TensorFlow   API r1.4

tf.contrib.metrics.auc_using_histogram

```
auc_using_histogram(
    boolean_labels,
    scores,
    score_range,
    nbins=100,
    collections=None,
    check_shape=True,
    name=None
)
```

Defined in `tensorflow/contrib/metrics/python/ops/histogram_ops.py`.

See the guide: Metrics (contrib) > Metric `Ops`

AUC computed by maintaining histograms.

Rather than computing AUC directly, this Op maintains Variables containing histograms of the scores associated with `True` and `False` labels. By comparing these the AUC is generated, with some discretization error. See: "Efficient AUC Learning Curve Calculation" by Bouckaert.

This AUC Op updates in `O(batch_size + nbins)` time and works well even with large class imbalance. The accuracy is limited by discretization error due to finite number of bins. If scores are concentrated in a fewer bins, accuracy is lower. If this is a concern, we recommend trying different numbers of bins and comparing results.

Args:

- `boolean_labels` : 1-D boolean `Tensor` . Entry is `True` if the corresponding record is in class.
- `scores` : 1-D numeric `Tensor` , same shape as boolean_labels.
- `score_range` : `Tensor` of shape `[2]` , same dtype as `scores` . The min/max values of score that we expect. Scores outside range will be clipped.
- `nbins` : Integer number of bins to use. Accuracy strictly increases as the number of bins increases.
- `collections` : List of graph collections keys. Internal histogram Variables are added to these collections. Defaults to `[GraphKeys.LOCAL_VARIABLES]` .
- `check_shape` : Boolean. If `True` , do a runtime shape check on the scores and labels.
- `name` : A name for this Op. Defaults to "auc_using_histogram".

Returns:

- `auc` : `float32` scalar `Tensor` . Fetching this converts internal histograms to auc value.
- `update_op` : `Op` , when run, updates internal histograms.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**