注意力机制与外部记忆

两种注意力:

1. 自下而上的注意力

就是图像中鲜艳的颜色,人会自动注意这些地方.

2. 自上而下的注意力

就是带着问题去寻找答案.

鸡尾酒会效应

当一个人在吵闹的鸡尾酒会上和朋友聊天时,尽管周围噪音干扰很多,他还是可以听到朋友的谈话内容,而忽略其他人的声音.同时,如果未注意到的背景声中有重要的词(比如他的名字),他会马上注意到.

软性注意力机制

1. 计算注意力分布 α

$$\begin{aligned}\alpha_n &= p(z=n|X,q) \\ &= \text{softmax}(s(x_n,q)) \\ &= \frac{\exp(s(x_n,q))}{\sum_{j=1}^{n}\exp(s(x_j,q))}\end{aligned} \tag{1}$$

$s(x_n,q)$是一个打分函数

2. 根据 α 来计算输入信息的加权平均

$$
\begin{aligned}
att(X,q) &= \sum_{n=1}^{N} \alpha_n x_n, \\
&= \mathbb{E}_{z\sim p(z|x,q)}[x_z]
\end{aligned}
\tag{2}
$$

打分函数	公式
加性模型	$s(x,q)=v^T\tanh(Wx+Uq)$
点积模型	$s(x,q)=x^Tq$
缩放点积模型	$s(x,q)=\frac{x^Tq}{\sqrt{D}}$
双线性模型	$s(x,q)=x^TWq$

硬性注意力

就是软性注意力是加权平均的,硬性就是只算注意力最高的.

一般强化学习用,这里略.

键值对注意力

用$(K,V)=[(k_1,v_1),\cdots,(k_N,v_N)]$表示$N$个输入信息

$$
\begin{aligned}
att((K,V),q) &= \sum_{n=1}^{N} \alpha_n v_n \\
&= \sum_{n=1}^{N} \frac{\exp(s(k_n,q))}{\sum_j \exp(s(k_j,q))} v_n
\end{aligned}
\tag{3}
$$

多头注意力

利用多个查询$Q=[q_1,\cdots,q_M]$,同时从输入信息中选取多组信息.每个"注意力头"关注输入信息的不同部分.

$$\mathbf{att}((K,V),Q)=\mathbf{att}((K,V),q_1)\oplus\cdots\mathbf{att}((K,V),q_M) \tag{4}$$

自注意力机制

上面所讲的注意力机制里面都有一个查询q,是需要外部提供的,而自注意力机制的q是自己生成的.

普通的自注意力机制

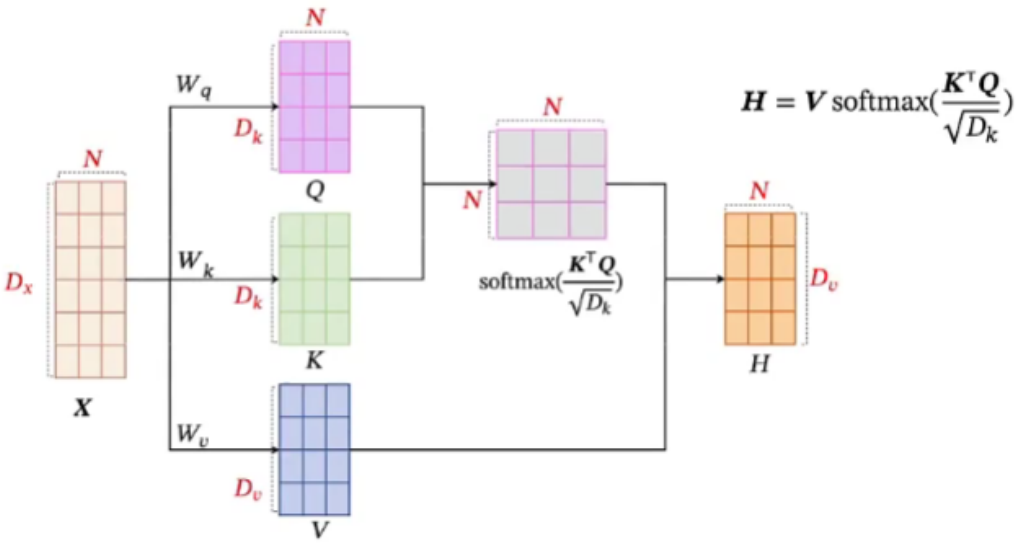
普通的自注意力机制,没有参数需要学习,且一个输入做了全部内容.

$$X'=X\cdot\mathbf{softmax}(X^TX) \tag{5}$$

KQV模式

普通的自注意力机制中,我们的输入既做了Q,又做了V,又做了V,即一个表示做了三个功能,我们可以将其拆开.

KQV模式如下图所示



1. 输入序列 $X=[x_1,\cdots,x_N]\in\mathbb{R}^{(D_x\times N)}$
2. 首先生成三个向量序列

$$\begin{aligned} Q&=W_qX\in\mathbb{R}^{(D_k\times N)}\\ K&=W_kX\in\mathbb{R}^{(D_k\times N)}\\ V&=W_vx\in\mathbb{R}^{(D_k\times N)} \end{aligned}\tag{6}$$

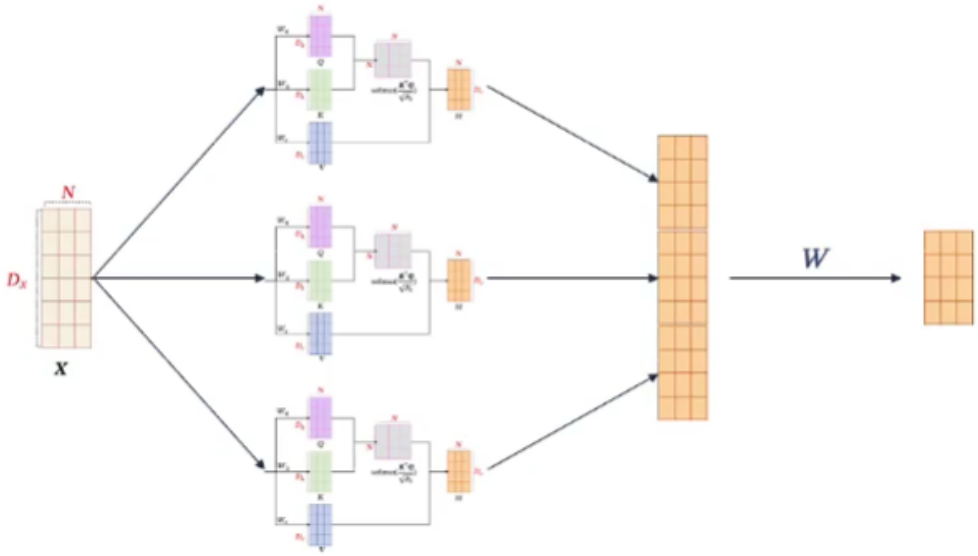
3. 计算 h_n

$$\begin{aligned} h_n&=\mathbf{att}((K,V),q_n)\\ &=\sum_{j=1}^{N}\alpha_{nj}v_j\\ &=\sum_{j=1}^{N}\mathbf{softmax}(s(k_j,q_n))v_j \end{aligned}\tag{7}$$

> 注:若使用缩放点积:
>
> $$H=V\,\mathbf{softmax}(\frac{K^TQ}{\sqrt{D_k}})\tag{8}$$

多头自注意力模型

就是自注意力机制,这样多来几组,有点像卷积的inception模块(用不同的卷积核提取特征,然后拼接在一起).



Transformer

就是不用其他的模型,只用注意力机制.

> 仅仅依靠注意力机制是不够的.例如,将输入中的x_1, x_2交换位置,则没有变化.
>
> 因此需要一些其他操作

其他操作

✅位置编码

例如,一句话中,给每个词进行序号标记,如“我今天很帅”,则额外添加“我为1,今天为2”这种位置信息,跟原始的词向量进行一个加法.

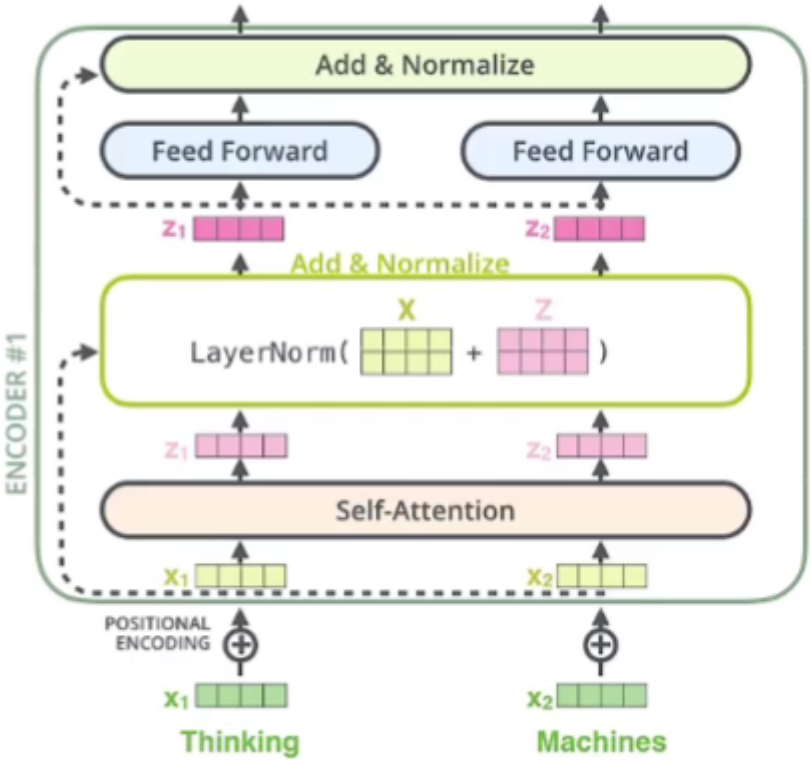
✅层归一化

✅直连边

✅逐位的FFN

就是每个词向量做一个多层的全连接.

下面是一个Transformer Encoder的一个block示例.

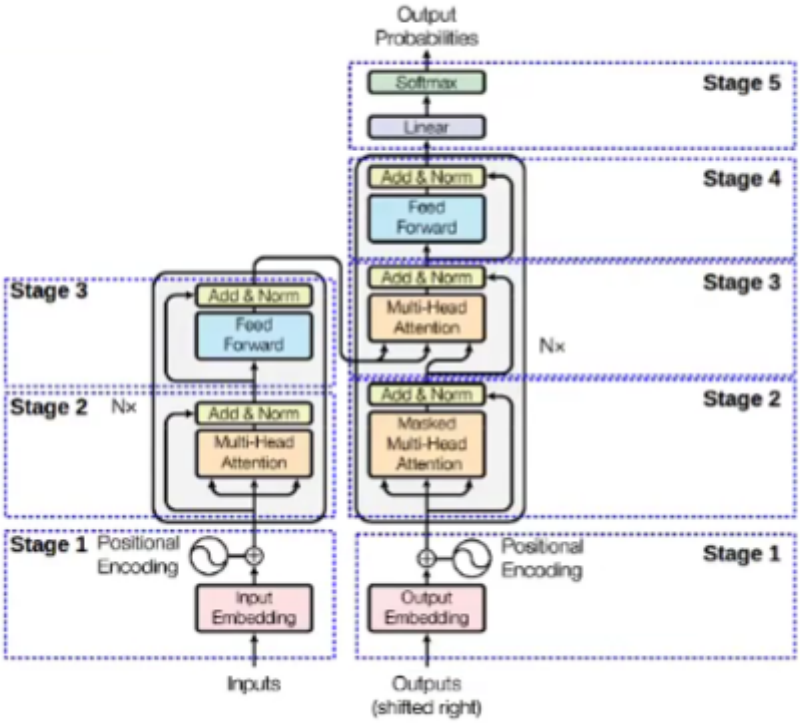


复杂度分析

模型	每层复杂度	序列操作数	最大路径长度
CNN	$O(kLD^2)$	$O(1)$	$O(\log_k^{(L)})$
RNN	$O(LD^2)$	$O(L)$	$O(L)$
Transformer	$O(L^2D)$	$O(1)$	$O(1)$

其中k表示卷积核大小,L表示序列长度,D表示维度

Transformer



外部记忆

记忆:外界信息在人脑中的内部表示

记忆过程

- 工作记忆(短期记忆)
- 情景记忆
- 结构记忆(长期记忆)

特点:联想记忆

记忆网络

略