tf.contrib.kernel_methods.sparse_multiclass_hinge_loss

```
sparse_multiclass_hinge_loss(
    labels,
    logits,
    weights=1.0,
    scope=None,
    loss_collection=tf.GraphKeys.LOSSES,
    reduction=losses.Reduction.SUM_BY_NONZERO_WEIGHTS
)
```

Defined in `tensorflow/contrib/kernel_methods/python/losses.py`.

Adds Ops for computing the multiclass hinge loss.

The implementation is based on the following paper: On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines by Crammer and Singer. link: http://jmlr.csail.mit.edu/papers/volume2/crammer01a/crammer01a.pdf

This is a generalization of standard (binary) hinge loss. For a given instance with correct label c*, the loss is given by: loss = max_{c != c*} logits_c - logits_{c*} + 1. or equivalently loss = max_c { logits_c - logits_{c*} + I_{c != c*} } where I_{c != c*} = 1 if c != c* and 0 otherwise.

Args:

- `labels`: `Tensor` of shape [batch_size] or [batch_size, 1]. Corresponds to the ground truth. Each entry must be an index in `[0, num_classes)`.
- `logits`: `Tensor` of shape [batch_size, num_classes] corresponding to the unscaled logits. Its dtype should be either `float32` or `float64`.
- `weights`: Optional (python) scalar or `Tensor`. If a non-scalar `Tensor`, its rank should be either 1 ([batch_size]) or 2 ([batch_size, 1]).
- `scope`: The scope for the operations performed in computing the loss.
- `loss_collection`: collection to which the loss will be added.
- `reduction`: Type of reduction to apply to loss.

Returns:

Weighted loss float `Tensor`. If `reduction` is `NONE`, this has the same shape as `labels`; otherwise, it is a scalar.

Raises:

- `ValueError`: If `logits`, `labels` or `weights` have invalid or inconsistent shapes.
- `ValueError`: If `labels` tensor has invalid dtype.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**