



TensorFlow

张量限幅

主讲：龙良曲

Outline

- clip_by_value
 - relu
 - clip_by_norm
 - gradient clipping
-

clip_by_value



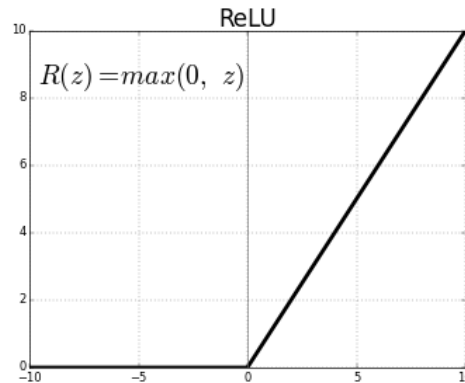
```
In [149]: a  
Out[149]: <tf.Tensor: id=422, shape=(10,), dtype=int32, numpy=array([0, 1, 2, 3,  
4, 5, 6, 7, 8, 9], dtype=int32)>
```

```
In [150]: tf.maximum(a,2)  
Out[150]: <tf.Tensor: id=426, shape=(10,), dtype=int32, numpy=array([2, 2, 2, 3,  
4, 5, 6, 7, 8, 9], dtype=int32)>
```

```
In [151]: tf.minimum(a,8)  
Out[151]: <tf.Tensor: id=429, shape=(10,), dtype=int32, numpy=array([0, 1, 2, 3,  
4, 5, 6, 7, 8, 8], dtype=int32)>
```

```
In [152]: tf.clip_by_value(a,2,8)  
Out[152]: <tf.Tensor: id=434, shape=(10,), dtype=int32, numpy=array([2, 2, 2, 3,  
4, 5, 6, 7, 8, 8], dtype=int32)>
```

relu



```
In [153]: a=a-5
```

```
Out[154]: <tf.Tensor: id=437, shape=(10,), dtype=int32, numpy=array([-5, -4, -3, -2, -1,  0,  1,  2,  3,  4], dtype=int32)>
```

```
In [155]: tf.nn.relu(a)
```

```
Out[155]: <tf.Tensor: id=439, shape=(10,), dtype=int32, numpy=array([0, 0, 0, 0, 0, 0, 1, 2, 3, 4], dtype=int32)>
```

```
In [156]: tf.maximum(a,0)
```

```
Out[156]: <tf.Tensor: id=442, shape=(10,), dtype=int32, numpy=array([0, 0, 0, 0, 0, 0, 1, 2, 3, 4], dtype=int32)>
```

clip_by_norm



```
In [157]: a=tf.random.normal([2,2],mean=10)
<tf.Tensor: id=449, shape=(2, 2), dtype=float32, numpy=
array([[12.217459 , 10.1498375],
       [10.84643  , 10.972536 ]], dtype=float32)>
```

```
In [159]: tf.norm(a)
Out[159]: <tf.Tensor: id=455, shape=(), dtype=float32, numpy=22.14333>
```

```
In [161]: aa=tf.clip_by_norm(a,15)
<tf.Tensor: id=473, shape=(2, 2), dtype=float32, numpy=
array([[8.276167 , 6.8755493],
       [7.3474245, 7.43285  ]], dtype=float32)>
```

```
In [162]: tf.norm(aa)
Out[162]: <tf.Tensor: id=496, shape=(), dtype=float32, numpy=15.000001>
```

Gradient clipping

- Gradient Exploding or vanishing
 - set lr=1
 - `new_grads, total_norm = tf.clip_by_global_norm(grads, 25)`
-

Before

```

(x, y), _ = datasets.mnist.load_data()
x = tf.convert_to_tensor(x, dtype=tf.float32) / 50.
```

```

i@z68:~/TutorialsCN/code_TensorFlow2.0/lesson18-数据限幅$ python main.py
2.0.0-dev20190225
```

```

x: (60000, 28, 28) y: (60000, 10)
```

```

sample: (128, 28, 28) (128, 10)
```

```

==before==
```

```

tf.Tensor(89.03711, shape=(), dtype=float32)
```

```

tf.Tensor(2.6175494, shape=(), dtype=float32)
```

```

tf.Tensor(118.17449, shape=(), dtype=float32)
```

```

tf.Tensor(2.1617627, shape=(), dtype=float32)
```

```

tf.Tensor(134.27968, shape=(), dtype=float32)
```

```

tf.Tensor(2.5254946, shape=(), dtype=float32)
```

```

0 loss: 28.88848876953125
```

```

==before==
```

```

tf.Tensor(1143.292, shape=(), dtype=float32)
```

```

tf.Tensor(35.841225, shape=(), dtype=float32)
```

```

tf.Tensor(1279.236, shape=(), dtype=float32)
```

```

tf.Tensor(24.312374, shape=(), dtype=float32)
```


```

tf.Tensor(1185.6311, shape=(), dtype=float32)
```

```

tf.Tensor(17.80448, shape=(), dtype=float32)
```

Gradient Clipping



```
print('==before==')
for g in grads:
    print(tf.norm(g))
grads, _ = tf.clip_by_global_norm(grads, 15)

print('==after==')
for g in grads:
    print(tf.norm(g))
```

After:

```
i@z68:~/TutorialsCN/code_TensorFlow2.0/lesson18-数据限幅$ python main.py
2.0.0-dev20190225
x: (60000, 28, 28) y: (60000, 10)
sample: (128, 28, 28) (128, 10)
==before==
tf.Tensor(118.00854, shape=(), dtype=float32)
tf.Tensor(3.5821552, shape=(), dtype=float32)
tf.Tensor(146.76697, shape=(), dtype=float32)
tf.Tensor(2.830059, shape=(), dtype=float32)
tf.Tensor(183.28879, shape=(), dtype=float32)
tf.Tensor(3.4088597, shape=(), dtype=float32)
==after==
tf.Tensor(6.734187, shape=(), dtype=float32)
tf.Tensor(0.20441659, shape=(), dtype=float32)
tf.Tensor(8.375294, shape=(), dtype=float32)
tf.Tensor(0.16149803, shape=(), dtype=float32)
tf.Tensor(10.45942, shape=(), dtype=float32)
tf.Tensor(0.19452743, shape=(), dtype=float32)
0 loss: 41.25679016113281
```

**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

高阶特性

Thank You.
