



PyTorch

索引与切片

主讲人：龙良曲

Indexing

索引

- dim 0 first

照片宏观到微观

方式一

方式二

方式三

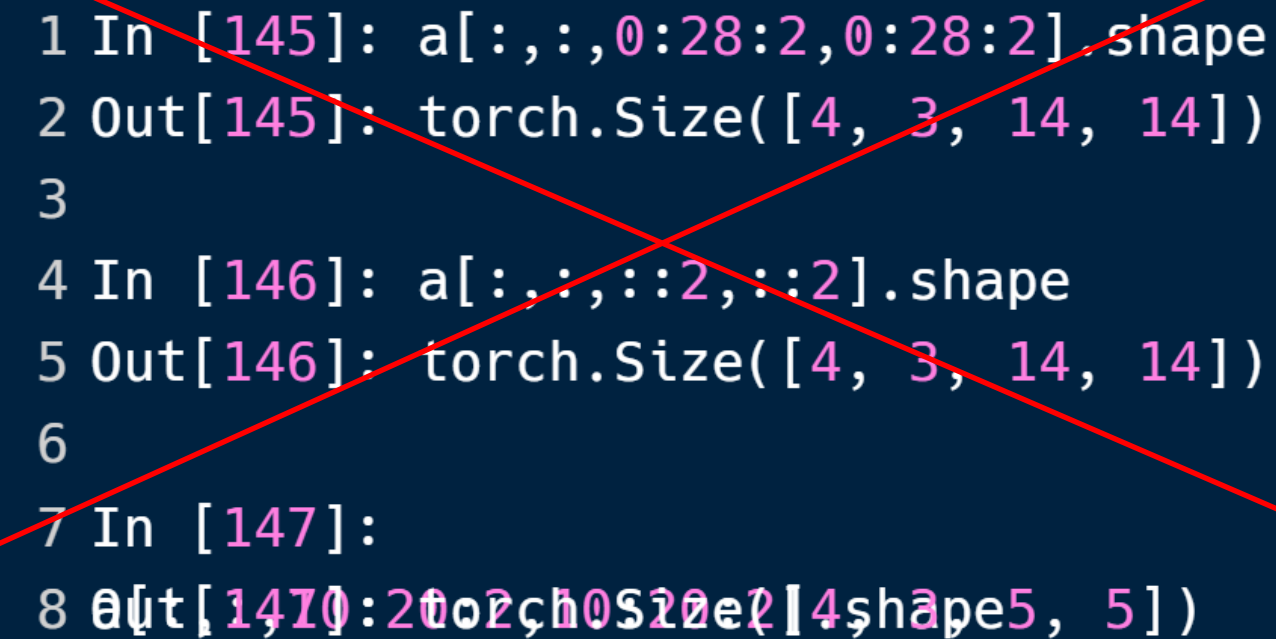
```
1 In [130]:  
2 a=torch.rand(4,3,28,28)  
3 In [131]: a[0].shape  
4 Out[131]: torch.Size([3, 28, 28])  
5  
6 In [138]: a[0,0].shape  
7 Out[138]: torch.Size([28, 28])  
8  
9 In [139]: a[0,0,2,4]  
10 Out[139]: tensor(0.8082)
```

select first/last N



```
1 In [140]: a.shape
2 Out[140]: torch.Size([4, 3, 28,
3 28])
4 In [141]: a[:2].shape
5 Out[141]: torch.Size([2, 3, 28,
6 28])
7 In [142]: a[:2,:1,:,:].shape
8 Out[142]: torch.Size([2, 1, 28,
9 28])
10 In [143]: a[:2,1,:,:].shape
11 Out[143]: torch.Size([2, 2, 28,
12 28])
13 In [144]: a[:2,-1,:,:].shape
14 Out[144]: torch.Size([2, 1, 28,
15 28])
```

~~select by steps~~



```
1 In [145]: a[:, :, 0:28:2, 0:28:2].shape
2 Out[145]: torch.Size([4, 3, 14, 14])
3
4 In [146]: a[:, :, ::2, ::2].shape
5 Out[146]: torch.Size([4, 3, 14, 14])
6
7 In [147]:
8 Out[147]: torch.Size([4, 5, 5])
```

select by specific index



```
1 In [149]: a.shape
2 Out[149]: torch.Size([4, 3, 28, 28])
3
4 In [159]: a.index_select(0,
5 Out[159]: torch.Size([1, 3, 28, 28])
6
7 In [167]: a.index_select(1,
8 Out[167]: torch.Size([4, 1, 28, 28])
9
10 In [168]: a.index_select(2, torch.arange(28)).shape
11 Out[168]: torch.Size([4, 3, 28, 28])
12
13 In [169]: a.index_select(2, torch.arange(8)).shape
14 Out[169]: torch.Size([4, 3, 8, 28])
```

...

```
1 In [149]: a.shape
2 Out[149]: torch.Size([4, 3, 28,
3 28])
4 In [150]: a[...].shape
5 Out[150]: torch.Size([4, 3, 28,
6 28])
7 In [151]: a[0,...].shape
8 Out[151]: torch.Size([3, 28, 28])
9
10 In [152]: a[:,1,...].shape
11 Out[152]: torch.Size([4, 28, 28])
12
13 In [155]: a[..., :2].shape
14 Out[155]: torch.Size([4, 3, 28, 2])
```

select by mask

- `.masked_select()`

```
1 In [170]: x = torch.randn(3, 4)
2 tensor([[ -1.3911,  -0.7871,  -1.6558,  -0.2542],
3         [-0.9011,   0.5404,  -0.6612,   0.3917],
4         [-0.3854,   0.2968,   0.6040,   1.5771]])
5
6 In [172]: mask = x.ge(0.5)
7 tensor([[0, 0, 0, 0],
8         [0, 1, 0, 0],
9         [0, 0, 1, 1]], dtype=torch.uint8)
10
11 In [174]: torch.masked_select(x, mask)
12 Out[174]: tensor([0.5404, 0.6040, 1.5771])
13
14 In [175]: torch.masked_select(x, mask).shape
15 Out[175]: torch.Size([3])
```

select by flatten index



```
1 In [176]: src = torch.tensor([[4, 3, 5],  
2     ...:                      [6, 7, 8]])  
3     ...:  
4  
5 In [177]: torch.take(src, torch.tensor([0, 2,  
6 Out[177]: tensor([4, 5, 8])
```

下一课时

Tensor变换

Thank You.
