1. Tensor的概念

Tensor 中文为张量。张量的意思是一个多维数组，它是标量、向量、矩阵的高维扩展。

标量可以称为 0 维张量，向量可以称为 1 维张量，矩阵可以称为 2 维张量，RGB 图像可以表示 3 维张量。你可以把张量看作多维数组。



1. Tensor的创建
2. 直接创建

torch.tensor(data, dtype=None, device=None, requires\_grad=False, pin\_ memory=False)

* data: 数据，可以是 list，numpy
* dtype: 数据类型，默认与 data 的一致
* device: 所在设备，cuda/cpu
* requires\_grad: 是否需要梯度
* pin\_memory: 是否存于锁页内存

1. 通过numpy数组创建

torch.from\_numpy(ndarray)

1. 根据数值创建

torch.zeros(\*size, out=None, dtype=None, layout=torch.strided, device= None, requires\_grad=False) 创建全为0的张量

torch.zeros\_like(input, dtype=None, layout=None, device=None, requires \_grad=False, memory\_format=torch.preserve\_format) 根据input的形状创建全0张量

torch.arange(start=0, end, step=1, out=None, dtype=None, layout= torch.strided, device=None, requires\_grad=False) 根据初始（start）结束（end）和步长（step）创建等差的一维张量

torch.linspace(start, end, steps=100, out=None, dtype=None, layout=tor ch.strided, device=None, requires\_grad=False) 根据初始和结束创建步长为1的一维张量

1. 创建随机序列

torch.randn(\*size, out=None, dtype=None, layout=torch.strided, device= None, requires\_grad=False) 生成形状为”size”的标准正态分布

randint(low=0, high, size, \*, generator=None, out=None,

dtype=None, layout=torch.strided, device=None, requires\_grad=False) 生成区间[low,height）上的整数均匀分布

1. tensor操作
2. 拼接

torch.cat(tensors, dim=0, out=None) 按照dim维度（类似于numpy中的axis）进行拼接

1. 划分

torch.chunk(input, chunks, dim=0) 按照dim维度划分为chunks份张量

1. 索引

torch.index\_select(input, dim, index, out=None) 按照dim维度取出index索引的数据并拼接成张量返回

index的数据类型为1\*n的张量

1. 变换

torch.reshape(input, shape) 类似于np.reshape