

tfdbg.LocalCLIDebugHook

Contents

Class LocalCLIDebugHook

Methods

`__init__`

`add_tensor_filter`

Class LocalCLIDebugHook

Inherits From: [SessionRunHook](#)

Defined in [tensorflow/python/debug/wrappers/hooks.py](#).

See the guide: [TensorFlow Debugger > Session wrapper class and SessionRunHook implementations](#)

Command-line-interface debugger hook.

Can be used as a monitor/hook for `tf.train.MonitoredSession` s and `tf.contrib.learn` 's `Estimator` s and `Experiment` s.

Methods

`__init__`

```
__init__(
    ui_type='curses',
    dump_root=None,
    thread_name_filter=None
)
```

Create a local debugger command-line interface (CLI) hook.

Args:

- `ui_type` : (str) user-interface type.
- `dump_root` : (str) optional path to the dump root directory. Must be a directory that does not exist or an empty directory. If the directory does not exist, it will be created by the debugger core during debug `run()` calls and removed afterwards.
- `thread_name_filter` : Regular-expression white list for threads on which the wrapper session will be active. See doc of `BaseDebugWrapperSession` for more details.

`add_tensor_filter`

```
add_tensor_filter(  
    filter_name,  
    tensor_filter  
)
```

Add a tensor filter.

See doc of `LocalCLIDebugWrapperSession.add_tensor_filter()` for details. Override default behavior to accommodate the possibility of this method being called prior to the initialization of the underlying `LocalCLIDebugWrapperSession` object.

Args:

- `filter_name`: See doc of `LocalCLIDebugWrapperSession.add_tensor_filter()` for details.
- `tensor_filter`: See doc of `LocalCLIDebugWrapperSession.add_tensor_filter()` for details.

after_create_session

```
after_create_session(  
    session,  
    coord  
)
```

Called when new TensorFlow session is created.

This is called to signal the hooks that a new session has been created. This has two essential differences with the situation in which `begin` is called:

- When this is called, the graph is finalized and ops can no longer be added to the graph.
- This method will also be called as a result of recovering a wrapped session, not only at the beginning of the overall session.

Args:

- `session`: A TensorFlow Session that has been created.
- `coord`: A Coordinator object which keeps track of all threads.

after_run

```
after_run(  
    run_context,  
    run_values  
)
```

before_run

```
before_run(run_context)
```

begin

```
begin()
```

end

```
end(session)
```

Called at the end of session.

The `session` argument can be used in case the hook wants to run final ops, such as saving a last checkpoint.

If `session.run()` raises exception other than `OutOfRangeError` or `StopIteration` then `end()` is not called. Note the difference between `end()` and `after_run()` behavior when `session.run()` raises `OutOfRangeError` or `StopIteration`. In that case `end()` is called but `after_run()` is not called.

Args:

- `session`: A TensorFlow Session that will be soon closed.

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](#)