tf.metrics.mean_cosine_distance

```
mean_cosine_distance(
    labels,
    predictions,
    dim,
    weights=None,
    metrics_collections=None,
    updates_collections=None,
    name=None
)
```

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

Computes the cosine distance between the labels and predictions.

The `mean_cosine_distance` function creates two local variables, `total` and `count` that are used to compute the average cosine distance between `predictions` and `labels`. This average is weighted by `weights`, and it is ultimately returned as `mean_distance`, which is an idempotent operation that simply divides `total` by `count`.

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

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

Args:

- `labels`: A `Tensor` of arbitrary shape.
- `predictions`: A `Tensor` of the same shape as `labels`.
- `dim`: The dimension along which the cosine distance is computed.
- `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). Also, dimension `dim` must be `1`.
- `metrics_collections`: An optional list of collections that the metric value variable should be added to.
- `updates_collections`: An optional list of collections that the metric update ops should be added to.
- `name`: An optional variable_scope name.

Returns:

- `mean_distance`: A `Tensor` representing the current mean, the value of `total` divided by `count`.
- `update_op`: An operation that increments the `total` and `count` variables appropriately.

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.