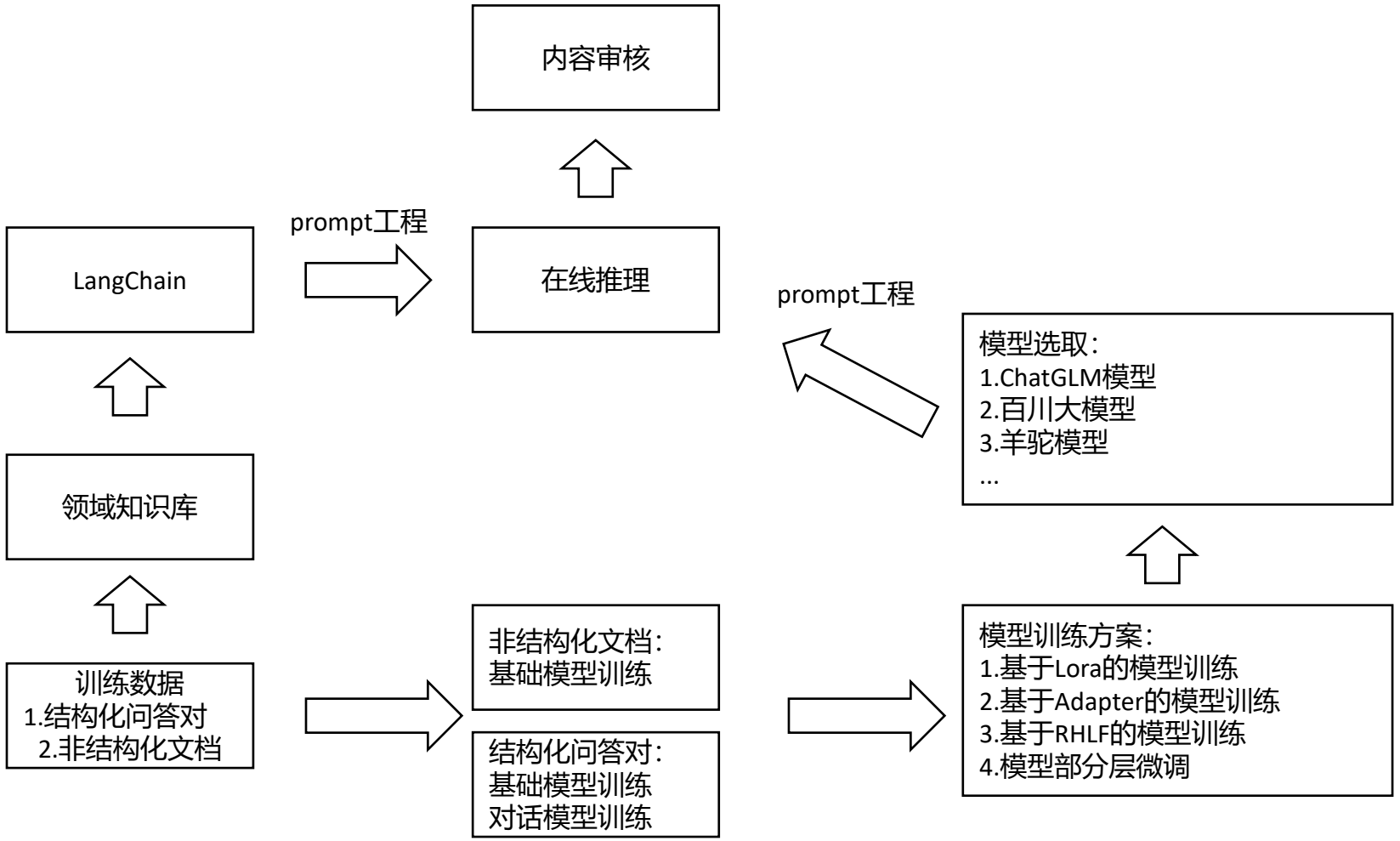
- 大模型的三要素

1.算法：模型结构，训练方法

2.数据：token计算方法，数据和模型效果之间的关系

3.算力：英伟达GPU介绍

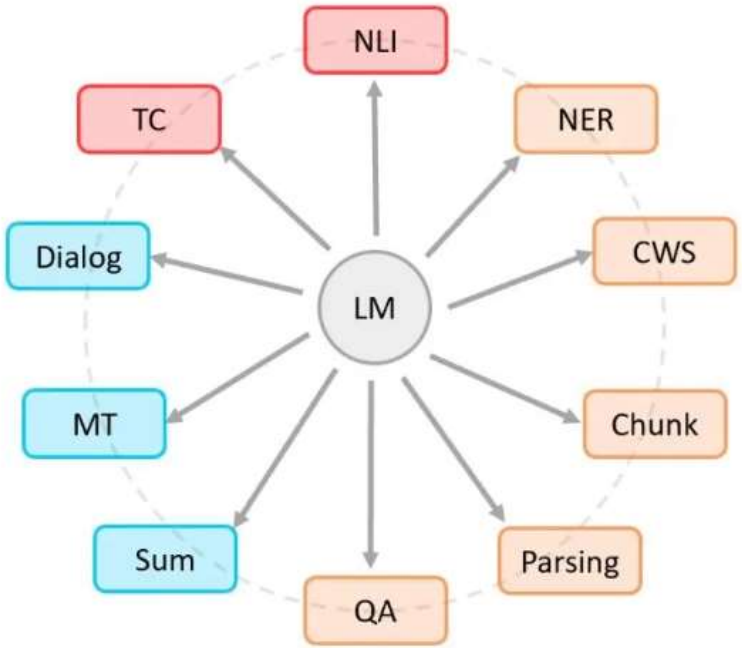
基于大模型对话的系统架构



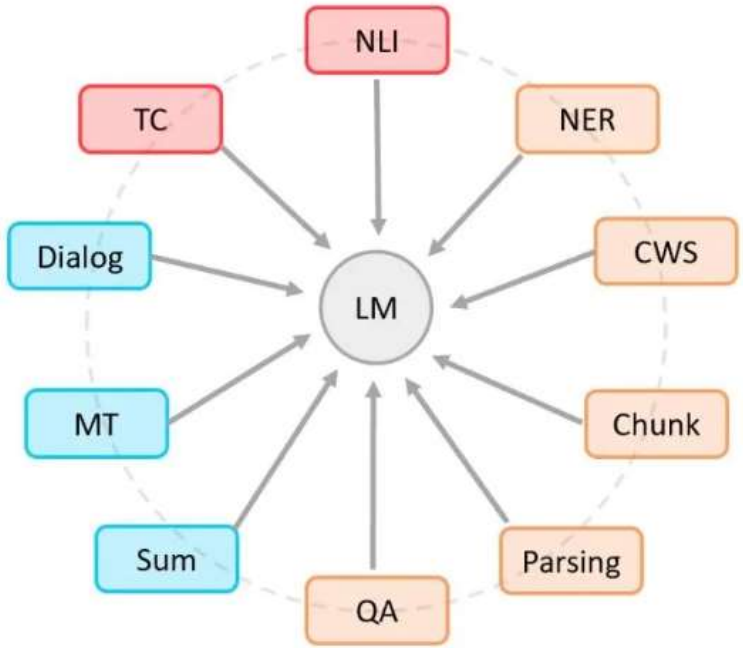
机器学习的四个范式

- 非神经网络时代的完全监督学习 （Fully Supervised Learning, Non-Neural Network）
- 基于神经网络的完全监督学习 (Fully Supervised Learning, Neural Network)
- 预训练，**精调**范式 (Pre-train, Fine-tune)
- 预训练，**提示**，预测范式（Pre-train, Prompt, Predict）

提示学习：提示工程的支撑



Fine-tuning



Prompting

范式	第1、2种范式：完全监督学习(Fully Supervised Learning；非神经网络和神经网络)	第3种范式：预训练-微调	第4种范式：预训练-提示学习
训练数据	目标任务数据集	大规模生语料，目标任务数据集	大规模生语料，目标任务数据集
输入	我是谁?	我是谁？，	[CLS]我是谁？[SEP] 主题是[MASK] [MASK] [SEP],
输出	,[0,0,1]	[0,0,1]	[CLS]哲学[SEP]
输出层	一个线性变换	一个线性变换	无新增结构
特点	依赖目标任务数据集来获得文本表示。	基于庞大的生语料拉来获得良好的文本表示。 基于目标任务数据获得下游任务知识。	基于庞大的生语料拉来获得良好的文本表示。 基于语言模型的文本生成能力，和下游任务特点，设计训练和推理策略。

大模型使用技巧：Prompt工程

提示工程(Prompt engineering)是一门相对较新的学科，旨在为各种应用和研究主题开发和优化提示，以有效地利用语言模型（LMs:language models ）。提示工程技能有助于更好地了解大型语言模型（LLMs:large language models ）的能力和局限性。研究人员使用提示工程来提高 LLMs 在各种常见和复杂任务（如问答和算术推理）上的能力。开发人员使用提示工程来设计稳健且有效的提示技术，与 LLMs 和其他工具进行交互。

Prompt:
完成这个句子：
天空是

Output:天空是指大气层上方的空间，通常是指人们在地面上所看到的天空，它的颜色和形态会随着时间、地点、季节和气象条件的不同而变化。

Prompt:
根据今天天气情况，完成描述今天天气的句子：
天空是

Output:天空是阴沉的，乌云密布，目前正在下雨。

Prompt:
根据上海今天晴天，完成描述今天天气的句子：
天空是

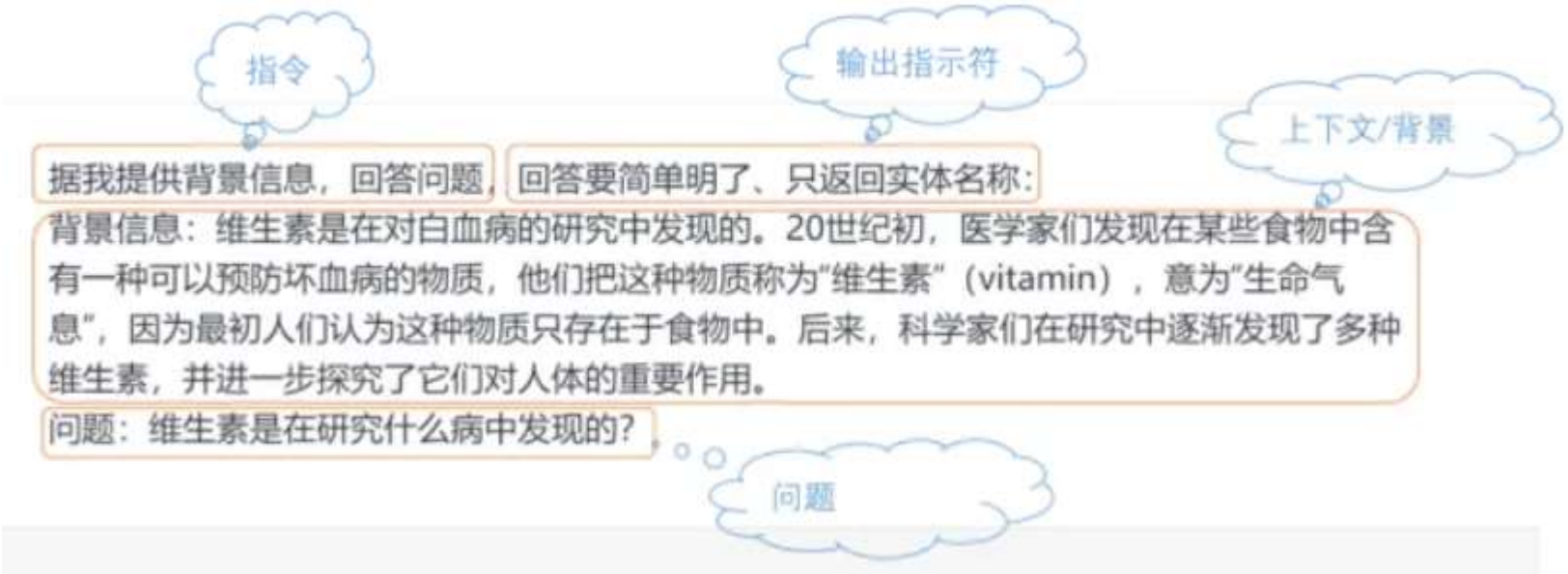
Output：天空是湛蓝色的，阳光明媚，没有云彩。

大模型使用技巧：Prompt工程

提示的组成部分(Elements of a Prompt)

提示可能包含以下任何组件：

- Instruction 指令：您想让模型执行的特定任务或指令
- Context 上下文：可以涉及外部信息或附加上下文，可以引导模型产生更好的响应
- Input Data 输入数据：我们有兴趣为其找到响应的输入或问题
- Output Indicator 输出指示符：指示输出的类型或格式。



大模型使用技巧：Prompt工程

基础prompt提示

- 文本摘要(Text Summarization)
- 信息提取(Information Extraction)
- 问答(Question Answering)
- 文本分类(Text Classification)
- 对话(Conversation)
- 代码生成(Code Generation)
- 推理(Reasoning)

大模型使用技巧：Prompt工程

高级prompt提示

- 零样本提示(Zero-shot Prompting)
- 少量样本提示(Few-shot Prompting)
- 思维链(COT)提示(Chain-of-Thought Prompting)
- 零样本CoT(Zero-shot CoT)
- 自洽性/自一致性(Self-Consistency)
- 生成知识提示(Generate Knowledge Prompting)
- 自动提示工程(Automatic Prompt Engineer)

少量样本提示(Few-shot Prompting)

Prompt:
巴黎是法国的首都，有艾菲尔铁塔、卢浮宫和香榭丽舍大道等著名景点。
纽约是美国的一座大城市，拥有自由女神像、时代广场和中央公园等著名景点。
东京是日本的首都，有天空树、浅草寺和新宿御苑等著名景点。
北京是?

Completion:北京是中国的首都，有故宫、长城和天坛等著名景点。

通过COT解决基本逻辑问题：

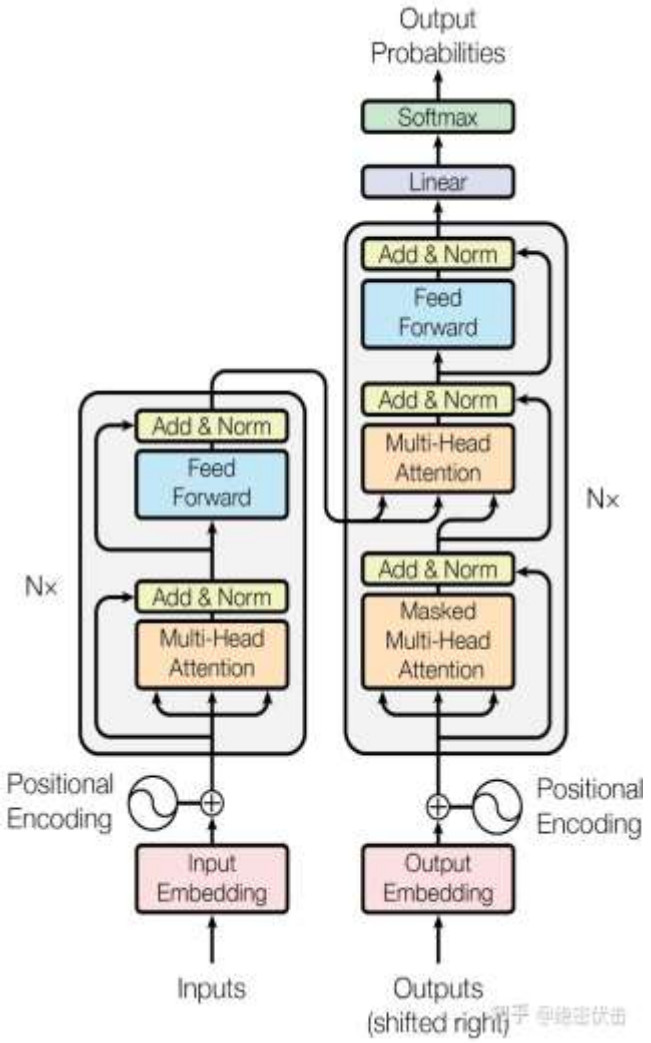
> Prompt:
> 煮一个鸡蛋需要2分钟，煮5个鸡蛋需要几分钟?
> Completion:
> 煮一个鸡蛋需要2分钟，因此煮5个鸡蛋所需的时间是5个鸡蛋的煮时间之和。所以煮5个鸡蛋需要5 x 2 = 10分钟。答案是10分钟。

自洽性/自一致性(Self-Consistency)

Prompt:
我6岁的时候，我妹妹只有我的一半大。现在我70岁了，我妹妹多大了?

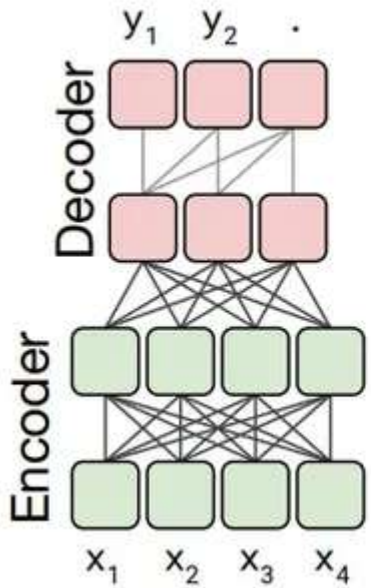
Completion:你的妹妹现在应该是64岁，因为当你6岁时，她只有3岁，相比之下，你已经增长到70岁，所以她应该比你小6岁。

大模型的内核：Transformer

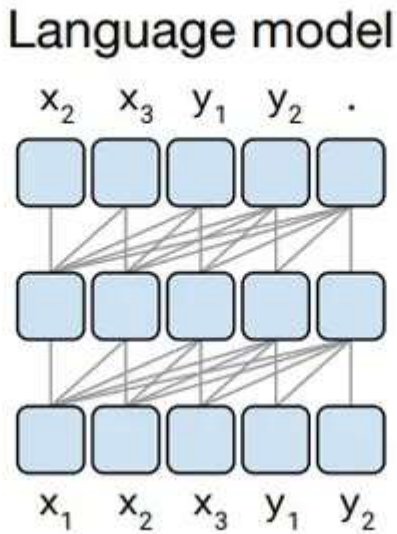


模型	结构	位置编码	激活函数	layer norm方法
原生 Transformer	Encoder-Decoder	Sinusoidal编码	ReLU	Post layer norm
BERT	Encoder	绝对位置编码	GeLU	Post layer norm
~~LLaMA~~	Casual decoder	RoPE	SwiGLU	Pre RMS Norm
ChatGLM-6B	Prefix decoder	RoPE	GeGLU	Post Deep Norm
Bloom	Casual decoder	ALiBi	GeLU	Pre Layer Norm

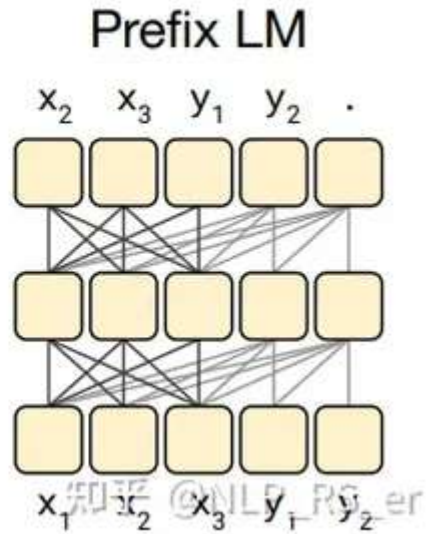
大模型的架构



BART
T5



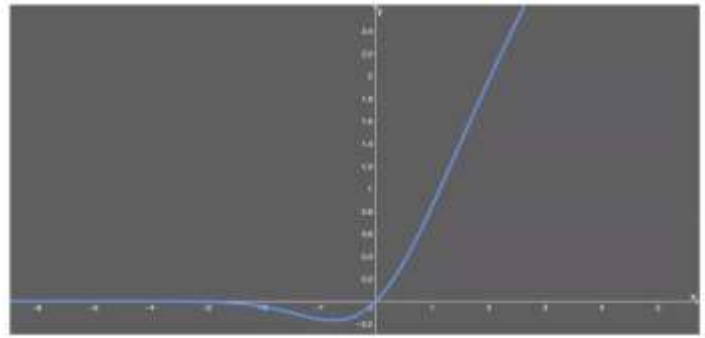
GPT
LLAMa



GLM模型

激活函数

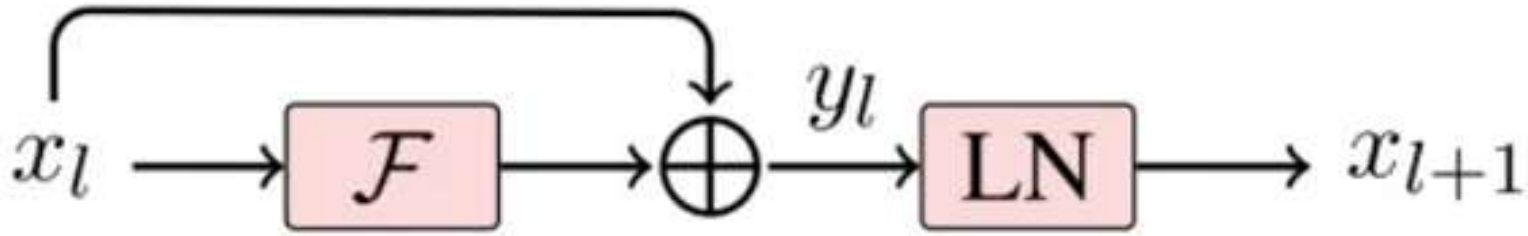
$$GELU(x) = x \times P(X <= x) = x \times \phi(x), x \sim N(0,1)$$
$$= 0.5x\left(1 + tanh\left(\sqrt{2/\pi}(x + 0.044715x^3)\right)\right)$$



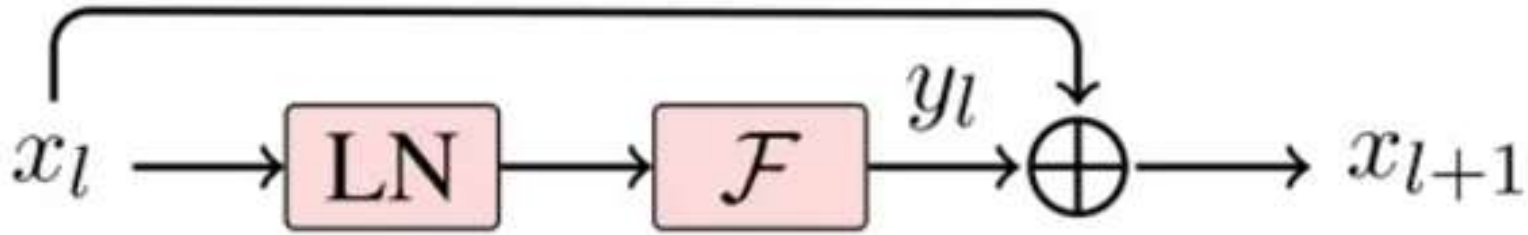
$$\text{GEGLU}(x, W, V, b, c) = \text{GELU}(xW + b) \otimes (xV + c)$$
$$\text{SwiGLU}(x, W, V, b, c, \beta) = \text{Swish}_\beta(xW + b) \otimes (xV + c)$$

这里 $Swish_\beta(x) = x\sigma(\beta x)$, β 为指定常数，如1

作者并没有对激活函数提出的原理和动机做过多描述，论文本身是对各类激活函数变种效果的对比尝试



post-norm residual unit



pre-norm residual unit

Deep Norm是对Post-LN的的改进，具体的：

```
def deepnorm(x):
    return LayerNorm(x * α + f(x))
```

与layerNorm相比，RMS Norm的主要区别在于去掉了减去均值的部分，计算公式为：

$$\bar{a}_i = \frac{a_i}{\mathrm{RMS}(\mathbf{a})} g_i, \quad \text{where } \mathrm{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}.$$

这里的 a_i 与Layer Norm中的 x 等价，作者认为这种模式在简化了Layer Norm的同时，可以在各个模型上减少约 7%~64% 的计算时间

位置编码

旋转位置编码

cos(a-b)=cosacosb+sinasinb

$$\langle \boldsymbol{f}(\boldsymbol{q}, m), \boldsymbol{f}(\boldsymbol{k}, n) \rangle = g(\boldsymbol{q}, \boldsymbol{k}, m-n)$$

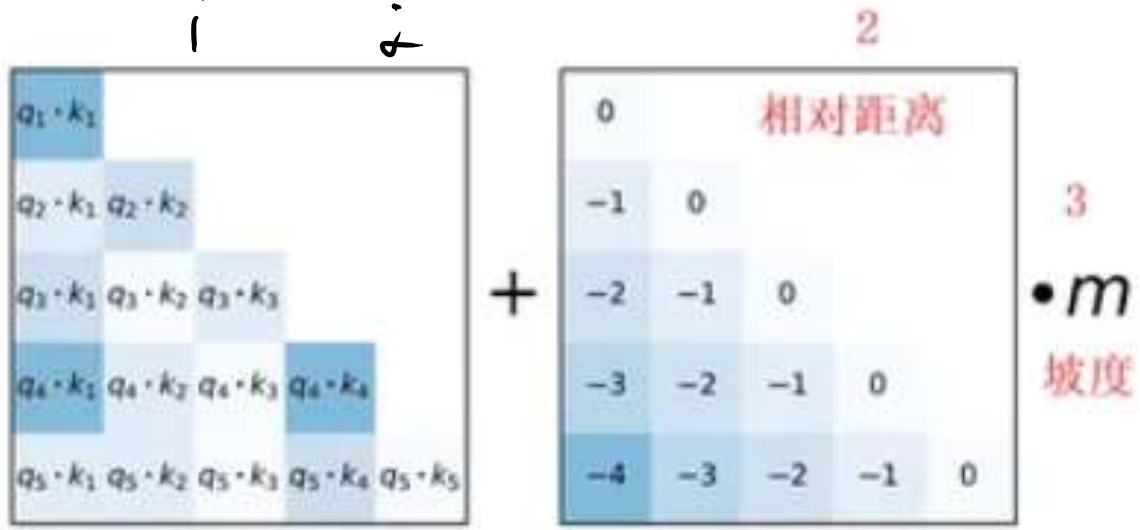
$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_0 \\ \cos m\theta_0 \\ \cos m\theta_1 \\ \cos m\theta_1 \\ \vdots \\ \cos m\theta_{d/2-1} \\ \cos m\theta_{d/2-1} \end{pmatrix} + \begin{pmatrix} -q_1 \\ q_0 \\ -q_3 \\ q_2 \\ \vdots \\ -q_{d-1} \\ q_{d-2} \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_0 \\ \sin m\theta_0 \\ \sin m\theta_1 \\ \sin m\theta_1 \\ \vdots \\ \sin m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{pmatrix}$$

位置编码

ALiBi

0 1 2 3 4 5
今天我在上课

$-|i-j|$



所有层都可以用

这是整体的一个key矩阵：

这是i-th query和所有key 向量的可能的相对位置的差异

$$\text{softmax}(\mathbf{q}_i\mathbf{K}^\top + m\cdot[-(i-1),\ldots,-2,-1,0])$$

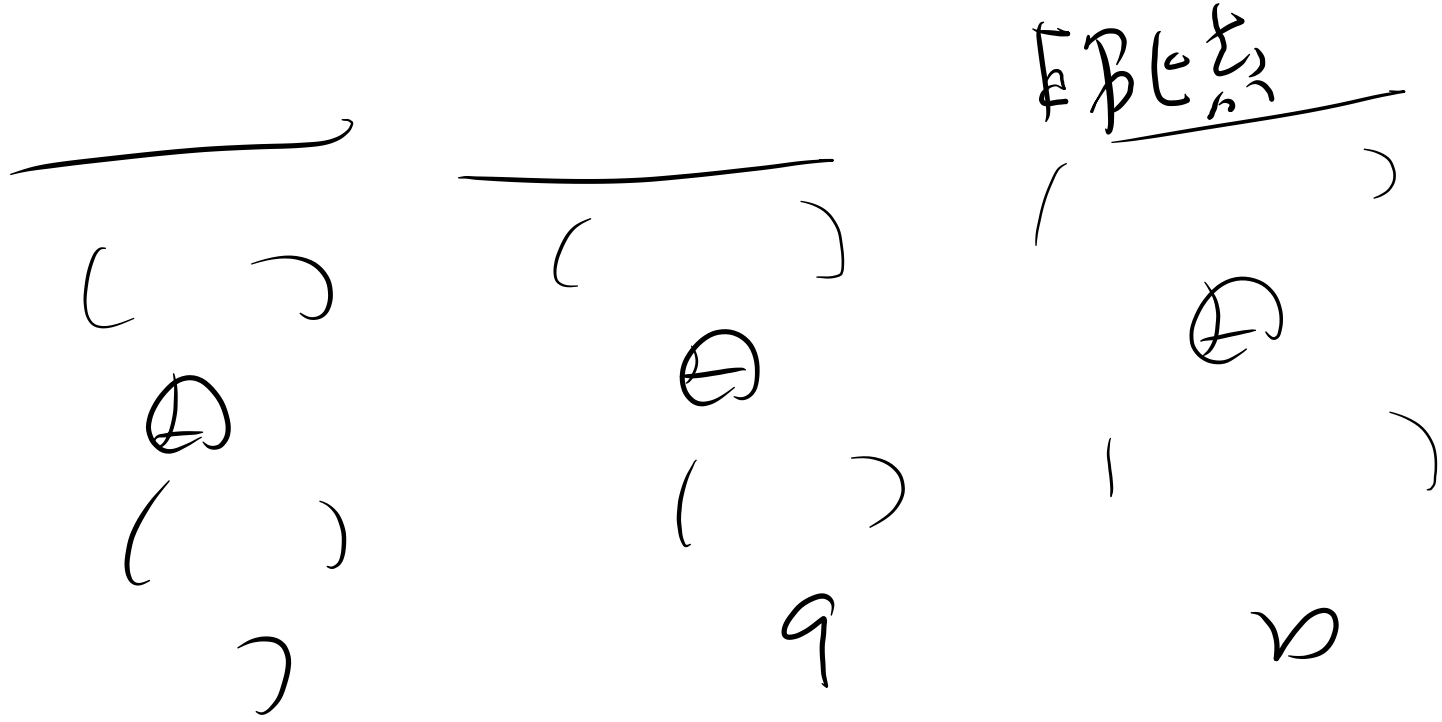
当前是第i个query

m是坡度

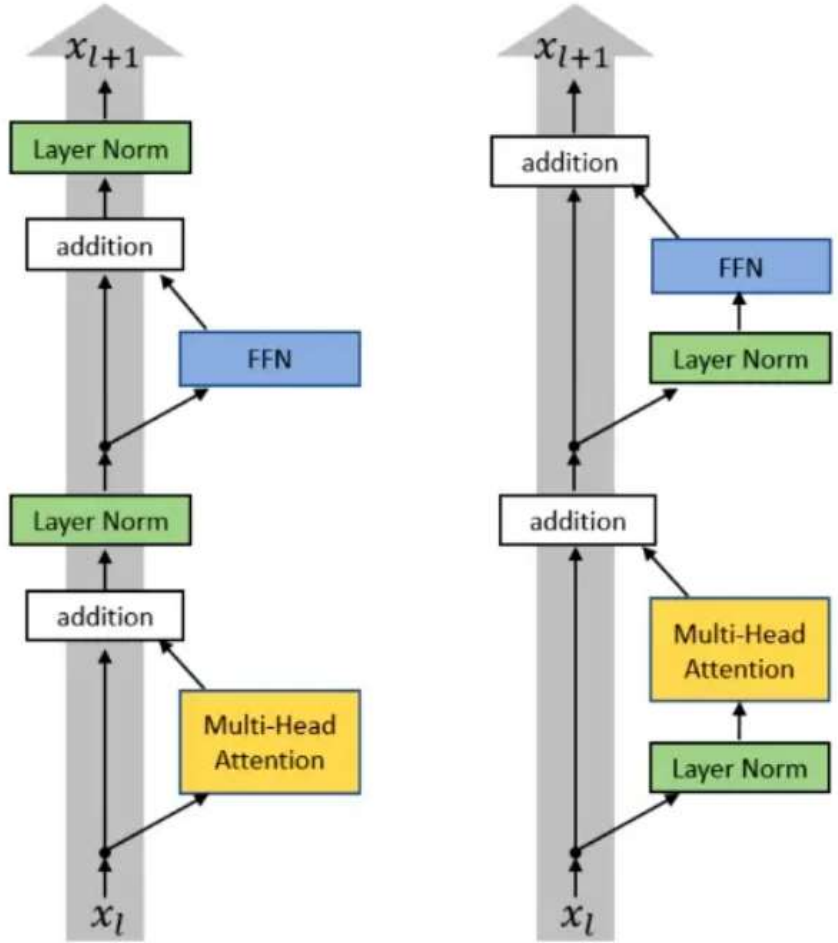
K，这里应该包含了，只能是取1, ..., i这些位置了...只适用于causal自回归

对于n个head的话，m的取值就是 $2^{\frac{-8}{n}}$ ：

$2^{\frac{-8}{1}}$，$2^{\frac{-8}{2}}$，$2^{\frac{-8}{3}}$，$\ldots, 2^{\frac{-8}{n}}$， 这样的m个坡度了。


q
p

LayerNorm编码



Transformer有效性分析