

tf.losses.softmax_cross_entropy

```
softmax_cross_entropy(  
    onehot_labels,  
    logits,  
    weights=1.0,  
    label_smoothing=0,  
    scope=None,  
    loss_collection=tf.GraphKeys.LOSSES,  
    reduction=Reduction.SUM_BY_NONZERO_WEIGHTS  
)
```

Defined in [tensorflow/python/ops/losses/losses_impl.py](#).

Creates a cross-entropy loss using `tf.nn.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.

If **label_smoothing** is nonzero, smooth the labels towards $1/\text{num_classes}$: $\text{new_onehot_labels} = \text{onehot_labels} * (1 - \text{label_smoothing}) + \text{label_smoothing} / \text{num_classes}$

Args:

- onehot_labels**: `[batch_size, num_classes]` target one-hot-encoded labels.
- logits**: `[batch_size, num_classes]` logits outputs of the network.
- weights**: Optional **Tensor** whose rank is either 0, or rank 1 and is broadcastable to the loss which is a **Tensor** of shape `[batch_size]`.
- label_smoothing**: If greater than 0 then smooth the labels.
- 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 shape `[batch_size]`; otherwise, it is scalar.

Raises:

- ValueError**: If the shape of **logits** doesn't match that of **onehot_labels** or if the shape of **weights** is invalid or if **weights** is None. Also if **onehot_labels** or **logits** is None.

Stay Connected

- Blog
- GitHub
- Twitter

Support

- Issue Tracker
- Release Notes
- Stack Overflow