推理引擎 - 模型转换与优化

模型转换IR表示

ZOMI

Talk Overview

1. **推理系统介绍**
   - 推理系统架构
   - 推理引擎叫故
2. **模型小型化**
   - CNN小型化结构
   - Transform小型化结构
3. **离线优化压缩**
   - 低比特量化
   - 模型剪枝
   - 知识蒸馏
4. **模型转换与优化**
   - 架构与流程
   - 模型转换技术细节
   - 模型离线优化
5. **Runtime与在线优化**
   - 动态batch
   - bin Packing
   - 多副本并行

Talk Overview

I. 模型格式转换

- 转换模块挑战与架构
- 模型序列化/反序列化
- protobuffer / flatbuffer 格式
- 自定义计算图 IR
- 转换流程和技术细节

- **工程理论**
- **知识概念**

Talk Overview

I. 模型格式转换

- 转换模块挑战与架构
- 模型序列化/反序列化
- protobuffer / flatbuffer 格式
- 自定义计算图 IR
- 转换流程和技术细节

- **技术细节**
- **核心内容**

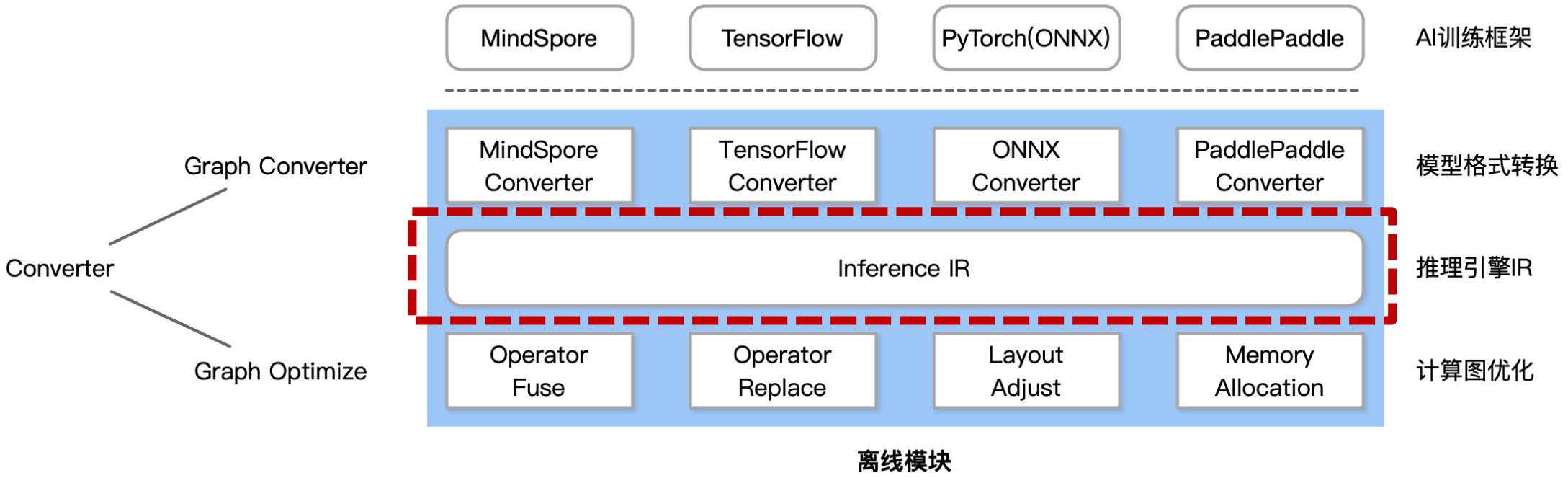
计算图回顾

Question?

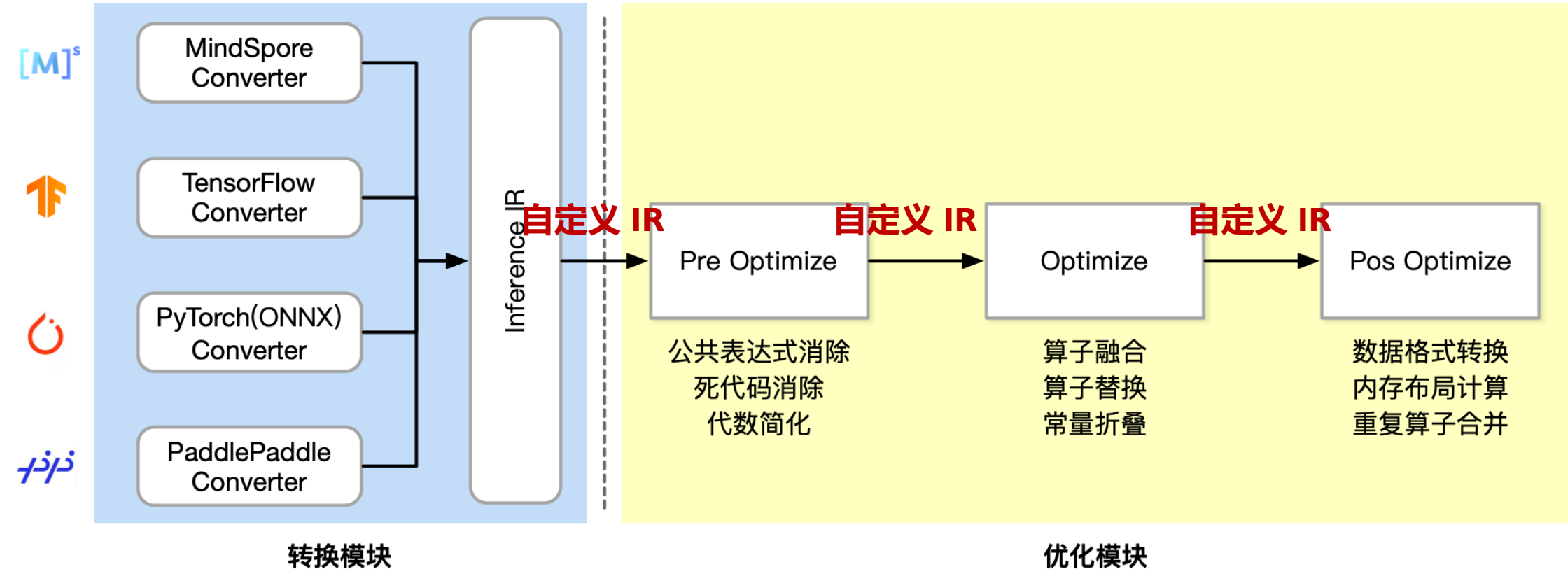
1. 为什么推理引擎需要自定义计算图？

转换模块架构

- Converter由Frontends和Graph Optimize构成。前者负责支持不同的AI 训练框架；后者通过算子融合、算子替代、布局调整等方式优化计算图：



转换模块的工作流程



AI框架之计算图

播放全部

4更新

络，框架需要解决诸多问题， 例如：如何实现自动求导，如何利用编译期分析对神经网络计算进行化
计算单元在加速器上的执行，如何将基本处理单元派发（dispatch）到特定的高效后端实现，如何...

默认排序　升序排序　编辑

计算图有哪些内容知识？【计算图】系列第一篇

789　2022-10-6

为什么AI框架都用计算图？什么是计算图？到底计算图有什

834　2022-10-8

计算图跟微分什么关系？怎么用计算图表示自动微分？AI框

653　2022-10-9

对计算图进行优化与执行调度！计算图优化跟AI编译器啥

556　2022-10-10

AI框架都是怎么表示控制流的？PyTorch和TF对计算图中

571　2022-10-11

AI框架如何实现动静统一？PyTorch和MindSpore动静统

419　2022-12-4

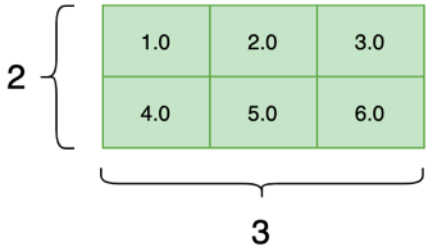
计算图未来将会走向何方？【计算图】第七篇

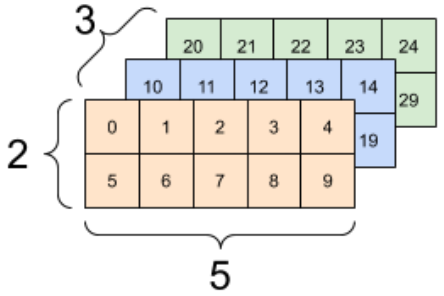
404　2022-10-12

基于计算图的AI框架：基本组成

基本数据结构：Tensor 张量

- Tensor形状： [2, 3, 4, 5]
- 元素类型：int, float, string, etc.





基本运算单元：Operator 算子

- 由最基本的代数算子组成
- 根据深度学习结构组成复杂算子
- N个输入Tensor，M个输出Tensor

Add	Log	While
Sub	MatMul	Merge
Mul	Conv	BroadCast
Div	BatchNorm	Reduce
Relu	Loss	Map
Floor	Sigmoid	…..

Question?

1. AI 框架与推理引擎的计算图有什么区别？

AI框架计算图 vs 推理引擎计算图

	AI框架计算图	推理引擎计算图
计算图组成	算子 + 张量 + 控制流	算子 + 张量 + 控制流
正反向	Forward + Backward	Forward
动静态	动态图 + 静态图 部分 AI 框架实现动静统一可以互相转换	以静态图为主
分布式并行	依托 AI 集群计算中心，计算图支持数据并行、张量并行、流水线并行等并行切分策略	以单卡推理服务为主，很少考虑分布式推理
使用场景	训练场景，以支持科研创新，模型训练和微调，提升算法精度	推理场景，以支持模型工业级部署应用，对外提供服务

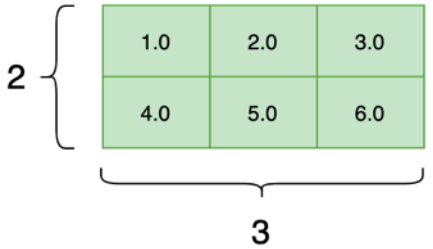
推理引擎

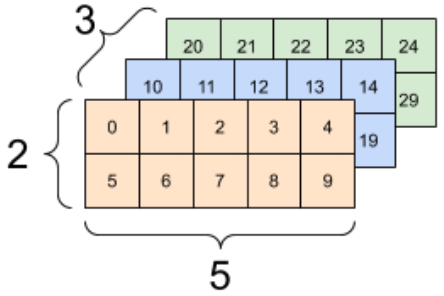
自定义计算图

基于计算图的AI框架：基本组成

基本数据结构：Tensor 张量

- Tensor形状： [2, 3, 4, 5]
- 元素类型：int, float, string, etc.





基本运算单元：Operator 算子

- 由最基本的代数算子组成
- 根据深度学习结构组成复杂算子
- N个输入Tensor，M个输出Tensor

Add	Log	While
Sub	MatMul	Merge
Mul	Conv	BroadCast
Div	BatchNorm	Reduce
Relu	Loss	Map
Floor	Sigmoid	…..

推理引擎计算图：Tensor 张量的表示

Tensor 数据存储格式

```
// 定义 Tensor 的数
enum DataType : i
  DT_INVALID = 0,
  DT_FLOAT = 1,
  DT_DOUBLE = 2,
  DT_INT32 = 3,
  DT_UINT8 = 4,
  DT_INT16 = 5,
  DT_INT8 = 6,
  // ...
}
```

Tensor 数据内存排布格式

```
// 定义 Tensor 数据排布格
enum DATA_FORMAT : byte
    ND,
    NCHW,
    NHWC,
    NC4HW4,
    NC1HWC0,
    UNKNOWN,
    // ...
}
```

Tensor 张量的定义

```
// 定义 Tensor
table Blob {
    // shape
    dims: [int];
    dataFormat: DATA_FORMAT;

    // data type
    dataType: DataType = DT_FLOAT;

    // extra
    // ...
}
```

推理引擎计算图： Operator 算子的表示

算子列表

```
37
38  // 推理引擎算子
39  enum OpType :
40      Const,
41      Convolut:
42      Convolut:
43      Deconvolu
44      Deconvolu
45      MatMul,
46      Padding,
47      // ...
48  }
49
```

算子公共属性和特殊算子列表

```
49
50  // 算子的公共属性和特
51  union OpParameter
52      WhileParam,
53      IfParam,
54      PadParam,
55      Range,
56      Act,
57      // ...
58  }
59
```

算子的基础定义

```
59
60  // 算子基础定义
61  table Op {
62      inputIndexes: [int];
63      outputIndexes: [int];
64      main: OpParameter;
65      type: OpType;
66      name: string;
67      // ...
68  }
69
```

推理引擎计算图： 计算图的表示

定义网络模型子图

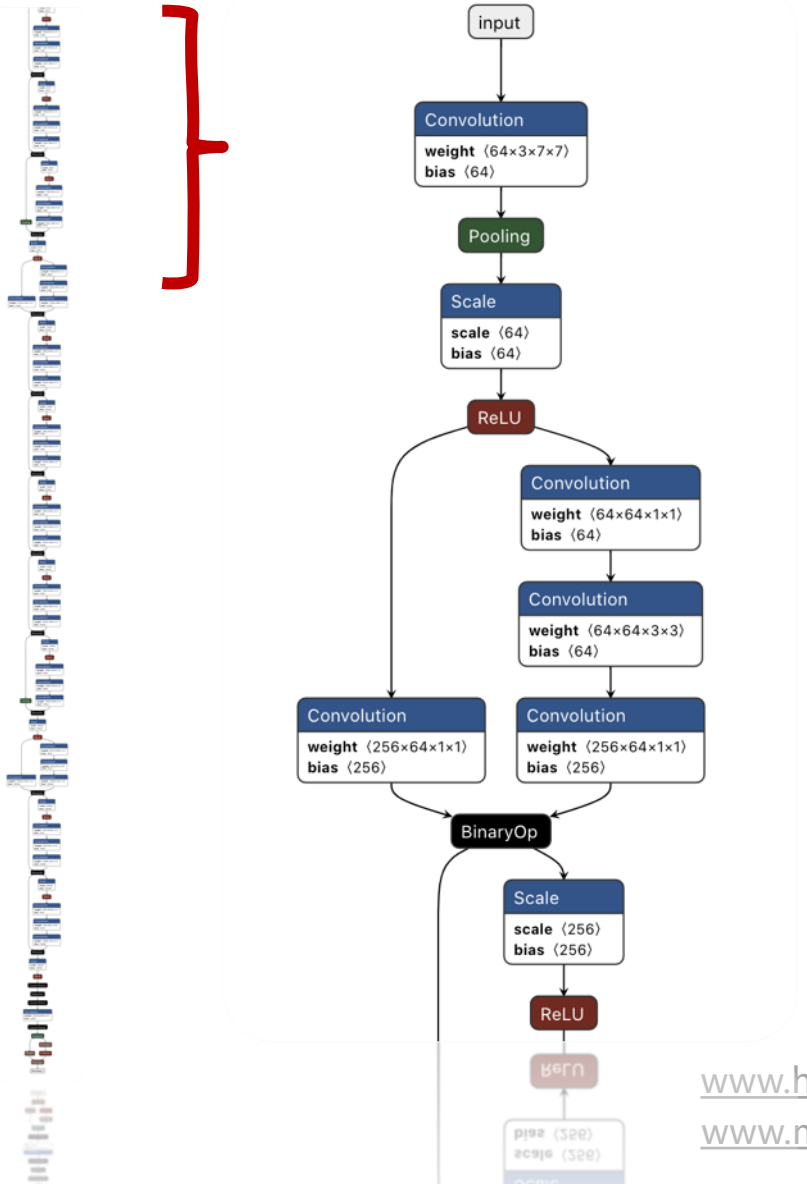
```
15  // 子图概念的定义
16  table SubGraph {
17      // Subgraph unique name.
18      name: string;
19      inputs: [int];
20      outputs: [int];
21
22      // All tensor names.
23      tensors: [string];
24
25      // Nodes of the subgraph.
26      nodes: [Op];
27  }
```

定义网络模型

```
2   // 网络模型定义
3   table Net {
4       name: string;
5       inputName: [string];
6       outputName: [string];
7       oplists: [Op];
8       sourceType: NetSource;
9
10      // Subgraphs of the Net.
11      subgraphs: [SubGraph];
12      // ...
13  }
14
```

自定义计算图

1. **构建计算图 IR**：根据自身推理引擎的特殊性和竞争力点，构建自定义的计算图
2. **解析训练模型**：通过解析 AI 框架导出的模型文件，使用 Protobuffer / flatbuffer 提供的API定义对接到自定义 IR 的对象
3. **生成自定义计算图**：通过使用 Protobuffer / flatbuffer 的API导出自定义计算图



参考文献

1. Huawei Technologies Co., Ltd. "Huawei MindSpore AI Development Framework." *Artificial Intelligence Technology*. Singapore: Springer Nature Singapore, 2022. 137-162.
2. Jiang, Xiaotang, et al. "Mnn: A universal and efficient inference engine." *Proceedings of Machine Learning and Systems* 2 (2020): 1-13.
3. https://onnx.ai/supported-tools
4. https://github.com/onnx/onnx/blob/main/docs/IR.md
5. https://gitee.com/mindspore/mindspore
6. https://github.com/alibaba/MNN
7. https://onnxruntime.ai/