tf.contrib.learn.RunConfig

Contents

Class **RunConfig**

Inherits From: `RunConfig`

Defined in `tensorflow/contrib/learn/python/learn/estimators/run_config.py`.

See the guide: Learn (contrib) > Graph actions

This class specifies the configurations for an `Estimator` run.

This class is the implementation of `tf.estimator.RunConfig` interface.

Properties

`cluster_spec`

`environment`

`evaluation_master`

`is_chief`

`keep_checkpoint_every_n_hours`

`keep_checkpoint_max`

`log_step_count_steps`

`master`

`model_dir`

`num_ps_replicas`

`num_worker_replicas`

`save_checkpoints_secs`

save_checkpoints_steps

save_summary_steps

service

Returns the platform defined (in TF_CONFIG) service dict.

session_config

task_id

task_type

tf_config

tf_random_seed

Methods

__init__

```
__init__(
    master=None,
    num_cores=0,
    log_device_placement=False,
    gpu_memory_fraction=1,
    tf_random_seed=None,
    save_summary_steps=100,
    save_checkpoints_secs=_USE_DEFAULT,
    save_checkpoints_steps=None,
    keep_checkpoint_max=5,
    keep_checkpoint_every_n_hours=10000,
    log_step_count_steps=100,
    evaluation_master='',
    model_dir=None,
    session_config=None
)
```

Constructor.

The superclass `ClusterConfig` may set properties like `cluster_spec`, `is_chief`, `master` (if `None` in the args), `num_ps_replicas`, `task_id`, and `task_type` based on the `TF_CONFIG` environment variable. See `ClusterConfig` for more details.

N.B.: If `save_checkpoints_steps` or `save_checkpoints_secs` is set, `keep_checkpoint_max` might need to be adjusted accordingly, especially in distributed training. For example, setting `save_checkpoints_secs` as 60 without adjusting `keep_checkpoint_max` (defaults to 5) leads to situation that checkpoint would be garbage collected after 5 minutes. In distributed training, the evaluation job starts asynchronously and might fail to load or find the checkpoint due to race condition.

Args:

- `master`: TensorFlow master. Defaults to empty string for local.
- `num_cores`: Number of cores to be used. If 0, the system picks an appropriate number (default: 0).

- `log_device_placement`: Log the op placement to devices (default: False).
- `gpu_memory_fraction`: Fraction of GPU memory used by the process on each GPU uniformly on the same machine.
- `tf_random_seed`: Random seed for TensorFlow initializers. Setting this value allows consistency between reruns.
- `save_summary_steps`: Save summaries every this many steps.
- `save_checkpoints_secs`: Save checkpoints every this many seconds. Can not be specified with `save_checkpoints_steps`.
- `save_checkpoints_steps`: Save checkpoints every this many steps. Can not be specified with `save_checkpoints_secs`.
- `keep_checkpoint_max`: The maximum number of recent checkpoint files to keep. As new files are created, older files are deleted. If None or 0, all checkpoint files are kept. Defaults to 5 (that is, the 5 most recent checkpoint files are kept.)
- `keep_checkpoint_every_n_hours`: Number of hours between each checkpoint to be saved. The default value of 10,000 hours effectively disables the feature.
- `log_step_count_steps`: The frequency, in number of global steps, that the global step/sec will be logged during training.
- `evaluation_master`: the master on which to perform evaluation.
- `model_dir`: directory where model parameters, graph etc are saved. If `None`, will use `model_dir` property in `TF_CONFIG` environment variable. If both are set, must have same value. If both are `None`, see `Estimator` about where the model will be saved.
- `session_config`: a ConfigProto used to set session parameters, or None. Note - using this argument, it is easy to provide settings which break otherwise perfectly good models. Use with care.

get_task_id

```
get_task_id()
```

Returns task index from `TF_CONFIG` environmental variable.

If you have a ClusterConfig instance, you can just access its task_id property instead of calling this function and re-parsing the environmental variable.

Returns:

`TF_CONFIG['task']['index']`. Defaults to 0.

replace

```
replace(**kwargs)
```

Returns a new instance of `RunConfig` replacing specified properties.

Only the properties in the following list are allowed to be replaced: - `model_dir`. - `tf_random_seed`, - `save_summary_steps`, - `save_checkpoints_steps`, - `save_checkpoints_secs`, - `session_config`, - `keep_checkpoint_max`, - `keep_checkpoint_every_n_hours`, - `log_step_count_steps`,

In addition, either `save_checkpoints_steps` or `save_checkpoints_secs` can be set (should not be both).

Args:

- `**kwargs`: keyword named properties with new values.

Raises:

- `ValueError`: If any property name in `kwargs` does not exist or is not allowed to be replaced, or both `save_checkpoints_steps` and `save_checkpoints_secs` are set.

Returns:

a new instance of `RunConfig`.

`uid`

```
uid(
    *args,
    **kwargs
)
```

Generates a 'Unique Identifier' based on all internal fields. (experimental)

THIS FUNCTION IS EXPERIMENTAL. It may change or be removed at any time, and without warning.

Caller should use the uid string to check `RunConfig` instance integrity in one session use, but should not rely on the implementation details, which is subject to change.

Args:

- `whitelist`: A list of the string names of the properties uid should not include. If `None`, defaults to `_DEFAULT_UID_WHITE_LIST`, which includes most properties user allowes to change.

Returns:

A uid string.