tf.contrib.lookup.HashTable

Contents

Class **HashTable**

Inherits From: `InitializableLookupTableBase`

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

A generic hash table implementation.

Example usage:

```
table = tf.HashTable(
    tf.KeyValueTensorInitializer(keys, values), -1)
out = table.lookup(input_tensor)
table.init.run()
print(out.eval())
```

Properties

`default_value`

The default value of the table.

`init`

The table initialization op.

`key_dtype`

The table key dtype.

`name`

The name of the table.

`table_ref`

Get the underlying table reference.

`value_dtype`

The table value dtype.

Methods

`__init__`

```
__init__(
    initializer,
    default_value,
    shared_name=None,
    name=None
)
```

Creates a non-initialized `HashTable` object.

Creates a table, the type of its keys and values are specified by the initializer. Before using the table you will have to initialize it. After initialization the table will be immutable.

Args:

- `initializer`: The table initializer to use. See `HashTable` kernel for supported key and value types.
- `default_value`: The value to use if a key is missing in the table.
- `shared_name`: If non-empty, this table will be shared under the given name across multiple sessions.
- `name`: A name for the operation (optional).

Returns:

A `HashTable` object.

`lookup`

```
lookup(
    keys,
    name=None
)
```

Looks up `keys` in a table, outputs the corresponding values.

The `default_value` is used for keys not present in the table.

Args:

- `keys`: Keys to look up. May be either a `SparseTensor` or dense `Tensor`.
- `name`: A name for the operation (optional).

Returns:

A `SparseTensor` if keys are sparse, otherwise a dense `Tensor`.

Raises:

- `TypeError`: when `keys` or `default_value` doesn't match the table data types.

`size`

```
size(name=None)
```

Compute the number of elements in this table.

Args:

- `name`: A name for the operation (optional).

Returns:

A scalar tensor containing the number of elements in this table.

Last updated November 2, 2017.