tf.contrib.legacy_seq2seq.embedding_tied_rnn_seq2seq

```
embedding_tied_rnn_seq2seq(
    encoder_inputs,
    decoder_inputs,
    cell,
    num_symbols,
    embedding_size,
    num_decoder_symbols=None,
    output_projection=None,
    feed_previous=False,
    dtype=None,
    scope=None
)
```

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

Embedding RNN sequence-to-sequence model with tied (shared) parameters.

This model first embeds encoder_inputs by a newly created embedding (of shape [num_symbols x input_size]). Then it runs an RNN to encode embedded encoder_inputs into a state vector. Next, it embeds decoder_inputs using the same embedding. Then it runs RNN decoder, initialized with the last encoder state, on embedded decoder_inputs. The decoder output is over symbols from 0 to num_decoder_symbols - 1 if num_decoder_symbols is none; otherwise it is over 0 to num_symbols - 1.

Args:

- `encoder_inputs`: A list of 1D int32 Tensors of shape [batch_size].
- `decoder_inputs`: A list of 1D int32 Tensors of shape [batch_size].
- `cell`: tf.nn.rnn_cell.RNNCell defining the cell function and size.
- `num_symbols`: Integer; number of symbols for both encoder and decoder.
- `embedding_size`: Integer, the length of the embedding vector for each symbol.
- `num_decoder_symbols`: Integer; number of output symbols for decoder. If provided, the decoder output is over symbols 0 to num_decoder_symbols - 1. Otherwise, decoder output is over symbols 0 to num_symbols - 1. Note that this assumes that the vocabulary is set up such that the first num_decoder_symbols of num_symbols are part of decoding.
- `output_projection`: None or a pair (W, B) of output projection weights and biases; W has shape [output_size x num_symbols] and B has shape [num_symbols]; if provided and feed_previous=True, each fed previous output will first be multiplied by W and added B.
- `feed_previous`: Boolean or scalar Boolean Tensor; if True, only the first of decoder_inputs will be used (the "GO" symbol), and all other decoder inputs will be taken from previous outputs (as in embedding_rnn_decoder). If False, decoder_inputs are used as given (the standard decoder case).
- `dtype`: The dtype to use for the initial RNN states (default: tf.float32).
- `scope`: VariableScope for the created subgraph; defaults to "embedding_tied_rnn_seq2seq".

Returns:

A tuple of the form (outputs, state), where: `outputs`: *A list of the same length as decoder_inputs of 2D Tensors with shape [batch_size x output_symbols] containing the generated outputs where output_symbols = num_decoder_symbols if num_decoder_symbols is not None otherwise output_symbols = num_symbols.* `state`: The state of each decoder cell at the final time-step. It is a 2D Tensor of shape [batch_size x cell.state_size].

Raises:

- `ValueError`: When output_projection has the wrong shape.

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.