

tf.sparse_softmax

```
sparse_softmax(  
    sp_input,  
    name=None  
)
```

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

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

Applies softmax to a batched N-D **SparseTensor**.

The inputs represent an N-D SparseTensor with logical shape `[..., B, C]` (where `N >= 2`), and with indices sorted in the canonical lexicographic order.

This op is equivalent to applying the normal `tf.nn.softmax()` to each innermost logical submatrix with shape `[B, C]`, but with the catch that *the implicitly zero elements do not participate*. Specifically, the algorithm is equivalent to:

(1) Applies `tf.nn.softmax()` to a densified view of each innermost submatrix with shape `[B, C]`, along the size-C dimension; (2) Masks out the original implicitly-zero locations; (3) Renormalizes the remaining elements.

Hence, the **SparseTensor** result has exactly the same non-zero indices and shape.

Example:

```
# First batch:  
# [ ?  e.]  
# [1.  ? ]  
# Second batch:  
# [e  ? ]  
# [e  e ]  
shape = [2, 2, 2] # 3-D SparseTensor  
values = np.asarray([[[0., np.e], [1., 0.]], [[np.e, 0.], [np.e, np.e]]])  
indices = np.vstack(np.where(values)).astype(np.int64).T  
  
result = tf.sparse_softmax(tf.SparseTensor(indices, values, shape))  
# ...returning a 3-D SparseTensor, equivalent to:  
# [ ?  1.]    [1  ?]  
# [1.  ? ] and [.5 .5]  
# where ? means implicitly zero.
```

Args:

- `sp_input`: N-D **SparseTensor**, where `N >= 2`.
- `name`: optional name of the operation.

Returns:

- `output`: N-D **SparseTensor** representing the results.

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