

tf.sparse_add

```
sparse_add(  
    a,  
    b,  
    thresh=0  
)
```

Defined in [tensorflow/python/ops/sparse_ops.py](#).

See the guide: [Sparse Tensors > Math Operations](#)

Adds two tensors, at least one of each is a **SparseTensor**.

If one **SparseTensor** and one **Tensor** are passed in, returns a **Tensor**. If both arguments are **SparseTensor**s, this returns a **SparseTensor**. The order of arguments does not matter. Use vanilla **tf.add()** for adding two dense **Tensor**s.

The shapes of the two operands must match: broadcasting is not supported.

The indices of any input **SparseTensor** are assumed ordered in standard lexicographic order. If this is not the case, before this step run **SparseReorder** to restore index ordering.

If both arguments are sparse, we perform "clipping" as follows. By default, if two values sum to zero at some index, the output **SparseTensor** would still include that particular location in its index, storing a zero in the corresponding value slot. To override this, callers can specify **thresh**, indicating that if the sum has a magnitude strictly smaller than **thresh**, its corresponding value and index would then not be included. In particular, **thresh == 0.0** (default) means everything is kept and actual thresholding happens only for a positive value.

For example, suppose the logical sum of two sparse operands is (densified):

```
[ 2]  
[.1 0]  
[ 6 -.2]
```

Then,

```
* `thresh == 0` (the default): all 5 index/value pairs will be returned.  
* `thresh == 0.11`: only .1 and 0 will vanish, and the remaining three  
  index/value pairs will be returned.  
* `thresh == 0.21`: .1, 0, and -.2 will vanish.
```

Args:

- **a**: The first operand; **SparseTensor** or **Tensor**.
- **b**: The second operand; **SparseTensor** or **Tensor**. At least one operand must be sparse.
- **thresh**: A 0-D **Tensor**. The magnitude threshold that determines if an output value/index pair takes space. Its dtype should match that of the values if they are real; if the latter are complex64/complex128, then the dtype should be float32/float64, correspondingly.

Returns:

A `SparseTensor` or a `Tensor`, representing the sum.

Raises:

- `TypeError`: If both `a` and `b` are `Tensor`s. Use `tf.add()` instead.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)