tf.contrib.legacy_seq2seq.model_with_buckets

```
model_with_buckets(
    encoder_inputs,
    decoder_inputs,
    targets,
    weights,
    buckets,
    seq2seq,
    softmax_loss_function=None,
    per_example_loss=False,
    name=None
)
```

Defined in `tensorflow/contrib/legacy_seq2seq/python/ops/seq2seq.py`.

Create a sequence-to-sequence model with support for bucketing.

The seq2seq argument is a function that defines a sequence-to-sequence model, e.g., seq2seq = lambda x, y: basic_rnn_seq2seq(x, y, rnn_cell.GRUCell(24))

Args:

- `encoder_inputs`: A list of Tensors to feed the encoder; first seq2seq input.
- `decoder_inputs`: A list of Tensors to feed the decoder; second seq2seq input.
- `targets`: A list of 1D batch-sized int32 Tensors (desired output sequence).
- `weights`: List of 1D batch-sized float-Tensors to weight the targets.
- `buckets`: A list of pairs of (input size, output size) for each bucket.
- `seq2seq`: A sequence-to-sequence model function; it takes 2 input that agree with encoder_inputs and decoder_inputs, and returns a pair consisting of outputs and states (as, e.g., basic_rnn_seq2seq).
- `softmax_loss_function`: Function (labels, logits) -> loss-batch to be used instead of the standard softmax (the default if this is None). **Note that to avoid confusion, it is required for the function to accept named arguments.**
- `per_example_loss`: Boolean. If set, the returned loss will be a batch-sized tensor of losses for each sequence in the batch. If unset, it will be a scalar with the averaged loss from all examples.
- `name`: Optional name for this operation, defaults to "model_with_buckets".

Returns:

A tuple of the form (outputs, losses), where: `outputs`: *The outputs for each bucket. Its j'th element consists of a list of 2D Tensors. The shape of output tensors can be either [batch_size x output_size] or [batch_size x num_decoder_symbols] depending on the seq2seq model used.* `losses`: List of scalar Tensors, representing losses for each bucket, or, if per_example_loss is set, a list of 1D batch-sized float Tensors.

Raises:

- `ValueError`: If length of encoder_inputs, targets, or weights is smaller than the largest (last) bucket.

Last updated November 2, 2017.