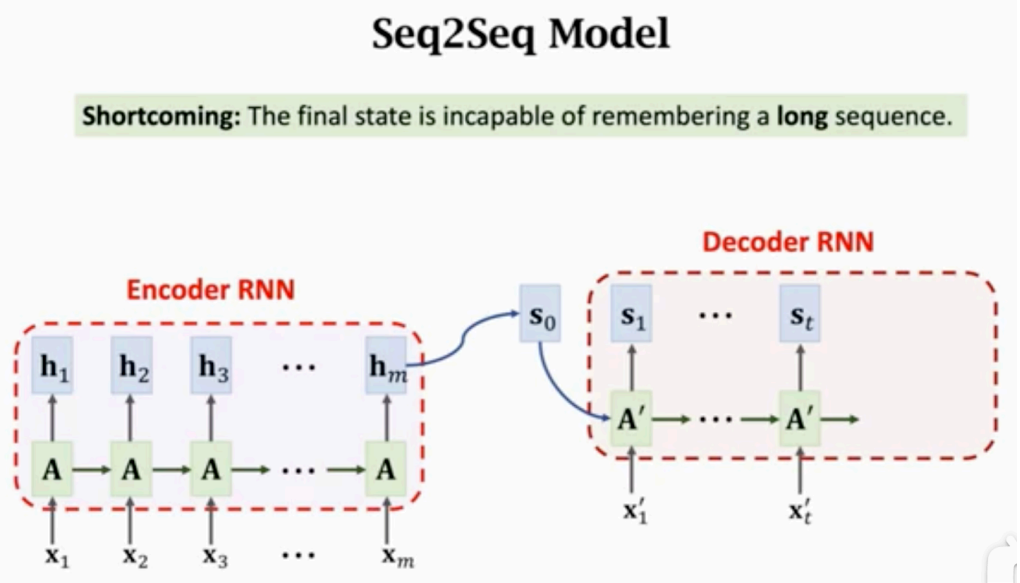
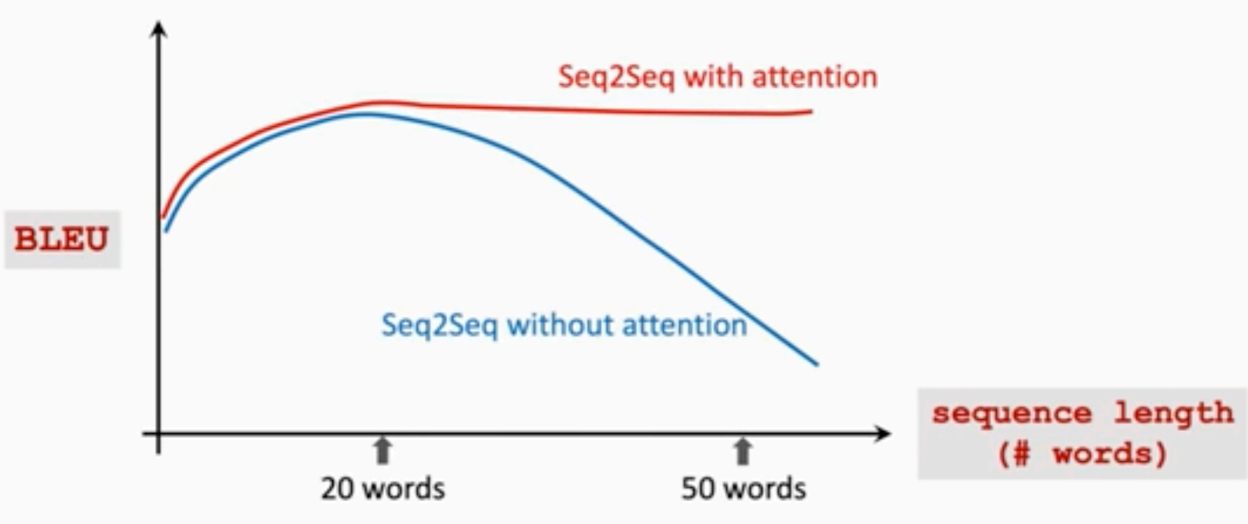
Attention



缺点：RNN模型在处理较长序列时会遗忘



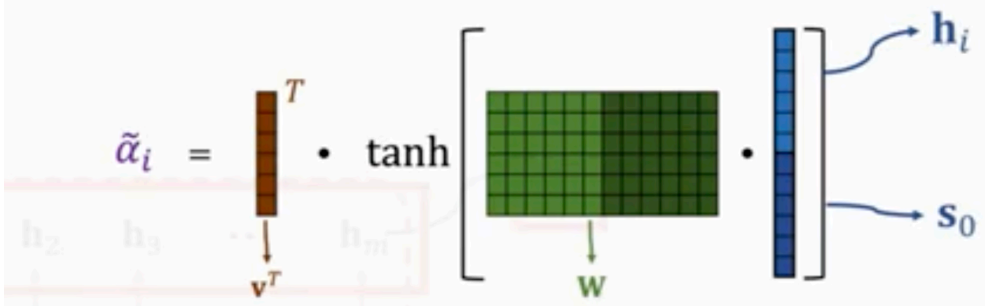
- Attention tremendously improves Seq2Seq model.
- With Attention, Seq2Seq model does not forget source input.
- With Attention, the decoder knows where to focus.
- Downside: much more computation.

计算相关性：$Weight:\alpha_i = align(h_i, s_0)$ 每个输入都计算一个α，所有α相加得到1

h_i是每个状态积累的不同信息

最后一个h_m积累了所有信息 并将其作为decoder RNN的输入s_0

Option 1 (used in the original paper):



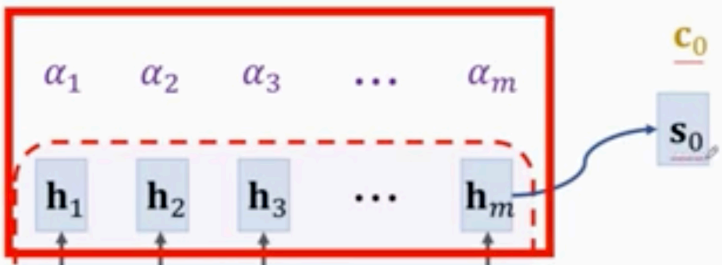
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

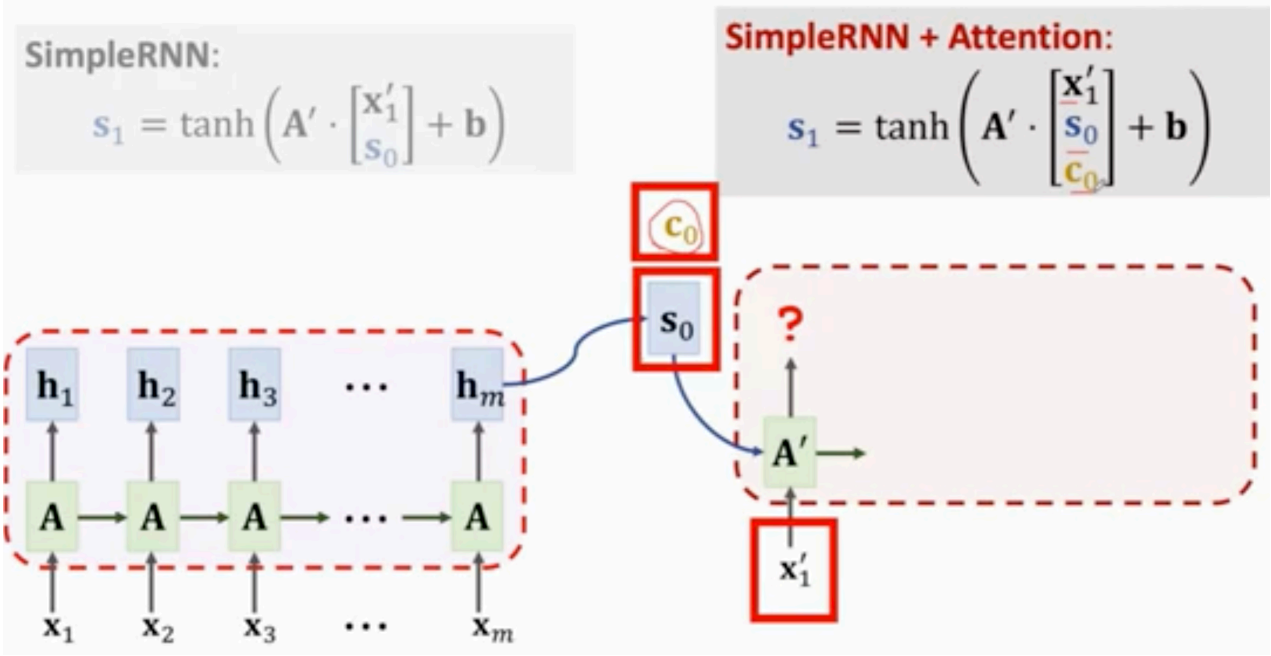
$$[\alpha_1, \cdots, \alpha_m] = \text{Softmax}([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]).$$

Option 2 (more popular; the same to Transformer):

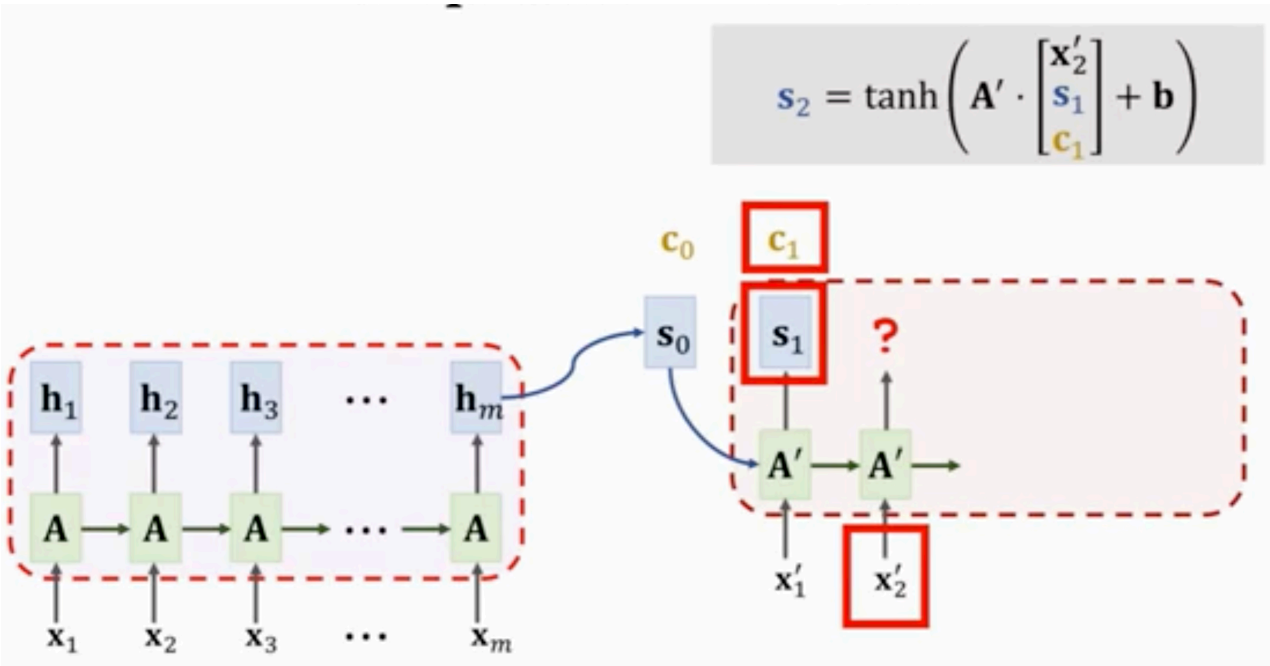
1. Linear maps:
   - $\mathbf{k}_i = \mathbf{W}_K \cdot \mathbf{h}_i$, for $i = 1$ to m.
   - $\mathbf{q}_0 = \mathbf{W}_Q \cdot \mathbf{s}_0$.
2. Inner product:
   - $\tilde{\alpha}_i = \mathbf{k}_i^T \mathbf{q}_0$, for $i = 1$ to m.
3. Normalization:
   - $[\alpha_1, \cdots, \alpha_m] = \text{Softmax}([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m])$.

Context vector: $\mathbf{c}_0 = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m$.

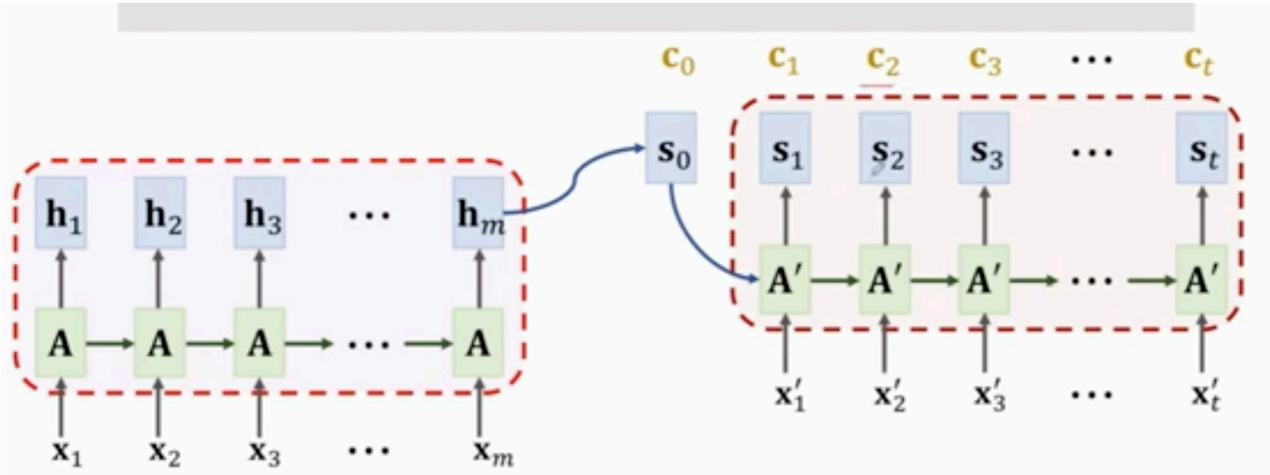




更新每一个decoder中的s_i状态都需要重新计算一次相关性



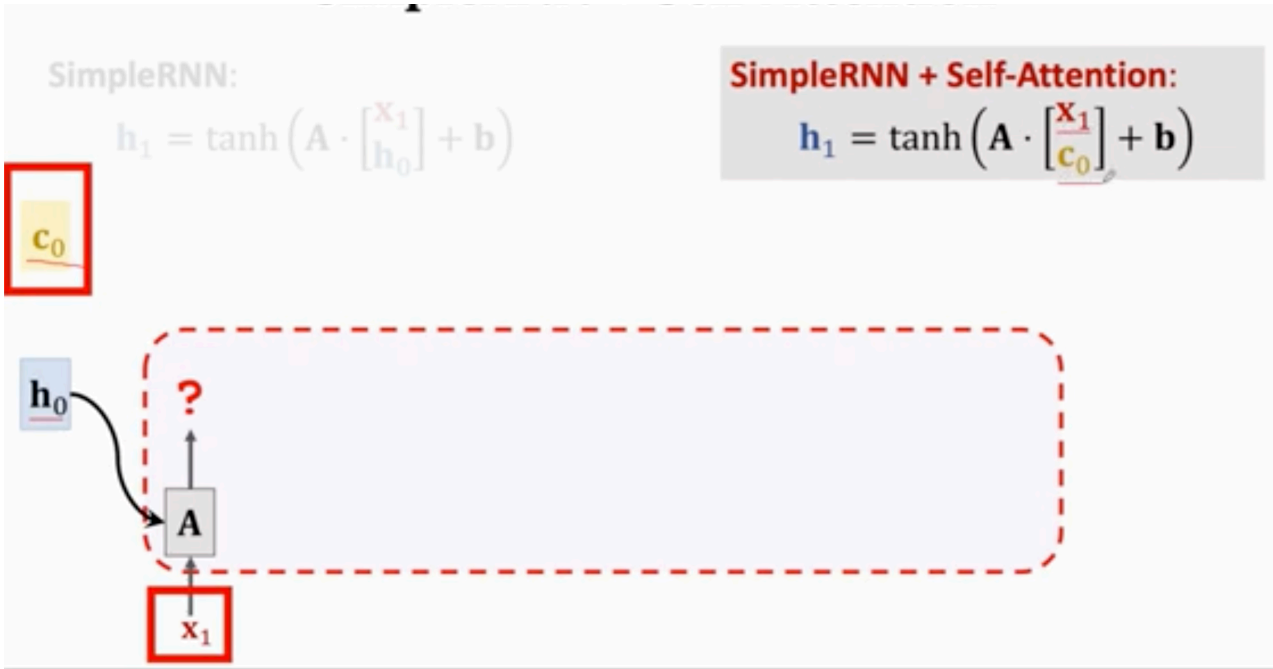
x'_i是前一个状态的输出　x'_1是起始符号



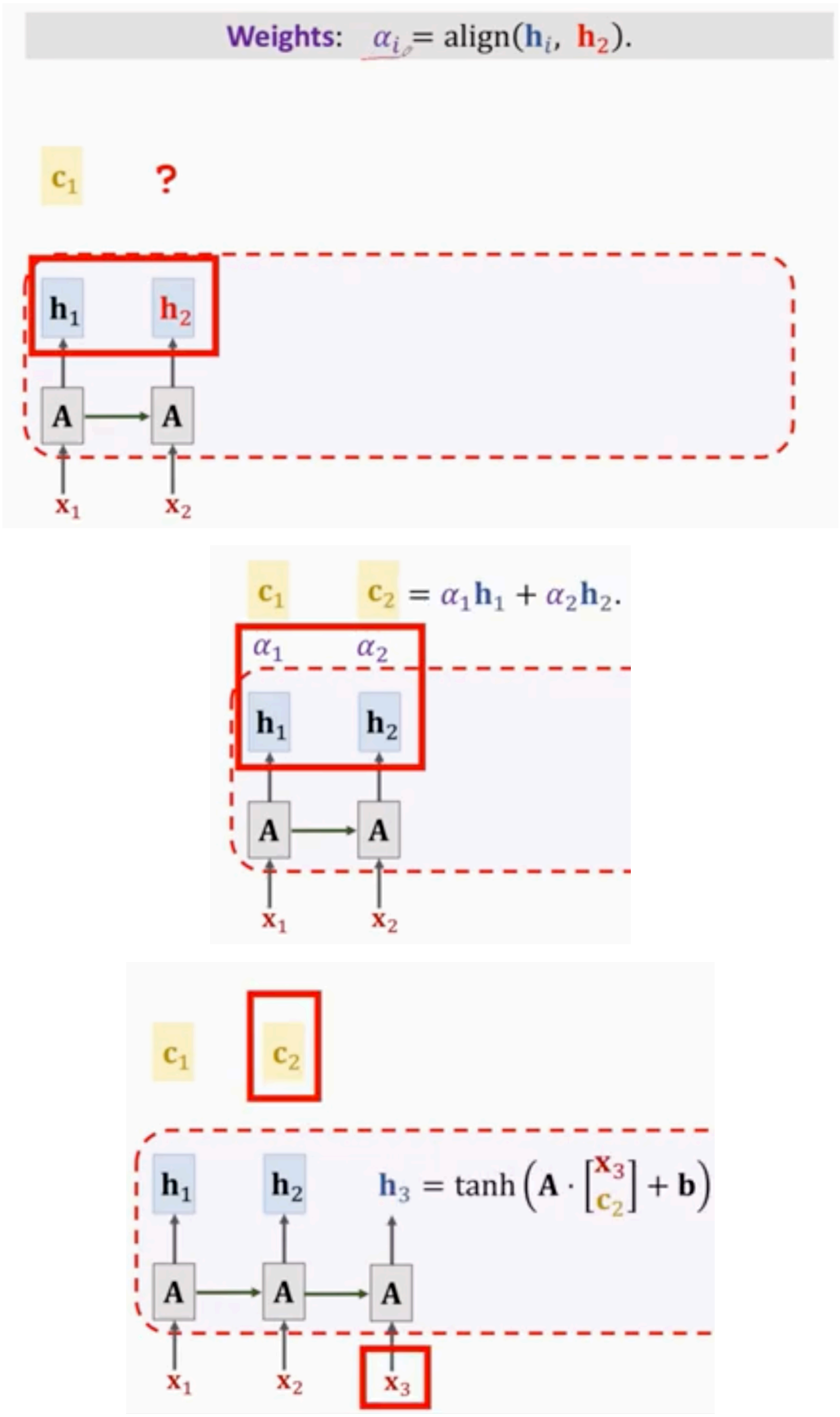
- Standard Seq2Seq model: the decoder looks at only its current state.
- Attention: decoder additionally looks at all the states of the encoder.
- Attention: decoder knows where to focus.

- Downside: higher time complexity.
  - m: source sequence length
  - t: target sequence length
  - Standard Seq2Seq: $O(m+t)$ time complexity
  - Seq2Seq + attention: $O(mt)$ time complexity

Self-Attention



初始状态c_0和h_0都是全零向量，A是参数矩阵


Weights: α_i = align($\mathbf{h}_i$, $\mathbf{h}_2$).
$\mathbf{c}_1$
?
$\mathbf{h}_1$
$\mathbf{h}_2$
A
A
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{c}_1$
$\mathbf{c}_2 = \alpha_1\mathbf{h}_1 + \alpha_2\mathbf{h}_2$.
α_1
α_2
$\mathbf{h}_1$
$\mathbf{h}_2$
A
A
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{c}_1$
$\mathbf{c}_2$
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3 = \tanh\left(\mathbf{A} \cdot \begin{bmatrix}\mathbf{x}_3 \\ \mathbf{c}_2\end{bmatrix} + \mathbf{b}\right)$
A
A
A
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$

计算流程

- 首先计算c_i （先求相似度得到α再加权平均）
- 再求得h_i

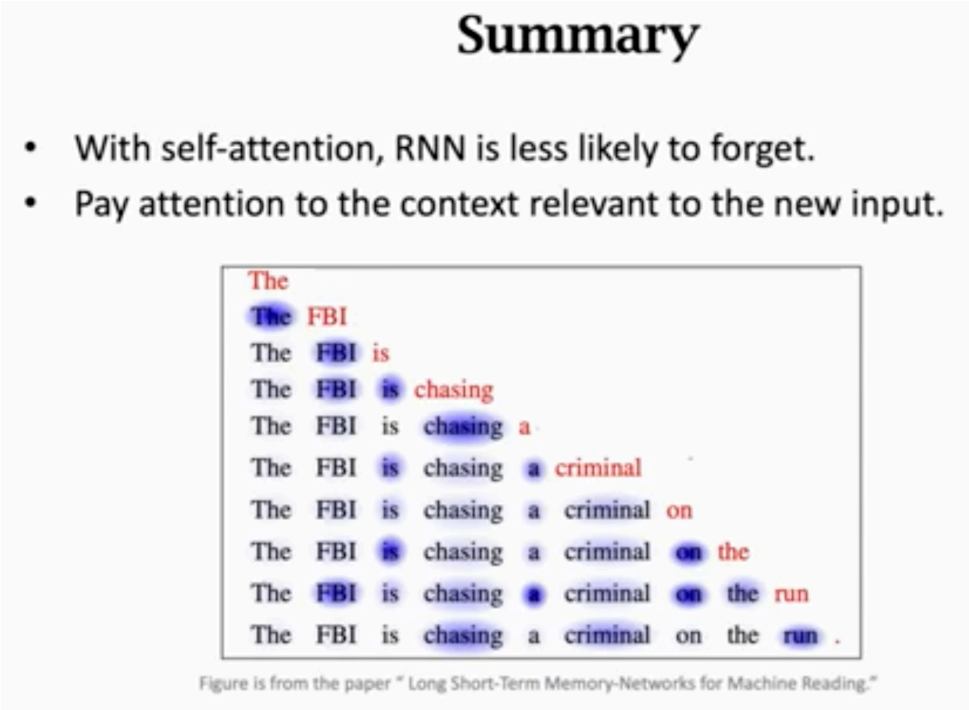


Figure is from the paper " Long Short-Term Memory-Networks for Machine Reading."

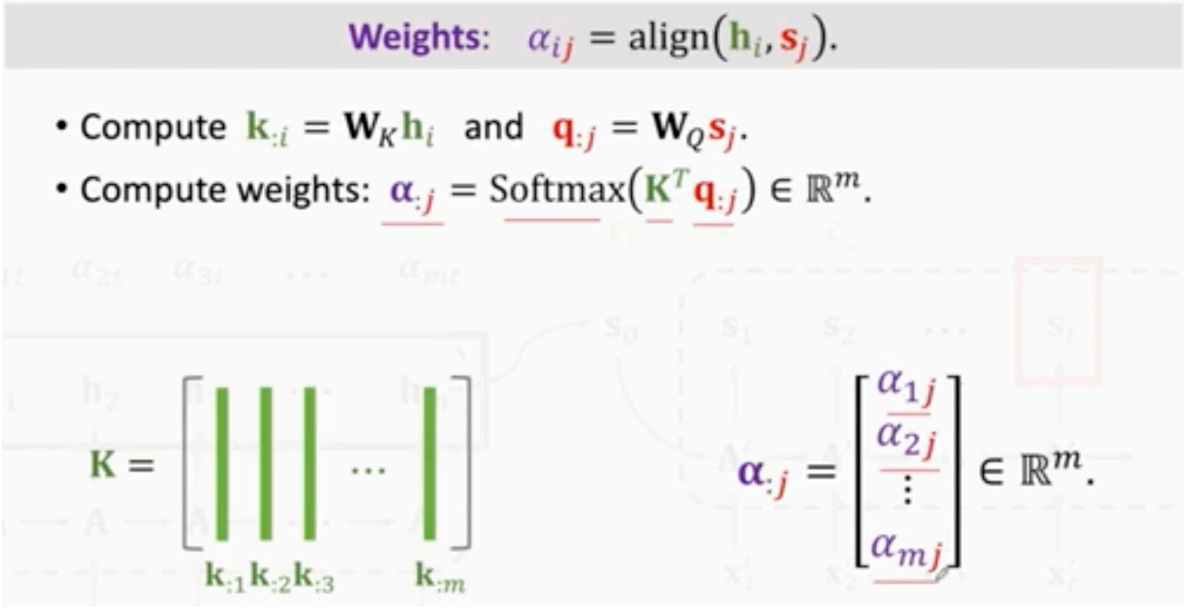
- self-attention可以用矩阵运算快速得到相似度
- $E * E^T$　　QKV（query、key、value）在self里面都是一样的

Attention is All You Need

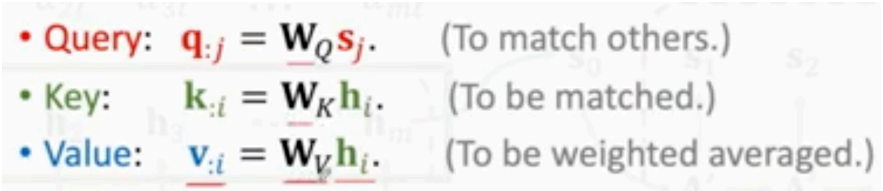
即 Transformer

- Transformer is a seq2seq model.
- Transformer is not RNN.
- Purely based on attention and dense layers. 只有Attention和全连接层
- Higher accuracy than RNNs on large datasets.

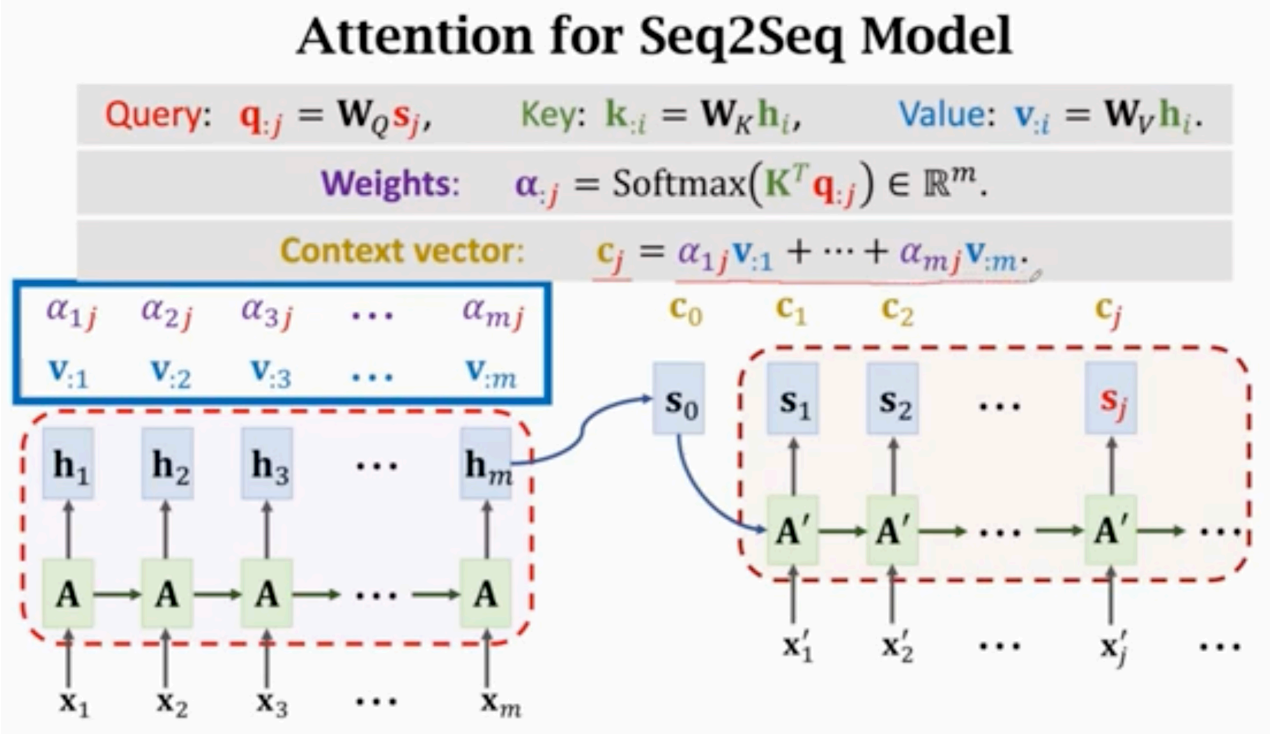
Attention for Seq2Seq Model



$\alpha_{:j}$中的元素都介于0-1之间，并且相加和为1



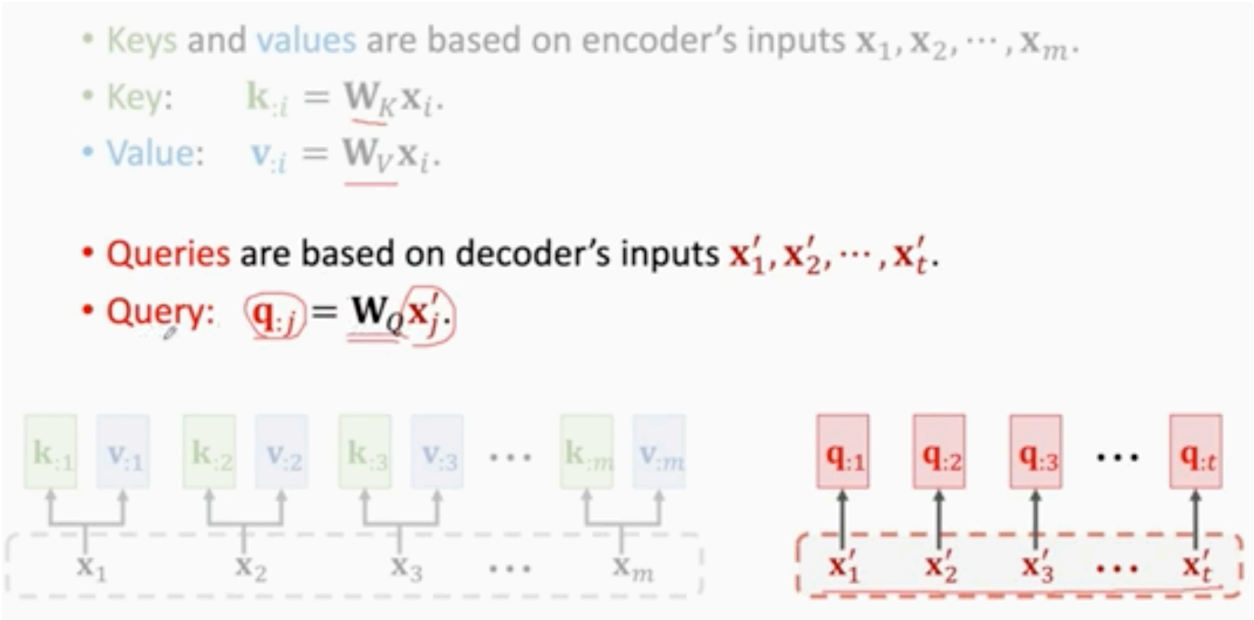
$\alpha_{:j}$说明的是Query和Key的匹配程度



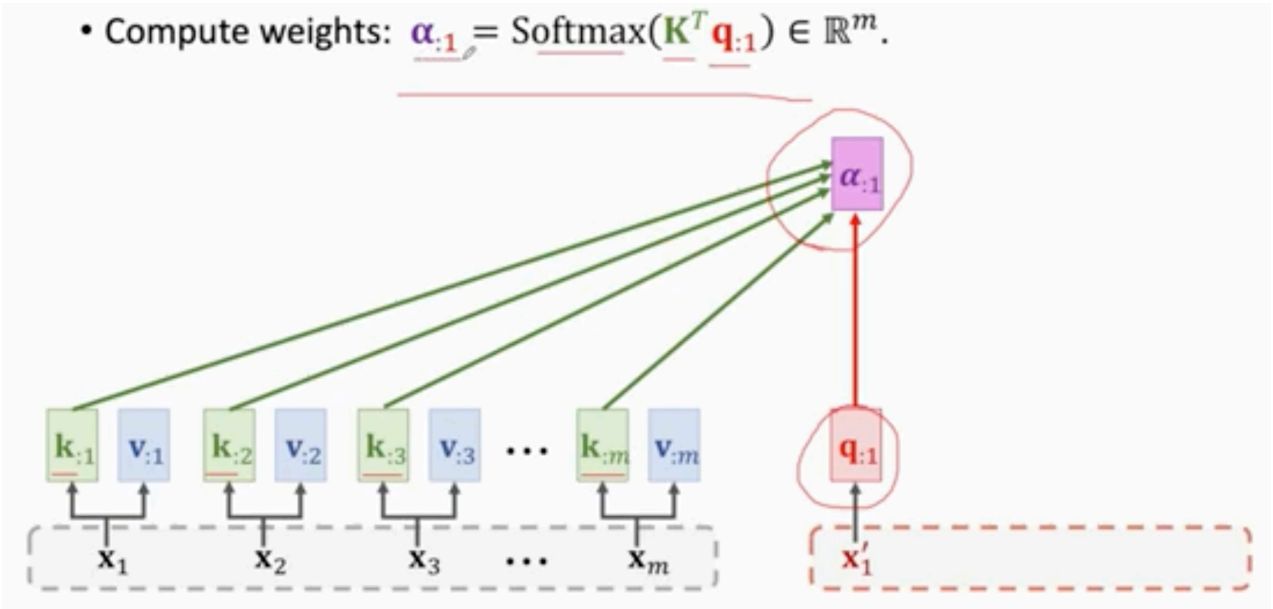
Attention Without RNN

具体步骤

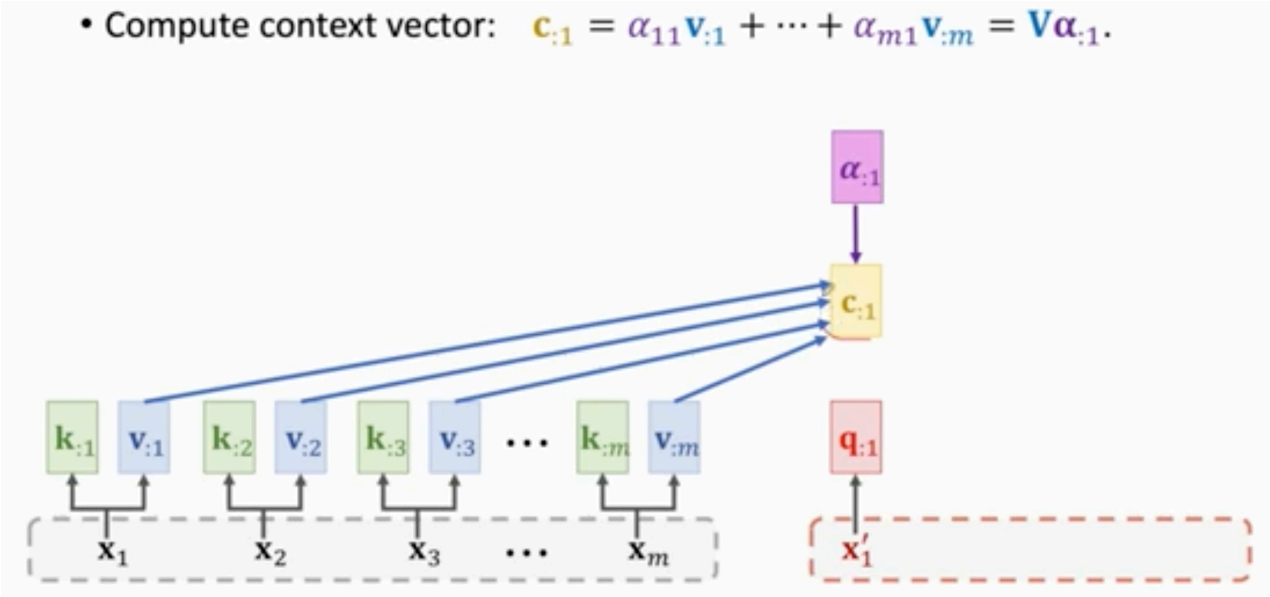
- 首先分别计算K V Q矩阵



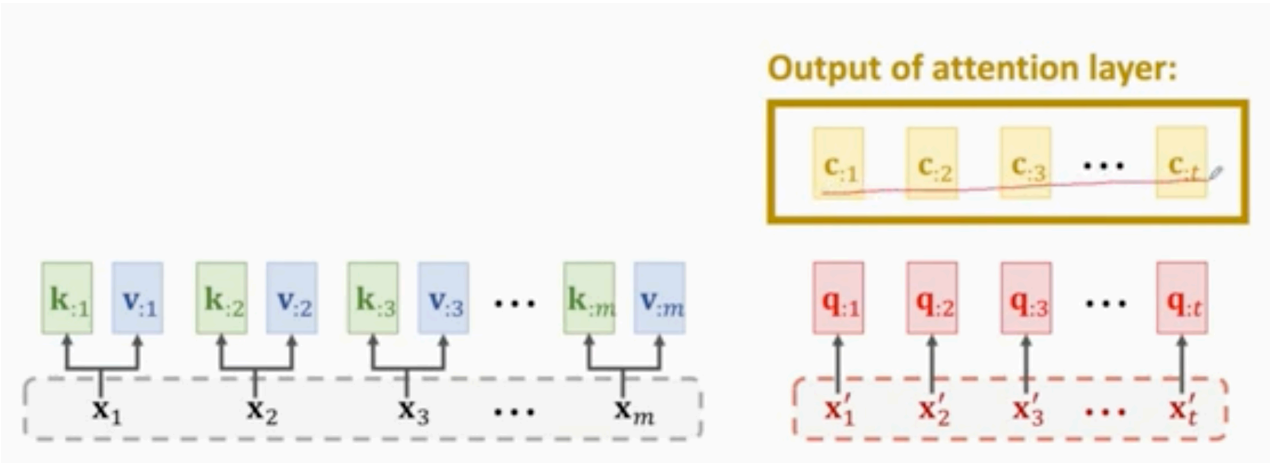
- 计算$\alpha_{:1}$



- 计算$c_{:1}$

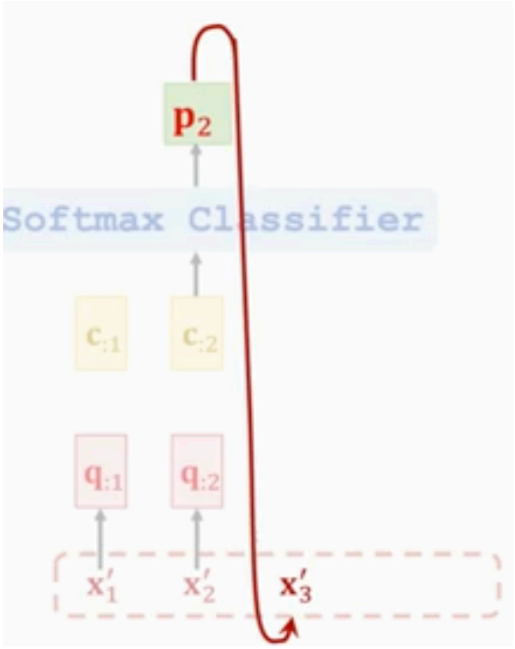


- 得到最终的c向量

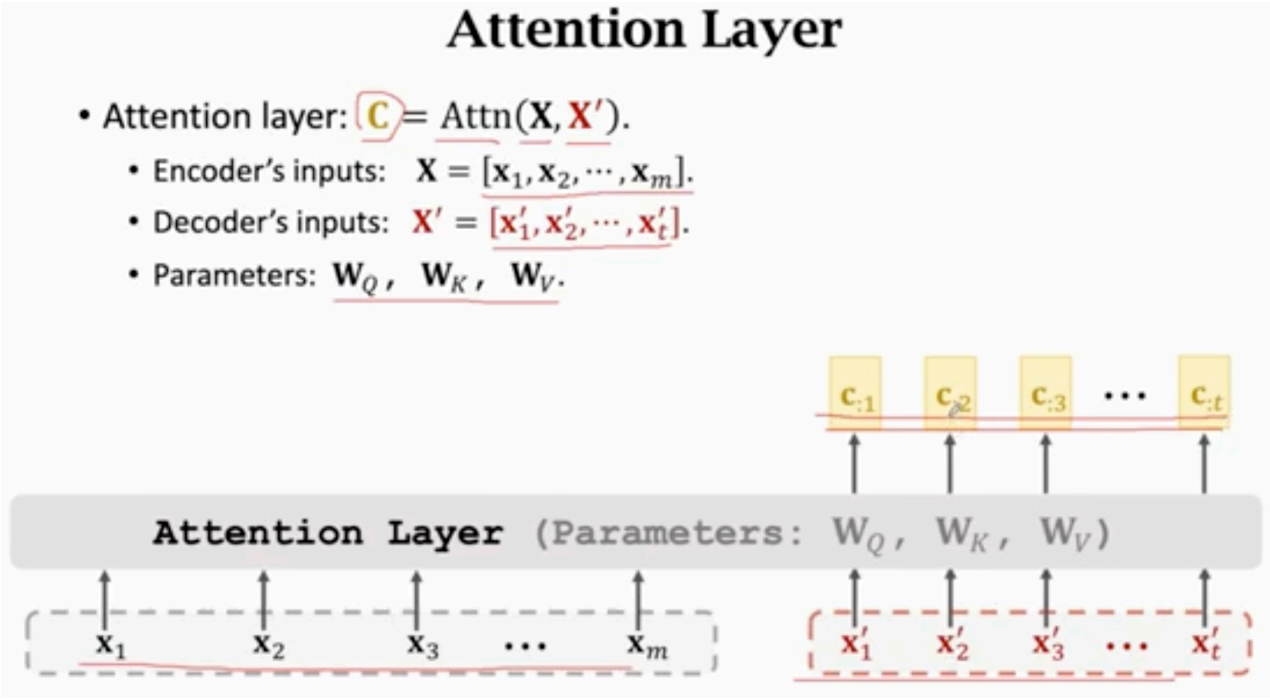


- Output of attention layer: $\mathbf{C} = [\mathbf{c}_{:1}, \mathbf{c}_{:2}, \mathbf{c}_{:3}, \cdots, \mathbf{c}_{:t}]$.
- Here, $\mathbf{c}_{:j} = \mathbf{V} \cdot \mathrm{Softmax}(\mathbf{K}^T \mathbf{q}_{:j})$.
- Thus, $\mathbf{c}_{:j}$ is a function of $\mathbf{x}'_j$ and $[\mathbf{x}_1, \cdots, \mathbf{x}_m]$.

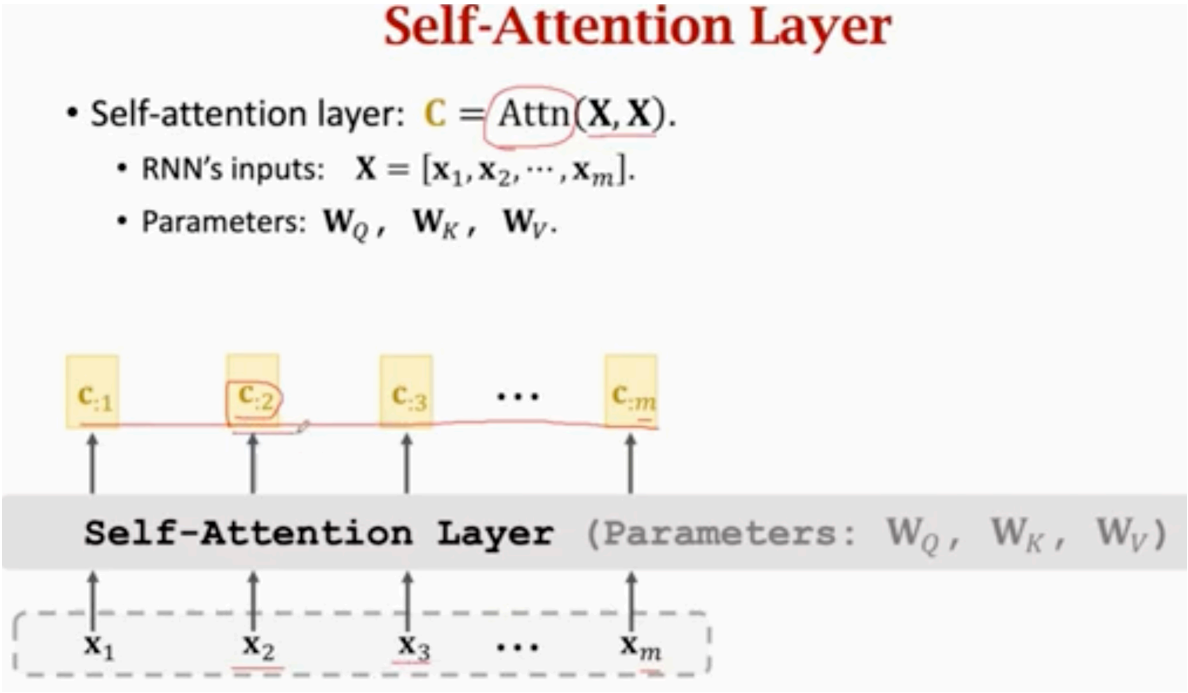
- 将C作为特征向量 这样的结构与RNN类似



Attention Layer的最终结构

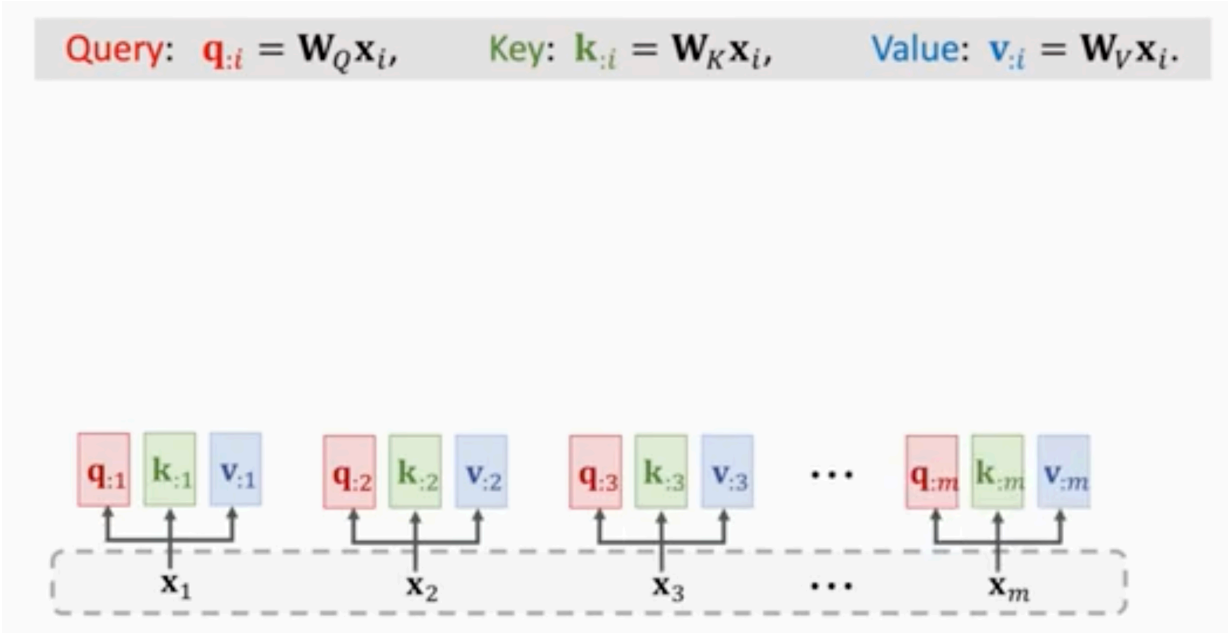


Self-Attention Without RNN

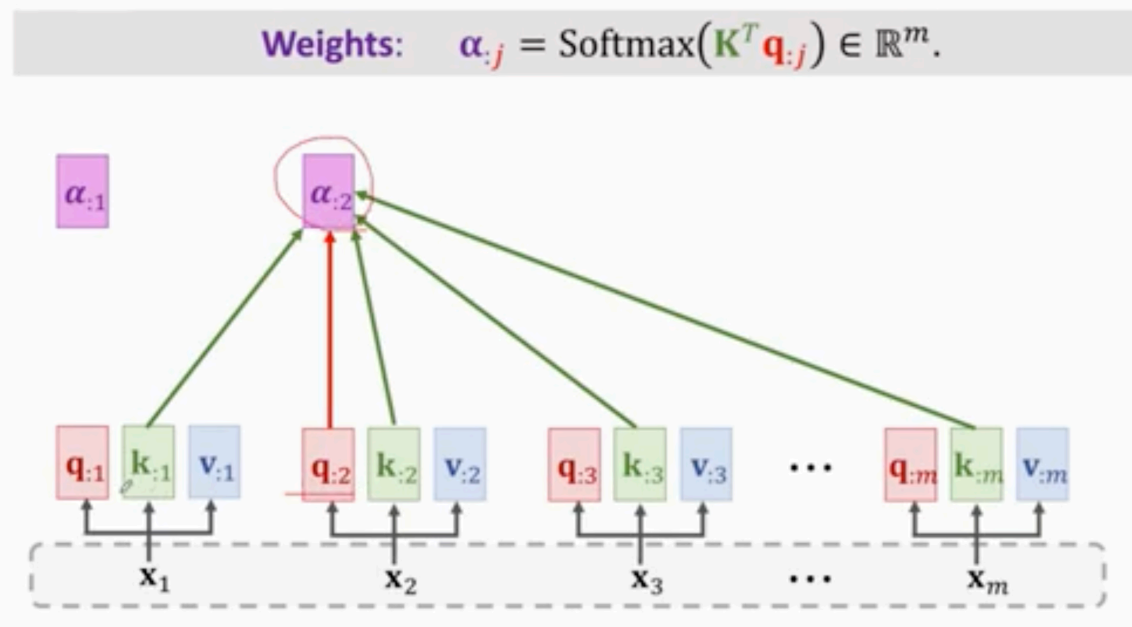


具体步骤

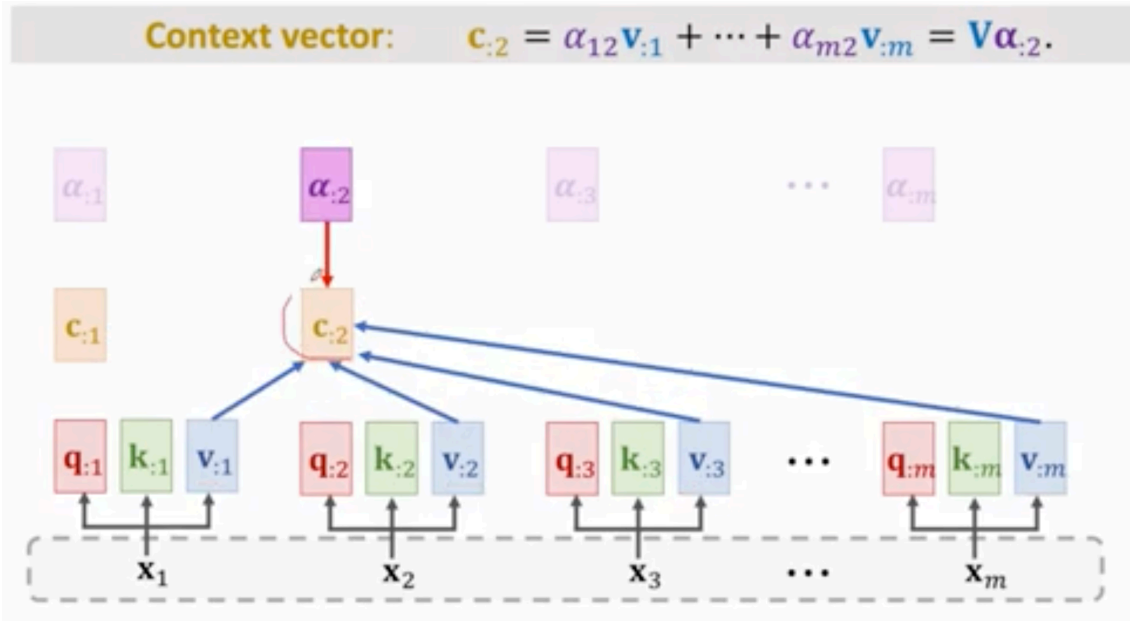
- 分别计算Q、K、V



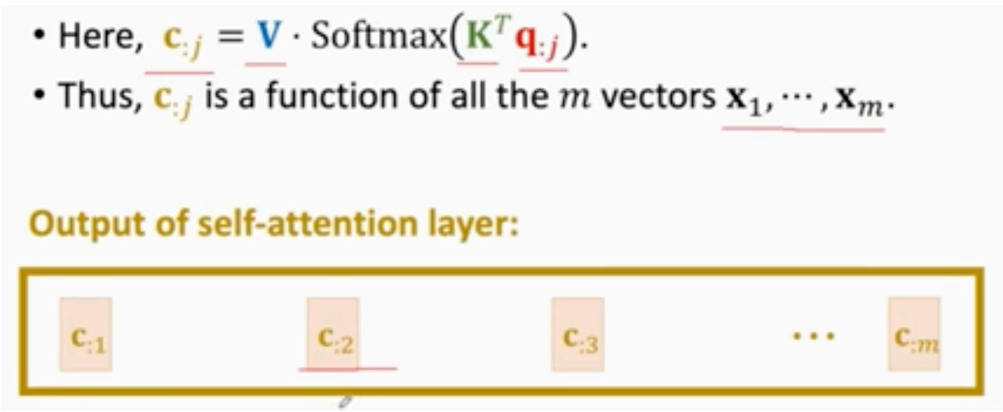
- 计算权重向量


Weights: $\alpha_{:j} = \mathrm{Softmax}(\mathbf{K}^T\mathbf{q}_{:j}) \in \mathbb{R}^m$.
$\alpha_{:1}$
$\alpha_{:2}$
$\mathbf{q}_{:1}$ $\mathbf{k}_{:1}$ $\mathbf{v}_{:1}$
$\mathbf{q}_{:2}$ $\mathbf{k}_{:2}$ $\mathbf{v}_{:2}$
$\mathbf{q}_{:3}$ $\mathbf{k}_{:3}$ $\mathbf{v}_{:3}$
$\mathbf{q}_{:m}$ $\mathbf{k}_{:m}$ $\mathbf{v}_{:m}$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_m$

- 计算C


Context vector: $\mathbf{c}_{:2} = \alpha_{12}\mathbf{v}_{:1} + \cdots + \alpha_{m2}\mathbf{v}_{:m} = \mathbf{V}\boldsymbol{\alpha}_{:2}$.
$\alpha_{:1}$ $\alpha_{:2}$ $\alpha_{:3}$ $\alpha_{:m}$
$\mathbf{c}_{:1}$ $\mathbf{c}_{:2}$
$\mathbf{q}_{:1}$ $\mathbf{k}_{:1}$ $\mathbf{v}_{:1}$
$\mathbf{q}_{:2}$ $\mathbf{k}_{:2}$ $\mathbf{v}_{:2}$
$\mathbf{q}_{:3}$ $\mathbf{k}_{:3}$ $\mathbf{v}_{:3}$
$\mathbf{q}_{:m}$ $\mathbf{k}_{:m}$ $\mathbf{v}_{:m}$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_m$

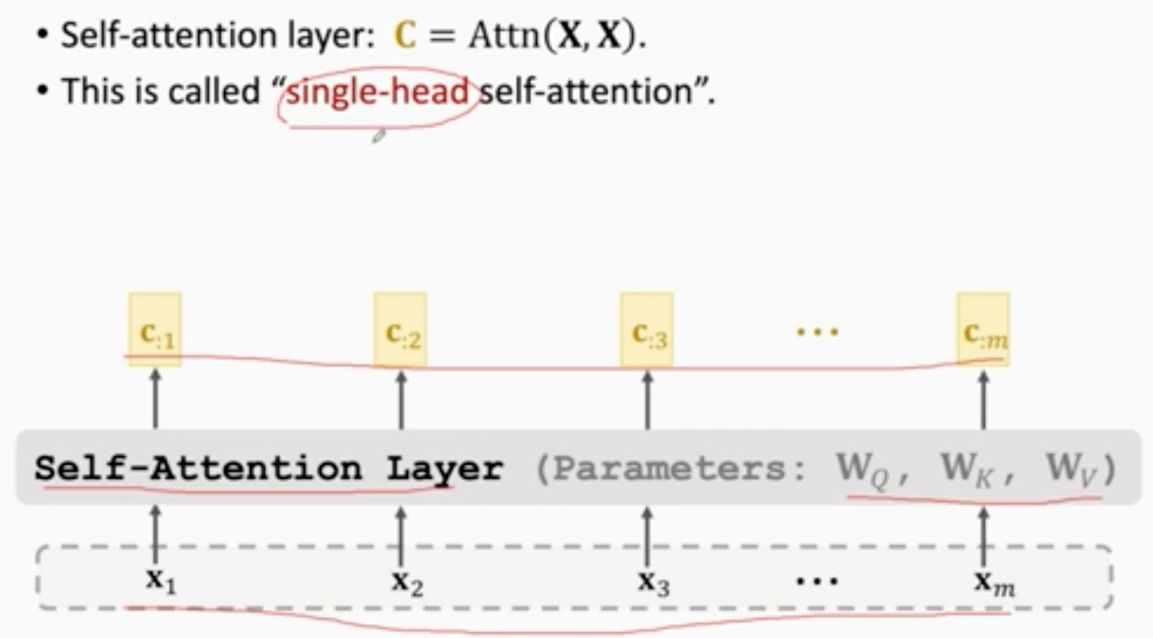
- 总结


• Here, $\mathbf{c}_{:j} = \mathbf{V} \cdot \mathrm{Softmax}(\mathbf{K}^T\mathbf{q}_{:j})$.
• Thus, $\mathbf{c}_{:j}$ is a function of all the m vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$.
Output of self-attention layer:
$\mathbf{c}_{:1}$ $\mathbf{c}_{:2}$ $\mathbf{c}_{:3}$ $\mathbf{c}_{:m}$

Summary

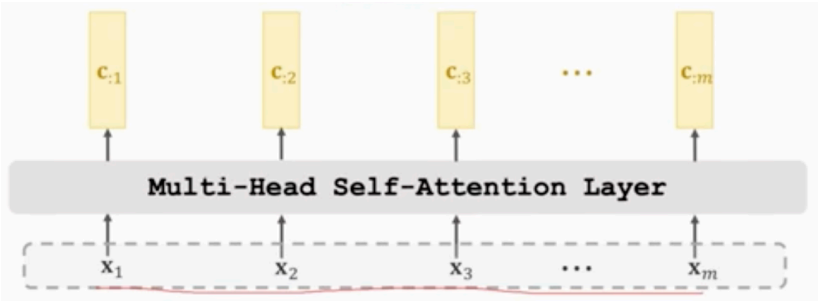
- Attention was originally develop for Seq2Seq RNN models.
- Self-attention: attention for all the RNN models (no necessarily Seq2Seq models).
- Attention can be used without RNN.

Transformer Model（From Shallow to Deep）

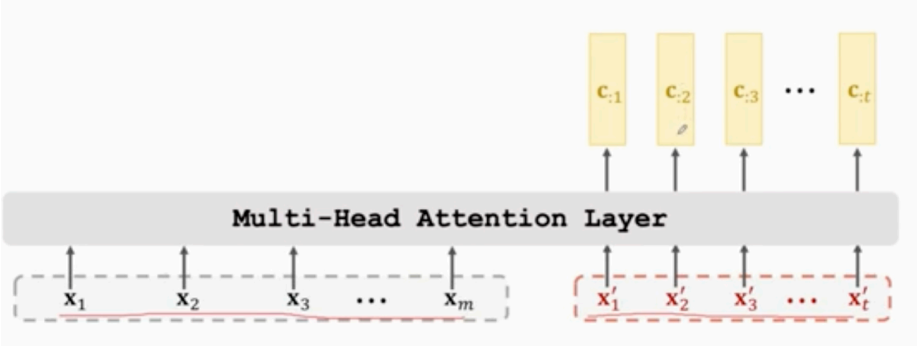


- Using l single-head self-attentions (which do not share parameters.)
  - A single-head self-attention has 3 parameter matrices: W_Q, W_K, W_V
  - Totally 3l parameters matrices.
  - Concatenating outputs of sigle-head self-attentions.
    - Suppose single-head self-attentions' outputs are d*m matrices.
    - Multi-head's output shape: (ld)*m.

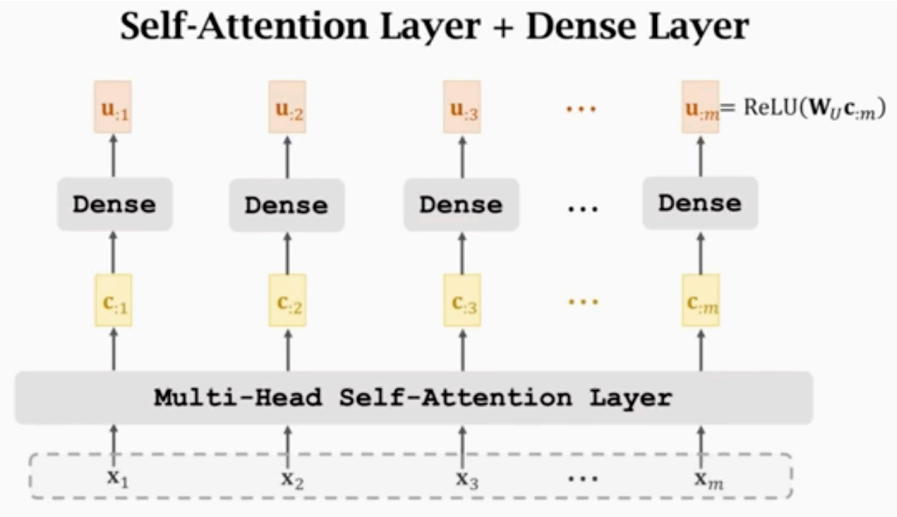
1. Multi-head Attention



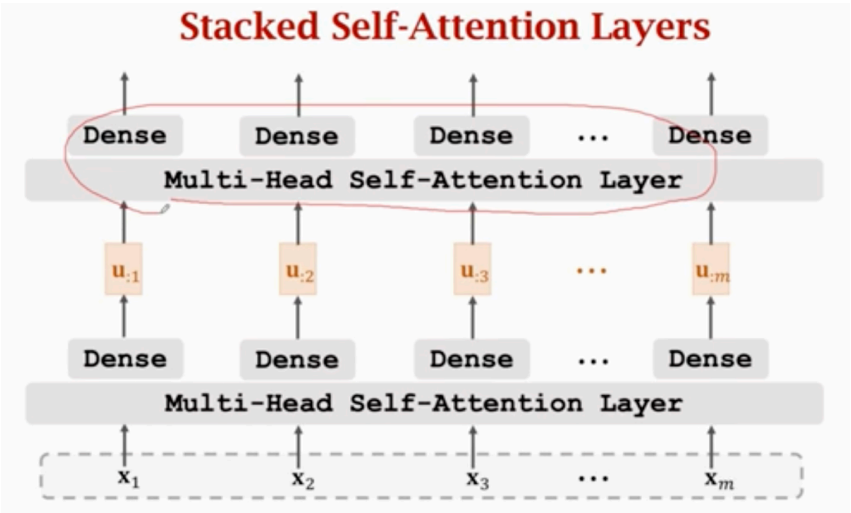
2. Multi-head Self-Attention



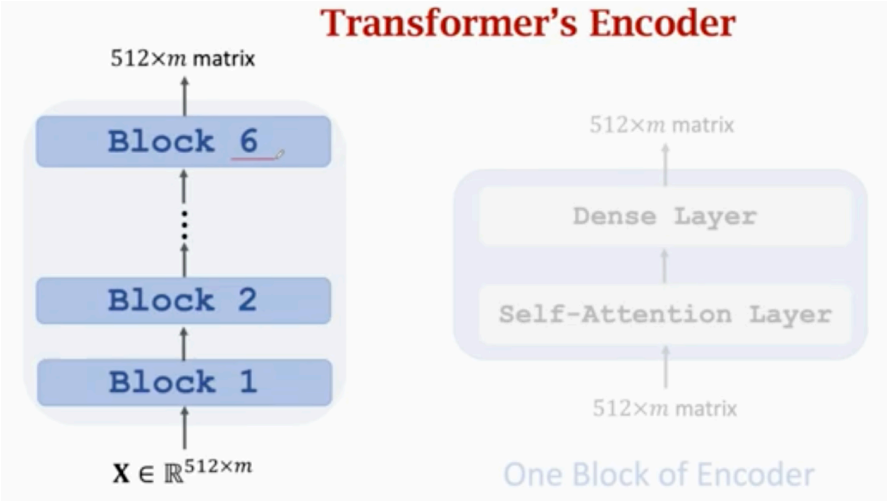
Stacked Self-Attention Layers



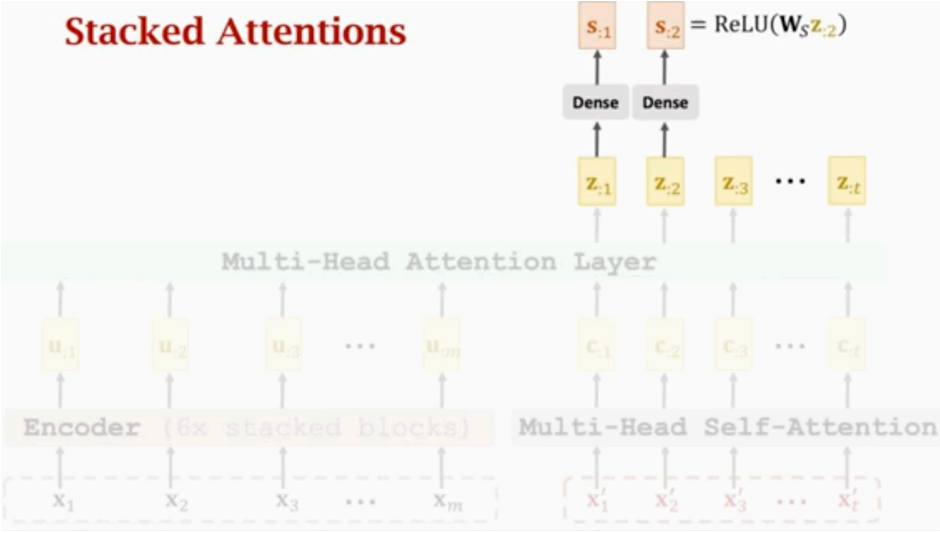
- Dense表示全连接层 每一个Dense层的权重都完全相同
- 输出u_i依赖于x_i . . . . . x_m， 改变任何一个x的值都会对u_i产生影响，但是对u_i影响最大的还是x_i



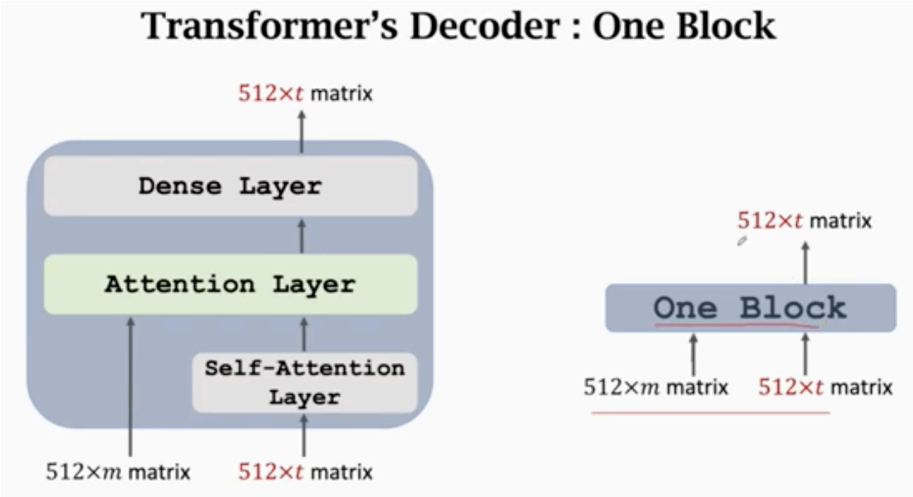
Transformer's Encoder



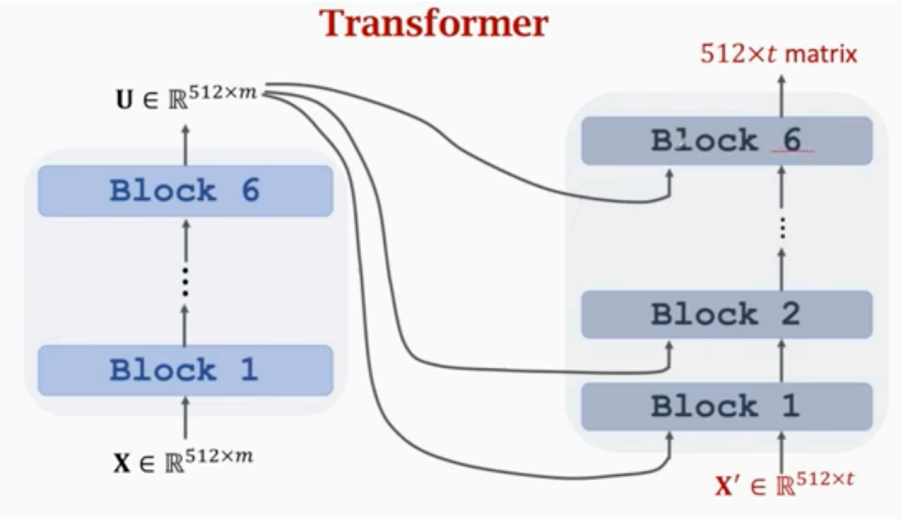
- block之间不共享参数
- Transformer's encoder contains 6 stacked blocks.
- 1block ≈ 1 multi-head attention layer + 1 dense layer.



两个输入
一个输出

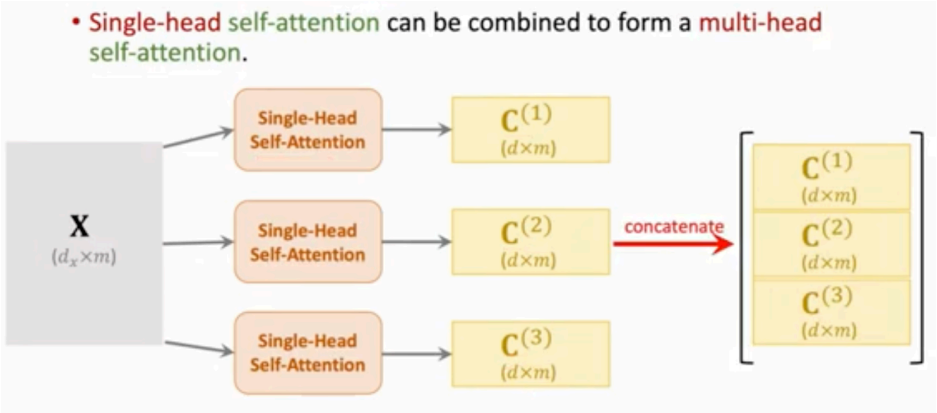


Put Everything Together

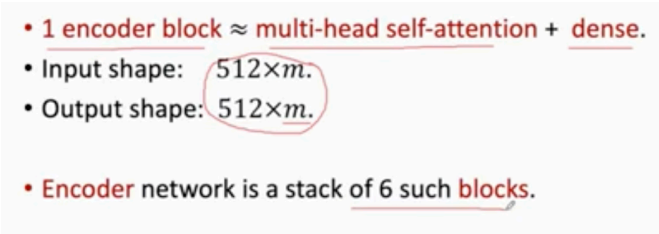


Summary

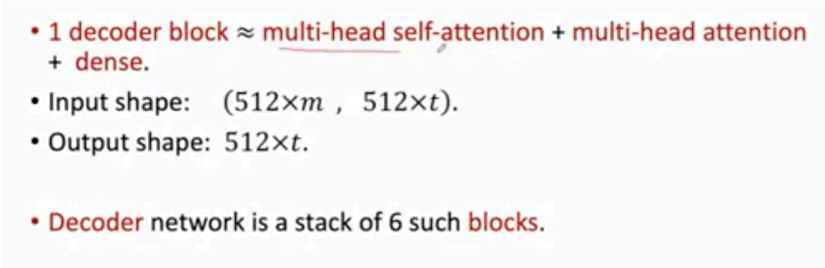
- From Single-Head to Multi-head



- Encoder Network of Transformer



- Decoder Network of Transformer



- Transformer Model

- Transformer is Seq2Seq model; it has an encoder and a decoder.
- Transformer model is not RNN.
- Transformer is based on attention and self-attention.
- Transformer outperforms all the state-of-the-art RNN models.