

团体标准

T/AII 008—2023

深度学习算法框架通用接口规范

General interface specification for deep learning algorithm frameworks

2023 - 12 - 26 发布

2023 - 12 - 26 实施

深圳市人工智能行业协会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 深度学习算法框架通用接口模型	3
5.1 算法接口目标	3
5.2 算法接口模型	3
6 深度学习算法框架通用接口要求	6
6.1 框架适配层	6
6.2 核心算法层	6
6.3 应用层	10
附录 A（规范性） 基础接口	12
A.1 张量 Tensor	12
A.2 模块 Module	13
附录 B（规范性） 接口参数表	16
B.1 数据读取	16
B.2 采样器	16
B.3 优化器	18
B.4 训练器	19
B.5 推理器	20
B.6 评估器	21
B.7 数据操作	22
参考文献	24
图 1 系统结构标准说明图	4
图 2 总体接口图	5
表 A.1 initialize 接口参数列表	12
表 A.2 index 接口参数列表	12
表 A.3 permute 接口参数列表	13

表 A.4	reshape 接口参数列表	13
表 A.5	forward 接口参数列表	14
表 A.6	get_parameters 接口参数列表	14
表 A.7	get_modules 接口参数列表	14
表 A.8	load_state_dict 接口参数列表	14
表 A.9	backward 接口参数列表	15
表 B.1	construct_dataset 接口参数列表	16
表 B.2	get_sample 接口参数列表	16
表 B.3	construct_sampler 接口参数列表	17
表 B.4	iterate_sample 接口参数列表	17
表 B.5	iterate_batch 接口参数列表	17
表 B.6	construct_optimizer 接口参数列表	18
表 B.7	update_learnable_parameter 接口参数列表	18
表 B.8	zero_gradient 接口参数列表	19
表 B.9	construct_trainer 接口参数列表	19
表 B.10	train_one_batch 接口参数列表	19
表 B.11	train_one_epoch 接口参数列表	20
表 B.12	construct_inference 接口参数列表	20
表 B.13	infer_one_sample 接口参数列表	21
表 B.14	infer_samples 接口参数列表	21
表 B.15	construct_evalvater 接口参数列表	22
表 B.16	evalvate 接口参数列表	22
表 B.17	construct_operator 接口参数列表	22
表 B.18	operate 接口参数列表	23

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。

本文件由深圳市人工智能行业协会（AII）提出并归口。

本文件起草单位：博瀚智能（深圳）有限公司、北京大学、深圳云天励飞技术股份有限公司、达闼机器人股份有限公司、深圳市酷开网络科技股份有限公司、麒麟合盛网络技术股份有限公司、软通智慧科技有限公司、上海致宇信息技术有限公司、深思考人工智能机器人科技（北京）有限公司、北京智境云创科技有限公司、深圳优地科技有限公司、深圳云译科技有限公司、中电金信软件（深圳）有限公司、深圳云知声信息技术有限公司、深圳市木愚科技有限公司、上海思岚科技有限公司、广州九四智能科技有限公司、深圳市华赛睿飞智能科技有限公司、静观照（深圳）科技有限公司、深圳市易行网数字科技有限公司、新译信息科技（北京）有限公司、超参数科技（深圳）有限公司、深圳玩智商科技有限公司、边无际（北京）科技有限公司、深圳市即构科技有限公司。

本文件主要起草人：陈鹏、王忠强、许佳、董豪、刘奕良、张衡、饶雪、张站朝、张伟、李晓榕、吴旭、张旭、邓小波、齐全、郭文鹏、陈思成、邹伟军、杨志明、王雪艳、王智武、罗沛、夏舸、刘开军、郭煜、梅华平、康恒、卢庆华、张丽萍、刘嗣平、汤丁青、林必毅、封其国、周艺恺、曾鹏、陈丹霞、田亮、刘劲松、刘永升、罗敏、王品、何英、陈永立、何建新、李凯。

引 言

随着深度学习应用越来越广泛，深度学习框架、计算硬件、操作系统、应用种类也呈现多样化趋势。其中，常用的深度学习框架有TensorFlow、PyTorch、昇思（MindSpore）、飞桨（PaddlePaddle）等，由于多任务、跨算法、不同框架之间的代码不通用，且用户学习多个框架的成本较高。因此，跨框架、跨平台的开发需求愈发强烈。根据跨框架、跨平台接口发展需求，博瀚智能（深圳）有限公司联合相关单位制定了《深度学习算法框架通用接口规范》。

本文件旨在为深度学习算法框架提供统一接口的标准，规范了面向多任务、跨算法框架通用应用接口目标、接口模型以及接口库的技术要求等。

深度学习算法框架通用接口规范

1 范围

本文件确立了面向深度学习算法框架通用应用接口模型、规定了接口功能及技术要求。
本文件适用于基于深度学习的算法开发和应用过程中所涉及应用编程接口的适配与调用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO/IEC 22989:2022 信息技术 人工智能 概念和术语 (Information technology — Artificial intelligence — Artificial intelligence concepts and terminology)

3 术语和定义

ISO/IEC 22989:2022界定的术语和定义适用于本文件。

3.1

深度学习 **deep learning**

通过训练具有多个隐藏层的神经网络来创建丰富的层次表示的方法。

[来源:ISO/IEC 22989:2022, 3.4.4]

3.2

模型训练 **model training**

基于机器学习算法，通过使用训练数据来确定或改进机器学习模型参数的过程。

[来源:ISO/IEC 22989:2022, 3.3.15]

3.3

训练数据 **training data**

用于训练机器学习模型的数据。

[来源:ISO/IEC 22989:2022, 3.3.16]

4 缩略语

下列缩略语适用于本文件。

ASR: 自动语音识别 (Automatic Speech Recognition)

BERT: 双向语言表征注意力模型 (Bidirectional Encoder Representation from Transformers)

ByteNet: 一种基于CNN的机器翻译模型 (Neural Machine Translation based on CNN)

CNN: 卷积神经网络 (Convolutional Neural Networks)

Conformer: 一种基于卷积增强的语音识别模型 (Convolution-augmented Transformer for Speech Recognition)

DeepLabv3: 一种基于空洞可分离卷积、编解码架构的语义分割模型 (Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation)

EfficientNet: 一种高效的卷积网络模型 (Efficient Net)

Faster RCNN: 一种基于区域建议网络的实时目标检测模型 (Towards Real-Time Object Detection with Region Proposal Networks)

HRNet: 一种高分辨率的人体姿态评估的深度模型 (Deep High-Resolution Representation Learning for Human Pose Estimation)

Mask RCNN: 一种图像实例分割模型

OCR: 光学字符识别 (Optical Character Recognition)

PFLD: 实用人脸关键点检测器 (A Practical Facial Landmark Detector)

ResNet: 残差网络 (Residual Network)

Resnet3D: 基于三维卷积的残差网络 (Residual Network with 3D CNN)

RetinaNet: 一种目标检测模型

RNNT: 一种用于语音识别的RNN变换模型 (Rnn-Transducer)

S3D: 可分离三维卷积网络 (Separable 3D CNN)

SSD: 单步多框检测器 (Single Shot MultiBox Detector)

T5: 文本到文本转换注意力模型 (Text-To-Text Transfer Transformer)

TextCNN: 一种用于文本分类的卷积网络模型 (Convolutional Neural Networks for Sentence Classification)

Transformer: 基于注意力机制的神经网络 (Neural Network based on Attention Mechanism)

TrOCR: 基于注意力的光学字符检测 (Transformer-based Optical Character Recognition)

TTS: 语音合成 (Text-to-Speech)

UNet: 一种用于生物医药领域图像分割的卷积网络 (Convolutional Networks for Biomedical Image Segmentation)

VGGNet: 由VGG组提出的图像分类网络模型 (Visual Geometry Group Network)

WaveNet: 一种用于生成原始音频波形的神经网络模型 (A Generative Model for Raw Audio)

XLNet: 一种自回归预训练语言模型 (Generalized Autoregressive Pretraining for Language Understanding)

YOLO: YOLO单阶段目标检测器 (You Only Look Once)

5 深度学习算法框架通用接口模型

5.1 算法接口目标

通过统一接口适配层，可实现下列目标。

- a) 算法支持使用不同的深度学习框架作为后端。
- b) 算法支持视觉、音频、文本等不同数据处理的深度学习任务。
- c) 在不同的硬件如服务器或分布式集群上正常运行。
- d) 算法支持不同数据的读入。

5.2 算法接口模型

从系统资源、框架适配、核心算法、应用四个层面描述了框架、数据、算法、模型之间的关系。图1给出了系统结构说明图，本文件主要规定的是框架适配、核心算法、应用三个层的技术要求。

- a) 系统资源层：包括硬件和软件两部分，提供计算机视觉系统必须的存储、计算和推理等功能。系统资源层的具体实现各个厂商有自己的方式，对此本文件不做进一步描述和规定。
- b) 框架适配层：使用统一的框架接口为上层系统过程提供算子、函数接口、优化接口、模型接口等服务接口，确保适配不同的深度学习框架。
- c) 核心算法层：包含数据、模型、训练、推理等基类定义，以及各领域的经典模型和算法，并确定数据、算法、流程等方面的基础协作逻辑。
- d) 应用层：完成图像、文本、音频等各领域、各任务的模型训练、模型推理等过程，在模型训练过程中需要调用本文件定义的各类接口。

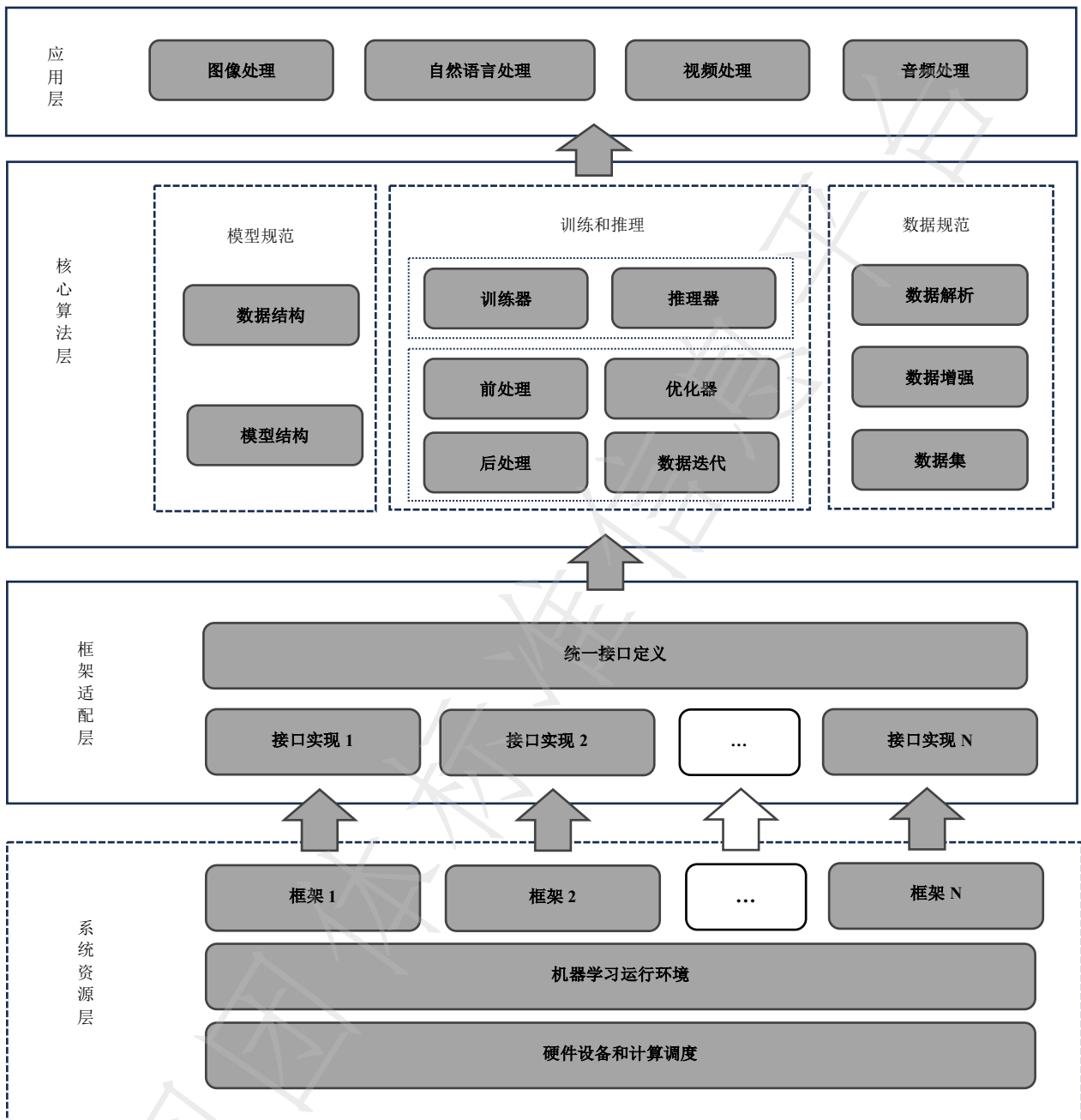


图 1 系统结构标准说明图

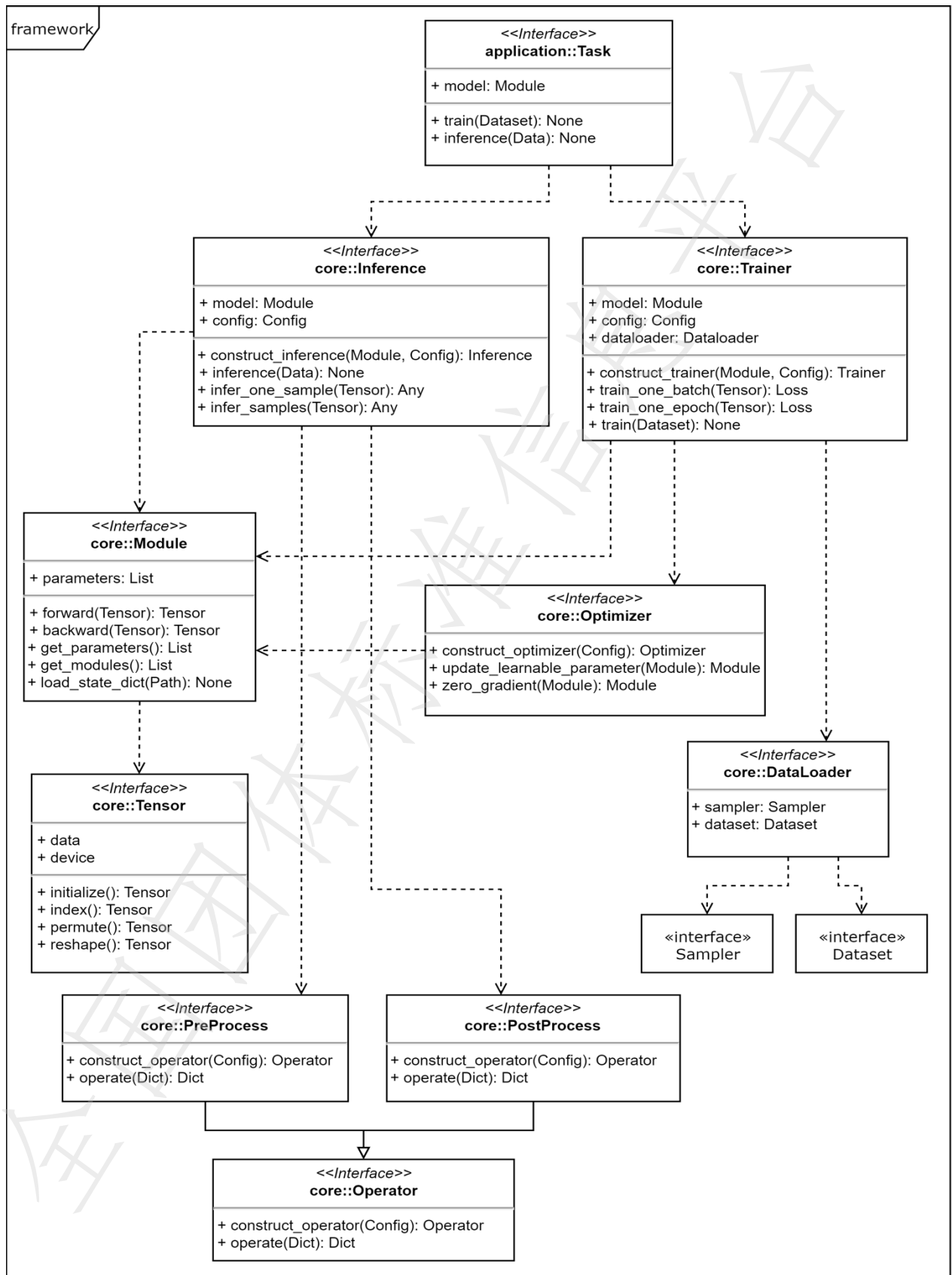


图 2 总体接口图

6 深度学习算法框架通用接口要求

6.1 框架适配层

6.1.1 统一接口规范

统一接口对各个框架进行统合，掩盖多框架之间的差异性。统一接口应具备下列核心功能。

- a) 支持各种张量初始化函数，如：
 - 1) 空值填充函数：即不做数值初始化；
 - 2) 指定值填充函数：零值填充、“1”值填充、输入数值填充、输入张量填充；
 - 3) 随机值填充函数：随机生成指定分布（平均分布、二项分布、高斯分布等）类型的数据，并作填充。
- b) 支持各种数学处理的基础函数，如：
 - 1) 逻辑函数：逻辑与、逻辑或、逻辑非、逻辑异或等；
 - 2) 三角函数：正弦函数、余弦函数、正切函数等；
 - 3) 反三角函数：反正弦函数、反余弦函数、反正切函数等；
 - 4) 双曲函数：双曲正弦、双曲余弦、双曲正切等；
 - 5) 比较函数：大于、小于、等于、不大于、不小于、不等于等；
 - 6) 平方函数、开方函数、指数函数、对数函数等。
- c) 支持各种张量处理的常用函数，如：
 - 1) 基础信息获取：张量形状、数据类型、设备信息等；
 - 2) 类型转换函数：基础数据类型转换、设备类型转换等；
 - 3) 聚合函数：求最大值、求最小值、求和、求平均；
 - 4) 激活函数：sigmoid、relu、tanh等；
 - 5) 其他函数：矩阵乘法、变形函数、拼接函数、堆叠函数、分片函数等。
- d) 支持各种常用模块和类：
 - 1) 常用激活模块：Sigmoid、Relu、Tanh等；
 - 2) 常用网络主体模块：多维CNN层、全连接层、RNN层等；
 - 3) 其他常用模块：Dropout层、BatchNorm层、最大/平均池化层、全局池化层等。
- e) 支持常用深度学习框架到统一接口的适配：pytorch、tensorflow、paddlepaddle、mindspore等。
- f) 支持后端框架的切换，如在启动时采用某个框架作为后台。

6.1.2 各框架接口实现

多个框架对统一接口的实现应当保持一致性。

6.2 核心算法层

6.2.1 模型规范

6.2.1.1 数据结构

数据结构作为计算机视觉的基本数据结构（Tensor），能够支持基本数据结构直接的计算，模型结构的输入输出均为该基本数据结构或该基本数据结构的集合。数据结构方面应具备下列核心功能。

- a) 支持常用数据形式：标量、向量、二维矩阵、多维矩阵。

- b) 支持该数据结构的初始化, 给定对应内容, 使用内容对该数据结构进行初始化。
- c) 支持随机初始化, 使用随机数对于该数据结构进行初始化, 例如: rand: 0~1之间的均匀采样; randint: 给定开始结束数字, 在其中之一均匀采样; randn: 均值0, 方差1的正态分布随机采样。
- d) 支持索引功能, 给定相关的维度索引, 给出具体的元素内容。
- e) 支持切片功能, 给定相关的维度索引, 遍历其他维度索引, 给出切片内容, 其结构仍为Tensor。
- f) 支持维度变换功能reshape, 给定新的维度与长度, 其元素总和与原元素总和一致, 将Tensor内元素按照新的维度与维度长度排布。
- g) 支持数据元素基本数学操作, 可进行一元数学基本操作, 例如取负、开根号、平方等。若两个Tensor维度与元素数量一致, 可进行+、-、*、/等二元基本数学操作。

基础接口见附录A。

6.2.1.2 模型结构

通过模型结构定义Module, 框架应该能够支持算法模型的定义。模型结构应具备下列核心功能。

- a) 支持前向计算, 模型执行一次前项计算, 通过输入经过模型计算得出输出。
- b) 支持反向计算, 模型执行一次反向计算, 通过输出的梯度反向计算出输入和权重的梯度。
- c) 支持获取可训练参数, 获取模型的可训练参数, 即权重。
- d) 支持获取训练状态, 获取模型的训练状态, 用于暂停训练时, 进行状态存储。
- e) 支持加载训练状态, 加载模型的训练状态, 用于恢复训练时, 进行状态回复。
- f) 支持获取子模块, 通过模块名称, 获取对应的子模块, 进行子模块操作。
- g) 支持下列模型结构定义。

1) 图像处理:

- 图像分类: ResNet、VGGNet、EfficientNet等;
- 目标检测: YOLO、SSD、Faster RCNN、RetinaNet等;
- 关键点检测: HRNet、PFLD等;
- 图像分割: DeepLabv3、UNet、Mask RCNN等;
- 光学字符识别: TrOCR等;
- 其他。

2) 自然语言处理:

- 文本相似度: BERT等;
- 文本分类: TextCNN、T5、BERT、XLNet等;
- 机器翻译: ByteNet、T5等;
- 文本生成: BERT等;
- 其他。

3) 音频处理:

- 自动语音识别: Conformer、RNNT等;
- 语音合成: WaveNet等;
- 其他。

4) 视频处理:

- 视频分类: S3D、Resnet3D等;
- 其他。

5) 其他模型。

基础接口见附录A。

6.2.2 数据规范

6.2.2.1 数据解析

通过数据解析接口，支持读取数据文件，并为数据转换做准备。数据解析接口应具备下列核心功能。

- a) 支持多种文本数据文件格式的读入和解析，如txt、json、xml等。
- b) 支持多种图像数据文件格式的读入和解析，如jpg、png、bmp等。
- c) 支持多种音频数据文件格式的读入和H.264/H.265/AV1格式的视频解码，如MP3、wmv、wav等。
- d) 支持多种视频数据文件格式的读入和解析，如MP4、mov等。
- e) 支持对数据进行基本校验等。

6.2.2.2 数据增强

为了提高模型对新数据的适应能力，通过数据增强接口对输入数据进行转换和增强，以此增加数据集的多样性。数据增强接口应采用基类与多个派生子类的设计模式，具备以下核心功能。

- a) 支持多种数据转换操作，例如：
 - 1) 图像数据，包括但不限于缩放、翻转、颜色扰动（如亮度、对比度调整）；
 - 2) 文本数据，例如分词、语法扰动；
 - 3) 音频数据，如变速、变调、添加背景噪声；
 - 4) 通用数据，归一化、特征缩放等。
 每种操作都应详细说明其对数据的具体影响，并提供示例代码。
- b) 支持数据和标签的同步转换，确保在转换过程中数据和标签保持一致，例如，当一张图像和另外一张图像进行了融合增强，相应的标签也应相应地更新。
- c) 支持对增强方法列表进行随机选择、排列和组合，以生成多样化的数据增强序列；接口应允许用户指定随机种子以确保结果的可重复性，并能够轻松地扩展新的数据增强方法。

通过这些功能，数据增强接口将为模型训练提供强大的支持，帮助提升模型在实际应用中的表现和鲁棒性。

接口参数见附录B。

6.2.2.3 数据集

通过数据集接口，支持将数据转换成模块要求的格式，例如张量。数据接口应具备外层接口和内层接口。框架应能够以迭代的形式为算法训练准备持续的训练数据。该接口应具备下列核心功能。

- a) 支持根据数据集路径以及数据集相关的参数构造数据集对象。
- b) 支持获取数据集中元素数量。
- c) 支持从存储设备或服务上读取数据集的部分或全部数据，例如标注文件、数据样本等。
- d) 支持对数据进行预处理操作，例如图像缩放、翻转、颜色扰动等。

接口参数见附录B。

6.2.3 训练和推理

6.2.3.1 优化器

优化器接口应支持网络模型优化的功能，根据不同的优化算法例如SGD、Adam、Momentum等，对模型参数进行优化更新。具体来说，应实现梯度计算和回传、参数更新、优化算法、学习率更新等功能。优化器接口应支持下列功能。

- a) 实现为一个类，其构造参数包括网络模型或者模型参数列表，以及所需的其他参数，例如学习率；这个类应该能够根据输入的模型和参数，选择和实施适当的流程和算法。

- b) 实现step()函数，用于进行一次参数优化更新；该函数的输入应该是当前的模型参数和梯度，输出应该是更新后的模型参数；当这个函数被调用之后，模型参数应根据输入的梯度进行更新。

接口参数见附录B。

6.2.3.2 训练接口

训练接口提供模型训练功能，接口应符合下列要求。

- a) 能组合数据、模型、优化器等关键模块。
- b) 支持前处理、模型推理、后处理的基础形态的过程，并允许子类继承和修改。
- c) 支持基本训练、评估过程。
- d) 支持模型的序列化和反序列化。
- e) 支持额外的前处理、后处理步骤。
- f) 支持下列类型模型的训练。
 - 1) 图像处理：
 - 图像分类：ResNet、VGGNet、EfficientNet等；
 - 目标检测：YOLO、SSD、Faster R-CNN、RetinaNet等；
 - 关键点检测：HRNet、PFLD等；
 - 图像分割：DeepLabv3、U-Net、Mask R-CNN等；
 - 光学字符识别：TrOCR等；
 - 其他。
 - 2) 自然语言处理：
 - 文本相似度：BERT等；
 - 文本分类：TextCNN、T5、BERT、XLNet等；
 - 机器翻译：ByteNet、T5等；
 - 文本生成：BERT等；
 - 其他。
 - 3) 音频处理：
 - 自动语音识别：Conformer、RNN-T等；
 - 语音合成：WaveNet等；
 - 其他。
 - 4) 视频处理：
 - 视频分类：S3D、Resnet3D等；
 - 其他。
 - 5) 其他模型。

接口参数见附录B。

6.2.3.3 推理接口

推理接口提供模型推理功能，接口应符合下列要求。

- a) 支持模型加载功能。
- b) 支持前处理、模型推理、后处理的基础形态的过程，并允许子类继承和修改。
- c) 支持批量数据推理过程。
- d) 支持额外的前处理、后处理步骤。

接口参数见附录B。

6.2.3.4 前处理

前处理接口表现为数据增强的组合，提供模型训练功能，接口应符合下列要求。

- a) 支持图像、音频、文本等类型的数据处理。
- b) 支持同时协同处理多种类型数据。
- c) 支持处理后的数据转换为Tensor类型。

6.2.3.5 后处理

不同的任务类型，其后处理过程往往不同。后处理接口应表现为一个基类和多个子类，提供将模型输出解析和转换到有语言意义的数据结构，接口应符合下列要求。

- a) 支持处理Tensor类型的数据。
- b) 支持图像分类、目标检测、图像分割、音频及文本任务的输出处理。

6.2.3.6 数据迭代器

通过数据接口，支持将数据转换成模块要求的格式，例如张量。数据接口应具备外层接口和内层接口。框架应能够以迭代的形式为算法训练准备持续的训练数据。该接口应具备下列核心功能。

- a) 支持实现一个可迭代的数据加载器类型，每次迭代返回值均为一个数据批。
- b) 支持根据训练要求从数据集中采样数据。

6.3 应用层

6.3.1 图像处理

图像处理应支持下列领域图像模型的基本训练和推理功能。

- a) 图像分类和识别。
- b) 目标检测。
- c) 图像分割和实例分割。
- d) 关键点检测，包含人脸关键点检测和人体姿态检测。
- e) 光学字符识别，包含字符分割和字符识别。
- f) 其他。

6.3.2 自然语言处理

自然语言处理应支持下列语言模型的基本训练和推理功能。

- a) 文本相似度计算。
- b) 文本分类。
- c) 机器翻译。
- d) 文本生成。
- e) 其他。

6.3.3 音频处理

音频处理应支持下列语音模型的基本训练和推理功能。

- a) 自动语音识别，即语音转文本。
- b) 语音合成，即文本转语音。
- c) 其他。

6.3.4 视频处理

视频处理应支持下列视频模型的基本训练和推理功能。

- a) 视频分类。
- b) 其他。

6.3.5 其它

可支持其他模态、用途、架构的模型。

全国团体标准信息平台

附录 A
(规范性)
基础接口

A.1 张量Tensor

A.1.1 内涵

Tensor是深度学习框架中所有数据的基础结构，所有操作传递的数据均为Tensor。

A.1.2 功能

Tensor数据结构为一个多维矩阵，每个Tensor应包括数据类型（Type）、维度（Rank or Dimension）和形状（Shape）。

A.1.2.1 初始化

接口名称：initialize。

接口功能描述：对Tensor进行初始化。

接口参数列表见表A.1。

表 A.1 initialize 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	data	数据列表，张量形状将根据数据层数和长度自动确认	必选
输入	type	类型(Type)主要包括 float32、float64 (double)、int64、int32、int16、int8、uint8、string、bool、complex64	可选
输出	tensor	返回 Tensor 基础类型	必选

异常处理：无。

附加说明：无。

A.1.2.2 切片索引

接口名称：index。

接口功能描述：通过切片的方式获取tensor中部分数据。

接口参数列表见表A.2。

表 A.2 index 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	tensor	被索引张量	必选
输入	index	Tensor[i, j, ...]表示维度 0 取第 i 元素，维度 1 取第 j 元素	可选
输出	tensor	返回 Tensor 的切片	必选

异常处理：无。

附加说明：无。

A.1.2.3 维度变换

接口名称：permute。

接口功能描述：通过变换维度顺序改变Tensor形状。

接口参数列表见表A.3。

表 A.3 permute 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	tensor	待切片张量	必选
输入	dims	Tensor 的维度顺序列表	必选
输出	tensor	返回维度顺序改变后的 Tensor	必选

异常处理：无。

附加说明：无。

A.1.2.4 形状变换

接口名称：reshape。

接口功能描述：不改变维度顺序而改变Tensor形状。

接口参数列表见表A.4。

表 A.4 reshape 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	tensor	待切片张量	必选
输入	shape	Tensor 的新形状，形状改变前后体积不变	必选
输出	tensor	返回形状改变后的 Tensor	必选

异常处理：无。

附加说明：无。

A.2 模块Module

A.2.1 内涵

Module是神经网络中的基本模块，神经网络模块基于该基础类进行图的构建。模块可以包含一个简单的计算，也可以是神经网络的一层，还可以是多层堆叠或组装而成的、具备完整功能的复杂网络模型。模型的类型一般为Module。构建模型需要做下列工作。

- 定义模型类，该子类继承Module基类。
- 初始化该子类中基类的部分。
- 定义和初始化模型的各个子部分。
- 如需，定义并覆盖基类中的计算图，决定子部分的执行流程；该步骤通常通过forward函数来实现。
- backward反向传播函数不需要显示声明，由计算图自动生成。

A.2.2 功能

进行前向计算和反向计算，并获取过程中的参数。

A.2.2.1 模块前向计算

接口名称：forward。

接口功能描述：模块执行一次前向计算并返回模块的计算结果。如果是训练状态，则在前向计算过程构造计算图以用于计算模块参数的梯度。

接口参数列表见表A. 5.

表 A. 5 forward 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	input	模块的输入，类型为元组或键值对	必选
输入	其他参数	键值对参数，类型不限	可选
输出	tensor	模块的输出，类型为张量	必选

异常处理：无。

附加说明：无。

A. 2. 2. 2 获取模块可训练参数

接口名称：get_parameters。

接口功能描述：返回模块的可训练参数。

接口参数列表见表A. 6。

表 A. 6 get_parameters 接口参数列表

参数类型	参数名称	参数说明	是否可选
输出	parameters	模块的可训练参数，类型为迭代器	必选

异常处理：无。

附加说明：无。

A. 2. 2. 3 获取模块及其子模块

接口名称：get_modules。

接口功能描述：可选接口。该接口返回一个迭代器，迭代返回模块本身以及模块的子模块，重复的模块只返回一次。

接口参数列表见表A. 7。

表 A. 7 get_modules 接口参数列表

参数类型	参数名称	参数说明	是否可选
输出	module	包含可训练参数的模块	必选

A. 2. 2. 4 加载模块状态

接口名称：load_state_dict。

接口功能描述：加载模块的状态，包括模块的参数以及缓冲。

接口参数列表见表A. 8。

表 A. 8 load_state_dict 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	state_dict	模块的参数字典	必选

异常处理，键值对不一致错误：state_dict键值对的名字不能和模块的参数名完全一致。

附加说明：无。

A.2.2.5 模块反向计算

接口名称：backward (grad_input, grad_output)。

接口功能描述：模块执行一次反向计算并返回模块的计算结果。如果是训练状态，则在反向计算过程计算模块参数的梯度。该函数为计算图自动生成，也可通过后续register_backward_function进行注册。

接口参数列表见表A.9。

表 A.9 backward 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	grad_input	模块的输入的梯度，类型为张量	必选
输出	grad_output	模块的输出的梯度，类型为张量	必选

异常处理：无。

附加说明：无。

附录 B
(规范性)
接口参数表

B.1 数据读取

B.1.1 数据集构造器

接口名称: `construct_dataset`。

功能描述: 根据数据集路径以及数据集相关的参数构造数据集。

接口参数列表见表B.1。

表 B.1 `construct_dataset` 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	<code>path</code>	数据集路径, 类型为字符串	必选
输入	其他参数	数据集相关的其他参数	可选
输出	<code>dataset</code>	数据集对象	必选

异常处理, 文件不存在错误: 数据集路径不存在。

附加说明: 无。

B.1.2 读取数据样本

接口名称: `get_sample`。

功能描述: 读取索引对应的数据样本并进行前处理。需在接口中完成包括但不限于样本的加载、前处理以及数据格式的转换。

接口参数列表见表B.2。

表 B.2 `get_sample` 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	<code>dataset</code>	数据集对象	必选
输入	<code>index</code>	数据集索引, 类型为整型	必选
输出	<code>sample</code>	数据集的一个样本, 类型为元组或键值对	必选

异常处理, 文件不存在错误: 样本对应的文件不存在。

附加说明: 无。

B.2 采样器

B.2.1 构造采样器

接口名称: `construct_sampler`。

功能描述: 根据`sampler_config`构造相应的`sampler`。

接口参数列表见表B.3。

表 B.3 construct_sampler 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	sampler_config	类型为键值对 一个典型的 sampler_config 包括批量化大小 batch_size, 是否进行 shuffle, 是否在读完数据集后重复开始, 以及是否在结束读取后 drop_last_batch	必选
输入	dataset	数据集对象, 类型 Dataset	必选
输出	sampler	采样器对象, 类型 Sampler, 支持对应接口即可, 记录当前采样状态	必选

异常处理: 无。

附加说明: 无。

B.2.2 采样器遍历单个数据

接口名称: iterate_sample。

功能描述: 按照既定顺序遍历数据样本, 并返回一个sample。

接口参数列表见表B.4。

表 B.4 iterate_sample 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	dataset	要采样的数据集	必选
输入	sampler	采样器对象, 类型 Sampler, 会记录该采样器的状态, 例如当前迭代至哪一个样本	可选
输出	sample	数据集的一个样本, 类型为字典或元组	必选

异常处理: 无。

附加说明: 无。

B.2.3 采样器遍历批量数据

接口名称: iterate_batch。

功能描述: 按照既定顺序遍历数据样本, 并返回一个sample。

接口参数列表见表B.5。

表 B.5 iterate_batch 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	dataset	要采样的数据集	必选
输入	sampler	采样器对象, 类型 Sampler, 会记录该采样器的状态, 例如当前迭代至哪一个样本	可选
输出	sample	数据集的批量样本, 类型为字典或元组列表。其元素数量等于 batch_size 大小	必选

异常处理: 无。

附加说明: 无。

B.3 优化器

B.3.1 构造优化器

接口名称: `construct_optimizer`。

功能描述: 基于深度学习模型`model`, 根据`optimizer_config`进行优化器`optimizer`的构建。

接口参数列表见表B.6。

表 B.6 `construct_optimizer` 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	<code>model</code>	深度学习模型, 类型为 <code>Module</code>	必选
输入	<code>optimizer_config</code>	优化器的定义参数, 类型为键值对 一个典型的 <code>optimizer_config</code> 包括优化器方法名称 <code>optimizer_method</code> , 例如 SGD 等, 基础学习率 <code>base_learningrate</code> , 动量系数 <code>momentum</code> , 参数衰减系数 <code>weight_decay</code>	必选
输出	<code>optimizer</code>	优化器, 优化器本身需支持后续更新 <code>model</code> 的可学习参数接口, 类型为 <code>Optimizer</code>	必选

异常处理:

- Config解析错误: `config`无法解析或者配置参数超出范围。
- Model错误: 深度学习模型定义错误。

附加说明: 无。

B.3.2 优化器更新可学习参数函数

接口名称: `update_learnable_parameter`。

功能描述: 基于深度学习模型`model`, 当基于某损失函数, `model`前向传播结果以及反向传播梯度已经计算完毕时, 使用`optimizer`内的策略对于`model`内的可学习参数进行更新。

接口参数列表见表B.7。

表 B.7 `update_learnable_parameter` 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	<code>model</code>	深度学习模型, 类型为 <code>Module</code>	必选
输入	<code>optimizer</code>	优化器, 类型为 <code>Optimizer</code>	必选
输出	<code>model</code>	深度学习模型, 类型为 <code>Module</code>	必选

异常处理:

- Model定义错误: 深度学习模型定义错误。
- Model梯度错误: 深度学习模型梯度计算未完成或发生错误。
- Model更新错误: `optimizer`定义策略无法正确更新模型的可学习参数, 例如进入NaN状态。

附加说明: 此处`model`既为输入也为输出。

B.3.3 可学习参数梯度清零

接口名称: `zero_gradient` (可选)。

功能描述: 基于深度学习模型`model`, 将`model`内的可学习参数的梯度清0。

接口参数列表见表B.8。

表 B.8 zero_gradient 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	model	深度学习模型，类型为 Module	必选
输入	optimizer	优化器，类型为 Optimizer	必选
输出	model	深度学习模型，类型为 Module	必选

异常处理：Model定义错误：深度学习模型定义错误。

附加说明：此处model既为输入也为输出。

B.4 训练器

B.4.1 构造训练器

接口名称：construct_trainer。

功能描述：基于深度学习模型model，根据trainer_config进行训练器trainer的构建。

接口参数列表见表B.9。

表 B.9 construct_trainer 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	model	深度学习模型，类型为 Module	必选
输入	trainer_config	训练器的定义参数，类型为键值对 一个典型的 trainer_config 包括优化器 optimizer，训练数据集 train_dataset，训练循环数 epochs，额外数据集 val_dataset，评估函数 metrics，损失函数 loss，输出文件夹 output_dir 等	必选
输出	trainer	训练器，训练器需支持训练、推理和评估流程，推理和评估过程将根据参数自动调用，类型为 Trainer	必选

异常处理：

a) Config解析错误：config无法解析或者配置参数超出范围。

b) Model错误：深度学习模型定义错误。

附加说明：无。

B.4.2 单次数据训练

接口名称：train_one_batch。

功能描述：对单批次数据data_batch，计算其损失函数，并自动执行反向传播过程，包括使用optimizer内的策略对于model内的可学习参数进行更新、调整学习率等。

接口参数列表见表B.10。

表 B.10 train_one_batch 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	data_batch	数据批次，类型为包含 X、Y 键的键值对	必选
输出	loss_value	损失函数值，类型为浮点数	必选

异常处理：

- a) Model梯度错误：深度学习模型梯度计算未完成或发生错误。
- b) Model更新错误：optimizer定义策略无法正确更新模型的可学习参数，例如进入NaN状态。

B.4.3 单次数据循环训练

接口名称：train_one_epoch。

功能描述：对构造过程中已说明的训练数据train_dataset，计算其损失函数，并自动执行反向传播过程，包括使用optimizer内的策略对于model内的可学习参数进行更新、调整学习率等。

接口参数列表见表B.11。

表 B.11 train_one_epoch 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	epoch_index	当前数据循环序号，类型为整型	必选
输出	loss_value	损失函数值，类型为浮点数	必选

异常处理：

- a) Model梯度错误：深度学习模型梯度计算未完成或发生错误。
- b) Model更新错误：optimizer定义策略无法正确更新模型的可学习参数，例如进入NaN状态。

B.4.4 全循环训练

接口名称：train。

功能描述：对构造过程中已说明的训练数据train_dataset，自动抽取数据批次，完成全流程训练步骤，包括：计算其损失函数，使用optimizer内的策略对于model内的可学习参数进行更新、调整学习率等，根据构造过程中的参数决定是否调用推理和评估过程、是否保存相关记录和权重到对应文件夹。

异常处理：

- a) Model梯度错误：深度学习模型梯度计算未完成或发生错误。
- b) Model更新错误：optimizer定义策略无法正确更新模型的可学习参数，例如进入NaN状态。

B.5 推理器

B.5.1 构造推理器

接口名称：construct_inference。

功能描述：基于深度学习模型model，根据infer_config进行推理器infer的构建。

接口参数列表见表B.12。

表 B.12 construct_inference 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	model	深度学习模型，类型为Module	必选
输入	infer_config	推理器的定义参数，类型为键值对 一个典型的 infer_config 包括前处理过程 preproc，后处理过程 postproc 等	必选
输出	infer	推理器，推理器需支持前、中、后处理的自动调用，类型为Infer	必选

异常处理:

- a) Config解析错误: config无法解析或者配置参数超出范围。
- b) Model错误: 深度学习模型定义错误。

附加说明: 无。

B.5.2 单样本数据推理

接口名称: infer_one_sample。

功能描述: 对单个样本进行推理, 过程中自动调用预设的前、中、后处理过程等。

接口参数列表见表B.13。

表 B.13 infer_one_sample 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	input	输入数据, 类型适应前处理步骤的输入类型	必选
输入	infer_config	推理器的定义参数, 类型为键值对 可包含临时变化的一些参数, 如某些阈值等	可选
输出	output	输出, 类型为后处理步骤的输出类型	必选

异常处理: 无。

B.5.3 多样本数据推理

接口名称: infer_samples。

功能描述: 对多个样本进行推理, 过程中自动调用预设的前、中、后处理过程等。

接口参数列表见表B.14。

表 B.14 infer_samples 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	inputs	输入数据, 类型为列表, 元素类型适配前处理步骤的输入类型	必选
输入	infer_config	推理器的定义参数, 类型为键值对 可包含临时变化的一些参数, 如某些阈值等	可选
输出	outputs	输出, 类型为列表, 元素类型适配后处理步骤的输出类型	必选

异常处理: 无。

B.6 评估器

B.6.1 构造评估器

接口名称: construct_evalvater。

功能描述: 基于推理器infer, 根据eval_config进行评估器evalvater的构建。

接口参数列表见表B.15。

表 B.15 construct_evalvater 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	infer	深度学习模型推理器	必选
输入	eval_config	评估器的定义参数，类型为键值对 一个典型的 eval_config 包括默认的数据集合 default_dataset，评估函数 eval_metric 等	必选
输出	evalvater	评估器，类型为 Evalvater	必选

异常处理，Config解析错误：config无法解析或者配置参数超出范围。

附加说明：无。

B.6.2 模型评估

接口名称：evalvate。

功能描述：对使用推理器对数据集内的所有样本进行推理，根据预设参数记录推理结果，并计算评估分数等。

接口参数列表见表B.16。

表 B.16 evalvate 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	dataset	评估数据集	可选
输入	eval_config	评估器的可选参数，类型为键值对 一个典型的 eval_config 包括评估函数 eval_metric 等	可选
输出	score	分数，即评估结果	必选

异常处理：无。

B.7 数据操作

B.7.1 构造数据操作

接口名称：construct_operator。

功能描述：根据op_config进数据操作operator的构建。

接口参数列表见表B.17。

表 B.17 construct_operator 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	op_config	数据操作的定义参数，类型为键值对	可选
输出	operator	评估器，类型为 Operator	必选

异常处理，Config解析错误：config无法解析或者配置参数超出范围。

B.7.2 执行数据操作

接口名称：operate。

功能描述：对输入数据进行操作，并输出处理后的数据。

接口参数列表见表B.18。

表 B.18 operate 接口参数列表

参数类型	参数名称	参数说明	是否可选
输入	indata	待操作的数据，类型为键值对 一个典型的 data 包括数据输入 inputs，拟合目标 targets 等	可选
输出	outdata	完成操作的数据，类型为键值对 一个典型的 data 包括数据输入 inputs，拟合目标 targets 等	必选

异常处理：无。

参考文献

- [1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
- [2] Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural machine translation in linear time[J]. arXiv preprint arXiv:1610.10099, 2016.
- [3] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [4] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented transformer for speech recognition[J]. arXiv preprint arXiv:2005.08100, 2020.
- [5] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.
- [6] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. PMLR, 2019: 6105-6114.
- [7] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [8] Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 5693-5703.
- [9] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [10] Guo X, Li S, Yu J, et al. PFLD: A practical facial landmark detector[J]. arXiv preprint arXiv:1902.10859, 2019.
- [11] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [12] Hara K, Kataoka H, Satoh Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 6546-6555.
- [13] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
- [14] Ghodsi M, Liu X, Apfel J, et al. RNN-transducer with stateless prediction network[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 7049-7053.
- [15] Xie S, Sun C, Huang J, et al. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 305-321.

- [16] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision - ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11 - 14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [17] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.
- [18] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
- [19] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [20] Li M, Lv T, Chen J, et al. Trocr: Transformer-based optical character recognition with pre-trained models[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(11): 13094-13102.
- [21] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015: 234-241.
- [22] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [23] Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J]. arXiv preprint arXiv:1609.03499, 2016.
- [24] Yang Z, Dai Z, Yang Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding[J]. Advances in neural information processing systems, 2019, 32.
- [25] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
-