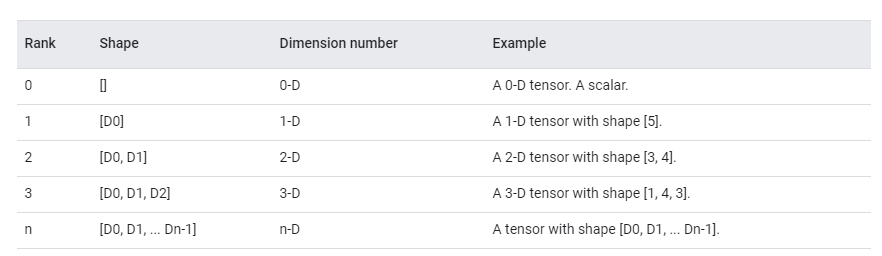
# 1.张量

TensorFlow 2.0 API 文档： https://www.tensorflow.org/api/r2.0

TensorFlow 使用一种叫tensor的数据结构去定义所有的数据，我们可以把tensor 看成是n维的array 或者list。在TensorFlow的各部分图形间流动传递的只能是tensor。

编写TensorFlow程序时，操纵并传递的主要对象是tf.Tensor：



**import** **tensorflow** **as** **tf**

print(tf.\_\_version\_\_)

2.0.0-alpha0

*# 是否可以在gpu上执行*

print(tf.test.is\_gpu\_available())

True

## 1.1.张量的创建

*# 零阶张量*

mammal=tf.Variable("Elephant",tf.string)

*# 输出张量的阶数*

tf.print(tf.rank(mammal))

*# 输出张量的形状*

tf.print(tf.shape(mammal))

0

[]

*# 直接打印张量*

print(tf.rank(mammal))

tf.Tensor(0, shape=(), dtype=int32)

*# 一阶张量*

mystr=tf.Variable(["Hello"],tf.string)

tf.print(tf.rank(mystr))

tf.print(tf.shape(mystr))

1

[1]

*# 二阶张量*

mymat=tf.Variable([[7],[11]],tf.int16)

tf.print(tf.rank(mymat))

tf.print(tf.shape(mymat))

2

[2 1]

更多创建方式

tf.constant([1,2,3],dtype=tf.int16)

<tf.Tensor: id=42, shape=(3,), dtype=int16, numpy=array([1, 2, 3], dtype=int16)>

tf.zeros((2,2),dtype=tf.int16)

<tf.Tensor: id=45, shape=(2, 2), dtype=int16, numpy=

array([[0, 0],

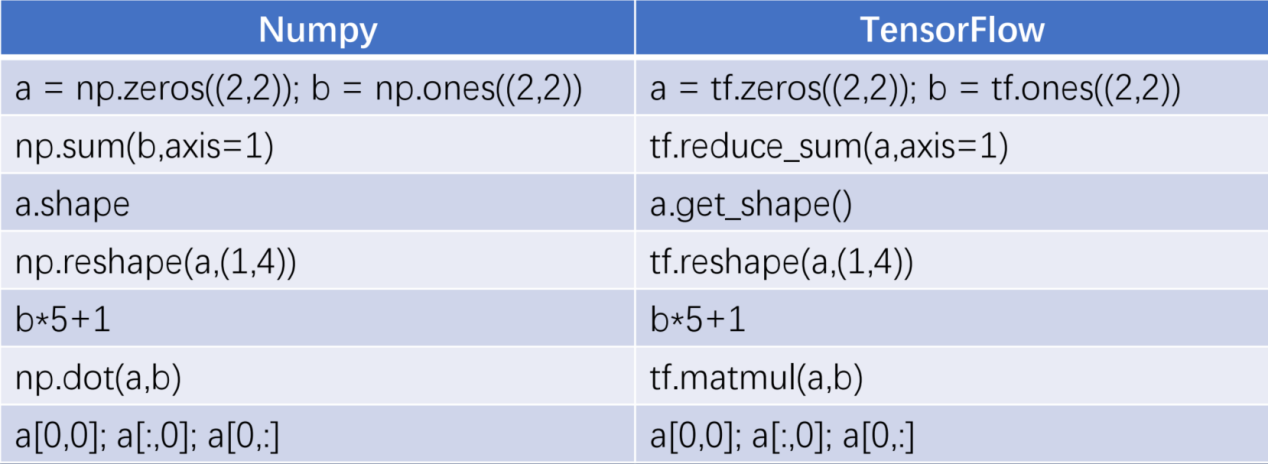
[0, 0]], dtype=int16)>

*#reshape*

rank\_three\_tensor=tf.ones([3,4,5])

matrix=tf.reshape(rank\_three\_tensor,[6,10])

Numpy与张量创建的比较



## 1.2.张量的操作

常用的操作：

tf.strings -- 与字符串相关操作

tf.debugging -- 断点相关操作

tf.dtypes -- 数据类型相关操作

tf.math -- 数学相关操作

tf.random -- 生成随机数相关操作

tf.feature\_column --特征结构相关操作

**tf.strings**

*# 单词切割，默认识别符号切割*

tf.strings.split('hello world')

<tf.Tensor: id=168, shape=(2,), dtype=string, numpy=array([b'hello', b'world'], dtype=object)>

*# 指定符号切割*

tf.strings.split('hello world', 'o')

<tf.Tensor: id=332, shape=(3,), dtype=string, numpy=array([b'hell', b' w', b'rld'], dtype=object)>

*#字符切割*

tf.strings.bytes\_split('hello')

<tf.Tensor: id=109, shape=(5,), dtype=string, numpy=array([b'h', b'e', b'l', b'l', b'o'], dtype=object)>

**tf.debugging**（使用并不多）

*#tf自带debug函数*

a=tf.random.uniform((10,10))

tf.debugging.assert\_equal(x=a.shape,y=(10,10))

*# 错误示范*

tf.debugging.assert\_equal(x=a.shape,y=(20,10))

**InvalidArgumentError**:

Condition x == y did not hold.

**tf.random**

*# 生成随机数，最小为0，最大为10*

a = tf.random.uniform(shape=(4,5),minval=0,maxval=10)

a

<tf.Tensor: id=781, shape=(10, 5), dtype=float32, numpy=

array([[7.337245 , 8.563206 , 0.12907624, 7.0185113 , 6.942792 ],

[8.801897 , 0.03558278, 0.71659565, 9.032879 , 2.803135 ],

[4.7294226 , 3.1436408 , 9.654947 , 5.030531 , 3.061322 ],

[7.6798067 , 1.7466497 , 4.6031084 , 5.355513 , 8.813373 ]],

dtype=float32)>

**tf.math**

a = tf.constant([[1,2],[3,4]])

b = tf.constant([[5,6],[7,8]])

tf.print(tf.math.add(a,b))

tf.print(tf.math.subtract(a,b))

tf.print(tf.math.multiply(a,b))

tf.print(tf.math.divide(a,b))

[[6 8]

[10 12]]

[[-4 -4]

[-4 -4]]

[[5 12]

[21 32]]

[[0.2 0.33333333333333331]

[0.42857142857142855 0.5]]

**tf.dtypes**

x =tf.constant([1.8,2.2],dtype=tf.float32)

x1=tf.dtypes.cast(x,tf.int32)

x1

<tf.Tensor: id=796, shape=(2,), dtype=int32, numpy=array([1, 2])>