

GPT-4里套娃LLaMA 2，OpenAI创始成员周末爆改「羊驼宝宝」，GitHub一日千星

量子位 · 2023-07-24 03:56

关注

C语言单文件500行代码，苹果M1笔记本直接跑
大神仅花一个周末训练微型**LLaMA 2**，并移植到**C语言**。

推理代码只有500行，在**苹果M1笔记本**上做到**每秒输出98个token**。

作者是OpenAI创始成员**Andrej Karpathy**，他把这个项目叫做**Baby LLaMA 2**（羊驼宝宝）。

**Andrej Karpathy** ✓

@karpathy

My fun weekend hack: llama2.c 🦙 😊

[github.com/karpathy/llama...](https://github.com/karpathy/llama2.c)

Lets you train a baby Llama 2 model in PyTorch, then inference it with one 500-line file with no dependencies, in pure C. My pretrained model (on TinyStories) samples stories in fp32 at 18 tok/s on my MacBook Air M1 CPU.

翻译推文



下午11:52 · 2023年7月23日 · 44.6万 查看

虽然它只有**1500万参数**，下载下来也只有**58MB**，但是已经能流畅讲故事。

You'll see text stream. On my M1 MacBook Air this runs at ~100 tokens/s, not bad for super naive fp32 single-threaded C code. Sample output:

Once upon a time, there was a boy named Timmy. Timmy loved to play sports with his friends. He was very good at throwing and catching balls. One day, Timmy's mom gave him a new shirt to wear to a party. Timmy thought it was impressive and asked his mom to explain what a shirt could be for. "A shirt is like a special suit for a basketball game," his mom said. Timmy was happy to hear that and put on his new shirt. He felt like a soldier going to the army and shouting. From that day on, Timmy wore his new shirt every time he played sports with his friends at the party. Once upon a time, there was a little girl named Lily. She loved to play outside with her friends. One day, Lily and her friend Emma were playing with a ball. Emma threw the ball too hard and it hit Lily's face. Lily felt embarrassed and didn't want to play anymore. Emma asked Lily what was wrong, and Lily told her about her memory. Emma told Lily that she was embarrassed because she had thrown the ball too hard. Lily felt bad achieved tok/s: 98.746993347843922



所有推理代码可以放在**C语言单文件**上，**没有任何依赖**，除了能在笔记本CPU上跑，还迅速被网友接力开发出了各种玩法。

llama.cpp的作者**Georgi Gerganov**搞出了**直接在浏览器里运行**的版本。



llama2.c compiled with Emscripten to run in a web page

generating ...

Once upon a time, there was a little boy named Timmy. Timmy loved to play in the park with his friends. One day, Timmy fell and hurt his knee. He cried out in pain as it hurt too much. His friend Billy came over and asked, "Are you okay, Timmy?" Timmy's mommy came over and saw his knee. She said, "Don't worry, Timmy. I am very helpful. I can heal your knee." Billy helped Timmy to his mommy and she put a bandage on his knee. It felt better! Timmy was surprised that Billy was so helpful. He said, "Thank you, Billy. You really know a lot about cars and airplanes." Billy smiled and said, "Yes, Timmy. Miss Sarah taught us that it's important to ask for help when we need it. We should always try to help others if they are in need." From that day on, Timmy and Billy always looked for ways to help others, even if they were different from other musicians. They knew that being helpful and kind was a kind thing to do. Once upon a time, there was a little Samantha-licks him.Ted upst shoulders.Mic happen. lwatting towistles happening toddling the next time ago. I fall.Ted upsticks upst heaven. It washadowed again. And foreshvizing that



提示工程师Alex Volkov甚至做到了在GPT-4代码解释器里跑Baby LLaMA 2。



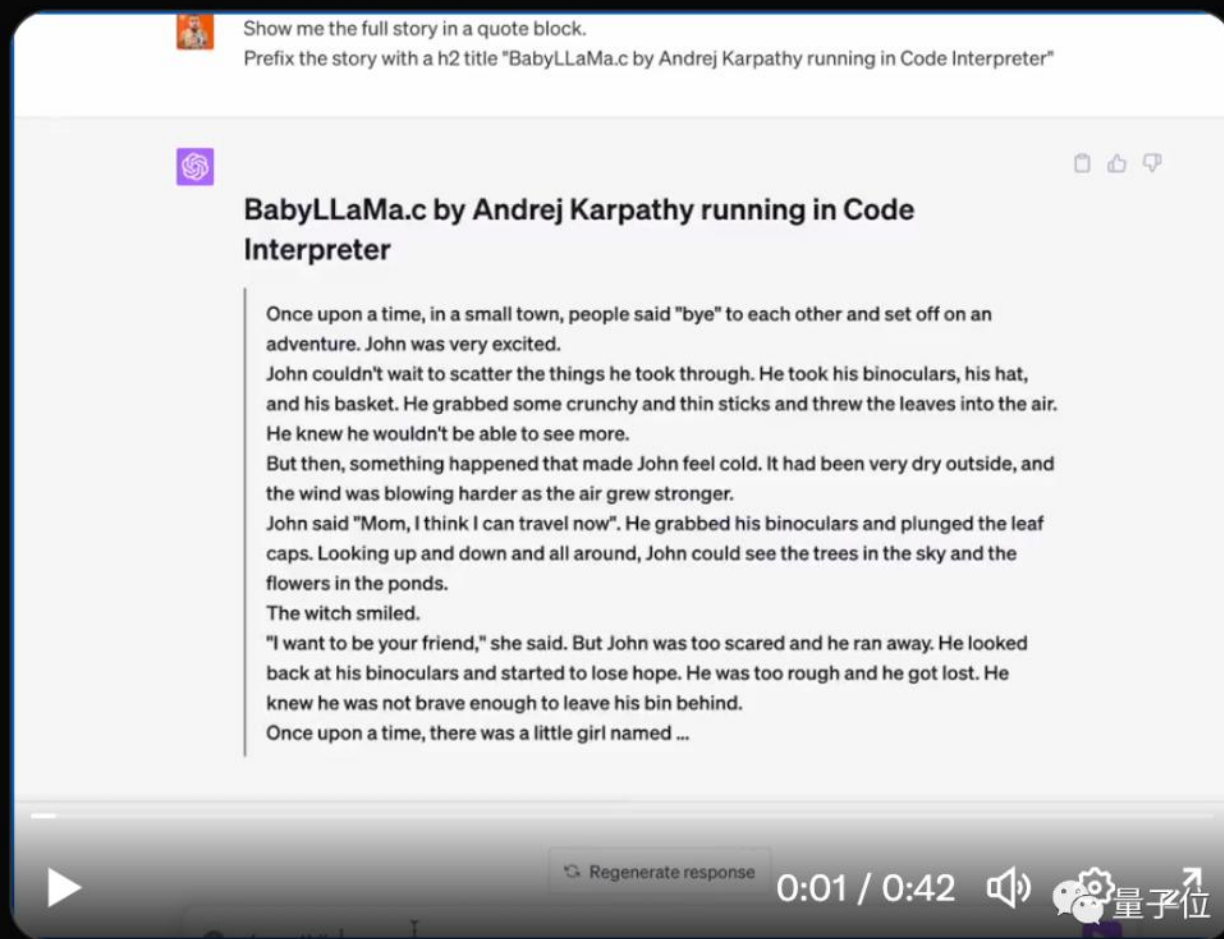
Alex Volkov - targum.video @altryne · 4小时

#codeinterpretercan run LLaMa.c by @karpathy 🙌

Steps:

- Follow instructions in readme on a linux machine
- Create a venv and install sentencepiece
- Zip all of it (weights, env and code)
- Be nice to code interpreter so it won't complain 😂

Anyone wants the session link? ;)



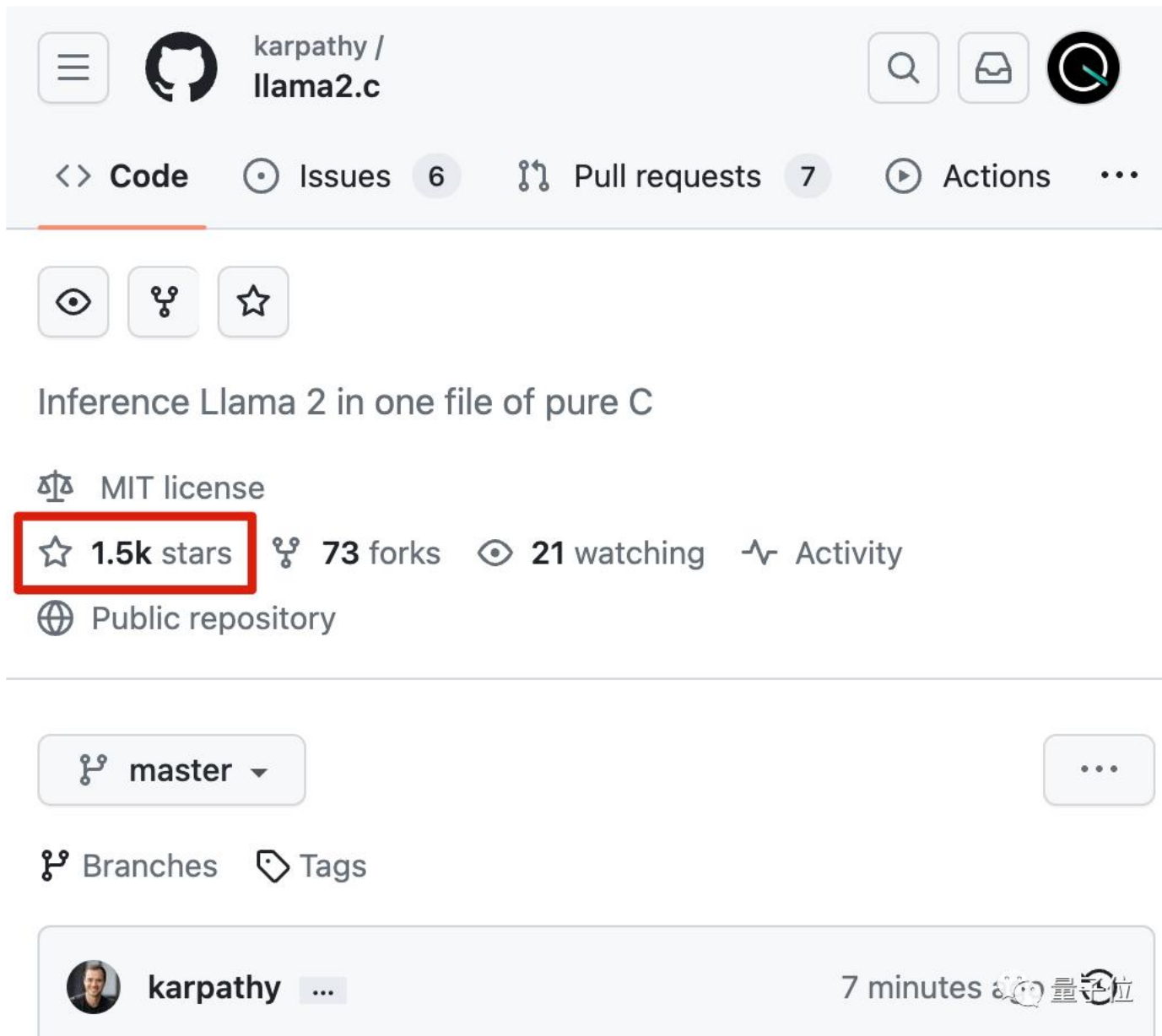
大模型套娃小模型，成了。

羊驼宝宝诞生记

据Karpathy分享，做这个项目的灵感正是来自llama.cpp。

训练代码来自之前他自己开发的nanoGPT，并修改成LLaMA 2架构。

推理代码直接开源在GitHub上了，不到24小时就狂揽1500+星。



karpathy / llama2.c

<> Code Issues 6 Pull requests 7 Actions ...

👁️ 🍴 ⭐

Inference Llama 2 in one file of pure C


📄 MIT license

⭐ 1.5k stars 🍴 73 forks 👁️ 21 watching 📈 Activity

🌐 Public repository

🔗 master ...

🔗 Branches 🏷️ Tags

 karpathy ... 7 minutes 🗨️ 📈 📊

训练数据集TinyStories则来自微软前一阵的研究。

2023新视野数学奖得主Ronen Eldan、2023斯隆研究奖得主李远志联手，**验证了1000万参数以下的小模型，在垂直数据上训练也可以学会正确的语法、生成流畅的故事、甚至获得推理能力。**

TinyStories: How Small Can Language Models Be and Still Speak Coherent English?

Ronen Eldan, Yuanzhi Li

Language models (LMs) are powerful tools for natural language processing, but they often struggle to produce coherent and fluent text when they are small. Models with around 125M parameters such as GPT-Neo (small) or GPT-2 (small) can rarely generate coherent and consistent English text beyond a few words even after extensive training. This raises the question of whether the emergence of the ability to produce coherent English text only occurs at larger scales (with hundreds of millions of parameters or more) and complex architectures (with many layers of global attention).

In this work, we introduce TinyStories, a synthetic dataset of short stories that only contain words that a typical 3 to 4-year-olds usually understand, generated by GPT-3.5 and GPT-4. We show that TinyStories can be used to train and evaluate LMs that are much smaller than the state-of-the-art models (below 10 million total parameters), or have much simpler architectures (with only one transformer block), yet still produce fluent and consistent stories with several paragraphs that are diverse and have almost perfect grammar, and demonstrate reasoning capabilities.

We also introduce a new paradigm for the evaluation of language models: We suggest a framework which uses GPT-4 to grade the content generated by these models as if those were stories written by students and graded by a (human) teacher. This new paradigm overcomes the flaws of standard benchmarks which often requires the model's output to be very structures, and moreover provides a multidimensional score for the model, providing scores for different capabilities such as grammar, creativity and consistency.

We hope that TinyStories can facilitate the development, analysis and research of LMs, especially for low-resource or specialized domains, and shed light on the emergence of language capabilities in LMs.

此外，开发过程中还有一个插曲。

Karpathy很久不写C语言已经生疏了，但是在GPT-4的帮助下，还是只用一个周末就完成了全部工作。



对此，英伟达科学家Jim Fan评价为：**现象级**。



最初，在CPU单线程运行、fp32推理精度下，Baby LLaMA 2每秒只能生成18个token。

在编译上使用一些优化技巧以后，直接提升到每秒98个token。



优化之路还未停止。

有人提出，可以通过GCC编译器的-funsafe-math-optimizations模式再次提速6倍。



除了编译方面外，也有人提议下一步增加LoRA、Flash Attention等模型层面流行的优化方法。

LoRA support? #11



psych0v0yager opened this issue 6 hours ago · 1 comment



psych0v0yager commented 6 hours ago

Are there any plans to offer LoRA support in the future? Currently I have been using this library (<https://github.com/cccntu/minLoRA>) with nanoGPT

By the way, huge fan of your videos. I loved the way you coded language models live as well as providing priceless intuition on all the core concepts. I would love to see a brief video on this repository as well as all the latest innovations in the llm space (alibi, rotary embeddings, flash attention, lora, quantization, etc)

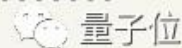


Baby LLaMA 2一路火到Hacker News社区，也引发了更多的讨论。

▲ gandalf 2 hours ago | prev | next [-]

Seems like this could be suitable for masochists like me who wish to run language models on retro computers :)

看来这可能适合像我这样希望在复古计算机上运行语言模型的受虐狂:)



有人提出，现在虽然只是一个概念验证，但本地运行的语言模型真的很令人兴奋。

虽然无法达到在云端GPU集群上托管的大模型的相同功能，但可以实现的玩法太多了。

This and the original is all absolutely awesome, it's obviously only a proof of concept with a tiny model, but *local first* LLMs are really exciting. I particularly love the idea of being able to build webapps with local inference.

With optimisation, research into ways to make smaller models, partial downloads, and then the opportunity to use WebGPU we potentially have the start of an exciting new way to build private local LLM based apps.

It's never going to be up to the same capabilities of hosted LLMs on massive clusters of top end GPUs, but there are so many use cases that this sort of thing will enable.

reply



在各种优化方法加持下，karpathy也透露已经开始尝试训练更大的模型，并表示：

70亿参数也许触手可及。

▲ karpathy 5 hours ago | next [-]

Yay fun to see it make its way to HN :) It turns out that my original checkpoint runs way faster than I expected (100 tok/s) on MacBook Air M1 with -O3 when compiling, so I am now training a bigger 44M model, which should still running interactively. Maybe the 7B Llama model is within reach... :thinking_emoji:



GitHub

<https://github.com/karpathy/llama2.c>

在浏览器运行Baby LLaMA 2

<https://ggerganov.com/llama2.c>

参考链接

[1]<https://twitter.com/karpathy/status/1683143097604243456>

[2]<https://twitter.com/gggerganov/status/1683174252990660610>

[3]<https://twitter.com/altryne/status/1683222517719384065>

[4]<https://news.ycombinator.com/item?id=36838051>