tf.parallel_stack

```
parallel_stack(
    values,
    name='parallel_stack'
)
```

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

See the guide: Tensor Transformations > Slicing and Joining

Stacks a list of rank-`R` tensors into one rank-`(R+1)` tensor in parallel.

Requires that the shape of inputs be known at graph construction time.

Packs the list of tensors in `values` into a tensor with rank one higher than each tensor in `values`, by packing them along the first dimension. Given a list of length `N` of tensors of shape `(A, B, C)`; the `output` tensor will have the shape `(N, A, B, C)`.

For example:

```
x = tf.constant([1, 4])
y = tf.constant([2, 5])
z = tf.constant([3, 6])
tf.parallel_stack([x, y, z])  # [[1, 4], [2, 5], [3, 6]]
```

The difference between `stack` and `parallel_stack` is that `stack` requires all the inputs be computed before the operation will begin but doesn't require that the input shapes be known during graph construction.

`parallel_stack` will copy pieces of the input into the output as they become available, in some situations this can provide a performance benefit.

Unlike `stack`, `parallel_stack` does NOT support backpropagation.

This is the opposite of unstack. The numpy equivalent is

```
tf.parallel_stack([x, y, z]) = np.asarray([x, y, z])
```

Args:

- `values`: A list of `Tensor` objects with the same shape and type.
- `name`: A name for this operation (optional).

Returns:

- `output`: A stacked `Tensor` with the same type as `values`.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**