

---

# **Cosmos Documentation**

***Release 1.0***

**Marcel van Gerven**

May 13, 2017



## CONTENTS

<b>1</b>	<b>supervised_learning package</b>	<b>3</b>
1.1	Submodules . . . . .	3
1.2	supervised_learning.agents module . . . . .	3
1.3	supervised_learning.iterators module . . . . .	3
1.4	supervised_learning.models module . . . . .	4
1.5	supervised_learning.networks module . . . . .	4
1.6	supervised_learning.unit_test module . . . . .	5
1.7	supervised_learning.world module . . . . .	5
1.8	Module contents . . . . .	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



Contents:



## SUPERVISED\_LEARNING PACKAGE

### 1.1 Submodules

### 1.2 supervised\_learning.agents module

**class** supervised\_learning.agents.**SupervisedAgent** (*model, optimizer, cutoff=None*)

Bases: object

Agent which trains on labelled data

**\_\_call\_\_** (*data*)

Runs networks in forward mode and applies optional output function

**Parameters** *data* –

**Returns** post-processed output

**reset\_state** ()

Resets persistent states

**test** (*data*)

Returns the loss for one batch

**Parameters** *data* –

**Returns** loss

**train** (*data*)

Train agent on one batch :param data: :return: loss

### 1.3 supervised\_learning.iterators module

**class** supervised\_learning.iterators.**RandomIterator** (*data, batch\_size=None*)

Bases: object

Generates random subsets of data

**next** ()

**class** supervised\_learning.iterators.**SequentialIterator** (*data, batch\_size=None*)

Bases: object

Generates subsets of data such that each batch contains data for the next time point

**next** ()

## 1.4 supervised\_learning.models module

**class** supervised\_learning.models.**Classifier** (*net*, *gpu=-1*)  
Bases: *supervised\_learning.models.Model*  
Wrapper for classification problems

**class** supervised\_learning.models.**Model** (*net*, *loss\_function*, *output\_function=<function <lambda>>*, *gpu=-1*)  
Bases: *chainer.link.Chain*  
Model which wraps a network to compute loss and generate actual predictions

**\_\_call\_\_** (*data*)  
Compute loss for minibatch of data  
**Parameters** *data* – list of minibatches (e.g. inputs and targets for supervised learning)  
**Returns** loss

**has\_state**  
Checks if a network has persistent states  
**Returns** bool

**predict** (*data*)  
Returns prediction, which can be different than raw output (e.g. for softmax function)  
**Parameters** *data* – minibatch or list of minibatches representing input to the model  
**Returns** prediction

**reset\_state** ()

**class** supervised\_learning.models.**Regressor** (*net*, *gpu=-1*)  
Bases: *supervised\_learning.models.Model*  
Wrapper for regression problems

## 1.5 supervised\_learning.networks module

**class** supervised\_learning.networks.**MLP** (*n\_input=None*, *n\_output=1*, *n\_hidden=10*)  
Bases: *chainer.link.Chain*  
Multilayer perceptron

**has\_state**  
Checks if a network has persistent states  
**Returns** bool

**class** supervised\_learning.networks.**RNN** (*n\_input=None*, *n\_output=1*, *n\_hidden=10*)  
Bases: *chainer.link.Chain*

**has\_state**  
Checks if a network has persistent states  
**Returns** bool

**reset\_state** ()  
Resets persistent states



## 1.6 supervised\_learning.unit\_test module

```
class supervised_learning.unit_test.UnitTest (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    test_gpu()  
        Test training procedure for stateless network on GPU  
  
    test_stateful_network()  
        Test training procedure for stateful network  
  
    test_stateless_network()  
        Test training procedure for stateless network
```

## 1.7 supervised\_learning.world module

```
class supervised_learning.world.World (agents, out='result')  
    Bases: object  
  
    Wrapper object which takes care of training and testing on some data iterator for one or more agents  
  
    test (test_iter)  
        Parameters test_iter – iterator over the test data  
        Returns test loss  
  
    train (train_iter, n_epochs, test_iter=None, snapshot=0)  
        Parameters  
        • train_iter – iterator over the training data  
        • n_epochs (int) – number of epochs to train on  
        • test_iter – optional iterator over the test data (returns optimal model)  
        • snapshot (int) – whether or not to save model after each epochs modulo snapshot  
        Returns train loss and optional test loss
```

## 1.8 Module contents



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## S

- `supervised_learning`, 5
- `supervised_learning.agents`, 3
- `supervised_learning.iterators`, 3
- `supervised_learning.models`, 4
- `supervised_learning.networks`, 4
- `supervised_learning.unit_test`, 5
- `supervised_learning.world`, 5



## Symbols

`__call__()` (supervised\_learning.agents.SupervisedAgent method), 3  
`__call__()` (supervised\_learning.models.Model method), 4

## C

Classifier (class in supervised\_learning.models), 4

## H

`has_state` (supervised\_learning.models.Model attribute), 4  
`has_state` (supervised\_learning.networks.MLP attribute), 4  
`has_state` (supervised\_learning.networks.RNN attribute), 4

## M

MLP (class in supervised\_learning.networks), 4  
 Model (class in supervised\_learning.models), 4

## N

`next()` (supervised\_learning.iterators.RandomIterator method), 3  
`next()` (supervised\_learning.iterators.SequentialIterator method), 3

## P

`predict()` (supervised\_learning.models.Model method), 4

## R

RandomIterator (class in supervised\_learning.iterators), 3  
 Regressor (class in supervised\_learning.models), 4  
`reset_state()` (supervised\_learning.agents.SupervisedAgent method), 3  
`reset_state()` (supervised\_learning.models.Model method), 4  
`reset_state()` (supervised\_learning.networks.RNN method), 4  
 RNN (class in supervised\_learning.networks), 4

## S

SequentialIterator (class in supervised\_learning.iterators), 3  
 supervised\_learning (module), 5  
 supervised\_learning.agents (module), 3  
 supervised\_learning.iterators (module), 3  
 supervised\_learning.models (module), 4  
 supervised\_learning.networks (module), 4  
 supervised\_learning.unit\_test (module), 5  
 supervised\_learning.world (module), 5  
 SupervisedAgent (class in supervised\_learning.agents), 3

## T

`test()` (supervised\_learning.agents.SupervisedAgent method), 3  
`test()` (supervised\_learning.world.World method), 5  
`test_gpu()` (supervised\_learning.unit\_test.UnitTest method), 5  
`test_stateful_network()` (supervised\_learning.unit\_test.UnitTest method), 5  
`test_stateless_network()` (supervised\_learning.unit\_test.UnitTest method), 5  
`train()` (supervised\_learning.agents.SupervisedAgent method), 3  
`train()` (supervised\_learning.world.World method), 5

## U

UnitTest (class in supervised\_learning.unit\_test), 5

## W

World (class in supervised\_learning.world), 5