

tf.contrib.rnn.FusedRNNCell

Contents

Class FusedRNNCell

Methods

`__call__`Class **FusedRNNCell**

Defined in `tensorflow/contrib/rnn/python/ops/fused_rnn_cell.py`.

See the guide: [RNN and Cells \(contrib\) > Core RNN Cell wrappers \(RNNCells that wrap other RNNCells\)](#)

Abstract object representing a fused RNN cell.

A fused RNN cell represents the entire RNN expanded over the time dimension. In effect, this represents an entire recurrent network.

Unlike RNN cells which are subclasses of `rnn_cell.RNNCell`, a `FusedRNNCell` operates on the entire time sequence at once, by putting the loop over time inside the cell. This usually leads to much more efficient, but more complex and less flexible implementations.

Every `FusedRNNCell` must implement `__call__` with the following signature.

Methods

`__call__`

```
__call__(
    inputs,
    initial_state=None,
    dtype=None,
    sequence_length=None,
    scope=None
)
```

Run this fused RNN on inputs, starting from the given state.

Args:

- `inputs`: 3-D tensor with shape `[time_len x batch_size x input_size]` or a list of `time_len` tensors of shape `[batch_size x input_size]`.
- `initial_state`: either a tensor with shape `[batch_size x state_size]` or a tuple with shapes `[batch_size x s]` for `s in state_size`, if the cell takes tuples. If this is not provided, the cell is expected to create a zero initial state of type `dtype`.
- `dtype`: The data type for the initial state and expected output. Required if `initial_state` is not provided or RNN state has a heterogeneous dtype.
- `sequence_length`: Specifies the length of each sequence in inputs. An `int32` or `int64` vector (tensor) size

`[batch_size]`, values in `[0, time_len)`. Defaults to `time_len` for each element.

- `scope` : `VariableScope` or `string` for the created subgraph; defaults to class name.

Returns:

A pair containing:

- Output: A 3-D tensor of shape `[time_len x batch_size x output_size]` or a list of `time_len` tensors of shape `[batch_size x output_size]`, to match the type of the `inputs`.
- Final state: Either a single 2-D tensor, or a tuple of tensors matching the arity and shapes of `initial_state`.

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](#)