#### TencorFlow

TensorFlow API r1.4

## tf.uniform\_unit\_scaling\_initializer

```
Contents

Class uniform_unit_scaling_initializer
   Aliases:

Methods
   __init__
   __call__
   from_config
   get_config
```

# Class uniform\_unit\_scaling\_initializer

Inherits From: Initializer

### Aliases:

- Class tf.initializers.uniform\_unit\_scaling
- Class tf.uniform\_unit\_scaling\_initializer

Defined in tensorflow/python/ops/init\_ops.py.

See the guide: Variables > Sharing Variables

Initializer that generates tensors without scaling variance.

When initializing a deep network, it is in principle advantageous to keep the scale of the input variance constant, so it does not explode or diminish by reaching the final layer. If the input is  $\mathbf{x}$  and the operation  $\mathbf{x} * \mathbf{W}$ , and we want to initialize  $\mathbf{W}$  uniformly at random, we need to pick  $\mathbf{W}$  from

```
[-sqrt(3) / sqrt(dim), sqrt(3) / sqrt(dim)]
```

to keep the scale intact, where <code>dim = W.shape[0]</code> (the size of the input). A similar calculation for convolutional networks gives an analogous result with <code>dim</code> equal to the product of the first 3 dimensions. When nonlinearities are present, we need to multiply this by a constant <code>factor</code>. See Sussillo et al., 2014 (pdf) for deeper motivation, experiments and the calculation of constants. In section 2.3 there, the constants were numerically computed: for a linear layer it's 1.0, relu: ~1.43, tanh: ~1.15.

### Args:

- factor: Float. A multiplicative factor by which the values will be scaled.
- seed: A Python integer. Used to create random seeds. See tf.set\_random\_seed for behavior.
- dtype: The data type. Only floating point types are supported.

### Methods

## \_\_init\_\_

```
__init__(
   factor=1.0,
   seed=None,
   dtype=tf.float32
)
```

#### DEPRECATED FUNCTION

THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: Use tf.initializers.variance\_scaling instead with distribution=uniform to get equivalent behavior.

## \_\_call\_\_

```
__call__(
    shape,
    dtype=None,
    partition_info=None
)
```

### from\_config

```
from_config(
   cls,
   config
)
```

Instantiates an initializer from a configuration dictionary.

Example:

```
initializer = RandomUniform(-1, 1)
config = initializer.get_config()
initializer = RandomUniform.from_config(config)
```

### Args:

• config: A Python dictionary. It will typically be the output of **get\_config**.

Returns:

An Initializer instance.

### get\_config

```
get_config()
```

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	