tf.ConditionalAccumulator

Class **`ConditionalAccumulator`**

Inherits From: `ConditionalAccumulatorBase`

Defined in `tensorflow/python/ops/data_flow_ops.py`.

See the guide: Inputs and Readers > Conditional Accumulators

A conditional accumulator for aggregating gradients.

Up-to-date gradients (i.e., time step at which gradient was computed is equal to the accumulator's time step) are added to the accumulator.

Extraction of the average gradient is blocked until the required number of gradients has been accumulated.

Properties

`accumulator_ref`

The underlying accumulator reference.

`dtype`

The datatype of the gradients accumulated by this accumulator.

`name`

The name of the underlying accumulator.

Methods

`__init__`

```
__init__(
    dtype,
    shape=None,
    shared_name=None,
    name='conditional_accumulator'
)
```

Creates a new ConditionalAccumulator.

Args:

- `dtype` : Datatype of the accumulated gradients.
- `shape` : Shape of the accumulated gradients.
- `shared_name` : Optional. If non-empty, this accumulator will be shared under the given name across multiple sessions.
- `name` : Optional name for the accumulator.

`apply_grad`

```
apply_grad(
    grad,
    local_step=0,
    name=None
)
```

Attempts to apply a gradient to the accumulator.

The attempt is silently dropped if the gradient is stale, i.e., local_step is less than the accumulator's global time step.

Args:

- `grad` : The gradient tensor to be applied.
- `local_step` : Time step at which the gradient was computed.
- `name` : Optional name for the operation.

Returns:

The operation that (conditionally) applies a gradient to the accumulator.

Raises:

- `ValueError` : If grad is of the wrong shape

`num_accumulated`

```
num_accumulated(name=None)
```

Number of gradients that have currently been aggregated in accumulator.

Args:

- `name` : Optional name for the operation.

Returns:

Number of accumulated gradients currently in accumulator.

`set_global_step`

```
set_global_step(
    new_global_step,
    name=None
)
```

Sets the global time step of the accumulator.

The operation logs a warning if we attempt to set to a time step that is lower than the accumulator's own time step.

Args:

- `new_global_step` : Value of new time step. Can be a variable or a constant
- `name` : Optional name for the operation.

Returns:

Operation that sets the accumulator's time step.

`take_grad`

```
take_grad(
    num_required,
    name=None
)
```

Attempts to extract the average gradient from the accumulator.

The operation blocks until sufficient number of gradients have been successfully applied to the accumulator.

Once successful, the following actions are also triggered:

- Counter of accumulated gradients is reset to 0.
- Aggregated gradient is reset to 0 tensor.
- Accumulator's internal time step is incremented by 1.

Args:

- `num_required` : Number of gradients that needs to have been aggregated
- `name` : Optional name for the operation

Returns:

A tensor holding the value of the average gradient.

Raises:

- `InvalidArgumentError` : If num_required < 1

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**