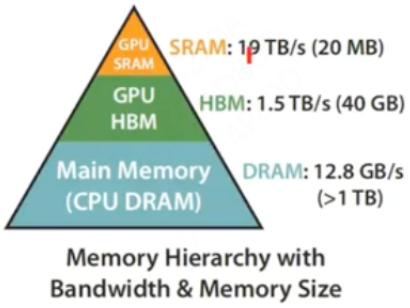
# Attention

- Flash-attention




CPU DRAM为内存；GPU HBM为显卡内存；GPU SRAM为显卡缓存

现状：self-attention的复杂度是序列长度的二次方->序列过长时，算法复杂度很高
目的：加速注意力计算并减少内存占用
原理：使用平铺和重计算等技术，将QKV切分后从HBM加载到SRAM执行计算，然后将结果更新会HBM。

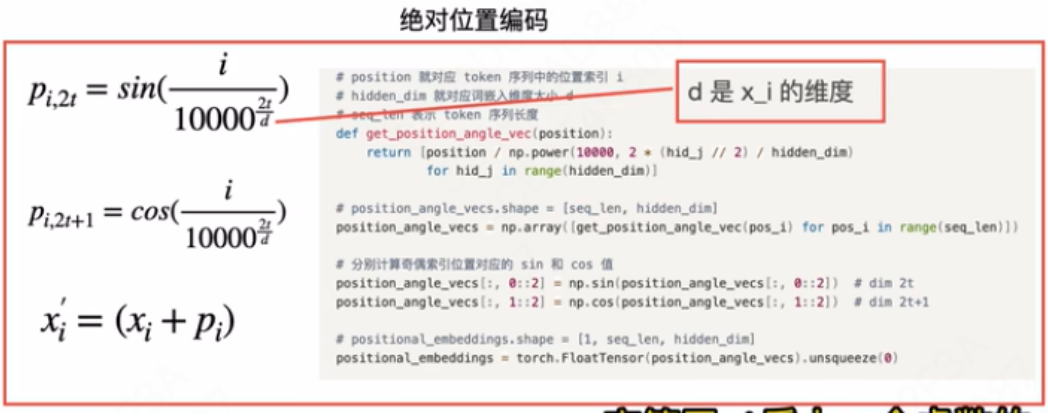
S和P与序列长度N的平方成正比，导致将中间结果S和P加载到SRAM中需要占用极大的GPU缓存和需要极大的时间。
QKV分块计算、softmax分块计算：1可以减少SRAM的占用 2分块计算可以跳过存储中间结果的过程，减少HBM的读写

- KV-Cache

在Self-attention中，因为attention mask会使得token的query向量只与其和其之前的token的key和value向量相乘，所以在生成下一个token的时候缓存之前token的key和value向量能够显著加速。

# 位置编码

- 正余弦编码




- 旋转位置编码Rope

在self-attention的过程中纳入位置编码

$$\langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n)\rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n)$$

RoPE寻找函数g令上式成立
设词向量维度为2维：

$$f_q(\boldsymbol{x}_m, m) = (\boldsymbol{W}_q \boldsymbol{x}_m)\, e^{im\theta}$$
$$f_k(\boldsymbol{x}_n, n) = (\boldsymbol{W}_k \boldsymbol{x}_n) e^{in\theta}$$
$$g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n) = \mathrm{Re}\left[(\boldsymbol{W}_q \boldsymbol{x}_m)(\boldsymbol{W}_k \boldsymbol{x}_n)^* e^{i(m-n)\theta}\right]$$

相乘之后可以表达成差

$f_q$可以表示为：

$$f_q(\boldsymbol{x}_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} W_q^{(1,1)} & W_q^{(1,2)} \\ W_q^{(2,1)} & W_q^{(2,2)} \end{pmatrix} \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix}$$
$$= \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_m^{(1)} \\ q_m^{(2)} \end{pmatrix}$$

旋转矩阵

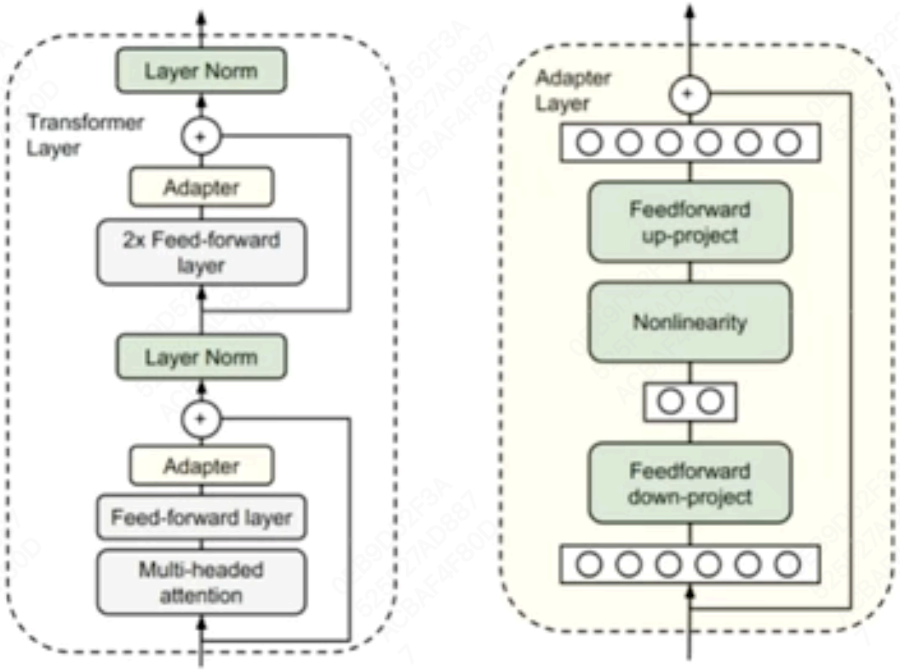
二维最终形式

$$g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n) = \begin{pmatrix} \boldsymbol{q}_m^{(1)} & \boldsymbol{q}_m^{(2)} \end{pmatrix} \begin{pmatrix} \cos((m-n)\theta) & -\sin((m-n)\theta) \\ \sin((m-n)\theta) & \cos((m-n)\theta) \end{pmatrix} \begin{pmatrix} k_n^{(1)} \\ k_n^{(2)} \end{pmatrix}$$

## 微调篇

- adapter

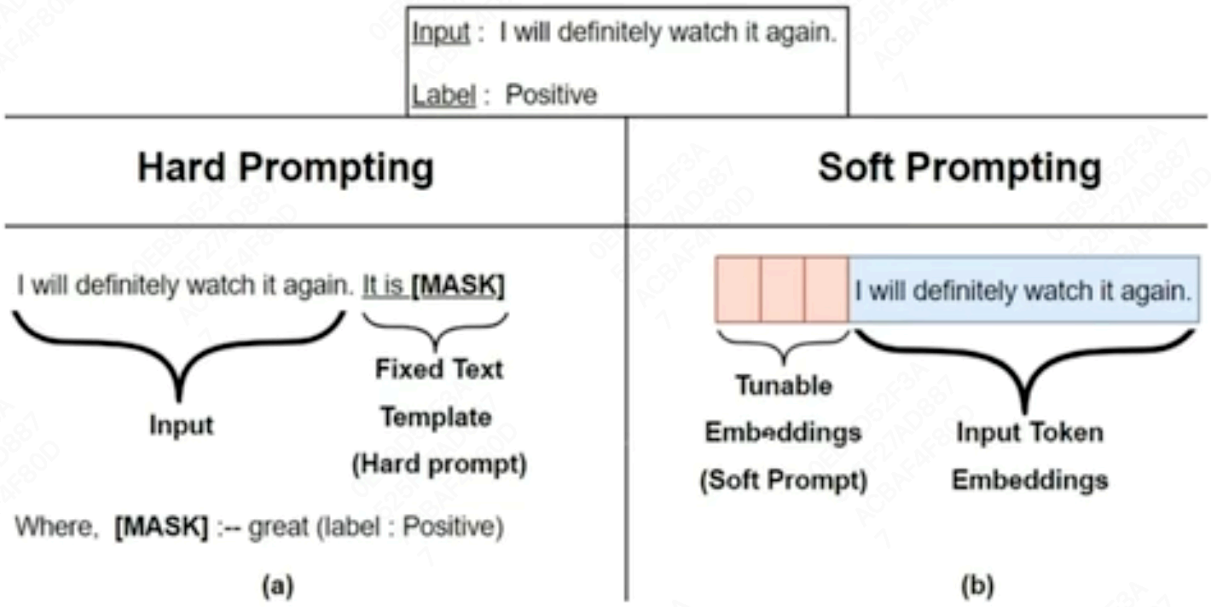
原理：针对每个Transformer层，增加两个Adapter结构（self-attention层和add+norm之间、feed-forward层和add+norm之间）。训练时，固定原来预训练模型参数不变，只对新增的Adapter结构和Layer norm层进行微调。
缺点：增加了额外的结构，这可能会导致推理时间的增加。




增加1个Adapter结构和1个Feed-forward层

- prefix tuning

原理：在每层transfomer的输入的前面插入一段virtual tokens作为prefix，训练时只更新prefix部分的参数。
缺点：插入 prefix 后，原本用于实际输入内容的 token 数量减少，这可能导致模型在处理长序列或复杂任务时出现信息丢失或表达受限的问题。



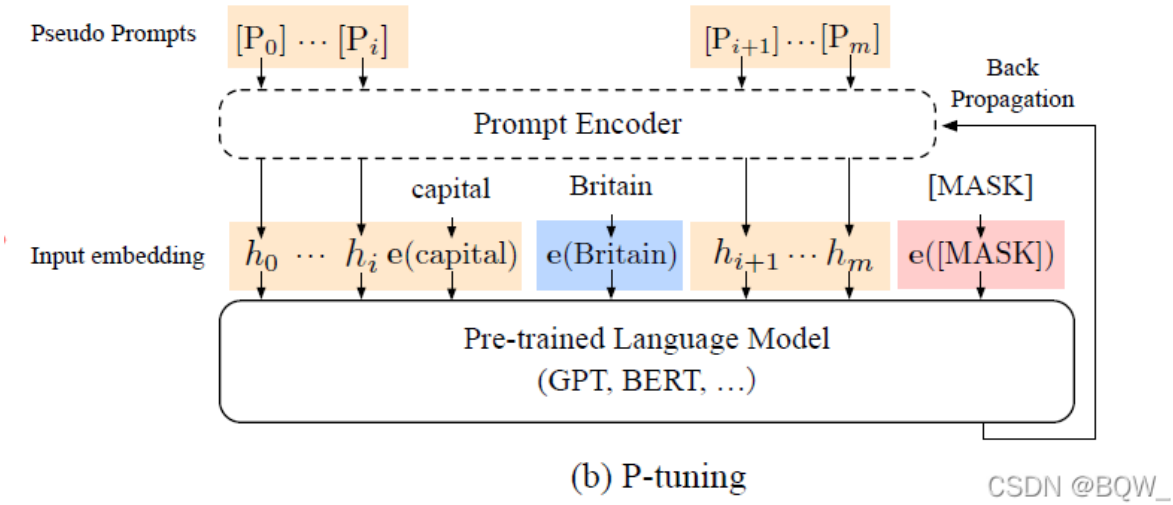

基于soft prompt的微调

- prompt tuning

原理：基于具体任务的需求，设计一个提示模板（Prompt Template）。该模板通常包含一些固定格式的文本，用于引导模型生成符合任务要求的输出。
缺点：任务高度定制化。

- p-tuning

原理：设计一个提示模板，该模板包含一系列虚拟标记（tokens）或嵌入向量，这些向量是待优化的参数。
缺点：




(b) P-tuning



- p-tuning v2

原理：在p-tuning的基础上，每一层Transformer都插入virtual tokens；移除重参数化编码器；不同任务提示长度不同。
缺点：微调后容易导致遗忘旧知识。

- lora

原理：在矩阵乘法的旁边加上一个数据通路。这个数据通路由两个矩阵组成，本质上是对权重矩阵的低秩分解。
缺点：