tf.gather_nd

```
gather_nd(
    params,
    indices,
    name=None
)
```

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

See the guide: Tensor Transformations > Slicing and Joining

Gather slices from `params` into a Tensor with shape specified by `indices`.

`indices` is an K-dimensional integer tensor, best thought of as a (K-1)-dimensional tensor of indices into `params`, where each element defines a slice of `params`:

```
output[i_0, ..., i_{K-2}] = params[indices[i0, ..., i_{K-2}]]
```

Whereas in `tf.gather` `indices` defines slices into the first dimension of `params`, in `tf.gather_nd`, `indices` defines slices into the first `N` dimensions of `params`, where `N = indices.shape[-1]`.

The last dimension of `indices` can be at most the rank of `params`:

```
indices.shape[-1] <= params.rank
```

The last dimension of `indices` corresponds to elements (if `indices.shape[-1] == params.rank`) or slices (if `indices.shape[-1] < params.rank`) along dimension `indices.shape[-1]` of `params`. The output tensor has shape

```
indices.shape[:-1] + params.shape[indices.shape[-1]:]
```

Some examples below.

Simple indexing into a matrix:

```
    indices = [[0, 0], [1, 1]]
    params = [['a', 'b'], ['c', 'd']]
    output = ['a', 'd']
```

Slice indexing into a matrix:

```
    indices = [[1], [0]]
    params = [['a', 'b'], ['c', 'd']]
    output = [['c', 'd'], ['a', 'b']]
```

Indexing into a 3-tensor:

```
    indices = [[1]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = [[['a1', 'b1'], ['c1', 'd1']]]

    indices = [[0, 1], [1, 0]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = [['c0', 'd0'], ['a1', 'b1']]

    indices = [[0, 0, 1], [1, 0, 1]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = ['b0', 'b1']
```

Batched indexing into a matrix:

```
    indices = [[[0, 0]], [[0, 1]]]
    params = [['a', 'b'], ['c', 'd']]
    output = [['a'], ['b']]
```

Batched slice indexing into a matrix:

```
    indices = [[[1]], [[0]]]
    params = [['a', 'b'], ['c', 'd']]
    output = [[['c', 'd']], [['a', 'b']]]
```

Batched indexing into a 3-tensor:

```
    indices = [[[1]], [[0]]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = [[[['a1', 'b1'], ['c1', 'd1']]],
              [[['a0', 'b0'], ['c0', 'd0']]]]

    indices = [[[0, 1], [1, 0]], [[0, 0], [1, 1]]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = [[['c0', 'd0'], ['a1', 'b1']],
              [['a0', 'b0'], ['c1', 'd1']]]

    indices = [[[0, 0, 1], [1, 0, 1]], [[0, 1, 1], [1, 1, 0]]]
    params = [[['a0', 'b0'], ['c0', 'd0']],
              [['a1', 'b1'], ['c1', 'd1']]]
    output = [['b0', 'b1'], ['d0', 'c1']]
```

Args:

- `params`: A `Tensor`. The tensor from which to gather values.
- `indices`: A `Tensor`. Must be one of the following types: `int32`, `int64`. Index tensor.
- `name`: A name for the operation (optional).

Returns:

A `Tensor`. Has the same type as `params`. Values from `params` gathered from indices given by `indices`, with shape `indices.shape[:-1] + params.shape[indices.shape[-1]:]`.

Last updated November 2, 2017.