tf.contrib.seq2seq.monotonic_attention

```
monotonic_attention(
    p_choose_i,
    previous_attention,
    mode
)
```

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

Compute monotonic attention distribution from choosing probabilities.

Monotonic attention implies that the input sequence is processed in an explicitly left-to-right manner when generating the output sequence. In addition, once an input sequence element is attended to at a given output timestep, elements occurring before it cannot be attended to at subsequent output timesteps. This function generates attention distributions according to these assumptions. For more information, see ``Online and Linear-Time Attention by Enforcing Monotonic Alignments''.

Args:

- `p_choose_i`: Probability of choosing input sequence/memory element i. Should be of shape (batch_size, input_sequence_length), and should all be in the range [0, 1].
- `previous_attention`: The attention distribution from the previous output timestep. Should be of shape (batch_size, input_sequence_length). For the first output timestep, preevious_attention[n] should be [1, 0, 0, ..., 0] for all n in [0, ... batch_size - 1].
- `mode`: How to compute the attention distribution. Must be one of 'recursive', 'parallel', or 'hard'.
    - 'recursive' uses tf.scan to recursively compute the distribution. This is slowest but is exact, general, and does not suffer from numerical instabilities.
    - 'parallel' uses parallelized cumulative-sum and cumulative-product operations to compute a closed-form solution to the recurrence relation defining the attention distribution. This makes it more efficient than 'recursive', but it requires numerical checks which make the distribution non-exact. This can be a problem in particular when input_sequence_length is long and/or p_choose_i has entries very close to 0 or 1.
    - 'hard' requires that the probabilities in p_choose_i are all either 0 or 1, and subsequently uses a more efficient and exact solution.

Returns:

A tensor of shape (batch_size, input_sequence_length) representing the attention distributions for each sequence in the batch.

Raises:

- `ValueError`: mode is not one of 'recursive', 'parallel', 'hard'.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**