tf.contrib.kfac.curvature_matrix_vector_products.CurvatureMatrixVectorProductCo.

Contents

Class `CurvatureMatrixVectorProductComputer`

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

Class for computing matrix-vector products for Fishers, GGNs and Hessians.

In other words we compute Mv *where M is the matrix, v is the vector, and* refers to standard matrix/vector multiplication (not element-wise multiplication).

The matrices are defined in terms of some differential quantity of the total loss function with respect to a provided list of tensors ("wrt_tensors"). For example, the Fisher associated with a log-prob loss w.r.t. the parameters.

The vecs argument to each method are lists of tensors that must be the size as the corresponding ones from "wrt_tensors". They represent the vector being multiplied.

"factors" of the matrix M are defined as matrices B such that B*B^T = M. Methods that multiply by the factor B take a "loss_inner_vecs" argument instead of vecs, which must be a list of tensors with shapes given by the corresponding XXX_inner_shapes property.

Note that matrix-vector products are not normalized by the batch size, nor are any damping terms added to the results. These things can be easily applied externally, if desired.

See for example: www.cs.utoronto.ca/~jmartens/docs/HF_book_chapter.pdf and https://arxiv.org/abs/1412.1193 for more information about the generalized Gauss-Newton, Fisher, etc., and how to compute matrix-vector products.

Properties

`fisher_factor_inner_shapes`

Shapes required by multiply_fisher_factor.

`generalized_gauss_newton_factor_inner_shapes`

Shapes required by multiply_generalized_gauss_newton_factor.

Methods

`__init__`

```
__init__(
    losses,
    wrt_tensors
)
```

Create a CurvatureMatrixVectorProductComputer object.

Args:

- `losses`: A list of LossFunction instances whose sum defines the total loss.
- `wrt_tensors`: A list of Tensors to compute the differential quantities defining the matrices with respect to (see class description).

multiply_fisher

```
multiply_fisher(vecs)
```

Multiply vecs by Fisher of total loss.

multiply_fisher_factor

```
multiply_fisher_factor(loss_inner_vecs)
```

Multiply loss_inner_vecs by factor of Fisher of total loss.

multiply_fisher_factor_transpose

```
multiply_fisher_factor_transpose(vecs)
```

Multiply vecs by transpose of factor of Fisher of total loss.

multiply_generalized_gauss_newton

```
multiply_generalized_gauss_newton(vecs)
```

Multiply vecs by generalized Gauss-Newton of total loss.

multiply_generalized_gauss_newton_factor

```
multiply_generalized_gauss_newton_factor(loss_inner_vecs)
```

Multiply loss_inner_vecs by factor of GGN of total loss.

multiply_generalized_gauss_newton_factor_transpose

```
multiply_generalized_gauss_newton_factor_transpose(vecs)
```

Multiply vecs by transpose of factor of GGN of total loss.

multiply_hessian

```
multiply_hessian(vecs)
```

Multiply vecs by Hessian of total loss.

Last updated November 2, 2017.