tf.keras.optimizers.Nadam

Contents

Class Nadam

Methods

__init__

from_config

Class **Nadam**

Inherits From: `Optimizer`

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

Nesterov Adam optimizer.

Much like Adam is essentially RMSprop with momentum, Nadam is Adam RMSprop with Nesterov momentum.

Default parameters follow those provided in the paper. It is recommended to leave the parameters of this optimizer at their default values.

Arguments:

- `lr`: float >= 0. Learning rate. beta_1/beta_2: floats, 0 < beta < 1. Generally close to 1.
- `epsilon`: float >= 0. Fuzz factor.

References: - Nadam report - On the importance of initialization and momentum in deep learning

Methods

`__init__`

```
__init__(
    lr=0.002,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-08,
    schedule_decay=0.004,
    **kwargs
)
```

`from_config`

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

get_config

```
get_config()
```

get_gradients

```
get_gradients(
    loss,
    params
)
```

get_updates

```
get_updates(
    loss,
    params
)
```

get_weights

```
get_weights()
```

Returns the current value of the weights of the optimizer.

Returns:

A list of numpy arrays.

set_weights

```
set_weights(weights)
```

Sets the weights of the optimizer, from Numpy arrays.

Should only be called after computing the gradients (otherwise the optimizer has no weights).

Arguments:

- `weights`: a list of Numpy arrays. The number of arrays and their shape must match number of the dimensions of the weights of the optimizer (i.e. it should match the output of `get_weights`).

Raises:

- `ValueError`: in case of incompatible weight shapes.

Last updated November 2, 2017.