

## tf.contrib.training.bucket\_by\_sequence\_length

```
bucket_by_sequence_length(  
    input_length,  
    tensors,  
    batch_size,  
    bucket_boundaries,  
    num_threads=1,  
    capacity=32,  
    bucket_capacities=None,  
    shapes=None,  
    dynamic_pad=False,  
    allow_smaller_final_batch=False,  
    keep_input=True,  
    shared_name=None,  
    name=None  
)
```

Defined in [tensorflow/contrib/training/python/training/bucket\\_ops.py](#).

See the guide: [Training \(contrib\) > Bucketing](#)

Lazy bucketing of inputs according to their length.

This method calls `tf.contrib.training.bucket` under the hood, after first subdividing the bucket boundaries into separate buckets and identifying which bucket the given `input_length` belongs to. See the documentation for `which_bucket` for details of the other arguments.

### Args:

- `input_length`: `int32` scalar `Tensor`, the sequence length of tensors.
- `tensors`: The list or dictionary of tensors, representing a single element, to bucket. Nested lists are not supported.
- `batch_size`: The new batch size pulled from the queue (all queues will have the same size). If a list is passed in then each bucket will have a different `batch_size`. (python int, `int32` scalar or iterable of integers of length `num_buckets`).
- `bucket_boundaries`: int list, increasing non-negative numbers. The edges of the buckets to use when bucketing tensors. Two extra buckets are created, one for `input_length < bucket_boundaries[0]` and one for `input_length >= bucket_boundaries[-1]`.
- `num_threads`: An integer. The number of threads enqueueing `tensors`.
- `capacity`: An integer. The maximum number of minibatches in the top queue, and also the maximum number of elements within each bucket.
- `bucket_capacities`: (Optional) None or a list of integers, the capacities of each bucket. If None, capacity is used (default). If specified, it must be a list of integers of length one larger than `bucket_boundaries`. Its *i*-th element is used as capacity for the *i*-th bucket queue.
- `shapes`: (Optional) The shapes for each example. Defaults to the inferred shapes for `tensors`.
- `dynamic_pad`: Boolean. Allow variable dimensions in input shapes. The given dimensions are padded upon dequeue so that tensors within a batch have the same shapes.
- `allow_smaller_final_batch`: (Optional) Boolean. If `True`, allow the final batches to be smaller if there are insufficient items left in the queues.

- `keep_input` : A `bool` scalar Tensor. If provided, this tensor controls whether the input is added to the queue or not. If it evaluates `True`, then `tensors` are added to the bucket; otherwise they are dropped. This tensor essentially acts as a filtering mechanism.
- `shared_name` : (Optional). If set, the queues will be shared under the given name across multiple sessions.
- `name` : (Optional) A name for the operations.

Returns:

A tuple `(sequence_length, outputs)` where `sequence_length` is a 1-D `Tensor` of size `batch_size` and `outputs` is a list or dictionary of batched, bucketed, outputs corresponding to elements of `tensors`.

Raises:

- `TypeError` : if `bucket_boundaries` is not a list of python integers.
- `ValueError` : if `bucket_boundaries` is empty or contains non-increasing values or if `batch_size` is a list and it's length doesn't equal the number of buckets.

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

## Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

## Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)