

tf.train.Scaffold

Contents

Class Scaffold

Properties

`init_feed_dict`

`init_fn`

Class **Scaffold**

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

See the guide: [Training > Distributed execution](#)

Structure to create or gather pieces commonly needed to train a model.

When you build a model for training you usually need ops to initialize variables, a **Saver** to checkpoint them, an op to collect summaries for the visualizer, and so on.

Various libraries built on top of the core TensorFlow library take care of creating some or all of these pieces and storing them in well known collections in the graph. The **Scaffold** class helps pick these pieces from the graph collections, creating and adding them to the collections if needed.

If you call the scaffold constructor without any arguments, it will pick pieces from the collections, creating default ones if needed when `scaffold.finalize()` is called. You can pass arguments to the constructor to provide your own pieces. Pieces that you pass to the constructor are not added to the graph collections.

The following pieces are directly accessible as attributes of the **Scaffold** object:

- **saver** : A `tf.train.Saver` object taking care of saving the variables. Picked from and stored into the **SAVERS** collection in the graph by default.
- **init_op** : An op to run to initialize the variables. Picked from and stored into the **INIT_OP** collection in the graph by default.
- **ready_op** : An op to verify that the variables are initialized. Picked from and stored into the **READY_OP** collection in the graph by default.
- **ready_for_local_init_op** : An op to verify that global state has been initialized and it is alright to run **local_init_op**. Picked from and stored into the **READY_FOR_LOCAL_INIT_OP** collection in the graph by default. This is needed when the initialization of local variables depends on the values of global variables.
- **local_init_op** : An op to initialize the local variables. Picked from and stored into the **LOCAL_INIT_OP** collection in the graph by default.
- **summary_op** : An op to run and merge the summaries in the graph. Picked from and stored into the **SUMMARY_OP** collection in the graph by default.
- **global_step** : A tensor containing the global step counter. Picked from and stored into the **GLOBAL_STEP** collection in the graph by default.

You can also pass the following additional pieces to the constructor:

- `init_feed_dict` : A session feed dictionary that should be used when running the init op.
- `init_fn` : A callable to run after the init op to perform additional initializations. The callable will be called as `init_fn(scaffold, session)` .

Properties

`init_feed_dict`

`init_fn`

`init_op`

`local_init_op`

`ready_for_local_init_op`

`ready_op`

`saver`

`summary_op`

Methods

`__init__`

```
__init__(
    init_op=None,
    init_feed_dict=None,
    init_fn=None,
    ready_op=None,
    ready_for_local_init_op=None,
    local_init_op=None,
    summary_op=None,
    saver=None,
    copy_from_scaffold=None
)
```

Create a scaffold.

Args:

- `init_op` : Optional op for initializing variables.
- `init_feed_dict` : Optional session feed dictionary to use when running the `init_op`.
- `init_fn` : Optional function to use to initialize the model after running the `init_op`. Will be called as `init_fn(scaffold, session)` .
- `ready_op` : Optional op to verify that the variables are initialized. Must return an empty 1D string tensor when the variables are initialized, or a non-empty 1D string tensor listing the names of the non-initialized variables.
- `ready_for_local_init_op` : Optional op to verify that the global variables are initialized and `local_init_op` can be run. Must return an empty 1D string tensor when the global variables are initialized, or a non-empty 1D string tensor listing the names of the non-initialized global variables.
- `local_init_op` : Optional op to initialize local variables.

- `summary_op` : Optional op to gather all summaries. Must return a scalar string tensor containing a serialized `Summary` proto.
- `saver` : Optional `tf.train.Saver` object to use to save and restore variables.
- `copy_from_scaffold` : Optional scaffold object to copy fields from. Its fields will be overwritten by the provided fields in this function.

finalize

```
finalize()
```

Creates operations if needed and finalizes the graph.

get_or_default

```
@staticmethod
get_or_default(
    arg_name,
    collection_key,
    default_constructor
)
```

Get from cache or create a default operation.

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](#)