极客大学

极客大学机器学习训练营
Jax 简介

王然

众微科技 AI Lab 负责人

二〇二一年四月二十四日

大纲

大纲

大纲

为什么需要 Jax

- 核心问题：在投入可行的人力基础上，提升单模型的训练效率；
- 根本原因：模型越来越大、训练越来越复杂；
- 其他希望：借用 TensorFlow 和 PyTorch 的经验构造一个更好的深度学习框架。

TensorFlow 的优缺点

- 静态图：提升运算效率（优点）；API 十分反人类且和人们熟知的 API 不同（缺点）；难以 debug（缺点）；
- TensorFlow XLA：一套基础良好的机器学习编译器（优点）；难以直接调用（缺点）；
- 代码库过于复杂（缺点）；
- 缺少效率调优的选项（缺点）。

PyTorch 的优缺点

- Eager Execution：容易 Debug（优点）；API 比较自然（优点）；较难进行性能调优（缺点）；
- 代码库过于复杂（缺点）；
- 缺少效率调优的选项（缺点）。

Jax/Flax

极客大学

- 一套尽可能轻薄的、可以对运行效率进行调优的深度学习框架；
- Jax 为主要的功能实现库；Flax 是在 Jax 的基础上做了包装以保证它可以使用类似于 PyTorch 的 API；
- 目前来说，Jax/Flax 的生态圈（文档/代码库）尚不成熟。

大纲

jax.numpy 和 numpy

极客大学

- jax.numpy 是 numpy 的 jax 版本，其API和 numpy 基本一致；
- 但是 jax.numpy 和 numpy 也有不同。

jax.numpy 和 numpy 的不同之处

- 不可以直接进行 inplace update：jax.numpy 是不能自动更新的（协助程序分析和优化），如果需要更新，需要使用 index_update 或 index_add；
- 随机种子的处理：jax 需要指定随机种子，同样的随机种子所导致的生成的随机数/张量是一样的；
- jax.numpy 和 lax 均带有类似的 api：lax api 更底层，不会对数据类型进行自动调整，一般建议使用 jax.numpy 的 api；
- jax 默认精度为 float32，如果需要使用双精度（float64），需要在程序开始时运行相应命令；
- 细节见 notebook。

jit

- Jax 提速的核心是 jit；
- jit 的内部运行机制是非常复杂的；
- 最简单的 JIT 例子见 notebook jit simple 部分；
- 注意：这部分代码中，第二次运行时并没有运行 print 相关语句，这是由于所有代码已经被编译成为 XLA 代码，所以真实被执行的不是 Python 代码而是相应 XLA 相关语句。

jit 的坑之一：副作用

极客大学

- 为了使 jit 命令可以发挥其最大的效用，需要把 jit 函数变成 pure function，即函数不得有副作用；
- 如果函数有副作用，程序会发生其他结果；
- 两个常见的问题是 print 函数和 iterator；
- 对于需要记载状态（state）的函数，一般避免该问题的方法是将新的状态返回；
- 具体内容见 notebook。

jit 的坑之二：tracing 和 static variable/operations

极客大学

- 在我们执行第一个 jit 的时候，我们看到打印出来的为 tracedarray 而非具体的值；
- 一般来说，jax 会对 array 的 dtype 和 rank 进行跟踪，但不会对具体的值进行跟踪；
- 一个常见的结果就是，对于涉及到判断的语句而言，jax 常常会 trace 失败；
- 解决方法之一：在 jit 时候指定 static variable，注意 static variable 的变换会导致全部程序重新编译；
- 解决方法二：采用 static operation（例如 numpy）；
- 一个一般的解决策略，打印并观察哪些是 traced/static；
- 具体内容见 notebook。

jit 的坑之三：control flow

极客大学

- 一般方法，采用 lax.cond 和 lax.scan 替代 if 和 for，这两者均可以支持 jit 和 autograd；
- 注意：普通的 if 一般不可以支持 jit；
- 注意：如果将 for 循环的上界在编译时候指定为 static argument，则 jax 会进行 loop unrolling；
- 具体内容见 notebook。

jit、primitive 和 jaxpr

见该文档.

PyTree

极客大学

- 定义见PyTree 定义；
- 最常见应用：记录复杂的神经网络层参数；
- 最重要的两个函数：tree_map 和 tree_multimap；
- 具体内容见 notebook。

Async Dispatch

- Jax 内部会根据 jit 的结果，决定是否顺序执行所指定的程序，还是采用一定的并行；
- 大部分时候，只要符合 jax 的撰写规范，我们可以依靠 jit 进行 async dispatch。

vmap 和 pmap

极客大学

- vmap 为自动并行化的操作、pmap 是指在多个计算单元之间的并行；
- vmap 最常见的操作是对 batch 中每个样本进行映射；
- 具体见 notebook。

大纲

Jax 实践

极客大学

- 在我们之前的讲座当中，我们已经介绍了部分 Jax 作为加速 Numpy 工具的一些用法；
- 在这一章当中，我们重点来实现 Entmax-α 的前向传导和后向传导过程；
- 为了达到这个目标，我们首先先对 Jax 当中 autograd 的机制做一些简要说明；
- 基本的 autograd 语句见 notebook。

构建自己的后向传导方式

细节见 notebook

实践：Entmax-α

极客大学

- 具体细节见Peters, Niculae, and Martins (2019)；
- 实现步骤：
  - 实现前向传导（不加 jit）；
  - 实现后向传导（不加 jit）；
  - 加入 jit 并测试。

大纲

大纲


Peters, Ben, Vlad Niculae, and André FT Martins (2019). "Sparse sequence-to-sequence models". In: *arXiv preprint arXiv:1905.05702*.