Client Base Classes#

Provide an abstract base class to allow the client to return summary statistics of the data. To define a new stats script, subclass this class and implement the following abstract methods: initialize(), abort(), and finalize() – inherited from ClientAlgoStats; can be optionally be implemented to help with lifecycle management of the class object. Call to abort the ClientAlgo training or evaluation. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. Call to finalize the ClientAlgo class. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. Call to initialize the ClientAlgo class. extra (Optional[dict]) – optional extra information, e.g. dict of ExtraItems.CLIENT_NAME and/or ExtraItems.APP_ROOT. Provide an abstract base class for defining algo to run on any platform. To define a new algo script, subclass this class and implement the following abstract methods: self.train() self.get_weights() self.evaluate() self.get_data_stats() (optional, inherited from ClientAlgoStats) initialize(), abort(), and finalize() - inherited from ClientAlgoStats - can be optionally be implemented to help with lifecycle management of the class object. Get evaluation metrics on test data. data (ExchangeObject) – ExchangeObject with network weights to use for evaluation. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. ExchangeObject with evaluation metrics. metrics Get current local weights or weight differences. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. current local weights or weight differences. ExchangeObject ExchangeObject example: Train network and produce new network from train data. data (ExchangeObject) – ExchangeObject containing current network weights to base training on. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. None None Get summary statistics about the local data. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. For example, requested statistics. summary statistics. ExchangeObject Extra dict example: Returned ExchangeObject example:

MONAI Bundle Reference Implementations#

Implementation of ClientAlgo to allow federated learning with MONAI bundle configurations. bundle_root (str) – path of bundle. local_epochs (int) – number of local epochs to execute during each round of local training; defaults to 1. send_weight_diff (bool) – whether to send weight differences rather than full weights; defaults to True. config_train_filename (Union[str, list, None]) – bundle training config path relative to bundle_root. Can be a list of files; defaults to “configs/train.json”. config_evaluate_filename (Union[str, list, None]) – bundle evaluation config path relative to bundle_root. Can be a list of files. If “default”, config_evaluate_filename = [“configs/train.json”, “configs/evaluate.json”] will be used; config_filters_filename (Union[str, list, None]) – filter configuration file. Can be a list of files; defaults to None. disable_ckpt_loading (bool) – do not use any CheckpointLoader if defined in train/evaluate configs; defaults to True. best_model_filepath (Optional[str]) – location of best model checkpoint; defaults “models/model.pt” relative to bundle_root. final_model_filepath (Optional[str]) – location of final model checkpoint; defaults “models/model_final.pt” relative to bundle_root. save_dict_key (Optional[str]) – If a model checkpoint contains several state dicts, the one defined by save_dict_key will be returned by get_weights; defaults to “model”. If all state dicts should be returned, set save_dict_key to None. seed (Optional[int]) – set random seed for modules to enable or disable deterministic training; defaults to None, i.e., non-deterministic training. benchmark (bool) – set benchmark to False for full deterministic behavior in

cuDNN components. Note, full determinism in federated learning depends also on deterministic behavior of other FL components, e.g., the aggregator, which is not controlled by this class. multi_gpu (bool) – whether to run MonaiAlgo in a multi-GPU setting; defaults to False. backend (str) – backend to use for torch.distributed; defaults to “nccl”. init_method (str) – init_method for torch.distributed; defaults to “env://”. tracking (Union[str, dict, None]) – enable the experiment tracking feature at runtime with optionally configurable and extensible. if “mlflow”, will add MLFlowHandler to the parsed bundle with default logging settings, if other string, treat it as file path to load the logging settings, if dict, treat it as logging settings, otherwise, use all the default settings. will patch the target config content with tracking handlers and the top-level items of configs. example of customized settings: tracking = { "handlers_id": { "trainer": {"id": "train#trainer", "handlers": "train#handlers"}, "validator": {"id": "evaluate#evaluator", "handlers": "evaluate#handlers"}, "evaluator": {"id": "evaluator", "handlers": "handlers"}, }, "configs": { "tracking_uri": "", "trainer": { "_target_": "MLFlowHandler", "tracking_uri": "@tracking_uri", "iteration_log": True, "output_transform": "$monai.handlers.from_engine(['loss'], first=True)", }, "validator": { "_target_": "MLFlowHandler", "tracking_uri": "@tracking_uri", "iteration_log": False, }, "evaluator": { "_target_": "MLFlowHandler", "tracking_uri": "@tracking_uri", "iteration_log": False, }, }, }, enable the experiment tracking feature at runtime with optionally configurable and extensible. if “mlflow”, will add MLFlowHandler to the parsed bundle with default logging settings, if other string, treat it as file path to load the logging settings, if dict, treat it as logging settings, otherwise, use all the default settings. will patch the target config content with tracking handlers and the top-level items of configs. example of customized settings: Abort the training or evaluation. :param extra: Dict with additional information that can be provided by the FL system. Evaluate on client’s local data. data (ExchangeObject) – ExchangeObject containing the current global model weights. extra – Dict with additional information that can be provided by the FL system. ExchangeObject containing evaluation metrics. return_metrics Finalize the training or evaluation. :param extra: Dict with additional information that can be provided by the FL system. Returns the current weights of the model. extra – Dict with additional information that can be provided by the FL system. ExchangeObject containing current weights (default)or load requested model type from disk (ModelType.BEST_MODEL or ModelType.FINAL_MODEL). or load requested model type from disk (ModelType.BEST_MODEL or ModelType.FINAL_MODEL). return_weights Initialize routine to parse configuration files and extract main components such as trainer, evaluator, and filters. extra – Dict with additional information that should be provided by FL system, i.e., ExtraItems.CLIENT_NAME and ExtraItems.APP_ROOT. Train on client’s local data. data (ExchangeObject) – ExchangeObject containing the current global model weights. extra – Dict with additional information that can be provided by the FL system. Implementation of ClientAlgoStats to allow federated learning with MONAI bundle configurations. bundle_root (str) – path of bundle. config_train_filename (Union[str, list, None]) – bundle training config path relative to bundle_root. Can be a list of files; defaults to “configs/train.json”. config_filters_filename (Union[str, list, None]) – filter configuration file. Can be a list of files; defaults to None. histogram_only (bool) – whether to only compute histograms. Defaults to False. Returns summary statistics about the local data. extra (Optional[dict]) – Dict with additional information that can be provided by the FL system. Both FlStatistics.HIST_BINS and FlStatistics.HIST_RANGE must be provided. ExchangeObject with summary statistics. stats Initialize routine to parse configuration files and extract main components such as trainer, evaluator, and filters. extra – Dict with additional information that should be provided by FL system, i.e., ExtraItems.CLIENT_NAME and ExtraItems.APP_ROOT. previous Auto3dseg next Model Bundle

MONAI Consortium.