

tensorflow接口研读math_ops(一)

math_ops函数使用，本篇为算术函数和基本数学函数。

1.1 tf.add(x, y, name=None)

功能：对应位置元素的加法运算。

输入：x,y具有相同尺寸的tensor，可以为`half`，`float32`，`float64`，`uint8`，`int8`，`int16`，`int32`，`int64`，`complex64`，`complex128`，`string`类型。

例：

```
x=tf.constant(1.0)
```

```
y=tf.constant(2.0)
```

```
z=tf.add(x,y)
```

```
z==>(3.0)
```

1.2 tf.subtract(x, y, name=None)

功能：对应位置元素的减法运算。

输入：x,y具有相同尺寸的tensor，可以为`half`，`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`，`string`类型。

例：

```
x=tf.constant([[1.0,-1.0]],tf.float64)
```

```
y=tf.constant([[2.2,2.3]],tf.float64)
```

```
z=tf.subtract(x,y)
```

```
z==>[[-1.2,-3.3]]
```

1.3 tf.multiply(x, y, name=None)

功能：对应位置元素的乘法运算。

输入：x,y具有相同尺寸的tensor，可以为`half`，`float32`，`float64`，`uint8`，`int8`，`uint16`，`int16`，`int32`，`int64`，`complex64`，`complex128`，`string`类型。

例：

```
x=tf.constant([[1.0,-1.0]],tf.float64)
```

```
y=tf.constant([[2.2,2.3]],tf.float64)
```

```
z=tf.multiply(x,y)
```

```
z==>[[2.2,-2.3]]
```

1.4 tf.scalar_mul(scalar, x)

功能：固定倍率缩放。

输入：scalar必须为0维元素，x为tensor。

例：

```
scalar=2.2
```

```
x=tf.constant([[1.2,-1.0]],tf.float64)
```

```
z=tf.scalar_mul(scalar,x)
```

```
z==>[[2.64,-2.2]]
```

1.5 tf.div(x, y, name=None)[推荐使用tf.divide(x, y)]

功能：对应位置元素的除法运算（使用python2.7除法算法，如果x,y有一个为浮点数，结果为浮点数；否则为整数，但使用该函数会报错）。

输入: x, y 具有相同尺寸的tensor, x 为被除数, y 为除数。

例:

```
x=tf.constant([[1,4,8]],tf.int32)
y=tf.constant([[2,3,3]],tf.int32)
z=tf.div(x,y)
```

```
z==>[[0,1,2]]
```

```
x=tf.constant([[1,4,8]],tf.int64)
y=tf.constant([[2,3,3]],tf.int64)
z=tf.divide(x,y)
```

```
z==>[[0.5,1.33333333,2.66666667]]
```

```
x=tf.constant([[1,4,8]],tf.float64)
y=tf.constant([[2,3,3]],tf.float64)
z=tf.div(x,y)
```

```
z==>[[0.5,1.33333333,2.66666667]]
```

1.6 tf.truediv(x, y, name=None)

功能: 对应位置元素的除法运算。(使用python3除法算法, 又叫真除, 结果为浮点数, 推荐使用tf.divide)

输入: x, y 具有相同尺寸的tensor, x 为被除数, y 为除数。

1.7 tf.floordiv(x, y, name=None)

功能: 对应位置元素的地板除法运算。返回不大于结果的最大整数

输入: x, y 具有相同尺寸的tensor, x 为被除数, y 为除数。

例:

```
x=tf.constant([[2,4,-1]],tf.int64) #float类型运行结果一致, 只是类型为浮点型
y=tf.constant([[3,3,3]],tf.int64)
z=tf.floordiv(x,y)
```

```
z==>[[0,1,-1]]
```

1.8 tf.realdiv(x, y, name=None)

功能: 对应位置元素的实数除法运算。实际情况不非官方描述, 与divide结果没区别,

输入: x, y 具有相同尺寸的tensor, 可以为`half`, `float32`, `float64`, `uint8`, `int8`, `int16`, `int32`, `int64`, `complex64`, `complex128`, `string`类型。

例:

```
x=tf.constant([[2+1j,4+2j,-1+3j]],tf.complex64)
y=tf.constant([[3+3j,3+1j,3+2j]],tf.complex64)
z=tf.realdiv(x,y)
```

```
z==>[[0.50000000-0.16666667j 1.39999998+0.2j 0.23076922+0.84615386j]]
```

1.9 tf.truncatediv(x, y, name=None)

功能: 对应位置元素的截断除法运算, 获取整数部分。(和手册功能描述不符, 符号位并不能转为0)

输入: x, y 具有相同尺寸的tensor, 可以为`uint8`, `int8`, `int16`, `int32`, `int64`, 类型。(只能为整型, 浮点型等并未注册, 和手册不符)

例:

```
x=tf.constant([[2,4,-7]],tf.int64)
y=tf.constant([[3,3,3]],tf.int64)
```

```
z=tf.truncatediv(x,y)
```

```
z==>[[0 1 -2]]
```

1.10 tf.floor_div(x, y, name=None)

功能：对应位置元素的地板除法运算。（和tf.floordiv运行结果一致，只是内部实现方式不一样）

输入：x,y具有相同尺寸的tensor，可以为`half`，`float32`，`float64`，`uint8`，`int8`，`int16`，`int32`，`int64`，`complex64`，`complex128`，`string`类型。

1.11 tf.truncatemod(x, y, name=None)

功能：对应位置元素的截断除法取余运算。

输入：x,y具有相同尺寸的tensor，可以为float32`，`float64`，`int32`，`int64`类型。

例：

```
x=tf.constant([[2.1,4.1,-1.1]],tf.float64)
```

```
y=tf.constant([[3,3,3]],tf.float64)
```

```
z=tf.truncatemod(x,y)
```

```
z==>[[2.1 1.1 -1.1]]
```

1.12 tf.floormod(x, y, name=None)

功能：对应位置元素的地板除法取余运算。

输入：x,y具有相同尺寸的tensor，可以为float32`，`float64`，`int32`，`int64`类型。

例：

```
x=tf.constant([[2.1,4.1,-1.1]],tf.float64)
```

```
y=tf.constant([[3,3,3]],tf.float64)
```

```
z=tf.floormod(x,y)
```

```
z==>[[2.1 1.1 1.9]]
```

1.13 tf.mod(x, y, name=None)

功能：对应位置元素的除法取余运算。若x和y只有一个小于0，则计算`floor (x/y) *y+mod(x,y)`。

输入：x,y具有相同尺寸的tensor，可以为`float32`，`float64`，`int32`，`int64`类型。

例：

```
x=tf.constant([[2.1,4.1,-1.1]],tf.float64)
```

```
y=tf.constant([[3,3,3]],tf.float64)
```

```
z=tf.mod(x,y)
```

```
z==>[[2.1 1.1 1.9]]
```

1.14 tf.cross(x, y, name=None)

功能：计算叉乘。最大维度为3。

输入：x,y具有相同尺寸的tensor，包含3个元素的向量

例：

```
x=tf.constant([[1,2,-3]],tf.float64)
```

```
y=tf.constant([[2,3,4]],tf.float64)
```

```
z=tf.cross(x,y)
```

```
z==>[[17. -10. -1]]#2×4-(-3)×3=17, -(1×4-(-3)×2)=-10, 1×3-2×2=-1。
```

1.15 tf.add_n(inputs, name=None)

功能：将所有输入的tensor进行对应位置的加法运算

输入：inputs：一组tensor，必须是相同类型和维度。

例：

```
x=tf.constant([[1,2,-3]],tf.float64)
```

```
y=tf.constant([[2,3,4]],tf.float64)
z=tf.constant([[1,4,3]],tf.float64)
xyz=[x,y,z]
z=tf.add_n(xyz)
```

```
z==>[[4. 9. 4.]]
```

1.16 tf.abs(x, name=None)

功能：求x的绝对值。

输入：x为张量或稀疏张量，可以为`float32`，`float64`，`int32`，`int64`类型。

例：

```
x=tf.constant([[1.1,2,-3]],tf.float64)
z=tf.abs(x)
```

```
z==>[[1.1 2. 3.]]
```

1.17 tf.negative(x, name=None)

功能：求x的负数。

输入：x为张量或稀疏张量，可以为`half`，`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[1.1,2,-3]],tf.float64)
z=tf.negative(x)
```

```
z==>[[-1.1. -2. 3.]]
```

1.18 tf.sign(x, name=None)

功能：求x的符号， $x > 0$ ，则 $y = 1$ ； $x < 0$ 则 $y = -1$ ； $x = 0$ 则 $y = 0$ 。

输入：x，为张量，可以为`half`，`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[1.1,0,-3]],tf.float64)
z=tf.sign(x)
```

```
z==>[[1. 0. -1.]]
```

1.19 tf.reciprocal(x, name=None)

功能：求x的倒数。

输入：x为张量，可以为`half`，`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[2,0,-3]],tf.float64)
z=tf.reciprocal(x)
```

```
z==>[[0.5 inf -0.33333333]]
```

1.20 tf.square(x, name=None)

功能：计算x各元素的平方。

输入：x为张量或稀疏张量，可以为`half`，`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[2,0,-3]],tf.float64)
z=tf.square(x)
```

```
z==>[[4. 0. 9.]]
```

1.21 tf.round(x, name=None)

功能：计算x各元素的距离其最近的整数，若在中間，則取偶数值。

输入：x为张量，可以为`float32`，`float64`类型。

例：

```
x=tf.constant([[0.9,1.1,1.5,-4.1,-4.5,-4.9]],tf.float64)
```

```
z=tf.round(x)
```

```
z==>[[1. 1. 2. -4. -4. -5.]]
```

1.22 tf.sqrt(x, name=None)

功能：计算x各元素的平方。

输入：x为张量，可以为`half`，`float32`，`float64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[2,3,-5]],tf.float64)
```

```
z=tf.sqrt(x)
```

```
z==>[[1.41421356 1.73205081 nan]]
```

1.23 tf.rsqrt(x, name=None)

功能：计算x各元素的平方根的倒数。

输入：x为张量或稀疏张量，可以为`half`，`float32`，`float64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[2,3,5]],tf.float64)
```

```
z=tf.rsqrt(x)
```

```
z==>[[0.70710678 0.57735027 0.4472136]]
```

1.24 tf.pow(x, y, name=None)

功能：计算x各元素的y次方。

输入：x, y为张量，可以为`float32`，`float64`，`int32`，`int64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[2,3,5]],tf.float64)
```

```
y=tf.constant([[2,3,4]],tf.float64)
```

```
z=tf.pow(x,y)
```

```
z==>[[4. 27. 625.]]
```

1.25 tf.exp(x, name=None)

功能：计算x各元素的自然指数，即 e^x 。

输入：x为张量，可以为`half`，`float32`，`float64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[0,1,-1]],tf.float64)
```

```
z=tf.exp(x)
```

```
z==>[[1. 2.71828183 0.36787944]]
```

1.26 tf.expml(x, name=None)

功能：计算x各元素的自然指数减1，即 e^x-1 。

输入：x为张量，可以为`half`，`float32`，`float64`，`complex64`，`complex128`类型。

例：

```
x=tf.constant([[0,1,-1]],tf.float64)
```

```
z=tf.expml(x)
```

```
z==>[[0. 1.71828183 -0.63212056]]
```

1.27 tf.log(x, name=None)

功能：计算x各元素的自然对数。

输入：x为张量，可以为`half`,`float32`,`float64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[1,2.71828183,10]],tf.float64)
```

```
z=tf.log(x)
```

```
z==>[[0. 1. 2.30258509]]
```

1.28 tf.log1p(x, name=None)

功能：计算x各元素加1后的自然对数。

输入：x为张量，可以为`half`,`float32`,`float64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[0,1.71828183,9]],tf.float64)
```

```
z=tf.log1p(x)
```

```
z==>[[0. 1. 2.30258509]]
```

1.29 tf.ceil(x, name=None)

功能：计算x各元素比x大的最小整数。

输入：x为张量，可以为`half`,`float32`,`float64`类型。

例：

```
x=tf.constant([[0.2, 0.8, -0.7]],tf.float64)
```

```
z=tf.ceil(x)
```

```
z==>[[1. 1. -0.]]
```

1.30 tf.floor(x, name=None)

功能：计算x各元素比其小的最大整数。

输入：x为张量，可以为`half`,`float32`,`float64`类型。

例：

```
x=tf.constant([[0.2, 0.8, -0.7]],tf.float64)
```

```
z=tf.floor(x)
```

```
z==>[[0. 0. -1.]]
```

1.31 tf.maximum(x, y, name=None)

功能：计算x,y对应位置元素较大的值。

输入：x, y为张量，可以为`half`,`float32`,`float64`,`int32`,`int64`类型。

例：

```
x=tf.constant([[0.2,0.8,-0.7]],tf.float64)
```

```
y=tf.constant([[0.2,0.5,-0.3]],tf.float64)
```

```
z=tf.maximum(x,y)
```

```
z==>[[0.2 0.8 -0.3]]
```

1.32 tf.minimum(x, y, name=None)

功能：计算x,y对应位置元素较小的值。

输入：x, y为张量，可以为`half`,`float32`,`float64`,`int32`,`int64`类型。

例：

```
x=tf.constant([[0.2,0.8,-0.7]],tf.float64)
```

```
y=tf.constant([[0.2,0.5,-0.3]],tf.float64)
```

```
z=tf.maximum(x,y)
```

```
z==>[[0.2 0.5 -0.7]]
```

1.33 tf.cos(x, name=None)

功能：计算x的余弦值。

输入：x为张量，可以为`half`,`float32`,`float64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[0,3.1415926]],tf.float64)
```

```
z=tf.cos(x)
```

```
z==>[[1. -1.]]
```

1.34 tf.sin(x, name=None)

功能：计算x的正弦值。

输入：x为张量，可以为`half`,`float32`,`float64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[0,1.5707963]],tf.float64)
```

```
z=tf.sin(x)
```

```
z==>[[0. 1.]]
```

1.35 tf.lbeta(x, name=None)

功能：计算 $\ln(|\text{Beta}(x)|)$ ，并以最末尺度进行归纳。

最末尺度 $z = [z_0, \dots, z_{K-1}]$ ，则 $\text{Beta}(z) = \prod_j \text{Gamma}(z_j) /$

$\text{Gamma}(\sum_j z_j)$

输入：x为秩为n+1的张量，可以为'float','double'类型。

例：

```
x=tf.constant([[4,3,3],[2,3,2]],tf.float64)
```

```
z=tf.lbeta(x)
```

```
z==>[-9.62377365 -5.88610403]
```

```
#ln(gamma(4)*gamma(3)*gamma(3)/gamma(4+3+3))=ln(6*2*2/362880)=-9.62377365
```

```
#ln(gamma(2)*gamma(3)*gamma(2)/gamma(2+3+2))=ln(2/720)=-5.88610403
```

1.36 tf.tan(x, name=None)

功能：计算tan(x)。

输入：x为张量，可以为`half`,`float32`,`float64`,`int32`,`int64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[0,0.785398163]],tf.float64)
```

```
z=tf.tan(x)
```

```
z==>[[0. 1.]]
```

1.37 tf.acos(x, name=None)

功能：计算acos(x)。

输入：x为张量，可以为`half`,`float32`,`float64`,`int32`,`int64`,`complex64`,`complex128`类型。

例：

```
x=tf.constant([[0,1,-1]],tf.float64)
```

```
z=tf.acos(x)
```

```
z==>[[1.57079633 0. 3.14159265]]
```

1.38 tf.asin(x, name=None)

功能：计算 $\arcsin(x)$ 。

输入：x为张量，可以为`half`、`float32`、`float64`、`int32`、`int64`、`complex64`、`complex128`类型。

例：

```
x=tf.constant([[0,1,-1]],tf.float64)
z=tf.asin(x)
```

```
z==>[[0. 1.57079633 -1.57079633]]
```

1.39 tf.atan(x, name=None)

功能：计算 $\arctan(x)$ 。

输入：x为张量，可以为`half`、`float32`、`float64`、`int32`、`int64`、`complex64`、`complex128`类型。

例：

```
x=tf.constant([[0,1,-1]],tf.float64)
```

```
z=tf.atan(x)
```

```
z==>[[0. 0.78539816 -0.78539816]]
```

1.40 tf.lgamma(x, name=None)

功能：计算 $\ln(\Gamma(x))$ 。

输入：x为张量，可以为`half`、`float32`、`float64`类型。

例：

```
x=tf.constant([[1,2,3]],tf.float64)
```

```
z=tf.lgamma(x)
```

```
z==>[[0. 0. 0.69314718]]
```

1.41 tf.digamma(x, name=None)

功能：计算 \lgamma 的导数，即 γ'/γ 。

输入：x, y为张量，可以为`half`、`float32`、`float64`类型。

例：

```
x=tf.constant([[1,2,3]],tf.float64)
```

```
z=tf.digamma(x)
```

```
z==>[[-0.57721566 0.42278434 0.92278434]]
```

1.42 tf.erf(x, name=None)

功能：计算x的高斯误差。

输入：x为张量或稀疏张量，可以为`half`、`float32`、`float64`类型。

例：

```
x=tf.constant([[-1,0,1,2,3]],tf.float64)
```

```
z=tf.erf(x)
```

```
z==>[[-0.84270079 0. 0.84270079 0.99532227 0.99997791]]
```

1.43 tf.erfc(x, name=None)

功能：计算x高斯互补误差。

输入：x为张量，可以为`half`、`float32`、`float64`类型。

例：

```
x=tf.constant([[-1,0,1,2,3]],tf.float64)
```

```
z=tf.erfc(x)
```



```
z==>[[1.84270079 1.00000000 0.15729920 4.67773498e-03 2.20904970e-05]]
```

1.44 tf.squared_difference(x, y, name=None)

功能: 计算 $(x-y)$ $(x-y)$ 。

输入: x为张量, 可以为`half`,`float32`,`float64`类型。

例:

```
x=tf.constant([[ -1, 0, 2]], tf.float64)
y=tf.constant([[ 2, 3, 4]], tf.float64)
z=tf.squared_difference(x, y)
```

```
z==>[[9. 9. 4.]]
```

1.45 tf.igamma(a, x, name=None)

功能: 计算 $\gamma(a, x) / \gamma(a)$, $\gamma(a, x) = \int_0^x t^{(a-1)} \exp(-t) dt$ 。

输入: x为张量, 可以为`float32`,`float64`类型。

例:

```
a=tf.constant(1, tf.float64)
x=tf.constant([[1, 2, 3, 4]], tf.float64)
z=tf.igamma(a, x)
```

```
z==>[[0.63212056 0.86466472 0.95021293 0.98168436]]
```

1.46 tf.igammac(a, x, name=None)

功能: 计算 $\gamma(a, x) / \gamma(a)$, $\gamma(a, x) = \int_x^\infty t^{(a-1)} \exp(-t) dt$ 。

输入: x为张量, 可以为`float32`,`float64`类型。

例:

```
x=tf.constant([[ -1, 0, 1, 2, 3]], tf.float64)
z=tf.erf(x)
```

```
z==>[[-0.84270079 0. 0.84270079 0.99532227 0.99997791]]
```

1.47 tf.zeta(x, q, name=None)

功能: 计算Hurwitz zeta函数。

输入: x为张量或稀疏张量, 可以为`float32`,`float64`类型。

例:

```
a=tf.constant(1, tf.float64)
x=tf.constant([[1, 2, 3, 4]], tf.float64)
z=tf.zeta(x, a)
```

```
z==>[[inf 1.64493407 1.2020569 1.08232323]]
```

1.48 tf.polygamma(a, x, name=None)

功能: 计算 $\psi^{(a)}(x)$, $\psi^{(a)}(x) = (d^a/dx^a) * \psi(x)$, ψ 即为polygamma。

输入: x为张量, 可以为`float32`,`float64`类型。a=tf.constant(1, tf.float64)

例:

```
x=tf.constant([[1, 2, 3, 4]], tf.float64) z=tf.polygamma(a, x)
```

```
z==>[[1.64493407 0.64493407 0.39493407 0.28382296]]
```

1.49 tf.betainc(a, b, x, name=None)

功能: 计算 $I_x(a, b)$ 。 $I_x(a, b) = \{B(x; a, b)\} / \{B(a, b)\}$ 。

$B(x; a, b) = \int_0^x t^{a-1} (1-t)^{b-1} dt$ 。

$B(a, b) = \int_0^1 t^{a-1} (1-t)^{b-1} dt$ 。即完全

beta函数。

输入: x为张量, 可以为`float32`, `float64`类型。a,b与x同类型。

例:

```
a=tf.constant(1,tf.float64)b=tf.constant(1,tf.float64)x=tf.constant([[0,0.5,1]],tf.float64)
```

```
z==>[[0. 0.5 1.]]
```

1.50 tf.rint(x, name=None)

功能: 计算离x最近的整数, 若为中间值, 取偶数值。

输入: x为张量, 可以为`half`, `float32`, `float64`类型。

例:

```
x=tf.constant([[-1.7,-1.5,-1.1,0.1,0.5,0.4,1.5]],tf.float64)
z=tf.rint(x)
```

```
z==>[[-2. -2. -1. 0. 0. 0. 2.]]
```