tf.unstack

```
unstack(
    value,
    num=None,
    axis=0,
    name='unstack'
)
```

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

See the guide: Tensor Transformations > Slicing and Joining

Unpacks the given dimension of a rank-`R` tensor into rank-`(R-1)` tensors.

Unpacks `num` tensors from `value` by chipping it along the `axis` dimension. If `num` is not specified (the default), it is inferred from `value`'s shape. If `value.shape[axis]` is not known, `ValueError` is raised.

For example, given a tensor of shape `(A, B, C, D)`;

If `axis == 0` then the i'th tensor in `output` is the slice `value[i, :, :, :]` and each tensor in `output` will have shape `(B, C, D)`. (Note that the dimension unpacked along is gone, unlike `split`).

If `axis == 1` then the i'th tensor in `output` is the slice `value[:, i, :, :]` and each tensor in `output` will have shape `(A, C, D)`. Etc.

This is the opposite of stack. The numpy equivalent is

```
tf.unstack(x, n) = np.unstack(x)
```

Args:

- `value`: A rank `R > 0` `Tensor` to be unstacked.
- `num`: An `int`. The length of the dimension `axis`. Automatically inferred if `None` (the default).
- `axis`: An `int`. The axis to unstack along. Defaults to the first dimension. Negative values wrap around, so the valid range is `[-R, R)`.
- `name`: A name for the operation (optional).

Returns:

The list of `Tensor` objects unstacked from `value`.

Raises:

- `ValueError`: If `num` is unspecified and cannot be inferred.
- `ValueError`: If `axis` is out of the range [-R, R).

Last updated November 2, 2017.