

tf.contrib.framework.nest.flatten_up_to

```
flatten_up_to(  
    shallow_tree,  
    input_tree  
)
```

Defined in [tensorflow/python/util/nest.py](#).

Flattens `input_tree` up to `shallow_tree`.

Any further depth in structure in `input_tree` is retained as elements in the partially flatten output.

If `shallow_tree` and `input_tree` are not sequences, this returns a single-element list: `[input_tree]`.

Use Case:

Sometimes we may wish to partially flatten a nested sequence, retaining some of the nested structure. We achieve this by specifying a shallow structure, `shallow_tree`, we wish to flatten up to.

The input, `input_tree`, can be thought of as having the same structure as `shallow_tree`, but with leaf nodes that are themselves tree structures.

Examples:

```
input_tree = [[[2, 2], [3, 3]], [[4, 9], [5, 5]]]  
shallow_tree = [[True, True], [False, True]]  
  
flattened_input_tree = flatten_up_to(shallow_tree, input_tree)  
flattened_shallow_tree = flatten_up_to(shallow_tree, shallow_tree)  
  
# Output is:  
# [[2, 2], [3, 3], [4, 9], [5, 5]]  
# [True, True, False, True]
```

```
input_tree = [[('a', 1), (('b', 2), (('c', 3), (('d', 4)))]]  
shallow_tree = [['level_1', ['level_2', ['level_3', ['level_4']]]]]  
  
input_tree_flattened_as_shallow_tree = flatten_up_to(shallow_tree, input_tree)  
input_tree_flattened = flatten(input_tree)  
  
# Output is:  
# [('a', 1), ('b', 2), ('c', 3), ('d', 4)]  
# ['a', 1, 'b', 2, 'c', 3, 'd', 4]
```

Non-Sequence Edge Cases:

```
flatten_up_to(0, 0) # Output: [0]  
flatten_up_to(0, [0, 1, 2]) # Output: [[0, 1, 2]]  
flatten_up_to([0, 1, 2], 0) # Output: TypeError  
flatten_up_to([0, 1, 2], [0, 1, 2]) # Output: [0, 1, 2]
```

Args:

- `shallow_tree` : a possibly pruned structure of `input_tree`.
- `input_tree` : an arbitrarily nested structure or a scalar object. Note, numpy arrays are considered scalars.

Returns:

A Python list, the partially flattened version of `input_tree` according to the structure of `shallow_tree`.

Raises:

- `TypeError` : If `shallow_tree` is a sequence but `input_tree` is not.
- `TypeError` : If the sequence types of `shallow_tree` are different from `input_tree`.
- `ValueError` : If the sequence lengths of `shallow_tree` are different from `input_tree`.

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](#)