AI基础知识

- ollama 安装和使用
- windows 系统下 mobaxterm 运行 ollama
- ollama 常用命令和操作

ollama 安装和使用

Ollama 安装与使用指南

Ollama 是一个强大的工具，允许你在本地运行各种大语言模型（**LLM**），如 LLaMA、Mistral、Gemma 等。以下是详细的安装和使用教程，涵盖 **Windows**、**macOS**、**Linux** 系统。

---

1. 安装 **Ollama**

□□ 方法 **1**：直接下载安装（推荐）

Windows / macOS

1. 访问 **Ollama** 官网，下载对应系统的安装包。
2. 运行安装程序，按照提示完成安装。
3. 安装完成后，打开终端（Windows 用 CMD/PowerShell，macOS 用 Terminal），输入：

   ```
   ollama --version
   ```

   如果显示版本号（如 `v0.1.30`），说明安装成功。

Linux（**Ubuntu/Debian/Rocky Linux**）

```
# 使用 curl 下载并安装
curl -fsSL https://ollama.com/install.sh | sh
```

安装完成后，运行：

```
ollama serve & # 启动服务（后台运行）
```

□□ 方法 **2**：使用 **Docker** 安装（适合开发者）

```
# 拉取 Ollama 镜像
docker pull ollama/ollama

# 运行容器（数据持久化到本地）
docker run -d -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama
```

- `-p 11434:11434`: Ollama 默认 API 端口。
- `-v ollama:/root/.ollama`: 存储模型数据，避免容器删除后丢失。

---

2. 下载模型

Ollama 支持多种模型，如:

- `llama2`（Meta 开源）
- `mistral`（轻量高效）
- `gemma`（Google 轻量级）
- `phi`（微软小模型）
- `qwen`（阿里通义千问）

◫ 下载模型

```
ollama pull llama2  # 下载 LLaMA 2
ollama pull mistral # 下载 Mistral
ollama pull gemma   # 下载 Google Gemma
```

⚠ 如果下载慢，可以使用代理:

```
export HTTP_PROXY=http://127.0.0.1:7890  # 替换为你的代理
export HTTPS_PROXY=http://127.0.0.1:7890
ollama pull llama2
```

---

3. 运行模型

◫ 方式 1：命令行交互

```
ollama run llama2  # 运行 LLaMA 2
```

输入问题，如：

```
>>> 用中文写一首关于春天的诗
```

模型会生成回答，按 `Ctrl+D` 退出。

方式 2：API 调用

Ollama 提供 REST API（默认端口 `11434`），可以用 `curl` 或 Python 调用：

```
curl -X POST http://localhost:11434/api/generate -d '{
  "model": "llama2",
  "prompt": "你好，你是谁？"
}'
```

Python 示例：

```
import requests

response = requests.post(
    "http://localhost:11434/api/generate",
    json={"model": "llama2", "prompt": "解释一下人工智能"}
)
print(response.text)
```

---

4. 自定义模型

你可以基于现有模型微调（Fine-tuning）：

1. 创建 `Modelfile`：

   ```
   FROM llama2
   SYSTEM """你是一个专业的中文助手，回答要简洁。"""
   ```

2. 构建自定义模型：

   ```
   ollama create my-llama -f Modelfile
   ```

3. 运行：

```
ollama run my-llama
```

---

5. 常见问题

□ □□ 1：`ollama` 命令找不到

- **Windows**：检查是否添加到了 `PATH`（安装时勾选）。
- **Linux/macOS**：手动添加环境变量：

```
export PATH=$PATH:/usr/local/bin
```

□□ 问题 2：下载模型太慢

- 方法 **1**：使用代理（见上文）。
- 方法 **2**：手动下载模型文件，放入 `~/.ollama/models`（Linux/macOS）或 `C:\Users\<用户名>\.ollama\models`（Windows）。

□□ 问题 3：端口冲突

如果 `11434` 端口被占用，可以修改端口：

```
OLLAMA_HOST=0.0.0.0:11435 ollama serve  # 改用 11435
```

---

6. 进阶用法

□□ 管理模型

```
ollama list      # 查看已安装的模型
ollama rm llama2  # 删除模型
```

⚙ 配置 GPU 加速

Ollama 支持 **NVIDIA CUDA** 和 **Apple Metal** 加速：

- **Linux (NVIDIA)**：安装 NVIDIA Container Toolkit，然后：

  ```
  docker run --gpus all -p 11434:11434 ollama/ollama
  ```

- **macOS (Metal)**：自动启用，无需额外配置。

---

7. 总结

操作	命令示例
安装 Ollama	`curl -fsSL https://ollama.com/install.sh | sh`
下载模型	`ollama pull llama2`
运行模型	`ollama run mistral`
调用 API	`curl -X POST http://localhost:11434/api/generate`
自定义模型	`ollama create my-model -f Modelfile`
查看已安装模型	`ollama list`

□□□□□□□□□□□□□□□□□□□ 如需更多模型，可查看 **Ollama** 官方库。

windows 系统下 mobaxterm 运行 ollama

在 Windows 系统下使用 **MobaXterm** 运行 **Ollama** 需要结合其类 Linux 终端环境进行操作。以下是详细步骤和注意事项：

---

1. 环境准备

(1) 安装 Ollama

- 官方安装：

  访问 Ollama 官网 下载 Windows 版安装包，默认安装后会自动添加到系统环境变量。

- 验证安装：

  在 MobaXterm 终端输入：

  ```
  ollama --version
  ```

  若提示命令不存在，需手动添加安装路径（如 `C:\Program Files\Ollama`）到 MobaXterm 的 `PATH` 中。

(2) 配置 MobaXterm

- 启用 **WSL** 集成（推荐）：

  MobaXterm 支持 WSL2，可原生运行 Linux 版 Ollama（性能更优）：

  ```
  wsl --install  # 确保已启用 WSL2
  wsl
  curl -fsSL https://ollama.com/install.sh | sh  # 在 WSL内安装
  ```

- 纯 **Windows** 模式：

  若需直接调用 Windows 版 Ollama，需在 MobaXterm 的 `Settings > Configuration > Terminal` 中勾选 **"Use Windows PATH"**。

---

2. 运行 Ollama

(1) 下载模型

```
ollama pull phi  # 示例: 下载Phi-2模型（约5GB）
ollama pull llama2:7b  # 下载Llama 2 7B
```

(2) 启动交互对话

```
ollama run phi "如何用Python读取Excel文件?"
```

- 快捷键：
  - `Ctrl+D` 退出交互模式
  - `Ctrl+C` 中断生成

(3) 后台服务模式

```
ollama serve  # 启动本地API服务（默认端口11434）
```

- 调用 **API**（另开终端）：

  ```
  curl http://localhost:11434/api/generate -d '{
    "model": "phi",
    "prompt": "解释量子计算的基本原理"
  }'
  ```

---

3. 常见问题解决

问题 1：GPU 未调用

- 现象：推理速度慢，任务管理器显示GPU利用率低。
- 解决：
  1. 确认已安装NVIDIA驱动+CUDA Toolkit。
  2. 显存不足时添加 `--num-gpu 1`（需Ollama版本≥0.1.20）：

     ```
     ollama run --num-gpu 1 phi
     ```

问题 2：端口冲突

- 修改默认端口：

```
set OLLAMA_HOST=0.0.0.0:12345  # 改为12345端口
ollama serve
```

问题3：中文支持差

- 加载中文优化模型：

```
ollama pull qwen:7b  # 阿里千问7B中文模型
ollama run qwen "写一篇关于人工智能的短文"
```

4. 性能优化建议

- 量化模型：减少显存占用（适合RTX 3060等消费级GPU）：

```
ollama pull llama2:7b-q4_0  # 4-bit量化版
```

- 限制**CPU**线程（避免卡顿）：

```
set OLLAMA_NUM_THREADS=4  # 限制为4线程
ollama run phi
```

5. 扩展功能

- 与 **Python** 集成：

```
import requests
response = requests.post(
    "http://localhost:11434/api/generate",
    json={"model": "phi", "prompt": "Python冒泡排序代码"}
)
print(response.json()["response"])
```

- **Docker** 部署（需启用MobaXterm的Docker插件）：

```
docker run -d -p 11434:11434 ollama/ollama
```

通过以上步骤，可在 MobaXterm 中高效运行 Ollama。如需针对特定场景（如长文本处理）进一步调优，可调整模型参数或结合 vLLM 等加速库。

ollama 常用命令和操作

以下是 **Ollama** 的常用命令和操作指南，涵盖模型管理、运行配置、 API 使用等核心功能，适用于本地大模型开发和测试：

---

1. 模型管理

下载模型

```
ollama pull <模型名    >[:版本  ]  # 默认下载最新版
```

常用模型示例 ：

```
ollama pull llama2        # Llama 2 7B
ollama pull phi           # Microsoft Phi-2 (2.7B)
ollama pull mistral       # Mistral 7B
ollama pull qwen:7b       # 阿里千问中文模型
```

查看已安装模型

```
ollama list
```

输出示例：

```
NAME          ID            SIZE    MODIFIED
llama2:latest   e6a7b3b4d5e6    3.8 GB  2 days ago
phi:latest      f1a2b3c4d5e6    1.9 GB  5 hours ago
```

删除模型

```
ollama rm <模型名    >
```

---

2. 运行模型

交互式对话

```
ollama run <模型名> "你的提示词"
```

示例：

```
ollama run llama2 "用 Python实现快速排序"
```

持续对话

直接运行不输入提示词，进入多轮对话模式：

```
ollama run phi
```

输入 `/bye` 退出对话。

带参数运行

```
ollama run --temperature 0.7 --num_ctx 2048 mistral
```

常用参数：

- `--temperature`：控制随机性（0-1，越高越有创意）
- `--num_ctx`：上下文长度（默认 2048）
- `--seed`：设置随机种子（用于复现结果）

---

3. 高级配置

启动本地 API服务

```
ollama serve
```

默认监听 `127.0.0.1:11434`，可通过 HTTP调用：

```
curl http://localhost:11434/api/generate -d '{
  "model": "llama2",
  "prompt": "为什么天空是蓝色的?"
}'
```

GPU加速

```
ollama run --gpu llama2 # 启用 NVIDIA GPU
```

需确保:

1. 已安装 NVIDIA驱动和 CUDA
2. Ollama版本 ≥0.1.20

量化模型运行

```
ollama pull llama2:7b-q4_0 # 下载 4-bit量化版
ollama run llama2:7b-q4_0 # 显存占用降低 50%
```

---

4. 自定义模型

从 **Modelfile**创建

1. 创建 `Modelfile`:
   ```
   FROM llama2
   PARAMETER temperature 0.8
   SYSTEM """
   你是一个专业的技术助手，回答需简洁准确。
   """
   ```
2. 构建自定义模型:
   ```
   ollama create my-llama -f Modelfile
   ```
3. 运行:
   ```
   ollama run my-llama
   ```

导入 /导出模型

```
ollama export llama2 llama2.tar # 导出
ollama import llama2.tar        # 导入
```

---

5. 实用技巧

查看模型信息

```
ollama show <模型名> --modelfile  # 显示模型配置
ollama show --license llama2  # 查看许可证
```

多模型并行

```
# 终端 1
ollama serve

# 终端 2
ollama run phi "解释量子力学"

# 终端 3
ollama run mistral "写一首英文诗"
```

日志调试

```
ollama serve > ollama.log 2>&1  # 输出日志到文件
```

---

6. 常见问题解决

内存不足

- 方案 **1**：使用量化模型（如 `q4` 版本）
- 方案 **2**：限制 CPU线程：

  ```
  OLLAMA_NUM_THREADS=4 ollama run llama2
  ```

中文输出不佳

1. 使用支持中文的模型：

   ```
   ollama pull qwen:7b
   ```

2. 明确提示词：

   ```
   ollama run qwen "用中文回答：如何学习机器学习？"
   ```

端口冲突

修改默认端口:

```
OLLAMA_HOST=0.0.0.0:12345 ollama serve
```

---

7. 常用模型推荐

模型名称	参数量	特点	适用场景
Llama 2	7B/13B	通用性强，英文优	文本生成、问答
Phi-2	2.7B	逻辑推理强，代码生成好	技术问答、代码辅助
Mistral	7B	性能接近 Llama 13B	长文本处理
Qwen	7B	中文支持优秀	中文对话、翻译
Gemma	2B/7B	Google轻量级模型	移动端部署

通过以上命令，您可以高效管理本地大语言模型。如需扩展功能（如接入 LangChain或 Web界面），可结合 Ollama的 API进一步开发。