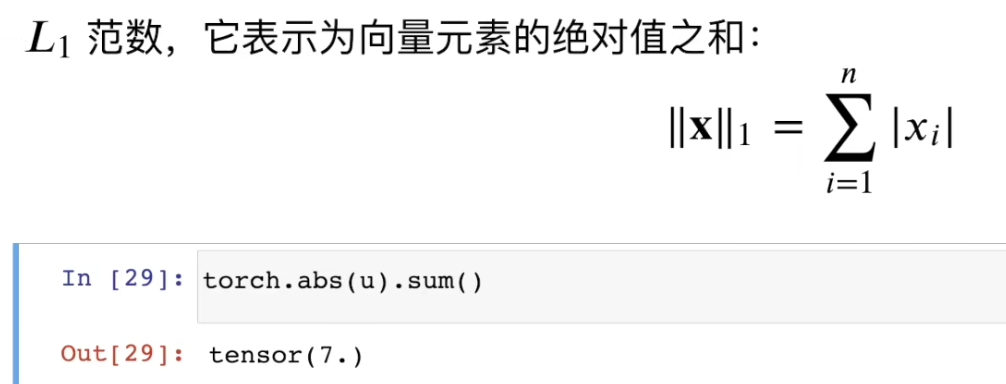
非降维求和：用keepdims=True来保证轴数不变，从而可以用广播机制。

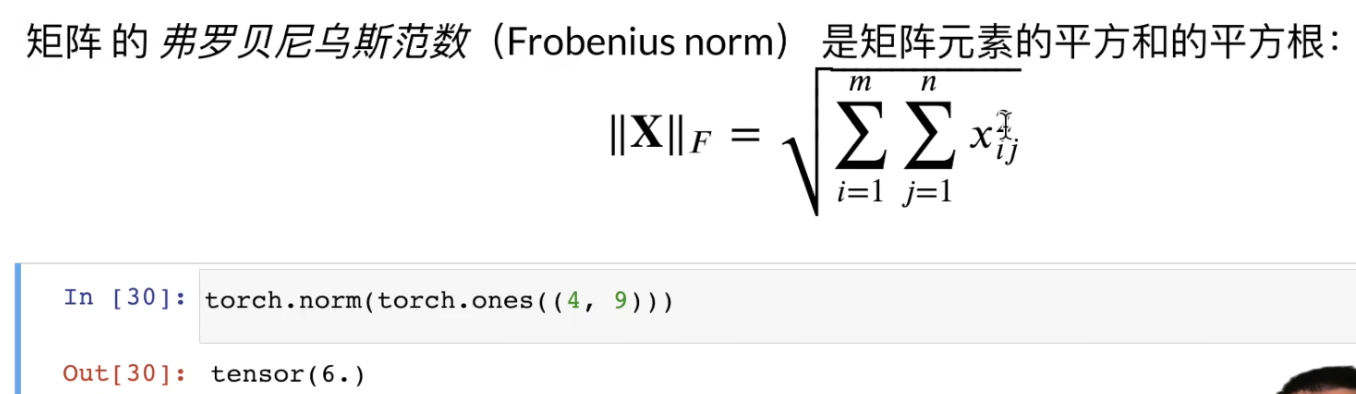
cumsum累积总和

范数：等于把矩阵拉成一个向量，再计算长度

向量范数：将向量映射到标量的函数。向量的范数表示一个向量的大小



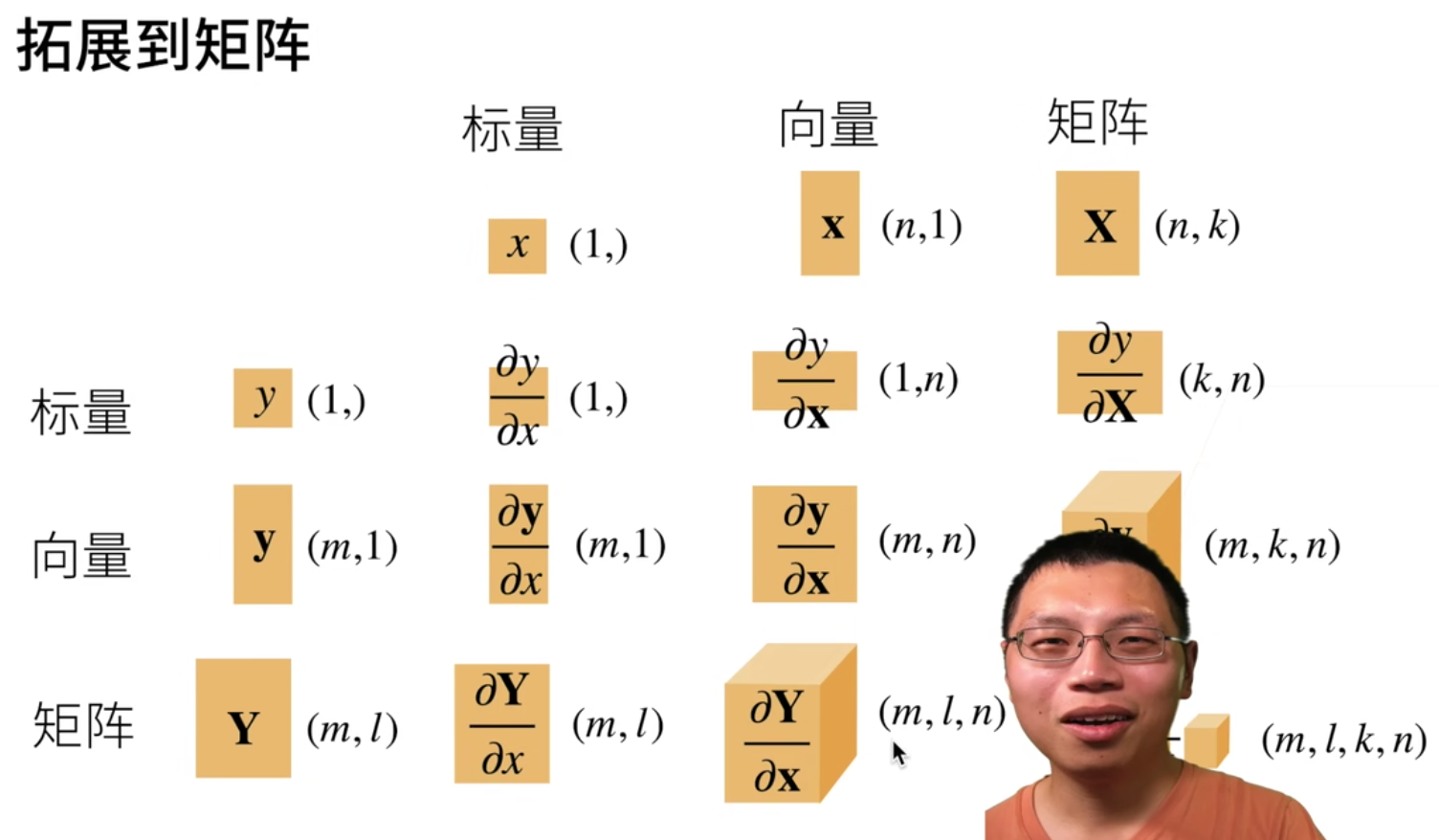




深度学习中的优化问题： 最大化分配给观测数据的概率; 最小化预测和真实观测之间的距离。 用向量表示物品（如单词、产品或新闻文章），以便最小化相似项目之间的距离，最大化不同项目之间的距离。 目标，或许是深度学习算法最重要的组成部分（除了数据），通常被表达为范数。

四、矩阵计算

标量导数



标量关于列向量的导数是行向量

梯度与等高线正交。梯度指向值变化最大的方向【机器学习求解的核心思想】

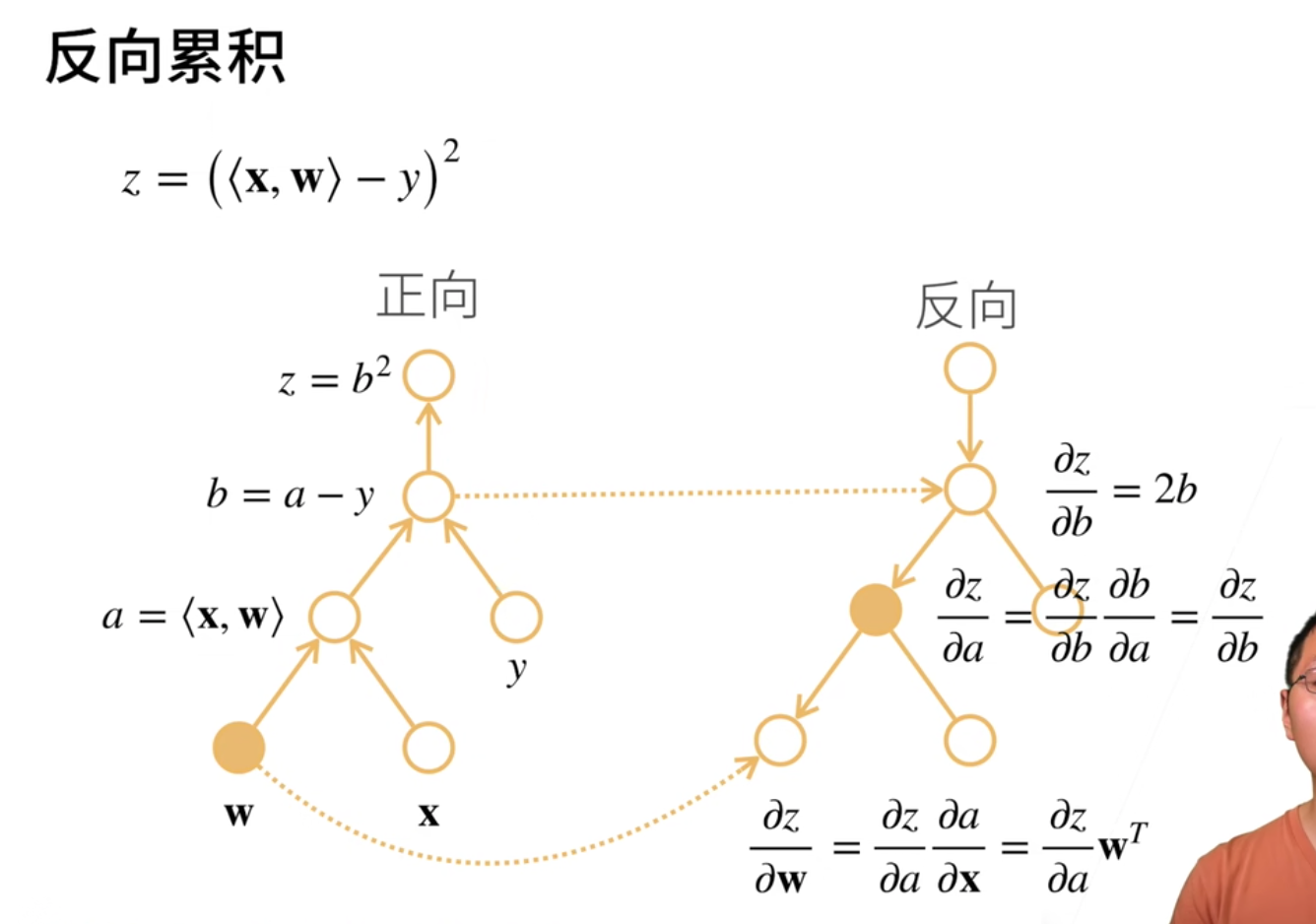
y是列向量，先把y拆解；然后每个y的元素对于x的导数是行向量；所以最终是矩阵。

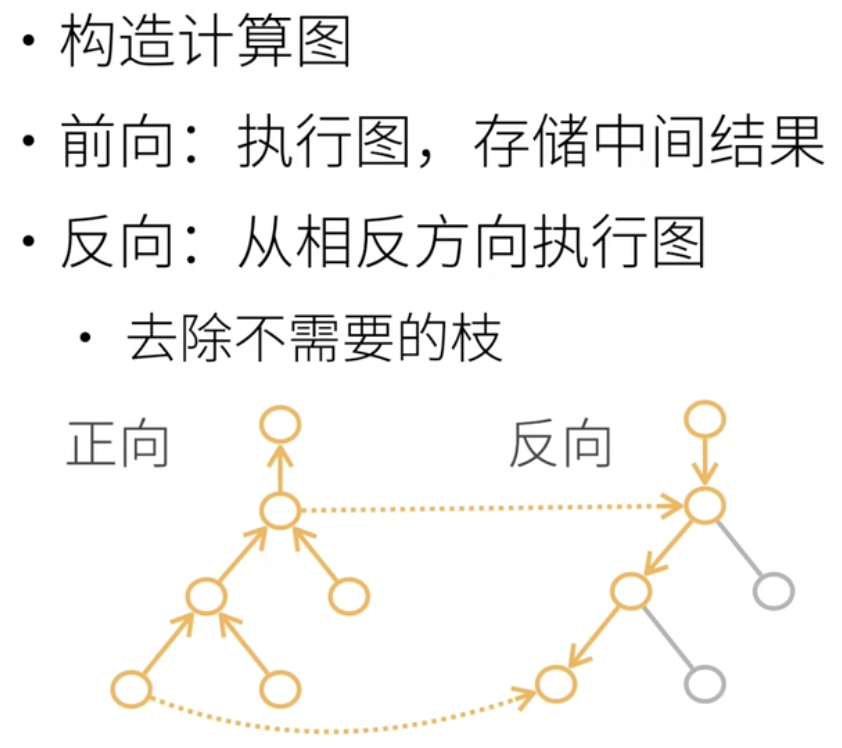
【如果下面是矩阵，结果需要把行列向量转一下；如果上面是矩阵，就不变】

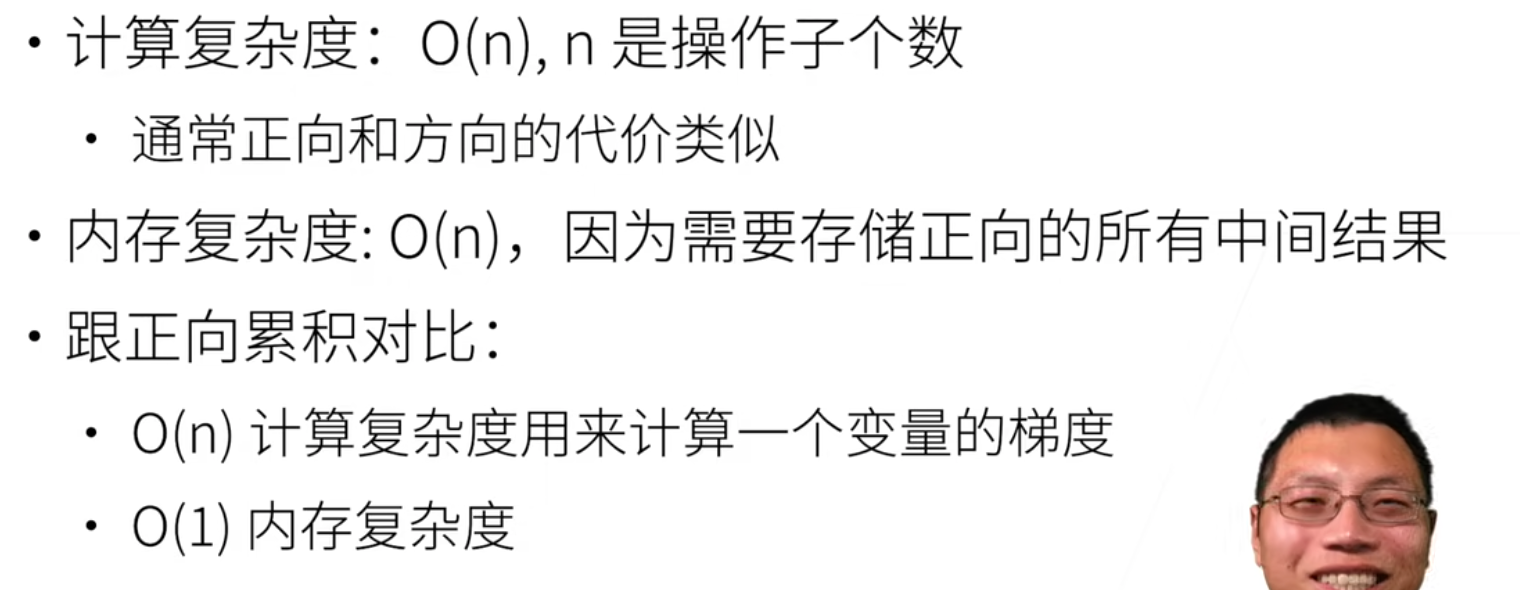
五、自动求导

计算一个函数在指定值上的导数，不需要知道导数的函数。

原理：计算图







神经网络的正向和反向的计算复杂度差不多，但是内存复杂度差很多，因为求梯度时需要存储前面的结果。（所以说深度神经网络耗gpu资源，祸源）

实现

X.grad存储梯度

X.grad.zero\_()梯度清零

Y.sum().backward()非标量的反向传播

U=y.detach()分离计算