Tensor advanced operation

- Where

- Gather

https://blog.openai.com/generative-models/

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

```
In [198]: cond
tensor([[0.6769, 0.7271],
        [0.8884, 0.4163]])

In [199]: a
tensor([[0., 0.],
        [0., 0.]])

In [200]: b
tensor([[1., 1.],
        [1., 1.]])

In [203]: torch.where(cond>0.5, a, b)
Out[203]:
tensor([[0., 0.],
        [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

- argmax（pred） to get relative labeling
- On some condition, our label is dinstinct from relative labeling

retrieve label

```
In [210]: prob=torch.randn(4,10)

In [213]: idx=prob.topk(dim=1, k=3)
(tensor([[2.4437, 1.5195, 1.3598],
         [1.6027, 1.4003, 0.7402],
         [2.8965, 0.3600, 0.1961],
         [2.2636, 0.9490, 0.3886]]), tensor([[7, 4, 9],
         [7, 4, 9],
         [8, 1, 3],
         [8, 6, 0]]))
In [215]: idx=idx[1]
tensor([[7, 4, 9],
        [7, 4, 9],
        [8, 1, 3],
        [8, 6, 0]])

In [216]: label=torch.arange(10)+100
Out[217]: tensor([100, 101, 102, 103, 104, 105, 106, 107, 108, 109])

In [220]: torch.gather(label.expand(4,10), dim=1, index=idx.long())
tensor([[107, 104, 109],
        [107, 104, 109],
        [108, 101, 103],
        [108, 106, 100]])
```