大模型面经——LLama2和chatGLM相对于transformer具体做了哪些优化?

原创 喜欢卷卷的瓦力 瓦力算法学研所 2024年04月18日 14:09 广东

◈ 面试干货专栏 ◈

作者：vivida

瓦力算法学研所
我们是一个致力于分享人工智能、机器学习和数据科学方面理论与应用知识的公众号。我...
117篇原创内容

公众号

本篇从模型结构、细节设计、注意力机制类型、位置编码、LayerNorm与激活函数优化角度，介绍目前较常用的框架LLama2和chatGLM原理，及其相对于transformer的优化。

本篇又是一道面经中的命题作文类题目，看似简单但暗含的细节非常多，回答的时候需要尽量注意有条理有框架。

这里给大家总结一些角度：可以从模型结构是decoder或是transformer类型的（encoder + decoder）、模型结构中更细节的设计、注意力机制类型、位置编码、LayerNorm、激活函数优化以及效率优化策略分别去聊。

跟前面的面经一样，本篇在写答案时也会尽量避免过于宽泛和官方的用词，并结合一些实际经验。

下面是一个快捷目录。

transformer

transformer估计做NLP出身的小伙伴都已经非常熟悉了，这里借鉴上面提到的角度来逐一讲一下transformer：

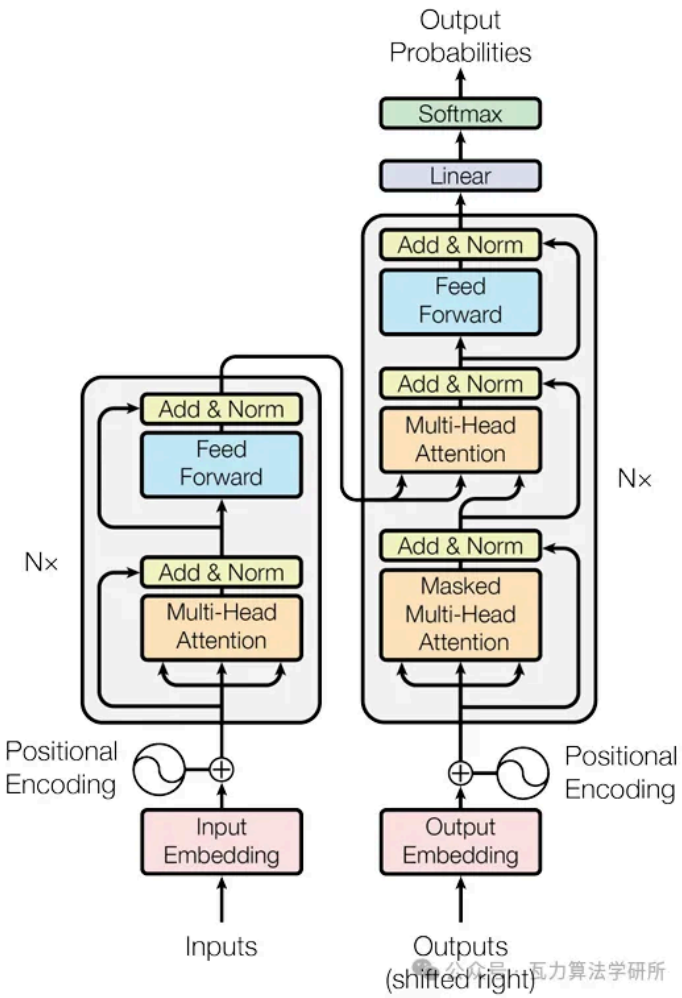
1. 模型结构设计

encoder和decoder

- Encoder：将输入序列转化为一组连续的、固定长度的内部向量，这些向量捕捉了输入序列中的上下文语义信息。Encoder 的输出是 Decoder 的输入。
- Decoder：基于 Encoder 的输出向量和已经生成的部分输出内容进行计算，生成最终的预测结果。

Transformer 的 Encoder 组件和 Decoder 组件都是由多个结构完全相同的原子组件堆叠而成。
一般情况下，有多少个 Encoder，就有多少个 Decoder；但在模型需要压缩的情况下，考虑到效果损失和效率，实际用到的Encoder个数会比 Decoder多。
Encoder 和 Decoder 是模型参数的主要载体。

这里直接上细节图并结合问题进行讲述：



这里主要是考验对模型细节的熟悉，面试官可能会问的问题，包括但不限于：

- **transformer中用的是postNorm还是preNorm，为什么**

  答案：为什么大模型结构设计中往往使用postNorm而不用preNorm?
- **decoder中有cross attention，请讲一下这个cross attention的输入和输出**

答案：输入是encoder的输出和历史上下文经过单向attention编码的向量，输出到FFN。

- **介绍一下transformer结构中用到的mask机制**

  答案：

  1）**预训练任务Masked Language Model (MLM)**：把输入的其中一部分词汇随机掩盖，模型的目标是预测这些掩盖词汇。这种训练方式使得每个位置的BERT都能学习到其上下文的信息。

  2）**self-attention中的Mask**：在attention中我们不希望一个token去注意到为了保持序列长度一致而出现的padding部分，所以attention中的mask通过在softmax前每个都设为-inf（或者实际的场景一个很小的数就可以），然后过完softmax后"padding"部分的权重就会接近于零，来处理掉这些无效的信息的。

  3）**下游任务的decoder中的Mask**：遮盖掉sequence中当前位置之后信息，以防止模型利用未来信息，也就是信息泄露。mask掉后模型的注意力只会集中在此前的序列上。

- **transformer结构常用的参数配置及其对应的训练数据量级**（考验对transformer应用熟悉度）

  答案：这里举例一个常用的架构：

  transformer-base: e6d6 embed_dim-512 ffn_embed_dim-2048 head-8

  一般是千万量级的数据训练用

3. 位置编码：正余弦编码

因为 Transformer 不采用 RNN 的结构，而是使用全局信息，不能利用单词的顺序信息，而这部分信息对于 NLP 来说非常重要。

所以 Transformer 中使用位置 Embedding 保存单词在序列中的相对或绝对位置

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d}\right)$$
$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d}\right)$$

公众号 · 乱力算法学研所

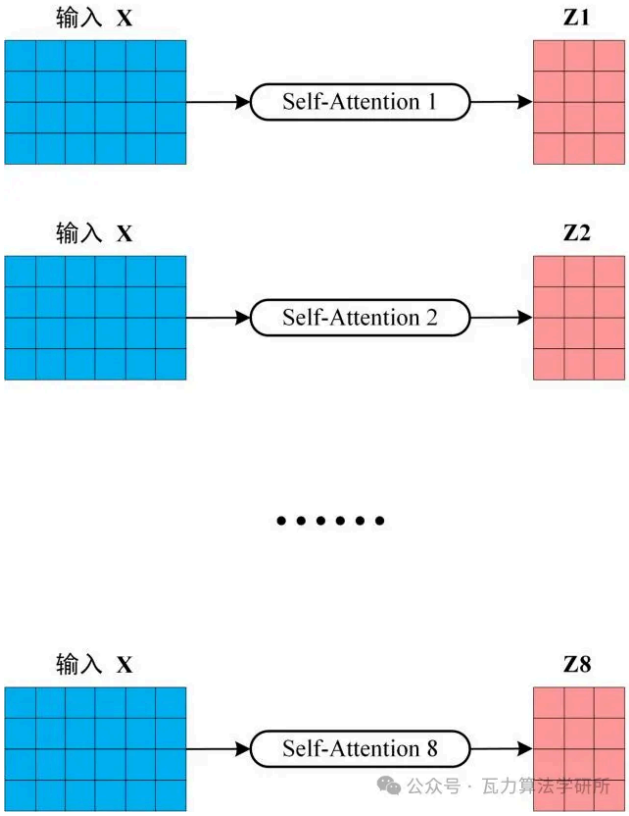
其中，pos 表示单词在句子中的位置，d 表示 PE的维度 (与词 Embedding 一样)，2i 表示偶数的维度，2i+1 表示奇数维度 (即 2i≤d, 2i+1≤d)。使用这种公式计算 PE 有以下的好处：

- **使 PE 能够适应比训练集里面所有句子更长的句子**，假设训练集里面最长的句子是有 20 个单词，突然来了一个长度为 21 的句子，则使用公式计算的方法可以计算出第 21 位的 Embedding。
- **可以让模型容易地计算出相对位置**，对于固定长度的间距 k，PE(pos+k) 可以用 PE(pos)计算得到。因为 Sin(A+B) = Sin(A)Cos(B) + Cos(A)Sin(B), Cos(A+B) = Cos(A)Cos(B) - Sin(A)Sin(B)。

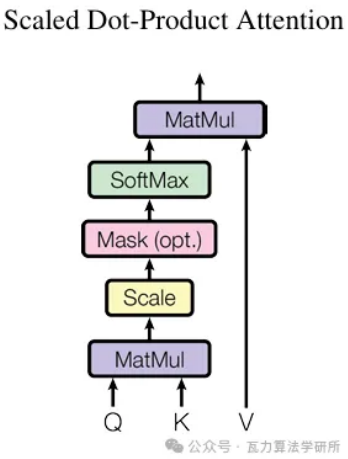
备注：将单词的词 Embedding 和位置 Embedding 相加，就可以得到单词的表示向量x，x就是 Transformer 的输入。

4. 注意力机制类型：MHA

Multi-Head Attention 是由多个 Self-Attention 组合形成的，首先将输入X分别传递到 h 个不同的 Self-Attention 中，计算得到 h 个输出矩阵Z。一般h是8，也就是8个头。



其中self-attention具体公式和结构图如下，



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

这里其实又容易引申出其他问题了：

- 在decoder的cross attention中 Q、K、V分别是什么
- MHA、GQA、MQA有什么区别

这里大家可以自行思考一下，答案基本都在文中了~

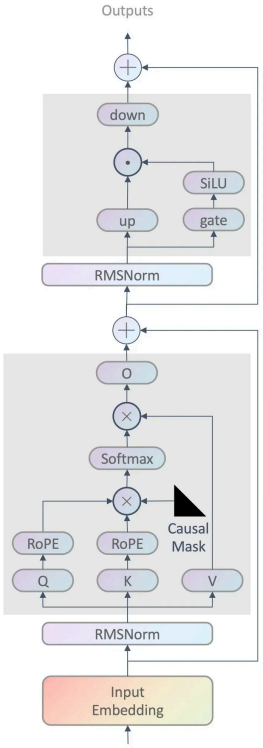
5. 激活函数

常用的是GELU激活函数，有时候考虑到效率也会用RELU

LLaMa2

1. 模型结构设计

llama2的架构比较容易理解，只用到decoder部分，具体结构跟GPT2类似；另外模型中已经变成了preNorm。



这里补充一下llama不同模型规模下超参数细节，有时候面试官也会为了考验熟悉程度随口一问。

参数规模	层数	自注意力头数	嵌入表示维度	学习率	全局批次大小	训练 Token 数
6.7B	32	32	4096	3.0e-4	400 万	1.0 万亿
13.0B	40	40	5120	3.0e-4	400 万	1.0 万亿
32.5B	60	52	6656	1.5e-4	400 万	1.4 万亿
65.2B	80	64	8192	1.5e-4	400 万	1.4 万亿

公众号 · 瓦力算法学研所

2. 位置编码：RoPE

具有较好的外推性。可以直接处理任意长的问题。LLaMA使用了Rotary Position Embedding。对于Q的第m个位置向量q，通过以下方法注入位置编码：

$$f(q,m) = \begin{bmatrix} q_0 \\ q_1 \\ \cdots \\ q_{d/2-1} \\ q_{d/2} \\ q_{d/2+1} \\ \cdots \\ q_{d-1} \end{bmatrix} \times \begin{bmatrix} cos(m\theta_0) \\ cos(m\theta_1) \\ \cdots \\ cos(m\theta_{d/2-1}) \\ cos(m\theta_0) \\ cos(m\theta_1) \\ \cdots \\ cos(m\theta_{d/2-1}) \end{bmatrix} + \begin{bmatrix} -q_{d/2} \\ -q_{d/2+1} \\ \cdots \\ -q_{d-1} \\ q_0 \\ q_{d_1} \\ \cdots \\ q_{d/2-1} \end{bmatrix} \times \begin{bmatrix} sin(m\theta_0) \\ sin(m\theta_1) \\ \cdots \\ sin(m\theta_{d/2-1}) \\ sin(m\theta_0) \\ sin(m\theta_1) \\ \cdots \\ sin(m\theta_{d/2-1}) \end{bmatrix}$$

公众号 · 瓦力算法学研所

RoPE形式上是绝对位置编码，即依赖其绝对位置m。

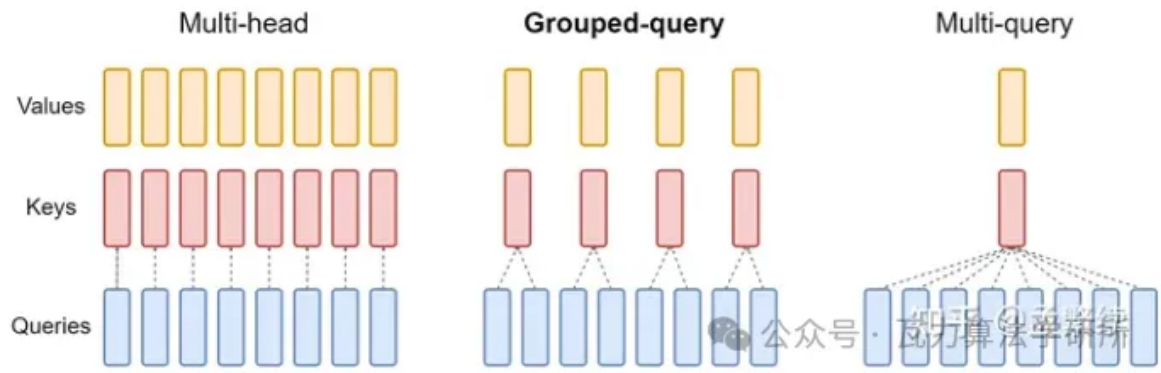
绝对位置编码的优点是计算速度快等，缺点是拓展长度比较麻烦，且绝对位置并没有什么实际意义。而相对位置编码对学习token之间的关系很有意义，比如距离的很远的两个token之间的关联大概率很小，使用相对位置编码往往能够获得更好的效果。

此外拓展长度也更容易，因为不论context size多长，只需关注最长距离以内的输入即可。相对位置编码的缺点是没有绝对位置编码计算速度快。

因此RoPE为q、k注入的绝对位置编码，计算得到的attention，却变成了相对位置编码；这个设计是非常巧妙的，推荐大家看大语言模型LLM为什么要用旋转位置编码（RoPE）—— 被LLaMA认可的长文本外推性（附代码）这篇有详细讲解。

2. 注意力机制类型：Group Query Attention(V2 only)

简单的说，就是一定头数 Q 共享一组 KV。



公众号 · 瓦力算法学研所

自回归模型生成回答时，需要前面生成的KV缓存起来，来加速计算。

多头注意力机制(MHA)需要的缓存量很大，Multi-Query Attention指出多个头之间可以共享KV对。

Group Query Attention没有像MQA一样极端，将query分组，组内共享KV，效果接近MHA，速度上与MQA可比较。

3. 归一化：RMSNorm

Layernorm换成了RMSNorm，而且是preNorm。

首先回顾一下LayerNorm公式

$$y = \frac{x - Mean(x)}{\sqrt{Var(x) + \epsilon}} * W + B$$

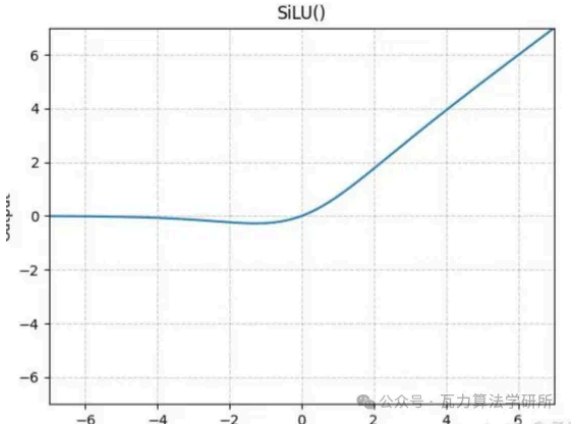
RMSNorm发现LayerNorm的中心偏移没什么用(减去均值等操作)。将其去掉之后，效果几乎不变，但是速度提升了40%。最终公式为

$$y = \frac{x}{\sqrt{Mean(x^2) + \epsilon}} * W$$

注意除了没有减均值，加偏置以外，分母上求的RMS而不是方差。

LLaMA2在 Attention Layer和MLP的输入上使用了RMSNorm，相比在输出上使用，也就是用preNorm替换postNorm，训练会更加稳定。

4. 激活函数：SwiGLU



SwiGLU，有时也被称为SiLU。公式为：

$$Sigmoid(x) * x$$

效果类似平滑版的ReLU。

ChatGLM

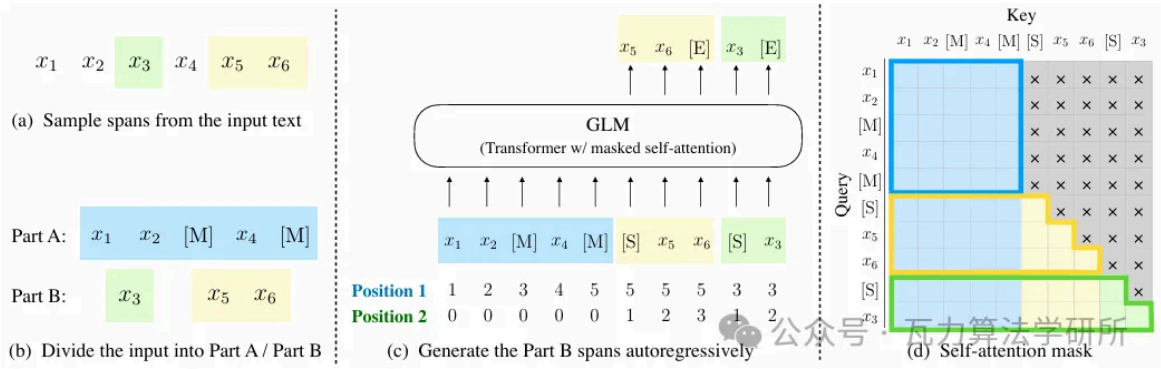
1. 模型结构设计

ChatGLM系列在原始single Transformer的基础上进行了一些修改：

1）重组了LN和残差连接的顺序；

2）使用单个线性层对输出token进行预测；

ChatGLM系列的亮点主要还是他的模型设计，融合了自编码、自回归、encoder-decoder各类思想，并且有精妙的span设计。



这里简述一下他的模板设计：

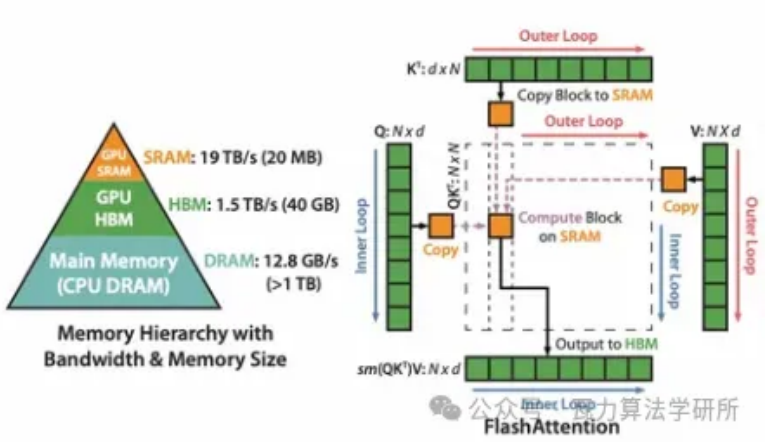
自回归：**采样span进行单向自回归预测，A为mask后文本，B为span**

自编码：**A、B互不可见，但内部可见，B单向可见**

2. 注意力机制：Flash attention、MQA（v2 only）

- Flash attention

flash-attention是一种快速、高效、可扩展的注意力机制，它利用了一种称为哈希感知（hash-aware）的技术，可以根据它们的相似性将输入序列中的元素分配到不同的桶（bucket）中。这样，模型只需要计算桶内元素之间的注意力权重，而不是整个序列。这大大减少了计算量和内存需求，同时保持了较高的精度和表达能力。



简单的说就是，计算softmax时候不需要全量input数据，可以分段计算； 反向传播的时候，不存储attention matrix (N^2的矩阵)，而是只存储softmax归一化的系数。

- MQA

让 Q 仍然保持原来的头数，但 K 和 V 只有一个头，相当于所有的 Q 头共享一组 K 和 V 头。

3. 激活函数：GeLU

总结

下面我们把上面提到的所有角度简略总结一下。

模型	结构	归一化位置	归一化类型	注意力机制	激活函数	位置编码
transformer	encoder-decoder	postNorm	layerNorm	MHA	RELU/GELU	正余弦
LLaMa2	decoder	preNorm	RMSNorm	GQA	SwiGLU	RoPE
ChatGLM2	encoder-decoder	postNorm	layerNorm	MQA、flash attention	GeLU	正余弦

公众号 · 瓦力算法学研所

欢迎大家关注公众号，更多面经干货将持续放送~

瓦力算法学研所
我们是一个致力于分享人工智能、机器学习和数据科学方面理论与应用知识的公众号。我...
117篇原创内容

公众号

面试干货 70

面试干货 · 目录

上一篇
AIGC算法工程师面经—公式理解篇（上）

下一篇
垂直领域大模型微调实践经验最全总结

修改于2024年04月19日