

TensorFlow 全新的数据读取方式

Dataset API 是 TensorFlow 1.3 版本中引入的一个新的模块，主要服务于数据读取，构建输入数据的 pipeline。此前，在 TensorFlow 中读取数据一般有两种方法：使用 placeholder 读内存中的数据；使用 queue 读硬盘中的数据。Dataset API 同时支持从内存和硬盘的读取，相比之前的两种方法在语法上更加简洁易懂。此外，如果想要用到 TensorFlow 新出的 Eager 模式，就必须使用 Dataset API 来读取数据。

Dataset API 的导入。在 TensorFlow 1.3 中，Dataset API 是放在 contrib 包中的，而在 TensorFlow 1.4 中，Dataset API 已经从 contrib 包中移除，变成了核心 API 的一员。从基础的类来了解 Dataset API。只需要关注两个最重要的基础类：Dataset 和 Iterator。Dataset 可以看作是相同类型“元素”的有序列表。在实际使用时，单个“元素”可以是向量，也可以是字符串、图片，甚至是 tuple 或者 dict。以 Dataset 的每一个元素是一个数字创建了一个 dataset，这个 dataset 中含有 5 个元素，分别是 1.0, 2.0, 3.0, 4.0, 5.0。从 Dataset 中示例化一个 Iterator 的元素取出，对 Iterator 进行迭代。对应的输出结果应该就是从 1.0 到 5.0。如果一个 dataset 中元素被读取完了，再尝试 sess.run (one_element) 的话，就会抛出 tf.errors.OutOfRangeError 异常，这个行为与使用队列方式读取数据的行为是一致的。在实际程序中，可以在外界捕捉这个异常以判断数据是否读取完。

Dataset 支持一类特殊的操作：Transformation。一个 Dataset 通过 Transformation 变成一个新的 Dataset。通常可以通过 Transformation 完成数据变换、打乱、组成 batch、生成 epoch 等一系列操作。常用的 Transformation 有：map、batch、shuffle、repeat。map 接收一个函数，Dataset 中的每个元素都会被当作这个函数的输入，并将函数返回值作为新的 Dataset。batch 就是将多个元素组合成 batch，将 dataset 中的每个元素组成了大小为 32 的 batch。shuffle 的功能为打乱 dataset 中的元素，它有一个参数 buffersize，表示打乱时使用的 buffer 的大小。repeat 的功能就是将整个序列重复多次，主要用来处理机器学习中的 epoch，假设原先的数据是一个 epoch，使用 repeat (5) 就可以将之变成 5 个 epoch，如果直接调

用 repeat() 的话，生成的序列就会无限重复下去，没有结束，因此也不会抛出 tf.errors.OutOfRangeError 异常。

最简单的创建 Iterator 的方法就是通过 dataset.make_one_shot_iterator() 来创建一个 one shot iterator。除了这种 one shot iterator 外，还有 3 个更复杂的 Iterator，即：initializable iterator、reinitializable iterator、feedable iterator。initializable iterator 必须要在使用前通过 sess.run() 来初始化。使用 initializable iterator，可以将 placeholder 代入 Iterator 中，这可以方便通过参数快速定义新的 Iterator。initializable iterator 还有一个功能：读入较大的数组。在使用 tf.data.Dataset.from_tensor_slices (array) 时，实际上发生的事情是将 array 作为一个 tf.constants 保存到了计算图中。当 array 很大时，会导致计算图变得很大，给传输、保存带来不便。这时可以用一个 placeholder 取代这里的 array，并使用 initializable iterator，只在需要时将 array 传进去，这样就可以避免把大数组保存在图里。reinitializable iterator 和 feedable iterator 相比 initializable iterator 更复杂，也更加少用。

在非 Eager 模式下，Dataset 中读出的一个元素一般对应一个 batch 的 Tensor，可以使用这个 Tensor 在计算图中构建模型。在 Eager 模式下，Dataset 建立 Iterator 的方式有所不同，此时通过读出的数据就是含有值的 Tensor，方便调试。作为兼容两种模式的 Dataset API，在今后应该会成为 TensorFlow 读取数据的主流方式。

iOS 10 种线程锁

在使用多线程的时候多个线程可能会访问同一块资源，这样就很容易引发数据错乱和数据安全等问题，这时候就需要保证每次只有一个线程访问这一块资源，锁应运而生。1.OSSpinLock（自旋锁）；2.os_unfair_lock（自旋锁）；3.dispatch_semaphore（信号量）；4.pthread_mutex（互斥锁）；5.NSLock（互斥锁、对象锁）；6.NSCondition（条件锁、对象锁）；7.NSConditionLock（条件锁、对象锁）；8.NSRecursiveLock（递归锁、对象锁）；9.@synchronized（条件锁）；10.pthread_mutex (recursive)（递归锁）。