tf.contrib.seq2seq.dynamic_decode

```
dynamic_decode(
    decoder,
    output_time_major=False,
    impute_finished=False,
    maximum_iterations=None,
    parallel_iterations=32,
    swap_memory=False,
    scope=None
)
```

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

See the guide: Seq2seq Library (contrib) > Dynamic Decoding

Perform dynamic decoding with `decoder`.

Calls initialize() once and step() repeatedly on the Decoder object.

Args:

- `decoder`: A `Decoder` instance.
- `output_time_major`: Python boolean. Default: `False` (batch major). If `True`, outputs are returned as time major tensors (this mode is faster). Otherwise, outputs are returned as batch major tensors (this adds extra time to the computation).
- `impute_finished`: Python boolean. If `True`, then states for batch entries which are marked as finished get copied through and the corresponding outputs get zeroed out. This causes some slowdown at each time step, but ensures that the final state and outputs have the correct values and that backprop ignores time steps that were marked as finished.
- `maximum_iterations`: `int32` scalar, maximum allowed number of decoding steps. Default is `None` (decode until the decoder is fully done).
- `parallel_iterations`: Argument passed to `tf.while_loop`.
- `swap_memory`: Argument passed to `tf.while_loop`.
- `scope`: Optional variable scope to use.

Returns:

`(final_outputs, final_state, final_sequence_lengths)`.

Raises:

- `TypeError`: if `decoder` is not an instance of `Decoder`.
- `ValueError`: if `maximum_iterations` is provided but is not a scalar.

Last updated November 2, 2017.