tf.contrib.tpu.batch_parallel

```
batch_parallel(
    computation,
    inputs=None,
    num_shards=1,
    infeed_queue=None,
    global_tpu_id=None,
    name=None
)
```

Defined in `tensorflow/contrib/tpu/python/tpu/tpu.py`.

Shards `computation` along the batch dimension for parallel execution.

Convenience wrapper around shard().

`inputs` must be a list of Tensors or None (equivalent to an empty list). Each input is split into `num_shards` pieces along the 0-th dimension, and computation is applied to each shard in parallel.

Tensors are broadcast to all shards if they are lexically captured by `computation`. e.g.,

x = tf.constant(7) def computation(): return x + 3 ... = shard(computation, ...)

The outputs from all shards are concatenated back together along their 0-th dimension.

Inputs and outputs of the computation must be at least rank-1 Tensors.

Args:

- `computation`: a Python function that builds a computation to apply to each shard of the input.
- `inputs`: a list of input tensors or None (equivalent to an empty list). The 0-th dimension of each Tensor must have size divisible by `num_shards`.
- `num_shards`: the number of shards.
- `infeed_queue`: if not None, the InfeedQueue from which to append a tuple of arguments as inputs to `computation`.
- `global_tpu_id`: if not None, a Numpy 2D array indicating the global id of each TPU device in the system. The outer dimension of the array is host task id, and the inner dimension is device ordinal, so e.g., global_tpu_id[x][y] indicates the global id of device /task:x/device:TPU_NODE:y.
- `name`: name of the operator.

Returns:

A list of output tensors.

Raises:

- `ValueError`: if num_shards <= 0

Last updated November 2, 2017.