



8.4 部分采样

索引和切片

索引

#一维张量
`a[1]`

#二维张量
`b[1][1]`
`b[1,1]`

#三维张量
`c[1][1][1]`
`c[1,1,1]`

例：手写数字图像数据集MNIST: **(60000, 28, 28)**

mnist[0]：取第1张图片中的数据

mnist[0][1]：取第1张图片中的第2行

mnist[0][1][2]：取第1张图片中的第2行的第3列



□ 切片

起始位置: 结束位置: 步长

- **起始位置: 结束位置**, 是**前闭后开**的, 切片中不包含结束位置
- 起始位置、结束位置、步长都可以省略
- 步长可以是**负数**, 这时起始位置的索引号, 应该大于结束位置

```
In [1]: import tensorflow as tf
```

```
In [2]: a=tf.range(10)
```

```
In [3]: a[::]
```

当3个参数全部省略时, 表示读取所有数据, 步长为1

```
Out[3]: <tf.Tensor: id=7, shape=(10,), dtype=int32, numpy=array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])>
```

```
In [4]: a[::2]
```

从第一个元素开始, 隔行采样, 直到最后一个元素为止

```
Out[4]: <tf.Tensor: id=11, shape=(5,), dtype=int32, numpy=array([0, 2, 4, 6, 8])>
```



□ 一维张量切片

In [5]: a[1::2] 读出所有的奇数

Out[5]: <tf.Tensor: id=15, shape=(5,), dtype=int32, numpy=array([1, 3, 5, 7, 9])>

In [6]: a[::-1] 从最后一个元素开始, 逆序取出所有元素

Out[6]: <tf.Tensor: id=19, shape=(10,), dtype=int32, numpy=array([9, 8, 7, 6, 5, 4, 3, 2, 1, 0])>

In [7]: a[::-2] 从最后一个元素开始, 逆序取出间隔的元素

Out[7]: <tf.Tensor: id=23, shape=(5,), dtype=int32, numpy=array([9, 7, 5, 3, 1])>



□ 下载鸢尾花数据集

下载鸢尾花数据集

```
In [8]: TRAIN_URL = "http://download.tensorflow.org/data/iris_training.csv"
train_path = tf.keras.utils.get_file(TRAIN_URL.split('/')[-1], TRAIN_URL)
```

```
In [9]: import pandas as pd
df_iris=pd.read_csv(train_path)
```

使用Pandas读取到二维表格对象df_iris中

```
In [10]: import numpy as np
np_iris=np.array(df_iris)
```

转换为NumPy数组np_iris

```
In [11]: iris=tf.convert_to_tensor(np_iris)
iris.shape
```

转换为张量

```
Out[11]: TensorShape([120, 5])
```

二维张量，共120个样本，
前4列是属性，最后1列是标记



□ 二维张量切片：维度之间用逗号隔开

```
In [12]: iris[0,:] 读取第一个样本的所有列（包括属性和标记）
```

```
Out[12]: <tf.Tensor: id=28, shape=(5,), dtype=float64, numpy=array([6.4, 2.8, 5.6, 2.2, 2. ])>
```

```
In [13]: iris[0:5,0:4] 读取前5个样本的所有属性
```

```
Out[13]: <tf.Tensor: id=32, shape=(5, 4), dtype=float64, numpy=
array([[6.4, 2.8, 5.6, 2.2],
       [5. , 2.3, 3.3, 1. ],
       [4.9, 2.5, 4.5, 1.7],
       [4.9, 3.1, 1.5, 0.1],
       [5.7, 3.8, 1.7, 0.3]])>
```



8.4 部分采样

```
In [14]: iris[:,0]
```

读取所有样本的第1个属性

```
Out[14]: <tf.Tensor: id=36, shape=(120,), dtype=float64, numpy=
array([[6.4, 5. , 4.9, 4.9, 5.7, 4.4, 5.4, 6.9, 6.7, 5.1, 5.2, 6.9, 5.8,
        5.4, 7.7, 6.3, 6.8, 7.6, 6.4, 5.7, 6.7, 6.4, 5.4, 6.1, 7.2, 5.2,
        5.8, 5.9, 5.4, 6.7, 6.3, 5.1, 6.4, 6.8, 6.2, 6.9, 6.5, 5.8, 5.1,
        4.8, 7.9, 5.8, 6.7, 5.1, 4.7, 6. , 4.8, 7.7, 4.6, 7.2, 5. , 6.6,
        6.1, 5. , 7. , 6. , 7.4, 5.8, 6.2, 5. , 5.6, 6.7, 6.3, 6.4, 6.2,
        7.3, 4.4, 7.2, 6.5, 5. , 4.7, 6.6, 5.5, 7.7, 6.1, 4.9, 5.5, 5.7,
        6. , 6.4, 5.4, 6.1, 6.5, 5.6, 6.3, 4.9, 6.8, 5.7, 6. , 5. , 6.5,
        6.1, 5.1, 4.6, 6.5, 4.6, 4.6, 7.7, 5.9, 5.1, 4.9, 4.9, 4.5, 5.8,
        5. , 5.2, 5.3, 5. , 5.6, 4.8, 6.3, 5.7, 5. , 6.3, 5. , 5.5, 5.7,
        4.4, 4.8, 5.5])>
```

```
In [15]: iris[0:10,0:4]
```

读取前10个样本的所有属性

```
Out[15]: <tf.Tensor: id=40, shape=(10, 4), dtype=float64, numpy=
array([[6.4, 2.8, 5.6, 2.2],
       [5. , 2.3, 3.3, 1. ],
       [4.9, 2.5, 4.5, 1.7],
       [4.9, 3.1, 1.5, 0.1],
       [5.7, 3.8, 1.7, 0.3],
       [4.4, 3.2, 1.3, 0.2],
       [5.4, 3.4, 1.5, 0.4],
       [6.9, 3.1, 5.1, 2.3],
       [6.7, 3.1, 4.4, 1.4],
       [5.1, 3.7, 1.5, 0.4]])>
```



□ 三维张量切片——手写数字数据集MNIST (60000,28,28)

`mnist[0, :, :]`: 第1张图片

`[0, :, :]`

为了更加简洁，两个冒号可以简写为一个冒号

`mnist[0:10, :, :]`: 前10张图片

`mnist[0:20, 0:28:2, :]`: 前20张图片的所有的行，隔行采样

`mnist[:, 0:28:2, 0:28:2]`: 对所有的图片，隔行采样，并且隔列采样



□ 三维张量切片——彩色图片lena (512, 512, 3)

`lena[:, :, 0]`: R通道的图片

`lena[:, :, 2]`: B通道的图片

□ 四维张量切片——多张彩色图片 (4, 512, 512, 3)

`image[0, :, :, 0]`: 第1张图片的R通道

`image[0:2, :, :, 2]`: 前2张图片的B通道

`image[0:2, 0:512:2, :, 2]`: 对前2张图片的B通道图片隔行采样

采用切片的方式，只能进行连续的、或者有规律的采样



■ 数据提取：根据索引，抽取出不规律的、特定的数据

□ **gather()**函数：用一个**索引列表**，将**给定张量**中对应索引值的元素提取出来

gather(params, indices)

输入张量

索引值列表

```
In [16]: a=tf.range(5)
```

```
In [17]: tf.gather(a, indices=[0, 2, 3])
```

从张量a中提取索引值分别为 0,2,3 的元素

```
Out[17]: <tf.Tensor: id=47, shape=(3, ), dtype=int32, numpy=array([0, 2, 3])>
```



□ 对多维张量采样——gather()、gather_nd()函数

gather(params, **axis**, indices)

说明在哪个轴上采样

gather()函数一次对**一个维度**进行索引

创建张量

```
In [18]: a=tf.range(20)  
b=tf.reshape(a, [4,5])  
b
```

创建二维张量

```
Out[18]: <tf.Tensor: id=53, shape=(4, 5), dtype=int32, numpy=  
array([[ 0,  1,  2,  3,  4],  
       [ 5,  6,  7,  8,  9],  
       [10, 11, 12, 13, 14],  
       [15, 16, 17, 18, 19]])>
```



8.4 部分采样

对axis=0的轴采样 (对行采样)

```
In [19]: tf.gather(b, axis=0, indices=[0, 2, 3])
```

```
Out[19]: <tf.Tensor: id=56, shape=(3, 5), dtype=int32, numpy=
array([[ 0,  1,  2,  3,  4],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])>
```

对axis=1的轴采样 (对列采样)

```
In [20]: tf.gather(b, axis=1, indices=[0, 2, 3])
```

```
Out[20]: <tf.Tensor: id=59, shape=(4, 3), dtype=int32, numpy=
array([[ 0,  2,  3],
       [ 5,  7,  8],
       [10, 12, 13],
       [15, 17, 18]])>
```

axis=1 →

0	[[0, 1, 2, 3, 4],
1	[5, 6, 7, 8, 9],
2	[10, 11, 12, 13, 14],
3	[15, 16, 17, 18, 19]]

axis=0 ↓

	[[0, 1, 2, 3, 4],
	[5, 6, 7, 8, 9],
	[10, 11, 12, 13, 14],
	[15, 16, 17, 18, 19]]
0	
1	
2	
3	
4	



□ 同时采样多个点——gather_nd()函数

```
[[0, 1, 2, 3, 4],  
 [5, 6, 7, 8, 9],  
 [10, 11, 12, 13, 14],  
 [15, 16, 17, 18, 19]]
```

```
[21]: tf.gather_nd(b, [[0, 0], [1, 1], [2, 3]])
```

```
[21]: <tf.Tensor: id=61, shape=(3,), dtype=int32, numpy=array  
      ([ 0, 6, 13])>
```

和使用3次索引的结果是一样的

```
In [22]: b[0, 0]
```

```
Out[22]: <tf.Tensor: id=65, shape=(), dtype=int32, numpy=0>
```

```
In [23]: b[1, 1]
```

```
Out[23]: <tf.Tensor: id=69, shape=(), dtype=int32, numpy=6>
```

```
In [24]: b[2, 3]
```

```
Out[24]: <tf.Tensor: id=73, shape=(), dtype=int32, numpy=13>
```

- 通过指定坐标，同时采样多个点
- 可以同时为多个维度进行索引



□ 选择采样维度

三维张量——彩色图片lena (512, 512, 3)

```
tf.gather_nd(lena, [[0, 0], [1, 1], [2, 3]])
```

对前两维采样，表示取得这些点所有通道的值

三维张量——手写数字数据集MNIST (60000,28,28)

```
tf.gather_nd(mnist, [[0],[2],[3]])
```

只对第一维采样，表示取到索引值为0,2,3的3张图片

