TensorFlow2教程-張量

由命名可知，TensorFlow是一個定義和運算涉及張量的計算的框架。TensorFlow將張量表示為基底資料型別的n維陣列。

tf.Tensor具有以下屬性：

- 資料類型（float32, int32, string等）
- 形狀

一個張量中的每個元素都是一樣的資料類型，且資料類型已知。形狀可能已知也可能部分已知。某些情況下，只有執行圖時才知道張量的形狀。

一些特殊的張量：

- tf.Variable
- tf.constant
- tf.placeholder
- tf.SparseTensor

1 Rank

rank是tf.Tensor物件的尺寸的數量。類似於order,degree或 n 維。TensorFlow中的rank和資料中矩陣的秩不同。如下表所示，TensorFlow每個rank對應一個不同的數學實體。

秩	數學實體
0	標量（僅幅度）
1	向量（大小和方向）
2	矩陣（數字表）
3	3維張量（數字的立方）
ñ	n維張量

1.1 rank 0

rank0張量創建

In [2]:
```
import tensorflow as tf
mammal = tf.Variable('Hongmeng', tf.string)
ignition = tf.Variable(360, tf.int16)
floating = tf.Variable(3.141592653, tf.float64)
its_complicated = tf.Variable(12.3 - 4.56j, tf.complex64)
```

注：在TensorFlow中，字串被視為單個物件，而不是字元序列。可能有標量字串，字串向量等。

1.2 rank 1

要創建rank 1的tf.Tensor對象，一般傳遞清單作為初始值

In [3]:
```
mystr = tf.Variable(["Hello"], tf.string)
cool_numbers  = tf.Variable([3.14159, 2.71828], tf.float32)
first_primes = tf.Variable([2, 3, 5, 7, 11], tf.int32)
its_very_complicated = tf.Variable([12.3 - 4.85j, 7.5 - 6.23j], tf.complex64)
```

1.3 rank 2

rank 2包含行跟列

In [5]:
```
mymat = tf.Variable([[7],[11]], tf.int16)
myxor = tf.Variable([[False, True],[True, False]], tf.bool)
linear_squares = tf.Variable([[4], [9], [16], [25]], tf.int32)
squarish_squares = tf.Variable([ [4, 9], [16, 25] ], tf.int32)
rank_of_squares = tf.rank(squarish_squares)
mymatC = tf.Variable([[7],[11]], tf.int32)
```

1.4 高階張量

高階張量由n維陣列組成。例如，在影像處理期間，使用了許多等級4的張量，其尺寸對應於批中示例，圖像高度，圖像寬度和顏色通道。

In [6]:
```
my_image = tf.zeros([10, 299, 299, 3])
```

1.5 獲取tf.Tensor對象的rank

要確定tf.Tensor物件的rank，可以調用tf.rank方法。例如，以下方法以程式設計方式確定

In [8]:
```
r = tf.rank(my_image)
print(r)
```

```
tf.Tensor(4, shape=(), dtype=int32)
```

1.6 tf.Tensor切片

由於 tf.Tensor是n維陣列，因此要訪問其中的某個元素，需要指定n個索引。

對於等級0的張量（標量），不需要索引，因為它已經是一個數字。

對於秩為1的張量（向量），傳遞單個索引可訪問數位：

In [11]:
```
my_scalar = first_primes[2]
```

對於2級或更高的張量，情況更加有趣。對於 tf.Tensor的rank為2的，傳遞兩個數字將按預期返回標量：

In [14]:
```
my_scalar = mymat[1, 0]
```

但是，傳遞單個數字將返回矩陣的子向量

In [15]:
```
my_row_vector = mymat[1]
my_column_vector = mymat[:, 0]
```

2 形狀

張量的形狀是每個維度中元素的數量。TensorFlow在圖形構建過程中自動推斷形狀。這些推斷的形狀可能具有已知或未知的等級。如果rank是已知的，則每個維度的大小可能是已知的或未知的。

TensorFlow文檔使用三種符號約定來描述張量維數：rank，形狀和維數。下表顯示了它們之間的相互關係：

rank	形狀	維數	例子
0	[]	0維	0維張量。標量。
1	[D0]	一維	形狀為[5]的一維張量。
2	[D0，D1]	2維	形狀為[3，4]的二維張量。
3	[D0，D1，D2]	3維	形狀為[1、4、3]的3-D張量。
ñ	[D0，D1，... Dn-1]	D維	形狀為[D0，D1，... Dn-1]的張量。

2.1 獲取tf.Tensor對象的形狀

有兩種訪問形狀tf.Tensor的方法。在構建圖形時，詢問關於張量形狀的已知知識通常很有用。

- 1、可以通過讀取物件的shape屬性來獲取tf.Tensor。此方法返回一個TensorShape物件，這是表示部分指定的形狀的便捷方式（因為在構建圖形時，並非所有形狀都將是完全已知的）。
- 2、也有可能在運行時獲得一個代表某個張量的形狀的tf.Tensor。這是通過調用tf.shape 操作來完成的。這樣，可以通過構建依賴於輸入的動態形狀的其他張量來完成網路構建。

例如，下面是如何製作一個零向量，其大小與給定矩陣中的列數相同：

In [17]:
```
zeros = tf.zeros(mymat.shape[1])
```

2.2 改變tf.Tensor的形狀

可以通過tf.reshape來改變張量的形狀，只需確保所有形狀的尺寸的乘積相同

In [18]:
```
rank_three_tensor = tf.ones([3, 4, 5])
matrix = tf.reshape(rank_three_tensor, [6, 10])
matrixB = tf.reshape(matrix, [3, -1])  .
matrixAlt = tf.reshape(matrixB, [4, 3, -1])
```

3 資料類型

除維數外，張量還具有資料類型。

具有一個以上的資料類型是不可能的。但是，可以將任意資料結構序列化為strings並將其存儲在tf.Tensors中。

可以使用tf.cast命令將tf.Tensors從一種資料類型轉換為另一種資料類型：

In [19]:
```
float_tensor = tf.cast(tf.constant([1, 2, 3]), dtype=tf.float32)
```

要檢查tf.Tensor的資料類型，請使用Tensor.dtype屬性。

tf.Tensor從python物件創建時，可以選擇指定資料類型。如果不設置，TensorFlow會選擇一種可以代表資料的資料類型。TensorFlow將Python整數轉換為tf.int32，並將python浮點數轉換為tf.float32。而在將numpy轉換為陣列時使用的相同規則。

4 列印張量

出於調試目的，可能需要列印tf.Tensor的值。TensorFlow2中可以直接用print列印

In [21]:
```
t = tf.add(3, 4)
print(t)
```

```
tf.Tensor(7, shape=(), dtype=int32)
```