

tf.contrib.seq2seq.BasicDecoder

Contents

Class BasicDecoder

Properties

batch_size

output_dtype

Class **BasicDecoder**Inherits From: [Decoder](#)Defined in [tensorflow/contrib/seq2seq/python/ops/basic_decoder.py](#).See the guide: [Seq2seq Library \(contrib\) > Dynamic Decoding](#)

Basic sampling decoder.

Properties

batch_size**output_dtype****output_size**

Methods

__init__

```
__init__(  
    cell,  
    helper,  
    initial_state,  
    output_layer=None  
)
```

Initialize BasicDecoder.

Args:

- `cell`: An [RNNCell](#) instance.
- `helper`: A [Helper](#) instance.
- `initial_state`: A (possibly nested tuple of...) tensors and TensorArrays. The initial state of the RNNCell.
- `output_layer`: (Optional) An instance of [tf.layers.Layer](#), i.e., [tf.layers.Dense](#). Optional layer to apply to the

RNN output prior to storing the result or sampling.

Raises:

- `TypeError` : if `cell`, `helper` or `output_layer` have an incorrect type.

finalize

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

initialize

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

Initialize the decoder.

Args:

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

Returns:

```
(finished, first_inputs, initial_state) .
```

step

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

Perform a decoding step.

Args:

- `time` : scalar `int32` tensor.
- `inputs` : A (structure of) input tensors.
- `state` : A (structure of) state tensors and TensorArrays.
- `name` : Name scope for any created operations.

Returns:

```
(outputs, next_state, next_inputs, finished) .
```

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](#)