ICS 35.240
CCS L 70

# 中华人民共和国国家标准

GB/T 43782—2024

# 人工智能　机器学习系统技术要求

**Artificial intelligence—Technical requirements for machine learning system**

2024-03-15 发布　　2024-03-15 实施

国家市场监督管理总局
国家标准化管理委员会　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。

请注意本文件的某些内容可能涉及专利。本文的发布机构不承担识别专利的责任。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：中国电子技术标准化研究院、华为技术有限公司、北京百度网讯科技有限公司、上海商汤智能科技有限公司、腾讯云计算(北京)有限责任公司、网易(杭州)网络有限公司、浪潮电子信息产业股份有限公司、北京眼神科技有限公司、中国工程物理研究院计算机应用研究所、沈阳东软智能医疗科技研究院有限公司、北京软件产品质量检测检验中心、山东省计算中心(国家超级计算济南中心)、上海燧原科技有限公司、美的集团(上海)有限公司、海信集团控股股份有限公司、上海计算机软件技术开发中心、清华大学、北京航天自动控制研究所、中国科学院软件研究所、上海人工智能研究院有限公司、郑州中业科技股份有限公司、北京智芯微电子科技有限公司、武汉精测电子集团股份有限公司、长威信息科技发展股份有限公司、江汉大学、飞腾信息技术(北京)有限公司、中国医学科学院生物医学工程研究所、北京林业大学、中国电子科技集团公司第二十八研究所、常州微亿智造科技有限公司、兴容(上海)信息技术股份有限公司。

本文件主要起草人：董建、王莞尔、马骋昊、曹晓琦、靳伟、张琦、符海芳、丁诚、谢永康、郑少秋、于琦、张军、蒋慧、刘海涛、樊峰峰、杨春林、吴庚、王丽媛、程万军、孔昊、漆莲芝、高永超、周昱瑶、王思善、车正平、徐洋、高雪松、陈敏刚、李涓子、薛云志、孟令中、宋海涛、鲍薇、马珊珊、李斌斌、王资凯、李介、袁福生、张胜森、戴文艳、谷潇聪、蒲江波、吴钰祥、赵雅倩、李仁刚、朱宝峰、马泽宇、张单、李亚坤、廖陈志、王丽娜、徐颂、黄超、高卉、马元巍、张恒星、夏寅力、卢国鸣、蒋锴、梁汝鹏。

# 人工智能　机器学习系统技术要求

## 1　范围

本文件提出了机器学习系统框架，规定了功能、可靠性、维护性、兼容性、安全性和可扩展性要求。

本文件适用于各领域机器学习支持服务的系统及相关解决方案的规划、研发、评估、选型及验收的依据。

## 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。

GB/T 17235.1　信息技术　连续色调静态图像的数字压缩及编码　第1部分：要求和指南

GB/T 33475.2　信息技术　高效多媒体编码　第2部分：视频

GB/T 33475.3　信息技术　高效多媒体编码　第3部分：音频

GB/T 41867—2022　信息技术　人工智能　术语

GB/T 42018—2022　信息技术　人工智能　平台计算资源规范

ISO/IEC 14496-10　信息技术　视听对象编码　第10部分：先进视频编码（Information technology—Coding of audio-visual objects—Part 10：Advanced video coding）

ISO/IEC 15948　信息技术　计算机图形和图像处理　便携式网络图形：功能规范[Information technology—Computer graphics and image processing—Portable Network Graphics（PNG）：Functional specification]

ISO/IEC 23008-2　信息技术　异构环境中的高效编码和媒体传输　第2部分：高效视频编码（Information technology—High efficiency coding and media delivery in heterogeneous environments—Part 2：High efficiency video coding）

ISO/IEC 23008-3　信息技术　异构环境中的高效编码和媒体传输　第3部分：3D音频（Information technology—High efficiency coding and media delivery in heterogeneous environments—Part 3：3D audio）

## 3　术语和定义

GB/T 41867—2022、GB/T 42018—2022界定的以及下列术语和定义适用于本文件。

3.1

**机器学习系统　machine learning system**

能运行或用于开发机器学习模型、算法和相关应用的软件系统。

3.2

**机器学习框架　machine learning framework**

利用预先构建和优化好的组件集合定义模型，实现对机器学习算法封装、数据调用处理和计算资源使用的软件库。

3.3

**机器学习服务　machine learning service**

利用机器学习模型算法及其系统作为工具为组织或个人提供一种其期望的便利的方式的价值的IT服务。

注：机器学习算法服务是机器学习服务的一种，用于接受用户的应用请求，对输入数据进行处理，返回处理结果。

3.4

**模型编译器　model compiler**

将机器学习模型定义的计算过程转换为能在特定人工智能计算资源上执行的代码序列的计算机程序。

注：本文件中定义的模型编译器仅用于机器学习领域。

[来源：ISO/IEC/IEEE 24765:2017，3.681]

3.5

**资源池　resource pool**

各类系统资源的集合体。

3.6

**作业　job**

机器学习训练或推理任务的逻辑组合。

注：一个作业属于且仅属于某一个资源池，一个作业包括一个或多个任务。

3.7

**任务　task**

实现特定目标所需要的活动。

注：任务用于完成一个相对独立的业务功能，一个任务属于且仅属于一个作业。

[来源：ISO/IEC 22989:2022，3.1.35，有修改]

## 4　缩略语

下列缩略语适用于本文件。

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

CPU：中央处理器（Central Processing Unit）

DAG：有向无环图（Directed Acyclic Graph）

FPGA：现场可编程逻辑门阵列（Field Programmable Gate Array）

GPU：图形处理器（Graphic Processing Unit）

IDE：集成开发环境（Integrated Development Environment）

JSON：JavaScript对象注记法（JavaScript Object Notation）

REST：表现层状态转换（Representational State Transfer）

RPC：远程过程调用（Remote Procedure Call）

SOA：面向服务的架构（Service-Oriented Architecture）

SQL：结构化查询语言（Structured Query Language）

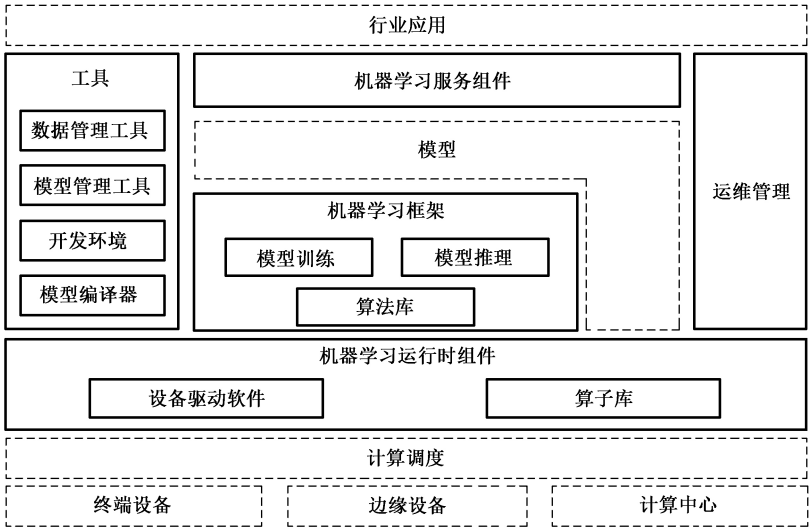
XML：可扩展置标语言（Extensible Markup Language）

## 5　系统框架

### 5.1　概述

机器学习系统包含机器学习运行时组件、机器学习框架、机器学习服务组件、工具和运维管理。提

供机器学习应用的开发、训练、部署、运行和管理能力，机器学习系统框架见图 1。




**注**：图中实线部分对应本文件相关规定，虚线部分仅为表明人工智能平台的系统组成，不属于本文件规定。

**图 1　机器学习系统框架**

## 5.2　机器学习运行时组件

机器学习运行时组件是为保障机器学习应用按照预期在特定机器学习系统上运行所必需的软件环境，包括设备驱动软件和算子库。

设备驱动软件负责机器学习各种类型任务的调度与执行，包括为机器学习任务分配提供资源管理通道，为应用提供存储管理、设备管理、执行流管理、事件管理和核函数执行功能。算子库提供机器学习算法在设备执行调度的最小计算单元，包括面向机器学习计算任务的通用算子和面向特定设备计算加速任务的优化算子。

## 5.3　机器学习框架

机器学习框架包含模型训练、模型推理及算法库三个模块，为机器学习应用开发、优化、验证和部署过程提供工具支撑。

模型训练用于机器学习应用设计开发阶段，该模块提供自动微分、损失函数和优化器等调用接口，提供模型定义、自动分布式并行训练和多硬件后端适配等能力。

模型推理用于机器学习应用的验证部署阶段，该模块提供模型加载、微调、性能评估和转换等接口，提供模型部署及推理加速等能力。

算法库面向机器学习训练、推理和模型性能优化任务，提供预先优化好的算法，以封装函数库的方式供用户调用，提升机器学习模型开发、优化、验证和部署的效率。

## 5.4　机器学习服务组件

机器学习服务是人工智能行业应用访问、利用机器学习能力和资源的主要方式，机器学习服务组件支持工作流管理、通用算法模板和应用部署。机器学习系统通过服务组件，进行服务部署、运行环境准备、运行状态汇报和服务容错等，并提供服务调用接口，供各领域上层应用调用。为满足应用场景的需求，机器学习系统可提供文本、图像、音频和视频及其他类型智能化操作的算法服务。

### 5.5 工具

#### 5.5.1 数据管理工具

数据管理工具提供人工智能数据的生存周期，包含定义、采集、预处理、模型构建、系统部署、系统维护、数据退出和系统退出的管理能力。提供各类数据源，包括结构化、半结构化和非结构化数据的接入、标注和质量控制，中间数据的管理、最终数据的管理、元数据的管理和数据使用溯源等能力，支持对海量结构化、非结构化数据的预处理与特征挖掘。

#### 5.5.2 模型管理工具

模型管理工具提供常用的机器学习模型及其变形，能按照一定的方式，如算法结构、应用范围，提供模型的分类检索；模型管理工具也可扩展支持模型导入、导出、更新、发布、迁移和版本控制等功能。在机器学习应用开发阶段，通过多模型组合开发、多模型集成、超参数设置和模型二次训练等方式支持模型优化与应用开发。

#### 5.5.3 开发环境

开发环境是机器学习全流程开发工具链，支持模型开发、算子开发和应用开发三个主流程中的开发任务，提供模型可视化、算力测试和 IDE 单机仿真调试等功能。

#### 5.5.4 模型编译器

模型编译器将计算过程的计算图和算子转换为环境兼容的中间表达或设备可执行的代码，支持编译优化、编译参数自动寻优、编译结果存储载入、自定义算子注册编译、模型格式转换等功能。

### 5.6 运维管理

运维管理提供系统所需的基本运维（例如安装部署、扩展、监控、报警、健康检查、问题及故障定位、升级和补丁、备份恢复和操作审计等）及管理功能（例如计算资源管理、权限管理、用户管理、日志管理、配置管理和安全管理等）。

## 6 功能要求

### 6.1 机器学习运行时组件

机器学习运行时组件的功能要求包括：

a) 应具备算法程序正常运行所需的基础软件组件，如设备驱动、通用算子库和操作系统等；
b) 应具备保障机器学习任务执行所需的设备管理及资源调度能力，包括设备管理、内存管理、事件管理、上下文管理、执行控制、故障感知与上报等；
c) 应具备基于设备定制开发的优化算子库；
d) 应具备算子级的执行控制和调度优化功能；
e) 应具备对环境中运行程序的访问权限控制和资源隔离功能；
f) 应具备计算资源的虚拟化与调度能力；
g) 应具备面向两种及以上机器学习框架模型格式的解析能力；
h) 应具备或集成集合通信库，以及单机多卡及多机多卡的计算平台架构。

### 6.2 机器学习框架

机器学习框架的功能要求包括以下内容。

a) 模型训练：
   1) 应具备对用户自定义数据的处理能力，包括图像的伸缩变换、音频特征提取和文本分词；
   2) 应具备用户自定义开发机器学习模型的能力，包括基本单元（如神经网络层）的基类、损失函数基类、用于参数更新的优化器基类；
   3) 应具备全连接层的调用和实例化功能，具备非线性激活函数的调用功能；
   4) 应提供接口获取训练过程信息，包括神经网络层的权重和偏置参数；
   5) 应具备静态图或动态图的执行模式；
   6) 应具备面向参数服务器和集合通信两种分布式架构的分布式并行能力；
   7) 应具备包括自动数据并行和模型并行结合的自动混合并行能力；
   8) 应具备时期和步骤粒度的数据处理回调功能；

   注 1：时期（Epoch）指训练时数据集的一次完整遍历。

   注 2：步骤（Step）指训练时完成一次前向计算和反向传播。

   9) 应具备自动混合精度（如 FP32 和 FP16）训练功能，面向不同的运算自动采用不同的数值精度按预期实施运算；
   10) 宜具备计算图重组等优化功能。

b) 模型推理：
   1) 如同时具备云侧和端侧推理能力，应提供云侧和端侧统一的中间表示，具备保存和加载该中间表示的能力；
   2) 应具备包括 CPU 和 GPU 的多种后端设备执行推理能力；
   3) 应具备至少两种编程语言接口，如 C＋＋、Python 和 Java 等；
   4) 宜具备多个模型的并发推理能力；
   5) 宜具备模型推理加速优化功能，如模型量化、内存复用和算子重新编排。

c) 算法库：
   1) 应具备模型评价函数，如准确度、精确度和平均绝对值误差等；
   2) 应具备损失函数，如回归损失和分类损失等；
   3) 应具备优化器算法；
   4) 应封装训练过程中常用的张量操作，包括池化运算和卷积操作等；
   5) 应提供激活函数，如线性单元激活函数、高斯误差线性单元激活函数；
   6) 应提供数学运算函数。

### 6.3 机器学习服务组件

机器学习系统提供通用服务能力，机器学习服务组件功能要求应包括：

a) 具备一种或多种算法服务；
b) 具备通用人工智能功能，如取流、解码、检测、识别分类、特征生成、特征比对和检索等功能；
c) 具备一种或多种单机服务，如模型自学习服务和批量推理服务等；
d) 具备一种或多种远程实时服务，如实时推理服务等；
e) 提供统一服务框架，如 SOA 和微服务等；
f) 提供统一、易用的算法服务接口，如 REST 和 RPC 等；
g) 具备常见的消息报文体格式，如 JSON 和 XML 等；
h) 具备同一算法服务的多实例部署功能；
i) 具备不同算法服务并发调用能力，各服务独立运行；
j) 具备多用户同时使用算法服务的功能，具备在多用户和高并发情况下的流量负载均衡，保证服务稳定运行；

k) 具备独立部署和运行能力，并具备服务动态扩容；
l) 具备服务容错能力，包括熔断、隔离、限流和降级等容错机制，来保证服务持续可用性；
m) 具备可扩展性，可方便增加新服务和动态调整服务节点等。

## 6.4 工具

### 6.4.1 数据管理工具

数据管理工具的功能要求包括：
a) 应具备各类数据源对接功能，包括结构化数据（例如传统关系型数据库）、半结构化数据、非结构化数据（例如文本、图像、音频和视频等）；
b) 应具备图像类数据格式采集功能，图像格式应符合 GB/T 17235.1 和 ISO/IEC 15948 的要求；
c) 应具备音频类数据格式采集功能，音频格式应符合 ISO/IEC 23008-3 和 GB/T 33475.3 的要求；
d) 应具备视频类数据格式采集功能，视频格式应符合 ISO/IEC 14496-10、ISO/IEC 23008-2 和 GB/T 33475.2 的要求；
e) 应具备对各类数据（例如文本、图像、音频和视频等）进行标注的能力；
f) 应具备引入和解析常见文件和数据格式的能力，如 parquet 和 carbondata 等；
g) 应具备多形态数据采集功能，包括单表采集、多表采集、增量采集、批数据采集和流数据采集；
h) 应具备对训练数据集、测试数据集和验证数据集独立提供数据生存周期管理的功能；
i) 应具备对原始数据、中间数据及产出数据进行增删改查及数据检索等操作的功能；
j) 应提供数据访问权限控制和版本控制能力，具备表粒度和字段粒度权限控制能力；
k) 应提供数据 IDE 工具，具备编写 SQL 和 Python 等脚本进行数据分析和探索的功能；
l) 应具备对敏感数据进行溯源管理功能；
m) 宜具备原始数据的诊断功能，如数据完整性检查、空值检查、规则校验和统计指标校验等；
n) 宜具备原始数据的相似度检测功能，过滤相似数据；
o) 宜具备多种元数据管理方法，如数据元信息生成、增删改查和血缘管理等；
p) 宜具备多种数据预处理手段，如数据的拆分、聚合、过滤和排序等；
q) 宜具备多种数据组合方法，如异构数据的组合、对齐和纠错等；
r) 宜具备用户数据集多版本管理功能；
s) 宜具备多人协同标注功能，并且具备多人协作任务的管理；
t) 宜具备不同数据集版本之间的数据分析统计功能，对比数据分布差异；
u) 宜具备推理结果数据结果回传模式。

### 6.4.2 模型管理工具

模型管理工具的功能要求包括：
a) 应具备模型的导入导出、更新、版本管理和权限控制等基础功能，模型导入导出地址应具备本地及远程对象存储等多种形式；
b) 应集成典型机器学习模型，具备模型的二次训练和保存模型多版本参数的能力；
c) 应基于多用户的权限控制，具备模型的安全管控能力；
d) 应提供模型封装和发布的能力，通过统一的接口提供模型服务的调用；
e) 应具备模型超参数的设置和保存功能；
f) 应提供用户友好的模型管理界面，展示模型的基本信息；
g) 应具备包括算法、超参数、参数、模型输入规范和模型输出规范五个要素的模型存储功能；

h） 宜提供多种形式的建模方式，如拖拽式 DAG 和 Notebook 等；
i） 宜具备多人协同建模能力；
j） 宜提供完整的模型分析报告，提高用户的模型选择和决策能力。

### 6.4.3 开发环境

开发环境的功能要求包括：
a） 应提供应用编程接口方式和图编排方式的应用开发方式，具备系统级调优、调试传输和异常分析等开发功能；
b） 应具备应用开发的单步调试功能；
c） 应具备自定义算子开发和算子级别性能分析功能或工具；
d） 应提供模型压缩、模型转换和模型显示输出工具；
e） 应提供模型训练调优工具；
f） 宜提供从模型训练到应用开发、调试、系统集成、构建打包和应用部署等的一站式应用集成开发环境；
g） 宜具备实时一站式图形界面调试环境，如当文本、图像、音频和视频等作为输入数据，开发环境可直接查看算法程序输出结果；
h） 宜具备边云协同的服务插件开发功能，如实现模型的边云同步和证书管理等；
i） 宜具备算子开发的自动调优、仿真调试调优和最优算子搜索工具。

### 6.4.4 模型编译器

模型编译器的功能要求包括：
a） 应提供编译器，对机器学习前端框架表达的计算过程进行图级和算子级编译；
b） 应具备多种机器学习算法模型和算子到设备可执行代码的自动映射功能；
c） 应具备机器学习算法程序的编译优化功能，如表达式化简和内存复用等；
d） 应具备自定义算子注册和编译功能；
e） 应具备计算图的自动切分功能；
f） 应具备编译结果的存储和载入功能；
g） 宜具备面向特定前端或硬件的定制优化规则接入机制；
h） 宜具备面向计算性能或内存空间的编译参数自动寻优功能。

## 6.5 运维管理

运维管理的功能要求包括：
a） 应提供多用户管理功能，具备多用户的权限管理能力，具备身份鉴别系统（例如 Kerberos）；
b） 应提供多租户管理功能，具备租户间的应用隔离、数据隔离、资源隔离和运行隔离等功能；
c） 应提供安装与升级功能，具备分发安装包、数据或模型参数文件，进行安装、升级、扩展和回滚；
d） 应提供备份与恢复功能，具备安装包、数据或模型参数文件的备份能力，以供故障后的系统恢复；
e） 应具备运行环境的监控能力，包括底层资源的统一监控，如 CPU 利用率和系统负载等；
f） 应提供日志管理功能，可根据日志进行故障定位及排查；
g） 应提供针对监控指标及日志的报警功能；
h） 宜提供主要监控指标的可视化展示功能。

## 7 可靠性要求

可靠性要求包括：

a) 应具备跟踪任务的执行状态，并对异常任务进行提示的能力；
b) 应具备资源受限或系统失效后持续提供或恢复服务的能力，如具备历史版本回滚、框架提供参数的保存能力等；
c) 应具备容错机制，具备系统在检测出异常输入或危险操作时的错误提示功能；
d) 应具备对误操作的抵御能力，确保误操作后系统的正常运行；
e) 应具备不同容量场景过载控制机制；
f) 应具备系统故障诊断能力，如机器学习框架可保存关键运行数据以用于故障定位和恢复；
g) 应具备系统故障隔离能力，如集群训练中，单一节点出现故障时可快速隔离；
h) 宜具备系统状态文件的冗余备份功能和容灾能力。

## 8 维护性要求

维护性要求包括：

a) 应具备数据集规模、均衡性、标注质量和污染情况对算法结果的影响分析功能；
b) 应具备在设计、实现和运行各阶段对应的性能度量指标和验证方法；
c) 应具备代码实现算法功能的正确性分析能力，包括代码规范性和代码漏洞检查；
d) 应具备系统实际运行中环境干扰的影响分析能力，包括噪声干扰和数据分布迁移等；
e) 宜具备异常数据的存储和导出能力。

## 9 兼容性要求

### 9.1 软件兼容性要求

软件兼容性要求包括：

a) 应具备软件服务兼容性，相互关联的软件服务能够正常运行，且在数据、信息和交互三个方面具有相互兼容的性质；
b) 不应依赖特定的软件运行环境；
c) 应具备系统运行的可移植性；
d) 应兼容主流操作系统，兼容多种编程语言；
e) 应兼容开源的通用接口，根据系统要求在最新版本中增强或优化；
f) 应具备模块间及模块内接口信息传递和互操作功能；
g) 应具备异源数据、异构数据库和新旧数据接口的转换功能；
h) 应兼容不同场景应用，兼容特定应用系统下的优化和扩展。

### 9.2 硬件兼容性要求

硬件兼容性要求包括：

a) 应兼容多种计算单元，例如 CPU、GPU、FPGA 和 ASIC 等；
b) 应兼容多种存储系统，例如分布式云存储和本地存储等；
c) 应兼容多种网络连接方式，例如以太网和 InfiniBand 网络；
d) 宜兼容多种计算平台，例如服务器、移动通信终端、平板式计算机和可穿戴设备等。

## 10 安全性要求

安全性要求包括：

a） 应提供对训练数据、部署模型、算法程序和服务接口的访问权限管理能力；

b） 应提供抵御对抗样本攻击和噪声污染的能力；

c） 应具备对访问用户的访问历史查询能力；

d） 应具备对权重文件的防篡改能力以及保护能力；

e） 应具备将任务详细状态输出到日志的能力；

f） 应具备对分布式任务的鉴别和加密通信能力；

g） 应具备部分模型的可解释能力；

h） 应具备部分模型的差分隐私训练能力；

i） 应具备部分模型和任务的稳健性评估能力；

j） 应屏蔽非法输入。

## 11 可扩展性要求

可扩展性要求包括：

a） 应具有标准格式的接口，降低维护和运行机器学习模型的成本；

b） 应具有模型部署到生产环境的标准流程，降低系统整合风险；

c） 应提供机器学习生存周期管理工具。

# 参 考 文 献

[1] ISO/IEC 22989:2022 Information technology—Artificial intelligence—Artificial intelligence concepts and terminology

[2] ISO/IEC/IEEE 24765:2017 Systems and software engineering—Vocabulary