

## tf.contrib.layers.xavier\_initializer

## Contents

## Aliases:

## Aliases:

- `tf.contrib.layers.xavier_initializer`
- `tf.contrib.layers.xavier_initializer_conv2d`

```
xavier_initializer(  
    uniform=True,  
    seed=None,  
    dtype=tf.float32  
)
```

Defined in [tensorflow/contrib/layers/python/layers/initializers.py](#).

See the guide: [Layers \(contrib\) > Initializers](#)

Returns an initializer performing "Xavier" initialization for weights.

This function implements the weight initialization from:

Xavier Glorot and Yoshua Bengio (2010): [Understanding the difficulty of training deep feedforward neural networks. International conference on artificial intelligence and statistics.](#)

This initializer is designed to keep the scale of the gradients roughly the same in all layers. In uniform distribution this ends up being the range: `x = sqrt(6. / (in + out)); [-x, x]` and for normal distribution a standard deviation of `sqrt(2. / (in + out))` is used.

## Args:

- `uniform`: Whether to use uniform or normal distributed random initialization.
- `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.

## Returns:

An initializer for a weight matrix.

---

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.

[Blog](#)

[GitHub](#)

[Twitter](#)

**Support**

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

**English**

[Terms](#) | [Privacy](#)