tf.keras.constraints.MinMaxNorm

Class `MinMaxNorm`

Inherits From: `Constraint`

Aliases:

- Class `tf.keras.constraints.MinMaxNorm`
- Class `tf.keras.constraints.min_max_norm`

Defined in `tensorflow/python/keras/_impl/keras/constraints.py`.

MinMaxNorm weight constraint.

Constrains the weights incident to each hidden unit to have the norm between a lower bound and an upper bound.

Arguments:

- `min_value`: the minimum norm for the incoming weights.
- `max_value`: the maximum norm for the incoming weights.
- `rate`: rate for enforcing the constraint: weights will be rescaled to yield `(1 - rate) * norm + rate * norm.clip(min_value, max_value)`. Effectively, this means that rate=1.0 stands for strict enforcement of the constraint, while rate<1.0 means that weights will be rescaled at each step to slowly move towards a value inside the desired interval.
- `axis`: integer, axis along which to calculate weight norms. For instance, in a `Dense` layer the weight matrix has shape `(input_dim, output_dim)`, set `axis` to `0` to constrain each weight vector of length `(input_dim,)`. In a `Conv2D` layer with `dim_ordering="channels_last"`, the weight tensor has shape `(rows, cols, input_depth, output_depth)`, set `axis` to `[0, 1, 2]` to constrain the weights of each filter tensor of size `(rows, cols, input_depth)`.

Methods

`__init__`

```
__init__(
    min_value=0.0,
    max_value=1.0,
    rate=1.0,
    axis=0
)
```

__call__

```
__call__(w)
```

get_config

```
get_config()
```

Last updated November 2, 2017.