《大模型与 Agent 开发实战》12月班

课程大纲

内容章节	内容模块	内容大纲
阶段一：全球顶尖大模型部署与调用方法		
【Part 1】大模型部署软硬件环境准备	大模型本地部署硬件配置指南	01 大模型应用需求分析
		02 硬件配置的选择标准
		03 主流显卡性能分析
		04 整机参考配置
		05 组装计算机硬件选型策略
	在线 GPU 白嫖与租赁指南	06 在线租赁 GPU 入门指南
		07 白嫖免费在线 GPU 策略
		08 阿里云、Kaggel、Colab GPU 平台使用方法
		09 付费租赁 GPU 平台介绍
		10 AutoDL 租赁与平台使用方法
		11 大模型 GPU 选配方法基本说明
		12 主流 GPU 分类及对应使用场景介绍
		13 主流 GPU 性能及性价比天梯图
		14 各类大模型显存占用及运行推荐配置
		15 个人或公司推荐大模型运行配置
	Windows&Ubuntu 双系统安装	16 大模型必备：Linux 操作系统入门介绍
		17 Ubuntu 桌面版操作系统入门介绍
		18 Ubuntu 22.04.3 镜像下载与启动盘制作
		19 【进阶】Windows&Ubuntu 双系统安装流程
【Part 2】大模型部署与调用方法	DeepSeek V3 & R1	01 DeepSeek v3 & R1 模型性能介绍
		02 Deepseek v3 & R1 使用方法：本地部署 & 在线 API
		03 Deepseek v3 & R1 主流本地部署方案(含硬件配置说明)
		04 DeepSeek 本地模型的四种部署方式
		05 使用 LM Studio 可视化工具离线运行 DeepSeek 模型
		06 DeepSeek API 账号注册与获取

		07 DeepSeek v3 & R1 API 参数详解
		08 DeepSeek v3 & R1 流式与结构化输出实操
		09 DeepSeek v3 & R1 实现多轮对话应用
		10 集成 DeepSeek v3 & R1 在线 API 实现流式对话问答
		11 集成 Ollama & vllm 启动 DeepSeek R1 蒸馏模型实现流式深度推理
		12 接入 Mysql 实现聊天会话的持久化存储
		13 应用并发量开发逻辑与压力测试详解
		14 DeepSeek R1 KTransformers 高性能部署实战
		15 DeepSeek 模型 KTransformers+Unsloth 联合部署方案
	Qwen3	01 Qwen3 模型核心特性介绍
		02 Qwen3 模型训练流程介绍
		03 Qwen3 各模型适用场景与硬件选择
		04【底层开发】Transformer 调用 Qwen3 模型流程详解
		05【个人使用】ollama 调用 Qwen3 模型完整流程详解
		06【企业级部署】vLLM 调用 Qwen3 模型流程详解
		07【CPU 推理】llama.cpp 调用 Qwen3 流程
		08【企业级前端】Open-WebUI 调用 Qwen3 流程
	Claude 4	01 anthropic 账号注册流程
		02 Claude 4 核心特性功能介绍
		03 Claude 4 的 Tool Use 实现流程
		04 Claude 4 的 streaming 流式响应技术
		05 Claude 4 的 Json Mode 模式应用方法
		06 用 Streamlit 打造前端页面可视化的私人 Claude 助理
	Gemini 2.5	01 Gemini 2.5 账号注册与模型调用方法

		02 Gemini 2.5 模型参数全解
		03 Gemini 2.5 模型多模态功能实现方法
		04 Gemini 2.5 调用外部工具实现方法
		05 基于 Vertex 平台调用 Gemini 2.5 模型流程
	GPT 4.1/o3/o4	01 OpenAI 账号注册与充值流程
		02 OpenAI 大模型技术生态介绍
		03 OpenAI 账户管理方法
		04 OpenAI Chat completion 模型基本介绍
		05 Chat Requestion body 与 Returns
		06 GPT 模型参数详解
		07 GPT tools 参数使用方法
		08 GPT 多模态功能实现流程
		09 多轮对话历史及面向对象多轮对话机器人实战
【Part 3】大模型部署策略与工具详解	大模型个人本地部署方案 Ollama	01 Ollama 项目完整介绍
		02 Ollama 核心接口 generate & chat
		03 Ollama OpenAI API 接口规范
		04 Ollama 在线 API 调用方法
		05 Ollama 模型服务接口压力测试
		06 Ollama 本地部署 DeepSeek-R1 模型
	大模型企业级部署方案 vllm	01 vLLM 框架整体概览
		02 Linux 系统本地部署 vLLM 环境配置
		03 vLLM 离线推理接口调用实战
		04 vLLM 服务器配置参数详解
		05 vLLM KV Cache 计算及 Scheduler 配置
		06 vLLM 推理类模型的在线推理调用
阶段二：微调&蒸馏垂域专属大模型实战		
【Part 4】大模型微调原理详解	有监督微调基本原理入门	01 fine-tunes 微调技术概念入门
		02 大语言模型的预训练与微调过程
		03 大语言模型底层输入输出解析
		04 双向自回归与自回归概念详解
		05 QLoR 原理实践性及原理分析
		06 Prefix-tuning 前缀微调基本原理介绍
		07 p-tuning 提示词微调法基本原理介

		绍
		08 LoRA 低秩适应微调方法基本原理介绍
		09 LoRA 微调有效性分析
	基于强化学习的微调方法基本原理入门	10 强化学习基本原理简介
		11 RLHF：基于人工反馈的强化学习微调方法
		12 RLHF 论文解读与原理介绍
		13 GRPO 原理解读与推理
		14 RLHF 完整训练流程介绍
		15 PPO 目标函数与约束策略
		16 PPO 基本原理介绍
		17 偏好函数及其优化方法介绍
		18 DPO 基本原理介绍
【Part 5】大模型微调数据集获取与创建方法	大模型微调数据集格式与基本要求	01 预训练、指令微调、偏好数据集基本说明
		02 大模型指令微调数据集基本格式要求
		03 instruction、input、output 基本字段说明
		04 数据格式映射与切割
		05 sharegpt 与 alpaca 格式解析说明
		06 数据集 json 格式表示方法
	微调数据集获取方法	07 Llama-Factory 开源微调项目介绍
		08 常用数据集下载与解析
		09 多语言能力微调通用数据集介绍
		10 代码能力微调数据集介绍
		11 Agent 能力微调数据集介绍
		12 知识点总结微调数据集介绍
		13 多功能微调数据集介绍
	微调数据集手动创建方法	14 deepseek Outputs 实现方法
		15 基于 Few-Shot 实现通用模型 Structured Outputs
		16 基于 Structured Outputs 创建问答对
		17 基于 Structured Outputs 创建偏好数据集
		18 微调数据集评估方法介绍
		19 常用微调数据集正确性校验流程
		20 COT 数据解构建

【Part 6】PEFT 手动微调实现流程	微调核心库使用方法介绍	01 Hugging Face Trainer 库下载与使用
		02 Hugging Face Trainer 训练与评估 API 调用
		03 PEFT 库基本介绍与下载方法
		04 PEFT 库高效微调方法介绍
		05 DeepSpeed 分布式训练基本概念介绍
		06 DeepSpeed 库下载与使用方法介绍
	PEFT 微调脚本编写方法	07 【Step 1】微调软硬件环境配置说明
		08 【Step 2】微调数据集加载与格式化
		09 【Step 3】加载预训练模型与集成 PEFT
		10【Step 4】配置训练参数与 DeepSpeed 参数
		11 【Step 5】使用 Trainer 进行训练与模型评估
		12 【Step 6】模型保存、参数合并与推理
		13 【Step 7】模型优化与调试
	微调模型并行策略介绍	14 【单机单卡】混合精度训练方法
		15 【单机单卡】梯度积累与内存优化方法
		16 【单机多卡】数据并行与模型并行方法
		17 【单机多卡】ZeRO 优化器使用方法
		18 【多机多卡】全局批量归一化执行流程
		19 【多机多卡】通信流程优化
【Part 7】借助 Llama-Factory 进行高效微调	Llama-Factory 项目部署与可视化微调	01 Llama-Factory 项目介绍与部署方法
		02 Llama-Factory 项目结构介绍与部分源码解读
		03 Llama-Factory 预训练、指令微调与强化学习微调方法
		04 Llama Board 可视化微调框架部署
		05 Llama Board 可视化微调流程
	Llama-Factory 高效微调实战	06 Llama-Factory SFT 基本微调流程介绍
		07 Llama-Factory SFT 脚本源码介绍与参数详解
		08 Llama-Factory 微调与参数合并流程
		09 Llama-Factory 微调模型推理与评估

		方法
		10 【实战】大模型语言能力微调
		11 【实战】大模型知识灌注微调
		12 【实战】大模型自主意识微调
		13 【实战】大模型 Agent 能力微调
	Llama-Facroty 预训练与强化学习微调实战	14 基于 Llama-Factory 的奖励模型微调方法
		15 Llama-Factory 的 PPO 训练流程
		16 Llama-Factory 的 DPO 训练流程
		17 Llama-Factory 的预训练流程与算力预估
		18 Llama-Factory 预训练数据集介绍
		19【实战】借助 Llama-Factory 从零到一预训练模型
【Part 8】借助 unsloth 进行高效微调	Unsloth 项目介绍与环境部署	01 Unsloth 项目简介与特性概述
		02 Unsloth 环境配置与部署方法
		03 Unsloth 项目结构与核心组件介绍
		04 Unsloth 与其他微调框架对比
	Unsloth 基础微调流程	05 Unsloth 数据准备与处理方法
		06 Unsloth 模型加载与优化技术详解
		07 Unsloth 微调参数配置全解析
		08 Unsloth 微调过程监控与可视化
	Unsloth 高级微调技术	09 Unsloth 的 LoRA、QLoRA 适配与实现
		10 Unsloth 模型量化技术深度解析
		11 Unsloth 分布式训练策略
		12 Unsloth 内存优化与性能调优技巧
	Unsloth 工程化与部署	13 Unsloth 微调模型的保存与加载
		14 Unsloth 模型部署最佳实践
		15 Unsloth 与其他框架的集成方法
		16 【实战】从零构建完整 Unsloth 微调工作流
阶段三：大模型 RAG 搭建高性能问答系统		
【Part 9】RAG 技术快速入门	基于大模型的企业级应用落地现状	01 新一代大模型应用开发范式
		02 大模型在真实行业中的创新应用案例
		03 大模型商业化落地的痛点与挑战
		04 有效避免大模型固有问题的解决方案
	大模型核心技术栈	05 开源大模型和闭源大模型如何抉择

		06 国内外开源大模型技术生态介绍
		07 【实战】大模型必备技能-函数调用（Tools）
		08 【实战】大模型必备技能-AI Agent 构建技巧
		09 【实战】大模型必备技能-高效微调
	大模型 RAG 技术的必备基础技能	10 Retrieval-Augmented Generation（RAG）技术出现的原因
		11 RAG 的经典结构与典型应用场景
		12 RAG 技术的核心构成-知识检索（Retrieval）
		13 RAG 技术的核心构成-内容生成（Generation）
		14 RAG 技术的核心构成-文本增强（Augmentation）
		15 构建 RAG 流程的两大阶段-数据与应用
		16 RAG 技术范式的演变过程
		17 基础 RAG 的构建流程与存在的问题
		18 高阶 RAG 在检索前、中、后期各阶段的优化技巧
		19 模块化 RAG 的核心编排思想
		20 详解主流 Embedding 模型（gte、bge）
		21 【实战】快速上手 OpenAI Embedding
		22 常用向量数据库以及类别介绍
		23 向量数据库的索引算法
		24 【实战】使用大模型的 In-Context Learning 解决特定任务需求
		25 工业级提示工程：思维链（Chain-of-Thought，CoT）回顾
		26 工业级提示工程：自洽性（Self-Consistency）回顾
		27 工业级提示工程：思维树（Tree-of-Thoughts，ToT）回顾
【Part 10】Embedding 模型与词向量数据库应用实战	词向量入门与开源词向量工具	01 词向量入门与核心概念介绍
		02 常用开源词向量库介绍
		03 词向量的数学原理与几何属性
		04 Word2Vec 的工作原理和特点
		05 text2vec-large 框架介绍与安装过程
		06 text2vec-large 词向量化实战

	OpenAI Embedding 模型 API 调用方法	07 大模型技术浪潮下的 Embedding 技术定位
		08 Embedding 技术入门介绍
		09 从 One-hot 到 Embedding
		10 Embedding 文本衡量与相似度计算
		11 OpenAI Embedding 模型与开源 Embedding 框架
		12 OpenAI Embedding 3 代模型介绍
		13 借助 Embedding 进行特征编码
		14 Embedding 结果的可视化展示与结果分析
		15 【实战】借助 Embedding 特征编码完成有监督预测
		16 【实战】借助 Embedding 进行推荐系统冷启动
		17 【实战】借助 Embedding 进行零样本分类与文本搜索
	常用开源词向量库安装与使用	18 GloVe 模型的介绍及其与 Word2Vec 的比较
		19 GloVe 预训练词向量创建方法
		20 FastText 的特色功能介绍与处理罕见词汇的能力
		21 ELMo 的动态词向量库介绍
		22 【实战】借助 ELMo 词向量以提升模型性能
	词向量数据库应用实战	23 Faiss 词向量库入门与安装
		24 【实战】使用 Faiss 进行高效相似性搜索
		25 自定义词向量数据库构建的最佳实践
		26 优化存储和搜索效率的策略
		27 Milvus 词向量数据库安装与使用方法
		28 Milvus 数据库索引算法介绍
		29 基于 Milvus 的相似度推荐流程
		30 【实战】大模型本地知识库问答系统中词向量库构建策略
【Part 11】RAG 应用进阶	RAG 项目的评估与优化	01 RAG 在 Query 阶段的优化（Query Transformations）
		02 RAG 如何构建优质的知识存储元数据
		03 RAG 在检索过程的 ranking 与

		refinement 增强
		04 什么是 Self-RAG
		05 RAG 的效果评估-质量指标
		06 RAG 的效果评估-能力指标
		07 RAG 的效果评估-评估工具
		09 【实战】使用 RAGAS 进行效果评估
		10 RAGAS 评估框架的缺点
	大模型应用开发框架	11 主流 RAG 构建框架：LangChain 与 LLamaIndex
		12 应用开发框架 LangChain 的基本介绍
		13 LangChain 核心抽象模块：Model I/O
		14 LangChain 核心抽象模块：Chains
		15 LangChain 核心抽象模块：Tools
		16 LangChain 核心抽象模块：Memory
		17 LangChain 核心抽象模块：Document loaders
		18 LangChain 核心抽象模块：Text splitters
		19 LangChain 核心抽象模块：Embedding Models
		20 LangChain 核心抽象模块：Vectorstores
		21 【实战】基于 LangChain 框架从 0 到 1 构建一个 RAG 系统
【Part 12】GraphRAG 应用	GraphRAG 本地源码部署	01 GraphRA G 基础入门与主流框架介绍
		02 GraphRAG 本地安装与配置文件说明
		03 GraphRAG CLI 索引构建方法与流程
		04 GrapRAG 源码安装及索引构建流程
	索引&检索源码解读与实战	05 索引构建源码解读：文档加载器及切分策略
		06 索引构建源码解读：实体关系抽取与 Pormpt 设计
		07 索引构建源码：社区检索算法与社区报告生成
		08 图数据库 Neo4j 源码安装及 Python 连接
		09 知识图谱基础概念与实战
		10 GraphRAG 自定义导入 Neo4j
		11 检索流程源码讲解：本地搜索

		12 检索流程源码讲解：全局搜索
	多模式文档接入与优化策略	13 .csv 格式索引流程源码解读
		14 数据库.csv 文件切分优化策略
		15 Auto Prompt Tuning 自适应垂类领域提示优化）
		16 结构化数据存储 Neo4j 流程规范
		17 PDFDocx 图片格式文件难点与解析方法概览
		18 MinerU 项目说明与本地部署
		19 MinerU 解析文件类型详解
		20 手动构建 PDF 文档加载器思路与源码
		21 动态结构感知分块策略源码接入及思路
	GraphRAG 工程化应用	22 GraphRAG API 接口源码概览
		23 Prompt 动态编排服务接口设计
		24 索引构建的 Python SDK 的自动化构建管道实现
		25 跨文档类型合并知识图谱解决方案-增量更新
		26 检索服务的工程化接口封装
阶段四：MCP 技术全体系深度实战		
【Part 13】大模型 MCP 应用入门实战	MCP 技术基础概念	01 MCP 与 Function Calling 的根本区别
		02 什么是 MCP Client
		03 什么是 MCP Server
		04 MCP 客户端和服务器的运行规范
		05 MCP 通信提示词模版设计
		06 什么是 MCP 的技术协议
		07 MCP 服务端 Resources、Tools 与 Prompts 类型
		08 MCP 的通信协议：Sutdio 和 SSE
		09 MCP 公开服务器资源整体概览
	从零构建 MCP 智能体开发实战	01 MCP 开发工具 SDK 快速入门
		02 uv 工具安装与使用实战
		03 MCP 基本客户端代码开发规范
		04 MCP 客户端接入开源&在线模型
		05 MCP 服务端执行流程详解
		06 通用外部工具注册 MCP 工具

		07 基于 Studio 协议的 MCP 本地服务器编写与启动方法
		08 客户端集成服务端自动化处理外部工具调用流程
		09 MCP Inspector Debug 工具使用技巧
		10 实战案例：MCP+GraphRAG 搭建检索增强智能体
		11 实战案例：基于 MCP 搭建 NL2SQL 数据分析智能体
	从零到一手搓 Mini Manus 开发实战	01 DeepSeek 模型调用参数详解
		02 DeepSeek 模型响应模式详解
		03 DeepSeek 并行调用与分发模式
		04 手动实现实时联网搜索功能
		05 本地 Python 代码解释器与 Mysql 环境配置
		06 手动实现实时 Github 代码搜索
		07 MiniManus 完整工作流开发逻辑
		08 多轮对话和自我认知功能测试
【Part 14】大模型 MCP 工程化应用进阶	MCP Python SDK 深度实战	01 MCP Python SDK 三大核心组件
		01 服务器端与客户端原语
		02 MCP 通讯协议详解
		03 MCP 数据隐私与沙箱执行
		04 Prompts 预定义交互提示模版
		05 Resources 与 Tools 的适用场景
		06 服务器基础架构和请求处理流程
		07 工具开发、资源与提示管理
		08 客户端与服务器连接和工具注册
		09 上下文管理及会话状态维护
		10 性能优化策略：缓存与并行处理
		11 性能优化策略：请求追踪与指标监测
		12 性能优化策略：Streamable HTTP
		13 性能优化策略：Docker 高并发部署
		14 实战：DeepResearch 构建复杂任务超长文本调研报告
	OpenAI Agents SDK 集成 MCP 实战	01 OpenAI Agents SDK 概述
		02 三大基本构件：Agents、Handoffs、Guardrails
		03 基础环境搭建与依赖管理
		04 工具调用、历史消息与异步执行模式
		05 Agent 配置参数与生命周期详解

		06 自定义 Agent 行为优化技巧
		07 Agents SDK 如何接入 MCP
		08 MCP 服务器架构与状态管理
		09 MCP 客户端配置与接入规范
		10 自定义大模型接入方法与跨平台响应
		11 多智能体系统设计与Handoffs 机制详解
		12 多智能体协作模式开发实战
		13 Guardrails 安全机制策略定制化开发
		14 实战：DeepSeek + MCP 搭建 Agentic RAG 问答系统
	LangChain&LangGraph 集成 MCP 实战	01 LangChain 集成 MCP 生态概览
		02 LangChain MCP 集成架构与本地安装
		03 单 & 多 MCP 服务器集成开发实战
		04 Studio & SSE 不同传输方式应用实战
		05 高级应用：构建 ReAct 模型 LangGraph MCP 服务器
		06 高级应用：自定义 Multi-Agent 工作流
		07 高级应用：自定义图结构与异步、流式应用输出
		08 实战：从 0 到 1 搭建 Multi-Agent 智能体系统
阶段五：必备十大主流 Agent 开发工具实战		
【Part 15】低代码开发框架	Coze 平台实战	01 Coze 平台概述
		02 Coze 的核心概念（Bot、工作流、插件）
		03 在 Coze 中创建和配置 AI Bot 的步骤
		04 如何编写有效的提示词
		05 设置 Bot 的人设和回复逻辑
		06 如何为 Bot 添加特定技能以增强功能
		07 创建和管理知识库，提升 Bot 的知识水平
		08 如何在 Coze 中预览 Bot 的功能
		09 如何将 Bot 发布到各类社交平台和

		通讯软件上
		10 管理和更新已发布的 Bot
		11 Coze 工作流的创建与管理
		12 理解工作流在 Coze 中的作用
		13 如何在工作流中使用代码节点
		14 编写自定义代码以实现特定功能
		15 如何安装 Coze 插件市场中的插件
		16 如何开发自定义插件
		17 将 AI Bot 部署到不同平台的具体步骤
		18 通过 API 将 Coze 的 AI Bot 与其他系统集成
		19 案例：构建 GitHub Star 管理助手
		20 案例：构建智能客服助手
		21 案例：创建小红书文案编写智能体
	Dify 工作流编排	01 Dify 平台整体介绍
		02 本地部署 Dify 系统
		03 Ollama 框架应用与部署
		04 在 Dify 中添加 Ollama 模型的步骤
		05 Xinference 框架的功能和优势
		06 将 Xinference 集成到 Dify 的方法
		07 使用 Dify 的知识库功能进行数据库管理
		08 导入和处理多种格式的文档（如 TXT、PDF、HTML ）
		09 设置分段清洗和索引方式
		10 配置 Embedding 模型和检索设置
		11 利用 Dify 的流程编排功能安排工作流
		12 工作流（Workflow）和对话流（Chatflow）的区别
		13 节点的类型和功能（如 LLM 节点、代码节点、模板节点、工具节点、迭代节点）
		14 使用条件分支和迭代节点处理复杂逻辑
		15 Ollama 框架与 Dify 打通
		16 案例：使用 Dify 进行文本解析

		17 案例：实现智能对话和问答系统
		18 案例：对接微信对话智能客服
		19 案例：对接企业级微信转接系统
		20 案例：企业注册 IP 打通全链路指导
		21 如何构建 Dify 的 API 进行系统集成
		22 Dify 源码架构解析
		23 Dify 核心模块剖析与数据流动
		24 Prompt 编排模块的实现原理
	自动化工作流-n8n	01 自动化工作流 n8n 框架详解
		02 n8n 两种部署方式：本地 / 云端部署
		03 n8n 核心组件详解
		04 n8n 数据流控制机制：输入输出变量解析 + 多路径流程演示
		05 n8n 实操：搭建「关键词监控 + 邮件预警系统」
		06 n8n 实战：飞书自动审批流程 + 人工加签节点
		07 n8n 实战组合：自动日报生成 + 发送 + 存储归档
		08 n8n 执行机制深入讲解：串行 / 并行 / 条件触发与流转
		09 n8n 节点玩法拓展：RAG 摘要、问答、改写
		10 n8n 与外部系统对接机制：OAuth 授权 / API Token 管理
		11 n8n 调试技巧与异常处理机制讲解
		12 n8n 企业案例：销售流程自动化搭建
【Part 16】AI Agent 基础理论与实现形式	AI Agent 的基本概念	01 AI Agent 基础概念介绍
		02 为什么会需要基于大模型的 AI Agents
		03 AI Agents 产品在市场上的应用价值
		04 AI Agents 的工作流程与设计模式
		05 构建 AI Agents 的核心组件
		06 大模型理论基础
		07 Agent 记忆能力（Memory）功能设计方法
		08 Agent 规划能力（Planning）设计方法

		09 工具调用（Tools）实现思路
		10 响应模式（Action）设计思路
		11 典型的 AI Agents 类型：反射代理、基于目标的代理
		12 基于效用的代理、基于模型的代理与学习代理
	AI Agent 的典型应用	13 当前 AI Agents 的实现形式
		14 ChatGPT 中的 AI Agent 应用
		15 热门 AI Agent 开源工具介绍
		16 典型工具：AutoGPT 的项目定位和产品价值
		17 典型工具：AutoGPT 与 ChatGPT 的关系和区别
		18 【实战】Linux/Ubuntu 系统私有化安装部署 AutoGPT
		19 【实战】Windows 系统私有化安装部署 AutoGPT
		20 【实战】OpenAI GPT API 获取与项目中设置方法
		21 【实战】应用 AutoGPT 快速构建可用于私人助理的 AI Agent 工作流
【Part 17】AI Agent 开发技巧与实战应用	AI Agent 应用开发范式	01 引导 LLMs 成为 AI Agents 的提示工程技巧
		02 Single IO （输入-输出模式）
		03 Instruction Prompting（指令提示）
		04 Self-Ask （自问）
		05 【实战】使用 LangChain 构建 Plan & Solve Agents
		06 【实战】使用 LangChain 构建 Self Ask Agents
		07 Several I-Os 提示工程
		08 ReAct 理论思想
		09 OpenAI Funcation Calling 原理
	从 0 到 1 构建文本分析的 AI Agent	10 项目整体架构与构建流程
		11 Ollama 框架概述及 LLaMA3 模型的应用
		12 掌握 Ollama 框架的命令行交互
		13 掌握 Ollama 框架的 REST API 对话服务
		14 封装大语言模型交互逻辑
		15 封装文本分析 AI Agent 核心类

		16 自定义构建 Tools 类
		17 基于 Streamlit 构建用户前端交互页面
	基于 ReAct 思想构建对话式 Agent 应用	18 ReAct 思想在 AI Agent 中的应用
		19 手动实现 ChatBot 基类
		20 构建基于 ReAct 思想的 Pormpt 设计
		21 定义并接入多个外部工具
		22 制定消息处理流程和输入输出规范
		23 多轮对话 Agent 应用功能用例测试
	OpenAI 函数调用结合 RAG 构建自主 AI Agent	24 Function Calling 与 AI Agent 的区别
		25 Haystack 开发框架介绍
		26 基于 Haystack 构建 RAG pipeline
		27 构建与向量数据库交互的 API Calls
		28 自定义工具列表
		29 构建完整链路及生成测试用例
		30 集成 Streamlit，实现类 ChatGPT 的聊天式应用程序
【Part 18】最强 Agent 框架 Assistant API	Assistant API 基础入门	01 Assistant API 框架的整体介绍
		02 Assistant 对象的创建方法
		03 Thread、Messges 及 Run 应用方法
		04 Run 运行时的状态转移机制
		05 实现 Run 状态的轮询方法，并实现 Assistant API 完整链路
		06 Assistant API 进阶应用方法介绍
		07 File Search 内置工具说明及文件管理
		08 基于 Assistant API 创建在线私有知识库
	Assistant 应用进阶	09 在 Assistant 和 Thread 定义 File Search 工具的四种策略
		10 如何处理 Assistant API 输出响应中的注释
		11 Code Interpreter（代码解释器）的应用技巧
		12 基于 Funcation Calling 实现本地代码解释器
		13 为什么企业级应用必须接入流式输出

		14 Assistant API 中流式输出的开启方法
		15 Assistant API 流式传输中的事件流原理细节
		16 如何在 Assistant API 流式传输中接入外部函数
		17 应用案例（1）：异步构建 Assistant 对象的工程化代码
		18 应用案例（2）：集成外部函数方法及项目完整功能介绍
【Part 19】主流 Agent 开发框架	LangChain	01 主流 RAG 构建框架：LangChain 与 LLamaIndex
		02 应用开发框架 LangChain 的基本介绍
		03 LangChain 核心抽象模块：Model I/O
		04 LangChain 核心抽象模块：Chains
		05 LangChain 核心抽象模块：Tools
		06 LangChain 核心抽象模块：Memory
		07 LangChain 核心抽象模块：Document loaders
		08 LangChain 核心抽象模块：Text splitters
		09 LangChain 核心抽象模块：Embedding Models
		10 LangChain 核心抽象模块：Vectorstores
		11 【实战】基于 LangChain 框架从 0 到 1 构建一个 RAG 系统
	LangGraph 底层框架	01 LangChain 的 AI Agent 开发框架架构设计
		02 LangGraph 的底层构建原理
		03 Langgraph 底层源码解析
		04 LangGraph 中如何接入大模型做问答流程
		05 LangGraph 中 State 的定义模式
		06 使用字典类型定义状态的方法与优劣势
		07 LangGraph 状态管理中 Reducer 函数的使用
		08 MessageGraph 源码功能解析
		09 LangSmith 基本原理与使用入门

	LangGraph 实现 Agent	10 LangGraph 代理架构及 Router Agent 介绍
		11 LangGraph 中可应用的三种结构化输出方法
		12 结合结构化输出构建 Router Agent（数据库）
		13 Tool Calling Agent 中 ToolNode 的使用
		14 Tool Calling Agent 的完整实现案例：实时搜索与数据库集成
		15 LangGraph 中 ReAct 的构建原理
		16 案例实操：构建复杂工具应用的 ReAct 自治代理
		17 LangGraph 中如何使用流式输出
		18 LangGraph 中的事件流
		19 Agent 长短期记忆认知
		20 LangGraph 的短期记忆及 Checkpointer(检查点)
		21 检查点的特定实现类型 -MemorySaver
		22 检查点的特定实现类型 -SqliteSaver
	LangGraph 高阶应用	23 长期记忆和 Store（仓库）
		24 LangGraph 知识点概述总结
		25 LangGraph 中的 HIL 实现思路
		26 标准图结构中如何添加断点
		27 复杂代理架构中如何添加动态断点
		28 案例：具备人机交互的完整 Agent 信息管理系统
		29 Single-Agent 存在的局限
		30 Multi-Agent 架构分类及子图的通信模式
		31 父、子图状态中无共同键的通信方式
		32 案例：基于网络架构实现智能 BI 数据分析多代理系统
		33 Supervisor 架构介绍与基本构建原理
		34 案例：基于 Supervisor 架构实现多代理系统
		35 GraphRAG 基本介绍与核心架构
		36 案例：Multi-Agent 实现混合多知识

		库检索
	AutoGen	01 AutoGen 开发框架整体介绍 02 配置 AutoGen 开发环境 03 ConversableAgent 源码解析 04 AutoGen 如何接入在线大模型 05 AutoGen 如何接入开源大模型 06 模型配置过滤器的使用方法 07 AutoGen 中 代码解释器的原理与接入方法 08 如何基于 AutoGen 构建自动 Debug 的数据分析器
实战项目一：企业级 RAG 知识库问答项目实战		
【Part 20】从零搭建企业级高性能 RAG 引擎：FuFan-Chat	**系统功能设计**	01 项目 Web 前端应用的个性化展示 02 消息队列与异步编程的实现方法 03 FuFan-Chat 中的会话状态管理 04 FuFan-Chat 中的缓存机制
	通用知识的问答流程	05 大模型前后端数据交互流程详解 06 在项目中融入 LangChain 开发规范 07 知识文档切分的常见算法 08 如何正确加载唯一会话对应的 Memory 记忆信息 09 工程化提示工程与自然语言提示工程的区别 10 优化提示工程大幅提升通用知识问答的回复准确率 11 使用 LangChain 框架集成 Mysql 数据库 12 如何从 Mysql 数据库中加载实时加载单一会话的 Memory
	本地知识库问答流程	13 检索增强生成（RAG）在 FuFan-ChatChat 项目中的构建方法 14 利用 GPT 4 构建问答对形式的私有知识 15 有效提升问答对知识的文本质量的方法 16 构建私有知识的关键 MetaData 信息 17 基于历史对话重新生成 Query 及其他 Query 优化相关策略 18 Faiss 向量数据库的接入及使用技巧

		19 使用 Query Transformations 进一步优化 RAG 检索效果
	AI 联网实时检索流程	20 精准切分.md 文件、PDF 文件等多种不同形式的文件格式
		21 使用 Milvus 高效存储大批量知识库文本
		22 在 RAG 流程中接入 Multi-Tools（API）
		23 实现联网检索功能，精准解析 Url 中的数据信息
		24 检索效果优化：实现 Reranking 重排
		25 使用 GLM4 标注检索数据并评估生成的回答
	大模型推荐系统流程	26 大模型在推荐系统中的应用形式
		27 教育领域问答数据的准备与预处理
		28 使用大模型对数据进行特征工程
		29 教育推荐系统的召回与精排
		30 RAG + AI Agents 的结合方式
		31 使用 RAG + Prompt 优化向量相似度筛选
	功能测试与部署上线	32 精准设计测试用例
		33 大模型应用部署：Docker 容器
		34 Docker 核心概念：镜像、容器和仓库
		35 Docker 快速上手（操作 + 掌握）
		36 通过 Docker 手动构建镜像
		37 通过 Docker 自动构建镜像
		38 Dokcer 镜像的使用方法
		39 实现 Docker 镜像推送
	应用落地的数据隐私与挑战	40 FuFan-ChatChat 智能问答系统的业务价值验证
		41 FuFan-ChatChat 智能问答系统的业务提效分析
		42 大模型应用的落地限制与数据隐私
		43 大语言模型的可解释性
		44 FuFan-ChatChat 智能问答系统的可优化空间
		45 FuFan-ChatChat 智能问答系统实际落地的挑战

实战项目二：智能客服 Agent 项目开发实战

【Part 21】从零搭建新一代智能客服 Agent：AssistGen	智能客服应用场景分析	01 项目介绍与课程安排
		02 生成式大模型特性与客服系统契合度分析
		03 目前国内外大模型智能客服系统简介
		04 智能客服系统场景分析：交易场景、客户维护场景、售后场景
		05 基于智能客服系统的新一代智能营销系统前瞻
	智能客服基座大模型	06 Ollama 本地部署 DeepSeek R1 模型完整流程
		07 Ollama REST API 核心接口：generate
		08 Ollama 兼容 OpenAI API 接口规范
		09 Deepseek v3 & R1 在线 API 调用方法
		10 AssistGen 项目结构与本地启动流程介绍
		11 AssistGen 多模型集成工厂函数规范
	智能客服系统数据集准备	12 GraphRAG 本地安装与配置文件说明
		13 图数据库 Neo4j 源码安装及 Python 连接
		14 Auto Prompt Tuning 自适应垂类领域提示优化)
		15 数据库.csv 文件切分优化策略
		16 结构化数据存储 Neo4j 流程规范
		17 PDFDocx 图片格式文件难点与解析方法概览
		18 MinerU 项目说明与本地部署
		19 手动构建 PDF 文档加载器思路与源码
		20 动态结构感知分块策略源码接入及思路
		21 索引构建的 Python SDK 的自动化构建管道实现
		22 跨文档类型合并知识图谱解决方案-增量更新
		23 检索服务的工程化接口封装
	AssistGen 项目架构	24 AssistGen 项目整体架构层级设计
		25 底层技术栈与核心逻辑说明

		26 对话、推理、视觉、Agent 模型参数配置
		27 项目启动方法与功能演示
	用户意图识别模块	28 电商场景的五大业务模块功能分类
		29 父、子图设计与数据流向
		30 基于 LangGraph 实现可扩展的意图识别节点
		31 意图识别的提示词设计与路由组件实现
		32 业务模型提示词设计与模型结构化输出
	自定义多代理工作流	33 通过动态 Neo4j Schema 增强语义
		34 实现复杂/多跳任务分解模块
		35 多工具统一接口规范与自定义工具接入方法
		36 Map-Reduce 构建并行工具执行机制
		37 图片识别功能接入与 LangGraph 图状态的集成
	混合知识库检索	38 人工 + 大模型智能生成预构建 Cypher 字典
		39 基于 neo4j-graphrag + LLM 自动生成 Cypher 流程
		40 Cypher 语法、权限、关系方向校验与自我纠正子图
		41 进阶：如何接入大模型进行 Cypher 异常与映射校验
		42 检索效率优化：如何预构建海量 Cypher 工具节点
		43 GraphRAG 检索 ResultFul API 构建自定义工具节点
实战项目三：“Manus”通用智能体项目开发实战		
【Part 22】从零搭建"Manus"通用智能体：InsightGen	通用智能体基础概念	01 类 Manus 智能体的发展与趋势
		02 大模型在市场分析中的应用场景剖析
		03 主流市场分析智能体系统对比分析
		04 市场分析核心场景：竞品分析、消费者洞察、趋势预测
		05 基于 AI 的市场分析新范式与方法论
	InsightGen 项目架构设计	06 InsightGen 整体架构与核心组件
		07 MCP 技术优势与应用场景分析

		08 多模型集成策略与参数配置优化
		09 开发环境与依赖配置全流程
		10 MCP 与传统 Function Calling 对比优势
	数据采集与处理引擎	11 市场数据源接入策略与架构设计
		12 公开 API 集成：社交媒体、新闻、行业报告
		13 自动网络爬虫工具开发与 MCP 注册
		14 非结构化数据预处理流程
		15 数据采集工具的错误处理与重试机制
	Google ADK 框架集成	16 InsightGen 的 MCP 通信协议配置
		17 工具、资源与提示词的 MCP 注册流程
		18 Studio 与 SSE 协议在市场分析场景中的选择策略
		19 基于 ADK 的市场分析代理创建
		20 代理参数优化与行为配置详解
		21 复杂市场分析任务的 Handoffs 设计
		22 Guardrails 安全机制在市场敏感数据中的应用
		23 市场分析多代理系统的协同工作流设计
		24 Google ADK 在市场分析中的应用架构
		25 复杂状态管理与条件分支设计
		26 动态分析工作流的实时调整机制
		27 自定义市场分析节点与边的实现
	高级工具接入与定制化开发	28 市场分析任务的层级分解策略
		28 专业市场分析工具接口规范化
		29 数据可视化工具集成与交互设计
		30 自定义 NL2SQL 市场数据查询工具
		31 自动化报告生成工具开发实战
		32 子任务协调与结果整合机制
		33 分布式任务处理与资源优化
	多模态数据处理引擎	34 市场图表识别与数据提取工具开发
		35 视频内容分析与关键信息提取
		36 多模态市场信息的语义整合方法
		37 音频市场资料的转录与分析流程
		38 跨模态一致性检验与冲突解决
	性能优化与部署方案	39 InsightGen 性能瓶颈分析与调优

		40 缓存策略设计与实现
		41 并行处理与异步执行优化
		42 Docker 容器化部署完整流程
		43 云服务与本地部署混合方案
实战项目四：AI 编程与数据分析 Agent 项目开发实战		
【Part 23】从零搭建多功能智能体：MateGen	**MateGen 多功能智能体项目设计**	01 通用智能体应用前景介绍 02 MateGen 项目功能整体介绍 03 第一代 MateGen 项目架构、功能与代码回顾 04 市场 Agent 开发框架功能回顾 05 Llama-index、AWEL 开发框架功能介绍 06 本地代码解释器、NL2SQL 功能介绍 07 智能体自主意识介绍
	MateGen WorkFlow 设计与开发	08 大模型工作流 WorkFlow 概念介绍 09 MateGen WorkFlow 功能规划 10 持久化线程设计与消息队列设计 11 多对话场景切换与中继修改功能开发 12 线程状态器设计与开发 13 基于 Function calling 的外部工具执行器开发 14 多代理模式下消息队列审查流程 15 用户意图识别与 MateGen 自主意识设计 16 MateGen 身份设定与自主意识提示词模板
	MateGen 关键环节提示词模板设计	17 借助提示工程自动编写外部函数 18 搭建基于大模型的函数功能验证流程 19 借助编程示例 Few-shot 提高大模型编程稳定性 20 借助多步提示法拆解用户需求 21 基于需求拆解结果自动编写 Prompt 22 基于 LtM 提示及需求拆解结果构建 Few-shot 23 基于多段提示大幅提高大模型编程稳定性 24 历史编程提示词汇总与本次存储

		25 【实战】全自动拆解需求与多段编程函数编写与应用 26 【实战】基于人类反馈的提示正例积累与应用 27 【实战】提示学习在 MateGen 中的实现方法
	MateGen 开发流程	28 Instructions 功能设计与提示词优化 29 外部函数执行流程与自动 debug 方法 30 无限多轮对话功能设计与开发 31 【知识库检索功能开发】借助 FuFan-Chat 构建 RAG 引擎 32 【知识库检索功能开发】代码文档、md 文档检索增强 33 【视觉能力开发】借助多模态大模型搭建视觉实现流程 34 【联网搜索功能开发】谷歌搜索 API 接入流程 35 【联网搜索功能开发】Kaggle API 接入流程 36 【联网搜索功能开发】爬虫设计与搜索知识库构建 37 【NL2SQL 功能开发】连接本地 MySQL 服务流程 38 【NL2SQL 功能开发】数据字典检索与 SQL 代码 debug 流程 39 【本地代码解释器功能开发】连接本地 Python 环境 40 【本地代码解释器功能开发】交互式编程逻辑开发 41 Action 功能开发：模型替换与日志文档 42 Action 功能开发创建核心主类，集成其他模块功能 43 MateGen 性能测试与自适应微调 44 MateGen 前端展示接口对接逻辑设计与开发