tf.SparseFeature

Contents

Class `SparseFeature`

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

See the guide: Inputs and Readers > Converting

Configuration for parsing a sparse input feature from an `Example`.

Note, preferably use `VarLenFeature` (possibly in combination with a `SequenceExample`) in order to parse out `SparseTensor`s instead of `SparseFeature` due to its simplicity.

Closely mimicking the `SparseTensor` that will be obtained by parsing an `Example` with a `SparseFeature` config, a `SparseFeature` contains a

- `value_key`: The name of key for a `Feature` in the `Example` whose parsed `Tensor` will be the resulting `SparseTensor.values`.
- `index_key`: A list of names - one for each dimension in the resulting `SparseTensor` whose `indices[i][dim]` indicating the position of the `i`-th value in the `dim` dimension will be equal to the `i`-th value in the Feature with key named `index_key[dim]` in the `Example`.
- `size`: A list of ints for the resulting `SparseTensor.dense_shape`.

For example, we can represent the following 2D `SparseTensor`

```
SparseTensor(indices=[[3, 1], [20, 0]],
             values=[0.5, -1.0]
             dense_shape=[100, 3])
```

with an `Example` input proto

```
features {
  feature { key: "val" value { float_list { value: [ 0.5, -1.0 ] } } }
  feature { key: "ix0" value { int64_list { value: [ 3, 20 ] } } }
  feature { key: "ix1" value { int64_list { value: [ 1, 0 ] } } }
}
```

and `SparseFeature` config with 2 `index_key`s

```
SparseFeature(index_key=["ix0", "ix1"],
              value_key="val",
              dtype=tf.float32,
              size=[100, 3])
```

Fields:

- `index_key` : A single string name or a list of string names of index features. For each key the underlying feature's type must be `int64` and its length must always match that of the `value_key` feature. To represent `SparseTensor` s with a `dense_shape` of `rank` higher than 1 a list of length `rank` should be used.
- `value_key` : Name of value feature. The underlying feature's type must be `dtype` and its length must always match that of all the `index_key` s' features.
- `dtype` : Data type of the `value_key` feature.
- `size` : A Python int or list thereof specifying the dense shape. Should be a list if and only if `index_key` is a list. In that case the list must be equal to the length of `index_key` . Each for each entry `i` all values in the `index_key` [i] feature must be in `[0, size[i])` .
- `already_sorted` : A Python boolean to specify whether the values in `value_key` are already sorted by their index position. If so skip sorting. False by default (optional).

Properties

`already_sorted`

Alias for field number 4

`dtype`

Alias for field number 2

`index_key`

Alias for field number 0

`size`

Alias for field number 3

`value_key`

Alias for field number 1

Methods

`__new__`

```
@staticmethod
__new__(
    cls,
    index_key,
    value_key,
    dtype,
    size,
    already_sorted=False
)
```

Last updated November 2, 2017.