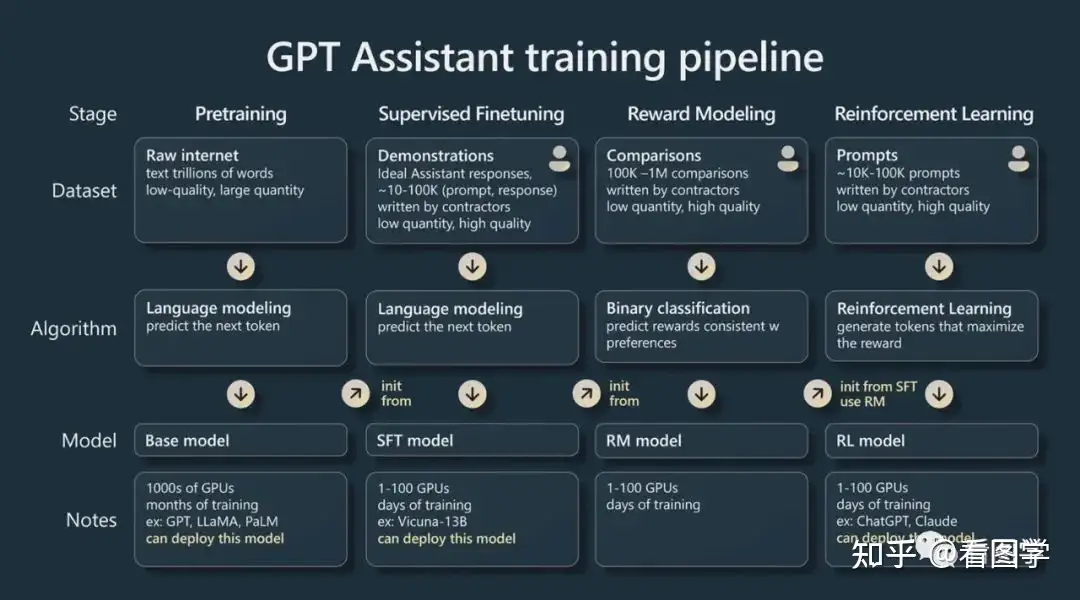
作者：看图学  
链接：https://www.zhihu.com/question/608820310/answer/3091336166  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

最近大模型发展卷的很，[王慧文](https://zhida.zhihu.com/search?q=%E7%8E%8B%E6%85%A7%E6%96%87&zhida_source=entity&is_preview=1" \t "_blank)都被整抑郁了。想要研究学习大模型，应该从哪里开始呢？

目前大模型发展生态最好的当属Meta的LLaMA模型。如果GPT系列是Windows操作系统(巧了，OpenAI的大东家目前就是微软)，那么LLaMA就是Linux。如果GPT系列是苹果手机，那么LLaMA就是安卓。如果你想基于大模型做一些事情，无论是创业还是研究，最好选择一个生态好的模型，毕竟有人用才有市场。

ChatGPT演化的路径如下图所示。



图片中分了4个阶段，但是第三个和第四个阶段一般都会放在一起，属于对齐阶段。所以一般会分成如下3个阶段：

* Stage 1: 预训练(Pretrain)
* Stage 2: 监督微调(SFT)
* Stage 3: 对齐(Reward Model + RLHF)

既然已经有了成功ChatGPT这一成功的案例，大家都想基于LLaMA把这条路再走一遍，以期望做出自己的ChatGPT。

所以**基于LLaMA的模型虽然很多，但是基本都可以放到上面3个框架当中**。本文就沿着预训练、监督微调、对齐(RW+RLHF)这一路径来梳理一下LLaMA生态中的各个模型。

主要是点出这些模型处在大模型训练的那一个阶段，以及都做了哪些创新性的工作，方便你根据自己的兴趣和资源来选择使用哪一个，对中文支持比较好的也都有注明。

## **Stage1 预训练: LLaMA 复现**

## **RedPajama**

* 参考LLaMA论文中的训练数据，收集并且开源可商用。
* [https://github.com/togethercomputer/RedPajama-Data](https://link.zhihu.com/?target=https://github.com/togethercomputer/RedPajama-Data" \t "_blank)

## **Baichuan-7B(支持中文)**

* 采用LLaMA的相同架构，在中文上做预训练。可商用。
* [王小川](https://zhida.zhihu.com/search?q=%E7%8E%8B%E5%B0%8F%E5%B7%9D&zhida_source=entity&is_preview=1" \t "_blank)这次做大模型的切入点其实挺不错的，绑定到LLaMA的生态上，然后在中文上有所突破。可能也在构思新三级火箭了吧。
* 目前Baichuan可以算是第一个LLaMA中文预训练模型，所以**后面的工作都可以在这上面都走一遍**，估计没多久Baichuan-Alapca, Baichuan-Vicuna就都出来了。
* [https://github.com/baichuan-inc/baichuan-7B](https://link.zhihu.com/?target=https://github.com/baichuan-inc/baichuan-7B" \t "_blank)

## **OpenLLaMA**

* 参考LLaMA的代码，在Apache 2.0 license下的重新实现和训练。使用了RedPajama训练集合。
* [https://github.com/openlm-research/open\_llama](https://link.zhihu.com/?target=https://github.com/openlm-research/open_llama" \t "_blank)

## **Lit-LLaMA ️**

* 参考LLaMA，在Apache 2.0 license下的只有代码的重新实现。同时支持加载原始LLaMA和OpenLLaMA的权重。
* [https://github.com/Lightning-AI/lit-llama](https://link.zhihu.com/?target=https://github.com/Lightning-AI/lit-llama" \t "_blank)

## **Stage 2: 监督微调**

因为预训练模型本质上还是个续写模型，所以并不能很好的满足人们的需求，所以监督微调的作用就是微调模型产生理想的回复。

在监督微调这里，大家目标都是一样的，但是做法有些不同，主要是有钱和没钱的区别。

有钱你可以[全参数微调](https://zhida.zhihu.com/search?q=%E5%85%A8%E5%8F%82%E6%95%B0%E5%BE%AE%E8%B0%83&zhida_source=entity&is_preview=1" \t "_blank)，没钱就只能使用一些低成本的方法，英文叫PEFT(Parameter-Efficient Fine-Tuning)。

PEFT确实是想我这种平民玩家的首选，但是有钱也可以用PEFT，它可以让你微调更大的模型。比如我们就只能玩玩10B的，有点小钱用PEFT玩个几十B的问题不大。

## **2.1 LLaMA + Instruction Finetuning(全量参数)**

### **Alpaca**

* llama7b + self-instruct数据指令微调。算是最早迈出LLaMA+SFT这一步的模型。最开始并没有提供权重，后来通过diff的方式给出，需要LLaMA原始模型才能恢复，github上有教程。
* 当时他们采用1张8卡A100(80G显存)，52k的数据，训练了3个小时。训练成本大概是100刀。
* [https://github.com/tatsu-lab/stanford\_alpaca](https://link.zhihu.com/?target=https://github.com/tatsu-lab/stanford_alpaca" \t "_blank)

### **Alpaca衍生模型**

* BELLE(支持中文): 最早是基于BLOOM的，后来也支持LLaMA [https://github.com/LianjiaTech/BELLE](https://link.zhihu.com/?target=https://github.com/LianjiaTech/BELLE" \t "_blank)
* openAlpaca: OpenLLaMA + databricks-dolly-15k dataset 进行指令微调 [https://github.com/yxuansu/OpenAlpaca](https://link.zhihu.com/?target=https://github.com/yxuansu/OpenAlpaca" \t "_blank)
* gpt4-x-alpaca: 用GPT4的数据微调，数据集为GPTeacher [https://huggingface.co/chavinlo/gpt4-x-alpaca](https://link.zhihu.com/?target=https://huggingface.co/chavinlo/gpt4-x-alpaca" \t "_blank)

### **Vicuna**

* llama13b + ShareGPT对话数据，微调
* 研发团队基于Vicuna发布了FastChat对话机器人。
* 和Alpaca一样，受协议限制，vicuna模型公布的权重也是个delta，每个参数要加上llama原来的权重才是模型权重。
* [https://github.com/lm-sys/FastChat](https://link.zhihu.com/?target=https://github.com/lm-sys/FastChat" \t "_blank)

### **Vicuna衍生模型**

* gpt4-x-vicuna-13b: 用GPT4的数据微调，数据集为GPTeacher [https://huggingface.co/NousResearch/gpt4-x-vicuna-13b](https://link.zhihu.com/?target=https://huggingface.co/NousResearch/gpt4-x-vicuna-13b" \t "_blank)

### **WizardLM**

* 采用了Evol-Instruct来构造指令，可以产生一些很难的指令.

1. 深度演化包括五种操作：添加约束、深化、具体化、增加推理步骤并使输入复杂化。
2. In-breadth Evolving 是突变，即根据给定的指令生成全新的指令
3. 进化是通过提示+LLM来实现的。

· [https://github.com/nlpxucan/WizardLM](https://link.zhihu.com/?target=https://github.com/nlpxucan/WizardLM" \t "_blank)

### **TÜLU**

* 使用LLaMA + Human/GPT data mix 微调
* 验证了很多结论，论文值得一看。[https://arxiv.org/abs/2306.04751](https://link.zhihu.com/?target=https://arxiv.org/abs/2306.04751" \t "_blank)
* [https://github.com/allenai/open-instruct](https://link.zhihu.com/?target=https://github.com/allenai/open-instruct" \t "_blank)

### **GPT4ALL**

* LLaMA用80w的[GPT3.5](https://zhida.zhihu.com/search?q=GPT3.5&zhida_source=entity&is_preview=1" \t "_blank)的数据(code, story, conversation)微调而来。
* [https://github.com/nomic-ai/gpt4all](https://link.zhihu.com/?target=https://github.com/nomic-ai/gpt4all" \t "_blank)

### **Koala**

* LLaMA13B基于ChatGPT Distillation Data和Open Source Data训练而来。
* 具体数据见下面：
  + [https://bair.berkeley.edu/blog/2023/04/03/koala/](https://link.zhihu.com/?target=https://bair.berkeley.edu/blog/2023/04/03/koala/" \t "_blank)

### **OpenBuddy(支持中文)**

* 基于LLaMA，Falcon, OpenLLaMA微调的，只说用了对话数据，细节没透漏。
* [https://github.com/OpenBuddy/OpenBuddy](https://link.zhihu.com/?target=https://github.com/OpenBuddy/OpenBuddy" \t "_blank)

### **Pygmalion 7B**

* 给予LLaMA微调，使用了不同来源的56MB 的对话数据，包含了人工和机器。
* [https://huggingface.co/PygmalionAI/pygmalion-7b](https://link.zhihu.com/?target=https://huggingface.co/PygmalionAI/pygmalion-7b" \t "_blank)

## **2.2 LLaMA + PEFT**

PEFT目前最流行的是LoRA，挺巧妙的架构，可以看看[https://arxiv.org/abs/2106.09685](https://link.zhihu.com/?target=https://arxiv.org/abs/2106.09685" \t "_blank)。

下面大多数的模型都是LLaMA+lora的架构，不只是文本，AIGC的头部网站[http://civitai.com](https://link.zhihu.com/?target=http://civitai.com" \t "_blank)上很多模型也都是基于lora的。

最近还出了QLoRA，在LoRA的基础上加入了量化，进一步降低显存的使用。[https://arxiv.org/abs/2305.14314](https://link.zhihu.com/?target=https://arxiv.org/abs/2305.14314" \t "_blank)。

### **Baize**

* LLaMA + Lora
* [https://github.com/project-baize/baize-chatbot](https://link.zhihu.com/?target=https://github.com/project-baize/baize-chatbot" \t "_blank)

### **LLaMA-Adapter**

* LLaMA + Adapter Layer
* [https://github.com/OpenGVLab/LLaMA-Adapter](https://link.zhihu.com/?target=https://github.com/OpenGVLab/LLaMA-Adapter" \t "_blank)

### **CalderaAI/30B-Lazarus**

* 似乎是多个LoRA的merge，但是没太公布太多细节。
* 在huggingface的leaderboard上排名还挺靠前。
* [https://huggingface.co/CalderaAI/30B-Lazarus](https://link.zhihu.com/?target=https://huggingface.co/CalderaAI/30B-Lazarus" \t "_blank)

### **Chinese-LLaMA-Alpaca(支持中文)**

* [https://arxiv.org/pdf/2304.08177.pdf](https://link.zhihu.com/?target=https://arxiv.org/pdf/2304.08177.pdf" \t "_blank)
* LLaMA + 扩词表 + lora
* Chinese LLaMA是属于局部参数预训练
  + Stage1: frozen encoder，只用来训练[Embedding层](https://zhida.zhihu.com/search?q=Embedding%E5%B1%82&zhida_source=entity&is_preview=1" \t "_blank)。
  + Stage2: 只训练Embedding, LM head, lora weights
* 在Chinese LLaMA的基础上，仿照Alpaca训练了Chinese Alpaca
* [https://github.com/ymcui/Chinese-LLaMA-Alpaca](https://link.zhihu.com/?target=https://github.com/ymcui/Chinese-LLaMA-Alpaca" \t "_blank)

### **Chinese-Vicuna(支持中文)**

* 基于：[https://github.com/tloen/alpaca-lora](https://link.zhihu.com/?target=https://github.com/tloen/alpaca-lora" \t "_blank)
* lora + 中文instruction数据
* chatv1的数据使用了50k中文指令+对话混合数据。
* 并没有扩充词表，据说Vicuna1.1并没有扩充词表，但是中文效果不错。
* [https://github.com/Facico/Chinese-Vicuna](https://link.zhihu.com/?target=https://github.com/Facico/Chinese-Vicuna" \t "_blank)

## **Stage 3: 对齐(LLaMA + FT + RHLF)**

* 这部分可以说是把ChatGPT的路径完整走了一遍。

## **StableVicuna**

* Vicuna = LLaMA + FT
* StableVicuna = Vicuna + RLHF
* [https://github.com/Stability-AI/StableLM](https://link.zhihu.com/?target=https://github.com/Stability-AI/StableLM" \t "_blank)

## **StackLLaMA**

* SFT: LLaMA + Lora
* RM: LLaMA + Lora + 分类
* [https://huggingface.co/blog/zh/stackllama](https://link.zhihu.com/?target=https://huggingface.co/blog/zh/stackllama" \t "_blank)

## **其他：LLaMA 推理优化**

### **llama.cpp**

* 用C/C++实现的推理，不依赖显卡。
* [https://github.com/ggerganov/llama.cpp](https://link.zhihu.com/?target=https://github.com/ggerganov/llama.cpp" \t "_blank)

### **GPTQ-for-LLaMA**

* 4 bits quantization of LLaMA using GPTQ.
* [https://github.com/qwopqwop200/GPTQ-for-LLaMa](https://link.zhihu.com/?target=https://github.com/qwopqwop200/GPTQ-for-LLaMa" \t "_blank)

## **写在最后**

上面的模型github中一般都有模型下载，但是国内的网络你懂得，有时候下载不下来。如果需要LLaMA模型的权重，可以看这一篇：**[ChatGPT平替模型：LLaMA（附下载地址，平民玩家和伸手党的福音！）](https://link.zhihu.com/?target=https://mp.weixin.qq.com/s?__biz=MzUyOTA5OTcwMg==&mid=2247484879&idx=1&sn=9babf498e543f41ff16b37a3baf63b65&scene=21%23wechat_redirect" \t "_blank)**