



PyTorch

高阶OP

主讲人：龙良曲

Tensor advanced operation

- Where
- Gather

where

```
torch.where(condition, x, y) → Tensor
```

Return a tensor of elements selected from either `x` or `y`, depending on `condition`.

The operation is defined as:

$$out_i = \begin{cases} x_i & \text{if } condition_i \\ y_i & \text{otherwise} \end{cases}$$

example

```
1 In [198]: cond
2 tensor([[0.6769, 0.7271],
3         [0.8884, 0.4163]])
4
5 In [199]: a
6 tensor([[0., 0.],
7         [0., 0.]])
8
9 In [200]: b
10 tensor([[1., 1.],
11         [1., 1.]])
12
13 In [203]: torch.where(cond>0.5, a, b)
14 Out[203]:
15 tensor([[0., 0.],
16         [0., 1.]])
```

gather

```
torch.gather(input, dim, index, out=None) → Tensor
```

Gathers values along an axis specified by *dim*.

For a 3-D tensor the output is specified by:

```
out[i][j][k] = input[index[i][j][k]][j][k]    # if dim == 0  
out[i][j][k] = input[i][index[i][j][k]][k]    # if dim == 1  
out[i][j][k] = input[i][j][index[i][j][k]]    # if dim == 2
```

retrieve global label

- $\text{argmax}(\text{pred})$ to get relative labeling
 - On some condition, our label is distinct from relative labeling
-

retrieve label

```
1 In [210]: prob=torch.randn(4,10)
2
3 In [213]: idx=prob.topk(dim=1, k=3)
4 (tensor([[2.4437, 1.5195, 1.3598],
5          [1.6027, 1.4003, 0.7402],
6          [2.8965, 0.3600, 0.1961],
7          [2.2636, 0.9490, 0.3886]]), tensor([[7, 4, 9],
8          [7, 4, 9],
9          [8, 1, 3],
10         [8, 6, 0]]))
11 In [215]: idx=idx[1]
12 tensor([[7, 4, 9],
13         [7, 4, 9],
14         [8, 1, 3],
15         [8, 6, 0]])
16
17 In [216]: label=torch.arange(10)+100
18 Out[217]: tensor([100, 101, 102, 103, 104, 105, 106, 107, 108, 109])
19
20 In [220]: torch.gather(label.expand(4,10), dim=1, index=idx.long())
21 tensor([[107, 104, 109],
22         [107, 104, 109],
23         [108, 101, 103],
24         [108, 106, 100]])
```

Thank You.
