术语表

广播操作(Broadcasting operation)

一种用[numpy-style broadcasting](http://docs.scipy.org/doc/numpy/user/basics.broadcasting.html)来保证tensor参数的形态兼容的操作。

Devices

一块可以用来运算并且拥有自己的地址空间的硬件，比如GPU和CPU。

eval

Tensor 的一个方法，返回 Tensor 的值。触发任意一个图计算都需要计算出这个值。只能在一个已经启动的会话的图中才能调用该 Tensor 值。

Feed

TensorFlow 的一个概念：把一个 Tensor 直接连接到一个会话图表中的任意节点。feed 不是在构建图(graph)的时候创建，而是在触发图的执行操作时去申请。一个 feed 临时替代一个带有 Tensor 值的节点。把feed数据作为run( )方法和eval( )方法的参数来初始化运算。方法运行结束后，替换的 feed 就会消失，而最初的节点定义仍然还在。可以通过tf.placeholder( )把特定的节点指定为 feed 节点来创建它们。详见[Basic Usage](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/get_started/basic_usage.md).

Fetch

TensorFlow中的一个概念：为了取回运算操作的输出结果。取回的申请发生在触发执行图操作的时候，而不是发生在建立图的时候。如果要取回一个或多个节点（node）的 Tensor 值，可以通过在 Session 对象上调用run( )方法并将待取回节点（node）的列表作为参数来执行图表(graph)。详见[Basic Usage](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/get_started/basic_usage.md)。

Graph(图)

把运算任务描述成一个直接的无环图形（DAG），图表中的节点（node）代表必须要实现的一些操作。图中的边代表数据或者可控的依赖。GratheDef 是系统中描述一个图表的协议(api)，它由一个 NodeDefs 集合组成。一个GraphDef可以转化成一个更容易操作的图表对象。

IndexedSlices（索引化切片）

在 Python API 中，TensorFlow 仅仅在第一维上对 Tensor 有所体现。如果一个 Tensor 有k维，那么一个 IndexedSlices 实例在逻辑上代表一个沿着这个 Tensor 第一维的(k-1)维切片的集合。切片的索引被连续储存在一个单独的一维向量中，而对应的切片则被拼接成一个单独的k维 Tensor。如果 sparsity 不是受限于第一维空间，请用 SparseTensor。

Node（节点）

图中的一个元素。 把启动一个特定操作的方式称为特定运算图表的一个节点，包括任何用来配置这个操作的属性的值。对于那些多形态的操作，这些属性包括能完全决定这个节点（Node）签名的充分信息。详见graph.proto。

操作（Op/operation）

在 TensorFlow 的运行时中，它是一种类似 add 或 matmul 或 concat的运算。可以用[how to add an op](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/how_tos/adding_an_op/index.md)中的方法来向运行时添加新的操作。

在 Python 的API中，它是图中的一个节点。在[tf.Operation](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/framework.md)类中列举出了这些操作。一个操作(Operation)的 type 属性决定这个节点（node）的操作类型，比如add和matmul。

Run

在一个运行的图中执行某种操作的行为。要求图必须运行在会话中。

在 Python 的 API 中，它是 Session 类的一个方法[tf.Session.run](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/client.md)。可以通过 Tensors 来订阅或获取run( )操作。

在C++的API中，它是[tensorflow::Session](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/client.md)类 的一个方法。

Session(会话)

启动图的第一步是创建一个 Session 对象。Session 提供在图中执行操作的一些方法。

在 Python API中，使用[tf.Session](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/client.md)。

在 C++ 的API中，[tensorflow::Session](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/cc/ClassSession.md)是用来创建一个图并运行操作的类：

Shape

Tensor 的维度和它们的大小。

在一个已经启动的图中，它表示流动在节点（node）之间的 Tensor 的属性。一些操作对 shape 有比较强的要求，如果没有 Shape 属性则会报告错误。

在 Python API中，用创建图的 API 来说明 Tensor 的 Shape 属性。Tensor 的Shape 属性要么只有部分已知，要么全部未知。详见[tf.TensroShape](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/framework.md)

在C++中，Shape 类用来表示 Tensor 的维度。[tensorflow::TensorShape](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/cc/ClassTensorShape.md)。

SparseTensor

在 Python API 中，它用来表示在 TensorFlow 中稀疏散落在任意地方的 Tensor 。SparseTensor 以字典-值格式来储存那些沿着索引的非空值。换言之，m个非空值，就包含一个长度为m的值向量和一个由m列索引(indices)组成的矩阵。为了提升效率，SparseTensor 需要将 indice（索引）按维度的增加来按序存储，比如行主序。如果稀疏值仅沿着第一维度，就用 IndexedSlices。

Tensor

Tensor是一种特定的多维数组。比如，一个浮点型的四维数组表示一小批由[batch,height，width，channel]组成的图片。

在一个运行的图(graph)中，它是一种流动在节点（node）之间的数据。 在 Python 中，Tensor 类表示添加到图的操作中的输入和输出，见[tf.Tensor](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/python/framework.md)，这样的类不持有数据。

在C++中，Tensor是方法[Session::Run( )](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/cc/ClassSession.md)的返回值，见[tensorflow::Tensor](https://github.com/jikexueyuanwiki/tensorflow-zh/blob/master/SOURCE/api_docs/cc/ClassTensor.md)，这样的 Tensor 持有数据。