# tf.norm

## Aliases:

- `tf.linalg.norm`
- `tf.norm`

```
norm(
    tensor,
    ord='euclidean',
    axis=None,
    keep_dims=False,
    name=None
)
```

Defined in `tensorflow/python/ops/linalg_ops.py`.

See the guide: Math > Matrix Math Functions

Computes the norm of vectors, matrices, and tensors.

This function can compute several different vector norms (the 1-norm, the Euclidean or 2-norm, the inf-norm, and in general the p-norm for p > 0) and matrix norms (Frobenius, 1-norm, and inf-norm).

## Args:

- `tensor`: `Tensor` of types `float32`, `float64`, `complex64`, `complex128`
- `ord`: Order of the norm. Supported values are 'fro', 'euclidean', `1`, `2`, `np.inf` and any positive real number yielding the corresponding p-norm. Default is 'euclidean' which is equivalent to Frobenius norm if `tensor` is a matrix and equivalent to 2-norm for vectors. Some restrictions apply: a) The Frobenius norm `fro` is not defined for vectors, b) If axis is a 2-tuple (matrix norm), only 'euclidean', 'fro', `1`, `np.inf` are supported. See the description of `axis` on how to compute norms for a batch of vectors or matrices stored in a tensor.
- `axis`: If `axis` is `None` (the default), the input is considered a vector and a single vector norm is computed over the entire set of values in the tensor, i.e. `norm(tensor, ord=ord)` is equivalent to `norm(reshape(tensor, [-1]), ord=ord)`. If `axis` is a Python integer, the input is considered a batch of vectors, and `axis` determines the axis in `tensor` over which to compute vector norms. If `axis` is a 2-tuple of Python integers it is considered a batch of matrices and `axis` determines the axes in `tensor` over which to compute a matrix norm. Negative indices are supported. Example: If you are passing a tensor that can be either a matrix or a batch of matrices at runtime, pass `axis=[-2,-1]` instead of `axis=None` to make sure that matrix norms are computed.
- `keep_dims`: If True, the axis indicated in `axis` are kept with size 1. Otherwise, the dimensions in `axis` are removed from the output shape.
- `name`: The name of the op.

## Returns:

- `output`: A `Tensor` of the same type as tensor, containing the vector or matrix norms. If `keep_dims` is True then the rank of output is equal to the rank of `tensor`. Otherwise, if `axis` is none the output is a scalar, if `axis` is an integer, the rank of `output` is one less than the rank of `tensor`, if `axis` is a 2-tuple the rank of `output` is two less than the rank of `tensor`.

#### Raises:

- `ValueError`: If `ord` or `axis` is invalid.

### numpy compatibility

Mostly equivalent to numpy.linalg.norm. Not supported: ord <= 0, 2-norm for matrices, nuclear norm. Other differences: a) If axis is `None`, treats the flattened `tensor` as a vector regardless of rank. b) Explicitly supports 'euclidean' norm as the default, including for higher order tensors.

*Last updated November 2, 2017.*

**Stay Connected**

Blog

GitHub

Twitter

**Support**

Issue Tracker

Release Notes

Stack Overflow

**English**

**Terms** | **Privacy**