MindIR:
the intermediate representation of MindSpore

Zichun Ye
2012 labs, Huawei

COMPUTATIONAL ABSTRACTIONS FOR PROBABILISTIC AND DIFFERENTIABLE PROGRAMMING WORKSHOP
2022.07.09
MindSpore: Open Source All-Scenario AI Framework

Develop once
Deploy on device, edge, and cloud

All-scenario applications
5,000+ applications online

All-scenario deployment
Free from cross-platform model conversion

Large-scale model training native support, accelerating innovation

Auto parallel
80% ↓
Parallel code

Auto tuning
60% ↓
Tuning time

MindSpore for science, accelerating AI for Science

MindSporeElec
Electromagnetic simulation kit

MindSpore SPONGE
Molecular simulation kit

200-billion-parameter model

Mobile phone EM simulation performance

10x ↑
Biopharmaceutical compound computing efficiency

50% ↑
MindSpore: Open Source All-Scenario AI Framework

Multi-Domain Expansion
Friendly Development
Full-stack collaborative optimization
All-scenario Deployment
Diversified Hardware

MindSpore: Open Source All-Scenario AI Framework

MindArmour

Model Zoo

Graph Neural Network
Reinforcement Learning
Deep Probabilistic Programming
Differential Equation

MindData

MindExpress

Cangjie Frontend
Offline Optimization Tools

MindCompiler

Type Derivation
Automatic Differentiation
Automatic Parallelism
Second-order Optimization
Quantization/Pruning/

Memory Optimization
Graph Kernel Fusion
Pipeline Execution

MindR

Hardware-related Optimization
MindAKG (Automatic operator generation)

MindRE (Distributed architecture for device-edge-cloud on-demand collaboration, deployment, communication and operation)

MindRT (Distributed DAG parallel execution)
MindRT Lite/Micro

DFX

DFX

Network Debugging
Performance Debugging
Precision Optimization

CANN
CUDA
Eigen
Android/LoT
Harmony

Ascend 310
Ascend 010
GPU
CPU
Android
IOS

CAPP Workshop, 29th July 2022
MindCompiler: the graph compiler of MindSpore

• In the AI software stack, the graph compiler optimizes the processing of a forward, or backward pass over the computation graphs that describe deep learning models.

• MindCompiler is the graph compiler of MindSpore to achieve three major kinds of optimizations, including
  • hardware-independent optimization (type inference, automatic differentiation, expression simplification, etc.);
  • hardware-related optimization (automatic parallelism, memory optimization, graph kernel fusion, pipeline execution, etc.);
  • deployment and inference-related optimizations (quantization, pruning, etc.).

• MindCompiler based on the unified device-cloud MindIR.
MindIR: A bit of motivation

- Compared other “general purpose” IRs, the IR of the graph compiler of AI framework has some special requirements and challenges:
  - Tensor representation;
  - Automatic differentiation;
  - JIT;
  - Implicit parallelism.

- Some hints from previous studies:
  - Sea of nodes: to construct graphs via use-def chains. The graph form is a single tiered structure instead of a two-tiered Control-Flow Graph (CFG) containing basic blocks (tier 1) of instructions (tier 2). control and data dependencies have the same form and implementation.
  - Thorin: a functional graph-based IR that abandons explicit scope nesting in favor of a dependency graph. The relationship between free variables and subgraphs is obtained via data dependency on the graph.
  - A-Normal form: partition expressions into two forms, atomic expressions and complex expression. All complex expression must be let-bound, or else appear in tail position.

\[
\begin{align*}
\langle \text{aexp} \rangle & ::= \text{NUMBER} | \text{STRING} | \text{VAR} | \text{BOOLEAN} | \text{PRIMOP} \\
& \quad | (\text{lambda} (\text{VAR} \ldots) \langle \text{aexp} \rangle) \\
\langle \text{aexp} \rangle & ::= (\langle \text{aexp} \rangle \langle \text{aexp} \rangle \ldots) \\
& \quad | (\text{if} \ \langle \text{aexp} \rangle \ \langle \text{aexp} \rangle \ \langle \text{aexp} \rangle) \\
\langle \text{exp} \rangle & ::= (\text{let} ([\text{VAR} \ \langle \text{cexp} \rangle]) \ \langle \text{exp} \rangle) \ | \ \langle \text{cexp} \rangle \ | \ \langle \text{aexp} \rangle
\end{align*}
\]
MindIR: a function-style graph-based IR

• MindSpore IR (MindIR) is a function-style IR based on graph representation.
  • Function-style IR makes automatic differentiation and implicit parallelism analysis more convenient;
  • Graph-based IR with a single tiered structure is suitable for JIT optimizations;
  • A-Normal form gets rid of scope and is easy to read and check.

• BNF: An variant A-Normal form used in MindIR with ‘if’ being treated as a prim operation
  ➢  <Anode> ::= Scalar | Named | Tensor | Var | Prim | MetaFunc | Func | Type | Shape | Param
  ➢  <CNode> ::= (<Anode> ...)
  ➢  <AnfNode> ::= <CNode> | <ANode>

• ANode in a MindIR corresponds to the atomic expression of ANF.
  • ValueNode refers to a constant node, which can carry a constant value (such as a scalar, symbol, tensor, type, and dimension), a primitive function (Primitive), or a common function (FuncGraph).
  • ParameterNode refers to a parameter node, which indicates the formal parameter of a function.

• CNode in a MindIR corresponds to the compound expression of ANF, indicating a function call.
MindIR: An example

Python Code

```python
def func(x, y):
    return x / y

def test_f(x, y):
    a = x - 1
    b = a + y
    c = b * func(a, b)
    return c
```

In a MindIR, a function graph indicates the definition of a common function.

Each expression is bound as a node, and the dependency is represented by using the directed edges between nodes.
In functional programming, the function definition itself is a value.

In a MindIR, a function, defined by a subgraph, can be transferred as the input or output of other higher-order functions.

Higher-order semantics greatly improve the flexibility and simplicity of MindSpore representations.
MindIR: Free Variables and Closures

- In a MindIR, a code block is represented as a function graph.
- The scope environment can be considered as the context where the function is called.
- The capture method of free variables is value copy instead of reference.
• In a MindIR, control flows are expressed in the form of high-order function selection and calling.
• This form transforms a control flow into a data flow of higher-order functions, making automatic differentiation of control flows possible.
MindIR: Autodiff

- MindIR implements automatic differentiation based source transformation.
- Each function call is transformed to return an additional value, which is a closure called the ‘backpropagator’.
  - The backpropagator computes the derivative with respect to the inputs given the derivatives with respect to the outputs;
  - The backpropagators of primitives are known;
  - The backpropagators of user-defined functions is constructed by calling the backpropagators of the function calls in the body in reverse order by the chain rule.
- The design of MindIR help the automatic differential algorithm works in the case of control flows such as conditional jumps, loops, and recursion.

```python
3     def f(x, y):
4         a = pow(x, 3)
5         b = pow(y, 4)
6         c = mul(a, b)
7         return c

9     def ▷f(▷x, ▷y):
10        ▷a, ▷a = ▷pow(▷x, ▷3)
11        ▷b, ▷b = ▷pow(▷y, ▷4)
12        ▷c, ▷c = ▷mul(▷a, ▷b)
13     def ▼f▼c:
14         ▼x, ▼y, ▼a, ▼b, ▼c = 0...
15         ▼mul_fv, ▼a, ▼b += ▼c▼c
16         ▼pow_fv, ▼y, ▼exp_4 += ▼b▼b
17         ▼pow_fv, ▼x, ▼exp_3 += ▼a▼a
18         return ▼, ▼x, ▼y
19         return ▷c, ▷f
22     dfdx = ▼f(1.0)[1] = ▷f(x, y)[1](1.0)[1]
```

MindIR: Autodiff

Perform auto diff transform on origin function according the rule of Generating $\nabla f$ and Generating $\Delta f$.
MindIR: Execution Process

- As the unified model file of MindSpore, MindIR stores network structures and weight parameter values. In addition, it can be deployed on the on-cloud Serving and the on-device Lite platforms to execute inference tasks.

- A MindIR file supports the deployment of multiple hardware forms.
  - On-cloud deployment and inference on Serving: After MindSpore trains and generates a MindIR model file, the file can be directly sent to MindSpore Serving for loading and inference.
  - On-device inference and deployment on Lite: MindIR can be directly used for Lite deployment.
Thank you