LLM 基本调用参数

ChatOpenAI()

Completion Params

- **model**:

  > 指定 OpenAI 模型的名称，用于生成响应。
  >
  > - 缺省是 gpt-3.5-turbo（约 20RMB/百万汉字，2024-11-15，数据来源：https://agicto.com/model，下同）。
  > - 日常推荐 gpt-4o-mini（约 1RMB/百万汉字）。
  > - 复杂问题和多模态处理用 gpt-4o（约 36RMB/百万汉字）。

- **temperature**:

  > 生成内容的温度系数，用于控制响应的随机性。值越高，生成的内容越随机；值越低，生成的内容越保守。
  >
  > - 0 = 完全机械化；
  > - 0.7 = 正常；
  > - 1.5 会出现不可预期的结果。

- **max_tokens**:

  > 最大生成的 token 数量。可以限制输出的长度。

> 可有效防止出现用户输入类似“请帮我写一篇十万字的小说”时产生巨大费用。

Client Params

- **timeout**:

  > 请求超时时间，单位秒。一般正常的 LLM 调用会在 5~10 秒内完成，但某些复杂处理可长达 30 秒（图形处理）甚至更长（实际应用中遇到过超过 100 秒的情况）。

- **max_retries**:

  > 最大重试次数。在请求失败时，可以设置最大重试次数以确保请求成功。

- **api_key**:

  > OpenAI API 的密钥。如果未传入，将从环境变量 `OPENAI_API_KEY`读取。

- **base_url**:

  > API 请求的基础 URL。如果未传入，将从环境变量 `OPENAI_API_BASE`读取。

- **organization**:

  > OpenAI 组织 ID。如果未传入，将从环境变量 `OPENAI_ORG_ID`读取。

ChatOpenAI.invoke()

Invoke 参数

- **prompt**:

  > 用户的输入内容，作为生成响应的基础。通常为字符串或包含多个字符串的数组。

- **stop**:

> 一个字符串或字符串数组，指定生成响应时的结束标志。LLM 在遇到这些标志后将停止生成。

- **n**:

> 生成的响应数量。可以指定希望生成的响应数量，默认为 1。增加生成数量可能增加 API 调用的成本。

- **stream**:

> 布尔值，指定是否以流的方式返回响应。设置为 `True` 时，生成的内容将分块返回。

---

streaming 流式输出

stream 可以把非常长的答案（如重构一个一千行的代码）按 token 逐个输出，而不必等待整个处理过程完成。因此可以提供更快的响应速度。

> 在 Chatgpt 中看到的逐个输出的效果就是这样实现的。

示例代码

以下是一个最简单的示例，使用 `stream=True` 参数生成“Hello |world|!” 样式的流式输出：

```
from openai import ChatOpenAI

# 创建 ChatOpenAI 实例
llm = ChatOpenAI(model="gpt-4o-mini", stream=True)

# 使用流式输出打印 Hello |world|!
def print_streamed_response():
    prompt = "Please write 'Hello world!' with each word separated by a | symbol."
    response = llm.invoke(prompt=prompt)

    for chunk in response:
        print(chunk, end='|')

print_streamed_response()
```