tf.estimator.EvalSpec

Class **EvalSpec**

Defined in `tensorflow/python/estimator/training.py` .

Configuration for the "eval" part for the `train_and_evaluate` call.

`EvalSpec` combines details of evaluation of the trained model as well as its export. Evaluation consists of computing metrics to judge the performance of the trained model. Export writes out the trained model on to external storage.

Properties

`exporters`

Alias for field number 4

`hooks`

Alias for field number 3

`input_fn`

Alias for field number 0

`name`

Alias for field number 2

`start_delay_secs`

Alias for field number 5

`steps`

Alias for field number 1

throttle_secs

Alias for field number 6

Methods

`__new__`

```
@staticmethod
__new__(
    cls,
    input_fn,
    steps=100,
    name=None,
    hooks=None,
    exporters=None,
    start_delay_secs=120,
    throttle_secs=600
)
```

Creates a validated `EvalSpec` instance.

Args:

- `input_fn` : Evaluation input function returning a tuple of: features - `Tensor` or dictionary of string feature name to `Tensor` . labels - `Tensor` or dictionary of `Tensor` with labels.
- `steps` : Int. Positive number of steps for which to evaluate model. If `None` , evaluates until `input_fn` raises an end-of-input exception. See `Estimator.evaluate` for details.
- `name` : String. Name of the evaluation if user needs to run multiple evaluations on different data sets. Metrics for different evaluations are saved in separate folders, and appear separately in tensorboard.
- `hooks` : Iterable of `tf.train.SessionRunHook` objects to run on all workers (including chief) during training.
- `exporters` : Iterable of `Exporter` s, or a single one, or `None` . `exporters` will be invoked after each evaluation.
- `start_delay_secs` : Int. Start evaluating after waiting for this many seconds.
- `throttle_secs` : Int. Do not re-evaluate unless the last evaluation was started at least this many seconds ago. Of course, evaluation does not occur if no new checkpoints are available, hence, this is the minimum.

Returns:

A validated `EvalSpec` object.

Raises:

- `ValueError` : If any of the input arguments is invalid.
- `TypeError` : If any of the arguments is not of the expected type.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**