ICS 35.040
CCS L 71

团　　体　　标　　准

T/AI 132.2—2025

人工智能 神经网络编译器 第2部分：技术要求与测试方法

Artificial intelligence — Neural network compiler—Part 2: Technical requirements and test methods

2025 - 04 - 27 发布　　2025 - 04 - 27 实施

中关村视听产业技术创新联盟　　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是T/AI 132《人工智能 神经网络编译器》的第2部分，T/AI 132已经发布以下部分：

——第2部分：技术要求与测试方法。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由新一代人工智能产业技术创新战略联盟AI标准工作组提出。

本文件由中关村视听产业技术创新联盟归口。

本文件起草单位：浪潮电子信息产业股份有限公司、北京大学、北京百度网讯科技有限公司、北京大学长沙计算与数字经济研究院、中科寒武纪科技股份有限公司、上海燧原科技股份有限公司、华为技术有限公司。

本文件主要起草人：郭振华、赵雅倩、杨超、蒋晓琳、勾海鹏、胡帅、王思善、关贺、唐轶男、王丽、李仁刚。

引　言

T/AI 132《人工智能 神经网络编译器》旨在为人工智能上层深度学习框架和下层AI硬件加速设备之间的统一映射提供规范化指引和依据。拟由两个部分构成。

——第1部分：接口与优化技术规范。目的在于规范神经网络编译器前后端接口、中间表示接口以及编译器优化接口等内容。

——第2部分：技术要求与测试方法。目的在于规范神经网络编译器基础能力的技术要求和对应的测试方法。

人工智能 神经网络编译器 第2部分：技术要求与测试方法

1 范围

本文件规定了人工智能神经网络编译器的技术要求和测试方法。

本文件适用于神经网络编译器的设计参考与测试评估，为神经网络编译器技术发展提供参考规范和功能指引。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

计算图 computational graph

对一系列算子和数据流转编排之后形成的有向无环图的描述。

3.2

深度学习框架 deep learning framework

用于构建和训练神经网络模型的一套工具和库。

3.3

神经网络编译器 neural network compiler

输入深度学习框架模型定义文件，输出能够在不同硬件高效执行的代码的翻译器或解释器。

3.4

神经网络编译器前端 neural network compiler frontend

神经网络编译器的一部分，负责与硬件无关的处理，包括接收深度学习框架的模型输入、将计算图转换至统一的高层IR并进行硬件无关的优化、将高层IR送给神经网络编译器后端。

3.5

神经网络编译器后端 neural network compiler backend

神经网络编译器的一部分，负责将高层IR转化为低层IR、针对特定输出硬件执行低层IR的特定优化、生成对应的硬件代码指令、调用各硬件设备执行计算。

4 缩略语

下列缩略语适用于本文件。

AI 人工智能（Artificial Intelligence）

ASIC 专用集成电路（Application-Specific Integrated Circuit）

CPU 中央处理器（Central Processing Unit）

CUDA 统一计算设备架构（Compute Unified Device Architecture）

GPU 图形处理器（Graphics Processing Unit）

IR 中间表示（Intermediate Representation）

LLVM 底层虚拟机（Low Level Virtual Machine）

5 概述

神经网络编译器的通用设计架构主要包含两部分：编译器前端和编译器后端。在部署深度学习算法模型的训练和推理时，模型可以被深度学习框架表示为计算图，计算图在神经网络编译器中被转换为多级IR，其中高层IR在编译器前端，低层IR在编译器后端。

a) 编译前端主要负责与硬件无关的处理，其主要功能是接收来自不同深度学习框架的模型输入，并将计算图转换至统一的高层IR，之后对高层IR进行硬件无关的优化，再将其送给编译后端；
b) 编译后端的主要功能是将优化后的高层IR转化为低层IR，然后针对特定输出硬件执行低层IR的特定优化，并生成对应的硬件代码指令，最后通过不同的硬件接入方式调用各硬件设备执行计算。
c) 神经网络编译器通用架构图见图1，其描述了神经网络编译器在编译执行过程中所处的位置与作用。本文件对图1中神经网络编译器（黑色实现框内）的内容做出规定，不涉及其他内容。

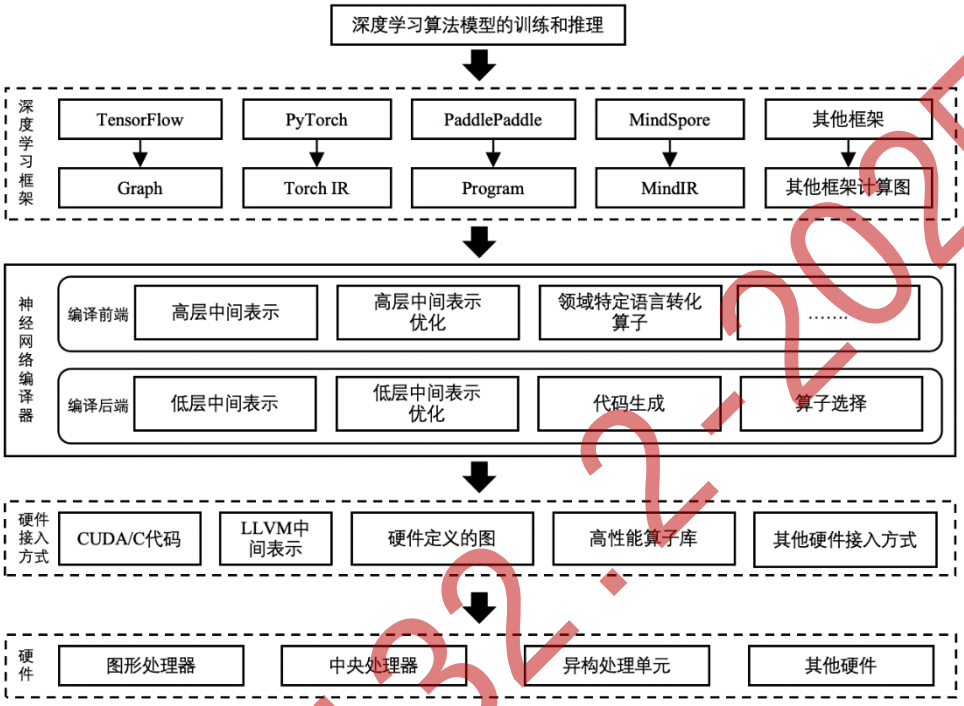

图1 神经网络编译器通用架构图

6 技术要求

6.1 一般要求

神经网络编译器一般要求如下：
a) 应支持主流的深度学习框架算法模型到后端AI硬件设备的编译映射；
b) 系统从功能设计上应符合高可靠性（编译运行时准确生成中间表示语言，算法功能正确）、统一性（统一的框架接口和算子接口）、可扩展性（从深度学习框架支持、高层算子映射、低层算子接口、硬件设备支持等支持灵活部署和弹性扩展）等核心要求；
c) 提供相关开发文档支持。

6.2 安装部署要求

神经网络编译器安装部署要求如下：
a) 应不依赖于各深度学习框架，可独立安装部署；
b) 应支持命令行安装和源码编译安装，并提供神经网络编译器的命令行安装和源码编译安装的安装指导说明文档；
c) 应提供对应不同软件版本的已经编译好的神经网络编译器安装包；
d) 应支持基于容器的安装部署。

6.3 功能要求

6.3.1 编译前端要求

神经网络编译器前端的要求如下：

a） 应提供高层中间表示；
b） 应支持不少于 3 种深度学习框架，能够实现不少于 3 种主流深度学习框架（如 Tensorflow、Pytorch、Paddlepaddle 等）算法模型计算任务到后端 AI 硬件设备的映射；
c） 高层中间表示应提供统一的算子接口，算子定义与硬件设备无关；
d） 应支持常规图层级编译优化技术，如算子消除、算子融合、内存分配优化等。

6.3.2 编译后端要求

神经网络编译器后端应：

a） 支持市场通用 CPU、GPU 硬件计算设备；
b） 支持包括 GPU 在内的不少于 3 种不同架构的 AI 硬件加速设备（如 GPU、ASIC、FPGA（现场可编程逻辑门阵列）等）；
c） 支持每个硬件设备的后端算子实现；
d） 支持自动生成算子代码和手工编写算子代码；
e） 支持调用硬件设备专门优化的算子计算库（如 cuDNN 等）；
f） 提供后端算子代码选择策略；
g） 支持常见的算子层优化技术，比如算子自动调度优化，循环优化，缓存优化等。

6.3.3 模型训练和推理要求

神经网络编译器对深度学习算法模型的训练和推理要求如下：

a） 应支持主流深度学习应用场景（CV（计算机视觉）、NLP（自然语言处理）等）算法任务到编译器后端硬件设备的映射；
b） 应支持多种类型深度学习算法（如卷积神经网络、递归神经网络等）；
c） 应提供各深度学习框架对应的推理任务深度学习算法模型库；
d） 应支持深度学习算法模型推理任务在 GPU、ASIC 等 AI 硬件设备上的分布式执行；
e） 宜支持深度学习算法模型在 GPU、ASIC 等 AI 硬件设备上的分布式训练。

6.4 兼容性要求

6.4.1 软件兼容性要求

神经网络编译器软件兼容性要求如下：

a） 应支持兼容市场主流 Linux 操作系统（如 CentOS，Ubuntu）；
b） 应支持系统数据的选择、提取、构建。

6.4.2 硬件兼容性要求

神经网络编译器硬件兼容性要求如下：

a） 应支持不少于 3 种不同型号 AI 硬件设备的适配；
b） 应支持不少于 3 种不同型号的服务器。

6.5 可扩展性要求

神经网络编译器可扩展性要求如下：

a） 应支持编译前端深度学习框架扩展，能够提供新的深度学习框架注册接口和注册说明；
b） 应支持编译后端 AI 硬件设备扩展，提供硬件设备注册添加接口和注册说明；
c） 应支持编译器高层 IR 的高层次算子扩展，提供高层 IR 的算子注册接口和注册添加说明；
d） 应支持编辑器低层 IR 的低层次算子扩展，提供低层 IR 的算子注册接口和注册添加说明。

7 测试方法

7.1 测试要求

测试要求如下：

a) 在服务器操作系统上安装部署不少于 3 种主流深度学习框架，并确保正确安装；
b) 在服务器内安装不少于 3 种不同架构的 AI 硬件加速设备，并安装相应的驱动程序，并确保操作系统可正确识别这些硬件设备；
c) 将编译器源码与编译器安装程序、或者支持编译器运行的容器镜像放置于操作系统可识别路径上，准备进行神经网络编译器的测试。

7.2 安装部署测试

神经网络编译器安装部署测试方法如下：
a) 测试方法：按照提供的安装说明文档，分别尝试源码编译安装和命令行安装部署编译器，验证是否支持源码编译安装和命令行安装；
b) 预期结果：编译器源码编译安装成功，编译器命令行安装部署成功。

7.3 功能测试

7.3.1 编译前端测试

神经网络编译器前端测试方法如下：
a) 测试方法：分别尝试运行不同深度学习框架的深度学习算法模型，验证深度学习算法任务是否映射到神经网络编译器的后端硬件设备，并正确执行计算；
b) 预期结果：能够支持不少于 3 种深度学习框架对应的深度学习算法模型计算任务到神经网络编译器后端 AI 硬件设备的映射，并正确执行计算。

7.3.2 编译后端测试

神经网络编译器后端测试方法如下：
a) 测试方法：在编译器前端撰写自定义测试实例，并依次尝试指定 3 种不同的后端设备，验证实例是否正常编译运行；
b) 预期结果：能在不少于 3 种不同的 AI 硬件加速设备上正确运行自定义测试实例。

7.3.3 模型训练和推理测试

7.3.3.1 训练任务测试

训练任务测试方法如下：
a) 测试方法：部署编译器分布式训练环境，选择支持的深度学习算法模型，验证是否支持该网络模型的分布式训练；
b) 预期结果：能够实现 AI 分布式训练任务到编译器后端硬件设备的映射，各设备正确执行训练过程。

7.3.3.2 推理任务测试

推理任务测试方法如下：
a) 测试方法：从模型库中选择支持的深度学习算法模型，尝试指定不同的后端设备执行推理计算；
b) 预期结果：能够实现深度学习算法模型推理任务在 GPU、ASIC 等 AI 硬件设备上的分布式计算正确执行。

7.4 兼容性测试

7.4.1 软件平台支持测试

神经网络编译器软件平台支持测试方法如下：
a) 测试方法：分别尝试在 2 种不同的 Linux 系统中安装部署编译器；
b) 预期结果：支持在 2 种 Linux 系统安装部署。

7.4.2 硬件平台支持测试

神经网络编译器硬件平台支持测试方法如下：

a) 测试方法：分别尝试基于 3 种不同型号的 AI 硬件设备执行编译器自定义测试实例，验证测试实例是否在不同型号的 AI 硬件设备上正确执行；
b) 预期结果：能够支持不少于 3 种不同型号 CPU、加速器硬件产品的适配，测试实例可正确执行。

7.5 可扩展性测试

7.5.1 编译前端测试

神经网络编译器前端测试方法如下：
a) 测试方法：按照说明文档，尝试在编译器前端添加新的深度学习框架支持接口；
b) 预期结果：深度学习框架的扩展添加成功。

7.5.2 编译后端测试

神经网络编译器后端测试方法如下：
a) 测试方法：按照说明文档，尝试在编译器后端注册添加新的 AI 硬件设备支持接口，验证新添加设备是否可用；
b) 预期结果：后端 AI 硬件设备扩展成功，且新添加设备可用。

7.5.3 编译器高层 IR 测试

神经网络编译器高层 IR 测试方法如下：
a) 测试方法：按照说明文档，尝试在编译器高层中间表示层添加注册新的自定义算子，验证新的算子是否可用；
b) 预期结果：用户自定义高层次算子添加成功且可使用。

7.5.4 编译器低层 IR 测试

神经网络编译器低层IR测试方法如下：
a) 测试方法：按照说明文档，尝试在编译器低层中间表示层添加注册新的低级别算子，验证新的算子是否可用；
b) 预期结果：后端低层次算子添加成功且可用。