人工智能技术（大模型部署工程师）

职业能力等级评价标准

（试行稿）

1 项目概况

1.1 项目名称

人工智能技术（大模型部署工程师）

1.2 项目定义

从事大模型的部署环境搭建与配置，将人工智能技术与互联网、金融、医疗、工业等领域的实际工作岗位进行应用适配并提供服务开发、运行监控、性能优化等服务的技术人员。

1.3 能力等级

本项目共设三个等级，分别为：初级、中级、高级。

1.4 能力特征

具备针对不同硬件（GPU、TPU 及边缘设备等）平台和业务场景，对大模型进行本地或容器化部署的能力。具备对大模型进行轻量化处理及适配优化的能力。具备将大模型封装为 API 服务，与业务系统进行集成的能力。具备使用工具对模型服务性能进行监控的能力。具备对运行资源进行调度，调整模型服务性能的能力。

1.5 职业能力等级评价要求

1.5.1 申报条件

具备以下条件之一者，可申报初级：

（1）累计从事相关职业工作 1 年（含）以上。

（2）相关专业在校学生。

具备以下条件之一者，可申报中级：

（1）取得本项目或相关职业初级评价证书（含职业资格证书、职业技能等级证书等）后，累计从事相关职业工作 2 年（含）以上。

（2）累计从事相关职业工作 4 年（含）以上。

（3）取得相关专业毕业证书。

具备以下条件之一者，可申报高级：

（1）取得本项目或相关职业中级评价证书（含职业资格证书、职业技能等级证书等）后，累计从事相关职业工作 3 年（含）以上。

（2）累计从事相关职业工作 6 年（含）以上。

（3）具有高等职业学校、高级技工学校、技师学院相关专业毕业证书，并取得本项目或相关职业中级评价证书（含职业资格证书、职业技能等级证等）。

（4）具有大专及以上相关专业毕业证书，并取得本项目或相关职业中级评价证书（含职业资格证书、职业技能等级证书等）后，累计从事相关职业工作 1 年（含）以上。

1.5.2 申报条件注释

（1）满足本项目高级别申报条件可申报本项目低级别。

（2）相关职业:大模型应用相关职业。

（3）参考《普通高等学校高等职业教育专业目录（2021 年）》涉及相关专业的归类[1]。

1.5.3 评价方式

职业能力等级评价考试包括理论知识、技能操作两个科目，较高等级必要时可增加综合评审。

理论知识考试以笔试为主，可以机考，条件成熟时试点开展网络考试，主要考核从业人员从事本职业应掌握的基本要求和相关知识要求。技能操作考核主要采用现场操作、模拟操作、面试答辩等方式进行，主要考核从业人员从事本职业应具备的技能水平。综合评审通常采取审阅申报材料、技术答辩等方式进行全面评议和审查。

理论知识考试和技能操作考核均实行百分制，成绩皆达 60 分（含）以上者算合格。

1.5.4 监考人员、考评人员与考生配比

理论知识考试和技能操作考核中的监考人员与考生配比不低于 1：15，且每个考场不少于 2 名监考人员。技能操作考核中考评人员为 3 人以上单数。

1.5.5 评价时间

理论知识考试时间不少于 90 分钟；技能操作考核时间:初级不少于 90 分钟，中级/高级不少于 120 分钟。

1.5.6 评价场所设备

理论知识考试：在标准教室或标准联网多媒体计算机教室进行。

技能操作考核：在配备必要办公软件及技能操作所需的工具的联网多媒体计算机教室进行，考试结束后能还原考试环境。

[1] 根据《普通高等学校高等职业教育专业目录（2021年）》,涉及的相关专业有人工智能技术应用（510209）、大数据技术（510205）、云计算技术应用（510206）、信息安全技术应用（510207）、物联网应用技术（510102）、软件技术（510203）、计算机网络技术（510202）、数字媒体技术（510204）、工业互联网技术（510211）、区块链技术应用（510212）、电子信息工程（080701）、电子科学与技术（080702）、通信工程（080703）、人工智能（080717T）、计算机科学与技术（080901）、软件工程（080902）、信息安全（080904K）、物联网工程（080905）等。

2 基本要求

2.1 职业道德

2.1.1 职业道德的基本知识

（1）遵纪守法，诚信从业。

（2）质量至上，精益求精。

（3）钻研业务，勇于创新。

（4）遵循伦理，合规安全。

2.1.2 职业守则

（1）专注专业，持续学习。

（2）刻苦钻研，勇于创新。

（3）敬业严谨，恪尽职守。

（4）遵规守纪，合法合规。

2.2 基础知识

2.2.1 硬件与系统知识

（1）GPU、TPU 架构特性与计算能力分析。

（2）CPU、GPU、内存等硬件资源监控与调度方法。

（3）Windows 及基于 Linux 操作系统权限配置、服务部署等系统管理知识。

（4）模型并行、流水线并行的硬件适配策略。

（5）边缘计算设备资源限制与模型轻量化部署。

（6）面向大模型数据读写的分布式存储系统支持技术。

2.2.2 人工智能与大模型理论知识

（1）Transformer 架构及注意力机制的核心原理与运行逻辑。

（2）扩散生成模型的核心原理与运行逻辑。

（3）大语言模型、多模态模型的结构组成与工作机制。

（4）大模型数据处理流程、训练算法及模型评估指标体系。

（5）模型量化、剪枝和蒸馏等压缩技术的原理与适用场景。

（6）分布式训练与推理的实现方式及数据同步方法。

（7）模型参数计算、计算复杂度分析与性能评估方法

2.2.3 容器技术基础知识

（1）容器化技术的构建、部署与管理方法

（2）Kubernetes 集群资源调度与服务管理策略

（3）云平台资源配置、弹性伸缩方案技术概述

（4）微服务架构设计原则及大模型服务化拆分技术概述

2.2.4 工程部署与模型微调技术

（1）python 虚拟环境管理知识

（2）vllm 等推理引擎优化部署技术要点

（3）Lora 等微调方法适用场景

（4）DeepSpeed 等微调工具应用知识

2.2.5 信息技术应用与数据处理

（1）计算机网络架构、通信协议及网络故障排查方法

（2）网络安全概念、常见攻击类型与防护技术手段

（3）关系型与非关系型数据库操作及架构设计方法

（4）数据采集系统架构、传感器选型与数据传输原理

（5）数据清洗、挖掘、可视化的技术与工具应用

（6）数据加密、备份恢复策略与实施方法

2.2.6 法律法规知识

（1）《中华人民共和国个人信息保护法》中与人工智能技术相关的知识。

（2）《中华人民共和国网络安全法》中与人工智能技术相关的知识。

（3）《中华人民共和国数据安全法》中与人工智能技术相关的知识。

（4）《中华人民共和国知识产权法》中与人工智能技术相关的知识。

3 工作要求

本标准初级、中级、高级的技能要求和相关知识要求依次递进，高级别涵盖低级别的要求。

3.1 初级

职业功能	工作内容	技能要求	相关知识要求
1. 模型适配与优化	1.1 硬件及模型适配	1.1.1 能根据任务需求，选择HuggingFace Hub、ModelScope等平台的预训练模型，及以满足基础业务场景需求 1.1.2 能根据硬件规格（如GPU（图形处理器）、TPU（张量处理器）型号）安装Python环境、驱动程序及加速库，以实现基础模型部署	1.1.1 主流模型架构适用场景 1.1.2 主流推理框架与硬件的适配关系知识
	1.2 轻量化处理	1.2.1 能使用Transformers/vLLM/llama.cpp等工具中的预设参数，对模型进行量化操作，以减少模型大小 1.2.2 能应用预置剪枝脚本减少模型参数规模，以提升推理速度	1.2.1 模型量化技术的原理与工具使用知识 1.2.2 模型优化算法的基本应用场景
	1.3 模型微调	1.3.1 能使用Llama Factory等图形化微调工具加载预训练模型并执行基础微调脚本，以适配简单任务 1.3.2 能使用Easy DataSet等大模型训练数据处理工具，制作符合微调格式的数据集，以确保数据兼容性	1.3.1 预训练模型微调的基本流程与参数配置知识 1.3.2 微调数据集的制作与格式转换
2.环境搭建与部署	2.1 环境配置	2.1.1 能通过命令行进行用户管理与环境变量设置，以配置基础系统环境 2.1.2 能按文档创建Python虚拟环境，以隔离依赖	2.1.1 系统环境变量配置方法 2.1.2 虚拟环境创建方法
	2.2 本地部署	2.2.1 能在本地环境分配基础计算资源（如CPU、内存），以运行模型 2.2.2 能安装Ollama、LM Studio等图形化工具，下载对应模型的权重文件，以提供大模型服务	2.2.1 计算资源分配 2.2.2 模型权重文件管理 2.2.3 常用图形化推理框架的安装及使用

	2.3 容器化部署	2.3.1 能使用Docker等容器技术构建基础镜像，以实现环境一致性 2.3.2 能基于容器部署模型推理服务，并通过Docker Compose进行简单服务编排	2.3.1 Docker等容器化技术的构建与管理方法 2.3.2 Docker Compose等服务编排配置方法
	2.4 云平台部署	2.4.1 能基于阿里云、华为云等云平台创建基础计算资源，以部署模型 2.4.2 能基于云平台环境安装推理框架，以运行模型服务	2.4.1 云等云平台的资源创建与管理流程 2.4.2 云平台上推理框架部署知识
3. 服务化与接口开发	3.1 API服务封装	3.1.1 能基于FastAPI等服务化框架，开发模型服务API接口 3.1.2 能将训练好的模型部署为RESTful API，以支持业务调用	3.1.1 服务化框架开发API的流程与规范 3.1.2 模型服务化封装的技术要点
	3.2 管理平台开发	3.2.1 能使用Vue等前端框架开发基础参数配置界面（如温度、top_p），以简化用户操作 3.2.2 能配置Nginx反向代理服务器设置请求频率阈值，以实现简单流量控制	3.2.1 前端界面开发技术 3.2.2 nginx流量控制方法
	3.3 系统集成	3.3.1 能使用Postman等工具测试API接口并与业务系统对接，以确保数据正确返回 3.3.2 能使用Wireshark等网络抓包工具调试接口问题（如超时），以定位错误根源	3.3.1 系统集成的接口规范与流程 3.3.2 接口调试的常用工具与方法
4.监控与维护	4.1 性能监控	4.1.1 能使用top、nvidia-smi等命令行工具监控CPU、GPU资源使用情况，以识别瓶颈 4.1.2 能通过日志文件分析模型运行状态，以生成基础报告	4.1.1 系统资源监控的基本方法 4.1.2 日志分析的技术与工具
	4.2 异常处理	4.2.1 能使用tail命令或ELK（Elasticsearch, Logstash, Kibana）等日志分析工具分析服务日志，定位问题根源 4.2.2 能分析与处理进程停止等常见服务中断问题，保障服务可	4.2.1 模型部署常见异常的原因与解决方案 4.2.2 服务故障的基础排查流程

		用性	
	4.3 资源调度	4.3.1 能根据大模型运行时的监控数据，调整批处理大小等参数，以优化资源使用 4.3.2 能根据大模型运行时服务器状态数据，重新分配CPU核心数、内存等基础算力资源，以提升效率	4.3.1 大模型推理参数与性能相关知识 4.3.2 资源分配策略

3.2 中级

职业功能	工作内容	技能要求	相关知识要求
1. 模型优化与适配	1.1 硬件及模型适配	1.1.1 能根据业务需求（如时延、精度）选择最佳参数规模的模型，以提升场景适配性 1.1.2 能使用nvidia-smi等工具配置GPU参数，进行显存分配等操作，以优化计算效率	1.1.1 模型性能指标解读方法 1.1.2 瓶颈诊断工具使用知识
	1.2 轻量化处理	1.2.1 能使用Transformers/vLLM/llama.cpp等推理框架中的量化工具，并选用合适的量化算法，对大模型进行量化，以适应特定业务场景的性能需求 1.2.2 能应用蒸馏技术（Knowledge Distillation）提升小参数规模模型（如qwen3-8b）的效率，以在资源受限设备上运行	1.2.1 模型压缩技术综合应用知识 1.2.2 蒸馏原理与实施方法
	1.3 模型微调	1.3.1 能使用命令行工具（如Hugging Face CLI）调整微调参数（学习率、批次大小），以提升模型精度 1.3.2 能处理复杂数据集（如多模态数据）并进行增强，以优化微调效果	1.3.1 微调参数优化方法 1.3.2 数据增强技术
2. 环境搭建与部署 1a1	2.1 高级环境管理	2.1.1 能使用Conda等工具管理多版本Python环境，以支持不同模型需求 2.1.2 能配置系统权限和服务守护进程，以确保环境稳定性	2.1.1 环境管理工具使用知识 2.1.2 系统服务配置原理
	2.2 本地部署优化	2.2.1 能优化本地资源分配（如GPU显存分区），以提升模型并行效率 2.2.2 能配置vLLM 、SGLang等推理引擎的张量并行/流水线并行等分布式参数，实现模型切分与计算资源协同调度，以优化吞吐量	2.2.1 资源优化技术 2.2.2 分布式推理相关知识

		与延迟性能	
	2.3 容器化部署	2.3.1 能配置Kubernetes集群部署模型服务，以实现高可用性 2.3.2 能优化容器资源限制（如CPU配额），以提升服务稳定性	2.3.1 Kubernetes服务管理策略 2.3.2 容器资源调度原理
	2.4 云平台部署	2.4.1 能配置云平台弹性伸缩策略，以自动调整资源应对流量变化 2.4.2 能使用云监控工具优化模型吞吐量，以提升服务效率	2.4.1 弹性伸缩方案技术 2.4.2 云性能监控方法
3. 服务化与接口开发	3.1 API 服务封装	3.1.1 能基于Celery等工具实现异步推理接口，以处理高并发请求 3.1.2 能通过缓存机制等措施优化API响应时间，以提升用户体验	3.1.1 高并发API架构的设计原则 3.1.2 API性能优化的技术与方法
	3.2 管理平台优化	3.2.1 能开发实时监控面板（如Prometheus集成），以展示模型服务状态 3.2.2 能实现高级流量管理（如QoS策略），以防止服务过载	3.2.1 监控工具集成知识 3.2.2 QoS策略原理
	3.3 集成优化	3.3.1 能实现API与数据库（如MySQL）集成，以支持动态数据交互 3.3.2 能诊断复杂集成问题（如数据格式冲突），并提供解决方案	3.3.1 数据库集成技术 3.3.2 问题诊断流程
4. 监控与维护	4.1 性能监控	4.1.1 能使用Prometheus、Grafana等工具实现实时性能监控，以预测潜在问题 4.1.2 能根据监控指标分析模型推理延迟原因，并提出优化建议	4.1.1 实时监控系统知识 4.1.2 延迟诊断方法
	4.2 复杂问题诊断	4.2.1 能诊断内存泄漏等资源竞争问题，并使用Valgrind等工具	4.2.1 高级诊断工具知识 4.2.2 分布式系统故障处理

		修复 4.2.2 能分析并处理数据同步失败等分布式系统错误，以恢复服务	
	4.3 资源优化调度	4.3.1 能使用Kubernetes调度器动态调整资源，以应对负载波动 4.3.2 能根据大模型运行时服务器状态数据，优化并发数等模型推理参数，以平衡性能和成本	4.3.1 动态调度技术 4.3.2 成本-性能优化方法

3.3 高级

职业功能	工作内容	技能要求	相关知识要求
1. 模型适配与优化	1.1 硬件及模型适配	1.1.1 能基于性能测试数据和成本模型，对比选择不同云平台或边缘硬件的最优配置组合，实现业务场景下成本与性能的最佳平衡 1.1.2 能基于硬件特性和模型结构，应用框架级加速技术（如算子优化、图优化），深度优化推理核心计算路径，最大化硬件利用率和推理速度	1.1.1 硬件性能指标与成本模型分析 1.1.2 模型推理底层优化原理
	1.2 轻量化策略设计	1.2.1 能基于精度容忍度和硬件限制，实施量化与剪枝的组合优化方案并进行精细校准，在可控精度损失下显著提升模型推理效率 1.2.2 能基于任务需求，设计和实施定制化知识蒸馏流程（优化损失函数、训练策略），训练出在目标硬件高效运行的高性能小模型	1.2.1 高级模型压缩技术原理与应用 1.2.2 知识蒸馏策略设计与调优
	1.3 模型微调	1.3.1 能基于参数高效微调技术（PEFT）和自动化工具，构建自动化微调流水线优化超参数，高效适配复杂任务并减少人工成本 1.3.2 能基于领域知识，设计和实现复杂数据增强与清洗流程，构建高质量微调数据集，提升模型在特定场景的泛化能力和鲁棒性	1.3.1 参数高效微调与自动化调参技术 1.3.2 高级数据增强与数据集构建方法
2.环境搭建与部署	2.1 环境策略实施	2.1.1 能基于团队需求，构建标准化、可复用的容器镜像及部署脚本（支持多环境），实现环境快速重建与部署一致性 2.1.2 能实施容器镜像安全扫描、最小化依赖管理和运行时安全策略配置，提升部署环境安全	2.1.1 容器化最佳实践与环境标准化 2.1.2 容器安全加固与漏洞管理基础

		性并降低风险	
	2.2 大规模部署管理	2.2.1 能基于负载特征和集群资源，配置和优化推理引擎的并行参数（张量并行、流水线并行、批次调度），在高并发下满足性能目标 2.2.2 能设计和实施大模型权重、数据的高效存储、分发与缓存策略（利用分布式存储/云存储），优化I/O性能满足大规模需求	2.2.1 大模型分布式推理配置与调优 2.2.2 大模型数据存储与访问优化策略
	2.3 容器化生产管理	2.3.1 能基于Kubernetes，配置服务治理组件（服务网格/API网关），实现流量管理、弹性伸缩和熔断限流，提升服务可靠性与可观测性 2.3.2 能基于资源监控和成本分析，优化Kubernetes资源配置（Requests/Limits、调度策略），实现集群资源高效利用和成本控制	2.3.1 Kubernetes服务治理与高可用策略 2.3.2 Kubernetes资源优化与成本控制方法
	2.4 云成本与效能优化	2.4.1 能基于云监控和成本数据，识别资源浪费并实施优化策略（自动启停、Spot实例），显著降低云资源使用成本 2.4.2 能配置跨区域负载均衡和故障转移策略，构建高可用、低延迟的多区域部署架构，提升服务韧性和用户体验	2.4.1 云成本分析与优化实战策略 2.4.2 多云/多区域高可用架构基础
3. 服务化与接口开发	3.1 高性能API架构	3.1.1 能基于高效通信协议和框架，设计和实现支持批处理、流式输出的低延迟、高吞吐模型API服务，满足不同交互需求 3.1.2 能实施细粒度的API访问控制策略（认证、授权、限流）并集成审计日志，保障API服务的安	3.1.1 高性能网络服务架构 3.1.2 API安全架构与实施

		全性和可追溯性	
	3.2 管理平台标准化	3.2.1 能开发可复用的模型服务监控、告警、配置管理功能模块，构建标准化管理界面模板，加速平台开发 3.2.2 能基于用户反馈和性能数据，诊断并优化管理平台交互体验和前端性能，提升用户满意度	3.2.1 可复用组件开发与监控集成 3.2.2 用户体验优化与前端性能调优基础
	3.3 复杂系统集成	3.3.1 能基于异步消息或事件驱动架构，实现大模型服务与业务系统的松耦合、高可靠集成，处理数据流转与错误恢复 3.3.2 能设计和维护清晰的接口契约文档，并实施接口兼容性测试，减少集成过程中的问题和故障	3.3.1 异步集成模式与可靠性设计 3.3.2 API契约管理与接口测试策略
	4.1 全链路监控实施	4.1.1 能配置和集成覆盖基础设施、服务、模型推理的全链路监控体系（指标、日志、追踪），实现问题快速定位与根因分析 4.1.2 能基于业务目标定义关键性能指标基线，配置智能告警规则和分级策略，确保关键问题及时响应并减少误报	4.1.1 全链路可观测性技术原理与应用 4.1.2 性能基线设定与智能告警配置
4. 监控与维护	4.2 高可用与容错设计	4.2.1 能设计和执行故障注入实验，验证系统容错能力并实施加固措施（重试、熔断、降级），提升系统韧性 4.2.2 能开发和维护覆盖核心功能、性能与异常场景的自动化测试套件，集成到CI/CD中保障服务变更质量与稳定	4.2.1 容错模式与混沌工程基础 4.2.2 大模型服务自动化测试策略
	4.3 资源效能规划	4.3.1 能基于历史负载分析与预测，预估未来资源需求，制定资源扩容或技术升级规划，支撑业务平滑扩展。	4.3.1 容量规划与预测方法基础 4.3.2 系统级性能瓶颈诊断与优化方法

		4.3.2 能基于深度性能剖析数据，诊断系统级资源瓶颈（计算、内存、通信），实施优化提升单点或集群整体计算效能	

4 权重表

4.1 理论知识权重表

项目		技能等级		
		初级	中级	高级
		（%）	（%）	（%）
基本要求	职业道德	5	5	5
	基础知识	20	15	10
相关知识	模型适配与优化	10	20	30
	环境搭建与部署	30	20	15
	服务化与接口开发	15	20	20
	监控与维护	20	20	20
合计		100	100	100

4.2 技能要求权重表

项目		技能等级		
		初级	中级	高级
		（%）	（%）	（%）
技能要求	模型适配与优化	25	30	35
	环境搭建与部署	30	25	20
	服务化与接口开发	20	25	20
	监控与维护	25	20	25
合计		100	100	100