tf.nn.embedding_lookup

```
embedding_lookup(
    params,
    ids,
    partition_strategy='mod',
    name=None,
    validate_indices=True,
    max_norm=None
)
```

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

See the guide: Neural Network > Embeddings

Looks up `ids` in a list of embedding tensors.

This function is used to perform parallel lookups on the list of tensors in `params`. It is a generalization of `tf.gather`, where `params` is interpreted as a partitioning of a large embedding tensor. `params` may be a `PartitionedVariable` as returned by using `tf.get_variable()` with a partitioner.

If `len(params) > 1`, each element `id` of `ids` is partitioned between the elements of `params` according to the `partition_strategy`. In all strategies, if the id space does not evenly divide the number of partitions, each of the first `(max_id + 1) % len(params)` partitions will be assigned one more id.

If `partition_strategy` is `"mod"`, we assign each id to partition `p = id % len(params)`. For instance, 13 ids are split across 5 partitions as: `[[0, 5, 10], [1, 6, 11], [2, 7, 12], [3, 8], [4, 9]]`

If `partition_strategy` is `"div"`, we assign ids to partitions in a contiguous manner. In this case, 13 ids are split across 5 partitions as: `[[0, 1, 2], [3, 4, 5], [6, 7, 8], [9, 10], [11, 12]]`

The results of the lookup are concatenated into a dense tensor. The returned tensor has shape `shape(ids) + shape(params)[1:]`.

Args:

- `params`: A single tensor representing the complete embedding tensor, or a list of P tensors all of same shape except for the first dimension, representing sharded embedding tensors. Alternatively, a `PartitionedVariable`, created by partitioning along dimension 0. Each element must be appropriately sized for the given `partition_strategy`.
- `ids`: A `Tensor` with type `int32` or `int64` containing the ids to be looked up in `params`.
- `partition_strategy`: A string specifying the partitioning strategy, relevant if `len(params) > 1`. Currently `"div"` and `"mod"` are supported. Default is `"mod"`.
- `name`: A name for the operation (optional).
- `validate_indices`: DEPRECATED. If this operation is assigned to CPU, values in `indices` are always validated to be within range. If assigned to GPU, out-of-bound indices result in safe but unspecified behavior, which may include raising an error.
- `max_norm`: If provided, embedding values are l2-normalized to the value of max_norm.

Returns:

A `Tensor` with the same type as the tensors in `params`.

Raises:

- `ValueError`: If `params` is empty.

Last updated November 2, 2017.