See playlist (https://www.youtube.com/playlist?list=PLQY2H8rRoyvxtOJfAzfs0Cq5FqBuYO8e8)

tf.keras.losses.SparseCategoricalCrossentropy

TensorFlow 1 version (/versions/r1.15/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)

View source on GitHub (https://github.c L759)

Computes the crossentropy loss between the labels and predictions.

Inherits From: `Loss` (https://www.tensorflow.org/api_docs/python/tf/keras/losses/Loss)

View aliases

Main aliases

`tf.losses.SparseCategoricalCrossentropy` (https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)

Compat aliases for migration

See Migration guide (https://www.tensorflow.org/guide/migrate) for more details.

`tf.compat.v1.keras.losses.SparseCategoricalCrossentropy` (https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)

```
tf.keras.losses.SparseCategoricalCrossentropy(
    from_logits=False, reduction=losses_utils.ReductionV2.AUTO,
    name='sparse_categorical_crossentropy'
)
```

Used in the notebooks

Used in the guide	Used in the tutorials

Used in the guide	Used in the tutorials
- Migrate your TensorFlow 1 code to TensorFlow 2 (https://www.tensorflow.org/guide/migrate) - Training and evaluation with the built-in methods (https://www.tensorflow.org/guide/keras/train_and_evaluate) - Recurrent Neural Networks (RNN) with Keras (https://www.tensorflow.org/guide/keras/rnn) - Writing a training loop from scratch (https://www.tensorflow.org/guide/keras/writing_a_training_loop_from_scratch) - Use TPUs (https://www.tensorflow.org/guide/tpu)	- Load text (https://www.tensorflow.org/tu - Distributed training with Keras (https://www.tensorflow.org/tutorials/di - Image segmentation (https://www.tensorflow.org/tutorials/im - Save and load a model using a distributio (https://www.tensorflow.org/tutorials/di - Image classification (https://www.tensorflow.org/tutorials/im

Use this crossentropy loss function when there are two or more label classes. We expect labels to be provided as integers. If you want to provide labels using `one-hot` representation, please use `CategoricalCrossentropy` loss. There should be `# classes` floating point values per feature for `y_pred` and a single floating point value per feature for `y_true`.

In the snippet below, there is a single floating point value per example for `y_true` and `# classes` floating pointing values per example for `y_pred`. The shape of `y_true` is `[batch_size]` and the shape of `y_pred` is `[batch_size, num_classes]`.

Standalone usage:

```
>>> y_true = [1, 2]
>>> y_pred = [[0.05, 0.95, 0], [0.1, 0.8, 0.1]]
>>> # Using 'auto'/'sum_over_batch_size' reduction type.
>>> scce = tf.keras.losses.SparseCategoricalCrossentropy()
>>> scce(y_true, y_pred).numpy()
1.177
```

```
>>> # Calling with 'sample_weight'.
>>> scce(y_true, y_pred, sample_weight=tf.constant([0.3, 0.7])).numpy()
0.814
```

```
>>> # Using 'sum' reduction type.
>>> scce = tf.keras.losses.SparseCategoricalCrossentropy(
...     reduction=tf.keras.losses.Reduction.SUM)
>>> scce(y_true, y_pred).numpy()
2.354
```

```
>>> # Using 'none' reduction type.
>>> scce = tf.keras.losses.SparseCategoricalCrossentropy(
...     reduction=tf.keras.losses.Reduction.NONE)
>>> scce(y_true, y_pred).numpy()
array([0.0513, 2.303], dtype=float32)
```

Usage with the `compile()` API:

```
model.compile(optimizer='sgd',
              loss=tf.keras.losses.SparseCategoricalCrossentropy())
```

Args	
`from_logits`	Whether `y_pred` is expected to be a logits tensor. By default, we assume that `y_pred` encodes a probability distribution.
`reduction`	(Optional) Type of `tf.keras.losses.Reduction` (https://www.tensorflow.org/api_docs/python/tf/keras/losses/Reduction) to apply to loss. Default value is `AUTO`. `AUTO` indicates that the reduction option will be determined by the usage context. For almost all cases this defaults to `SUM_OVER_BATCH_SIZE`. When used with `tf.distribute.Strategy` (https://www.tensorflow.org/api_docs/python/tf/distribute/Strategy), outside of built-in training loops such as `tf.keras` (https://www.tensorflow.org/api_docs/python/tf/keras) `compile` and `fit`, using `AUTO` or `SUM_OVER_BATCH_SIZE` will raise an error. Please see this custom training tutorial (https://www.tensorflow.org/tutorials/distribute/custom_training) for more details.
`name`	Optional name for the op. Defaults to 'sparse_categorical_crossentropy'.

Methods

`from_config`

View source (https://github.com/tensorflow/tensorflow/blob/v2.5.0/tensorflow/python/keras/losses.py#L159-L169)

```
@classmethod
from_config(
    config
)
```

Instantiates a `Loss` from its config (output of `get_config()`).

Args	
`config`	Output of `get_config()`.

Returns

A `Loss` instance.

`get_config`

View source (https://github.com/tensorflow/tensorflow/blob/v2.5.0/tensorflow/python/keras/losses.py#L261-L266)

```
get_config()
```

Returns the config dictionary for a `Loss` instance.

`__call__`

View source (https://github.com/tensorflow/tensorflow/blob/v2.5.0/tensorflow/python/keras/losses.py#L120-L157)

```
__call__(
    y_true, y_pred, sample_weight=None
)
```

Invokes the `Loss` instance.

Args	
`y_true`	Ground truth values. shape = `[batch_size, d0, .. dN]`, except sparse loss functions such as sparse categorical crossentropy where shape = `[batch_size, d0, .. dN-1]`
`y_pred`	The predicted values. shape = `[batch_size, d0, .. dN]`
`sample_weight`	Optional `sample_weight` acts as a coefficient for the loss. If a scalar is provided, then the loss is simply scaled by the given value. If `sample_weight` 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 `sample_weight` vector. If the shape of `sample_weight` is `[batch_size, d0, .. dN-1]` (or can be broadcasted to this shape), then each loss element of `y_pred` is scaled by the corresponding value of `sample_weight`. (Note on`dN-1`: all loss functions reduce by 1 dimension, usually axis=-1.)

Returns
Weighted loss float `Tensor`. If `reduction` is `NONE`, this has shape `[batch_size, d0, .. dN-1]`; otherwise, it is scalar. (Note `dN-1` because all loss functions reduce by 1 dimension, usually axis=-1.)

Raises	
`ValueError`	If the shape of `sample_weight` is invalid.

Last updated 2021-05-14 UTC.