tf.losses.huber_loss

```
huber_loss(
    labels,
    predictions,
    weights=1.0,
    delta=1.0,
    scope=None,
    loss_collection=tf.GraphKeys.LOSSES,
    reduction=Reduction.SUM_BY_NONZERO_WEIGHTS
)
```

Defined in `tensorflow/python/ops/losses/losses_impl.py`.

Adds a Huber Loss term to the training procedure.

For each value x in `error=labels-predictions`, the following is calculated:

```
  0.5 * x^2                  if |x| <= d
  0.5 * d^2 + d * (|x| - d)  if |x| > d
```

where d is `delta`.

See: https://en.wikipedia.org/wiki/Huber_loss

`weights` acts as a coefficient for the loss. If a scalar is provided, then the loss is simply scaled by the given value. If `weights` is a tensor of size [batch_size], then the total loss for each sample of the batch is rescaled by the corresponding element in the `weights` vector. If the shape of `weights` matches the shape of `predictions`, then the loss of each measurable element of `predictions` is scaled by the corresponding value of `weights`.

Args:

- `labels`: The ground truth output tensor, same dimensions as 'predictions'.
- `predictions`: The predicted outputs.
- `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 `losses` dimension).
- `delta`: `float`, the point where the huber loss function changes from a quadratic to linear.
- `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 scalar.

Raises:

- `ValueError`: If the shape of `predictions` doesn't match that of `labels` or if the shape of `weights` is invalid. Also

if `labels` or `predictions` is None.

Last updated November 2, 2017.