
Tensorforce Documentation

Release 0.5.0

Tensorforce Team

Sep 08, 2019

1	Installation	3
2	Getting started	5
2.1	Training	5
2.2	Evaluation / application	6
2.3	Runner utility	6
3	Module specification	7
3.1	How to specify modules	7
3.2	Static vs dynamic hyperparameters	8
4	run.py – Runner	9
4.1	Required arguments	9
4.2	Optional arguments	9
5	tune.py – Hyperparameter tuner	11
5.1	Required arguments	11
5.2	Optional arguments	11
6	Agent interface	13
7	Constant Agent	17
8	Random Agent	19
9	Tensorforce Policy Agent	21
10	Deep Q-Network	25
11	Vanilla Policy Gradient	27
12	Proximal Policy Optimization	29
13	Trust-Region Policy Optimization	31
14	Distributions	33
15	Layers	35
15.1	Convolutional layers	35

15.2	Dense layers	36
15.3	Embedding layers	37
15.4	Recurrent layers	38
15.5	Pooling layers	40
15.6	Normalization layers	41
15.7	Misc layers	41
15.8	Layers with internal states	43
15.9	Special layers	45
16	Memories	47
17	Networks	49
18	Objectives	51
19	Optimizers	53
20	Parameters	59
21	Preprocessing	63
22	Policies	67
23	Environment interface	69
24	Arcade Learning Environment	71
25	Maze Explorer	73
26	Open Sim	75
27	OpenAI Gym	77
28	OpenAI Retro	79
29	PyGame Learning Environment	81
30	ViZDoom	83
	Index	85

Tensorforce is an open-source deep reinforcement learning framework, with an emphasis on modularized flexible library design and straightforward usability for applications in research and practice. Tensorforce is built on top of [Google's TensorFlow framework](#) and compatible with Python 3 (Python 2 support was dropped with version 0.5).

Tensorforce follows a set of high-level design choices which differentiate it from other similar libraries:

- **Modular component-based design:** Feature implementations, above all, strive to be as generally applicable and configurable as possible, potentially at some cost of faithfully resembling details of the introducing paper.
- **Separation of RL algorithm and application:** Algorithms are agnostic to the type and structure of inputs (states/observations) and outputs (actions/decisions), as well as the interaction with the application environment.
- **Full-on TensorFlow models:** The entire reinforcement learning logic, including control flow, is implemented in TensorFlow, to enable portable computation graphs independent of application programming language, and to facilitate the deployment of models.

CHAPTER 1

Installation

A stable version of Tensorforce is periodically updated on PyPI and installed as follows:

```
pip install tensorforce
```

To always use the latest version of Tensorforce, install the GitHub version instead:

```
git clone https://github.com/tensorforce/tensorforce.git
cd tensorforce
pip install -e .
```

Tensorforce is built on top of [Google's TensorFlow](#) and requires that either `tensorflow` or `tensorflow-gpu` is installed, currently as version 1.13.1. To include the correct version of TensorFlow with the installation of Tensorforce, simply add the flag `tf` for the normal CPU version or `tf_gpu` for the GPU version:

```
# PyPI version plus TensorFlow CPU version
pip install tensorforce[tf]

# GitHub version plus TensorFlow GPU version
pip install -e .[tf_gpu]
```

Some environments require additional packages, for which there are also options available (`mazeexp`, `gym`, `retro`, `vizdoom`; or `envs` for all environments), however, some require other tools to be installed (see [environments documentation](#)).

2.1 Training

```
from tensorforce.agents import Agent
from tensorforce.environments import Environment

# Setup environment
# (Tensorforce or custom implementation, ideally using the Environment interface)
environment = Environment.create(environment='environment.json')

# Create and initialize agent
agent = Agent.create(agent='agent.json', environment=environment)
agent.initialize()

# Reset agent and environment at the beginning of a new episode
agent.reset()
states = environment.reset()
terminal = False

# Agent-environment interaction training loop
while not terminal:
    actions = agent.act(states=states)
    states, terminal, reward = environment.execute(actions=actions)
    agent.observe(terminal=terminal, reward=reward)

# Close agent and environment
agent.close()
environment.close()
```

2.2 Evaluation / application

```
# Agent-environment interaction evaluation loop
while not terminal:
    actions = agent.act(states=states, evaluation=True)
    states, terminal, reward = environment.execute(actions=actions)
```

2.3 Runner utility

```
from tensorforce.execution import Runner

# Tensorforce runner utility
runner = Runner(agent='agent.json', environment='environment.json')

# Run training
runner.run(num_episodes=500)

# Close runner
runner.close()
```

Module specification

Agents are instantiated via `Agent.create(agent=...)`, with either of the specification alternatives presented below (agent acts as type argument). It is recommended to pass as second argument `environment` the application Environment implementation, which automatically extracts the corresponding states, actions and `max_episode_timesteps` arguments of the agent.

3.1 How to specify modules

3.1.1 Dictionary with module type and arguments

```
Agent.create(...
    network=dict(type='layered', layers=[dict(type='dense', size=32)]),
    memory=dict(type='replay', capacity=10000), ...
)
```

3.1.2 JSON specification file (plus additional arguments)

```
Agent.create(...
    network='network.json',
    memory=dict(type='memory.json', capacity=10000), ...
)
```

3.1.3 Module path (plus additional arguments)

```
Agent.create(...
    network='my_module.TestNetwork',
    memory=dict(type='tensorflow.core.memories.Replay', capacity=10000), ...
)
```

3.1.4 Callable or Type (plus additional arguments)

```
Agent.create(...
    network=TestNetwork,
    memory=dict(type=Replay, capacity=10000), ...
)
```

3.1.5 Default module: only arguments or first argument

```
Agent.create(...
    network=[dict(type='dense', size=32)],
    memory=dict(capacity=10000), ...
)
```

3.2 Static vs dynamic hyperparameters

Tensorforce distinguishes between agent/module arguments (primitive types: bool/int/long/float) which specify either part of the TensorFlow model architecture, like the layer size, or a value within the architecture, like the learning rate. Whereas the former are statically defined as part of the agent initialization, the latter can be dynamically adjusted afterwards. These dynamic hyperparameters are indicated by `parameter` as part of their type specification in the documentation, and can alternatively be assigned a `parameter module` instead of a constant value, for instance, to specify a decaying learning rate.

4.1 Required arguments

#1: **agent** (*string*) – Agent (configuration JSON file, name, or library module)

#2: **environment** (*string*) – Environment (name, configuration JSON file, or library module)

4.2 Optional arguments

4.2.1 Agent arguments

–**[n]etwork** (*string, default: not specified*) – Network (configuration JSON file, name, or library module)

4.2.2 Environment arguments

–**[l]evel** (*string, default: not specified*) – Level or game id, like `CartPole-v1`, if supported

–**[i]mport-modules** (*string, default: not specified*) – Import comma-separated modules required for environment

–**visualize** (*bool, default: false*) – Visualize agent–environment interaction, if supported

4.2.3 Runner arguments

–**[t]imesteps** (*int, default: not specified*) – Number of timesteps

–**[e]pisodes** (*int, default: not specified*) – Number of episodes

–**[m]ax-episode-timesteps** (*int, default: not specified*) – Maximum number of timesteps per episode

–**mean-horizon** (*int, default: 10*) – Number of timesteps/episodes for mean reward computation

–**[v]aluation** (*bool, default: false*) – Evaluation mode

-[s]ave-best-agent (*bool, default: false*) – Save best-performing agent

4.2.4 Logging arguments

-[r]eppeat (*int, default: 1*) – Number of repetitions

-[p]ath (*string, default: not specified*) – Logging path, directory plus filename without extension

-seaborn (*bool, default: false*) – Use seaborn

tune.py – Hyperparameter tuner

5.1 Required arguments

#1: **environment** (*string*) – Environment (name, configuration JSON file, or library module)

5.2 Optional arguments

–**[l]evel** (*string*, *default: not specified*) – Level or game id, like `CartPole-v1`, if supported

–**[m]ax-repeats** (*int*, *default: 1*) – Maximum number of repetitions

–**[n]um-iterations** (*int*, *default: 1*) – Number of BOHB iterations

–**[d]irectory** (*string*, *default: “tuner”*) – Output directory

–**[r]estore** (*string*, *default: not specified*) – Restore from given directory

–**id** (*string*, *default: “worker”*) – Unique worker id

Agent interface

```
class tensorforce.agents.Agent (states, actions, max_episode_timesteps=None, parallel_interactions=1, buffer_observe=True, seed=None, recorder=None)
```

Tensorforce agent interface.

```
act (states, parallel=0, deterministic=False, independent=False, evaluation=False, query=None, **kwargs)
```

Returns action(s) for the given state(s), needs to be followed by `observe (. . .)` unless `independent` is true.

Parameters

- **states** (*dict[state]*) – Dictionary containing state(s) to be acted on ().
- **parallel** (*int*) – Parallel execution index (: 0).
- **deterministic** (*bool*) – Whether to apply exploration and sampling (: false).
- **independent** (*bool*) – Whether action is not remembered, and this call is thus not followed by `observe` (: false).
- **evaluation** (*bool*) – Whether the agent is currently evaluated, implies and overwrites `deterministic` and `independent` (: false).
- **query** (*list[str]*) – Names of tensors to retrieve (: none).
- **kwargs** – Additional input values, for instance, for dynamic hyperparameters.

Returns Dictionary containing action(s), plus queried tensor values if requested.

Return type (dict[action], plus optional list[str])

```
close ()
```

Closes the agent.

```
static create (agent=None, environment=None, **kwargs)
```

Creates an agent from a specification.

Parameters

- **agent** (*specification*) – JSON file, specification key, configuration dictionary, library module, or Agent subclass (: Policy agent).
- **environment** (*Environment*) – Environment which the agent is supposed to be trained on, environment-related arguments like state/action space specifications will be extract if given.
- **kwargs** – Additional arguments.

get_available_summaries ()

Returns the summary labels provided by the agent.

Returns Available summary labels.

Return type list[str]

get_output_tensors (*function*)

Returns the names of output tensors for the given function.

Parameters **function** (*str*) – Function name ().

Returns Names of output tensors.

Return type list[str]

get_query_tensors (*function*)

Returns the names of queryable tensors for the given function.

Parameters **function** (*str*) – Function name ().

Returns Names of queryable tensors.

Return type list[str]

initialize ()

Initializes the agent.

observe (*reward*, *terminal=False*, *parallel=0*, *query=None*, ***kwargs*)

Observes reward and whether a terminal state is reached, needs to be preceded by `act (. . .)`.

Parameters

- **reward** (*float*) – Reward ().
- **terminal** (*bool | 0 | 1 | 2*) – Whether a terminal state is reached or 2 if the episode was aborted (: false).
- **parallel** (*int*) – Parallel execution index (: 0).
- **query** (*list[str]*) – Names of tensors to retrieve (: none).
- **kwargs** – Additional input values, for instance, for dynamic hyperparameters.

Returns Whether an update was performed, plus queried tensor values if requested.

Return type (bool, optional list[str])

reset ()

Resets the agent to start a new episode.

restore (*directory=None*, *filename=None*)

Restores the agent.

Parameters

- **directory** (*str*) – Checkpoint directory (: directory specified for TensorFlow saver).
- **filename** (*str*) – Checkpoint filename (: latest checkpoint in directory).

save (*directory=None, filename=None, append_timestep=True*)

Saves the current state of the agent.

Parameters

- **directory** (*str*) – Checkpoint directory (: directory specified for TensorFlow saver).
- **filename** (*str*) – Checkpoint filename (: filename specified for TensorFlow saver).
- **append_timestep** – Whether to append the current timestep to the checkpoint file (: true).

Returns Checkpoint path.

Return type str

Constant Agent

```
class tensorflowforce.agents.ConstantAgent (states, actions, max_episode_timesteps=None, action_values=None, name='agent', device=None, summarizer=None, seed=None, recorder=None)
    Agent returning constant action values (specification key: constant).
```

Parameters

- **states** (*specification*) – States specification (**required**), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (**required**), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – ?
- **action_values** (*dict[value]*) – Constant value per action (: false for binary boolean actions, 0 for discrete integer actions, 0.0 for continuous actions).
- **seed** (*int*) – Random seed to set for Python, NumPy and TensorFlow (: none).
- **name** (*string*) – Agent name, used e.g. for TensorFlow scopes (: “agent”).
- **device** (*string*) – Device name (: TensorFlow default).
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer):

Random Agent

```
class tensorflowforce.agents.RandomAgent (states, actions, max_episode_timesteps=None,  
                                         name='agent', device=None, summarizer=None,  
                                         seed=None, recorder=None)
```

Agent returning random action values (specification key: `random`).

Parameters

- **states** (*specification*) – States specification (**required**), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (**required**), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – ?
- **seed** (*int*) – Random seed to set for Python, NumPy and TensorFlow (: none).
- **name** (*string*) – Agent name, used e.g. for TensorFlow scopes (: “agent”).
- **device** (*string*) – Device name (: TensorFlow default).
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer):

Tensorforce Policy Agent

```
class tensorforce.agents.PolicyAgent (states, actions, update, objective, reward_estimation,
                                     max_episode_timesteps=None, policy=None, net-
                                     work='auto', memory=None, optimizer='adam',
                                     baseline_policy=None, baseline_network=None, base-
                                     line_optimizer=None, baseline_objective=None,
                                     preprocessing=None, exploration=0.0, vari-
                                     able_noise=0.0, l2_regularization=0.0, en-
                                     tropy_regularization=0.0, name='agent',
                                     device=None, parallel_interactions=1,
                                     buffer_observe=True, seed=None, execution=None,
                                     saver=None, summarizer=None, recorder=None)
```

Policy Agent (specification key: `policy`).

Base class for a broad class of deep reinforcement learning agents, which act according to a policy parametrized by a neural network, leverage a memory module for periodic updates based on batches of experience, and optionally employ a baseline/critic/target policy for improved reward estimation.

Parameters

- **states** (*specification*) – States specification (), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Maximum number of timesteps per episode (: not given).
- **policy** (*specification*) – Policy configuration, currently best to ignore and use the *network* argument instead.
- **network** (*"auto" | specification*) – Policy network configuration, see [networks](#) (: “auto”, automatically configured network).

- **memory** (*int* | *specification*) – Memory configuration, see [memories](#) (: replay memory with given or inferred capacity).
- **update** (*int* | *specification*) – Model update configuration with the following attributes (: timesteps batch size):
- **optimizer** (*specification*) – Optimizer configuration, see [optimizers](#) (: Adam optimizer).
- **objective** (*specification*) – Optimization objective configuration, see [objectives](#) ().
- **reward_estimation** (*specification*) – Reward estimation configuration with the following attributes ():
- **baseline_policy** ("same" | "equal" | *specification*) – Baseline policy configuration, “same” refers to reusing the main policy as baseline, “equal” refers to using the same configuration as the main policy (: none).
- **baseline_network** ("same" | "equal" | *specification*) – Baseline network configuration, see [networks](#), “same” refers to reusing the main network as part of the baseline policy, “equal” refers to using the same configuration as the main network (: none).
- **baseline_optimizer** ("same" | "equal" | *specification*) – Baseline optimizer configuration, see [optimizers](#), “same” refers to reusing the main optimizer for the baseline, “equal” refers to using the same configuration as the main optimizer (: none).
- **baseline_objective** ("same" | "equal" | *specification*) – Baseline optimization objective configuration, see [objectives](#), “same” refers to reusing the main objective for the baseline, “equal” refers to using the same configuration as the main objective (: none).
- **preprocessing** (*dict*[*specification*]) – Preprocessing as layer or list of layers, see [preprocessing](#), specified per state-type or -name and for reward (: none).
- **exploration** (*parameter* | *dict*[*parameter*], *float* >= 0.0) – Exploration, global or per action, defined as the probability for uniformly random output in case of *bool* and *int* actions, and the standard deviation of Gaussian noise added to every output in case of *float* actions (: 0.0).
- **variable_noise** (*parameter*, *float* >= 0.0) – Standard deviation of Gaussian noise added to all trainable float variables (: 0.0).
- **l2_regularization** (*parameter*, *float* >= 0.0) – Scalar controlling L2 regularization (: 0.0).
- **entropy_regularization** (*parameter*, *float* >= 0.0) – Scalar controlling entropy regularization, to discourage the policy distribution being too “certain” / spiked (: 0.0).
- **name** (*string*) – Agent name, used e.g. for TensorFlow scopes (: “agent”).
- **device** (*string*) – Device name (: TensorFlow default).
- **parallel_interactions** (*int* > 0) – Maximum number of parallel interactions to support, for instance, to enable multiple parallel episodes, environments or (centrally controlled) agents within an environment (: 1).
- **buffer_observe** (*bool* | *int* > 0) – Maximum number of timesteps within an episode to buffer before executing internal observe operations, to reduce calls to TensorFlow for improved performance (: max_episode_timesteps or 1000, unless summarizer specified).
- **seed** (*int*) – Random seed to set for Python, NumPy and TensorFlow (: none).

- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer):
- **recorder** (*specification*) – Experience traces recorder configuration with the following attributes (: no recorder):

CHAPTER 10

Deep Q-Network

```
class tensorforce.agents.DeepQNetwork(states, actions, max_episode_timesteps=None,  
                                     network='auto', memory=10000, batch_size=32,  
                                     update_frequency=4, start Updating=None, learn-  
                                     ing_rate=0.0003, huber_loss=0.0, n_step=0,  
                                     discount=0.99, estimate_terminal=False,  
                                     target_sync_frequency=10000, tar-  
                                     get_update_weight=1.0, preprocessing=None,  
                                     exploration=0.0, variable_noise=0.0,  
                                     l2_regularization=0.0, entropy_regularization=0.0,  
                                     name='agent', device=None, paral-  
                                     lel_interactions=1, seed=None, execution=None,  
                                     saver=None, summarizer=None, recorder=None)
```

Deep Q-Network agent (specification key: `dqn`).

Vanilla Policy Gradient

```
class tensorforce.agents.VanillaPolicyGradient(states, actions, max_episode_timesteps,  
                                              network='auto', batch_size=10,  
                                              update_frequency=None, learning  
                                              rate=0.0003, discount=0.99,  
                                              estimate_terminal=False,  
                                              critic_network=None,  
                                              critic_optimizer=None, preprocess-  
                                              ing=None, exploration=0.0, vari-  
                                              able_noise=0.0, l2_regularization=0.0,  
                                              entropy_regularization=0.0,  
                                              name='agent', device=None, par-  
                                              allel_interactions=1, seed=None,  
                                              execution=None, saver=None, summa-  
                                              rizer=None, recorder=None)
```

Vanilla Policy Gradient agent (specification key: vpg).

Proximal Policy Optimization

```
class tensorforce.agents.ProximalPolicyOptimization(states, actions,  
                                                    max_episode_timesteps, net-  
                                                    work='auto', batch_size=10,  
                                                    update_frequency=None,  
                                                    learning_rate=0.0003, sub-  
                                                    sampling_fraction=0.33,  
                                                    optimization_steps=10, like-  
                                                    lihood_ratio_clipping=0.2,  
                                                    discount=0.99, esti-  
                                                    mate_terminal=False,  
                                                    critic_network=None,  
                                                    critic_optimizer=None, pre-  
                                                    processing=None, explo-  
                                                    ration=0.0, variable_noise=0.0,  
                                                    l2_regularization=0.0, en-  
                                                    tropy_regularization=0.0,  
                                                    name='agent', device=None,  
                                                    parallel_interactions=1,  
                                                    seed=None, execution=None,  
                                                    saver=None, summarizer=None,  
                                                    recorder=None)
```

Proximal Policy Optimization agent (specification key: ppo).

Trust-Region Policy Optimization

```
class tensorforce.agents.TrustRegionPolicyOptimization(states, actions,
                                                    max_episode_timesteps,
                                                    network='auto',
                                                    batch_size=10, up-
                                                    date_frequency=None,
                                                    learning_rate=0.001, like-
                                                    lihood_ratio_clipping=0.2,
                                                    discount=0.99, esti-
                                                    mate_terminal=False,
                                                    critic_network=None,
                                                    critic_optimizer=None,
                                                    preprocessing=None,
                                                    exploration=0.0,
                                                    variable_noise=0.0,
                                                    l2_regularization=0.0,
                                                    entropy_regularization=0.0,
                                                    name='agent', de-
                                                    vice=None, paral-
                                                    lel_interactions=1,
                                                    seed=None, execu-
                                                    tion=None, saver=None,
                                                    summarizer=None,
                                                    recorder=None)
```

Trust Region Policy Optimization agent (specification key: `trpo`).


```
class tensorforce.core.distributions.Bernoulli(name, action_spec, embedding_size,  
                                              summary_labels=None)  
    Bernoulli distribution, for binary boolean actions (specification key: bernoulli).
```

Parameters

- **name** (*string*) – Distribution name ().
- **action_spec** (*specification*) – Action specification ().
- **embedding_size** (*int* > 0) – Embedding size ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.distributions.Beta(name, action_spec, embedding_size, sum-  
                                         mary_labels=None)  
    Beta distribution, for bounded continuous actions (specification key: beta).
```

Parameters

- **name** (*string*) – Distribution name ().
- **action_spec** (*specification*) – Action specification ().
- **embedding_size** (*int* > 0) – Embedding size ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.distributions.Categorical(name, action_spec, embedding_size,  
                                                  summary_labels=None)  
    Categorical distribution, for discrete integer actions (specification key: categorical).
```

Parameters

- **name** (*string*) – Distribution name ().
- **action_spec** (*specification*) – Action specification ().
- **embedding_size** (*int* > 0) – Embedding size ().

- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.distributions.**Gaussian**(*name, action_spec, embedding_size, summary_labels=None*)

Gaussian distribution, for unbounded continuous actions (specification key: `gaussian`).

Parameters

- **name** (*string*) – Distribution name ().
- **action_spec** (*specification*) – Action specification ().
- **embedding_size** (*int > 0*) – Embedding size ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

Default layer: Function with default argument function

15.1 Convolutional layers

```
class tensorflow.core.layers.Conv1d(name, size, window=3, stride=1, padding='same',
                                     bias=True, activation='relu', dropout=0.0,
                                     is_trainable=True, use_cudnn_on_gpu=True,
                                     input_spec=None, summary_labels=None,
                                     l2_regularization=None)
```

1-dimensional convolutional layer (specification key: conv1d).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **size** (*int* ≥ 0) – Layer output size, 0 implies additionally removing the axis ().
- **window** (*int* > 0) – Window size (: 3).
- **stride** (*int* > 0) – Stride size (: 1).
- **padding** ('same' | 'valid') – Padding type, see [TensorFlow docs](#) (: 'same').
- **bias** (*bool*) – Whether to add a trainable bias variable (: true).
- ('crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (*activation*) – 'softmax' | 'softplus' | 'softsign' | 'swish' | 'tanh'): Activation nonlinearity (: "relu").
- **dropout** (*parameter*, $0.0 \leq \text{float} < 1.0$) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **use_cudnn_on_gpu** (*bool*) – Whether to use cuDNN on GPU (: true).
- **input_spec** (*specification*) – Input tensor specification ().

- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorforce.core.layers.Conv2d(name, size, window=3, stride=1, padding='same',
                                     dilation=1, bias=True, activation='relu', dropout=0.0,
                                     is_trainable=True,          use_cudnn_on_gpu=True,
                                     input_spec=None,            summary_labels=None,
                                     l2_regularization=None)
```

2-dimensional convolutional layer (specification key: conv2d).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **size** (*int >= 0*) – Layer output size, 0 implies additionally removing the axis ().
- **window** (*int > 0 | (int > 0, int > 0)*) – Window size (: 3).
- **stride** (*int > 0 | (int > 0, int > 0)*) – Stride size (: 1).
- **padding** (*'same' | 'valid'*) – Padding type, see [TensorFlow docs](#) (: 'same').
- **dilation** (*int > 0 | (int > 0, int > 0)*) – Dilation value (: 1).
- **bias** (*bool*) – Whether to add a trainable bias variable (: true).
- (*'crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (activation)*) – 'softmax' | 'softplus' | 'softsign' | 'swish' | 'tanh': Activation nonlinearity (: "relu").
- **dropout** (*parameter, 0.0 <= float < 1.0*) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **use_cudnn_on_gpu** (*bool*) – Whether to use cuDNN on GPU (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).

15.2 Dense layers

```
class tensorforce.core.layers.Dense(name, size, bias=True, activation='relu', dropout=0.0,
                                    is_trainable=True,          input_spec=None,      sum-
                                    mary_labels=None, l2_regularization=None)
```

Dense fully-connected layer (specification key: dense).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **size** (*int >= 0*) – Layer output size, 0 implies additionally removing the axis ().
- **bias** (*bool*) – Whether to add a trainable bias variable (: true).

- (`'crelu'` | `'elu'` | `'leaky-relu'` | `'none'` | `'relu'` | `'selu'` | `'sigmoid'` | *(activation)* – `'softmax'` | `'softplus'` | `'softsign'` | `'swish'` | `'tanh'`): Activation nonlinearity (: `"relu"`).
- `dropout` (*parameter*, $0.0 \leq \text{float} < 1.0$) – Dropout rate (: 0.0).
- `is_trainable` (*bool*) – Whether layer variables are trainable (: `true`).
- `input_spec` (*specification*) – Input tensor specification ().
- `summary_labels` (`'all'` | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- `l2_regularization` ($\text{float} \geq 0.0$) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorforce.core.layers.Linear(name, size, bias=True, is_trainable=True,
                                     input_spec=None, summary_labels=None,
                                     l2_regularization=None)
```

Linear layer (specification key: `linear`).

Parameters

- `name` (*string*) – Layer name (: internally chosen).
- `size` ($\text{int} \geq 0$) – Layer output size, 0 implies additionally removing the axis ().
- `bias` (*bool*) – Whether to add a trainable bias variable (: `true`).
- `is_trainable` (*bool*) – Whether layer variables are trainable (: `true`).
- `input_spec` (*specification*) – Input tensor specification ().
- `summary_labels` (`'all'` | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- `l2_regularization` ($\text{float} \geq 0.0$) – Scalar controlling L2 regularization (: inherit value of parent module).

15.3 Embedding layers

```
class tensorforce.core.layers.Embedding(name, size, num_embeddings=None, partition_strategy='mod',
                                         max_norm=None, bias=False, activation='tanh', dropout=0.0,
                                         is_trainable=True, input_spec=None, summary_labels=None, l2_regularization=None)
```

Embedding layer (specification key: `embedding`).

Parameters

- `name` (*string*) – Layer name (: internally chosen).
- `size` ($\text{int} \geq 0$) – Layer output size, 0 implies additionally removing the axis ().
- `num_embeddings` ($\text{int} > 0$) – If set, specifies the number of embeddings (: `none`).
- `partition_strategy` (`'mod'` | `'div'`) – Partitioning strategy, see [TensorFlow docs](#) (: `'mod'`).
- `max_norm` (*float*) – If set, embeddings are clipped if their L2-norm is larger (: `none`).
- `bias` (*bool*) – Whether to add a trainable bias variable (: `false`).

- (`'crelu'` | `'elu'` | `'leaky-relu'` | `'none'` | `'relu'` | `'selu'` | `'sigmoid'` | (*activation*) – `'softmax'` | `'softplus'` | `'softsign'` | `'swish'` | `'tanh'`): Activation nonlinearity (: `"tanh"`).
- **dropout** (*parameter*, $0.0 \leq \text{float} < 1.0$) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: `true`).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (`'all'` | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** ($\text{float} \geq 0.0$) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for potential parent class.

15.4 Recurrent layers

```
class tensorforce.core.layers.Gru(name, size, return_final_state=True, bias=False,
                                  activation=None, dropout=0.0, is_trainable=True,
                                  input_spec=None, summary_labels=None,
                                  l2_regularization=None, **kwargs)
```

Gated recurrent unit layer (specification key: `gru`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** (`'gru'` | `'lstm'`) – The recurrent cell type ().
- **size** ($\text{int} \geq 0$) – Layer output size, 0 implies additionally removing the axis ().
- **return_final_state** (*bool*) – Whether to return the final state instead of the per-step outputs (: `true`).
- **bias** (*bool*) – Whether to add a trainable bias variable (: `false`).
- (`'crelu'` | `'elu'` | `'leaky-relu'` | `'none'` | `'relu'` | `'selu'` | `'sigmoid'` | (*activation*) – `'softmax'` | `'softplus'` | `'softsign'` | `'swish'` | `'tanh'`): Activation nonlinearity (: `none`).
- **dropout** (*parameter*, $0.0 \leq \text{float} < 1.0$) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: `true`).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (`'all'` | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** ($\text{float} \geq 0.0$) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras GRU layer, see [TensorFlow docs](#).

```
class tensorforce.core.layers.Lstm(name, size, return_final_state=True, bias=False,
                                   activation=None, dropout=0.0, is_trainable=True,
                                   input_spec=None, summary_labels=None,
                                   l2_regularization=None, **kwargs)
```

Long short-term memory layer (specification key: `lstm`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** (*'gru' | 'lstm'*) – The recurrent cell type ().
- **size** (*int >= 0*) – Layer output size, 0 implies additionally removing the axis ().
- **return_final_state** (*bool*) – Whether to return the final state instead of the per-step outputs (: true).
- **bias** (*bool*) – Whether to add a trainable bias variable (: false).
- (*'crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (activation)*) – ‘softmax’ | ‘softplus’ | ‘softsign’ | ‘swish’ | ‘tanh’): Activation nonlinearity (: none).
- **dropout** (*parameter, 0.0 <= float < 1.0*) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras LSTM layer, see [TensorFlow docs](#).

```
class tensorforce.core.layers.Rnn(name, cell, size, return_final_state=True, bias=False,
                                activation=None, dropout=0.0, is_trainable=True,
                                input_spec=None, summary_labels=None,
                                l2_regularization=None, **kwargs)
```

Recurrent neural network layer (specification key: rnn).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** (*'gru' | 'lstm'*) – The recurrent cell type ().
- **size** (*int >= 0*) – Layer output size, 0 implies additionally removing the axis ().
- **return_final_state** (*bool*) – Whether to return the final state instead of the per-step outputs (: true).
- **bias** (*bool*) – Whether to add a trainable bias variable (: false).
- (*'crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (activation)*) – ‘softmax’ | ‘softplus’ | ‘softsign’ | ‘swish’ | ‘tanh’): Activation nonlinearity (: none).
- **dropout** (*parameter, 0.0 <= float < 1.0*) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras RNN layer, see [TensorFlow docs](#).

15.5 Pooling layers

class tensorforce.core.layers.**Flatten**(*name*, *input_spec=None*, *summary_labels=None*)
Flatten layer (specification key: `flatten`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.**Pooling**(*name*, *reduction*, *input_spec=None*, *summary_labels=None*)
Pooling layer (global pooling) (specification key: `pooling`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **reduction** ('concat' | 'max' | 'mean' | 'product' | 'sum') – Pooling type ().
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.**Pool1d**(*name*, *reduction*, *window=2*, *stride=2*, *padding='same'*, *input_spec=None*, *summary_labels=None*)
1-dimensional pooling layer (local pooling) (specification key: `pool1d`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **reduction** ('average' | 'max') – Pooling type ().
- **window** (*int* > 0) – Window size (: 2).
- **stride** (*int* > 0) – Stride size (: 2).
- **padding** ('same' | 'valid') – Padding type, see [TensorFlow docs](#) (: 'same').
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.**Pool2d**(*name*, *reduction*, *window=2*, *stride=2*, *padding='same'*, *input_spec=None*, *summary_labels=None*)
2-dimensional pooling layer (local pooling) (specification key: `pool2d`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **reduction** ('average' | 'max') – Pooling type ().
- **window** (*int* > 0 | (*int* > 0, *int* > 0)) – Window size (: 2).
- **stride** (*int* > 0 | (*int* > 0, *int* > 0)) – Stride size (: 2).

- **padding** ('same' | 'valid') – Padding type, see [TensorFlow docs](#) (: 'same').
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

15.6 Normalization layers

```
class tensorforce.core.layers.ExponentialNormalization(name,          decay=0.999,
                                                    axes=None,          in-
                                                    put_spec=None,          sum-
                                                    mary_labels=None)
```

Normalization layer based on the exponential moving average (specification key: `exponential_normalization`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **decay** (*parameter*, $0.0 \leq \text{float} \leq 1.0$) – Decay rate (: 0.999).
- **axes** (*iter[int >= 0]*) – Normalization axes, excluding batch axis (: all but last axis).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorforce.core.layers.InstanceNormalization(name,          axes=None,          in-
                                                    put_spec=None,          sum-
                                                    mary_labels=None)
```

Instance normalization layer (specification key: `instance_normalization`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **axes** (*iter[int >= 0]*) – Normalization axes, excluding batch axis (: all).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

15.7 Misc layers

```
class tensorforce.core.layers.Activation(name, nonlinearity, input_spec=None, sum-
                                                    mary_labels=None)
```

Activation layer (specification key: `activation`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).

- ('crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | *nonlinearity*) – 'softmax' | 'softplus' | 'softsign' | 'swish' | 'tanh': `Nonlinearity()`.
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.Clipping (*name*, *upper*, *lower=None*, *input_spec=None*, *summary_labels=None*)

Clipping layer (specification key: clipping).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **upper** (*parameter*, *float*) – Upper clipping value ().
- **lower** (*parameter*, *float*) – Lower clipping value (: negative upper value).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.Deltafier (*name*, *concatenate=False*, *input_spec=None*, *summary_labels=None*)

Deltafier layer computing the difference between the current and the previous input; can only be used as preprocessing layer (specification key: deltafier).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **concatenate** (*False* | *int* >= 0) – Whether to concatenate instead of replace deltas with input, and if so, concatenation axis (: false).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.Dropout (*name*, *rate*, *input_spec=None*, *summary_labels=None*)

Dropout layer (specification key: dropout).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **rate** (*parameter*, *0.0* <= *float* < *1.0*) – Dropout rate ().
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class tensorforce.core.layers.Image (*name*, *height=None*, *width=None*, *grayscale=False*, *input_spec=None*, *summary_labels=None*)

Image preprocessing layer (specification key: image).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **height** (*int*) – Height of resized image (: no resizing or relative to width).

- **width** (*int*) – Width of resized image (: no resizing or relative to height).
- **grayscale** (*bool* | *iter[float]*) – Turn into grayscale image, optionally using given weights (: false).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Sequence(name, length, axis=-1, concatenate=True, input_spec=None, summary_labels=None)
```

Sequence layer stacking the current and previous inputs; can only be used as preprocessing layer (specification key: sequence).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **length** (*int* > 0) – Number of inputs to concatenate ().
- **axis** (*int* >= 0) – Concatenation axis, excluding batch axis (: last axis).
- **concatenate** (*bool*) – Whether to concatenate inputs at given axis, otherwise introduce new sequence axis (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

15.8 Layers with internal states

```
class tensorforce.core.layers.InternalGru(name, size, bias=False, activation=None, dropout=0.0, is_trainable=True, input_spec=None, summary_labels=None, l2_regularization=None, **kwargs)
```

Internal state GRU cell layer (specification key: internal_gru).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** ('gru' | 'lstm') – The recurrent cell type ().
- **size** (*int* >= 0) – Layer output size, 0 implies additionally removing the axis ().
- **length** (*parameter*, *long* > 0) – ???+1 ().
- **bias** (*bool*) – Whether to add a trainable bias variable (: false).
- ('crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | *activation*) – 'softmax' | 'softplus' | 'softsign' | 'swish' | 'tanh': Activation nonlinearity (: none).
- **dropout** (*parameter*, 0.0 <= *float* < 1.0) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

- **l2_regularization** (*float* ≥ 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras GRU layer, see [TensorFlow docs](#).

```
class tensorforce.core.layers.InternalLstm(name, size, bias=False, activation=None,
                                          dropout=0.0, is_trainable=True, input_spec=None,
                                          summary_labels=None, l2_regularization=None, **kwargs)
```

Internal state LSTM cell layer (specification key: `internal_lstm`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** (*'gru' | 'lstm'*) – The recurrent cell type ().
- **size** (*int* ≥ 0) – Layer output size, 0 implies additionally removing the axis ().
- **length** (*parameter, long* > 0) – $???+1$ ().
- **bias** (*bool*) – Whether to add a trainable bias variable (: false).
- (*'crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (activation)*) – ‘softmax’ | ‘softplus’ | ‘softsign’ | ‘swish’ | ‘tanh’): Activation nonlinearity (: none).
- **dropout** (*parameter, 0.0* \leq *float* < 1.0) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float* ≥ 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras LSTM layer, see [TensorFlow docs](#).

```
class tensorforce.core.layers.InternalRnn(name, cell, size, length, bias=False, activation=None,
                                          dropout=0.0, is_trainable=True, input_spec=None,
                                          summary_labels=None, l2_regularization=None, **kwargs)
```

Internal state RNN cell layer (specification key: `internal_rnn`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **cell** (*'gru' | 'lstm'*) – The recurrent cell type ().
- **size** (*int* ≥ 0) – Layer output size, 0 implies additionally removing the axis ().
- **length** (*parameter, long* > 0) – $???+1$ ().
- **bias** (*bool*) – Whether to add a trainable bias variable (: false).
- (*'crelu' | 'elu' | 'leaky-relu' | 'none' | 'relu' | 'selu' | 'sigmoid' | (activation)*) – ‘softmax’ | ‘softplus’ | ‘softsign’ | ‘swish’ | ‘tanh’): Activation nonlinearity (: none).
- **dropout** (*parameter, 0.0* \leq *float* < 1.0) – Dropout rate (: 0.0).
- **is_trainable** (*bool*) – Whether layer variables are trainable (: true).

- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (float >= 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras RNN cell layer, see [TensorFlow docs](#).

15.9 Special layers

```
class tensorforce.core.layers.Function (name,      function,      output_spec=None,      in-
                                         put_spec=None,          summary_labels=None,
                                         l2_regularization=None)
```

Custom TensorFlow function layer (specification key: function).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **function** (*lambda[x -> x]*) – TensorFlow function ().
- **output_spec** (*specification*) – Output tensor specification containing type and/or shape information (: same as input).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (float >= 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorforce.core.layers.Keras (name, layer, input_spec=None, summary_labels=None,
                                       l2_regularization=None, **kwargs)
```

Keras layer (specification key: keras).

Parameters

- **layer** (*string*) – Keras layer class name, see [TensorFlow docs](#) ().
- **kwargs** – Arguments for the Keras layer, see [TensorFlow docs](#).

```
class tensorforce.core.layers.Register (name,      tensor,      input_spec=None,      sum-
                                         mary_labels=None)
```

Tensor retrieval layer, which is useful when defining more complex network architectures which do not follow the sequential layer-stack pattern, for instance, when handling multiple inputs (specification key: register).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **tensor** (*string*) – Name under which tensor will be registered ().
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Retrieve (name, tensors, aggregation='concat', axis=0, input_spec=None, summary_labels=None)
```

Tensor retrieval layer, which is useful when defining more complex network architectures which do not follow the sequential layer-stack pattern, for instance, when handling multiple inputs (specification key: `retrieve`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **tensors** (*iter[string]*) – Names of global tensors to retrieve, for instance, state names or previously registered global tensor names ().
- **aggregation** (*'concat' | 'product' | 'stack' | 'sum'*) – Aggregation type in case of multiple tensors (: `'concat'`).
- **axis** (*int >= 0*) – Aggregation axis, excluding batch axis (: 0).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Reuse (name, layer, input_spec=None)
```

Reuse layer (specification key: `reuse`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **layer** (*string*) – Name of a previously defined layer ().
- **input_spec** (*specification*) – Input tensor specification ().

Default memory: Replay with default argument `capacity`

```
class tensorforce.core.memories.Recent (name, capacity, values_spec, device=None, summary_labels=None)
```

Batching memory which always retrieves most recent experiences (specification key: `recent`).

Parameters

- **name** (*string*) – Memory name ().
- **capacity** (*int* > 0) – Memory capacity, in experience timesteps ().
- **values_spec** (*specification*) – Values specification ().
- **device** (*string*) – Device name (: inherit value of parent module).
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.memories.Replay (name, capacity, values_spec, device=None, summary_labels=None)
```

Replay memory which randomly retrieves experiences (specification key: `replay`).

Parameters

- **name** (*string*) – Memory name ().
- **capacity** (*int* > 0) – Memory capacity, in experience timesteps ().
- **values_spec** (*specification*) – Values specification ().
- **device** (*string*) – Device name (: inherit value of parent module).
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

Default network: LayeredNetwork with default argument layers

```
class tensorflow.core.networks.AutoNetwork (name,          inputs_spec,          size=64,
                                             depth=2,          final_size=None,          fi-
                                             nal_depth=1,    internal_rnn=False,    de-
                                             vice=None,          summary_labels=None,
                                             l2_regularization=None)
```

Network which is automatically configured based on its input tensors, offering high-level customization (specification key: auto).

Parameters

- **name** (*string*) – Network name ().
- **inputs_spec** (*specification*) – Input tensors specification ().
- **size** (*int* > 0) – Layer size, before concatenation if multiple states (: 64).
- **depth** (*int* > 0) – Number of layers per state, before concatenation if multiple states (: 2).
- **final_size** (*int* > 0) – Layer size after concatenation if multiple states (: layer size).
- **final_depth** (*int* > 0) – Number of layers after concatenation if multiple states (: 1).
- **internal_rnn** (*false* | *parameter*, *long* >= 0) – Whether to add an internal state LSTM cell as last layer, and if so, horizon of the LSTM (: false).
- **device** (*string*) – Device name (: inherit value of parent module).
- **summary_labels** ('all' | *iter*[*string*]) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float* >= 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorflow.core.networks.LayeredNetwork (name,    layers,    inputs_spec,    de-
                                              vice=None,    summary_labels=None,
                                              l2_regularization=None)
```

Network consisting of Tensorforce layers, which can be specified as either a list of layer specifications in the case of a standard sequential layer-stack architecture, or as a list of list of layer specifications in the case of a more complex architecture consisting of multiple sequential layer-stacks (specification key: `custom` or `layered`).

Parameters

- **name** (*string*) – Network name ().
- **layers** (*iter[specification] | iter[iter[specification]]*) – Layers configuration, see [layers](#) ().
- **inputs_spec** (*specification*) – Input tensors specification ().
- **device** (*string*) – Device name (: inherit value of parent module).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).

Objectives

```
class tensorforce.core.objectives.ActionValue (name, huber_loss=0.0,
                                             mean_over_actions=False, summary_labels=None)
```

State-action-value / Q-value objective, which minimizes the L2-distance between the state-action-value estimate and target reward value (specification key: `action_value`).

Parameters

- **name** (*string*) – Module name ().
- **huber_loss** (*parameter, float > 0.0*) – Huber loss threshold (: no huber loss).
- **mean_over_actions** (*bool*) – Whether to compute objective for mean of state-action-values instead of per state-action-value (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.objectives.Plus (name, objective1, objective2, summary_labels=None)
```

Additive combination of two objectives (specification key: `plus`).

Parameters

- **name** (*string*) – Module name ().
- **objective1** (*specification*) – First objective configuration ().
- **objective2** (*specification*) – Second objective configuration ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.objectives.PolicyGradient (name, ratio_based=False,
                                                  clipping_value=0.0,
                                                  mean_over_actions=False, summary_labels=None)
```

Policy gradient objective, which maximizes the log-likelihood or likelihood-ratio scaled by the target reward value (specification key: `policy_gradient`).

Parameters

- **name** (*string*) – Module name ().
- **ratio_based** (*bool*) – Whether to scale the likelihood-ratio instead of the log-likelihood (: false).
- **clipping_value** (*parameter, float > 0.0*) – Clipping threshold for the maximized value (: no clipping).
- **mean_over_actions** (*bool*) – Whether to compute objective for mean of likelihoods instead of per likelihood (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.objectives.StateValue (name,                                huber_loss=0.0,  
                                              mean_over_actions=False,      sum-  
                                              mary_labels=None)
```

State-value objective, which minimizes the L2-distance between the state-value estimate and target reward value (specification key: `state_value`).

Parameters

- **name** (*string*) – Module name ().
- **huber_loss** (*parameter, float > 0.0*) – Huber loss threshold (: no huber loss).
- **mean_over_actions** (*bool*) – Whether to compute objective for mean of state-values instead of per state-value (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

Default optimizer: MetaOptimizerWrapper

```
class tensorforce.core.optimizers.ClippingStep(name, optimizer, threshold,
                                              mode='global_norm', summary_labels=None)
```

Clipping-step meta optimizer, which clips the updates of the given optimizer (specification key: `clipping_step`).

Parameters

- **name** (*string*) – Module name ().
- **optimizer** (*specification*) – Optimizer configuration ().
- **threshold** (*parameter, float > 0.0*) – Clipping threshold ().
- **mode** (*'global_norm' | 'norm' | 'value'*) – Clipping mode (: `'global_norm'`).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.Evolutionary(name, learning_rate, num_samples=1,
                                              unroll_loop=False, summary_labels=None)
```

Evolutionary optimizer, which samples random perturbations and applies them either as positive or negative update depending on their improvement of the loss (specification key: `evolutionary`).

Parameters

- **name** (*string*) – Module name ().
- **learning_rate** (*parameter, float > 0.0*) – Learning rate ().
- **num_samples** (*parameter, int > 0*) – Number of sampled perturbations (: 1).
- **unroll_loop** (*bool*) – Whether to unroll the sampling loop (: `false`).

- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.GlobalOptimizer(name, optimizer, summary_labels=None)
```

Global meta optimizer, which applies the given optimizer to the local variables, then applies the update to a corresponding set of global variables, and subsequently updates the local variables to the value of the global variables; will likely change in the future (specification key: global_optimizer).

Parameters

- **name** (string) – Module name ().
- **optimizer** (specification) – Optimizer configuration ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.MetaOptimizerWrapper(name, optimizer, multi_step=1, subsampling_fraction=1.0, clipping_threshold=None, optimizing_iterations=0, summary_labels=None, **kwargs)
```

Meta optimizer wrapper (specification key: meta_optimizer_wrapper).

Parameters

- **name** (string) – Module name ().
- **optimizer** (specification) – Optimizer configuration ().
- **multi_step** (parameter, int > 0) – Number of optimization steps (: single step).
- **subsampling_fraction** (parameter, 0.0 < float <= 1.0) – Fraction of batch timesteps to subsample (: no subsampling).
- **clipping_threshold** (parameter, float > 0.0) – Clipping threshold (: no clipping).
- **optimizing_iterations** (parameter, int >= 0) – Maximum number of line search iterations (: no optimizing).
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.MultiStep(name, optimizer, num_steps, unroll_loop=False, summary_labels=None)
```

Multi-step meta optimizer, which applies the given optimizer for a number of times (specification key: multi_step).

Parameters

- **name** (string) – Module name ().
- **optimizer** (specification) – Optimizer configuration ().
- **num_steps** (parameter, int > 0) – Number of optimization steps ().
- **unroll_loop** (bool) – Whether to unroll the repetition loop (: false).
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.NaturalGradient (name, learning_rate,
                                                    cg_max_iterations=10,
                                                    cg_damping=0.001,
                                                    cg_unroll_loop=False, summary_labels=None)
```

Natural gradient optimizer (specification key: `natural_gradient`).

Parameters

- **name** (*string*) – Module name ().
- **learning_rate** (*parameter, float > 0.0*) – Learning rate as KL-divergence of distributions between optimization steps ().
- **cg_max_iterations** (*int > 0*) – Maximum number of conjugate gradient iterations. (: 10).
- **cg_damping** (*float > 0.0*) – Conjugate gradient damping factor. (: 1e-3).
- **cg_unroll_loop** (*bool*) – Whether to unroll the conjugate gradient loop (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.OptimizingStep (name, optimizer,
                                                    ls_max_iterations=10,
                                                    ls_accept_ratio=0.9,
                                                    ls_mode='exponential',
                                                    ls_parameter=0.5,
                                                    ls_unroll_loop=False, summary_labels=None)
```

Optimizing-step meta optimizer, which applies line search to the given optimizer to find a more optimal step size (specification key: `optimizing_step`).

Parameters

- **name** (*string*) – Module name ().
- **optimizer** (*specification*) – Optimizer configuration ().
- **ls_max_iterations** (*parameter, int > 0*) – Maximum number of line search iterations (: 10).
- **ls_accept_ratio** (*parameter, float > 0.0*) – Line search acceptance ratio (: 0.9).
- **ls_mode** (*'exponential' | 'linear'*) – Line search mode, see line search solver (: 'exponential').
- **ls_parameter** (*parameter, float > 0.0*) – Line search parameter, see line search solver (: 0.5).
- **ls_unroll_loop** (*bool*) – Whether to unroll the line search loop (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.Plus (name, optimizer1, optimizer2, summary_labels=None)
```

Additive combination of two optimizers (specification key: `plus`).

Parameters

- **name** (*string*) – Module name ().

- **optimizer1** (*specification*) – First optimizer configuration ().
- **optimizer2** (*specification*) – Second optimizer configuration ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.SubsamplingStep(name, optimizer, fraction, summary_labels=None)
```

Subsampling-step meta optimizer, which randomly samples a subset of batch instances before applying the given optimizer (specification key: `subsampling_step`).

Parameters

- **name** (*string*) – Module name ().
- **optimizer** (*specification*) – Optimizer configuration ().
- **fraction** (*parameter*, $0.0 < \text{float} \leq 1.0$) – Fraction of batch timesteps to subsample ().
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.Synchronization(name, sync_frequency=1, update_weight=1.0, summary_labels=None)
```

Synchronization optimizer, which updates variables periodically to the value of a corresponding set of source variables (specification key: `synchronization`).

Parameters

- **name** (*string*) – Module name ().
- **optimizer** (*specification*) – Optimizer configuration ().
- **sync_frequency** (*parameter*, $\text{int} > 0$) – Timestep interval between updates which also perform a synchronization step (: every time).
- **update_weight** (*parameter*, $0.0 < \text{float} \leq 1.0$) – Update weight (: 1.0).
- **summary_labels** ('all' | iter[string]) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.optimizers.TFOptimizer(name, optimizer, learning_rate=0.0003, gradient_norm_clipping=1.0, summary_labels=None, **kwargs)
```

TensorFlow optimizer (specification key: `tf_optimizer`, `adadelata`, `adagrad`, `adam`, `gradient_descent`, `momentum`, `proximal_adagrad`, `proximal_gradient_descent`, `rmsprop`).

Parameters

- **name** (*string*) – Module name ().
- **optimizer** ('adadelata' | 'adagrad' | 'adam' | 'gradient_descent' | 'momentum' | 'proximal_adagrad' | 'proximal_gradient_descent' | 'rmsprop') – TensorFlow optimizer name, see [TensorFlow docs](#) ().
- **learning_rate** (*parameter*, $\text{float} > 0.0$) – Learning rate (: 3e-4).
- **gradient_norm_clipping** (*parameter*, $\text{float} > 0.0$) – Clip gradients by the ratio of the sum of their norms (: 1.0).

- **summary_labels** (*'all' / iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **kwargs** – Arguments for the TensorFlow optimizer, see [TensorFlow docs](#).

Default parameter: Constant

class `tensorforce.core.parameters.Constant` (*name*, *value*, *dtype*, *summary_labels=None*)
Constant hyperparameter.

Parameters

- **name** (*string*) – Module name ().
- **value** (*dtype-dependent*) – Constant hyperparameter value ().
- **dtype** ("bool" | "int" | "long" | "float") – Tensor type ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

class `tensorforce.core.parameters.Decaying` (*name*, *dtype*, *unit*, *decay*, *initial_value*, *decay_steps*, *increasing=False*, *inverse=False*, *scale=1.0*, *summary_labels=None*, ***kwargs*)

Decaying hyperparameter.

Parameters

- **name** (*string*) – Module name ().
- **dtype** ("bool" | "int" | "long" | "float") – Tensor type ().
- **unit** ("timesteps" | "episodes" | "updates") – Unit of decay schedule ().
- **decay** ("cosine" | "cosine_restarts" | "exponential" | "inverse_time" | "linear_cosine" | "linear_cosine_noisy" | "natural_exponential" | "polynomial") – Decay type, see [TensorFlow docs](#) ().
- **initial_value** (*float*) – Initial value ().
- **decay_steps** (*long*) – Number of decay steps ().
- **increasing** (*bool*) – Whether to subtract the decayed value from 1.0 (: false).

- **inverse** (*bool*) – Whether to take the inverse of the decayed value (: false).
- **scale** (*float*) – Scaling factor for (inverse) decayed value (: 1.0).
- **summary_labels** ("all" | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **kwargs** – Additional arguments depend on decay mechanism. Cosine decay:
Cosine decay with restarts:
Exponential decay:
Inverse time decay:
Linear cosine decay:
Natural exponential decay:
Noisy linear cosine decay:
Polynomial decay:

```
class tensorforce.core.parameters.OrnsteinUhlenbeck(name, dtype, theta=0.15,  
                                                    sigma=0.3, mu=0.0, summary_labels=None)
```

Ornstein-Uhlenbeck process.

Parameters

- **name** (*string*) – Module name ().
- **dtype** ("bool" | "int" | "long" | "float") – Tensor type ().
- **theta** (*float* > 0.0) – Theta value (: 0.15).
- **sigma** (*float* > 0.0) – Sigma value (: 0.3).
- **mu** (*float*) – Mu value (: 0.0).
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.parameters.PiecewiseConstant(name, dtype, unit, boundaries,  
                                                    values, summary_labels=None)
```

Piecewise-constant hyperparameter.

Parameters

- **name** (*string*) – Module name ().
- **dtype** ("bool" | "int" | "long" | "float") – Tensor type ().
- **unit** ("timesteps" | "episodes" | "updates") – Unit of interval boundaries ().
- **boundaries** (*iter[long]*) – Strictly increasing interval boundaries for constant segments ().
- **values** (*iter[dtype-dependent]*) – Interval values of constant segments, one more than ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.parameters.Random(name, dtype, distribution, shape=(), summary_labels=None, **kwargs)
```

Random hyperparameter.

Parameters

- **name** (*string*) – Module name ().
- **dtype** ("bool" | "int" | "long" | "float") – Tensor type ().
- **distribution** ("normal" | "uniform") – Distribution type for random hyperparameter value ().
- **shape** (*iter[int > 0]*) – Tensor shape (: scalar).
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **kwargs** – Additional arguments dependent on distribution type. Normal distribution:
Uniform distribution:


```
class tensorforce.core.layers.Activation(name, nonlinearity, input_spec=None, summary_labels=None)
```

Activation layer (specification key: activation).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- (**'crelu'** | **'elu'** | **'leaky-relu'** | **'none'** | **'relu'** | **'selu'** | **'sigmoid'** | (*nonlinearity*) – 'softmax' | 'softplus' | 'softsign' | 'swish' | 'tanh'): Nonlinearity ().
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (**'all'** | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Clipping(name, upper, lower=None, input_spec=None, summary_labels=None)
```

Clipping layer (specification key: clipping).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **upper** (*parameter, float*) – Upper clipping value ().
- **lower** (*parameter, float*) – Lower clipping value (: negative upper value).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (**'all'** | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Deltafier(name, concatenate=False, input_spec=None, summary_labels=None)
```

Deltafier layer computing the difference between the current and the previous input; can only be used as preprocessing layer (specification key: deltafier).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **concatenate** (*False | int >= 0*) – Whether to concatenate instead of replace deltas with input, and if so, concatenation axis (: false).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Dropout(name, rate, input_spec=None, summary_labels=None)
```

Dropout layer (specification key: dropout).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **rate** (*parameter, 0.0 <= float < 1.0*) – Dropout rate ().
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.ExponentialNormalization(name, decay=0.999, axes=None, input_spec=None, summary_labels=None)
```

Normalization layer based on the exponential moving average (specification key: exponential_normalization).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **decay** (*parameter, 0.0 <= float <= 1.0*) – Decay rate (: 0.999).
- **axes** (*iter[int >= 0]*) – Normalization axes, excluding batch axis (: all but last axis).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float >= 0.0*) – Scalar controlling L2 regularization (: inherit value of parent module).

```
class tensorforce.core.layers.Image(name, height=None, width=None, grayscale=False, input_spec=None, summary_labels=None)
```

Image preprocessing layer (specification key: image).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **height** (*int*) – Height of resized image (: no resizing or relative to width).
- **width** (*int*) – Width of resized image (: no resizing or relative to height).
- **grayscale** (*bool | iter[float]*) – Turn into grayscale image, optionally using given weights (: false).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.InstanceNormalization (name, axes=None, input_spec=None, summary_labels=None)
```

Instance normalization layer (specification key: `instance_normalization`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **axes** (*iter[int >= 0]*) – Normalization axes, excluding batch axis (: all).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.layers.Sequence (name, length, axis=-1, concatenate=True, input_spec=None, summary_labels=None)
```

Sequence layer stacking the current and previous inputs; can only be used as preprocessing layer (specification key: `sequence`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **length** (*int > 0*) – Number of inputs to concatenate ().
- **axis** (*int >= 0*) – Concatenation axis, excluding batch axis (: last axis).
- **concatenate** (*bool*) – Whether to concatenate inputs at given axis, otherwise introduce new sequence axis (: true).
- **input_spec** (*specification*) – Input tensor specification ().
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

Default policy: `ParametrizedDistributions`

```
class tensorflow.core.policies.ParametrizedDistributions (name,      states_spec,
                                                         actions_spec,      net-
                                                         work='auto',      dis-
                                                         tributions=None,
                                                         device=None,      sum-
                                                         mary_labels=None,
                                                         l2_regularization=None)
```

Policy which parametrizes independent distributions per action conditioned on the output of a central states-processing neural network (supports both stochastic and action-value-based policy interface) (specification key: `parametrized_distributions`).

Parameters

- **name** (*string*) – Module name ().
- **states_spec** (*specification*) – States specification ().
- **actions_spec** (*specification*) – Actions specification ().
- **network** ('auto' | *specification*) – Policy network configuration, see [networks](#) (: 'auto', automatically configured network).
- **distributions** (*dict[specification]*) – Distributions configuration, see [distributions](#), specified per action-type or -name (: per action-type, Bernoulli distribution for binary boolean actions, categorical distribution for discrete integer actions, Gaussian distribution for unbounded continuous actions, Beta distribution for bounded continuous actions).
- **device** (*string*) – Device name (: inherit value of parent module).
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).
- **l2_regularization** (*float* >= 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).

Environment interface

class `tensorflow.environments.Environment`

Tensorforce environment interface.

actions ()

Returns the action space specification.

Returns Arbitrarily nested dictionary of action descriptions with the following attributes:

Return type specification

close ()

Closes the environment.

static create (*environment*, ***kwargs*)

Creates an environment from a specification.

Parameters

- **environment** (*specification*) – JSON file, specification key, configuration dictionary, library module, or `Environment` subclass ().
- **kwargs** – Additional arguments.

execute (*actions*)

Executes the given action(s) and advances the environment by one step.

Parameters **actions** (*dict[action]*) – Dictionary containing action(s) to be executed ().

Returns Dictionary containing next state(s), whether a terminal state is reached or 2 if the episode was aborted, and observed reward.

Return type ((dict[state], bool | 0 | 1 | 2, float))

max_episode_timesteps ()

Returns the maximum number of timesteps per episode.

Returns Maximum number of timesteps per episode.

Return type int

reset ()

Resets the environment to start a new episode.

Returns Dictionary containing initial state(s) and auxiliary information.

Return type dict[state]

states ()

Returns the state space specification.

Returns Arbitrarily nested dictionary of state descriptions with the following attributes:

Return type specification

Arcade Learning Environment

```
class tensorflow.environments.ArcadeLearningEnvironment (level,
                                                    life_loss_terminal=False,
                                                    life_loss_punishment=0.0,
                                                    re-
                                                    peat_action_probability=0.0,
                                                    visualize=False,
                                                    frame_skip=1,
                                                    seed=None)
```

Arcade Learning Environment adapter (specification key: ale, arcade_learning_environment).

May require:

```
sudo apt-get install libsdl1.2-dev libsdl-gfx1.2-dev libsdl-image1.2-dev cmake

mkdir build && cd build
cmake -DUSE_SDL=ON -DUSE_RLGLUE=OFF -DBUILD_EXAMPLES=ON ..
make -j 4

pip install git+https://github.com/mgbellemare/Arcade-Learning-Environment.git
```

Parameters

- **level** (*string*) – ALE rom file ().
- **loss_of_life_termination** – Signals a terminal state on loss of life (: false).
- **loss_of_life_reward** (*float*) – Reward/Penalty on loss of life (negative values are a penalty) (: 0.0).
- **repeat_action_probability** (*float*) – Repeats last action with given probability (: 0.0).
- **visualize** (*bool*) – Whether to visualize interaction (: false).
- **frame_skip** (*int* > 0) – Number of times to repeat an action without observing (: 1).
- **seed** (*int*) – Random seed (: none).

Maze Explorer

class `tensorforce.environments.MazeExplorer` (*level*, *visualize=False*)
`MazeExplorer` environment adapter (specification key: `mazeexp`, `maze_explorer`).

May require:

```
sudo apt-get install freeglut3-dev  
pip install mazeexp
```

Parameters

- **level** (*int*) – Game mode, see [GitHub](#) ().
- **visualize** (*bool*) – Whether to visualize interaction (: `false`).


```
class tensorforce.environments.OpenSim(level, visualize=False, integrator_accuracy=5e-05)  
    OpenSim environment adapter (specification key: osim, open_sim).
```

Parameters

- **level** (*'Arm2D' | 'L2Run' | 'Prosthetics'*) – Environment id ().
- **visualize** (*bool*) – Whether to visualize interaction (: false).
- **integrator_accuracy** (*float*) – Integrator accuracy (: 5e-5).


```
class tensorforce.environments.OpenAIGym(level,  
                                         visualize=False,  
                                         max_episode_timesteps=None,  
                                         terminal_reward=0.0,  
                                         reward_threshold=None,  
                                         tags=None,  
                                         monitor_directory=None,  
                                         **kwargs)
```

OpenAI Gym environment adapter (specification key: gym, openai_gym).

May require:

```
pip install gym[all]
```

Parameters

- **level** (*string*) – Gym id ().
- **visualize** (*bool*) – Whether to visualize interaction (: false).
- **max_episode_timesteps** (*false | int > 0*) – Whether to terminate an episode after a while, and if so, maximum number of timesteps per episode (: Gym default).
- **terminal_reward** (*float*) – Additional reward for early termination, if otherwise indistinguishable from termination due to maximum number of timesteps (: Gym default).
- **reward_threshold** (*float*) – Gym environment argument, the reward threshold before the task is considered solved (: Gym default).
- **tags** (*dict*) – Gym environment argument, a set of arbitrary key-value tags on this environment, including simple property=True tags (: Gym default).
- **monitor_directory** (*string*) – Monitor output directory (: none).
- **kwargs** – Additional Gym environment arguments.


```
class tensorforce.environments.OpenAIRetro (level, visualize=False, monitor_directory=None, **kwargs)
```

OpenAI Retro environment adapter (specification key: `retro`, `openai_retro`).

May require:

```
pip install gym-retro
```

Parameters

- **level** (*string*) – Game id ().
- **visualize** (*bool*) – Whether to visualize interaction (: `false`).
- **monitor_directory** (*string*) – Monitor output directory (: `none`).
- **kwargs** – Additional Retro environment arguments.

PyGame Learning Environment

```
class tensorflow.environments.PyGameLearningEnvironment (level, visualize=False,
                                                         frame_skip=1, fps=30)
    PyGame Learning Environment environment adapter (specification key: ple,
    pygame_learning_environment).
```

May require:

```
sudo apt-get install git python3-dev python3-setuptools python3-numpy python3-
↳opengl \
libsdl-image1.2-dev libsdl-mixer1.2-dev libsdl-ttf2.0-dev libsmpeg-dev libsdl1.2-
↳dev \
libportmidi-dev libswscale-dev libavformat-dev libavcodec-dev libtiff5-dev libx11-
↳6 \
libx11-dev fluid-soundfont-gm timgm6mb-soundfont xfonts-base xfonts-100dpi xfonts-
↳75dpi \
xfonts-cyrillic fontconfig fonts-freefont-ttf libfreetype6-dev

pip install git+https://github.com/pygame/pygame.git

pip install git+https://github.com/ntasfi/PyGame-Learning-Environment.git
```

Parameters

- **level** (string | subclass of `ple.games.base`) – Game instance or name of class in `ple.games`, like ‘doom’, ‘flappybird’, ‘monsterkong’, ‘catcher’, ‘pixelcopter’, ‘pong’, ‘puckworld’, ‘raycastmaze’, ‘snake’, ‘waterworld’ ().
- **visualize** (*bool*) – Whether to visualize interaction (: false).
- **frame_skip** (*int* > 0) – Number of times to repeat an action without observing (: 1).
- **fps** (*int* > 0) – The desired frames per second we want to run our game at (: 30).


```
class tensorforce.environments.ViZDoom(level, visualize=False, include_variables=False,  
                                       factored_action=False, frame_skip=12,  
                                       seed=None)
```

ViZDoom environment adapter (specification key: `vizdoom`).

Parameters

- **level** (*string*) – ViZDoom configuration file ().
- **include_variables** (*bool*) – Whether to include game variables to state (: `false`).
- **factored_action** (*bool*) – Whether to use factored action representation (: `false`).
- **visualize** (*bool*) – Whether to visualize interaction (: `false`).
- **frame_skip** (*int* > 0) – Number of times to repeat an action without observing (: 12).
- **seed** (*int*) – Random seed (: `none`).

A

`act()` (*tensorforce.agents.Agent* method), 13
`actions()` (*tensorforce.environments.Environment* method), 69
`ActionValue` (*class in tensorforce.core.objectives*), 51
`Activation` (*class in tensorforce.core.layers*), 41
`Agent` (*class in tensorforce.agents*), 13
`ArcadeLearningEnvironment` (*class in tensorforce.environments*), 71
`AutoNetwork` (*class in tensorforce.core.networks*), 49

B

`Bernoulli` (*class in tensorforce.core.distributions*), 33
`Beta` (*class in tensorforce.core.distributions*), 33

C

`Categorical` (*class in tensorforce.core.distributions*), 33
`Clipping` (*class in tensorforce.core.layers*), 42
`ClippingStep` (*class in tensorforce.core.optimizers*), 53
`close()` (*tensorforce.agents.Agent* method), 13
`close()` (*tensorforce.environments.Environment* method), 69
`Constant` (*class in tensorforce.core.parameters*), 59
`ConstantAgent` (*class in tensorforce.agents*), 17
`Conv1d` (*class in tensorforce.core.layers*), 35
`Conv2d` (*class in tensorforce.core.layers*), 36
`create()` (*tensorforce.agents.Agent* static method), 13
`create()` (*tensorforce.environments.Environment* static method), 69

D

`Decaying` (*class in tensorforce.core.parameters*), 59
`DeepQNetwork` (*class in tensorforce.agents*), 25
`Deltafier` (*class in tensorforce.core.layers*), 42
`Dense` (*class in tensorforce.core.layers*), 36
`Dropout` (*class in tensorforce.core.layers*), 42

E

`Embedding` (*class in tensorforce.core.layers*), 37
`Environment` (*class in tensorforce.environments*), 69
`Evolutionary` (*class in tensorforce.core.optimizers*), 53
`execute()` (*tensorforce.environments.Environment* method), 69
`ExponentialNormalization` (*class in tensorforce.core.layers*), 41

F

`Flatten` (*class in tensorforce.core.layers*), 40
`Function` (*class in tensorforce.core.layers*), 45

G

`Gaussian` (*class in tensorforce.core.distributions*), 34
`get_available_summaries()` (*tensorforce.agents.Agent* method), 14
`get_output_tensors()` (*tensorforce.agents.Agent* method), 14
`get_query_tensors()` (*tensorforce.agents.Agent* method), 14
`GlobalOptimizer` (*class in tensorforce.core.optimizers*), 54
`Gru` (*class in tensorforce.core.layers*), 38

I

`Image` (*class in tensorforce.core.layers*), 42
`initialize()` (*tensorforce.agents.Agent* method), 14
`InstanceNormalization` (*class in tensorforce.core.layers*), 41
`InternalGru` (*class in tensorforce.core.layers*), 43
`InternalLstm` (*class in tensorforce.core.layers*), 44
`InternalRnn` (*class in tensorforce.core.layers*), 44

K

`Keras` (*class in tensorforce.core.layers*), 45

L

LayeredNetwork (class in *tensorforce.core.networks*), 49

Linear (class in *tensorforce.core.layers*), 37

Lstm (class in *tensorforce.core.layers*), 38

M

max_episode_timesteps() (*tensorforce.environments.Environment* method), 69

MazeExplorer (class in *tensorforce.environments*), 73

MetaOptimizerWrapper (class in *tensorforce.core.optimizers*), 54

MultiStep (class in *tensorforce.core.optimizers*), 54

N

NaturalGradient (class in *tensorforce.core.optimizers*), 54

O

observe() (*tensorforce.agents.Agent* method), 14

OpenAIGym (class in *tensorforce.environments*), 77

OpenAIRetro (class in *tensorforce.environments*), 79

OpenSim (class in *tensorforce.environments*), 75

OptimizingStep (class in *tensorforce.core.optimizers*), 55

OrnsteinUhlenbeck (class in *tensorforce.core.parameters*), 60

P

ParametrizedDistributions (class in *tensorforce.core.policies*), 67

PiecewiseConstant (class in *tensorforce.core.parameters*), 60

Plus (class in *tensorforce.core.objectives*), 51

Plus (class in *tensorforce.core.optimizers*), 55

PolicyAgent (class in *tensorforce.agents*), 21

PolicyGradient (class in *tensorforce.core.objectives*), 51

Pool1d (class in *tensorforce.core.layers*), 40

Pool2d (class in *tensorforce.core.layers*), 40

Pooling (class in *tensorforce.core.layers*), 40

ProximalPolicyOptimization (class in *tensorforce.agents*), 29

PyGameLearningEnvironment (class in *tensorforce.environments*), 81

R

Random (class in *tensorforce.core.parameters*), 60

RandomAgent (class in *tensorforce.agents*), 19

Recent (class in *tensorforce.core.memories*), 47

Register (class in *tensorforce.core.layers*), 45

Replay (class in *tensorforce.core.memories*), 47

reset() (*tensorforce.agents.Agent* method), 14

reset() (*tensorforce.environments.Environment* method), 69

restore() (*tensorforce.agents.Agent* method), 14

Retrieve (class in *tensorforce.core.layers*), 45

Reuse (class in *tensorforce.core.layers*), 46

Rnn (class in *tensorforce.core.layers*), 39

S

save() (*tensorforce.agents.Agent* method), 14

Sequence (class in *tensorforce.core.layers*), 43

states() (*tensorforce.environments.Environment* method), 70

StateValue (class in *tensorforce.core.objectives*), 52

SubsamplingStep (class in *tensorforce.core.optimizers*), 56

Synchronization (class in *tensorforce.core.optimizers*), 56

T

TFOptimizer (class in *tensorforce.core.optimizers*), 56

TrustRegionPolicyOptimization (class in *tensorforce.agents*), 31

V

VanillaPolicyGradient (class in *tensorforce.agents*), 27

VizDoom (class in *tensorforce.environments*), 83