tf.uniform_unit_scaling_initializer

Contents

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 `x` and the operation `x * W`, and we want to initialize `W` uniformly at random, we need to pick `W` from

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

to keep the scale intact, where `dim = W.shape[0]` (the size of the input). A similar calculation for convolutional networks gives an analogous result with `dim` equal to the product of the first 3 dimensions. When nonlinearities are present, we need to multiply this by a constant `factor`. 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()
```