tf.contrib.opt.DropStaleGradientOptimizer

Contents

Class DropStaleGradientOptimizer
Methods
__init__
apply_gradients

Class `DropStaleGradientOptimizer`

Inherits From: `Optimizer`

Defined in `tensorflow/contrib/opt/python/training/drop_stale_gradient_optimizer.py`.

Wrapper optimizer that checks and drops stale gradient.

This optimizer records the global step for each worker before computing gradients and compares it with the global step at the time of applying the gradients. If the difference is larger than a threshold, it will drop all the computed gradients.

Methods

`__init__`

```
__init__(
    opt,
    staleness,
    use_locking=False,
    name='DropStaleGradient'
)
```

Constructs a new DropStaleGradientOptimizer.

Args:

- `opt`: The actual optimizer that will be used to compute and apply the gradients. Must be one of the Optimizer classes.
- `staleness`: The maximum staleness allowed for the optimizer.
- `use_locking`: If `True` use locks for clip update operations.
- `name`: Optional name prefix for the operations created when applying gradients. Defaults to "DropStaleGradient".

`apply_gradients`

```
apply_gradients(
    grads_and_vars,
    global_step=None,
    name=None
)
```

compute_gradients

```
compute_gradients(
    loss,
    *args,
    **kwargs
)
```

get_name

```
get_name()
```

get_slot

```
get_slot(
    *args,
    **kwargs
)
```

get_slot_names

```
get_slot_names(
    *args,
    **kwargs
)
```

minimize

```
minimize(
    loss,
    global_step=None,
    var_list=None,
    gate_gradients=GATE_OP,
    aggregation_method=None,
    colocate_gradients_with_ops=False,
    name=None,
    grad_loss=None
)
```

Add operations to minimize `loss` by updating `var_list`.

This method simply combines calls `compute_gradients()` and `apply_gradients()`. If you want to process the gradient before applying them call `compute_gradients()` and `apply_gradients()` explicitly instead of using this function.

Args:

- `loss`: A `Tensor` containing the value to minimize.

- `global_step`: Optional `Variable` to increment by one after the variables have been updated.
- `var_list`: Optional list or tuple of `Variable` objects to update to minimize `loss`. Defaults to the list of variables collected in the graph under the key `GraphKeys.TRAINABLE_VARIABLES`.
- `gate_gradients`: How to gate the computation of gradients. Can be `GATE_NONE`, `GATE_OP`, or `GATE_GRAPH`.
- `aggregation_method`: Specifies the method used to combine gradient terms. Valid values are defined in the class `AggregationMethod`.
- `colocate_gradients_with_ops`: If True, try colocating gradients with the corresponding op.
- `name`: Optional name for the returned operation.
- `grad_loss`: Optional. A `Tensor` holding the gradient computed for `loss`.

Returns:

An Operation that updates the variables in `var_list`. If `global_step` was not `None`, that operation also increments `global_step`.

Raises:

- `ValueError`: If some of the variables are not `Variable` objects.

Class Members

GATE_GRAPH

GATE_NONE

GATE_OP