Tensor Ranks, Shapes, and Types

TensorFlow programs use a tensor data structure to represent all data. You can think of a TensorFlow tensor as an n-dimensional array or list. A tensor has a static type and dynamic dimensions. Only tensors may be passed between nodes in the computation graph.

Rank

In the TensorFlow system, tensors are described by a unit of dimensionality known as *rank*. Tensor rank is not the same as matrix rank. Tensor rank (sometimes referred to as *order* or *degree* or *n-dimension*) is the number of dimensions of the tensor. For example, the following tensor (defined as a Python list) has a rank of 2:

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

A rank two tensor is what we typically think of as a matrix, a rank one tensor is a vector. For a rank two tensor you can access any element with the syntax `t[i, j]`. For a rank three tensor you would need to address an element with `t[i, j, k]`.

Rank	Math entity	Python example
0	Scalar (magnitude only)	`s = 483`
1	Vector (magnitude and direction)	`v = [1.1, 2.2, 3.3]`
2	Matrix (table of numbers)	`m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
3	3-Tensor (cube of numbers)	`t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]`
n	n-Tensor (you get the idea)	`....`

Shape

The TensorFlow documentation uses three notational conventions to describe tensor dimensionality: rank, shape, and dimension number. The following table shows how these relate to one another:

Rank	Shape	Dimension number	Example
0	[]	0-D	A 0-D tensor. A scalar.
1	[D0]	1-D	A 1-D tensor with shape [5].
2	[D0, D1]	2-D	A 2-D tensor with shape [3, 4].
3	[D0, D1, D2]	3-D	A 3-D tensor with shape [1, 4, 3].
n	[D0, D1, ... Dn-1]	n-D	A tensor with shape [D0, D1, ... Dn-1].

Shapes can be represented via Python lists / tuples of ints, or with the `tf.TensorShape` (https://www.tensorflow.org/api_docs/python/tf/TensorShape).

Data types

In addition to dimensionality, Tensors have a data type. You can assign any one of the following data types to a tensor:

Data type	Python type	Description
`DT_FLOAT`	`tf.float32`	32 bits floating point.
`DT_DOUBLE`	`tf.float64`	64 bits floating point.
`DT_INT8`	`tf.int8`	8 bits signed integer.
`DT_INT16`	`tf.int16`	16 bits signed integer.
`DT_INT32`	`tf.int32`	32 bits signed integer.
`DT_INT64`	`tf.int64`	64 bits signed integer.
`DT_UINT8`	`tf.uint8`	8 bits unsigned integer.
`DT_UINT16`	`tf.uint16`	16 bits unsigned integer.
`DT_STRING`	`tf.string`	Variable length byte arrays. Each element of a Tensor is a byte array.
`DT_BOOL`	`tf.bool`	Boolean.
`DT_COMPLEX64`	`tf.complex64`	Complex number made of two 32 bits floating points: real and imaginary parts.
`DT_COMPLEX128`	`tf.complex128`	Complex number made of two 64 bits floating points: real and imaginary parts.

Data type	Python type	Description
`DT_QINT8`	`tf.qint8`	8 bits signed integer used in quantized Ops.
`DT_QINT32`	`tf.qint32`	32 bits signed integer used in quantized Ops.
`DT_QUINT8`	`tf.quint8`	8 bits unsigned integer used in quantized Ops.

上次更新日期：三月 8, 2017