

tf.contrib.learn.learn_runner.run

```
run(  
    experiment_fn,  
    output_dir=None,  
    schedule=None,  
    run_config=None,  
    hparams=None  
)
```

Defined in [tensorflow/contrib/learn/python/learn/learn_runner.py](#).

Make and run an experiment.

It creates an Experiment by calling `experiment_fn`. Then it calls the function named as `schedule` of the Experiment.

If schedule is not provided, then the default schedule for the current task type is used. The defaults are as follows:

- 'ps' maps to 'serve'
- 'worker' maps to 'train'
- 'master' maps to 'local_run'

If the experiment's config does not include a task type, then an exception is raised.

Example with `run_config` (Recommended):

```
def _create_my_experiment(run_config, hparams):  
  
    # You can change a subset of the run_config properties as  
    #   run_config = run_config.replace(save_checkpoints_steps=500)  
  
    return tf.contrib.learn.Experiment(  
        estimator=my_estimator(config=run_config, hparams=hparams),  
        train_input_fn=my_train_input,  
        eval_input_fn=my_eval_input)  
  
learn_runner.run(  
    experiment_fn=_create_my_experiment,  
    run_config=run_config_lib.RunConfig(model_dir="some/output/dir"),  
    schedule="train_and_evaluate",  
    hparams=_create_default_hparams())
```

or simply as

```
learn_runner.run(  
    experiment_fn=_create_my_experiment,  
    run_config=run_config_lib.RunConfig(model_dir="some/output/dir"))
```

if `hparams` is not used by the `Estimator`. On a single machine, `schedule` defaults to `train_and_evaluate`.

Example with `output_dir` (deprecated):

```
def _create_my_experiment(output_dir):
    return tf.contrib.learn.Experiment(
        estimator=my_estimator(model_dir=output_dir),
        train_input_fn=my_train_input,
        eval_input_fn=my_eval_input)

learn_runner.run(
    experiment_fn=_create_my_experiment,
    output_dir="some/output/dir",
    schedule="train")
```

Args:

- `experiment_fn`: A function that creates an `Experiment`. It could be one of the two following signatures: 1) [Deprecated] It accepts an argument `output_dir` which should be used to create the `Estimator` (passed as `model_dir` to its constructor). It must return an `Experiment`. For this case, `run_config` and `hparams` must be `None`. 2) It accepts two arguments `run_config` and `hparams`, which should be used to create the `Estimator` (`run_config` passed as `config` to its constructor; `hparams` used as the hyper-parameters of the model). It must return an `Experiment`. For this case, `output_dir` must be `None`.
- `output_dir`: Base output directory [Deprecated].
- `schedule`: The name of the method in the `Experiment` to run.
- `run_config`: `RunConfig` instance. The `run_config.model_dir` must be non-empty. If `run_config` is set, `output_dir` must be `None`.
- `hparams`: `HParams` instance. The default hyper-parameters, which will be passed to the `experiment_fn` if `run_config` is not `None`.

Returns:

The return value of function `schedule`.

Raises:

- `ValueError`: If both `output_dir` and `run_config` are empty or set, `schedule` is `None` but no task type is set in the built experiment's config, the task type has no default, `run_config.model_dir` is empty or `schedule` doesn't reference a member of `Experiment`.
- `TypeError`: `schedule` references non-callable member.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)