推理引擎 OpenPPL 实战训练营

商汤 sensetime

编程工作坊：基于 OpenPPL 的模型推理与应用部署

2021年12月17日星期五

课程安排

课程安排	主讲人	课程时间
第一期：商汤自研AI推理引擎 OpenPPL 的实践之路	高洋	2021年12月07日
第二期：编程工作坊：基于 OpenPPL 的模型推理与应用部署	**欧国宇**	**2021年12月16日**
第三期：OpenPPL 性能优化：通用架构下的性能优化概要	许志耿	2021年12月28日
第四期：模型大小与推理速度的那些事儿	田子宸	2022年01月06日
第五期：性能调优实战（x86篇）	梁杰鑫	敬请期待
第六期：性能调优实战（CUDA篇）	李天健	敬请期待
第七期：OpenPPL+RISC-V 指令集初探	焦明俊/杨阳	敬请期待
第八期：OpenPPL 在 ARM Server 上的技术实践	许志耿/邱君仪	敬请期待
第九期：量化工具实践	纪喆	敬请期待

「商汤学术」公众号

可以回复“抽奖”试试哦

实战训练营项目

项目亮点

- **全面讲解**：商汤资深研究员倾情讲授，基础知识一应俱全
- **项目实践**：拒绝纸上谈兵！多个**课程体验 Demo**，学完即可直接上手实操
- **实时答疑**：课程期间设有答疑环节，更有互动社群随时交流
- **专属社群**：9 期课程专属群，主讲人**实时解答**，更有多种互动好礼等你

社群有礼

- **实名社群**：亮出你的身份才能在社群内交到更多朋友哦
- **社群互动**：配合训练营安排，小助手将按时**提醒进展、分享资料、发布任务、解答疑问**
- **打榜好礼**：打卡课程、体验 Demo、参与互动均可收获 **“P 币”**。训练营期间，群内将定期送出**互动好礼**；结营之时，P 币累积在**总排行榜前十**的同学，更会收到**商汤精美定制大礼包**一份

实战训练营

我已经给你安排好了

如何获得P币?

项目	环节	P币	参加凭证
直播课程	参与弹幕互动	每期：每次发弹幕+1，上限不超过5	截图发至小助手
	参与QA提问	一个提问+3	截图发至小助手
	参与课程反馈	+3	后台实名查验
实战加码	一般任务	+30	以PR形式提交至master分支
	复杂任务	+60，并有机会获得实习绿色通道	以PR形式提交至master分支
学习心得	撰写体验文章	+30	公众号/知乎/博客
	撰写模块分析	+60，并有机会获得实习绿色通道	公众号/知乎/博客
社群互动	分享优质文章	+3	群内统计
	群内提出问题	+3	群内统计
	解答群员提问	+5	群内统计
更多活动，敬请期待……			

主讲人介绍

欧国宇

商汤高性能计算组 OpenPPL 开源负责人

- 商汤科技高性能计算组 OpenPPL 开源负责人，中国科学院研究生院硕士毕业
- 在商汤目前负责高性能推理引擎 PPL 的研发，其开源版本已于今年 6 月对社区开源

课程提纲

第二期课程将重点介绍人工智能推理的基本原理、开源高性能推理引擎 OpenPPL 的体系架构，并通过实际案例解释如何测试生成的训练模型，并将其部署到实际应用中。

1. 什么是推理引擎

- 从训练到推理
- ONNX 模型介绍

2. OpenPPL 介绍

- 支持的平台及算子
- 模块介绍
- 编程接口介绍

3. Demo 演示

实战训练营

基于 OpenPPL 的模型推理与应用部署

欧国宇

2021.12.16

OpenPPL

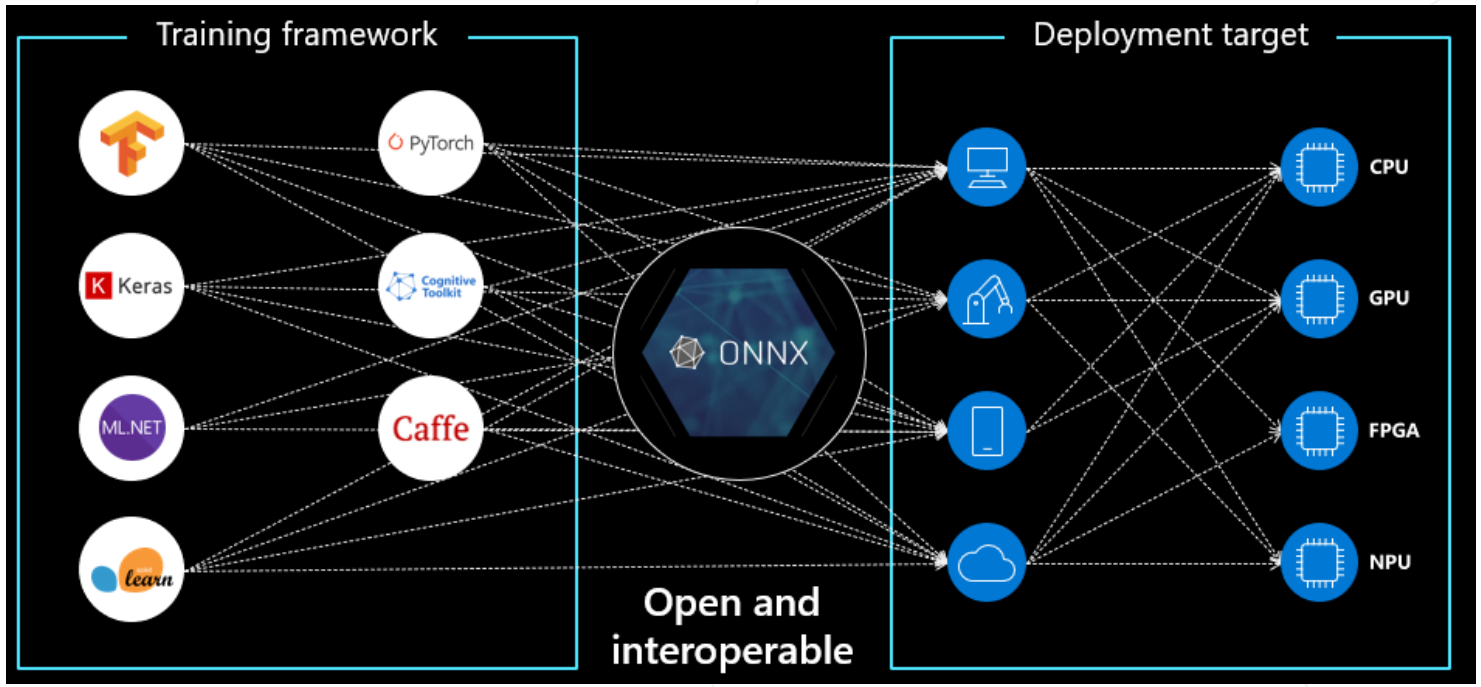
SensePPL 是商汤 HPC 团队从 2015 年开始精心打造的多后端深度学习推理部署引擎。训练平台训练好的模型，可以通过转换成 ONNX 等标准格式，使用 **SensePPL** 进行快速推理部署。

OpenPPL 是 **SensePPL** 的开源版本，能够让人工智能应用高效可靠地运行在现有的 CPU、GPU 等计算平台上，为云端场景提供人工智能推理服务。第一个版本从 v0.1 开始，包含了对 x86 架构 fp32 数据类型，以及 NVIDIA GPU 的 Turing 架构 fp16 数据类型的基本支持。

GitHub: https://github.com/openppl-public

ONNX

ONNX（Open Neural Network Exchange）是一种针对机器学习所设计的开放文件格式，用于存储训练好的模型，使得不同的人工智能框架可以采用相同格式存储模型数据并交互。



tensorflow-onnx Public
Convert TensorFlow, Keras, Tensorflow.js and Tflite models to ONNX
export · deep-learning · tensorflow · convert · keras · onnx · tflite
Jupyter Notebook · Apache-2.0 · 292 · 1,265 · 55 · 13 · Updated 4 hours ago

onnx-mlir Public
Representation and Reference Lowering of ONNX Models in MLIR Compiler Infrastructure
C++ · Apache-2.0 · 114 · 207 · 89 (2 issues need help) · 8 · Updated 5 hours ago

backend-scoreboard Public
Scoreboard for ONNX Backend Compatibility
onnx
Python · Apache-2.0 · 21 · 20 · 0 · 4 · Updated 8 hours ago

onnx-tensorrt Public
ONNX-TensorRT: TensorRT backend for ONNX
deep-learning · nvidia · onnx
C++ · Apache-2.0 · 402 · 1,780 · 113 · 11 · Updated 3 days ago

GitHub: https://github.com/openppl-public

ONNX

- 定义了模型结构和节点类型
  - ModelProto，GraphProto，NodeProto，...
- 标准算子定义
  - 由 domain，type 和 version 三者唯一指定
  - 包含循环和条件分支，四则运算等
- 支持多种数据类型
  - int8，int16，int32，int64，float，double
  - string，sequence，map

ai.onnx (default)

Operator	Since version
Abs	13, 6, 1
Acos	7
Acosh	9
Add	14, 13, 7, 6, 1
And	7, 1
ArgMax	13, 12, 11, 1
ArgMin	13, 12, 11, 1
Asin	7
Asinh	9
Atan	7

GitHub: https://github.com/openppl-public

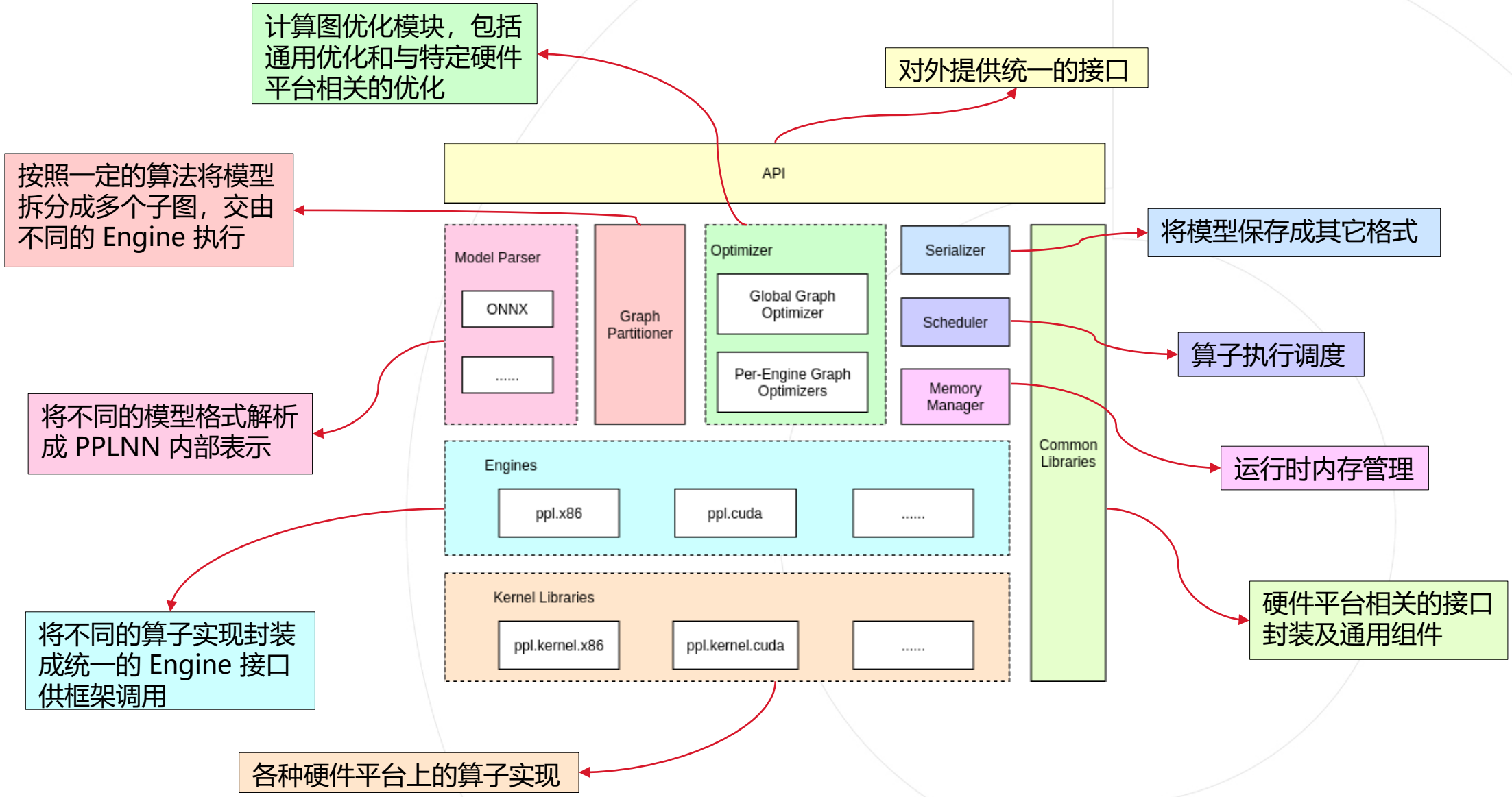
支持的平台和算子

- 操作系统
  - Linux，Windows，MacOS
- 设备平台
  - X86，CUDA
  - ARM server，RISCV（即将发布）
- 支持的算子列表

- ONNX

Op Type	Op Set	Linux/Windows/Darwin X86-64	Linux/Windows CUDA
Add	7~12	✓	✓
And	7~16	✓	✓
ArgMax	11	✓	✓
AveragePool	11~16	✓	✓
BatchNormalization	9~13	✓	✓
Cast	9~12	✓	✓
Ceil	6~12	✓	✓
Clip	11	✓	✓
Concat	11~12	✓	✓
Constant	9~16	✓	
ConstantOfShape	9~16	✓	✓
Conv	11~16	✓	✓
ConvTranspose	11~16	✓	✓
DepthToSpace	11~12	✓	✓
Div	7~12	✓	✓
Equal	11~12	✓	✓
Exp	6~12	✓	✓
Expand	8~12	✓	✓

模块介绍



GitHub: https://github.com/openppl-public

编程接口

- 兼顾功能和易用性
  - 常规流程
    - 初始化：runtime = runtime_builder.CreateRuntime();
    - 设置输入：runtime->GetInput(0)->CopyFromHost(data);
    - 执行推理：runtime->Run();
    - 获取输出：runtime->GetOutput(0)->CopyToHost();
  - 可配置
    - runtime->Configure(options, ...);
  - 性能
    - runtime->GetOutput(0)->GetBufferPtr();
- 支持语言
  - C++
  - Python
  - Lua

编程接口 – C++

- RuntimeBuilder
  - runtime_builder = OnnxRuntimeBuilderFactory.Create (model, engines.data(), engines.size());
- Runtime
  - runtime = runtime_builder->CreateRuntime();
- 设置输入
  - input_count = runtime->GetInputCount();
  - runtime->GetInput(0) ->ConvertFromHost(data);
- 执行推理
  - runtime->Run();
- 获取输出
  - output_count = runtime->GetOutputCount();
  - output = runtime->GetOutput(0) ->ConvertToHost();

编程接口 - Python

- RuntimeBuilder
  - runtime_builder = OnnxRuntimeBuilderFactory.CreateFromFile(model, engines)
- Runtime
  - runtime = runtime_builder.CreateRuntime();
- 设置输入
  - input_count = runtime.GetInputCount();
  - runtime.GetInput(0).ConvertFromHost(data);
- 执行推理
  - runtime.Run();
- 获取输出
  - output_count = runtime.GetOutputCount();
  - output = runtime.GetOutput(0).ConvertToHost();

编程接口 - Lua

- RuntimeBuilder
  - runtime_builder = OnnxRuntimeBuilderFactory:CreateFromFile(model, engines)
- Runtime
  - runtime = runtime_builder:CreateRuntime();
- 设置输入
  - input_count = runtime:GetInputCount();
  - runtime.GetInput(0):ConvertFromHost(data, dims, data_type);
- 执行推理
  - runtime:Run();
- 获取输出
  - output_count = runtime:GetOutputCount();
  - output = runtime:GetOutput(0):ConvertToHost();

Demo 演示

- C++ API
- 使用 OpenPPL 推理 OpenMMLab 经典检测网络 Mask R-CNN

P 币任务

根据 Demo 及配套教程，完成以下任一任务，即可获得商汤「P 币」

- 以 pull request 形式提交至 master 分支
  - （一般）基于 OpenPPL 最新版本更新英文文档
  - （复杂）基于 OpenPPL 最新版本，使用 OpenMMLab 的检测模型推理 demo

- 发表文章（公众号，知乎，博客平台均可）
  - （一般）基于 demo 部署撰写 OpenPPL 体验文章
  - （困难）OpenPPL 源码分析
  - 其它形式的文章

Q&A

https://github.com/openppl-public
https://openppl.ai/
https://www.zhihu.com/people/openppl