Lesson 5

Tools

Haoxiang Sun @ RUC CS

使用 GPT 与 Copilot 来编写复杂 Python 代码

- 市面上常见的模型：lmsys.org
- OpenAI 公司：**GPT 4o**, GPT 4o-mini, **o1-preview**, o1-mini
- Anthropic 公司：**Claude 3.5 Sonnet**
- Google 公司：**Gemini Experimental 1121**
- 通义千问：Qwen 2.5 72B Instruct, QwQ 32B, etc.
- 深度求索：deepseek-coder, deepseek-math
- 智谱：GLM-4-Plus
- 经典开源模型系列 Meta Llama：Llama 3.2 90B

常见的代码 AI 工具

- 直接与各大公司的模型进行对话
- GitHub Copilot （使用学生优惠，可以免费使用）
- Cursor

- 主打网页搜索的 Perplexity

Prompt Engineering

- 在与国外的模型进行对话时，尽量使用英语
- 准确清晰地描述需求
- Reflection（需要用户自己有辨别能力）
- 实践中真实有效的方法：
- 1. Chain of Thought: “Let’s think step by step.”
- Chain-of-Thought Prompting Elicits Reasoning in LLMs. (2201.11903, Google)
- 2. Few shot: 示范
- Language Models are Few-Shot Learners. (2005.14165, OpenAI)

总结

- 可以多问 GPT 有关 Python 的各种流行包的问题，他们非常擅长 Python 编程
- 以官方文档为准！LLM 的幻觉问题仍难以抑制

Numpy

- 支持大规模矩阵运算与数组运算
- 很多高性能计算包是用 C/C++ 实现的，调用它们可以得到更快的计算速度
- 有高性能计算卡？可以尝试 CuPy（基于CUDA包，cuBLAS，etc）
- （由于数据转移等操作，CuPy可能更慢）

Numpy

demo

```
import numpy as np

a = np.array([1, 2, 3])
print(a.shape)
print(a)
print(type(a))

b = np.array([[1, 2, 3], [4, 5, 6]])
print(b.shape)
print(b)
print(b[1])
print(b[1, 1])

print(np.zeros((2, 2)))
print(np.ones((1, 2)))
print(np.full((2, 2), 7))
print(np.eye(4))
print(np.random.random((2, 2)))
```

Numpy

demo

```
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
print(a[:2, 1:3])
print(a[:, 3:1:-1])
b = a[:2, 1:3]  # A view into the same data
b[0][0] = -10
print(a)

a = [1, 2, 3, 4]
b = a[1:3]
b[0] = -10
print(a)

a = np.array([[1, 2], [3, 4], [5, 6]])
print(a[[0, 1, 2], [0, 1, 0]])
a[[0, 1, 2], [0, 1, 0]] -= 50
print(a)

b = a > 2
print(b)
print(a[b])
```

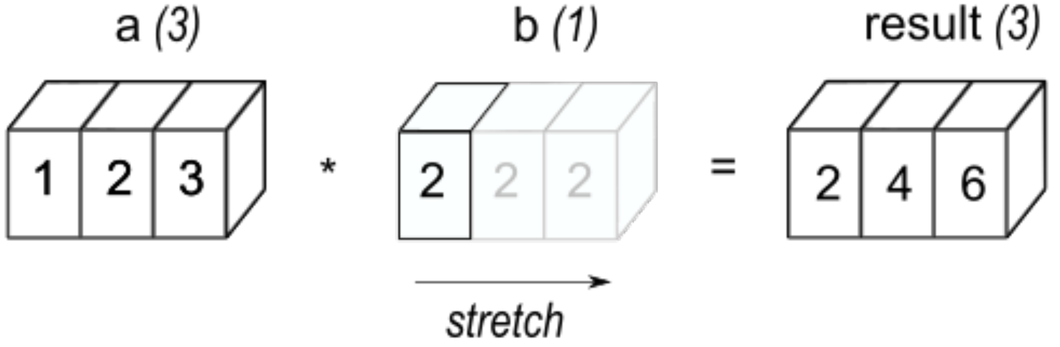
Numpy

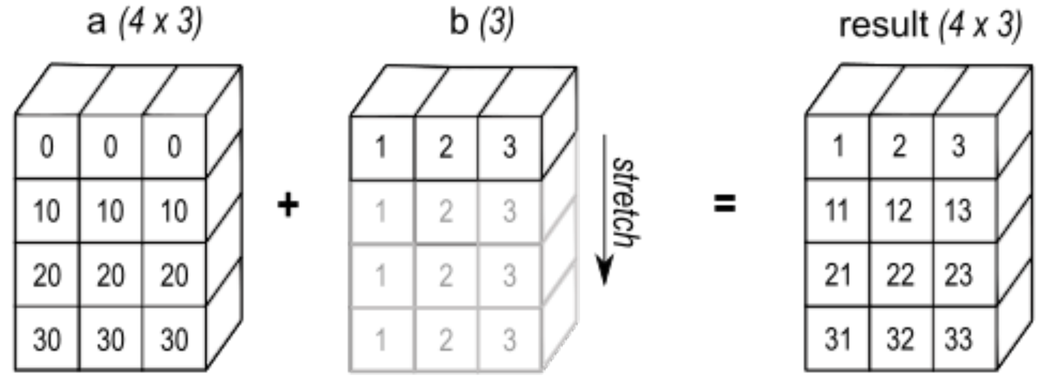
demo

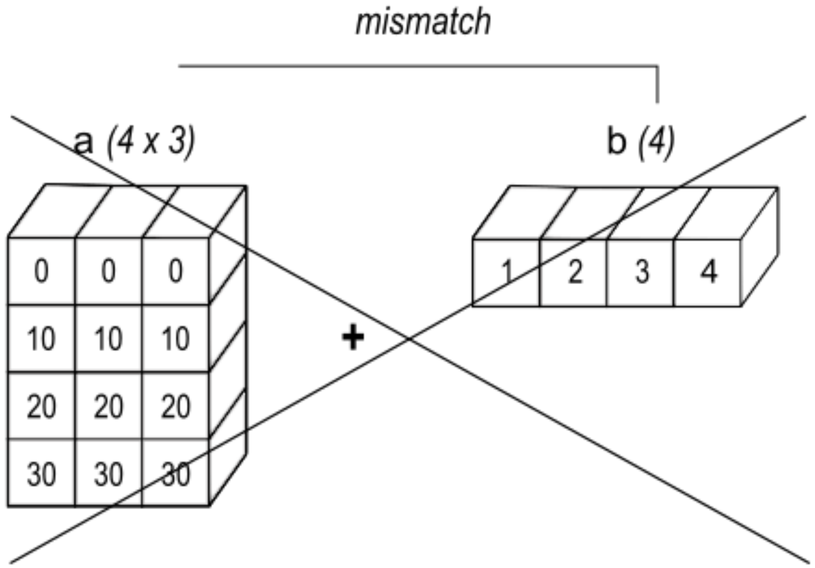
```
x = np.array([1, 2])
print(x.dtype)
x = np.array([1.0, 2.0])
print(x.dtype)
y = np.array([10.0, 12.5], dtype=np.int64)
print(y.dtype)
print(y)

print(x + y)
print(x - y)
print(x * y)
print(x / y)
print(np.sqrt(x))
print(np.log(x))

x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])
print(x * y)
print(x @ y)
print(np.array([[1, 2, 3]]) @ np.array([1, 2, 3]))

print(np.sum(x))
print(np.sum(x, axis=0))
print(np.sum(x, axis=1))
print(x.T)
print(np.array([[1, 2, 3], [4, 5, 6]]).T)
```

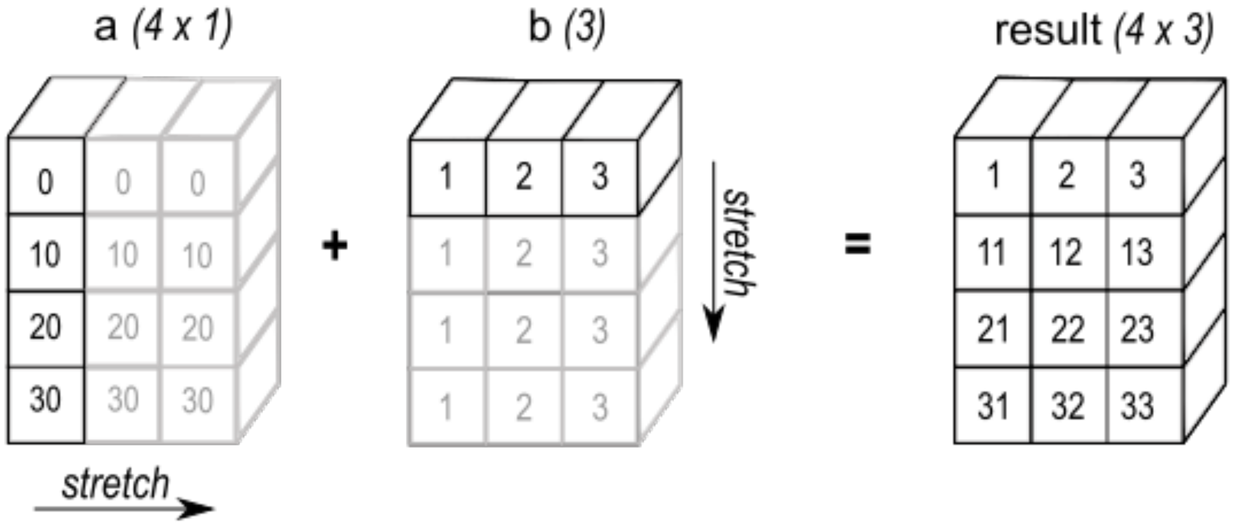
Numpy 广播机制

print(np.array([[1, 2, 3, 4]]) * 2)









Playwright

- HTML文件的格式?

- 可以用来爬取动态网页
- 支持 headless 模式
- 可以进行网页截图等操作

demo

```python
import json
import os
from playwright.async_api import async_playwright

async def run(page, k):
    url = f"https://loj.ac/s/{k.__str__()}"
    await page.goto(url)
    await page.wait_for_timeout(8000)  # 等待 8 秒

    code_text = await page.query_selector_all("._codeBoxContent_122zh_12")
    status_text = await page.query_selector_all(".statustext")

    if len(code_text) == 0 or len(status_text) == 0:
        return None

    status = await status_text[0].inner_text()
    code = await code_text[0].inner_text()

    # 判断是不是 Accepted
    if status.strip() == "Accepted":
        return code

    return None

playwright = await async_playwright().start()
browser = await playwright.chromium.launch(headless=False)
page = await browser.new_page()

with open(f"./data/codes.jsonl", "w", encoding="utf-8") as f:
    for k in range(2196563, 2196613): # 选择一段爬取的评测结果的 id
        result = await run(page, k)
        if result is not None:
            f.write(repr(result))
            f.write("\n")

await browser.close()
await playwright.stop()
```