
DRM Documentation

Release 0.1

Michele D'Asaro, Umut Guclu, Marcel van Gerven

April 02, 2017

CONTENTS

1	Introduction	1
2	DRM package	3
3	Indices and tables	7
	Python Module Index	9
	Index	11

**CHAPTER
ONE**

INTRODUCTION

This should contain an introduction to DRM.

DRM PACKAGE

class `DRM.base.DRM(drm_net)`

Bases: `object`

wrapper object that trains and analyses the model at hand

estimate (`data_iter, val_iter=None, n_epochs=1, cutoff=None`)

Estimation via truncated backprop

Parameters

- **data_iter** – iterator which generates sensations/responses at some specified resolution
- **val_iter** – optional iterator which generates sensations/responses at some specified resolution used for validation
- **n_epochs** – number of training epochs
- **cutoff** – cutoff for truncated backpropagation

Returns train loss and validation loss

forward (`data_iter`)

Forward propagation

Parameters `data_iter` –

Returns generated response and population activity

class `DRM.base.DRMLoss`

Bases: `torch.nn.modules.module.Module`

MSE loss which ignores missing data

forward (`prediction, target`)

Computes loss on a prediction and a target

Computes MSE loss but ignores those terms where the target is equal to nan, indicating missing data.

Parameters

- **prediction** (*Variable*) – Prediction of output
- **target** (*Variable*) – Target output

Returns MSE loss

Return type `Variable`

class `DRM.base.DRMNet(populations, ws, Wp, readout)`

Bases: `torch.nn.modules.container.Sequential`

```
detach_()
    Detach gradients for truncation

forward(x)
    Forward propagation

    Parameters x – sensory input at this point in time (zeros for no input); numpy array

    Returns predicted output measurements

reset()
    Reset states of model components

class DRM.base.DRMNode(n_in=1, n_out=1)
    Bases: torch.nn.modules.module.Module

    Base class for populations, readouts and connections

detach_()
    Detach gradients for truncation

forward(x)
    Forward pass for this node

    Parameters x – input data

    Returns output data

reset()
    The function that is called when resetting internal state

class DRM.connection.DRMConnection(n_in=1, n_out=1, delay=1)
    Bases: DRM.base.DRMNode

detach_()
    Detach gradients for truncation

forward(x)
    Forward propagation

    Parameters x – input to connection

    Returns connection output

reset()
    Reset state

class DRM.iterators.DRMIterator(resolution, stimulus, stim_time, response=None, resp_time=None,
    batch_size=None, n_batches=None)
    Bases: object

__iter__()
    Initializes data generator. Should be invoked at the start of each epoch

    Returns self

is_final()
    Flags if final iteration is reached

    Returns boolean if final batch is reached

next()
    Produces next data item

    Returns dictionary containing the stimulus and the response as torch variables
```

```
class DRM.population.DRMPopulation (n_in=1, n_out=1, delay=1)
Bases: DRM.base.DRMNode

forward(x)
    Forward propagation

    Parameters x (list of afferent population outputs) – population input

    Returns population output

class DRM.readout.DRMReadout (n_in=1, n_out=1)
Bases: DRM.base.DRMNode

forward(x)
    Forward propagation

    Parameters x (list of afferent population outputs) – readout input

    Returns predicted measurements
```

**CHAPTER
THREE**

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

`DRM.base`, 3
`DRM.connection`, 4
`DRM.iterators`, 4
`DRM.population`, 4
`DRM.readout`, 5

Symbols

`__iter__()` (DRM.iterators.DRMIterator method), 4

D

`detach_()` (DRM.base.DRMNet method), 3
`detach_()` (DRM.base.DRMNode method), 4
`detach_()` (DRM.connection.DRMConnection method), 4
`DRM` (class in DRM.base), 3
`DRM.base` (module), 3
`DRM.connection` (module), 4
`DRM.iterators` (module), 4
`DRM.population` (module), 4
`DRM.readout` (module), 5
`DRMConnection` (class in DRM.connection), 4
`DRMIterator` (class in DRM.iterators), 4
`DRMLoss` (class in DRM.base), 3
`DRMNet` (class in DRM.base), 3
`DRMNode` (class in DRM.base), 4
`DRMPopulation` (class in DRM.population), 4
`DRMReadout` (class in DRM.readout), 5

E

`estimate()` (DRM.base.DRM method), 3

F

`forward()` (DRM.base.DRM method), 3
`forward()` (DRM.base.DRMLoss method), 3
`forward()` (DRM.base.DRMNet method), 4
`forward()` (DRM.base.DRMNode method), 4
`forward()` (DRM.connection.DRMConnection method), 4
`forward()` (DRM.population.DRMPopulation method), 5
`forward()` (DRM.readout.DRMReadout method), 5

I

`is_final()` (DRM.iterators.DRMIterator method), 4

N

`next()` (DRM.iterators.DRMIterator method), 4

R

`reset()` (DRM.base.DRMNet method), 4
`reset()` (DRM.base.DRMNode method), 4
`reset()` (DRM.connection.DRMConnection method), 4