

tf.contrib.learn.ModelFnOps

Contents

Class ModelFnOps

Properties

eval_metric_ops

loss

Class ModelFnOps

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

See the guide: [Learn \(contrib\) > Estimators](#)

Ops returned from a model_fn.

Properties

eval_metric_ops

Alias for field number 3

loss

Alias for field number 1

mode

Alias for field number 8

output_alternatives

Alias for field number 4

predictions

Alias for field number 0

scaffold

Alias for field number 7

train_op

Alias for field number 2

training_chief_hooks

Alias for field number 5

training_hooks

Alias for field number 6

Methods

__new__

```
@staticmethod
__new__(
    cls,
    mode,
    predictions=None,
    loss=None,
    train_op=None,
    eval_metric_ops=None,
    output_alternatives=None,
    training_chief_hooks=None,
    training_hooks=None,
    scaffold=None
)
```

Creates a validated `ModelFnOps` instance.

For a multi-headed model, the predictions dict here will contain the outputs of all of the heads. However: at serving time, requests will be made specifically for one or more heads, and the RPCs used for these requests may differ by problem type (i.e., regression, classification, other). The purpose of the output_alternatives dict is to aid in exporting a SavedModel from which such head-specific queries can be served. These output_alternatives will be combined with input_alternatives (see `saved_model_export_utils`) to produce a set of `SignatureDef`s specifying the valid requests that can be served from this model.

For a single-headed model, it is still advisable to provide output_alternatives with a single entry, because this is how the problem type is communicated for export and serving. If output_alternatives is not given, the resulting SavedModel will support only one head of unspecified type.

Args:

- `mode`: One of `ModeKeys`. Specifies if this training, evaluation or prediction.
- `predictions`: Predictions `Tensor` or dict of `Tensor`.
- `loss`: Training loss `Tensor`.
- `train_op`: Op for the training step.
- `eval_metric_ops`: Dict of metric results keyed by name. The values of the dict are the results of calling a metric function, such as `Tensor`.
- `output_alternatives`: a dict of `{submodel_name: (problem_type, {tensor_name: Tensor})}`, where `submodel_name` is a submodel identifier that should be consistent across the pipeline (here likely taken from the name of each `Head`, for models that use them), `problem_type` is a `ProblemType`, `tensor_name` is a symbolic name for an output Tensor possibly but not necessarily taken from `PredictionKey`, and `Tensor` is the corresponding

output Tensor itself.

- `training_chief_hooks` : A list of `SessionRunHook` objects that will be run on the chief worker during training.
- `training_hooks` : A list of `SessionRunHook` objects that will be run on all workers during training.
- `scaffold` : A `tf.train.Scaffold` object that can be used to set initialization, saver, and more to be used in training.

Returns:

A validated `ModelFnOps` object.

Raises:

- `ValueError` : If validation fails.

`estimator_spec`

```
estimator_spec(default_serving_output_alternative_key=None)
```

Creates an equivalent `EstimatorSpec` .

Args:

- `default_serving_output_alternative_key` : Required for multiple heads. If you have multiple entries in `output_alternatives` dict (comparable to multiple heads), `EstimatorSpec` requires a default head that will be used if a Servo request does not explicitly mention which head to infer on. Pass the key of the output alternative here that you want to designate as default. A separate `ExportOutput` for this default head will be added to the `export_outputs` dict with the special key `signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY`, unless there is already an entry in `output_alternatives` with this special key.

Returns:

Instance of `EstimatorSpec` that is equivalent to this `ModelFnOps`

Raises:

- `ValueError` : If problem type is unknown.

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](#)