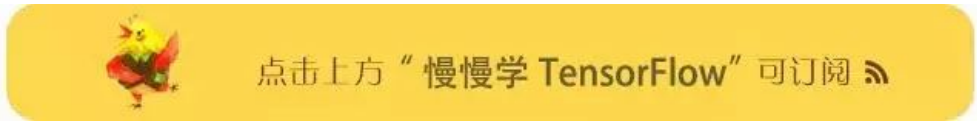
TensorFlow 2017 年大事记

原创 *2017-12-31 慢慢* 慢慢学TensorFlow



2017 年最后一天，我们来回顾下 TensorFlow 这一年都有哪些大的改变。

一、TensorFlow 1.0 发布

1 月 9 日，TF 1.0 alpha 版本发布，标志着 0.x 时代的终结。API 相比之前有了较大调整，本公众号也记录了一些使用新 API 过程中遇到的坑，详见以下文章：

《TensorFlow 1.0.0rc1 入坑记》

《TensorFlow 1.0.0rc1 入坑记（续）》

二、TensorFlow 全球开发者峰会

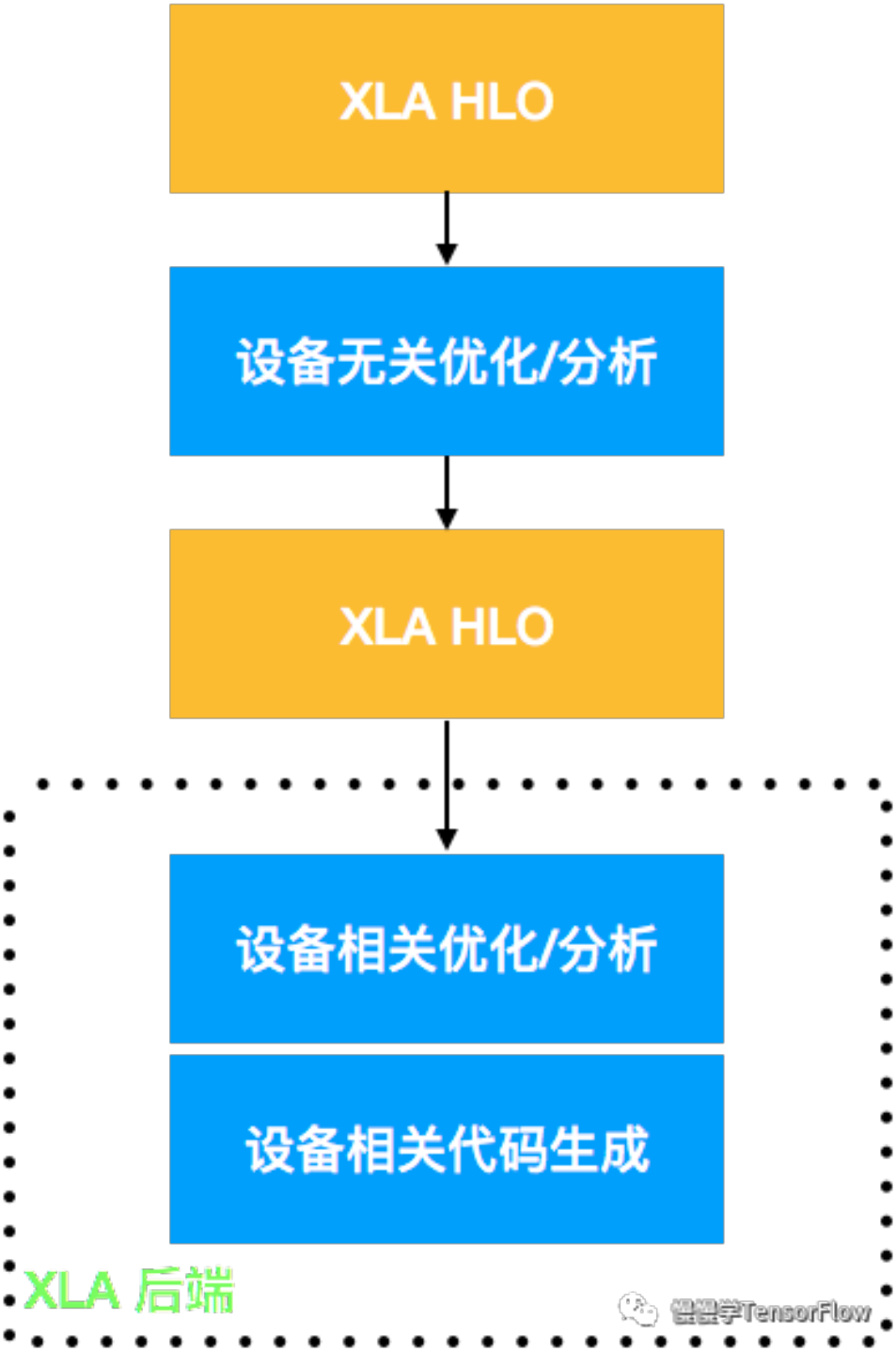
2 月 15 日，TensorFlow 第一届全球开发者峰会在加利福尼亚州山景城召开，同天发布了 TF 1.0 正式版。峰会从四个不同角度讲述了 TF：社区，应用，部署，工具和技巧。

首先 Jeff Dean Keynote 介绍了 TF 发展历程，对比了 TF 与上一代 DistBelief 区别。接着 Google 开发人员分别介绍了 XLA、TensorBoard、High-Level API、Serving、移动端等组件以及 TF 在皮肤癌图像分类、医学成像、广告推荐、音乐创作等领域的应用。

公众号后台回复“20171231”获取峰会视频资源链接。

三、TensorFlow XLA

加速线性代数(Accelerated Linear Algebra, XLA) 是领域相关编译器，可以用来优化 TensorFlow 计算图，改善速度、内存占用、可移植性（移动端和云端部署）。 XLA 特性迎合了利用新硬件（如 FPGA、DSP、AI ASIC 等）加速 TF 的开发者的口味。



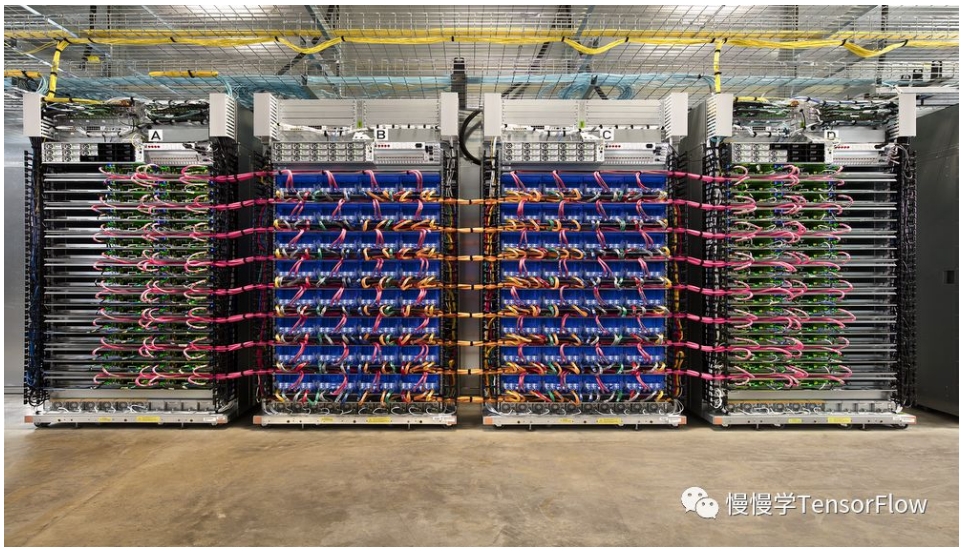
XLA 编译过程如上图。输入为 HLO IR（高层次优化器中间表示，可以看作编译器中间表示）。XLA 将 HLO 定义的计算图转换为不同硬件指令，在 TF 源码树中已经集成了 x86_64、ARM64、Nvidia GPU 后端。设备无关优化步骤会做一些诸如运算符合并、缓存分析等与具体硬件无关的优化，产生新的优化过的 HLO IR 送入后端进行下一步优化。后端可以执行进一步 HLO 分析和优化，此时会考虑硬件相关特性和限制，例如 XLA GPU 后端可能会根据 GPU 编程模型来决定如何将计算切分为多个 stream。最后一步是设备相关代码生成，采用 LLVM 为 CPU/GPU 提供低级 IR、优化、最终代码生成。

XLA 仍在试验阶段，需要使能 XLA 特性时，只能通过**源码编译**安装 TensorFlow。

四、Cloud TPU

5 月 17 日，Google 宣布第二代张量处理单元（TPU, Tensor Processing Units）部署到谷歌云来专门加速机器学习任务。相比第一代 TPU 增加了对训练的支持，每个新 TPU 设备可以提供 180 TFLOPS 浮点计算能力。

每个 TPU 包含定制高速网络接口，64 个 TPU 可以组网变成一个 TPU Pod，提供 11.5 PFLOPS 浮点处理能力，具备了小型超算的规模。



你可以登录 Google 云（https://cloud.google.com/tpu/ ）或注册 TensorFlow 研究云（https://www.tensorflow.org/tfrc/） 获取 Cloud TPU 使用资格。

TensorFlow 中使用 TPU 计算的例程：https://github.com/tensorflow/tpu-demos

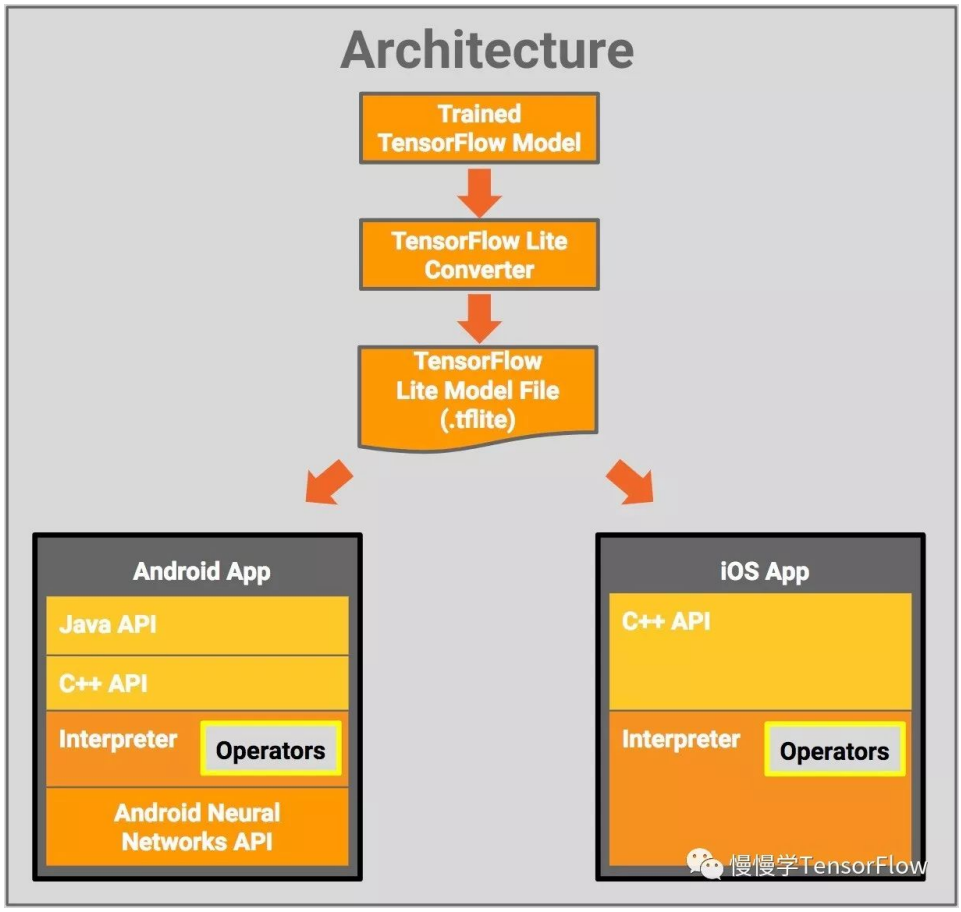
五、移动端设备支持

TensorFlow 不仅能在云端通过 TPU 来高效运行，还可以将训练好的模型部署到移动端，用户可以直接利用手机的传感器（相机、麦克风）采集实际数据，调用 TensorFlow API 实现实现更广泛的终端应用。

为了降低模型尺寸，最简单的解决方法是对模型进行压缩和量化，参考《TensorFlow 1.0.0rc1上玩量化神经网络》，另外今年 Google 还发布了《用于移动和嵌入式视觉应用的 MobileNets》从模型结构设计上降低模型尺寸。

模型准备好后，需要编译移动端工程，TensorFlow 提供了两种途径：TF for Mobile 和 TF Lite。其中 TF Lite 是今年 11 月发布的，是 TF for Mobile 的进化版，具有相对较小的app 尺寸，更少依赖，和更好的性能。但 TF Lite 还未产品化，TF for Mobile 已经产品化。

TF Lite 代码位置：tensorflow/contrib/lite，其架构图如下：

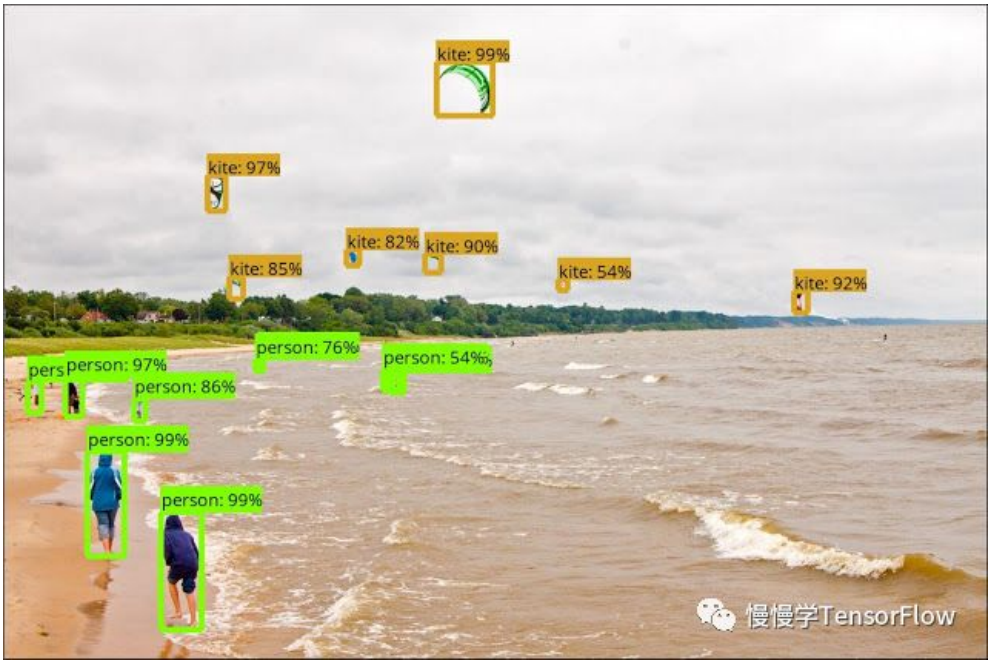


TF Mobile 代码位置：tensorflow/examples/android 和 tensorflow/examples/ios

本公众号之前的文章《如何在移动设备上运行 TensorFlow》介绍了如何在 TF 1.0 上编译 iOS app。

六、TensorFlow 目标检测 API

计算机视觉中目标检测是近年比较活跃的问题，多种检测框架（RCNN、Fast RCNN、Faster RCNN、Mask RCNN、SSD、YOLO、YOLOv2、R-FCN……）相继登场，各领风骚。TensorFlow Object Detection API 则简化了构建这些系统的步骤，可以灵活实现不同基础网络（MobileNets/ResNet/Inception）、不同检测方法（Faster RCNN/SSD/R-FCN）的混搭，实现。



搭建该系统的步骤可参考《TensorFlow Object Detection API 实践》。

七、动态计算图

TensorFlow Eager Execution 是今年 10 月份发布的动态计算图解决方案，从此可以告别 sess.run(result) 这种反人类的用法，而是更接近 Python 的方式，直接在调用处求值，降低初学者学习难度，也让研究人员用更直观的方式实现想法，方便调试。

Eager Execution 例程：tensorflow/contrib/eager/python/examples/mnist/mnist.py

八、结语

短短一年时间，TF 发生了很多变化，这篇文章列举的只是冰山一角，还有很多 TF 模型也发生了悄然变化（NMT、GAN、WaveNet……），慢慢学习中。。。

2018，让 TF 新 feature 来的更猛烈些吧！


慢慢学TensorFlow