



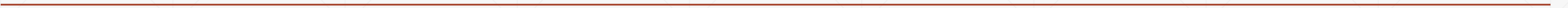
TensorFlow

高阶OP

主讲：龙良曲

Outline

- where
- scatter_nd
- meshgrid



Where(tensor)

True	False	False
False	True	False
False	False	True

where

```
In [3]: a=tf.random.normal([3,3])
<tf.Tensor: id=11, shape=(3, 3), dtype=float32, numpy=
array([[ 1.6420907 ,  0.43938753, -0.31872085],
       [ 1.144599 , -0.02425919, -0.9576591 ],
       [ 1.5931814 ,  0.1182256 , -0.39948994]], dtype=float32)>

In [5]: mask=a>0
<tf.Tensor: id=14, shape=(3, 3), dtype=bool, numpy=
array([[ True,  True, False],
       [ True, False, False],
       [ True,  True, False]])>

In [7]: tf.boolean_mask(a,mask)
<tf.Tensor: id=42, shape=(5,), dtype=float32, numpy=
array([1.6420907 , 0.43938753, 1.144599 , 1.5931814 , 0.1182256 ],
      dtype=float32)>

In [8]: indices=tf.where(mask)
<tf.Tensor: id=44, shape=(5, 2), dtype=int64, numpy=
array([[0, 0],
       [0, 1],
       [1, 0],
       [2, 0],
       [2, 1]])>

In [10]: tf.gather_nd(a,indices)
<tf.Tensor: id=46, shape=(5,), dtype=float32, numpy=
array([1.6420907 , 0.43938753, 1.144599 , 1.5931814 , 0.1182256 ],
      dtype=float32)>
```

where

- `where(cond, A, B)`

true 从A取
false从B取



```
In [11]: mask
<tf.Tensor: id=14, shape=(3, 3), dtype=bool, numpy=
array([[ True,  True, False],
       [ True, False, False],
       [ True,  True, False]])>

In [12]: A=tf.ones([3,3])

In [13]: B=tf.zeros([3,3])

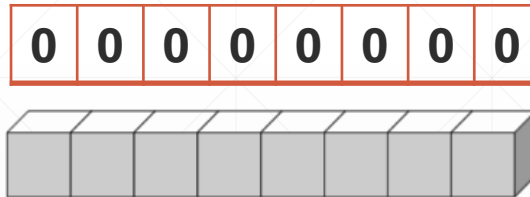
In [14]: tf.where(mask, A, B)
<tf.Tensor: id=55, shape=(3, 3), dtype=float32, numpy=
array([[1., 1., 0.],
       [1., 0., 0.],
       [1., 1., 0.]], dtype=float32)>
```

scatter_nd

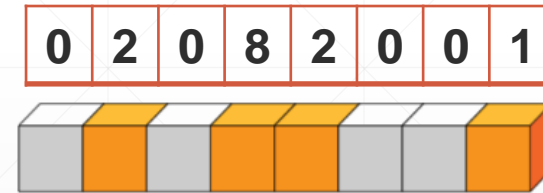
- `tf.scatter_nd(`
- `indices,`
- `updates,`
- `shape`



updates



shape



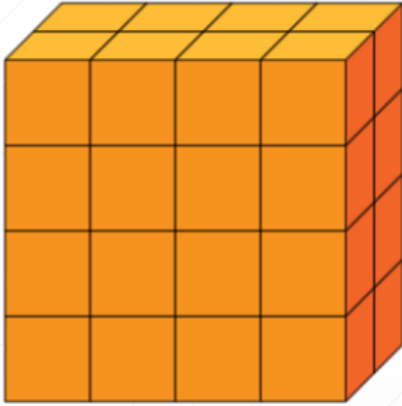
output

scatter_nd

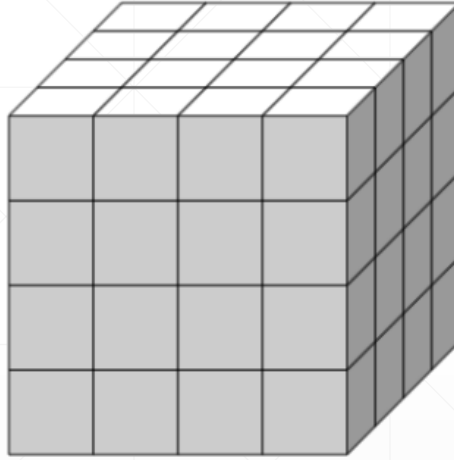
```
In [17]: indices = tf.constant([[4], [3], [1], [7]])
In [18]: updates = tf.constant([9, 10, 11, 12])
In [19]: shape = tf.constant([8])

In [20]: tf.scatter_nd(indices, updates, shape)
Out[20]: <tf.Tensor: id=60, shape=(8,), dtype=int32, numpy=array([ 0, 11,  0, 10,
 9,  0,  0, 12], dtype=int32)>
```

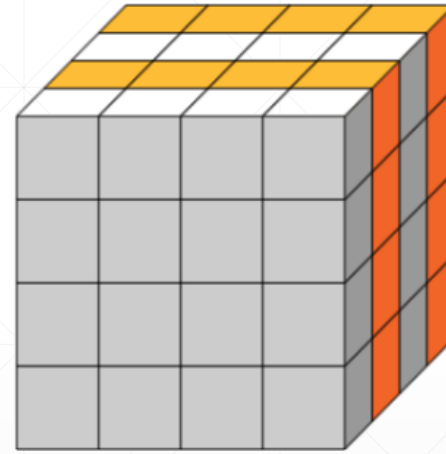
scatter_nd



updates

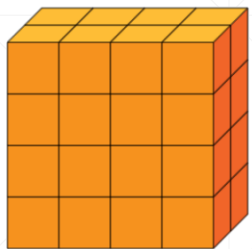


shape

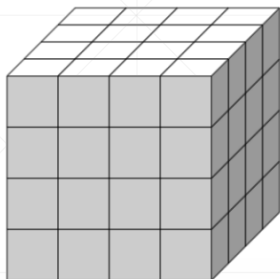


output

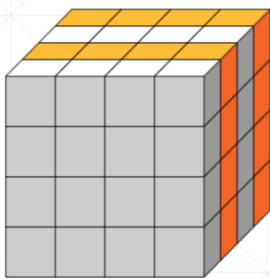
scatter_nd



updates



shape



output

```
In [21]: indices = tf.constant([[0], [2]])
```

```
In [22]: updates = tf.constant([ [[5, 5, 5, 5], [6, 6, 6, 6],  
....:                             [7, 7, 7, 7], [8, 8, 8, 8]],  
....:  
....:                             [[5, 5, 5, 5], [6, 6, 6, 6],  
....:                             [7, 7, 7, 7], [8, 8, 8, 8]]])
```

```
In [24]: updates.shape
```

```
Out[24]: TensorShape([2, 4, 4])
```

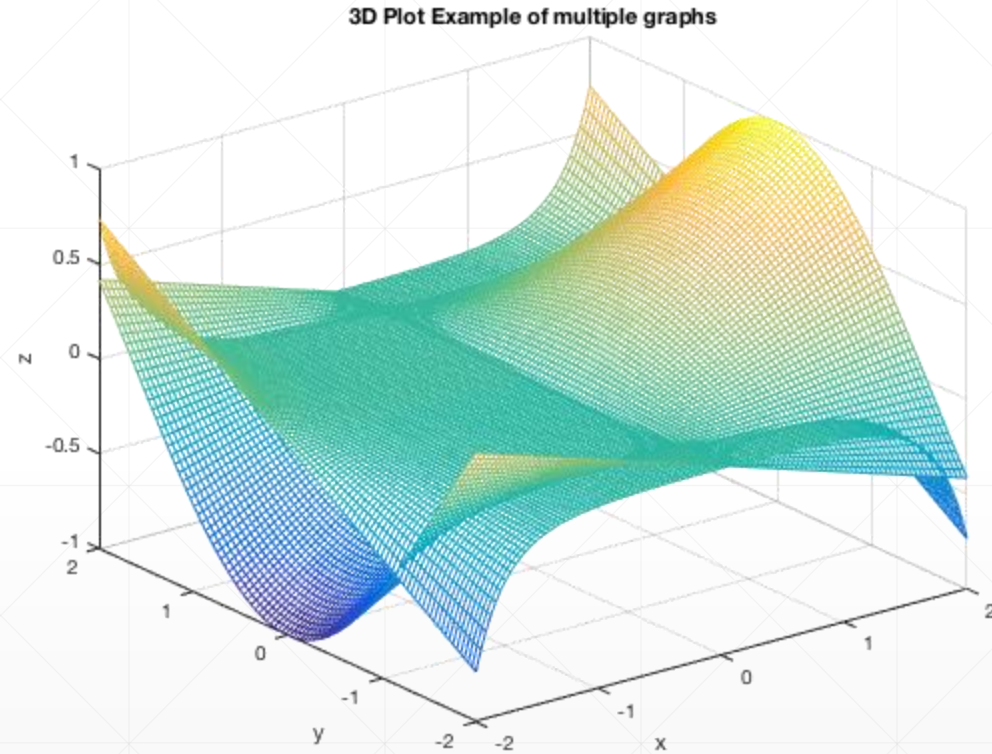
```
In [23]: shape = tf.constant([4, 4, 4])
```

```
In [25]: tf.scatter_nd(indices, updates, shape)
```

```
<tf.Tensor: id=65, shape=(4, 4, 4), dtype=int32, numpy=  
array([[[5, 5, 5, 5],  
        [6, 6, 6, 6],  
        [7, 7, 7, 7],  
        [8, 8, 8, 8]],  
       [[0, 0, 0, 0],  
        [0, 0, 0, 0],  
        [0, 0, 0, 0],  
        [0, 0, 0, 0]],  
       [[5, 5, 5, 5],  
        [6, 6, 6, 6],  
        [7, 7, 7, 7],  
        [8, 8, 8, 8]],  
       [[0, 0, 0, 0],  
        [0, 0, 0, 0],  
        [0, 0, 0, 0],  
        [0, 0, 0, 0]]], dtype=int32)>
```

meshgrid

- $[-2, -2]$
- $[-1, -2]$
- $[0, -2]$
- $[-2, -1]$
- $[-1, -1]$
- ...
- $[2, 2]$

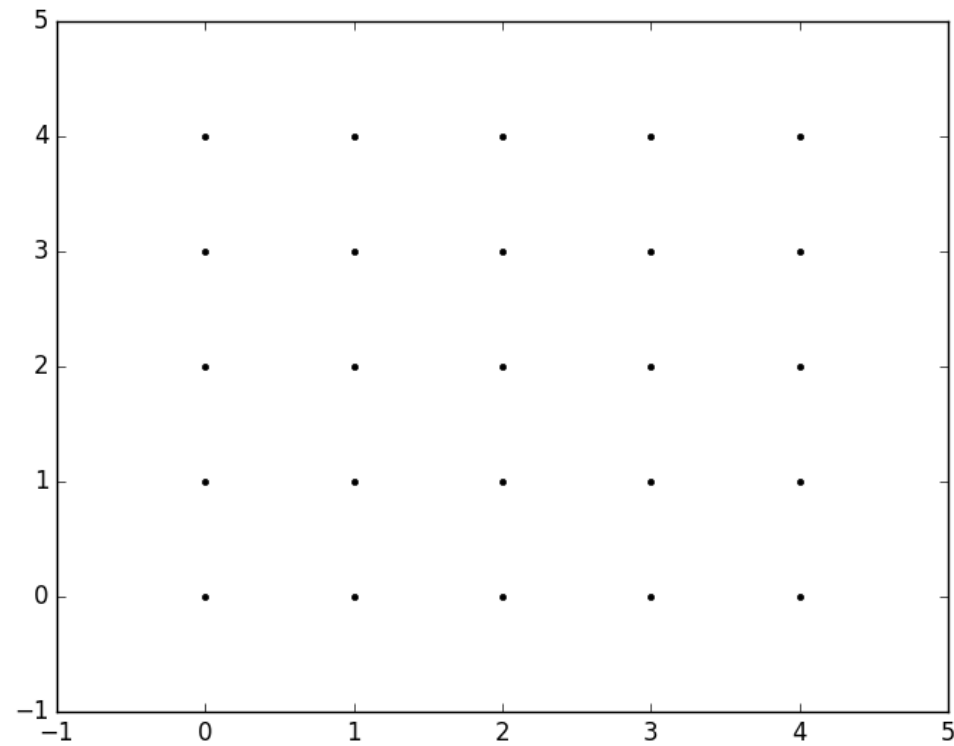


Points

- $[y, x, 2]$

- $[5, 5, 2]$

- $[N, 2]$



Numpy



```
points = []
```

```
for y in np.linspace(-2, 2, 5):  
    for x in np.linspace(-2, 2, 5):  
        points.append([x, y])
```

```
return np.array(points)
```

GPU acceleration

- $x: [-2 \sim 2]$
 - $y: [-2 \sim 2]$
 - Points: $[N, 2]$
-



```
In [39]: y=tf.linspace(-2.,2,5)
```

```
In [40]: y
```

```
Out[40]: <tf.Tensor: id=136, shape=(5,), dtype=float32, numpy=array([-2., -1.,  
    0.,  1.,  2.], dtype=float32)>
```

```
In [41]: x=tf.linspace(-2.,2,5)
```

```
In [42]: points_x, points_y=tf.meshgrid(x,y)
```

```
In [43]: points_x.shape
```

```
Out[43]: TensorShape([5, 5])
```

```
In [44]: points_x  
<tf.Tensor: id=162, shape=(5, 5), dtype=float32, numpy=  
array([[ -2.,  -1.,   0.,   1.,   2.],  
       [ -2.,  -1.,   0.,   1.,   2.],  
       [ -2.,  -1.,   0.,   1.,   2.],  
       [ -2.,  -1.,   0.,   1.,   2.],  
       [ -2.,  -1.,   0.,   1.,   2.]], dtype=float32)>
```

```
In [45]: points_y  
<tf.Tensor: id=163, shape=(5, 5), dtype=float32, numpy=  
array([[ -2.,  -2.,  -2.,  -2.,  -2.],  
       [ -1.,  -1.,  -1.,  -1.,  -1.],  
       [  0.,   0.,   0.,   0.,   0.],  
       [  1.,   1.,   1.,   1.,   1.],  
       [  2.,   2.,   2.,   2.,   2.]], dtype=float32)>
```

Points:[N, 2]



```
[51]: points_x.shape
```

```
Out[51]: TensorShape([5, 5])
```

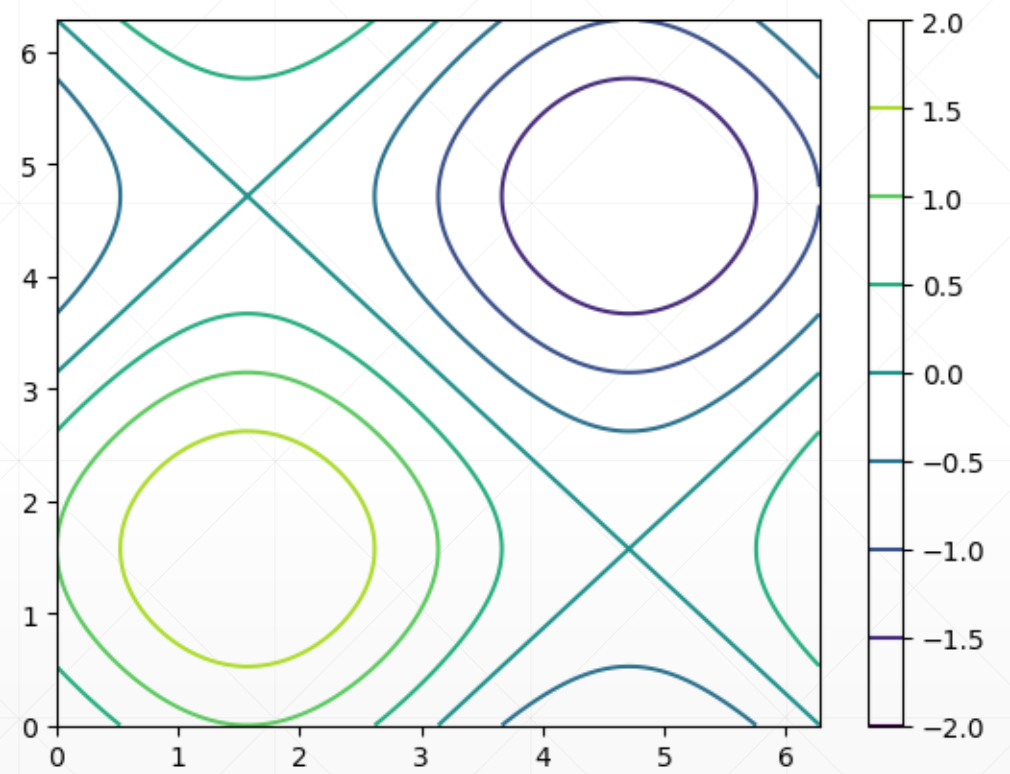
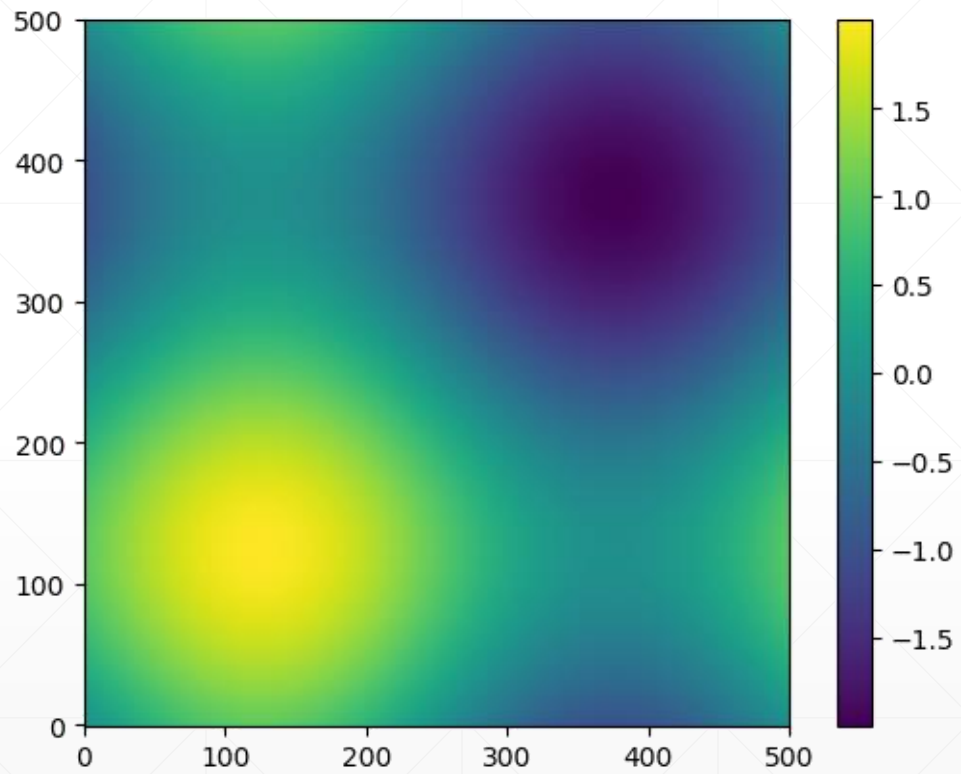
```
In [52]: points=tf.stack([points_x,points_y], axis=2)
```

```
<tf.Tensor: id=176, shape=(5, 5, 2), dtype=float32, numpy=
```

```
array([[[ -2., -2.],
         [ -1., -2.],
         [  0., -2.],
         [  1., -2.],
         [  2., -2.]],
```

```
        [[ -2., -1.],
         [ -1., -1.],
         [  0., -1.],
         [  1., -1.],
         [  2., -1.]])
```


$$z = \sin(x) + \sin(y)$$



**JUST
DO
IT.**

The text "JUST DO IT." is rendered in a bold, black, sans-serif font. The letters are heavily textured with a splatter or ink-blot effect, giving them a gritty, dynamic appearance. The background is a light gray with a subtle, repeating diamond-shaped grid pattern. The overall composition is centered and minimalist.

下一课时

数据加载

Thank You.
