



基本运算

主讲人：龙良曲

Math operation

- Add/minus/multiply/divide
- Matmul
- Pow
- Sqrt/rsqrt
- Round

basic

```
1 In [2]: a=torch.rand(3,4)
2 In [3]: b=torch.rand(4)
3
4 In [4]: a+b
5 Out[4]:
6 tensor([[0.7114, 1.7711, 1.1941, 0.8124],
7         [0.8159, 0.8690, 1.2145, 0.5990],
8         [0.3545, 1.3203, 0.4354, 0.1347]])
9
10 In [5]: torch.add(a,b)
11 Out[5]:
12 tensor([[0.7114, 1.7711, 1.1941, 0.8124],
13         [0.8159, 0.8690, 1.2145, 0.5990],
14         [0.3545, 1.3203, 0.4354, 0.1347]])
15
16 In [6]: torch.all(torch.eq(a-b, torch.sub(a,b)))
17 Out[6]: tensor(1, dtype=torch.uint8)
18
19 In [7]: torch.all(torch.eq(a*b, torch.mul(a,b)))
20 Out[7]: tensor(1, dtype=torch.uint8)
21
22 In [8]: torch.all(torch.eq(a/b, torch.div(a,b)))
23 Out[8]: tensor(1, dtype=torch.uint8)
24
```

matmul

- `Torch.mm`
 - only for 2d
 - `Torch.matmul`
 - `@`
-

```
1 In [17]: a
2 tensor([[3., 3.],
3         [3., 3.]])
4
5 In [18]: b=torch.ones(2,2)
6 tensor([[1., 1.],
7         [1., 1.]])
8
9 In [20]: torch.mm(a,b)
10 tensor([[6., 6.],
11         [6., 6.]])
12
13 In [21]: torch.matmul(a,b)
14 Out[21]:
15 tensor([[6., 6.],
16         [6., 6.]])
17
18 In [22]: a@b
19 Out[22]:
20 tensor([[6., 6.],
21         [6., 6.]])
```

An example



```
1 In [24]: a=torch.rand(4,784)
2
3 In [25]: x=torch.rand(4,784)
4
5 In [26]: w=torch.rand(512,784)
6
7 In [27]: (x@w.t()).shape
8 Out[27]: torch.Size([4, 512])
9
```

> 2d tensor matmul?

```
1 In [28]: a=torch.rand(4,3,28,64)
2 In [29]: b=torch.rand(4,3,64,32)
3
4 In [30]: torch.mm(a,b).shape
5 -----
6 RuntimeError                                Traceback (most recent call last)
7 <ipython-input-30-18c616c4b668> in <module>()
8 ----> 1 torch.mm(a,b).shape
9 RuntimeError: matrices expected, got 4D, 4D tensors at /opt/conda/conda-
  bld/pytorch_1544174967633/work/aten/src/TH/generic/THTensorMath.cpp:935
10
11 In [31]: torch.matmul(a,b).shape
12 Out[31]: torch.Size([4, 3, 28, 32])
13
14 In [32]: b=torch.rand(4,1,64,32)
15 In [33]: torch.matmul(a,b).shape
16 Out[33]: torch.Size([4, 3, 28, 32])
17
18 In [34]: b=torch.rand(4,64,32)
19 In [35]: torch.matmul(a,b).shape
20 -----
21 RuntimeError                                Traceback (most recent call last)
22 <ipython-input-35-9aa246c961de> in <module>()
23 ----> 1 torch.matmul(a,b).shape
24 RuntimeError: The size of tensor a (3) must match the size of tensor b (4) at non-
  singleton dimension 1
```

Power

```
1 In [9]: a=torch.full([2,2],3)
2 In [11]: a.pow(2)
3 tensor([[9., 9.],
4         [9., 9.]])
5
6 In [12]: a**2
7 tensor([[9., 9.],
8         [9., 9.]])
9
10 In [13]: aa=a**2
11 In [14]: aa.sqrt()
12 tensor([[3., 3.],
13         [3., 3.]])
14
15 In [15]: aa.rsqrt()
16 tensor([[0.3333, 0.3333],
17         [0.3333, 0.3333]])
18
19 In [16]: aa**(0.5)
20 tensor([[3., 3.],
21         [3., 3.]])
```

Exp log



```
1 In [116]: a=torch.exp(torch.ones(2,2))
2
3 In [117]: a
4 Out[117]:
5 tensor([[2.7183, 2.7183],
6         [2.7183, 2.7183]])
7
8 In [118]: torch.log(a)
9 Out[118]:
10 tensor([[1., 1.],
11         [1., 1.]])
```

Approximation

- `.floor()` `.ceil()`
- `.round()`
- `.trunc()` `.frac()`

```
1 In [121]: a=torch.tensor(3.14)
2
3 In [124]: a.floor(),a.ceil(),a.trunc(),a.frac()
4 Out[124]: (tensor(3.), tensor(4.), tensor(3.), tensor(0.1400))
5
6 In [125]: a=torch.tensor(3.499)
7
8 In [126]: a.round()
9 Out[126]: tensor(3.)
10
11 In [127]: a=torch.tensor(3.5)
12
13 In [128]: a.round()
14 Out[128]: tensor(4.)
```

clamp

- gradient clipping
 - (min)
 - (min, max)
-

```
In [2]: grad=torch.rand(2,3)*15
```

```
In [3]: grad.max()  
Out[3]: tensor(14.8737)
```

```
In [4]: grad.median()  
Out[4]: tensor(10.1571)
```

```
In [5]: grad.clamp(10)  
Out[5]:  
tensor([[14.8737, 10.1571, 10.0000],  
        [11.3591, 10.0000, 14.0524]])
```

```
In [6]: grad  
Out[6]:  
tensor([[14.8737, 10.1571,  4.4872],  
        [11.3591,  8.9101, 14.0524]])
```

```
In [7]: grad.clamp(0, 10)  
Out[7]:  
tensor([[10.0000, 10.0000,  4.4872],  
        [10.0000,  8.9101, 10.0000]])
```

Thank You.
