

## tf.contrib.kfac.fisher\_blocks.FullyConnectedDiagonalFB

### Contents

Class `FullyConnectedDiagonalFB`

### Methods

`__init__`

`instantiate_factors`

`multiply`

`multiply_inverse`

`tensors_to_compute_grads`

## Class `FullyConnectedDiagonalFB`

Inherits From: `FisherBlock`

Defined in `tensorflow/contrib/kfac/python/ops/fisher_blocks.py`.

FisherBlock for fully-connected (dense) layers using a diagonal approx.

Unlike `NaiveDiagonalFB` this uses the low-variance "sum of squares" estimator that is computed using the well-known trick.

## Methods

### `__init__`

```
__init__(
    layer_collection,
    inputs,
    outputs,
    has_bias=False
)
```

Creates a `FullyConnectedDiagonalFB` block.

Args:

- `layer_collection`: The collection of all layers in the K-FAC approximate Fisher information matrix to which this `FisherBlock` belongs.
- `inputs`: The Tensor of input activations to this layer.
- `outputs`: The Tensor of output pre-activations from this layer.
- `has_bias`: Whether the component Kronecker factors have an additive bias. (Default: False)

### `instantiate_factors`

```
instantiate_factors(  
    grads_list,  
    damping  
)
```

## multiply

```
multiply(vector)
```

## multiply\_inverse

```
multiply_inverse(vector)
```

## tensors\_to\_compute\_grads

```
tensors_to_compute_grads()
```

---

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](#)