tfdbg.GradientsDebugger

Contents

Class **`GradientsDebugger`**

Defined in `tensorflow/python/debug/lib/debug_gradients.py`.

Gradients Debugger.

Allows retrieval of gradient tensors created by TensorFlow's automatic differentiation algorithm, i.e., `tf.gradients` and optimizer classes that use it.

Properties

`graph`

`y_tensor`

Methods

`__init__`

```
__init__(y_tensor=None)
```

Constructor of GradientsDebugger.

Args:

- `y_tensor`: optional: the `tf.Tensor` to be differentiated, i.e., the tensor on the numerator of the differentiation.

`__enter__`

```
__enter__()
```

`__exit__`

```
__exit__(
    unused_type,
    unused_value,
    unused_traceback
)
```

`gradient_tensor`

```
gradient_tensor(x_tensor)
```

Get the gradient tensor of an x-tensor.

Args:

- `x_tensor` : (**`tf.Tensor`** , **`tf.Variable`** or **`str`**) The x-tensor object or its name. x-tensor refers to the independent **`tf.Tensor`** , i.e., the tensor on the denominator of the differentiation.

Returns:

If found, the gradient tensor.

Raises:

- `TypeError` : If **`x_tensor`** is not a **`tf.Tensor`** , **`tf.Variable`** or **`str`** .
- `LookupError` : If the **`x_tensor`** has not been registered with a gradient tensor.

`gradient_tensors`

```
gradient_tensors()
```

Get the gradient tensors that this object is aware of.

Returns:

A dict mapping x-tensor names to gradient tensor objects. x-tensor refers to the tensors on the denominator of the differentation.

`identify_gradient`

```
identify_gradient(input_tensor)
```

Create a debug identity tensor that registers and forwards gradients.

The side effect of this method is that when gradient tensor(s) are created with respect to the any paths that include the **`input_tensor`** , the gradient tensor(s) with repsect to **`input_tensor`** will be registered with this this **`GradientsDebugger`** instance and can later be retrieved, with the methods **`gradient_tensor`** and **`gradient_tensors`** .

Example:

```
x = tf.Variable(1.0)
y = tf.add(x, x)

grad_debugger = tf_debug.GradientsDebugger()
debug_y = grad_debugger.identify_gradient(y)
z = tf.square(debug_y)

# Create a train op under the grad_debugger context.
with grad_debugger:
  train_op = tf.train.GradientDescentOptimizer(z)

# Now we can reflect through grad_debugger to get the gradient tensor
# with respect to y.
y_grad = grad_debugger.gradient_tensor(y)
```

Args:

- `input_tensor`: the input `tf.Tensor` object whose related gradient tensors are to be reigstered with this `GradientsDebugger` instance when they are created, e.g., during `tf.gradients` calls or the construction of optimization (training) op that uses `tf.gradients`.

Returns:

A forwarded identity of `input_tensor`, as a `tf.Tensor`.

Raises:

- `ValueError`: If an op with name that duplicates the gradient-debugging op already exists in the graph (highly unlikely).

`register_gradient_tensor`

```
register_gradient_tensor(
    x_tensor_name,
    gradient_tensor
)
```

Register the gradient tensor for an x-tensor.

Args:

- `x_tensor_name`: (`str`) the name of the independent `tf.Tensor`, i.e., the tensor on the denominator of the differentiation.
- `gradient_tensor`: the gradient `tf.Tensor`.

`watch_gradients_by_tensor_names`

```
watch_gradients_by_tensor_names(
    graph,
    tensor_name_regex
)
```

Watch gradient tensors by name(s) of the x-tensor(s).

The side effect of this method is that when gradient tensor(s) are created with respect to the x-tensors, the gradient tensor(s) will be registered with this `GradientsDebugger` instance and can later be retrieved.

Unlike the `identify_gradient` method, this method is used after the construction of the forward graph has completed. Unlike the `watch_gradients_by_tensor` method, this method does not use handles to the tensors of interest; it uses their names.

This method is the same as `watch_gradients_by_tensors` except that the x-tensors are specified by name patterns, instead of `tf.Tensor` or `tf.Variable` objects.

Example:

```
x = tf.Variable(1.0, name="x")
y = tf.add(x, x, name="y")
z = tf.square(debug_y)

# Create a train op under the grad_debugger context.
grad_debugger = tf_debug.GradientsDebugger()
with grad_debugger.watch_gradients_by_tensor_names(r"(x|y):0$"):
  train_op = tf.train.GradientDescentOptimizer(z)

# Now we can reflect through grad_debugger to get the gradient tensor
# with respect to x and y.
x_grad = grad_debugger.gradient_tensor("x:0")
y_grad = grad_debugger.gradient_tensor("y:0")
```

Args:

- `graph`: the `tf.Graph` to watch the gradients on.
- `tensor_name_regex`: the regular-expression pattern of the name(s) of the x-tensor(s) to watch. x-tensor refers to the tensors on the denominator of the differentiation.

Returns:

The GradientsDebugger instance itself.

watch_gradients_by_tensors

```
watch_gradients_by_tensors(
    graph,
    tensors
)
```

Watch gradient tensors by x-tensor(s).

The side effect of this method is that when gradient tensor(s) are created with respect to the any paths that include the `x_tensor`s, the gradient tensor(s) with repsect to the tensor will be registered with this this `GradientsDebugger` instance and can later be retrieved, with the methods `gradient_tensor` and `gradient_tensors`.

Unlike the method `identify_gradient`, this method is used to retrieve gradient tensors after the construction of the forward subgraph has completed (but before the construction of the backward subgraph).

This method is the same as `watch_gradients_by_x_tensor_names` except that the tensors are specified by the Python `tf.Tensor` or `tf.Variable` objects, instead by name patterns.

Example:

```
x = tf.Variable(1.0)
y = tf.add(x, x, name="y")
z = tf.square(debug_y)

# Create a train op under the grad_debugger context.
grad_debugger = tf_debug.GradientsDebugger()
with grad_debugger.watch_gradients_by_tensors(y):
  train_op = tf.train.GradientDescentOptimizer(z)

# Now we can reflect through grad_debugger to get the gradient tensor
# with respect to y.
y_grad = grad_debugger.gradient_tensor(y)
# or
y_grad = grad_debugger.gradient_tensor("y:0")
```

Args:

- `graph` : the `tf.Graph` to watch the gradients on.
- `tensors` : a `tf.Tensor` or `tf.Variable` object, or a list of such objects.

Returns:

The GradientsDebugger instance itself.