TORCH.NORM

`torch.norm(input, p='fro', dim=None, keepdim=False, out=None, dtype=None)` [SO RCE]

Returns the matrix norm or vector norm of a given tensor.

• WARNING

torch.norm is eprecate an may e remove in a future PyTorch release. se `torch.linalg.norm()` instea , ut note that `torch.linalg.norm()` has a ifferent signature an slightly ifferent ehavior that is more consistent with NumPy's numpy.linalg.norm.

Parameters

- **in ut** (*Tensor*) – the input tensor
- (*int, float, inf, -inf, 'fro', 'nuc', optional*) –

  the or er of norm. Default: `'fro'` The following norms can e calculate :

or	matrix norm	vector norm
None	Fro enius norm	2-norm
'fro'	Fro enius norm	–
'nuc'	nuclear norm	–
Other	as vec norm when im is None	sum(a s(x)**or)**(1./or)

- **im** (*int, 2-tuple of python:ints, 2-list of python:ints, optional*) – If it is an int, vector norm will e calculate , if it is 2-tuple of ints, matrix norm will e calculate . If the value is None, matrix norm will e calculate when the input tensor only has two imensions, vector norm will e calculate when the input tensor only has one imension. If the input tensor has more than two imensions, the vector norm will e applie to last imension.
- **kee   im** (*bool, optional*) – whether the output tensors have `dim` retaine or not. Ignore if `dim = None` an `out = None`. Default: `False`
- **out** (*Tensor, optional*) – the output tensor. Ignore if `dim = None` an `out = None`.
- **ty e** (`torch.dtype`, optional) – the esire   ata type of returne tensor. If specifie , the input tensor is caste to :attr:' type' while performing the operation. Default: None.

Example:

```
>>> import torch
>>> a = torch.arange(9, dtype= torch.float) - 4
>>> b = a.reshape((3, 3))
>>> torch.norm(a)
tensor(7.7460)
>>> torch.norm(b)
tensor(7.7460)
>>> torch.norm(a, float('inf'))
tensor(4.)
>>> torch.norm(b, float('inf'))
tensor(4.)
>>> c = torch.tensor([[ 1, 2, 3],[-1, 1, 4]] , dtype= torch.float)
>>> torch.norm(c, dim=0)
tensor([1.4142, 2.2361, 5.0000])
>>> torch.norm(c, dim=1)
tensor([3.7417, 4.2426])
>>> torch.norm(c, p=1, dim=1)
tensor([6., 6.])
>>> d = torch.arange(8, dtype= torch.float).reshape(2,2,2)
>>> torch.norm(d, dim=(1,2))
tensor([ 3.7417, 11.2250])
>>> torch.norm(d[0, :, :]), torch.norm(d[1, :, :])
(tensor(3.7417), tensor(11.2250))
```

Previous Next

Access com rehensive evelo er ocumentation for PyTorch

View Docs

Get in- e th tutorials for eginners an a vance evelo ers

View Tutorials

Fin evelo ment resources an get your uestions answere

View Resources