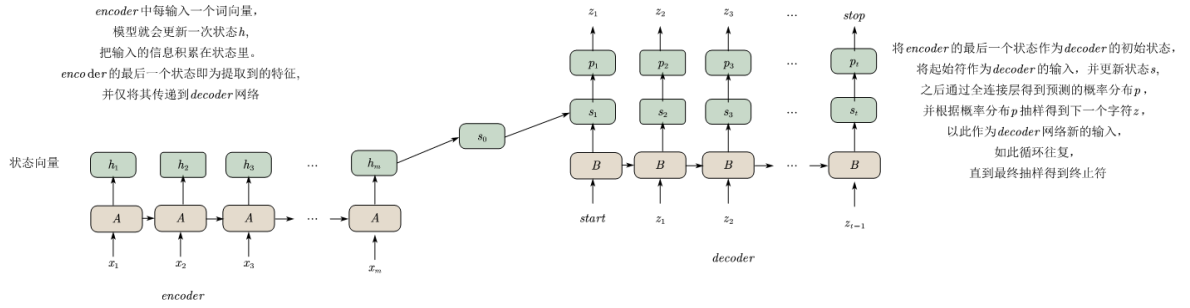
Transformer

1.Attention without RNN

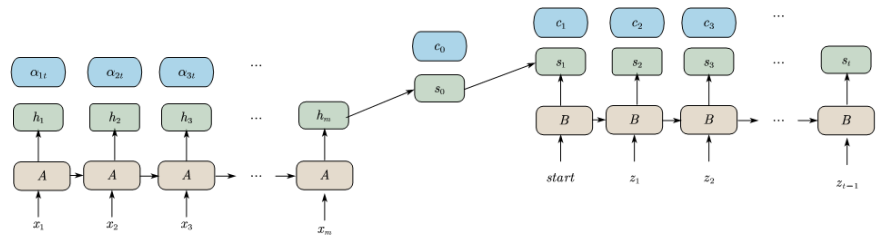
1.1Attention for seq2seq model

Seq2Seq模型由编码器和解码器组成，其中编码器处理序列输入并将信息压缩到固定长度的上下文向量，上下文向量被认为是输入序列的语义概要，而解码器由上下文向量初始化，并每次产生一个解码输出。具体来说在编码器中，每输入一个词向量，模型就会更新一次状态h，把输入的信息压缩到状态向量h中，最后一个状态向量h_m是对所有输入的概括，它将被传递到解码器中作为解码器的初始状态，在解码器中，h_m连同起始符共同作为输入，更新状态s,之后通过全连接层得到预测的概率分布p，并依据概率分布p抽样得到下一个字符z，以此作为decoder中新的输入，如此循环往复，直到最终抽样得到终止符结束。



Attention，即注意力机制，是使我们更加关注具有高解析度或者高辨识度的区域，并根据所观察到的内容进行推断周围区域的情况，反映了人们如何关注图片的某些区域，注意力机制借助重要性权重向量来判断所预测或者推断的元素与其他元素的关联性强度如何，然后对加权的向量求和来近似逼近最终的目标值。

如果在$seq2seq$模型中加入$attention$的话，
则需要计算$context\ vector\ c$，
每计算出一个状态s，就相应的可计算出c



具体计算公式：

$weight: \alpha_{ij} = align(h_i, s_j)$

计算$k_{:i} = W_k h_i (Key, 被q匹配)$，$q_{:j} = W_q s_j (Query, 匹配k)$，$v_{:i} = W_v h_i (Value, 与权重相乘相加)$

$K = [k_{:1}\ \ k_{:2}\ \cdots\ k_{:m}]$

$\alpha_{:j} = soft\max(K^T q_{:j}) \in R^m$

$context\ vector\ c_{:j} = \sum_{i=1}^{m} \alpha_{ij} v_{:i}$

这里的意思就是说我们要拿一个$Query$向量去与m个k向量做对比，
计算出他们之间的相似度（匹配程度）

下面我们剥离RNN，来考虑attention layer。

keys 和values是基于encoder的输入$x_1, x_2, x_3, \ldots x_m$

key: $k_{:i} = W_k * x_i$ (video title)

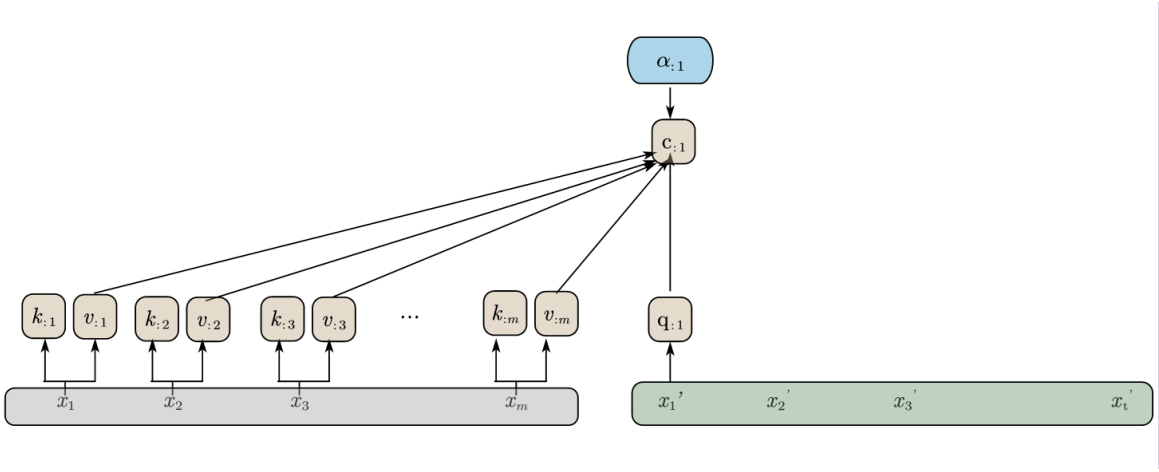
value:$v_{:i} = W_v * x_i$ (video context)

queries是基于decoder的输入$x_1^{'}, x_2^{'}, x_3^{'}, \ldots x_t^{'}$

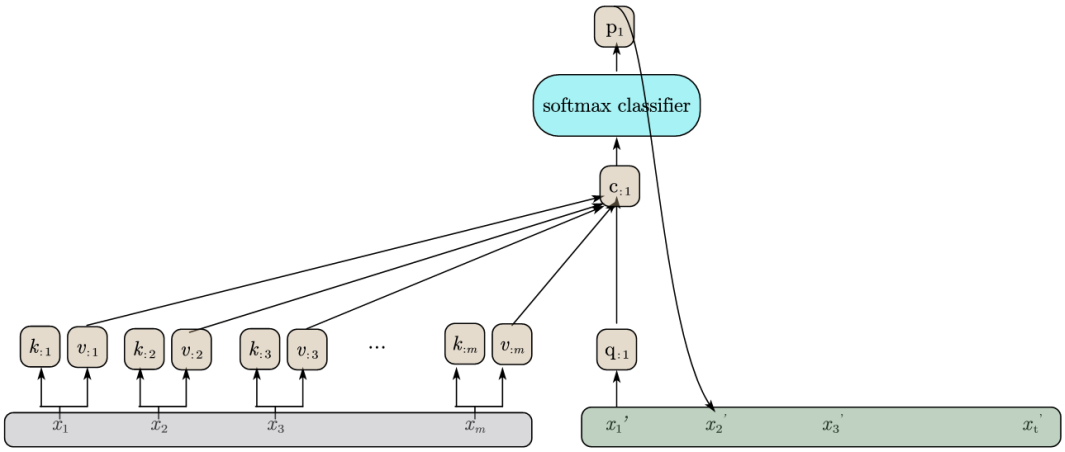
query:$q_{:j} = W_q * x_j^{'}$ (search text)

同上，计算权重$\alpha_{:j} = softmax(K^T * q_{:j})$,并利用的到的权重向量计算context vector:
$c_{:j} = \sum_{i=1}^{m} \alpha_{ij} v_{:i}$

即：$c_{:j} = V * softmax(K^T * q_{:j})$



不难发现output of attention layer:C=[$c_{:1}, c_{:2}, \ldots, c_{:t}$]$\in R_{embed*t}$



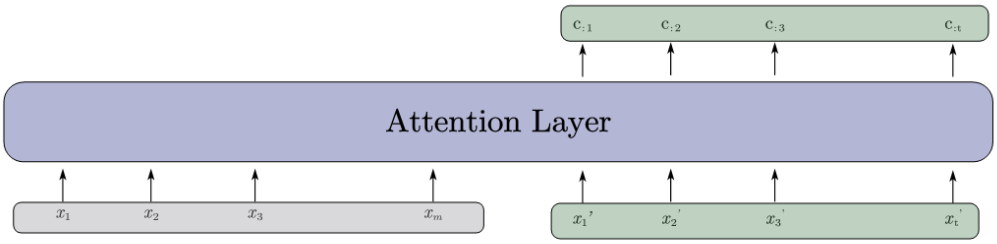
上面我们即构造了一个attention layer:

$C = Attn(X, X')$

Encoder's inputs:$X = [x_1, x_2, \ldots, x_m]$

Decoder's inputs:$X' = [x_1', x_2', \ldots, x_m']$

parameters:W_q, W_k, W_v

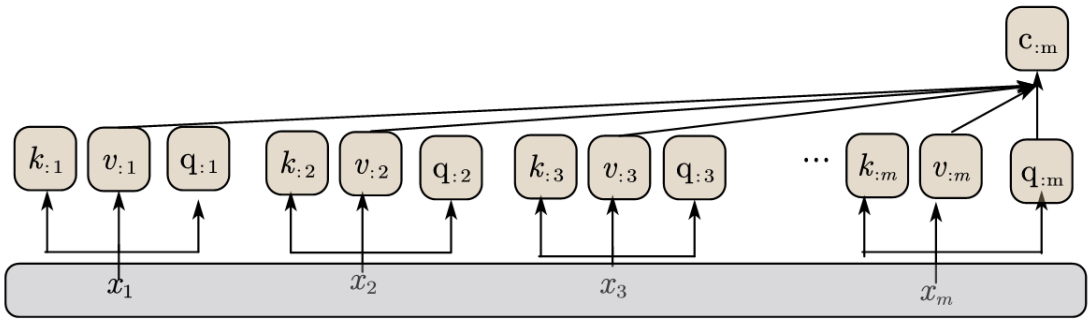


2.Self-Attention without RNN

$C = Attn(X, X)$

Encoder's inputs:$X = [x_1, x_2, \ldots, x_m]$

parameters:W_q, W_k, W_v



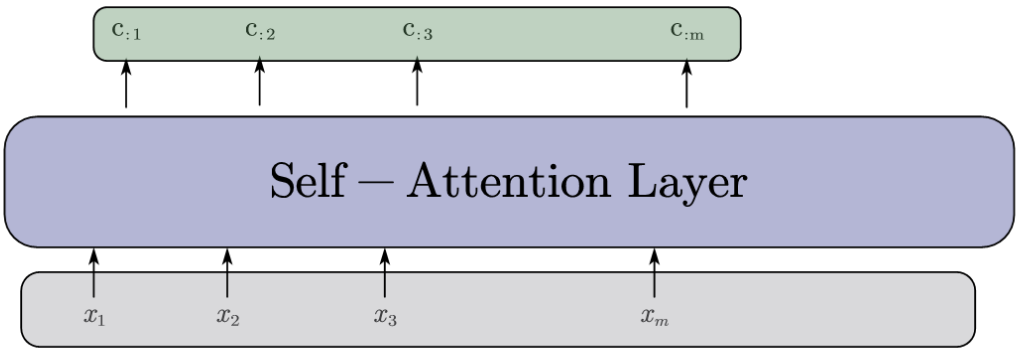
key: $k_{:i} = W_k * x_i$ (video title)

value:$v_{:i} = W_v * x_i$ (video context)

query:$q_{:j} = W_q * x_j$ (search text)

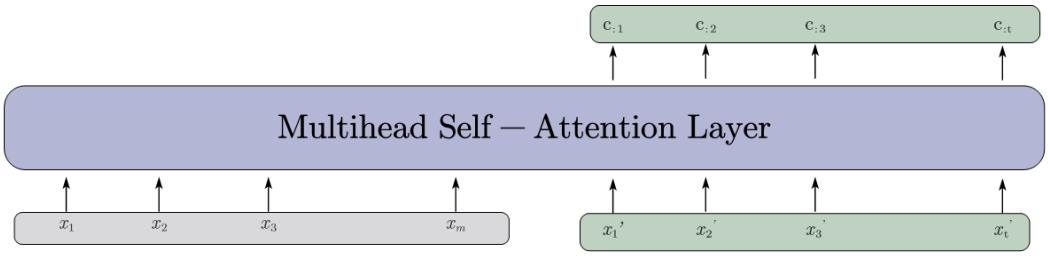
同上，计算权重$\alpha_{:j} = softmax(K^T * q_{:j})$,并利用的到的权重向量计算context vector:
$c_{:j} = \sum_{i=1}^{m} \alpha_{ij} v_{:i}$

即：$c_{:j} = V * softmax(K^T * q_{:j})$

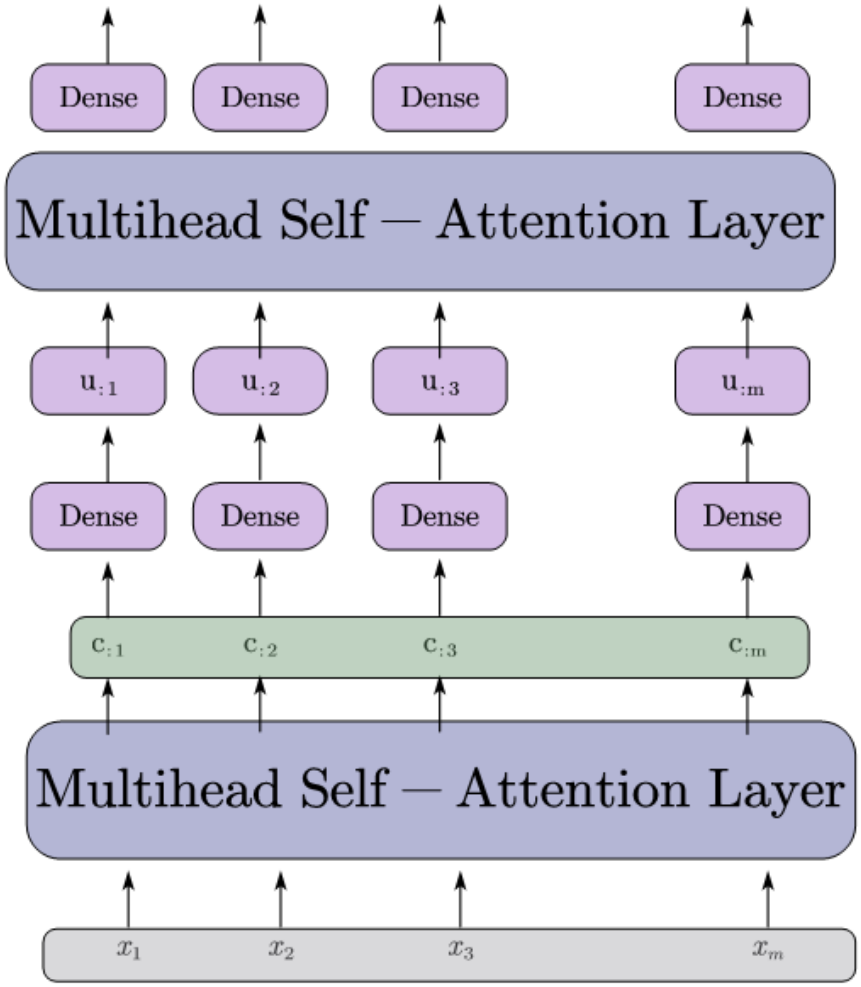


3.Multi-head self-attention

使用l个single head self-attention,每个self-attention之间不共享参数，由于每个单头有三个参数向量，因此该多头注意力机制共有3l个参数向量。每个单头注意力机制都有相同的输入，但是他们的参数矩阵各不相同，因此输出的c也不相同，把这l个单头的输出连接起来作为多头注意力机制的输出，如果每个单头的输出都是$d * m$维的矩阵，那么多头的输出的形状即为$(ld) * m$。



4.Stacked Self-Attention Layers



5.transformer

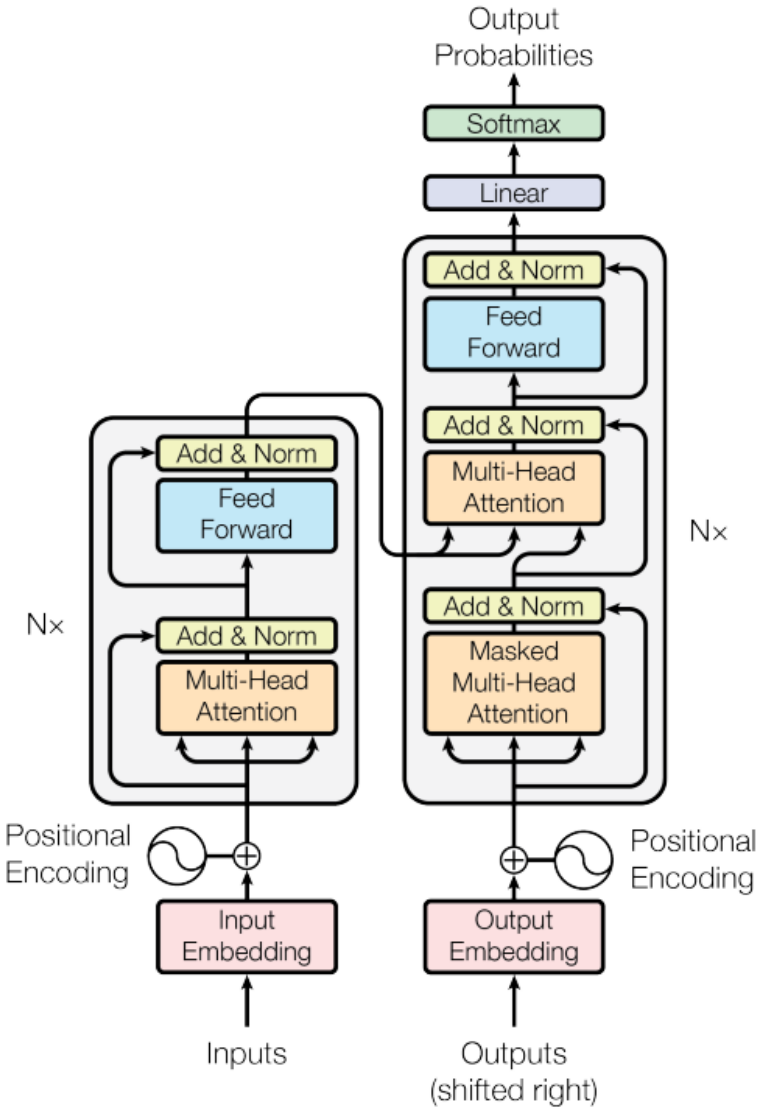


Figure 1: The Transformer - model architecture.