
Tensorforce Documentation

Release 0.5.5

Tensorforce Team

Jun 16, 2020

1	Installation	3
2	Getting started	5
2.1	Initializing an environment	5
2.2	Initializing an agent	6
2.3	Training and evaluation	7
3	Agent specification	9
3.1	States and actions specification	9
3.2	How to specify modules	9
3.3	Static vs dynamic hyperparameters	10
4	Features	13
4.1	Multi-input and non-sequential network architectures	13
4.2	Action masking	14
4.3	Parallel environment execution	14
4.4	Record & pretrain	15
4.5	Save & restore	15
4.6	TensorBoard	16
5	run.py – Runner	17
5.1	Agent arguments	17
5.2	Environment arguments	17
5.3	Parallel execution arguments	17
5.4	Runner arguments	18
5.5	Logging arguments	18
6	tune.py – Hyperparameter tuner	19
6.1	Required arguments	19
6.2	Optional arguments	19
7	Agent interface	21
7.1	Initialization and termination	21
7.2	Main reinforcement learning interface	21
7.3	Required for evaluation at episode start	22
7.4	Loading and saving	22
7.5	Get and assign variables	23

7.6	Custom summaries	24
7.7	Advanced functions for specialized use cases	24
7.8	Others	25
8	Constant Agent	27
9	Random Agent	29
10	Tensorforce Agent	31
11	Deep Q-Network	35
12	Dueling DQN	39
13	Vanilla Policy Gradient	43
14	Actor-Critic	47
15	Advantage Actor-Critic	51
16	Deterministic Policy Gradient	55
17	Proximal Policy Optimization	59
18	Trust-Region Policy Optimization	63
19	Distributions	67
20	Layers	69
20.1	Convolutional layers	69
20.2	Dense layers	70
20.3	Embedding layers	71
20.4	Recurrent layers	72
20.5	Pooling layers	73
20.6	Normalization layers	75
20.7	Misc layers	75
20.8	Layers with internal states	77
20.9	Special layers	79
21	Memories	81
22	Networks	83
23	Objectives	85
24	Optimizers	87
25	Parameters	93
26	Preprocessing	97
27	Policies	101
28	Runner	103
29	Environment interface	107
29.1	Initialization and termination	107
29.2	Attributes	108

29.3 Interaction functions	108
30 Arcade Learning Environment	109
31 Maze Explorer	111
32 Open Sim	113
33 OpenAI Gym	115
34 OpenAI Retro	117
35 PyGame Learning Environment	119
36 ViZDoom	121
Index	123

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](#) version 2.0 (!) 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:

```
pip3 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
pip3 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
pip3 install tensorforce[tf]

# GitHub version plus TensorFlow GPU version
pip3 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 Initializing an environment

It is recommended to initialize an environment via the `Environment.create(...)` interface.

```
from tensorforce.environments import Environment
```

For instance, the OpenAI CartPole environment can be initialized as follows:

```
environment = Environment.create(  
    environment='gym', level='CartPole', max_episode_timesteps=500  
)
```

Gym's pre-defined versions are also accessible:

```
environment = Environment.create(environment='gym', level='CartPole-v1')
```

Alternatively, an environment can be specified as a config file:

```
{  
    "environment": "gym",  
    "level": "CartPole"  
}
```

Environment config files can be loaded by passing their file path:

```
environment = Environment.create(  
    environment='environment.json', max_episode_timesteps=500  
)
```

Custom Gym environments can be used in the same way, but require the corresponding class(es) to be imported and registered accordingly.

Finally, it is possible to implement a custom environment using Tensorforce's `Environment` interface:

```
class CustomEnvironment(Environment):

    def __init__(self):
        super().__init__()

    def states(self):
        return dict(type='float', shape=(8,))

    def actions(self):
        return dict(type='int', num_values=4)

    # Optional, should only be defined if environment has a natural maximum
    # episode length
    def max_episode_timesteps(self):
        return super().max_episode_timesteps()

    # Optional
    def close(self):
        super().close()

    def reset(self):
        state = np.random.random(size=(8,))
        return state

    def execute(self, actions):
        next_state = np.random.random(size=(8,))
        terminal = np.random.random() < 0.5
        reward = np.random.random()
        return next_state, terminal, reward
```

Custom environment implementations can be loaded by passing their module path:

```
environment = Environment.create(
    environment='custom_env.CustomEnvironment', max_episode_timesteps=10
)
```

It is strongly recommended to specify the `max_episode_timesteps` argument of `Environment.create(...)` unless specified by the environment (or for evaluation), as otherwise more agent parameters may require specification.

2.2 Initializing an agent

Similarly to environments, it is recommended to initialize an agent via the `Agent.create(...)` interface.

```
from tensorforce.agents import Agent
```

For instance, the generic Tensorforce agent can be initialized as follows:

```
agent = Agent.create(
    agent='tensorforce', environment=environment, update=64,
    objective='policy_gradient', reward_estimation=dict(horizon=20)
)
```

Other pre-defined agent classes can alternatively be used, for instance, [Proximal Policy Optimization](#):

```
agent = Agent.create(
    agent='ppo', environment=environment, batch_size=10, learning_rate=1e-3
)
```

Alternatively, an agent can be specified as a config file:

```
{
    "agent": "tensorforce",
    "update": 64,
    "objective": "policy_gradient",
    "reward_estimation": {
        "horizon": 20
    }
}
```

Agent config files can be loaded by passing their file path:

```
agent = Agent.create(agent='agent.json', environment=environment)
```

It is recommended to pass the environment object returned by `Environment.create(...)` as `environment` argument of `Agent.create(...)`, so that the `states`, `actions` and `max_episode_timesteps` argument are automatically specified accordingly.

2.3 Training and evaluation

It is recommended to use the execution utilities for training and evaluation, like the [Runner utility](#), which offer a range of configuration options:

```
from tensorforce.execution import Runner
```

A basic experiment consisting of training and subsequent evaluation can be written in a few lines of code:

```
runner = Runner(
    agent='agent.json',
    environment=dict(environment='gym', level='CartPole'),
    max_episode_timesteps=500
)

runner.run(num_episodes=200)

runner.run(num_episodes=100, evaluation=True)

runner.close()
```

The execution utility classes take care of handling the agent-environment interaction correctly, and thus should be used where possible. Alternatively, if more detailed control over the agent-environment interaction is required, a simple training and evaluation loop can be written as follows:

```
# Create agent and environment
environment = Environment.create(
    environment='environment.json', max_episode_timesteps=500
)
agent = Agent.create(agent='agent.json', environment=environment)
```

(continues on next page)

(continued from previous page)

```
# Train for 200 episodes
for _ in range(200):
    states = environment.reset()
    terminal = False
    while not terminal:
        actions = agent.act(states=states)
        states, terminal, reward = environment.execute(actions=actions)
        agent.observe(terminal=terminal, reward=reward)

# Evaluate for 100 episodes
sum_rewards = 0.0
for _ in range(100):
    states = environment.reset()
    internals = agent.initial_internals()
    terminal = False
    while not terminal:
        actions, internals = agent.act(states=states, internals=internals,
↪evaluation=True)
        states, terminal, reward = environment.execute(actions=actions)
        sum_rewards += reward

print('Mean episode reward:', sum_rewards / 100)

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

Agent 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 States and actions specification

A state/action value is specified as dictionary with mandatory attributes `type` (one of `'bool'`: binary, `'int'`: discrete, or `'float'`: continuous) and `shape` (a positive number or tuple thereof). Moreover, `'int'` values should additionally specify `num_values` (the fixed number of discrete options), whereas `'float'` values can specify bounds via `min/max_value`. If the state or action consists of multiple components, these are specified via an additional dictionary layer. The following example illustrates both possibilities:

```
states = dict(
    observation=dict(type='float', shape=(16, 16, 3)),
    attributes=dict(type='int', shape=(4, 2), num_values=5)
)
actions = dict(type='float', shape=10)
```

Note: Ideally, the agent arguments `states` and `actions` are specified implicitly by passing the `environment` argument.

3.2 How to specify modules

3.2.1 Dictionary with module type and arguments

```
Agent.create(...
    policy=dict(network=dict(type='layered', layers=[dict(type='dense', size=32)])),
```

(continues on next page)

(continued from previous page)

```
memory=dict(type='replay', capacity=10000), ...  
)
```

3.2.2 JSON specification file (plus additional arguments)

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

3.2.3 Module path (plus additional arguments)

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

3.2.4 Callable or Type (plus additional arguments)

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

3.2.5 Default module: only arguments or first argument

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

3.3 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.

3.3.1 Example: exponentially decaying exploration


```
Agent.create(...
    exploration=dict(
        type='decaying', unit='timesteps', decay='exponential',
        initial_value=0.1, decay_steps=1000, decay_rate=0.5
    ), ...
)
```

3.3.2 Example: linearly increasing horizon

```
Agent.create(...
    reward_estimation=dict(horizon=dict(
        type='decaying', dtype='long', unit='episodes', decay='polynomial',
        initial_value=10.0, decay_steps=1000, final_value=50.0, power=1.0
    ), ...
)
```


4.1 Multi-input and non-sequential network architectures

Multi-input and other non-sequential networks are specified as list of layer lists, as opposed to simply a list of layers for sequential networks. The following example illustrates how to specify such a more complex network, by using the special layers `Register` and `Retrieve` to combine the multiple sequential layer stacks.

```
Agent.create(  
    states=dict(  
        observation=dict(type='float', shape=(16, 16, 3)),  
        attributes=dict(type='int', shape=(4, 2), num_values=5)  
    ),  
    ...  
    policy=[  
        [  
            dict(type='retrieve', tensors='observation'),  
            dict(type='conv2d', size=16),  
            dict(type='flatten'),  
            dict(type='register', tensor='obs-embedding')  
        ],  
        [  
            dict(type='retrieve', tensors='attributes'),  
            dict(type='embedding', size=16),  
            dict(type='flatten'),  
            dict(type='register', tensor='attr-embedding')  
        ],  
        [  
            dict(  
                type='retrieve', tensors=['obs-embedding', 'attr-embedding'],  
                aggregation='concat'  
            ),  
            dict(type='dense', size=32)  
        ]  
    ],  
    ...  
)
```

(continues on next page)

(continued from previous page)

```

    ...
)

```

4.2 Action masking

```

agent = Agent.create(
    states=dict(type='float', shape=(10,)),
    actions=dict(type='int', shape=(), num_actions=3), ...
)
...
states = dict(
    state=np.random.random_sample(size=(10,)), # regular state
    action_mask=[True, False, True] # mask as '[ACTION-NAME]_mask'
)
action = agent.act(states=states)
assert action != 1

```

4.3 Parallel environment execution

Execute multiple environments running locally in one call / batched:

```

Runner(
    agent='benchmarks/configs/ppol.json', environment='CartPole-v1',
    num_parallel=5
)
runner.run(num_episodes=100, batch_agent_calls=True)

```

Execute environments running in different processes whenever ready / unbatched:

```

Runner(
    agent='benchmarks/configs/ppol.json', environment='CartPole-v1',
    num_parallel=5, remote='multiprocessing'
)
runner.run(num_episodes=100)

```

Execute environments running on different machines, here using `run.py` instead of `Runner`:

```

# Environment machine 1
python run.py --environment gym --level CartPole-v1 --remote socket-server \
    --port 65432

# Environment machine 2
python run.py --environment gym --level CartPole-v1 --remote socket-server \
    --port 65433

# Agent machine
python run.py --agent benchmarks/configs/ppol.json --episodes 100 \
    --num-parallel 2 --remote socket-client --host 127.0.0.1,127.0.0.1 \
    --port 65432,65433 --batch-agent-calls

```

4.4 Record & pretrain

```
agent = Agent.create(...
    recorder=dict(
        directory='data/traces',
        frequency=100 # record a traces file every 100 episodes
    ), ...
)
...
agent.close()

# Pretrain agent on recorded traces
agent = Agent.create(...)
agent.pretrain(
    directory='data/traces',
    num_iterations=100 # perform 100 update iterations on traces (more_
↳ configurations possible)
)
```

4.5 Save & restore

4.5.1 TensorFlow saver (full model)

```
agent = Agent.create(...
    saver=dict(
        directory='data/checkpoints',
        frequency=600 # save checkpoint every 600 seconds (10 minutes)
    ), ...
)
...
agent.close()

# Restore latest agent checkpoint
agent = Agent.load(directory='data/checkpoints')
```

4.5.2 NumPy / HDF5 (only weights)

```
agent = Agent.create(...
    saver=dict(
        directory='data/checkpoints',
        frequency=600 # save checkpoint every 600 seconds (10 minutes)
    ), ...
)
...
agent.save(directory='data/checkpoints', format='numpy', append='episodes')

# Restore latest agent checkpoint
agent = Agent.load(directory='data/checkpoints', format='numpy')
```

4.6 TensorBoard

```
Agent.create(...
    summarizer=dict(
        directory='data/summaries',
        # list of labels, or 'all'
        labels=['graph', 'entropy', 'kl-divergence', 'losses', 'rewards'],
        frequency=100, # store values every 100 timesteps
        # (infrequent update summaries every update; other configurations possible)
        custom=dict( # custom summaries which need to be recorded explicitly
            custom_summary1=dict(type='image', max_outputs=10), ...
        )
    ), ...
)
...
# custom summary recording
agent.summarize(summary='custom_summary1', value=image)
```

5.1 Agent arguments

- [a]gent** (*string*, **required** unless “socket-server” remote mode) – Agent (name, configuration JSON file, or library module)
- [n]etwork** (*string*, *default: not specified*) – Network (name, configuration JSON file, or library module)

5.2 Environment arguments

- [e]nvironment** (*string*, **required** unless “socket-client” remote mode) – Environment (name, configuration JSON file, or library module)
- [l]evel** (*string*, *default: not specified*) – Level or game id, like `CartPole-v1`, if supported
- [m]ax-episode-timesteps** (*int*, *default: not specified*) – Maximum number of timesteps per episode
- visualize** (*bool*, *default: false*) – Visualize agent–environment interaction, if supported
- visualize-directory** (*bool*, *default: not specified*) – Directory to store videos of agent–environment interaction, if supported
- import-modules** (*string*, *default: not specified*) – Import comma-separated modules required for environment

5.3 Parallel execution arguments

- num-parallel** (*int*, *default: no parallel execution*) – Number of environment instances to execute in parallel
- batch-agent-calls** (*bool*, *default: false*) – Batch agent calls for parallel environment execution
- sync-timesteps** (*bool*, *default: false*) – Synchronize parallel environment execution on timestep-level
- sync-episodes** (*bool*, *default: false*) – Synchronize parallel environment execution on episode-level

–remote (*str*, *default: local execution*) – Communication mode for remote environment execution of parallelized environment execution: “multiprocessing” | “socket-client” | “socket-server”. In case of “socket-server”, runs environment in server communication loop until closed.

–blocking (*bool*, *default: false*) – Remote environments should be blocking

–host (*str*, *only for “socket-client” remote mode*) – Socket server hostname(s) or IP address(es), single value or comma-separated list

–port (*str*, *only for “socket-client/server” remote mode*) – Socket server port(s), single value or comma-separated list, increasing sequence if single host and port given

5.4 Runner arguments

–e[v]aluation (*bool*, *default: false*) – Run environment (last if multiple) in evaluation mode

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

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

–[u]pdates (*int*, *default: not specified*) – Number of agent updates

–mean-horizon (*int*, *default: 1*) – Number of episodes progress bar values and evaluation score are averaged over

–save-best-agent (*bool*, *default: false*) – Directory to save the best version of the agent according to the evaluation score

5.5 Logging arguments

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

–path (*string*, *default: not specified*) – Logging path, directory plus filename without extension

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

6.1 Required arguments

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

6.2 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

7.1 Initialization and termination

static `TensorforceAgent.create` (*agent='tensorforce', environment=None, **kwargs*)
Creates an agent from a specification.

Parameters

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

`TensorforceAgent.close()`
Closes the agent.

7.2 Main reinforcement learning interface

`TensorforceAgent.act` (*states, internals=None, parallel=0, independent=False, deterministic=False, evaluation=False, query=None, **kwargs*)
Returns action(s) for the given state(s), needs to be followed by `observe(...)` unless independent mode set via `independent/evaluation`.

Parameters

- **states** (*dict[state] | iter[dict[state]]*) – Dictionary containing state(s) to be acted on ().
- **internals** (*dict[internal] | iter[dict[internal]]*) – Dictionary containing current internal agent state(s), either given by `initial_internals()` at the

beginning of an episode or as return value of the preceding `act (. . .)` call (if independent mode and agent has internal states).

- **parallel** (*int* | *iter[int]*) – Parallel execution index (: 0).
- **independent** (*bool*) – Whether act is not part of the main agent-environment interaction, and this call is thus not followed by observe (: false).
- **deterministic** (*bool*) – If independent mode, whether to act deterministically, so no exploration and sampling (: false).
- **evaluation** (*bool*) – Whether the agent is currently evaluated, implies independent and deterministic (: false).
- **query** (*list[str]*) – Names of tensors to retrieve (: none).
- **kwargs** – Additional input values, for instance, for dynamic hyperparameters.

Returns `dict[action]` | `iter[dict[action]]`, `dict[internal]` | `iter[dict[internal]]` if `internals` argument given, plus optional `list[str]`: Dictionary containing action(s), dictionary containing next internal agent state(s) if independent mode, plus queried tensor values if requested.

`TensorforceAgent.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* | *iter[float]*) – Reward ().
- **terminal** (*bool* | 0 | 1 | 2 | *iter[...]*) – Whether a terminal state is reached or 2 if the episode was aborted (: false).
- **parallel** (*int*, *iter[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` | `int`, optional `list[str]`)

7.3 Required for evaluation at episode start

`TensorforceAgent.initial_internals ()`

Returns the initial internal agent state(s), to be used at the beginning of an episode as `internals` argument for `act (. . .)` in independent mode

Returns Dictionary containing initial internal agent state(s).

Return type `dict[internal]`

7.4 Loading and saving

static `TensorforceAgent.load (directory=None, filename=None, format=None, environment=None, **kwargs)`

Restores an agent from a specification directory/file.

Parameters

- **directory** (*str*) – Checkpoint directory (: current directory “.”).

- **filename** (*str*) – Checkpoint filename, with or without append and extension (: “agent”).
- **format** (*"tensorflow" | "numpy" | "hdf5" | "pb-actonly"*) – File format, “pb-actonly” loads an act-only agent based on a Protobuf model (: format matching directory and filename, required to be unambiguous).
- **environment** (*Environment object*) – Environment which the agent is supposed to be trained on, environment-related arguments like state/action space specifications and maximum episode length will be extract if given (unless “pb-actonly” format).
- **kwargs** – Additional arguments, invalid for “pb-actonly” format.

`TensorforceAgent.save(directory=None, filename=None, format='tensorflow', append=None)`

Saves the agent to a checkpoint.

Parameters

- **directory** (*str*) – Checkpoint directory (: directory specified for TensorFlow saver, otherwise current directory).
- **filename** (*str*) – Checkpoint filename, without extension (: filename specified for TensorFlow saver, otherwise name of agent).
- **format** (*"tensorflow" | "numpy" | "hdf5"*) – File format, “tensorflow” uses TensorFlow saver to store variables, graph meta information and an optimized Protobuf model with an act-only graph, whereas the others only store variables as NumPy/HDF5 file (: TensorFlow format).
- **append** (*"timesteps" | "episodes" | "updates"*) – Append current timestep/episode/update to checkpoint filename (: none).

Returns Checkpoint path.

Return type `str`

7.5 Get and assign variables

`TensorforceAgent.get_variables()`

Returns the names of all agent variables.

Returns Names of variables.

Return type `list[str]`

`TensorforceAgent.get_variable(variable)`

Returns the value of the variable with the given name.

Parameters **variable** (*string*) – Variable name ().

Returns Variable value.

Return type `numpy-array`

`TensorforceAgent.assign_variable(variable, value)`

Assigns the given value to the variable with the given name.

Parameters

- **variable** (*string*) – Variable name ().
- **value** (*variable-compatible value*) – Value to assign to variable ().

7.6 Custom summaries

`TensorforceAgent.summarize(summary, value, step=None)`

Records a value for the given custom summary label (as specified via `summarizer[custom]`).

Parameters

- **variable** (*string*) – Custom summary label ().
- **value** (*summary-compatible value*) – Summary value to record ().
- **step** (*int*) – Summary recording step (: current timestep).

7.7 Advanced functions for specialized use cases

`TensorforceAgent.experience(states, actions, terminal, reward, internals=None, query=None, **kwargs)`

Feed experience traces.

Parameters

- **states** (*dict[array[state]]*) – Dictionary containing arrays of states ().
- **actions** (*dict[array[action]]*) – Dictionary containing arrays of actions ().
- **terminal** (*array[bool]*) – Array of terminals ().
- **reward** (*array[float]*) – Array of rewards ().
- **internals** (*dict[state]*) – Dictionary containing arrays of internal agent states (: no internal states).
- **query** (*list[str]*) – Names of tensors to retrieve (: none).
- **kwargs** – Additional input values, for instance, for dynamic hyperparameters.

`TensorforceAgent.update(query=None, **kwargs)`

Perform an update.

Parameters

- **query** (*list[str]*) – Names of tensors to retrieve (: none).
- **kwargs** – Additional input values, for instance, for dynamic hyperparameters.

`TensorforceAgent.pretrain(directory, num_iterations, num_traces=1, num_updates=1)`

Naive pretraining approach as a combination of `experience()` and `update`, uses experience traces obtained e.g. via recorder argument.

Parameters

- **directory** (*path*) – Directory with experience traces, e.g. obtained via recorder; episode length has to be consistent with agent configuration ().
- **num_iterations** (*int > 0*) – Number of iterations consisting of loading new traces and performing multiple updates ().
- **num_traces** (*int > 0*) – Number of traces to load per iteration; has to at least satisfy the update batch size (: 1).
- **num_updates** (*int > 0*) – Number of updates per iteration (: 1).

7.8 Others

`TensorforceAgent.reset()`

Resets all agent buffers and discards unfinished episodes.

`TensorforceAgent.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]`

`TensorforceAgent.get_available_summaries()`

Returns the summary labels provided by the agent.

Returns Available summary labels.

Return type `list[str]`

Constant Agent

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

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

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **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 (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: `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): , specification with either type `"scalar"`, type `"histogram"` with optional `"buckets"`, type `"image"` with optional `"max_outputs"` (default: 3), or type `"audio"` (default: no custom summaries).

`labels (<i>"all" | iter[string]</i>) – all or list of summaries to record, from the following labels (default: only "graph"); "graph": graph summary "parameters": parameter scalars`

- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Random Agent

```
class tensorflow.agents.RandomAgent (states, actions, max_episode_timesteps=None,  
                                     name='agent', device=None, seed=None, summar-  
                                     izer=None, recorder=None, config=None)
```

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

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: 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): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).labels (<i>“all” | iter[string]</i>) – all or list of summaries to record, from the following labels (default: only

“graph”): ”graph”: graph summary ”parameters”: parameter scalars
‘

- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Tensorforce Agent

```
class tensorforce.agents.TensorforceAgent (states, actions, update, ob-  
jective, reward_estimation,  
max_episode_timesteps=None, pol-  
icy='default', memory=None, opti-  
mizer='adam', baseline_policy=None,  
baseline_optimizer=None, base-  
line_objective=None, preprocess-  
ing=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, execu-  
tion=None, saver=None, summarizer=None,  
recorder=None, config=None)
```

Tensorforce agent (specification key: `tensorforce`).

Highly configurable agent and basis 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 (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).

- **policy** (*specification*) – Policy configuration, see [policies](#) (: “default”, action distributions parametrized by an automatically configured network).
- **memory** (*int* | *specification*) – Memory configuration, see [memories](#) (: replay memory with either given or minimum 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** (*specification*) – Baseline policy configuration, main policy will be used as baseline if none (: none).
- **baseline_optimizer** (*float* > 0.0 | *specification*) – Baseline optimizer configuration, see [optimizers](#), main optimizer will be used for baseline if none, a float implies none and specifies a custom weight for the baseline loss (: none).
- **baseline_objective** (*specification*) – Baseline optimization objective configuration, see [objectives](#), main objective will be used for baseline if none (: 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 and saver default filename (: “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 (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).

- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels** (*all* | iter[string]) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Deep Q-Network

```
class tensorflow.agents.DeepQNetwork(states, actions, memory, batch_size,
                                     max_episode_timesteps=None, network='auto',
                                     update_frequency=None, start Updating=None,
                                     learning_rate=0.0003, huber_loss=0.0, horizon=0,
                                     discount=0.99, estimate_terminal=False, target_sync_frequency=1,
                                     target_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, parallel_interactions=1, seed=None,
                                     execution=None, saver=None, summarizer=None, recorder=None, config=None)
```

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

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **memory** (*int > 0*) – Replay memory capacity, has to fit at least maximum `batch_size` + maximum network/estimator horizon + 1 timesteps ().
- **batch_size** (*parameter, long > 0*) – Number of timesteps per update batch ().

- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: batch_size).
- **start_updating** (*parameter*, *long* >= *batch_size*) – Number of timesteps before first update (: none).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 3e-4).
- **huber_loss** (*parameter*, *float* > 0.0) – Huber loss threshold (: no huber loss).
- **horizon** ("episode" | *parameter*, *long* >= 0) – Horizon of discounted-sum reward estimation before critic estimate (: 0).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **target_sync_frequency** (*parameter*, *int* > 0) – Interval between target network updates (: every update).
- **target_update_weight** (*parameter*, 0.0 < *float* <= 1.0) – Target network update weight (: 1.0).
- **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 and saver default filename (: "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).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):

- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
- labels (<i>”all” | iter[string]</i>) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 ”distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 ”dropout”: dropout zero fraction
 ”entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 ”graph”: graph summary
 ”kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 ”losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 ”parameters”: parameter scalars
 ”relu”: ReLU activation zero fraction
 ”rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 ”update-norm”: update norm
 ”updates”: update mean and variance scalars
 ”updates-histogram”: update histograms
 ”variables”: variable mean and variance scalars
 ”variables-histogram”: variable histograms

- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):


```
class tensorforce.agents.DuelingDQN(states, actions, memory, batch_size,
                                     max_episode_timesteps=None, network='auto',
                                     update_frequency=None, start_updating=None,
                                     learning_rate=0.0003, huber_loss=0.0, horizon=0,
                                     discount=0.99, estimate_terminal=False, target_sync_frequency=1,
                                     target_update_weight=1.0, pre-processing=None,
                                     exploration=0.0, variable_noise=0.0,
                                     l2_regularization=0.0, entropy_regularization=0.0,
                                     name='agent', device=None, parallel_interactions=1,
                                     seed=None, execution=None, saver=None, summarizer=None,
                                     recorder=None, config=None)
```

Dueling DQN agent (specification key: `dueling_dqn`).

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int* > 0) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **memory** (*int* > 0) – Replay memory capacity, has to fit at least maximum `batch_size` + maximum network/estimator horizon + 1 timesteps ().
- **batch_size** (*parameter*, *long* > 0) – Number of timesteps per update batch ().
- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: “auto”, automatically configured network).

- **update_frequency** (*"never" | parameter, long > 0*) – Frequency of updates (: *batch_size*).
- **start_updating** (*parameter, long >= batch_size*) – Number of timesteps before first update (: *none*).
- **learning_rate** (*parameter, float > 0.0*) – Optimizer learning rate (: *3e-4*).
- **huber_loss** (*parameter, float > 0.0*) – Huber loss threshold (: *no huber loss*).
- **horizon** (*"episode" | parameter, long >= 0*) – Horizon of discounted-sum reward estimation before critic estimate (: *0*).
- **discount** (*parameter, 0.0 <= float <= 1.0*) – Discount factor for future rewards of discounted-sum reward estimation (: *0.99*).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: *false*).
- **target_sync_frequency** (*parameter, int > 0*) – Interval between target network updates (: *every update*).
- **target_update_weight** (*parameter, 0.0 < float <= 1.0*) – Target network update weight (: *1.0*).
- **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 and saver default filename (: “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*).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: *none*).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: *standard*): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: *no saver*):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: *no summarizer*): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional

“max_outputs” (default: 3), or type “audio” (default: no custom summaries).

labels (*all* | iter[string]) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):

- “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
- “dropout”: dropout zero fraction
- “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
- “graph”: graph summary
- “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
- “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
- “parameters”: parameter scalars
- “relu”: ReLU activation zero fraction
- “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
- “update-norm”: update norm
- “updates”: update mean and variance scalars
- “updates-histogram”: update histograms
- “variables”: variable mean and variance scalars
- “variables-histogram”: variable histograms

- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Vanilla Policy Gradient

```
class tensorforce.agents.VanillaPolicyGradient (states, actions, max_episode_timesteps,
                                              batch_size, network='auto',
                                              use_beta_distribution=True, memory=None,
                                              update_frequency=None, learning_rate=0.0003,
                                              discount=0.99, estimate_terminal=False,
                                              baseline_network=None, baseline_optimizer=None,
                                              preprocessing=None, exploration=0.0, variable_noise=0.0,
                                              l2_regularization=0.0, entropy_regularization=0.0,
                                              name='agent', device=None, parallel_interactions=1,
                                              seed=None, execution=None, saver=None,
                                              summarizer=None, recorder=None, config=None)
```

Vanilla Policy Gradient aka REINFORCE agent (specification key: vpg).

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **batch_size** (*parameter, long > 0*) – Number of episodes per update batch ().

- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).
- **memory** (*int* > 0) – Batch memory capacity, has to fit at least maximum batch_size + 1 episodes (: minimum capacity, usually does not need to be changed).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: batch_size).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 3e-4).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **baseline_network** (*specification*) – Baseline network configuration, see [networks](#), main policy will be used as baseline if none (: none).
- **baseline_optimizer** (*float* > 0.0 | *specification*) – Baseline optimizer configuration, see [optimizers](#), main optimizer will be used for baseline if none, a float implies none and specifies a custom weight for the baseline loss (: 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 and saver default filename (: "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).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):

- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
- labels (<i>”all” | iter[string]</i>) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 ”distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 ”dropout”: dropout zero fraction
 ”entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 ”graph”: graph summary
 ”kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 ”losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 ”parameters”: parameter scalars
 ”relu”: ReLU activation zero fraction
 ”rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 ”update-norm”: update norm
 ”updates”: update mean and variance scalars
 ”updates-histogram”: update histograms
 ”variables”: variable mean and variance scalars
 ”variables-histogram”: variable histograms

- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Actor-Critic

```

class tensorforce.agents.ActorCritic(states, actions, batch_size,
                                     max_episode_timesteps=None, network='auto',
                                     use_beta_distribution=True, memory=None, up-
                                     date_frequency=None, learning_rate=0.0003, hori-
                                     zon=0, discount=0.99, state_action_value=False,
                                     estimate_terminal=False, critic_network='auto',
                                     critic_optimizer=1.0, preprocessing=None, explo-
                                     ration=0.0, variable_noise=0.0, l2_regularization=0.0,
                                     entropy_regularization=0.0, name='agent', de-
                                     vice=None, parallel_interactions=1, seed=None,
                                     execution=None, saver=None, summarizer=None,
                                     recorder=None, config=None)

```

Actor-Critic agent (specification key: `ac`).

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **batch_size** (*parameter, long > 0*) – Number of timesteps per update batch ().
- **network** (*"auto" | specification*) – Policy network configuration, see [networks](#) (: “auto”, automatically configured network).

- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: *true*).
- **memory** (*int* > 0) – Batch memory capacity, has to fit at least maximum batch_size + maximum network/estimator horizon + 1 timesteps (: minimum capacity, usually does not need to be changed).
- **update_frequency** (*"never" | parameter, long* > 0) – Frequency of updates (: batch_size).
- **learning_rate** (*parameter, float* > 0.0) – Optimizer learning rate (: 3e-4).
- **horizon** (*"episode" | parameter, long* >= 0) – Horizon of discounted-sum reward estimation before critic estimate (: 0).
- **discount** (*parameter, 0.0* <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **state_action_value** (*bool*) – Whether to estimate state-action values instead of state values (: *false*).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: *false*).
- **critic_network** (*specification*) – Critic network configuration, see [networks](#) (: "auto").
- **critic_optimizer** (*float* > 0.0 | *specification*) – Critic optimizer configuration, see [optimizers](#), a float instead specifies a custom weight for the critic loss (: 1.0).
- **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 and saver default filename (: "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).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: *none*).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: *standard*): ...

- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels (*all* | iter[string]) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Advantage Actor-Critic

```
class tensorforce.agents.AdvantageActorCritic(states,          actions,          batch_size,
                                              max_episode_timesteps=None,
                                              network='auto',
                                              use_beta_distribution=True,          memory=None,
                                              update_frequency=None,
                                              learning_rate=0.0003, horizon=0, discount=0.99,
                                              state_action_value=False,
                                              estimate_terminal=False,
                                              critic_network='auto',
                                              critic_optimizer=1.0,          preprocessing=None,
                                              exploration=0.0,          variable_noise=0.0,
                                              l2_regularization=0.0,
                                              entropy_regularization=0.0,
                                              name='agent',          device=None,
                                              parallel_interactions=1,          seed=None,
                                              execution=None,          saver=None,
                                              summarizer=None,          recorder=None,
                                              config=None)
```

Advantage Actor-Critic agent (specification key: a2c).

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:
- **max_episode_timesteps** (*int > 0*) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent`).

```
create(...)).
```

- **batch_size** (*parameter*, *long* > 0) – Number of timesteps per update batch ().
- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).
- **memory** (*int* > 0) – Batch memory capacity, has to fit at least maximum batch_size + maximum network/estimator horizon + 1 timesteps (: minimum capacity, usually does not need to be changed).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: batch_size).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 3e-4).
- **horizon** ("episode" | *parameter*, *long* >= 0) – Horizon of discounted-sum reward estimation before critic estimate (: 0).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **state_action_value** (*bool*) – Whether to estimate state-action values instead of state values (: false).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **critic_network** (*specification*) – Critic network configuration, see [networks](#) (: "auto").
- **critic_optimizer** (*float* > 0.0 | *specification*) – Critic optimizer configuration, see [optimizers](#), a float instead specifies a custom weight for the critic loss (: 1.0).
- **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 and saver default filename (: "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).

- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels (*all* | *iter[string]*) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Deterministic Policy Gradient

```
class tensorforce.agents.DeterministicPolicyGradient (states,      actions,      mem-
                                                    ory,              batch_size,
                                                    max_episode_timesteps=None,
                                                    network='auto',
                                                    use_beta_distribution=True,
                                                    update_frequency=None,
                                                    start_updating=None,  learn-
                                                    ing_rate=0.0003,      hori-
                                                    zon=0,              discount=0.99,
                                                    estimate_terminal=False,
                                                    critic_network='auto',
                                                    critic_optimizer=1.0,
                                                    preprocessing=None,
                                                    exploration=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,          summa-
                                                    rizer=None,  recorder=None,
                                                    config=None)
```

Deterministic Policy Gradient agent (specification key: `dpg`). Action space is required to consist of only a single float action.

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via

`environment` argument for `Agent.create(...)`, arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:

- **max_episode_timesteps** (*int* > 0) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **memory** (*int* > 0) – Replay memory capacity, has to fit at least maximum `batch_size` + maximum network/estimator horizon + 1 timesteps ().
- **batch_size** (*parameter*, *long* > 0) – Number of timesteps per update batch ().
- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: `batch_size`).
- **start Updating** (*parameter*, *long* >= `batch_size`) – Number of timesteps before first update (: none).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 3e-4).
- **horizon** ("episode" | *parameter*, *long* >= 0) – Horizon of discounted-sum reward estimation before critic estimate (: 0).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **critic_network** (*specification*) – Critic network configuration, see [networks](#) (: none).
- **critic_optimizer** (*float* > 0.0 | *specification*) – Critic optimizer configuration, see [optimizers](#), a float instead specifies a custom weight for the critic loss (: 1.0).
- **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 and saver default filename (: "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).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels** (*all* | `iter[string]`) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Proximal Policy Optimization

```
class tensorforce.agents.ProximalPolicyOptimization(states, actions,
                                                    max_episode_timesteps,
                                                    batch_size, network='auto',
                                                    use_beta_distribution=True,
                                                    memory=None, update_frequency=None, learning_rate=0.0003,
                                                    subsampling_fraction=0.33, optimization_steps=10, likelihood_ratio_clipping=0.2,
                                                    discount=0.99, estimate_terminal=False,
                                                    critic_network=None, critic_optimizer=None, pre_processing=None, exploration=0.0,
                                                    variable_noise=0.0, l2_regularization=0.0, entropy_regularization=0.0,
                                                    name='agent', device=None, parallel_interactions=1,
                                                    seed=None, execution=None, saver=None, summarizer=None,
                                                    recorder=None, config=None)
```

Proximal Policy Optimization agent (specification key: ppo).

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of

action descriptions (usually taken from `Environment.actions()`) with the following attributes:

- **max_episode_timesteps** (*int* > 0) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **batch_size** (*parameter*, *long* > 0) – Number of episodes per update batch ().
- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).
- **memory** (*int* > 0) – Batch memory capacity, has to fit at least maximum `batch_size` + 1 episodes (: minimum capacity, usually does not need to be changed).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: `batch_size`).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 3e-4).
- **subsampling_fraction** (*parameter*, 0.0 < *float* <= 1.0) – Fraction of batch timesteps to subsample (: 0.33).
- **optimization_steps** (*parameter*, *int* > 0) – Number of optimization steps (: 10).
- **likelihood_ratio_clipping** (*parameter*, *float* > 0.0) – Likelihood-ratio clipping threshold (: 0.2).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **critic_network** (*specification*) – Critic network configuration, see [networks](#), main policy will be used as critic if none (: none).
- **critic_optimizer** (*float* > 0.0 | *specification*) – Critic optimizer configuration, see [optimizers](#), main optimizer will be used for critic if none, a float implies none and specifies a custom weight for the critic loss (: 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 and saver default filename (: “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).
- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to fit at least set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels** (*all* | *iter[string]*) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

Trust-Region Policy Optimization

```
class tensorforce.agents.TrustRegionPolicyOptimization (states, actions,
                                                         max_episode_timesteps,
                                                         batch_size, network='auto',
                                                         use_beta_distribution=True,
                                                         memory=None, up-
                                                         date_frequency=None,
                                                         learning_rate=0.001,
                                                         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, con-
                                                         fig=None)
```

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

Parameters

- **states** (*specification*) – States specification (, better implicitly specified via `environment` argument for `Agent.create(...)`), arbitrarily nested dictionary of state descriptions (usually taken from `Environment.states()`) with the following attributes:
- **actions** (*specification*) – Actions specification (, better implicitly specified via

`environment` argument for `Agent.create(...)`, arbitrarily nested dictionary of action descriptions (usually taken from `Environment.actions()`) with the following attributes:

- **max_episode_timesteps** (*int* > 0) – Upper bound for number of timesteps per episode (: not given, better implicitly specified via `environment` argument for `Agent.create(...)`).
- **batch_size** (*parameter*, *long* > 0) – Number of episodes per update batch ().
- **network** ("auto" | *specification*) – Policy network configuration, see [networks](#) (: "auto", automatically configured network).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).
- **memory** (*int* > 0) – Batch memory capacity, has to fit at least maximum `batch_size` + 1 episodes (: minimum capacity, usually does not need to be changed).
- **update_frequency** ("never" | *parameter*, *long* > 0) – Frequency of updates (: `batch_size`).
- **learning_rate** (*parameter*, *float* > 0.0) – Optimizer learning rate (: 1e-3).
- **discount** (*parameter*, 0.0 <= *float* <= 1.0) – Discount factor for future rewards of discounted-sum reward estimation (: 0.99).
- **estimate_terminal** (*bool*) – Whether to estimate the value of (real) terminal states (: false).
- **critic_network** (*specification*) – Critic network configuration, see [networks](#), main policy will be used as critic if none (: none).
- **critic_optimizer** (*float* > 0.0 | *specification*) – Critic optimizer configuration, see [optimizers](#), main optimizer will be used for critic if none, a float implies none and specifies a custom weight for the critic loss (: 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 and saver default filename (: "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).

- **seed** (*int*) – Random seed to set for Python, NumPy (both set globally!) and TensorFlow, environment seed has to be set separately for a fully deterministic execution (: none).
- **execution** (*specification*) – TensorFlow execution configuration with the following attributes (: standard): ...
- **saver** (*specification*) – TensorFlow saver configuration for periodic implicit saving, as alternative to explicit saving via `agent.save(...)`, with the following attributes (: no saver):
- **summarizer** (*specification*) – TensorBoard summarizer configuration with the following attributes (: no summarizer): , specification with either type “scalar”, type “histogram” with optional “buckets”, type “image” with optional “max_outputs” (default: 3), or type “audio” (default: no custom summaries).
 - labels (*all* | *iter[string]*) – all excluding “*-histogram” labels, or list of summaries to record, from the following labels (default: only “graph”):
 - “distributions” or “bernoulli”, “categorical”, “gaussian”, “beta”: distribution-specific parameters
 - “dropout”: dropout zero fraction
 - “entropies” or “entropy”, “action-entropies”: entropy of policy distribution(s)
 - “graph”: graph summary
 - “kl-divergences” or “kl-divergence”, “action-kl-divergences”: KL-divergence of previous and updated policy distribution(s)
 - “losses” or “loss”, “objective-loss”, “regularization-loss”, “baseline-loss”, “baseline-objective-loss”, “baseline-regularization-loss”: loss scalars
 - “parameters”: parameter scalars
 - “relu”: ReLU activation zero fraction
 - “rewards” or “timestep-reward”, “episode-reward”, “raw-reward”, “empirical-reward”, “estimated-reward”: reward scalar
 - “update-norm”: update norm
 - “updates”: update mean and variance scalars
 - “updates-histogram”: update histograms
 - “variables”: variable mean and variance scalars
 - “variables-histogram”: variable histograms
- **recorder** (*specification*) – Experience traces recorder configuration, currently not including internal states, with the following attributes (: no recorder):

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

Parameters

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

```
class tensorflowforce.core.distributions.Beta(name, action_spec, embedding_shape, summary_labels=None)
    Beta distribution, for bounded continuous actions (specification key: beta).
```

Parameters

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

```
class tensorflowforce.core.distributions.Categorical(name, action_spec, embedding_shape, advance_based=False, summary_labels=None)
    Categorical distribution, for discrete integer actions (specification key: categorical).
```

Parameters

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

- **action_spec** (*specification*) – Action specification ().
- **embedding_shape** (*iter[int > 0]*) – Embedding shape ().
- **advantage_based** (*bool*) – Whether to compute action values as state value plus advantage (: false).
- **summary_labels** (*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

[illegible]

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

Parameters

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

Default layer: Function with default argument function

20.1 Convolutional layers

```
class tensorflow.core.layers.Conv1d(name, size, window=3, stride=1, padding='same',
                                     dilation=1, bias=True, activation='relu', dropout=0.0,
                                     is_trainable=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').
- **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*, $0.0 \leq \text{float} < 1.0$) – 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** (*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, 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).
- **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).

20.2 Dense layers

```
class tensorforce.core.layers.Dense(name, size, bias=True, activation='tanh', dropout=0.0,
                                   is_trainable=True, input_spec=None, summary_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 (: tanh).
- **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).

```
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** (*int* >= 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** (*float* >= 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).

20.3 Embedding layers

```
class tensorforce.core.layers.Embedding(name, size, num_embeddings=None,
                                         max_norm=None, bias=True, 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** (*int* >= 0) – Layer output size, 0 implies additionally removing the axis ().
- **num_embeddings** (*int* > 0) – If set, specifies the number of embeddings (: none).
- **max_norm** (*float*) – If set, embeddings are clipped if their L2-norm is larger (: none).
- **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 (: tanh).
- **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 potential parent class.

20.4 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** (*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 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*, $0.0 \leq \text{float} < 1.0$) – 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** (*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=True,
                                activation='tanh', 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 (: true).
- (**'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** (*float* ≥ 0.0) – Scalar controlling L2 regularization (: inherit value of parent module).
- **kwargs** – Additional arguments for Keras RNN layer, see [TensorFlow docs](#).

20.5 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).

20.6 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).

20.7 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.Reshape (name, shape, input_spec=None, summary_labels=None)
```

Reshape layer (specification key: reshape).

Parameters

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

20.8 Layers with internal states

```
class tensorforce.core.layers.InternalGru (name, size, length, 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) – For truncated backpropagation through time ().
- **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, length, 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) – For truncated backpropagation through time ().
- **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.InternalRnn(name, cell, size, length, bias=True, activation='tanh', 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) – For truncated backpropagation through time ().
- **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 (: 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 Keras RNN cell layer, see [TensorFlow docs](#).

20.9 Special layers

class tensorforce.core.layers.**Block** (*name*, *layers*, *input_spec=None*)
Block of layers (specification key: block).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **layers** (*iter[specification]*) – Layers configuration, see [layers](#) ().
- **input_spec** (*specification*) – Input tensor specification ().

class tensorforce.core.layers.**Function** (*name*, *function*, *output_spec=None*, *input_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** ($\text{float} \geq 0.0$) – 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*, *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: 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, is_trainable=True, input_spec=None)
```

Reuse layer (specification key: `reuse`).

Parameters

- **name** (*string*) – Layer name (: internally chosen).
- **layer** (*string*) – Name of a previously defined layer ().
- **is_trainable** (*bool*) – Whether reused layer variables are kept trainable (: true).
- **input_spec** (*specification*) – Input tensor specification ().

Default memory: Replay with default argument capacity

```
class tensorforce.core.memories.Recent (name,      capacity=None,    values_spec=None,
                                         min_capacity=0,    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 (: minimum capacity).
- **values_spec** (*specification*) – Values specification ().
- **min_capacity** (*int* >= 0) – Minimum memory capacity ().
- **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=None,    values_spec=None,
                                         min_capacity=0,    device=None,    summary_labels=None)
```

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

Parameters

- **name** (*string*) – Memory name ().
- **capacity** (*int* > 0) – Memory capacity (: minimum capacity).
- **values_spec** (*specification*) – Values specification ().
- **min_capacity** (*int* >= 0) – Minimum memory capacity ().
- **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, final_depth=1,
                                             internal_rnn=False, device=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 for truncated backpropagation through time (: *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 tensorforce.core.networks.LayeredNetwork (name, layers, inputs_spec, device=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.DeterministicPolicyGradient` (*name*, *summary_labels=None*)
 Deterministic policy gradient objective (specification key: `det_policy_gradient`).

Parameters

- **name** (*string*) – Module name ().
- **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*, *early_reduce=True*, *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).
- **early_reduce**(*bool*) – Whether to compute objective for reduced likelihoods instead of per likelihood (: true).
- **summary_labels**(*'all' | iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.objectives.Value(name,          value='state',      huber_loss=0.0,  
                                     early_reduce=True, summary_labels=None)
```

Value approximation objective, which minimizes the L2-distance between the state-(action-)value estimate and the target reward value (specification key: *value*).

Parameters

- **name**(*string*) – Module name ().
- **value**(*"state" | "action"*) – Whether to approximate the state- or state-action-value (: “state”).
- **huber_loss**(*parameter*, *float* ≥ 0.0) – Huber loss threshold (: no huber loss).
- **early_reduce**(*bool*) – Whether to compute objective for reduced values instead of value per action (: true).
- **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** (*0.0 <= float <= 1.0*) – 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, 0.0 <= float <= 1.0*) – Line search acceptance ratio (: 0.9).
- **ls_mode** (*'exponential' | 'linear'*) – Line search mode, see line search solver (: 'exponential').
- **ls_parameter** (*parameter, 0.0 <= float <= 1.0*) – 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 <= float <= 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, int >= 1*) – Interval between updates which also perform a synchronization step (: every update).
- **update_weight** (*parameter, 0.0 <= float <= 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`, `adamax`, `adamw`, `ftrl`, `lazyadam`, `nadam`, `radam`, `ranger`, `rmsprop`, `sgd`, `sgdw`)

Parameters

- **name** (*string*) – Module name ().
- **optimizer** (`adadelata` | `adagrad` | `adam` | `adamax` | `adamw` | `ftrl` | `lazyadam` | `nadam` | `radam` | `ranger` | `rmsprop` | `sgd` | `sgdw`) – TensorFlow optimizer name, see [TensorFlow docs](#) and [TensorFlow Addons docs](#) (unless given by specification key).
- **learning_rate** (*parameter, float >= 0.0*) – Learning rate (: 3e-4).
- **gradient_norm_clipping** (*parameter, 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, special values “decoupled_weight_decay”, “lookahead” and “moving_average”, see [TensorFlow docs](#) and [TensorFlow Addons docs](#).

Parameters

Default parameter: Constant

```
class tensorflow.core.parameters.Constant(name, value, dtype, min_value=None,
                                         max_value=None, summary_labels=None)
```

Constant hyperparameter.

Parameters

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

```
class tensorflow.core.parameters.Decaying(name, dtype, unit, decay, initial_value,
                                         decay_steps, increasing=False, inverse=False,
                                         scale=1.0, min_value=None, max_value=None,
                                         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" | "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).
- **min_value** (*dtype-compatible value*) – Lower parameter value bound ().
- **max_value** (*dtype-compatible value*) – Upper parameter value bound ().
- **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, absolute=False, min_value=None,
                                                    max_value=None, 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).
- **absolute** (*bool*) – Absolute value (: false).
- **min_value** (*dtype-compatible value*) – Lower parameter value bound ().
- **max_value** (*dtype-compatible value*) – Upper parameter value bound ().
- **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, min_value=None,
                                                    max_value=None, 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 ().
- **min_value** (*dtype-compatible value*) – Lower parameter value bound ().
- **max_value** (*dtype-compatible value*) – Upper parameter value bound ().
- **summary_labels** ('all' | *iter[string]*) – Labels of summaries to record (: inherit value of parent module).

```
class tensorforce.core.parameters.Random(name, dtype, distribution, shape=(),
                                         min_value=None, max_value=None, 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 ().
- **min_value** (*dtype-compatible value*) – Lower parameter value bound ().
- **max_value** (*dtype-compatible value*) – Upper parameter value bound ().
- **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*, $0.0 \leq \text{float} < 1.0$) – 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*, $0.0 \leq \text{float} \leq 1.0$) – 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.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, network='auto',
                                                         distributions=None,
                                                         temperature=0.0,
                                                         use_beta_distribution=True,
                                                         infer_state_value=False,
                                                         device=None, summary_labels=None,
                                                         l2_regularization=None,
                                                         states_spec=None, actions_spec=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 ().
- **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).
- **temperature** (*parameter | dict[parameter], float >= 0.0*) – Sampling temperature, global or per action (: 0.0).
- **use_beta_distribution** (*bool*) – Whether to use the Beta distribution for bounded continuous actions by default. (: true).

- **infer_state_value** (*False* | *"action-values"* | *"distribution"*) – Whether to infer the state value from either the action values or (experimental) the distribution parameters (: *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).
- **states_spec** (*specification*) – States specification ().
- **actions_spec** (*specification*) – Actions specification ().

```
class tensorforce.execution.Runner(agent, environment=None, max_episode_timesteps=None,  
                                   evaluation=False, num_parallel=None, environ-  
                                   ments=None, remote=None, blocking=False, host=None,  
                                   port=None)
```

Tensorforce runner utility.

Parameters

- **agent** (*specification | Agent object*) – Agent specification or object, the latter is not closed automatically as part of `runner.close()`, `parallel_interactions` is implicitly specified as / expected to be at least `num_parallel`, -1 if evaluation ().
- **environment** (*specification | Environment object*) – Environment specification or object, the latter is not closed automatically as part of `runner.close()` (, or alternatively `environments`, invalid for “socket-client” remote mode).
- **max_episode_timesteps** (*int > 0*) – Maximum number of timesteps per episode, overwrites the environment default if defined (: environment default, invalid for “socket-client” remote mode).
- **evaluation** (*bool*) – Whether to run the (last if multiple) environment in evaluation mode (: no evaluation).
- **num_parallel** (*int > 0*) – Number of environment instances to execute in parallel (: no parallel execution, implicitly specified by `environments`).
- **environments** (*list[specification | Environment object]*) – Environment specifications or objects to execute in parallel, the latter are not closed automatically as part of `runner.close()` (: no parallel execution, alternatively specified via `environment` and `num_parallel`, invalid for “socket-client” remote mode).
- **remote** (*"multiprocessing" | "socket-client"*) – Communication mode for remote environment execution of parallelized environment execution, not compatible with environment(s) given as `Environment` objects, “socket-client” mode requires a corresponding “socket-server” running (: local execution).

- **blocking** (*bool*) – Whether remote environment calls should be blocking, only valid if remote mode given (: not blocking, invalid unless “multiprocessing” or “socket-client” remote mode).
- **host** (*str*, *iter[str]*) – Socket server hostname(s) or IP address(es) (only for “socket-client” remote mode).
- **port** (*int*, *iter[int]*) – Socket server port(s), increasing sequence if single host and port given (only for “socket-client” remote mode).

run (*num_episodes=None*, *num_timesteps=None*, *num_updates=None*, *batch_agent_calls=False*, *sync_timesteps=False*, *sync_episodes=False*, *num_sleep_secs=0.001*, *callback=None*, *callback_episode_frequency=None*, *callback_timestep_frequency=None*, *use_tqdm=True*, *mean_horizon=1*, *evaluation=False*, *save_best_agent=None*, *evaluation_callback=None*)
Run experiment.

Parameters

- **num_episodes** (*int* > 0) – Number of episodes to run experiment (: no episode limit).
- **num_timesteps** (*int* > 0) – Number of timesteps to run experiment (: no timestep limit).
- **num_updates** (*int* > 0) – Number of agent updates to run experiment (: no update limit).
- **batch_agent_calls** (*bool*) – Whether to batch agent calls for parallel environment execution (: separate call per environment).
- **sync_timesteps** (*bool*) – Whether to synchronize parallel environment execution on timestep-level, implied by `batch_agent_calls` (: not synchronized unless `batch_agent_calls`).
- **sync_episodes** (*bool*) – Whether to synchronize parallel environment execution on episode-level (: not synchronized).
- **num_sleep_secs** (*float*) – Sleep duration if no environment is ready (: one milliseconds).
- **callback** ((*Runner*, *parallel*) -> *bool*) – Callback function taking the runner instance plus parallel index and returning a boolean value indicating whether execution should continue (: callback always true).
- **callback_episode_frequency** (*int*) – Episode interval between callbacks (: every episode).
- **callback_timestep_frequency** (*int*) – Timestep interval between callbacks (: not specified).
- **use_tqdm** (*bool*) – Whether to display a tqdm progress bar for the experiment run (: true), with the following additional information (averaged over number of episodes given via `mean_horizon`):
- **mean_horizon** (*int*) – Number of episodes progress bar values and evaluation score are averaged over (: not averaged).
- **evaluation** (*bool*) – Whether to run in evaluation mode, only valid if a single environment (: no evaluation).
- **save_best_agent** (*string*) – Directory to save the best version of the agent according to the evaluation score (: best agent is not saved).

- **evaluation_callback**(*int* / *Runner* -> *float*) – Callback function taking the runner instance and returning an evaluation score (: cumulative evaluation reward averaged over mean_horizon episodes).

29.1 Initialization and termination

static `Environment.create(environment=None, max_episode_timesteps=None, remote=None, blocking=False, host=None, port=None, **kwargs)`

Creates an environment from a specification. In case of “socket-server” remote mode, runs environment in server communication loop until closed.

Parameters

- **environment** (*specification | Environment class/object*) – JSON file, specification key, configuration dictionary, library module, `Environment` class/object, or `gym.Env()`.
- **max_episode_timesteps** (*int > 0*) – Maximum number of timesteps per episode, overwrites the environment default if defined (: environment default, invalid for “socket-client” remote mode).
- **remote** (*"multiprocessing" | "socket-client" | "socket-server"*) – Communication mode for remote environment execution of parallelized environment execution, “socket-client” mode requires a corresponding “socket-server” running, and “socket-server” mode runs environment in server communication loop until closed (: local execution).
- **blocking** (*bool*) – Whether remote environment calls should be blocking (: not blocking, invalid unless “multiprocessing” or “socket-client” remote mode).
- **host** (*str*) – Socket server hostname or IP address (only for “socket-client” remote mode).
- **port** (*int*) – Socket server port (only for “socket-client/server” remote mode).
- **kwargs** – Additional arguments.

`Environment.close()`

Closes the environment.

29.2 Attributes

`Environment.states()`

Returns the state space specification.

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

Return type specification

`Environment.actions()`

Returns the action space specification.

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

Return type specification

`Environment.max_episode_timesteps()`

Returns the maximum number of timesteps per episode.

Returns Maximum number of timesteps per episode.

Return type int

29.3 Interaction functions

`Environment.reset()`

Resets the environment to start a new episode.

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

Return type dict[state]

`Environment.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

Arcade Learning Environment

```
class tensorforce.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  
  
git clone https://github.com/mgbellemare/Arcade-Learning-Environment.git  
cd Arcade-Learning-Environment  
  
mkdir build && cd build  
cmake -DUSE_SDL=ON -DUSE_RLGLUE=OFF -DBUILD_EXAMPLES=ON ..  
make -j 4  
cd ..  
  
pip3 install .
```

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  
  
pip3 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_steps=None,  
                                         terminal_reward=0.0, reward_threshold=None,  
                                         drop_states_indices=None, visualize_directory=None, **kwargs)
```

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

May require:

```
pip3 install gym  
pip3 install gym[all]
```

Parameters

- **level** (*string* | *gym.Env*) – Gym id or instance ().
- **visualize** (*bool*) – Whether to visualize interaction (: false).
- **max_episode_steps** (*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).
- **drop_states_indices** (*list[int]*) – Drop states indices (: none).
- **visualize_directory** (*string*) – Visualization output directory (: none).
- **kwargs** – Additional Gym environment arguments.


```
class tensorforce.environments.OpenAIRetro (level, visualize=False, visual-  
                                           ize_directory=None, **kwargs)  
    OpenAI Retro environment adapter (specification key: retro, openai_retro).
```

May require:

```
pip3 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

pip3 install git+https://github.com/pygame/pygame.git

pip3 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 “Catcher”, “Doom”, “FlappyBird”, “MonsterKong”, “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).

May require:

```
sudo apt-get install g++ build-essential libsdl2-dev zlib1g-dev libmpg123-dev  
↳ libjpeg-dev      libsndfile1-dev nasm tar libbz2-dev libgtk2.0-dev make cmake  
↳ git chrpath timidity libfluidsynth-dev libgme-dev libopenal-dev timidity  
↳ libwildmidi-dev unzip libboost-all-dev liblua5.1-dev  
  
pip3 install 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()` (*tensorflow.agents.TensorforceAgent* method), 21

`actions()` (*tensorflow.environments.Environment* method), 108

Activation (class in *tensorflow.core.layers*), 75

ActorCritic (class in *tensorflow.agents*), 47

AdvantageActorCritic (class in *tensorflow.agents*), 51

ArcadeLearningEnvironment (class in *tensorflow.environments*), 109

`assign_variable()` (*tensorflow.agents.TensorforceAgent* method), 23

AutoNetwork (class in *tensorflow.core.networks*), 83

B

Bernoulli (class in *tensorflow.core.distributions*), 67

Beta (class in *tensorflow.core.distributions*), 67

Block (class in *tensorflow.core.layers*), 79

C

Categorical (class in *tensorflow.core.distributions*), 67

Clipping (class in *tensorflow.core.layers*), 75

ClippingStep (class in *tensorflow.core.optimizers*), 87

`close()` (*tensorflow.agents.TensorforceAgent* method), 21

`close()` (*tensorflow.environments.Environment* method), 107

Constant (class in *tensorflow.core.parameters*), 93

ConstantAgent (class in *tensorflow.agents*), 27

Conv1d (class in *tensorflow.core.layers*), 69

Conv2d (class in *tensorflow.core.layers*), 70

`create()` (*tensorflow.agents.TensorforceAgent* static method), 21

`create()` (*tensorflow.environments.Environment* static method), 107

D

Decaying (class in *tensorflow.core.parameters*), 93

DeepQNetwork (class in *tensorflow.agents*), 35

Deltafier (class in *tensorflow.core.layers*), 76

Dense (class in *tensorflow.core.layers*), 70

DeterministicPolicyGradient (class in *tensorflow.agents*), 55

DeterministicPolicyGradient (class in *tensorflow.core.objectives*), 85

Dropout (class in *tensorflow.core.layers*), 76

DuelingDQN (class in *tensorflow.agents*), 39

E

Embedding (class in *tensorflow.core.layers*), 71

Evolutionary (class in *tensorflow.core.optimizers*), 87

`execute()` (*tensorflow.environments.Environment* method), 108

`experience()` (*tensorflow.agents.TensorforceAgent* method), 24

ExponentialNormalization (class in *tensorflow.core.layers*), 75

F

Flatten (class in *tensorflow.core.layers*), 73

Function (class in *tensorflow.core.layers*), 79

G

Gaussian (class in *tensorflow.core.distributions*), 68

`get_available_summaries()` (*tensorflow.agents.TensorforceAgent* method), 25

`get_output_tensors()` (*tensorflow.agents.TensorforceAgent* method), 25

`get_variable()` (*tensorflow.agents.TensorforceAgent* method), 23

`get_variables()` (*tensorforce.agents.TensorforceAgent* method), 23

`GlobalOptimizer` (class in *tensorforce.core.optimizers*), 88

`Gru` (class in *tensorforce.core.layers*), 72

I

`Image` (class in *tensorforce.core.layers*), 76

`initial_internals()` (*tensorforce.agents.TensorforceAgent* method), 22

`InstanceNormalization` (class in *tensorforce.core.layers*), 75

`InternalGru` (class in *tensorforce.core.layers*), 77

`InternalLstm` (class in *tensorforce.core.layers*), 78

`InternalRnn` (class in *tensorforce.core.layers*), 78

K

`Keras` (class in *tensorforce.core.layers*), 79

L

`LayeredNetwork` (class in *tensorforce.core.networks*), 83

`Linear` (class in *tensorforce.core.layers*), 71

`load()` (*tensorforce.agents.TensorforceAgent* static method), 22

`Lstm` (class in *tensorforce.core.layers*), 72

M

`max_episode_timesteps()` (*tensorforce.environments.Environment* method), 108

`MazeExplorer` (class in *tensorforce.environments*), 111

`MetaOptimizerWrapper` (class in *tensorforce.core.optimizers*), 88

`MultiStep` (class in *tensorforce.core.optimizers*), 88

N

`NaturalGradient` (class in *tensorforce.core.optimizers*), 88

O

`observe()` (*tensorforce.agents.TensorforceAgent* method), 22

`OpenAIGym` (class in *tensorforce.environments*), 115

`OpenAIRetro` (class in *tensorforce.environments*), 117

`OpenSim` (class in *tensorforce.environments*), 113

`OptimizingStep` (class in *tensorforce.core.optimizers*), 89

`OrnsteinUhlenbeck` (class in *tensorforce.core.parameters*), 94

P

`ParametrizedDistributions` (class in *tensorforce.core.policies*), 101

`PiecewiseConstant` (class in *tensorforce.core.parameters*), 94

`Plus` (class in *tensorforce.core.objectives*), 85

`Plus` (class in *tensorforce.core.optimizers*), 89

`PolicyGradient` (class in *tensorforce.core.objectives*), 85

`Pool1d` (class in *tensorforce.core.layers*), 74

`Pool2d` (class in *tensorforce.core.layers*), 74

`Pooling` (class in *tensorforce.core.layers*), 74

`pretrain()` (*tensorforce.agents.TensorforceAgent* method), 24

`ProximalPolicyOptimization` (class in *tensorforce.agents*), 59

`PyGameLearningEnvironment` (class in *tensorforce.environments*), 119

R

`Random` (class in *tensorforce.core.parameters*), 95

`RandomAgent` (class in *tensorforce.agents*), 29

`Recent` (class in *tensorforce.core.memories*), 81

`Register` (class in *tensorforce.core.layers*), 79

`Replay` (class in *tensorforce.core.memories*), 81

`reset()` (*tensorforce.agents.TensorforceAgent* method), 25

`reset()` (*tensorforce.environments.Environment* method), 108

`Reshape` (class in *tensorforce.core.layers*), 77

`Retrieve` (class in *tensorforce.core.layers*), 80

`Reuse` (class in *tensorforce.core.layers*), 80

`Rnn` (class in *tensorforce.core.layers*), 73

`run()` (*tensorforce.execution.Runner* method), 104

`Runner` (class in *tensorforce.execution*), 103

S

`save()` (*tensorforce.agents.TensorforceAgent* method), 23

`Sequence` (class in *tensorforce.core.layers*), 77

`states()` (*tensorforce.environments.Environment* method), 108

`SubsamplingStep` (class in *tensorforce.core.optimizers*), 90

`summarize()` (*tensorforce.agents.TensorforceAgent* method), 24

`Synchronization` (class in *tensorforce.core.optimizers*), 90

T

`TensorforceAgent` (class in *tensorforce.agents*), 31

`TFOptimizer` (class in *tensorforce.core.optimizers*), 90

`TrustRegionPolicyOptimization` (class in *tensorforce.agents*), 63

U

`update()` (*tensorforce.agents.TensorforceAgent*
method), [24](#)

V

`Value` (*class in tensorforce.core.objectives*), [86](#)

`VanillaPolicyGradient` (*class in tensor-*
force.agents), [43](#)

`VizDoom` (*class in tensorforce.environments*), [121](#)