tf.slice

```
slice(
    input_,
    begin,
    size,
    name=None
)
```

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

See the guide: Tensor Transformations > Slicing and Joining

Extracts a slice from a tensor.

This operation extracts a slice of size `size` from a tensor `input` starting at the location specified by `begin`. The slice `size` is represented as a tensor shape, where `size[i]` is the number of elements of the 'i'th dimension of `input` that you want to slice. The starting location (`begin`) for the slice is represented as an offset in each dimension of `input`. In other words, `begin[i]` is the offset into the 'i'th dimension of `input` that you want to slice from.

Note that `tf.Tensor.__getitem__` is typically a more pythonic way to perform slices, as it allows you to write `foo[3:7, :-2]` instead of `tf.slice([3, 0], [4, foo.get_shape()[1]-2])`.

`begin` is zero-based; `size` is one-based. If `size[i]` is -1, all remaining elements in dimension i are included in the slice. In other words, this is equivalent to setting:

`size[i] = input.dim_size(i) - begin[i]`

This operation requires that:

`0 <= begin[i] <= begin[i] + size[i] <= Di for i in [0, n]`

For example:

```
t = tf.constant([[[1, 1, 1], [2, 2, 2]],
                 [[3, 3, 3], [4, 4, 4]],
                 [[5, 5, 5], [6, 6, 6]]])
tf.slice(t, [1, 0, 0], [1, 1, 3])  # [[[3, 3, 3]]]
tf.slice(t, [1, 0, 0], [1, 2, 3])  # [[[3, 3, 3],
                                   #   [4, 4, 4]]]
tf.slice(t, [1, 0, 0], [2, 1, 3])  # [[[3, 3, 3]],
                                   #  [[5, 5, 5]]]
```

Args:

- `input_`: A `Tensor`.
- `begin`: An `int32` or `int64` `Tensor`.
- `size`: An `int32` or `int64` `Tensor`.
- `name`: A name for the operation (optional).

Returns:

A `Tensor` the same type as `input`.

Last updated November 2, 2017.