

tf.contrib.seq2seq.one2many_rnn_seq2seq

```

one2many_rnn_seq2seq(
    encoder_inputs,
    decoder_inputs_dict,
    enc_cell,
    dec_cells_dict,
    num_encoder_symbols,
    num_decoder_symbols_dict,
    embedding_size,
    feed_previous=False,
    dtype=None,
    scope=None
)

```

Defined in [tensorflow/contrib/legacy_seq2seq/python/ops/seq2seq.py](#).

One-to-many RNN sequence-to-sequence model (multi-task).

This is a multi-task sequence-to-sequence model with one encoder and multiple decoders. Reference to multi-task sequence-to-sequence learning can be found here: <http://arxiv.org/abs/1511.06114>

Args:

- `encoder_inputs`: A list of 1D int32 Tensors of shape `[batch_size]`.
- `decoder_inputs_dict`: A dictionary mapping decoder name (string) to the corresponding decoder_inputs; each decoder_inputs is a list of 1D Tensors of shape `[batch_size]`; num_decoders is defined as `len(decoder_inputs_dict)`.
- `enc_cell`: `tf.nn.rnn_cell.RNNCell` defining the encoder cell function and size.
- `dec_cells_dict`: A dictionary mapping encoder name (string) to an instance of `tf.nn.rnn_cell.RNNCell`.
- `num_encoder_symbols`: Integer; number of symbols on the encoder side.
- `num_decoder_symbols_dict`: A dictionary mapping decoder name (string) to an integer specifying number of symbols for the corresponding decoder; `len(num_decoder_symbols_dict)` must be equal to num_decoders.
- `embedding_size`: Integer, the length of the embedding vector for each symbol.
- `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 of the initial state for both the encoder and encoder rnn cells (default: `tf.float32`).
- `scope`: `VariableScope` for the created subgraph; defaults to "one2many_rnn_seq2seq"

Returns:

A tuple of the form `(outputs_dict, state_dict)`, where: `outputs_dict`: A mapping from decoder name (string) to a list of the same length as `decoder_inputs_dict[name]`; each element in the list is a 2D Tensors with shape `[batch_size x num_decoder_symbol_list[name]]` containing the generated outputs. `state_dict`: A mapping from decoder name (string) to the final state of the corresponding decoder RNN; it is a 2D Tensor of shape `[batch_size x cell.state_size]`.

Raises:

- `TypeError` : if `enc_cell` or any of the `dec_cells` are not instances of `RNNCell`.
- `ValueError` : if `len(dec_cells) != len(decoder_inputs_dict)`.

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](#)