



链滴

# TensorFlow 数据类型转换

作者: [nosqlcoco](#)

原文链接: <https://ld246.com/article/1498556150279>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

TensorFlow 为开发者提供了多种数据类型转换方法，具体如下：

- `tf.string_to_number(string_tensor, out_type=None, name=None)`
- `tf.to_double(x, name='ToDouble')`
- `tf.to_float(x, name='ToFloat')`
- `tf.to_bfloat16(x, name='ToBFloat16')`
- `tf.to_int32(x, name='ToInt32')`
- `tf.to_int64(x, name='ToInt64')`
- `tf.cast(x, dtype, name=None)`

---

## `tf.string_to_number(string_tensor, out_type=None, name=None)`

将输入张量中的每个字符串转换为指定的数值类型张量。

### Args:

- `string_tensor` : 一个数值字符串类型张量
- `out_type` : 参数类型为 `tf.DType`, 可选 `tf.float32`, `tf.int32` 默认为 `tf.float32` (optional)
- `name` : 操作名称 (optional)

### Returns:

一个类型 `out_type` 的张量。

```
import tensorflow as tf
x = ['1','2','7']
with tf.Session() as sess:
    y = tf.string_to_number(x, out_type = tf.int32)
    print(y.eval()) ==> [1 2 7]
```

---

## `tf.to_double(x, name='ToDouble')`

将一个张量转换为 `float64`

### Args:

- `x` : 张量或稀疏张量
- `name` : 操作名称 (optional)

### Returns:

返回一个类型为 `float64` 、大小相同的张量或稀疏张量。

```
import tensorflow as tf
x = [1, 2, 7]
```

```
with tf.Session() as sess:  
    y = tf.to_double(x)  
    print(y.eval()) ==> [ 1.  2.  7.]
```

---

## tf.to\_float(x, name='ToFloat')

将一个张量转换为 `float32`

### Args:

- `x`: 张量或稀疏张量
- `name`: 操作名称 (optional)

### Returns:

返回一个类型为 `float32`、大小相同的张量或稀疏张量。

```
import tensorflow as tf  
x = [1, 2, 7]  
with tf.Session() as sess:  
    y = tf.to_double(x)  
    print(y.eval()) ==> [ 1.  2.  7.]
```

---

## tf.to\_bfloat16(x, name='ToBFloat16')

将一个张量转换为 `bfloat16`，输入必须是浮点型张量。

### Args:

- `x`: 张量或稀疏张量
- `name`: 操作名称 (optional)

### Returns:

返回一个类型为 `bfloat16`、大小相同的张量或稀疏张量。

```
import tensorflow as tf  
x = [1.0, 2.0, 7.0]  
with tf.Session() as sess:  
    y = tf.to_bfloat16(x)  
    print(y.eval()) ==> [16256 16384 16608]
```

---

## tf.to\_int32(x, name='ToInt32')

将一个张量转换为 `int32`。

### Args:

- **x**: 张量或稀疏张量
- **name**: 操作名称 (optional)

**Returns:**

返回一个类型为 **int32**、大小相同的张量或稀疏张量。

```
import tensorflow as tf
x = [1.0, 2, 7]
with tf.Session() as sess:
    y = tf.to_int32(x)
    print(y)
#输出 Tensor("ToInt32_3:0", shape=(3,), dtype=int32)
```

---

## **tf.to\_int64(x, name='ToInt64')**

将一个张量转换为 **int64**。

**Args:**

- **x**: 张量或稀疏张量
- **name**: 操作名称 (optional)

**Returns:**

返回一个类型为 **int64**、大小相同的张量或稀疏张量。

```
import tensorflow as tf
x = [1.0, 2, 7]
with tf.Session() as sess:
    y = tf.to_int64(x)
    print(y.eval())
#输出 Tensor("ToInt64_3:0", shape=(3,), dtype=int64)
```

---

## **tf.cast(x, dtype, name=None)**

将一个张量转换新类型，新类型由 **dtype** 参数指定。

**Args:**

- **x**: 张量或稀疏张量
- **dtype**: 目标类型, **tf.DType**
- **name**: 操作名称 (optional)

**Returns:**

返回一个类型为 **dtype**、大小相同的张量或稀疏张量。

```
import tensorflow as tf
```

```
x = [1.0, 2, 7]
with tf.Session() as sess:
    # 等同于 tf.to_int32(x)
    y = tf.cast(x, tf.int32)
    print(y)
    #输出 Tensor("Cast:0", shape=(3,), dtype=int32)
```