tf.contrib.training.GreedyLoadBalancingStrategy

Contents

Class `GreedyLoadBalancingStrategy`

Defined in `tensorflow/contrib/training/python/training/device_setter.py`.

Returns the least-loaded ps task for op placement.

The load is calculated by a user-specified load function passed in at construction. There are no units for load, and the load function is responsible for providing an internally consistent measure.

Note that this strategy is very sensitive to the exact order in which ps ops (typically variables) are created, as it greedily places ops on the least-loaded ps at the point each op is processed.

One reasonable heuristic is the `byte_size_load_fn`, which estimates load as the number of bytes that would be used to store and transmit the entire variable. More advanced load functions could consider the difference in access patterns across ops, or trade off CPU-intensive ops with RAM-intensive ops with network bandwidth.

This class is intended to be used as a `ps_strategy` in `tf.train.replica_device_setter`.

Methods

`__init__`

```
__init__(
    num_tasks,
    load_fn
)
```

Create a new `LoadBalancingStrategy`.

Args:

- `num_tasks`: Number of ps tasks to cycle among.
- `load_fn`: A callable that takes an `Operation` and returns a numeric load value for that op.

`__call__`

```
__call__(op)
```

Choose a ps task index for the given `Operation`.

Args:

- `op` : A `Operation` to be placed on ps.

Returns:

The next ps task index to use for the `Operation`. Greedily places the op on the least-loaded ps task so far, as determined by the load function.
