tf.metrics.auc

```
auc(
    labels,
    predictions,
    weights=None,
    num_thresholds=200,
    metrics_collections=None,
    updates_collections=None,
    curve='ROC',
    name=None,
    summation_method='trapezoidal'
)
```

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

Computes the approximate AUC via a Riemann sum.

The `auc` function creates four local variables, `true_positives`, `true_negatives`, `false_positives` and `false_negatives` that are used to compute the AUC. To discretize the AUC curve, a linearly spaced set of thresholds is used to compute pairs of recall and precision values. The area under the ROC-curve is therefore computed using the height of the recall values by the false positive rate, while the area under the PR-curve is the computed using the height of the precision values by the recall.

This value is ultimately returned as `auc`, an idempotent operation that computes the area under a discretized curve of precision versus recall values (computed using the aforementioned variables). The `num_thresholds` variable controls the degree of discretization with larger numbers of thresholds more closely approximating the true AUC. The quality of the approximation may vary dramatically depending on `num_thresholds`.

For best results, `predictions` should be distributed approximately uniformly in the range [0, 1] and not peaked around 0 or 1. The quality of the AUC approximation may be poor if this is not the case. Setting `summation_method` to 'minoring' or 'majoring' can help quantify the error in the approximation by providing lower or upper bound estimate of the AUC.

For estimation of the metric over a stream of data, the function creates an `update_op` operation that updates these variables and returns the `auc`.

If `weights` is `None`, weights default to 1. Use weights of 0 to mask values.

Args:

- `labels`: A `Tensor` whose shape matches `predictions`. Will be cast to `bool`.
- `predictions`: A floating point `Tensor` of arbitrary shape and whose values are in the range `[0, 1]`.
- `weights`: Optional `Tensor` whose rank is either 0, or the same rank as `labels`, and must be broadcastable to `labels` (i.e., all dimensions must be either `1`, or the same as the corresponding `labels` dimension).
- `num_thresholds`: The number of thresholds to use when discretizing the roc curve.
- `metrics_collections`: An optional list of collections that `auc` should be added to.
- `updates_collections`: An optional list of collections that `update_op` should be added to.
- `curve`: Specifies the name of the curve to be computed, 'ROC' [default] or 'PR' for the Precision-Recall-curve.
- `name`: An optional variable_scope name.

- `summation_method`: Specifies the Riemann summation method used, 'trapezoidal' [default] that applies the trapezoidal rule, 'minoring' that applies left summation for increasing intervals and right summation for decreasing intervals or 'majoring' that applies the opposite.

Returns:

- `auc`: A scalar `Tensor` representing the current area-under-curve.
- `update_op`: An operation that increments the `true_positives`, `true_negatives`, `false_positives` and `false_negatives` variables appropriately and whose value matches `auc`.

Raises:

- `ValueError`: If `predictions` and `labels` have mismatched shapes, or if `weights` is not `None` and its shape doesn't match `predictions`, or if either `metrics_collections` or `updates_collections` are not a list or tuple.

Last updated November 2, 2017.