HyperParam (Dimension)

_cb

register_callback(p_cb)
callback_on_change(p_value)

HyperParamSpace (ESpace)

HyperParamTupel (Element)

Model (Log, LoadSave, Plottable, ScientificObject)

C_TYPE = 'Model'
C_NAME = '????'
C_BUFFER_CLS = Buffer
C_SCIREF_TYPE = C_SCIREF_TYPE_NONE
_adaptivity
_adapted:bool
_hyperparam_space: HyperParamSpace
_hyperparam_tupel: HyperParamTupel
_buffer

__init__(p_buffer_size=0,
         p_ada=True,
         p_logging=True)
_init_hyperparam()
get_hyperparam()
switch_adaptivity(p_ada)
set_random_seed(p_seed=None)
get_adapted()
_set_adapted(p_adapted)
adapt(*p_args)
_adapt(*p_args)
clear_buffer()

Mode (Log)

C_MODE_SIM = 0
C_MODE_REAL = 1
_mode

__init__(p_mode, p_logging=True)
get_mode()
set_mode(p_mode)

AdaptiveFunction (Model, Function)

_threshold
_mappings_total
_mappings_good

__init__(p_input_space:MSpace,
         p_output_space:MSpace,
         p_output_elem_cls=Element,
         p_threshold=0,
         p_buffer_size=0,
         p_ada=True,
         p_logging=True)
adapt(p_input: Element,
      p_output: Element): Bool
_adapt(p_input: Element,
       p_output: Element): Bool
get_maturity()

TrainingResults (Saveable)

scenario: Scenario
ts_start
ts_end
duration
score
trained_model: Model

__init__(p_scenario:Scenario)

Scenario (Mode, LoadSave)

C_TYPE = 'Scenario'
C_NAME = '????'
_model: Model

__init__(p_mode,
         p_ada=True,
         p_logging=True)
_setup(p_mode, p_ada, p_logging): Adaptive
get_model(): Model
_set_mode(p_mode)
reset(p_seed=1)
_reset(p_seed)
run_step(): Bool
_run_step(): Bool
run( )

Training (Log)

C_TYPE = 'Training'
C_NAME = '????'
_results: TrainingResults

__init__(p_scenario:Scenario,
         p_hpt:HyperParamTuner=None,
         p_hpt_trials=0,
         p_logging=True)
get_scenario(): Scenario
run(): TrainingResults
_run(): TrainingResults
get_results(): TrainingResults

HyperParamTuner (Log)

C_TYPE = 'HPT'
C_NAME = '????'

maximize(p_ofct,
         p_model: Model,
         p_num_trials): TrainingResult
_maximize( ): TrainingResults

WrHPTHyperopt (HyperParamTuner)

C_NAME = 'Hyperopt'

_maximize( ): TrainingResults
_ofct_hyperopt(...): float

WrHPTOptuna (HyperParamTuner)

C_NAME = 'Optuna'

_maximize( ): TrainingResults
_ofct_optuna(p_trial): float

WrHPTRayTune (HyperParamTuner)

C_NAME = 'RAY Tune'

_maximize( ): TrainingResults
_ofct_raytune(...): float

Legend:

→ Depends on
▷ Inherits from
method() New/inherited method/attribute to be implemented
method() Method/attribute is already implemented