
tf-mdp Documentation

Release v0.5.2-alpha

Thiago P. Bueno

Apr 17, 2019

Contents:

1	tfmdp package	1
1.1	Subpackages	1
1.1.1	tfmdp.model package	1
1.1.1.1	Subpackages	1
1.1.1.2	Submodules	3
1.1.1.3	tfmdp.model.sequential.montecarlo module	3
1.1.1.4	tfmdp.model.sequential.mrm module	3
1.1.1.5	Module contents	3
1.1.2	tfmdp.planning package	3
1.1.2.1	Submodules	3
1.1.2.2	tfmdp.planning.pdplanner module	3
1.1.2.3	tfmdp.planning.planner module	3
1.1.2.4	Module contents	3
1.1.3	tfmdp.policy package	3
1.1.3.1	Subpackages	3
1.1.3.2	Submodules	4
1.1.3.3	tfmdp.policy.drp module	4
1.1.3.4	tfmdp.policy.feedforward module	6
1.1.3.5	Module contents	7
1.2	Submodules	7
1.3	tfmdp.utils module	7
1.4	Module contents	7
2	Indices and tables	9
	Python Module Index	11

1.1 Subpackages

1.1.1 tfmdp.model package

1.1.1.1 Subpackages

tfmdp.model.cell package

Submodules

tfmdp.model.cell.basic_cell module

```
class tfmdp.model.cell.basic_cell.BasicMarkovCell (compiler:
                                                    rddl2tf.compiler.Compiler, policy:
                                                    tfmdp.policy.drp.DeepReactivePolicy,
                                                    config: Optional[Dict] = None)
```

Bases: tensorflow.python.ops.rnn_cell_impl.RNNCell

BasicMarkovCell implements a 1-step MDP transition function as an RNNCell whose hidden state is the MDP current state and output is a tuple with next state, action, intermediate fluents, and reward.

Parameters

- **compiler** (*rddl2tf.compiler.Compiler*) – RDDDL2TensorFlow compiler.
- **config** (*Dict*) – The cell configuration parameters.

```
__call__ (inputs:          tensorflow.python.framework.ops.Tensor,      state:          Se-
           quence[tensorflow.python.framework.ops.Tensor],             scope:          Optional[str]    =
           None)      →      Tuple[Tuple[Sequence[tensorflow.python.framework.ops.Tensor],      Se-
           quence[tensorflow.python.framework.ops.Tensor], Sequence[tensorflow.python.framework.ops.Tensor],
           tensorflow.python.framework.ops.Tensor], Sequence[tensorflow.python.framework.ops.Tensor]]
Returns the cell's output tuple and next state tensors.
```

Output tuple packs together the next state, action, interms, and reward tensors in order.

Parameters

- **inputs** (*tf.Tensor*) – The timestep input tensor.
- **state** (*Sequence[tf.Tensor]*) – The current state tensors.
- **scope** (*Optional[str]*) – The cell name scope.

Returns A pair with the cell’s output tuple and next state.

Return type (CellOutput, CellState)

action_size

Returns the MDP action size.

graph

Returns the cell’s computation graph.

interm_size

Returns the MDP intermediate state size.

output_size

Returns the simulation cell output size.

state_size

Returns the MDP state size.

class tfmdp.model.cell.basic_cell.**OutputTuple** (*state, action, interms, reward*)

Bases: tuple

__getnewargs__ ()

Return self as a plain tuple. Used by copy and pickle.

static **__new__** (*_cls, state, action, interms, reward*)

Create new instance of OutputTuple(state, action, interms, reward)

__repr__ ()

Return a nicely formatted representation string

__asdict ()

Return a new OrderedDict which maps field names to their values.

classmethod **__make** (*iterable, new=<built-in method __new__ of type object>, len=<built-in function len>*)

Make a new OutputTuple object from a sequence or iterable

__replace (***kws*)

Return a new OutputTuple object replacing specified fields with new values

action

Alias for field number 1

interms

Alias for field number 2

reward

Alias for field number 3

state

Alias for field number 0

tfmdp.model.cell.basic_cell.**cell_size** (*sizes: Sequence[Sequence[int]]*) → Sequence[Union[Sequence[int], int]]

`tfmdp.model.cell.basic_cell.to_tensor` (*fluents*)

Module contents

1.1.1.2 Submodules

1.1.1.3 `tfmdp.model.sequential.montecarlo` module

1.1.1.4 `tfmdp.model.sequential.mrm` module

1.1.1.5 Module contents

1.1.2 `tfmdp.planning` package

1.1.2.1 Submodules

1.1.2.2 `tfmdp.planning.pdplanner` module

1.1.2.3 `tfmdp.planning.planner` module

1.1.2.4 Module contents

1.1.3 `tfmdp.policy` package

1.1.3.1 Subpackages

`tfmdp.policy.layers` package

Submodules

`tfmdp.policy.layers.action_layer` module

class `tfmdp.policy.layers.action_layer.ActionLayer` (*action_size: int*)

Bases: `tensorflow.python.layers.base.Layer`

ActionLayer should be used as the output layer in a DRP.

It generates multi-head dense output layers with the same shape as action fluents. Otionally, it restricts the output tensors based on action bounds.

Parameters `action_size` (*Sequence[Sequence[int]]*) – The list of action fluent sizes.

`_get_output_tensor` (*tensor: tensorflow.python.framework.ops.Tensor, bounds: Tuple[Optional[tensorflow.python.framework.ops.Tensor], Optional[tensorflow.python.framework.ops.Tensor]]*) → *tensorflow.python.framework.ops.Tensor*

Returns the value constrained output tensor.

Parameters

- **`tensor`** (*tf.Tensor*) – The layer’s output tensor corresponding to an action fluent.
- **`bounds`** (*Tuple[Optional[tf.Tensor], Optional[tf.Tensor]]*) – The action fluent bounds.

Returns the constrained output tensor.

Return type (tf.Tensor)

call (*inputs*: tensorflow.python.framework.ops.Tensor, *action_bounds*: Optional[Sequence[Tuple[Optional[tensorflow.python.framework.ops.Tensor], Optional[tensorflow.python.framework.ops.Tensor]]]] = None) → Sequence[tensorflow.python.framework.ops.Tensor]
Returns the tensors of the multi-head layer's output.

Parameters

- **inputs** (tf.Tensor) – A hidden layer's output.
- **action_bounds** (Optional[Sequence[Tuple[Optional[tf.Tensor], Optional[tf.Tensor]]]]) – The action bounds.

Returns A tuple of action tensors.

Return type Sequence[tf.Tensor]

trainable_variables

Returns the list of all layer variables/weights.

tfmdp.policy.layers.state_layer module

class tfmdp.policy.layers.state_layer.StateLayer (*input_layer_norm*: bool = False)

Bases: tensorflow.python.layers.base.Layer

StateLayer should be used as an input layer in a DRP.

It flattens each state fluent and returns a single concatenated tensor.

Parameters **input_layer_norm** (bool) – The boolean flag for enabling layer normalization.

call (*inputs*: Sequence[tensorflow.python.framework.ops.Tensor]) → tensorflow.python.framework.ops.Tensor
Returns the concatenation of all state fluent tensors previously flatten.

Parameters **inputs** (Sequence[tf.Tensor]) – A tuple of state fluent tensors.

Returns A single output tensor.

Return type tf.Tensor

trainable_variables

Returns the list of all layer variables/weights.

Module contents

1.1.3.2 Submodules

1.1.3.3 tfmdp.policy.drp module

class tfmdp.policy.drp.DeepReactivePolicy (*compiler*: rddl2tf.compiler.Compiler, *config*: Dict)

Bases: object

DeepReactivePolicy abstract base class.

It defines the basic API for building, saving and restoring reactive policies implemented as deep neural nets.

A reactive policy defines a mapping from current state fluents to action fluents.

Parameters

- **compiler** (`rddl2tf.compiler.Compiler`) – RDDDL2TensorFlow compiler.
- **config** (`Dict`) – The reactive policy configuration parameters.

__call__ (*state*: `Sequence[tensorflow.python.framework.ops.Tensor]`, *timestep*: `tensorflow.python.framework.ops.Tensor`) → `Sequence[tensorflow.python.framework.ops.Tensor]`
Returns action fluents for the current *state* and *timestep*.

Parameters

- **state** (`Sequence[tf.Tensor]`) – A tuple of state fluents.
- **timestep** (`tf.Tensor`) – The current timestep.

Returns A tuple of action fluents.

Return type `Sequence[tf.Tensor]`

build() → `None`

Create the DRP layers and trainable weights.

classmethod from_json (*compiler*: `rddl2tf.compiler.Compiler`, *json_config*: `str`) → `tfmdp.policy.drp.DeepReactivePolicy`
Instantiates a DRP from a *json_config* string.

Parameters

- **compiler** (`rddl2tf.compiler.Compiler`) – RDDDL2TensorFlow compiler.
- **json_config** (`str`) – A DRP configuration encoded in JSON format.

Returns A DRP object.

Return type `tfmdp.policy.drp.DeepReactivePolicy`

graph

name

Returns the canonical DRP name.

restore (*sess*: `tensorflow.python.client.session.Session`, *path*: `Optional[str] = None`) → `None`
Restores previously saved DRP trainable variables.

If path is not provided, restores from last saved checkpoint.

Parameters

- **sess** (`tf.Session`) – A running session.
- **path** (`Optional[str]`) – An optional path to a checkpoint directory.

save (*sess*: `tensorflow.python.client.session.Session`, *path*: `str`) → `str`
Serializes all DRP trainable variables into a checkpoint file.

Parameters

- **sess** (`tf.Session`) – A running session.
- **path** (`str`) – The path to a checkpoint directory.

Returns The path prefix of the newly created checkpoint file.

Return type `str`

size
Returns the number of trainable parameters.

summary () → None
Prints a string summary of the DRP.

to_json () → str
Returns the policy configuration parameters serialized in JSON format.

vars
Returns a list of the trainable variables.

1.1.3.4 tfmdp.policy.feedforward module

class tfmdp.policy.feedforward.**FeedforwardPolicy** (*compiler: rddl2tf.compiler.Compiler, config: dict*)

Bases: *tfmdp.policy.drp.DeepReactivePolicy*

FeedforwardPolicy implements a DRP as a multi-layer perceptron.

It is parameterized by the following configuration params:

- `config['layers']`: a list of number of units; and
- `config['activation']`: an activation function.

Parameters

- **compiler** (*rddl2tf.compiler.Compiler*) – RDDDL2TensorFlow compiler.
- **config** (*Dict*) – The policy configuration parameters.

__call__ (*state: Sequence[tensorflow.python.framework.ops.Tensor], timestep: tensorflow.python.framework.ops.Tensor*) → *Sequence[tensorflow.python.framework.ops.Tensor]*
Returns action fluents for the current *state* and *timestep*.

Parameters

- **state** (*Sequence[tf.Tensor]*) – A tuple of state fluents.
- **timestep** (*tf.Tensor*) – The current timestep.

Returns A tuple of action fluents.

Return type *Sequence[tf.Tensor]*

_build_hidden_layers () → None
Builds all hidden layers as *tf.layers.Dense* layers.

_build_input_layer () → None
Builds the DRP input layer using a *tfmdp.policy.layers.state_layer.StateLayer*.

_build_output_layer () → None
Builds the DRP output layer using a *tfmdp.policy.layers.action_layer.ActionLayer*.

build () → None
Create the DRP layers and trainable weights.

name
Returns the canonical DRP name.

size
Returns the number of trainable parameters.

vars

Returns a list of the trainable variables.

1.1.3.5 Module contents**1.2 Submodules****1.3 tfmdp.utils module**

`tfmdp.utils.get_params_string` (*config: Dict*) → str

Returns a canonical configuration string by concatenating its parameters.

1.4 Module contents

CHAPTER 2

Indices and tables

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

t

- `tfmdp`, 7
- `tfmdp.model`, 3
 - `tfmdp.model.cell`, 3
 - `tfmdp.model.cell.basic_cell`, 1
- `tfmdp.planning`, 3
- `tfmdp.policy`, 7
 - `tfmdp.policy.drp`, 4
 - `tfmdp.policy.feedforward`, 6
 - `tfmdp.policy.layers`, 4
 - `tfmdp.policy.layers.action_layer`, 3
 - `tfmdp.policy.layers.state_layer`, 4
- `tfmdp.utils`, 7

Symbols

`__call__()` (*tfmdp.model.cell.basic_cell.BasicMarkovCell* (class in *tfmdp.model.cell.basic_cell*), 1)
`__call__()` (*tfmdp.policy.drp.DeepReactivePolicy* (class in *tfmdp.policy.drp*), 5)
`__call__()` (*tfmdp.policy.feedforward.FeedforwardPolicy* (class in *tfmdp.policy.feedforward*), 6)
`__getnewargs__()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)
`__new__()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)
`__repr__()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)
`_asdict()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)
`_build_hidden_layers()` (*tfmdp.policy.feedforward.FeedforwardPolicy* (class in *tfmdp.policy.feedforward*), 6)
`_build_input_layer()` (*tfmdp.policy.feedforward.FeedforwardPolicy* (class in *tfmdp.policy.feedforward*), 6)
`_build_output_layer()` (*tfmdp.policy.feedforward.FeedforwardPolicy* (class in *tfmdp.policy.feedforward*), 6)
`_get_output_tensor()` (*tfmdp.policy.layers.action_layer.ActionLayer* (class in *tfmdp.policy.layers.action_layer*), 3)
`_make()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)
`_replace()` (*tfmdp.model.cell.basic_cell.OutputTuple* (class in *tfmdp.model.cell.basic_cell*), 2)

A

`action` (*tfmdp.model.cell.basic_cell.OutputTuple* attribute), 2
`action_size` (*tfmdp.model.cell.basic_cell.BasicMarkovCell* attribute), 2
`ActionLayer` (class in *tfmdp.policy.layers.action_layer*), 3

B

`BasicMarkovCell` (class in *tfmdp.model.cell.basic_cell*), 1
`build()` (*tfmdp.policy.drp.DeepReactivePolicy* (class in *tfmdp.policy.drp*), 5)
`build()` (*tfmdp.policy.feedforward.FeedforwardPolicy* (class in *tfmdp.policy.feedforward*), 6)

C

`call()` (*tfmdp.policy.layers.action_layer.ActionLayer* (class in *tfmdp.policy.layers.action_layer*), 4)
`call()` (*tfmdp.policy.layers.state_layer.StateLayer* (class in *tfmdp.policy.layers.state_layer*), 4)
`cell_size()` (in module *tfmdp.model.cell.basic_cell*), 2

D

`DeepReactivePolicy` (class in *tfmdp.policy.drp*), 4

F

`FeedforwardPolicy` (class in *tfmdp.policy.feedforward*), 6
`from_json()` (*tfmdp.policy.drp.DeepReactivePolicy* (class in *tfmdp.policy.drp*), 5)

G

`get_params_string()` (in module *tfmdp.utils*), 7
`graph` (*tfmdp.model.cell.basic_cell.BasicMarkovCell* attribute), 2
`graph` (*tfmdp.policy.drp.DeepReactivePolicy* attribute), 5

I

`interm_size` (*tfmdp.model.cell.basic_cell.BasicMarkovCell* attribute), 2
`interms` (*tfmdp.model.cell.basic_cell.OutputTuple* attribute), 2

N

`name` (*tfmdp.policy.drp.DeepReactivePolicy* attribute), 5

name (*tfmdp.policy.feedforward.FeedforwardPolicy* attribute), 6

O

output_size (*tfmdp.model.cell.basic_cell.BasicMarkovCell* attribute), 2

OutputTuple (class in *tfmdp.model.cell.basic_cell*), 2

R

restore() (*tfmdp.policy.drp.DeepReactivePolicy* method), 5

reward (*tfmdp.model.cell.basic_cell.OutputTuple* attribute), 2

S

save() (*tfmdp.policy.drp.DeepReactivePolicy* method), 5

size (*tfmdp.policy.drp.DeepReactivePolicy* attribute), 5

size (*tfmdp.policy.feedforward.FeedforwardPolicy* attribute), 6

state (*tfmdp.model.cell.basic_cell.OutputTuple* attribute), 2

state_size (*tfmdp.model.cell.basic_cell.BasicMarkovCell* attribute), 2

StateLayer (class in *tfmdp.policy.layers.state_layer*), 4

summary() (*tfmdp.policy.drp.DeepReactivePolicy* method), 6

T

tfmdp (module), 7

tfmdp.model (module), 3

tfmdp.model.cell (module), 3

tfmdp.model.cell.basic_cell (module), 1

tfmdp.planning (module), 3

tfmdp.policy (module), 7

tfmdp.policy.drp (module), 4

tfmdp.policy.feedforward (module), 6

tfmdp.policy.layers (module), 4

tfmdp.policy.layers.action_layer (module), 3

tfmdp.policy.layers.state_layer (module), 4

tfmdp.utils (module), 7

to_json() (*tfmdp.policy.drp.DeepReactivePolicy* method), 6

to_tensor() (in module *tfmdp.model.cell.basic_cell*), 2

trainable_variables
(*tfmdp.policy.layers.action_layer.ActionLayer* attribute), 4

trainable_variables
(*tfmdp.policy.layers.state_layer.StateLayer* attribute), 4

V

vars (*tfmdp.policy.drp.DeepReactivePolicy* attribute), 6

vars (*tfmdp.policy.feedforward.FeedforwardPolicy* attribute), 6