tf.losses.sparse_softmax_cross_entropy

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

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

Cross-entropy loss using `tf.nn.sparse_softmax_cross_entropy_with_logits`.

`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 shape [`batch_size`], then the loss weights apply to each corresponding sample.

Args:

- `labels`: `Tensor` of shape `[d_0, d_1, ..., d_{r-1}]` (where `r` is rank of `labels` and result) and dtype `int32` or `int64`. Each entry in `labels` must be an index in `[0, num_classes)`. Other values will raise an exception when this op is run on CPU, and return `NaN` for corresponding loss and gradient rows on GPU.
- `logits`: Unscaled log probabilities of shape `[d_0, d_1, ..., d_{r-1}, num_classes]` and dtype `float32` or `float64`.
- `weights`: Coefficients for the loss. This must be scalar or broadcastable to `labels` (i.e. same rank and each dimension is either 1 or the same).
- `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 `Tensor` of the same type as `logits`. If `reduction` is `NONE`, this has the same shape as `labels`; otherwise, it is scalar.

Raises:

- `ValueError`: If the shapes of `logits`, `labels`, and `weights` are incompatible, or if any of them are None.

Last updated November 2, 2017.