

tf.contrib.seq2seq.Decoder

Contents

Class Decoder

Properties

batch_size

output_dtype

Class **Decoder**

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

See the guide: [Seq2seq Library \(contrib\) > Dynamic Decoding](#)

An RNN Decoder abstract interface object.

Concepts used by this interface: - **inputs**: (structure of) tensors and TensorArrays that is passed as input to the RNNCell composing the decoder, at each time step. - **state**: (structure of) tensors and TensorArrays that is passed to the RNNCell instance as the state. - **finished**: boolean tensor telling whether each sequence in the batch is finished. - **outputs**: Instance of BasicDecoderOutput. Result of the decoding, at each time step.

Properties

batch_size

The batch size of input values.

output_dtype

A (possibly nested tuple of...) dtype[s].

output_size

A (possibly nested tuple of...) integer[s] or `TensorShape` object[s].

Methods

finalize

```
finalize(  
    outputs,  
    final_state,  
    sequence_lengths  
)
```

initialize

```
initialize(name=None)
```

Called before any decoding iterations.

This methods must compute initial input values and initial state.

Args:

- `name` : Name scope for any created operations.

Returns:

`(finished, initial_inputs, initial_state)` : initial values of 'finished' flags, inputs and state.

step

```
step(  
    time,  
    inputs,  
    state,  
    name=None  
)
```

Called per step of decoding (but only once for dynamic decoding).

Args:

- `time` : Scalar `int32` tensor. Current step number.
- `inputs` : RNNCell input (possibly nested tuple of) tensor[s] for this time step.
- `state` : RNNCell state (possibly nested tuple of) tensor[s] from previous time step.
- `name` : Name scope for any created operations.

Returns:

`(outputs, next_state, next_inputs, finished)` : `outputs` is an object containing the decoder output, `next_state` is a (structure of) state tensors and TensorArrays, `next_inputs` is the tensor that should be used as input for the next step, `finished` is a boolean tensor telling whether the sequence is complete, for each sequence in the batch.

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](#)