Module: tf.nest

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest#top_of_page)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest#functions)

Public API for tf.nest namespace.

Functions

[assert\_same\_structure(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/assert_same_structure): Asserts that two structures are nested in the same way.

[flatten(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/flatten): Returns a flat list from a given nested structure.

[is\_nested(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/is_nested): Returns true if its input is a collections.Sequence (except strings).

[map\_structure(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/map_structure): Applies func to each entry in structure and returns a new structure.

[pack\_sequence\_as(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/pack_sequence_as): Returns a given flattened sequence packed into a given structure.

# tf.nest.assert\_same\_structure

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/assert_same_structure#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/assert_same_structure#aliases)

Asserts that two structures are nested in the same way.

### Aliases:

* tf.compat.v1.nest.assert\_same\_structure
* tf.compat.v2.nest.assert\_same\_structure
* tf.nest.assert\_same\_structure

tf.nest.assert\_same\_structure(  
    nest1,  
    nest2,  
    check\_types=True,  
    expand\_composites=False  
)

Defined in [python/util/nest.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/util/nest.py).

Note that namedtuples with identical name and fields are always considered to have the same shallow structure (even with check\_types=True). For instance, this code will print True:

def nt(a, b):  
  return collections.namedtuple('foo', 'a b')(a, b)  
print(assert\_same\_structure(nt(0, 1), nt(2, 3)))

#### Args:

* **nest1**: an arbitrarily nested structure.
* **nest2**: an arbitrarily nested structure.
* **check\_types**: if True (default) types of sequences are checked as well, including the keys of dictionaries. If set to False, for example a list and a tuple of objects will look the same if they have the same size. Note that namedtuples with identical name and fields are always considered to have the same shallow structure. Two types will also be considered the same if they are both list subtypes (which allows "list" and "\_ListWrapper" from trackable dependency tracking to compare equal).
* **expand\_composites**: If true, then composite tensors such as [tf.SparseTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/sparse/SparseTensor) and [tf.RaggedTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/RaggedTensor) are expanded into their component tensors.

#### Raises:

* **ValueError**: If the two structures do not have the same number of elements or if the two structures are not nested in the same way.
* **TypeError**: If the two structures differ in the type of sequence in any of their substructures. Only possible if check\_types is True.

# tf.nest.flatten

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/flatten#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/flatten#aliases)

Returns a flat list from a given nested structure.

### Aliases:

* tf.compat.v1.nest.flatten
* tf.compat.v2.nest.flatten
* tf.nest.flatten

tf.nest.flatten(  
    structure,  
    expand\_composites=False  
)

Defined in [python/util/nest.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/util/nest.py).

If nest is not a sequence, tuple, or dict, then returns a single-element list: [nest].

In the case of dict instances, the sequence consists of the values, sorted by key to ensure deterministic behavior. This is true also for OrderedDict instances: their sequence order is ignored, the sorting order of keys is used instead. The same convention is followed in pack\_sequence\_as. This correctly repacks dicts and OrderedDicts after they have been flattened, and also allows flattening an OrderedDict and then repacking it back using a corresponding plain dict, or vice-versa. Dictionaries with non-sortable keys cannot be flattened.

Users must not modify any collections used in nest while this function is running.

#### Args:

* **structure**: an arbitrarily nested structure or a scalar object. Note, numpy arrays are considered scalars.
* **expand\_composites**: If true, then composite tensors such as tf.SparseTensor and tf.RaggedTensor are expanded into their component tensors.

#### Returns:

A Python list, the flattened version of the input.

#### Raises:

* **TypeError**: The nest is or contains a dict with non-sortable keys.

# tf.nest.is\_nested

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/is_nested#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/is_nested#aliases)

Returns true if its input is a collections.Sequence (except strings).

### Aliases:

* tf.compat.v1.nest.is\_nested
* tf.compat.v2.nest.is\_nested
* tf.nest.is\_nested

tf.nest.is\_nested(seq)

Defined in [python/util/nest.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/util/nest.py).

#### Args:

* **seq**: an input sequence.

#### Returns:

True if the sequence is a not a string and is a collections.Sequence or a dict.

# tf.nest.map\_structure

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/map_structure#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/map_structure#aliases)

Applies func to each entry in structure and returns a new structure.

### Aliases:

* tf.compat.v1.nest.map\_structure
* tf.compat.v2.nest.map\_structure
* tf.nest.map\_structure

tf.nest.map\_structure(  
    func,  
    \*structure,  
    \*\*kwargs  
)

Defined in [python/util/nest.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/util/nest.py).

Applies func(x[0], x[1], ...) where x[i] is an entry in structure[i]. All structures in structure must have the same arity, and the return value will contain results with the same structure layout.

#### Args:

* **func**: A callable that accepts as many arguments as there are structures.
* **\*structure**: scalar, or tuple or list of constructed scalars and/or other tuples/lists, or scalars. Note: numpy arrays are considered as scalars.
* **\*\*kwargs**: Valid keyword args are:
  + check\_types: If set to True (default) the types of iterables within the structures have to be same (e.g. map\_structure(func, [1], (1,)) raises a TypeError exception). To allow this set this argument to False. Note that namedtuples with identical name and fields are always considered to have the same shallow structure.
  + expand\_composites: If set to True, then composite tensors such as [tf.SparseTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/sparse/SparseTensor)and [tf.RaggedTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/RaggedTensor) are expanded into their component tensors. If False (the default), then composite tensors are not expanded.

#### Returns:

A new structure with the same arity as structure, whose values correspond to func(x[0], x[1], ...) where x[i] is a value in the corresponding location in structure[i]. If there are different sequence types and check\_types is False the sequence types of the first structure will be used.

#### Raises:

* **TypeError**: If func is not callable or if the structures do not match each other by depth tree.
* **ValueError**: If no structure is provided or if the structures do not match each other by type.
* **ValueError**: If wrong keyword arguments are provided.

# tf.nest.pack\_sequence\_as

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/pack_sequence_as#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/nest/pack_sequence_as#aliases)

Returns a given flattened sequence packed into a given structure.

### Aliases:

* tf.compat.v1.nest.pack\_sequence\_as
* tf.compat.v2.nest.pack\_sequence\_as
* tf.nest.pack\_sequence\_as

tf.nest.pack\_sequence\_as(  
    structure,  
    flat\_sequence,  
    expand\_composites=False  
)

Defined in [python/util/nest.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/util/nest.py).

If structure is a scalar, flat\_sequence must be a single-element list; in this case the return value is flat\_sequence[0].

If structure is or contains a dict instance, the keys will be sorted to pack the flat sequence in deterministic order. This is true also for OrderedDict instances: their sequence order is ignored, the sorting order of keys is used instead. The same convention is followed in flatten. This correctly repacks dicts and OrderedDicts after they have been flattened, and also allows flattening an OrderedDict and then repacking it back using a corresponding plain dict, or vice-versa. Dictionaries with non-sortable keys cannot be flattened.

#### Args:

* **structure**: Nested structure, whose structure is given by nested lists, tuples, and dicts. Note: numpy arrays and strings are considered scalars.
* **flat\_sequence**: flat sequence to pack.
* **expand\_composites**: If true, then composite tensors such as [tf.SparseTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/sparse/SparseTensor) and [tf.RaggedTensor](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/RaggedTensor) are expanded into their component tensors.

#### Returns:

* **packed**: flat\_sequence converted to have the same recursive structure as structure.

#### Raises:

* **ValueError**: If flat\_sequence and structure have different element counts.
* **TypeError**: structure is or contains a dict with non-sortable keys.