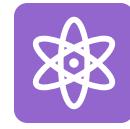


《数据挖掘技术》

# ★ CHX1 Self-Attention & Bert

→ Created by *Wang JingHui*

→ Version: 4.0



# 主要内容

- Attention
- Self Attention
- Transformer
- Bert



# Attention

Attention 机制最早的提出是针对序列模型的，出处是 Bengio 大神在2015年的这篇文章：

- Neural Machine Translation by jointly learning to align and translate, Bengio et. al. ICLR 2015
- [arxiv.org/pdf/1409.0473.pdf](https://arxiv.org/pdf/1409.0473.pdf)

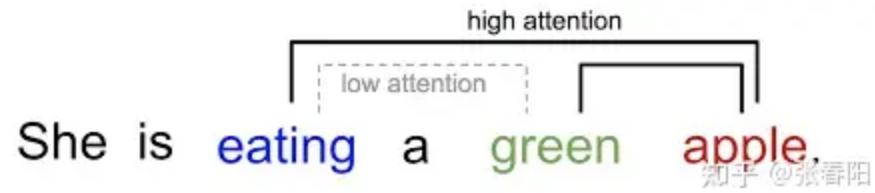
Bengio 大神借鉴了生物在观察、学习、思考行为中的过程的一种独特的生理机制，这种机制就是 Attention 机制。大家都能有感觉，我们在获取信息的时候，通常是先从宏观上建立一个比较模糊的认识，然后又在红馆认识下，发现一些比较重要的信息，对于这些重要的信息，我们花费更多的注意力进行观察、学习和思考。

# 机器视觉

在视觉任务，应该怎么去理解 Attention：



# 自然语言处理

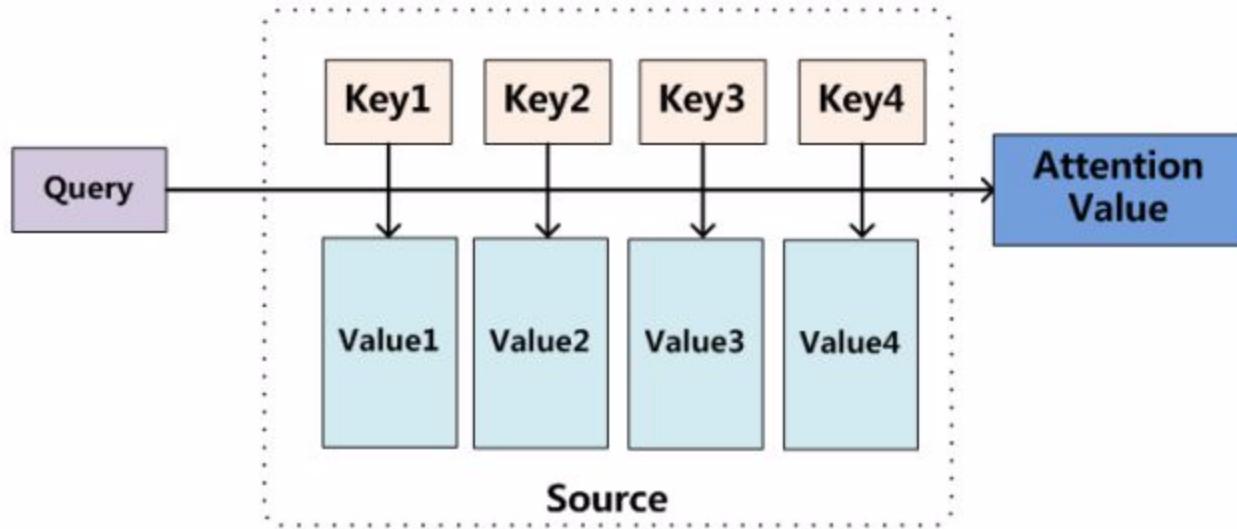


- “eating”和“apple”有很强的联系，那我们就希望在处理吃这个单词的时候，能够在语义中，包含一定的苹果的信息，这样能够帮助我们更好的理解“eating”这个动作。
- “green”和“apple”也是一样的；
- Attention的机制能够帮助我们在处理单个的 token 的时候，带有一定的上下文信息。

# 推荐系统

- query 当前要判断的商品的向量，key 就是用户购买序列中，每一个商品的向量。
- query 和 key 进行相似度计算，得到待判断商品和购买序列中商品的相关性分值；
- 将这个分值进行归一化(softmax)，得到一个商品注意力的分布，看看哪些商品是判断的重要依据；
- 使用注意力分布和 value 进行计算，得到一个融合注意力的更好的 value 值，这个值就是最终我们融合判断当前商品是否推荐购买的依据。

# Attention 工作机制



- 将Source中的构成元素想象成是由一系列的<Key,Value>数据对构成，给定Target中的某个元素Query，通过计算Query和各个Key的相似性或者相关性，得到每个Key对应Value的权重系数，然后对Value进行加权求和，即得到了最终的Attention数值。

- 本质上Attention机制是对Source中元素的Value值进行加权求和，而Query和Key用来计算对应Value的权重系数。即可以将其本质思想改写为如下公式：

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} \text{Similarity}(\text{Query}, \text{Key}_i) * \text{Value}_i$$

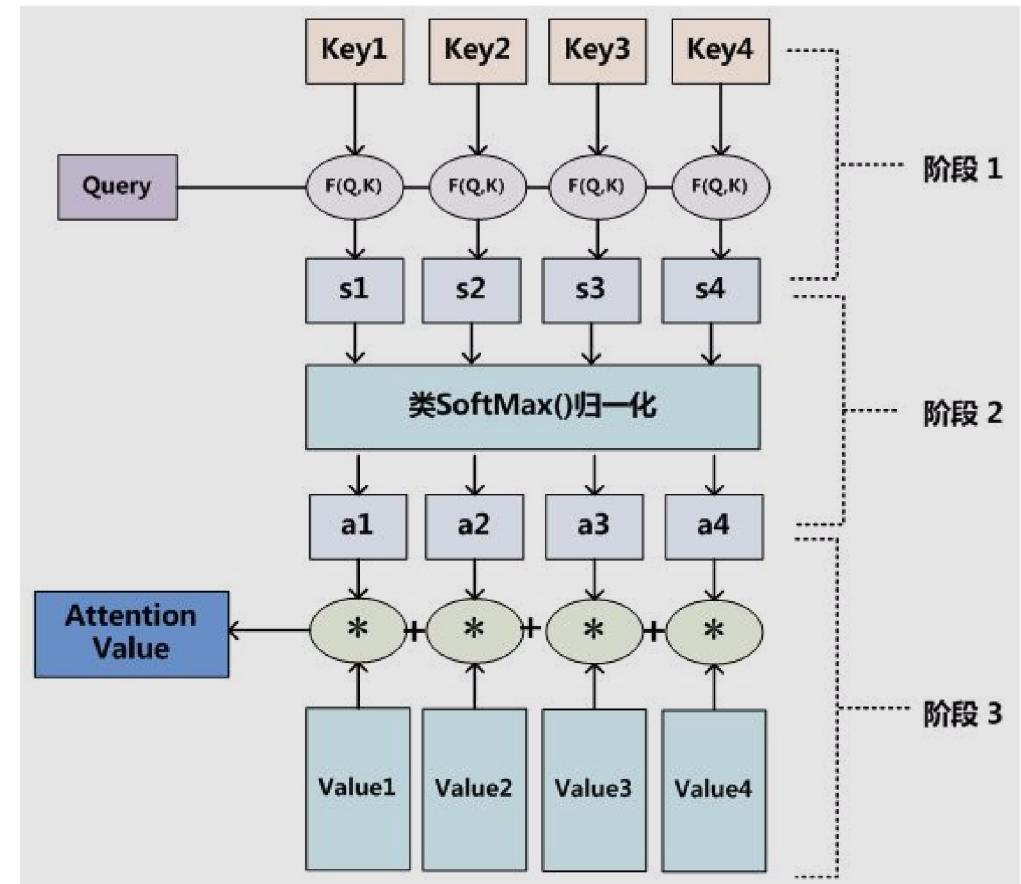
其中， $L_x = ||\text{Source}||$ 代表Source的长度。

- 机器翻译中，因为在计算Attention的过程中，Source中的Key和Value合二为一，指向的是同一个东西，也即输入句子中每个单词对应的语义编码，所以可能不容易看出这种能够体现本质思想的结构。
- 从概念上理解，把Attention仍然理解为从大量信息中有选择地筛选出少量重要信息并聚焦到这些重要信息上，忽略大多不重要的信息。聚焦的过程体现在权重系数的计算上，权重越大越聚焦于其对应的Value值上，即权重代表了信息的重要性，而Value是其对应的信息。

# 计算过程

一个 Attention 的计算过程有三步：

- query 和 key 进行相似度计算，得到一个 query 和 key 相关性的分值；
- 将这个分值进行归一化 (softmax)，得到一个注意力的分布；
- 使用注意力分布和 value 进行计算，得到一个融合注意力的更好的 value 值。



## 相似度

可以引入不同的函数和计算机制，根据Query和某个Key<sub>i</sub>，计算两者的相似性或者相关性，最常见的方法包括：求两者的向量点积、求两者的向量Cosine相似性或者通过再引入额外的神经网络来求值，即如下方式：

- 点积：

$$\text{Similarity}(\text{Query}, \text{Key}_i) = \text{Query} \cdot \text{Key}_i$$

- Cosine 相似性

$$\text{Similarity}(\text{Query}, \text{Key}_i) = \frac{\text{Query} \cdot \text{Key}_i}{\|\text{Query}\| \cdot \|\text{Key}_i\|}$$

- MLP网络：

$$\text{Similarity}(\text{Query}, \text{Key}_i) = \text{MLP}(\text{Query}, \text{Key}_i)$$

## 归一化

引入类似SoftMax的计算方式对第一阶段的得分进行数值转换，一方面可以进行归一化，将原始计算分值整理成所有元素权重之和为1的概率分布；另一方面也可以通过SoftMax的内在机制更加突出重要元素的权重。

## 计算结果

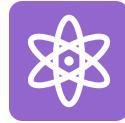
$a_i$ 即为 $value_i$ 对应的权重系数，然后进行加权求和即可得到Attention数值：

$$Attention(Query, Source) = \sum_{i=1}^{L_x} a_i \cdot Value_i$$

即可求出针对 $Query$ 的 $Attention$ 数值，目前绝大多数具体的注意力机制计算方法都符合上述的三阶段抽象计算过程。

## 说明

Attention顾名思义，指的不是Target和Source之间的Attention机制，而是Source内部元素之间或者Target内部元素之间发生的Attention机制，也可以理解为Target=Source这种特殊情况下的注意力计算机制。其具体计算过程是一样的，只是计算对象发生了变化而已。



# Self-attention

self-attention 最好的文章《Attention is all you need》，在这篇文章中，谷歌提出了 Transformer 这个模型，里面最重要的模块就是 self-attention。

不过这篇文章其实并不是最早提出类似架构的论文，但是它将 self-attention 发扬光大了。

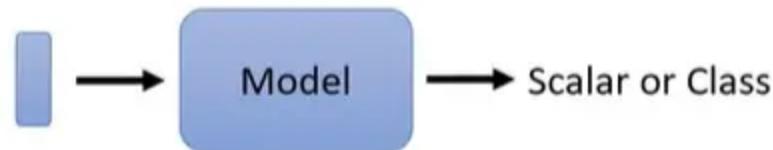
- NLP 中的句子是一条很长的序列，CV 中的图像是很大的像素矩阵。复杂输入意味着输入样本同时包含了多个元素的信息，而这些信息之间并不是独立的，而是存在程度不一的联系，互为上下文信息。
- 比如对于机器翻译而言，同一个词在不同语境下还存在一词多义的情况。
- Attention 机制的出现就是为了解决这个问题，在网络中紧邻输入层之后添加了一个 attention 层，让输入向量中的各元素都依次从彼此之间抽取有用信息后，再输入后续的网络中计算。

- 从计算细节上看，注意力模型能够实现并行运算，相比于 RNN 模型，大大加快了对序列的处理速度。具体来说，attention 层一共增加了 3 个待训练的参数矩阵： $W^q, W^k, W^v$ ，分别表示 query 的权重矩阵，key 的权重矩阵 和 vector 的权重矩阵，它们分别乘以输入矩阵  $I$  后得到了  $Q, K, V$  三个矩阵。
- 以 NLP 为例，这三个矩阵的引入，使得每一个单词都有了 query 向量，key 向量和 vector 向量，用于后续词之间关联度计算。而词与词之间的关联程度，就是通过当前单词的 query 向量和其他单词的 key 向量的点积求得，再经过激活函数归一化。和某单词点积越大的词向量意味着和该词的关联度越高，因而在提取信息的加权求和中占比就越大。最终的注意力权重计算公式为  $A' = \text{softmax}(k \cdot Q)$ ，而抽取后的输出需要再乘以 vector 矩阵，为  $O = VA'$ 。

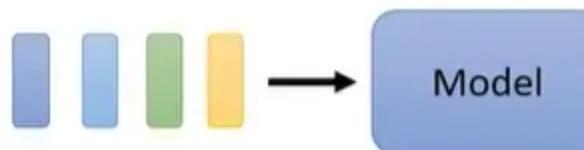
## Self-attention Input:

对于一些机器学习任务，可能面临复杂的 input，它们不止有一个向量，而且具体的向量的个数会随样本不同而改变，复杂的输入（不定个数的向量）。

- Input is a vector



- Input is a set of vectors



知乎 @橘Oran

图源：李宏毅老师的课件

举例1：文字处理中，每一个词都是一个向量，且句子长度不一样。

如何表示每个词呢？

最简单的方式就是 One-hot Encoding，但是这样表示是假设所有的词之间都没有关系，向量间没有相关信息。

另一种方式是 Word Embedding，这种方式可以降维，并且含有语义信息，使得类似得词聚成一团。

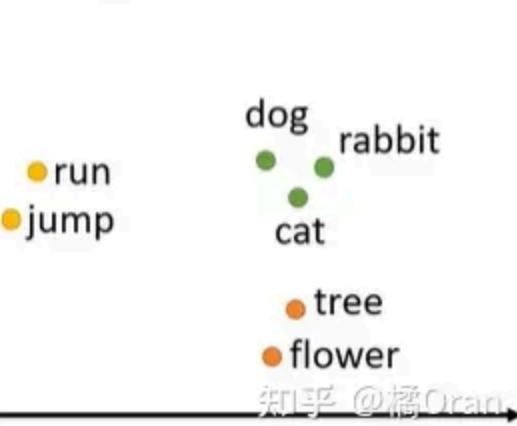
Vector Set as Input

this    is    a    cat  
      |    |    |  
      blue blue green yellow

One-hot Encoding

apple = [ 1 0 0 0 0 ..... ]  
bag = [ 0 1 0 0 0 ..... ]  
cat = [ 0 0 1 0 0 ..... ]  
dog = [ 0 0 0 1 0 ..... ]  
elephant = [ 0 0 0 0 1 ..... ]

Word Embedding



文字处理任务中单词的表示，OHE 或 word embedding

## 举例二：声音序列，也是如此，长度不一



知乎 @橘Oran

在语音序列上，我们会把一段窗口内的语音片段（frame）变成一个向量，1秒中的语音大概有 100 个 frame。

语音也是向量

- 图源：李宏毅老师的课件

### 举例3：图，也是一堆向量（结点和向量）

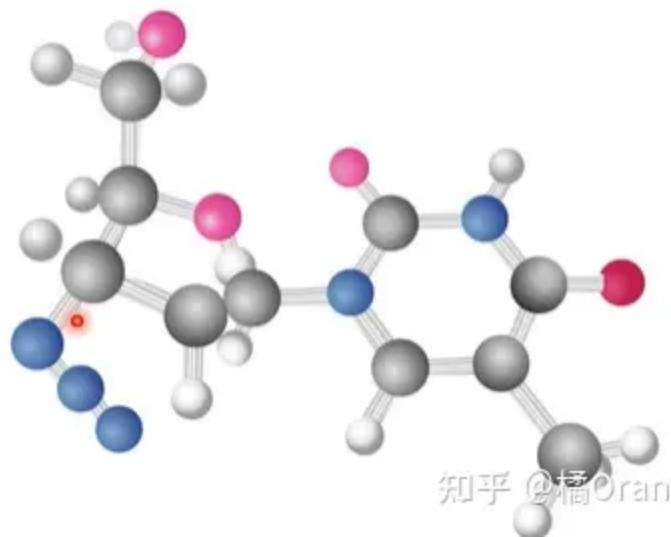
- Graph is also a set of vectors (consider each **node** as a **vector**)



图也是向量 图源：李宏毅老师的课件

## 举例4：分子，其结构是图，所以也是一堆向量

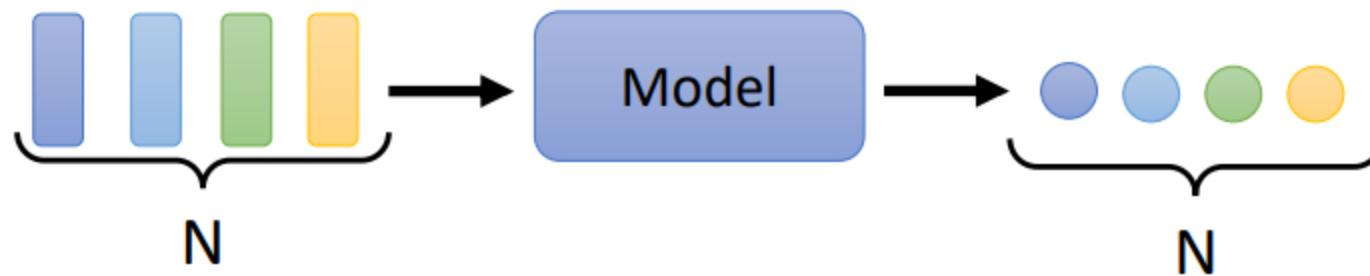
- Graph is also a set of vectors (consider each **node** as a **vector**)



分子的结构也是向量 图源：李宏毅老师的课件

## Output

第一种形式：  $m$  to  $m$ , 输入和输出的数目一样，每一个输入向量都有一个 label



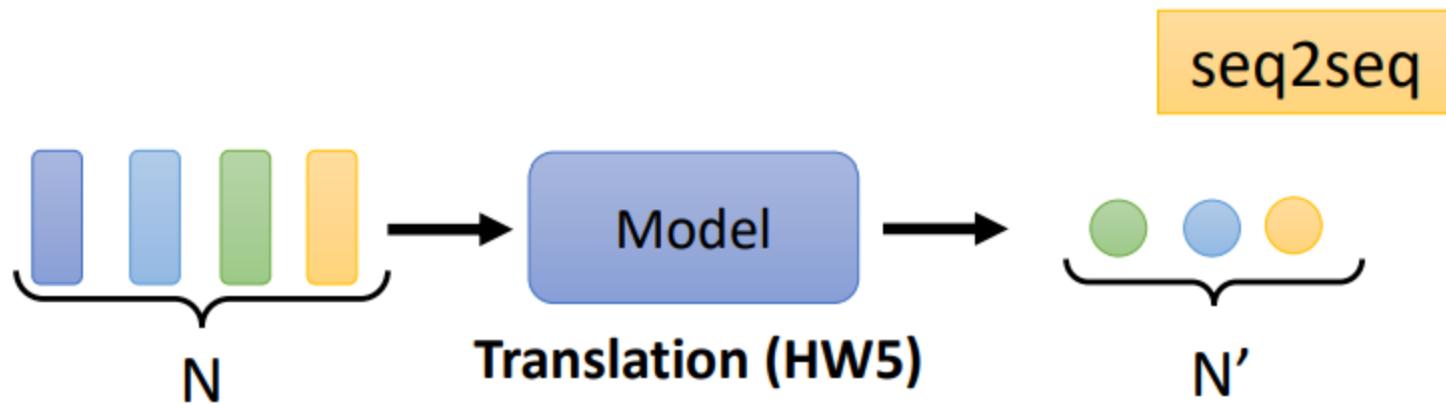
$m$  to  $m$  任务的输出，每个输入都对应一个输出.

第二种形式：  $m$  to 1， 多输入， 单输出



$m$  to 1 多输入单输出的情况

第三种形式： seq to seq， 输出的长度不一定， 比如机器翻译



seq to seq， 多个输入对应不定个数的输出

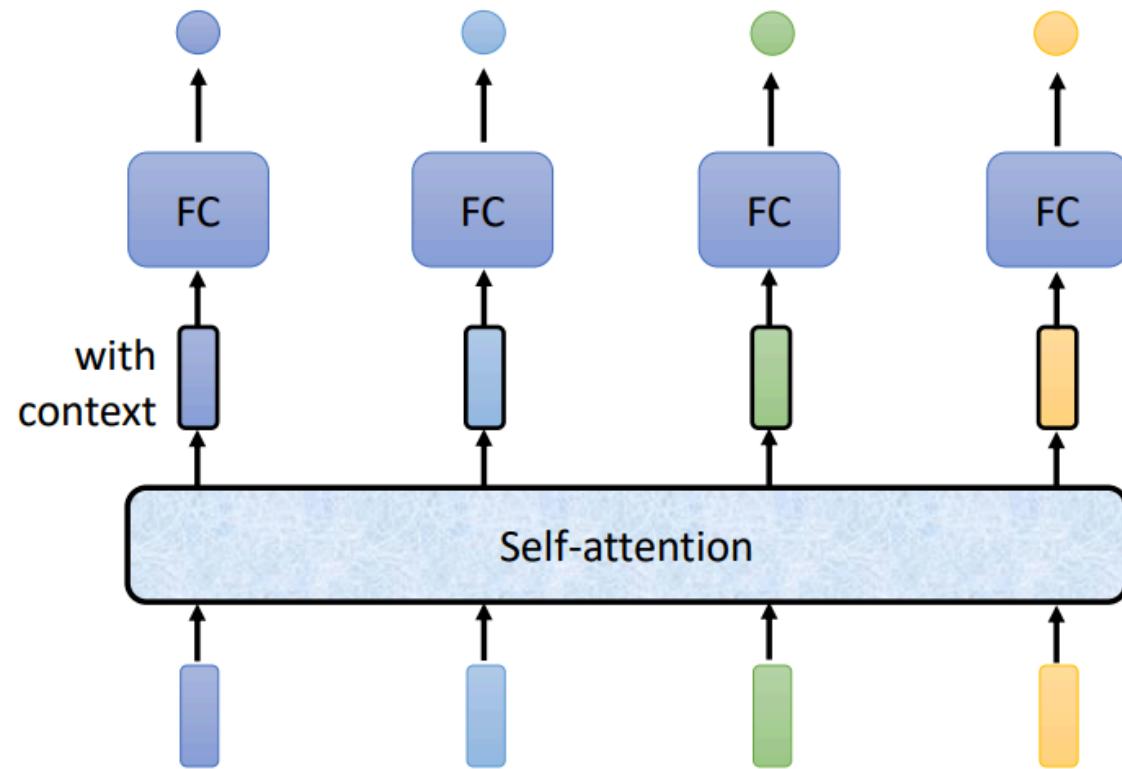
## 多输入多输出任务中的 Attention

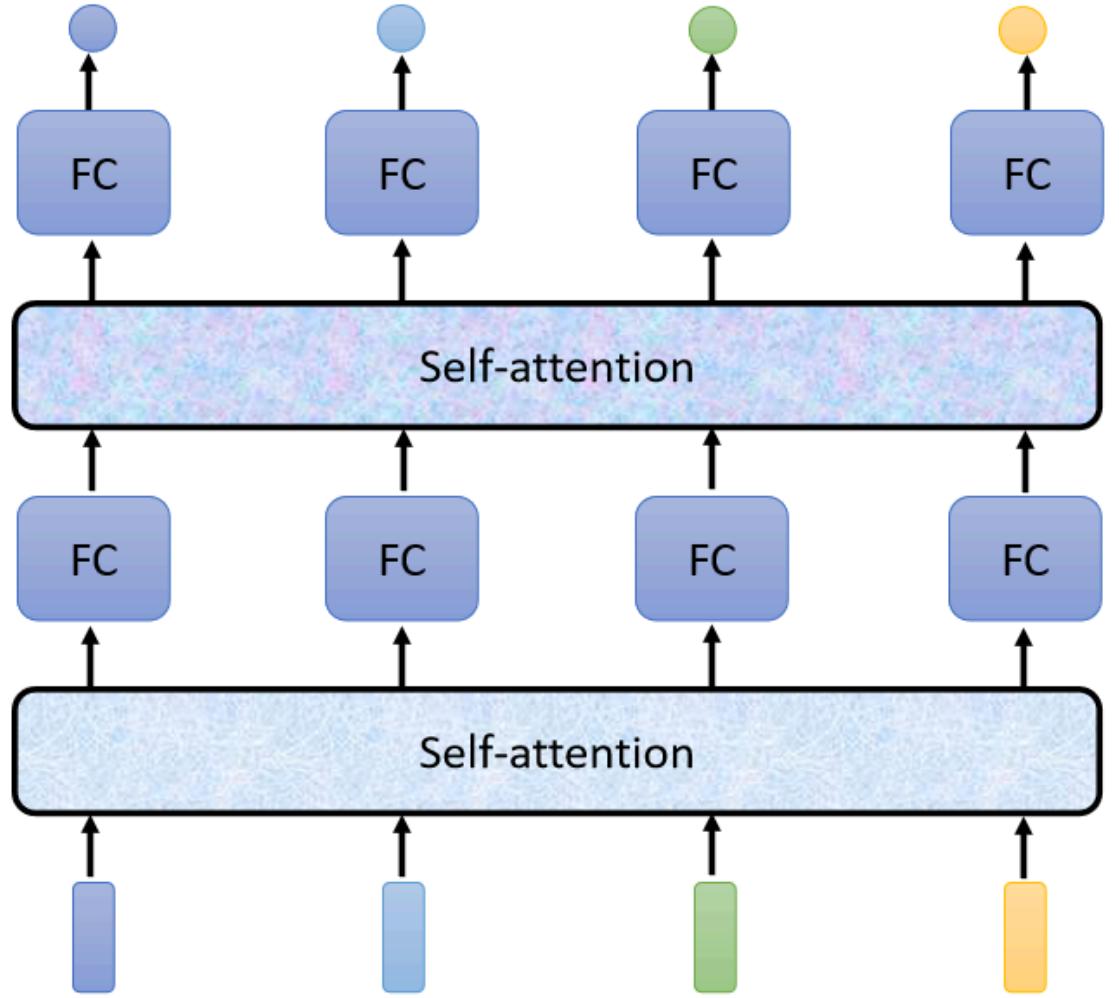
问题：在词性标注任务中，对于 Fully-connected 全连接网络来说，输入完全一样，输出也是一样，无法考虑不同位置的词性。

解决方法1：考虑一个固定长度的 window，在判断词性的时候，不仅只看当前词，还看固定 window 内的其他词，也就是这个词的上下文背景信息。

但这种直观处理方式的缺点是句子的长度是不定的，有的可能很长，window 设置得太小可能没有效果，若特别大则又容易过拟合。

因为要建立输入向量序列的长依赖关系，所以模型要考虑整个向量序列的信息。如下图所示，Self-Attention的输出序列长度是和输入序列的长度一样的，对应的输出向量考虑了整个输入序列的信息。然后将输出向量输入到Fully-Connected网络中，做后续处理。





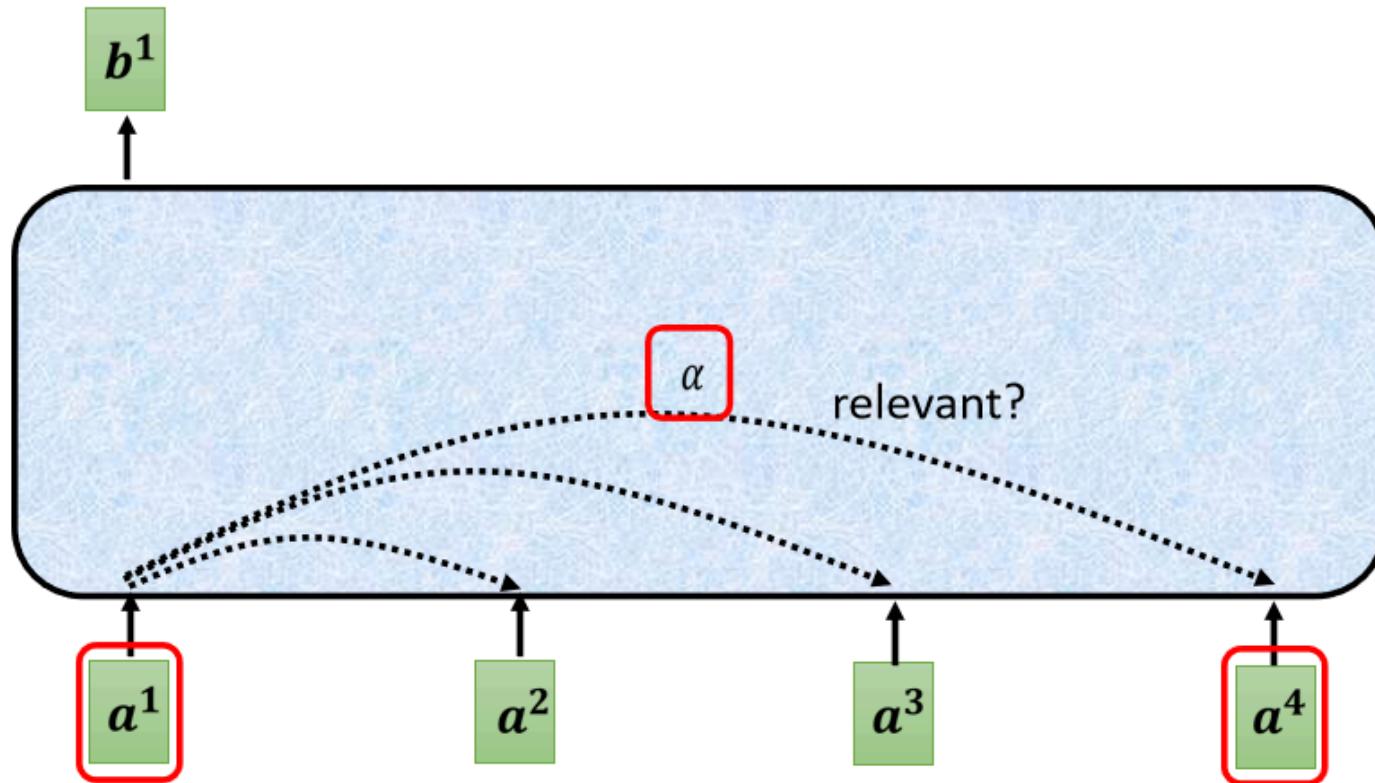
<https://arxiv.org/abs/1706.03762>

self-attention 最好的文章《Attention is all you need》，在这篇文章中，谷歌提出了 Transformer 这个模型，里面最重要的模块就是 self-attention。

不过这篇文章其实并不是最早提出类似架构的论文，但是它将 self-attention 发扬光大了。

# 计算步骤

1. 以  $a^1$  为例，根据  $a^1$  这个向量，找出整个 sequence 中跟  $a^1$  相关的其他向量  $\Rightarrow$  计算哪些部分是重要的，求出  $a^i$  和  $a^1$  的相关性（影响程度大的就多考虑点资讯），用  $\alpha$  表示



Find the relevant vectors in a sequence

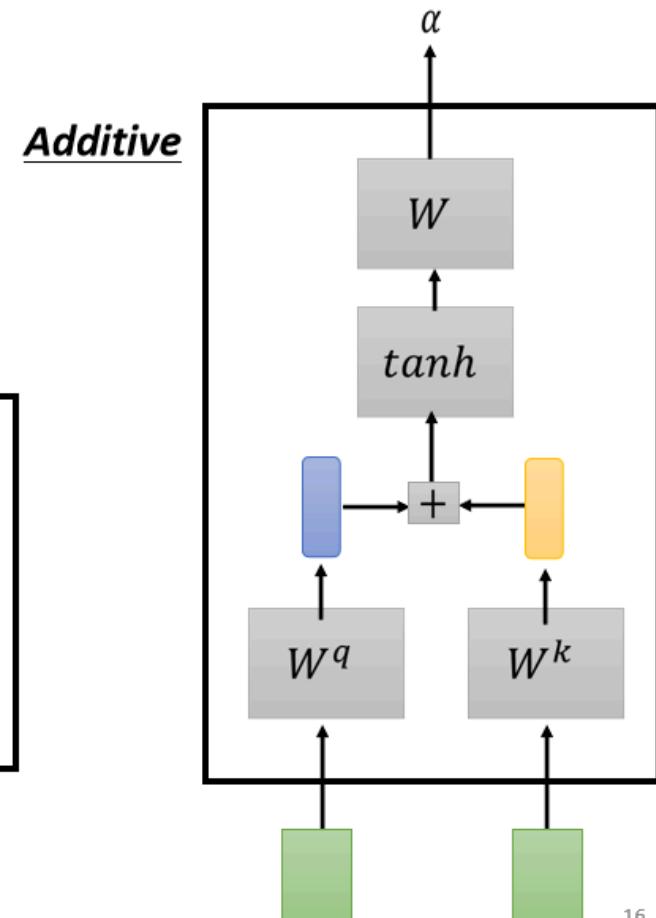
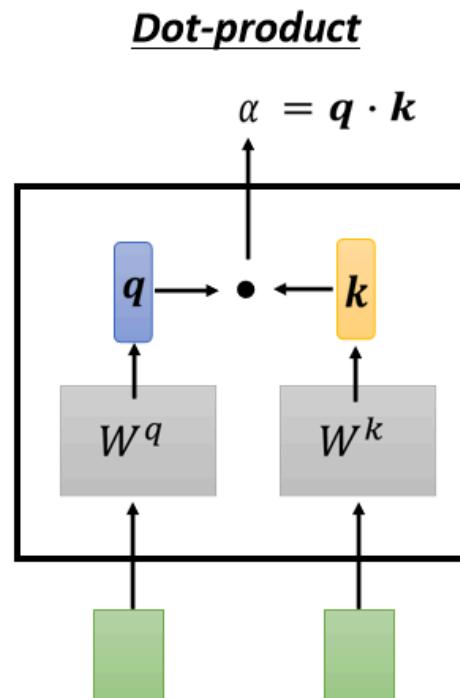
## self-attention 的结构

如何产生  $b_1$  这个向量？需要衡量输入向量之间的关联度  $a$ 。

- 计算 attention 的模组算出 $a$ （关联度大小），有很多方法，比如 Dot-product 方法和 Additive 方法。
- Attention is all you need 论文中 Transformer 用的方法使用 Dot-product 方法，也是最常用的方法。

- 左边这个向量乘上矩阵  $W^q$  得到矩阵  $q$  (query) , 右边这个向量乘上矩阵  $W^k$  得到矩阵  $k$  (key) , 再把  $q$  跟  $k$  做 dot product (点积) , 逐元素相乘后累加得到一个 scalar 就是 相关性  $\alpha$

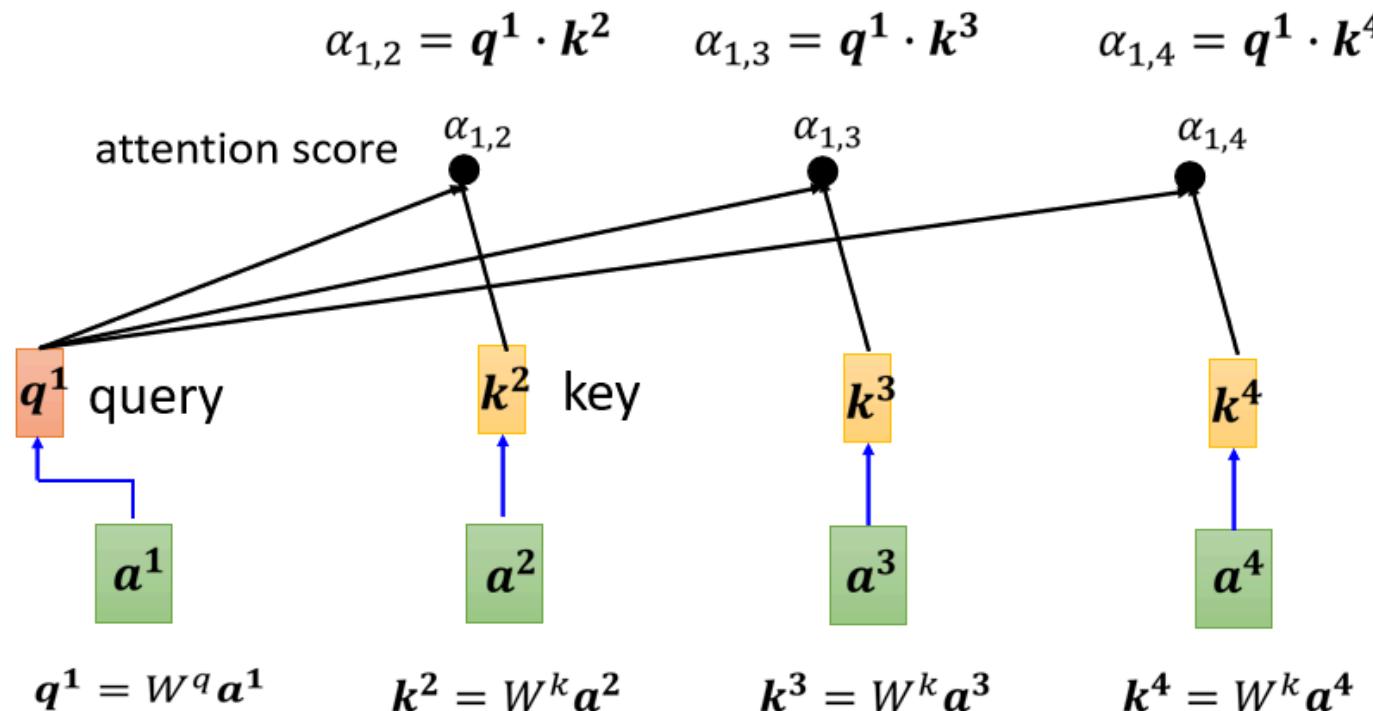
## Self-attention



## 注意力权重 $\alpha$ 的计算

为了提高模型能力，自注意力模型经常采用查询-键-值（Query-Key-Value,  $QKV$ ）模型，其计算过程如下所示：

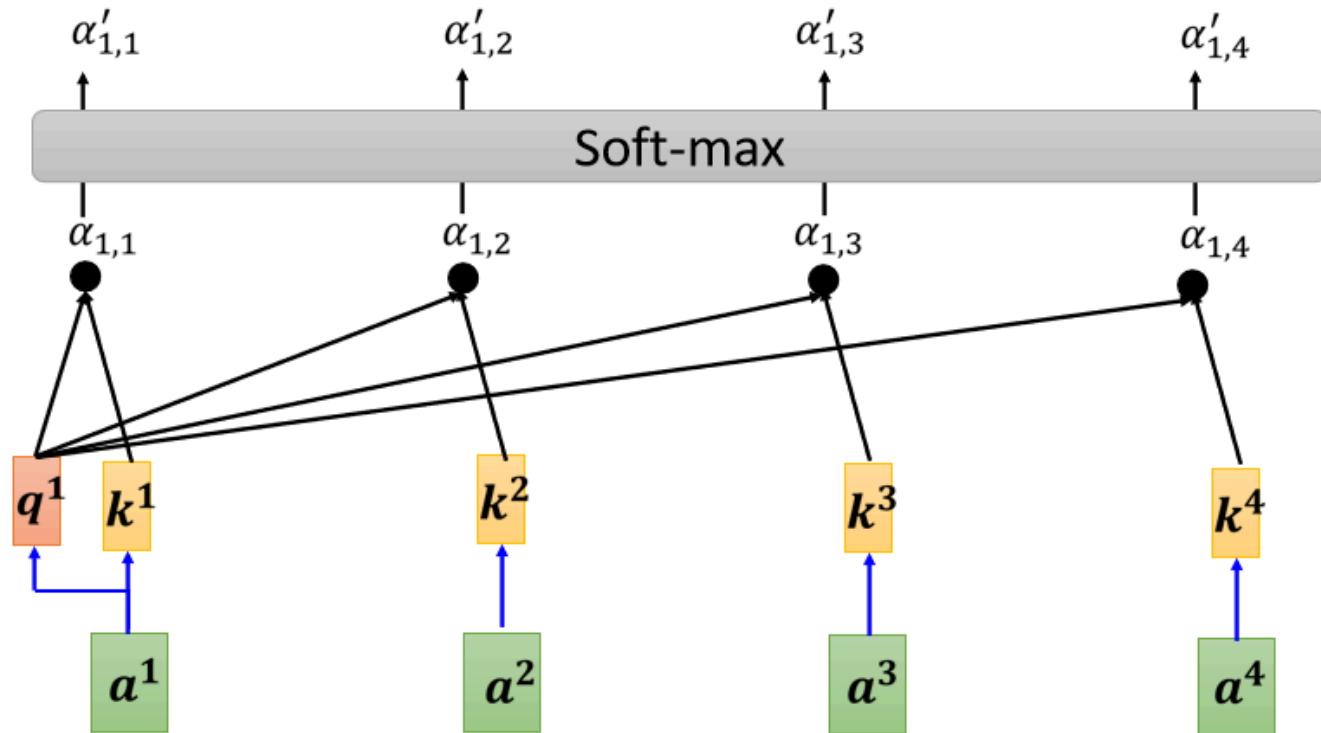
- 计算  $a^1$  与  $[a^2, a^3, a^4]$  的关联性  $\alpha$ （实践时一般也会计算与  $a^1$  自己的相关性）。以 Dot-product 为例，我们分别将  $[a^1, a^2], [a^1, a^3], [a^1, a^4]$  作为 Dot-product 的输入，求得对应的相关性  $[\alpha_{1,2}, \alpha_{1,3}, \alpha_{1,4}]$ ，如下图所示：



- 计算出 $a^1$ 跟每一个向量的关联性之后，将得到的关联性输入softmax中，这个 softmax 和分类使用的softmax时一样的，得到对应数量的 $\alpha'$ 。
- 计算激活后的注意力权重 attention score 得出所有的  $\alpha_{i,j}$  之后，这些数需要通过激活函数比如 Soft-max 函数，输出激活后的  $\alpha'$ ，也就是我们需要的注意力权重。这里的激活函数使用的是 Soft-max，目的是做 Normalization。
- 至于为什么使用 soft-max 并没有定论，也可以使用别的激活函数，比如 ReLU。

## Self-attention

$$\alpha'_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$$



$$q^1 = W^q a^1$$

$$k^2 = W^k a^2$$

$$k^3 = W^k a^3$$

$$k^4 = W^k a^4$$

$$k^1 = W^k a^1$$

- 最后根据  $\alpha'$ ，也就是激活后的 attention score，来抽取信息。

这里有引入另一个组件， $v^i$ ，它的计算方式也和 query 或 key 的计算很相似，它也有属于自己的权重矩阵  $W^v$ ，所以计算方式为： $v^i = W^v \cdot a^i$

- 最终的  $b_1$  计算过程：

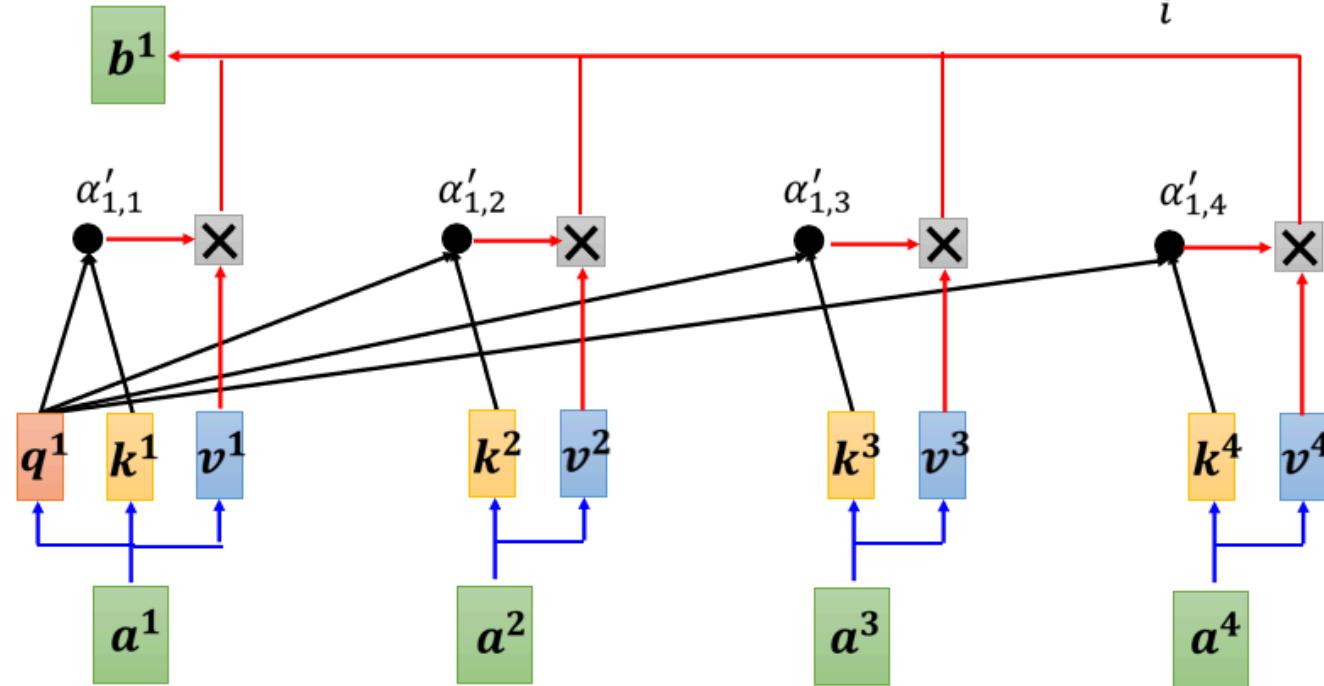
$b^1$  是加全求和，每也就是个注意力权重  $\alpha'_{1,j}$  乘以对应的  $v^i$ ，计算公式为：

$$b^1 = \sum_i \alpha'_{1,i} v^i = \alpha'_{1,1} v^1 + \alpha'_{1,2} v^2 + \dots + \alpha'_{1,4} v^4$$

## Self-attention

Extract information based  
on attention scores

$$b^1 = \sum_i \alpha'_{1,i} v^i$$



$$v^1 = W^v a^1$$

$$v^2 = W^v a^2$$

$$v^3 = W^v a^3$$

$$v^4 = W^v a^4$$

谁和当前词的关系越大，其信息就越会被抽出来。每一个输入向量都有其对应的  $q, k, v$  向量。

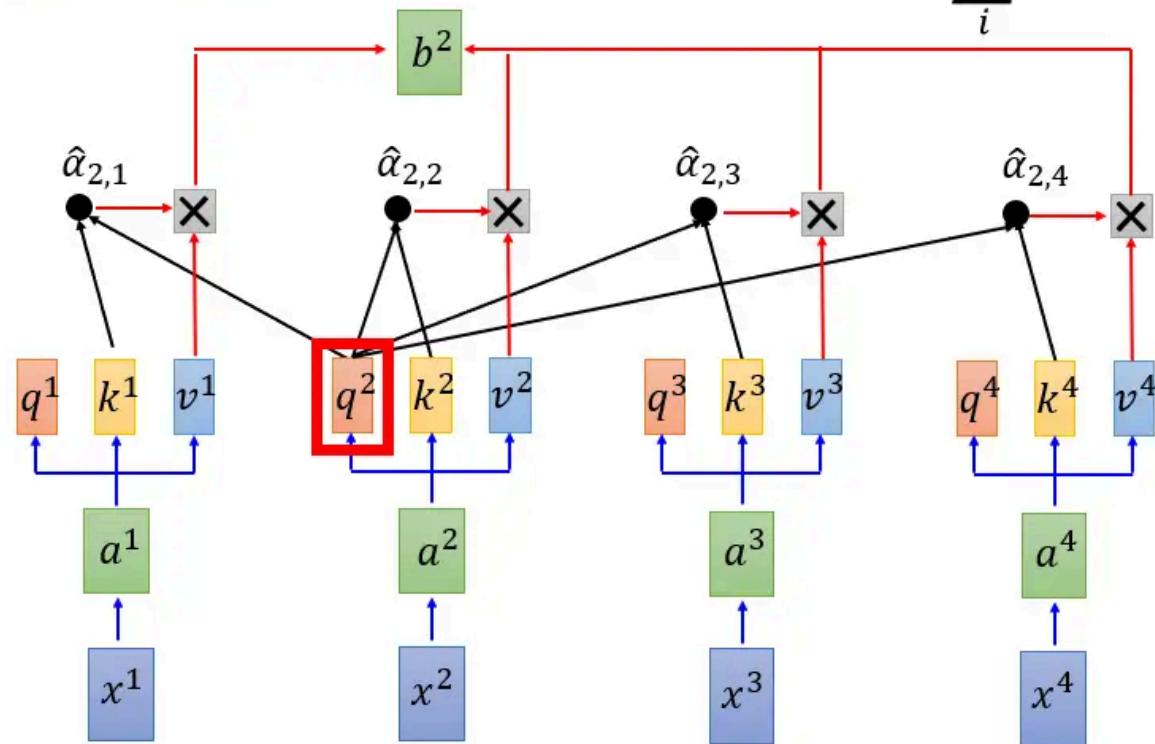
# 并行计算

这个过程可以并行计算，以  $b_2$  为例，重复一次计算过程：

## Self-attention

拿每個 query  $q$  去對每個 key  $k$  做 attention

$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$



- 以  $b^2$  为例，重复一次计算过程

1 类似于全连接网络中的矩阵计算，将每个  $q, k, v$  向量堆叠在一起，成为矩阵。

$$q^i = W^q a^i \quad q^1 | q^2 | q^3 | q^4 = \begin{matrix} W^q \\ Q \end{matrix} \quad a^1 | a^2 | a^3 | a^4$$

$$k^i = W^k a^i \quad k^1 | k^2 | k^3 | k^4 = \begin{matrix} W^k \\ K \end{matrix} \quad a^1 | a^2 | a^3 | a^4$$

$$v^i = W^v a^i \quad v^1 | v^2 | v^3 | v^4 = \begin{matrix} W^v \\ V \end{matrix} \quad a^1 | a^2 | a^3 | a^4$$

并行处理， $Q, K, V, W, I$  都是矩阵

整体来看，并行矩阵运算的过程：

## ***Self-attention***

$$q^1 \ q^2 \ q^3 \ q^4 = \begin{matrix} W^q \\ Q \end{matrix} \quad a^1 \ a^2 \ a^3 \ a^4 = \begin{matrix} W^q \\ I \end{matrix}$$

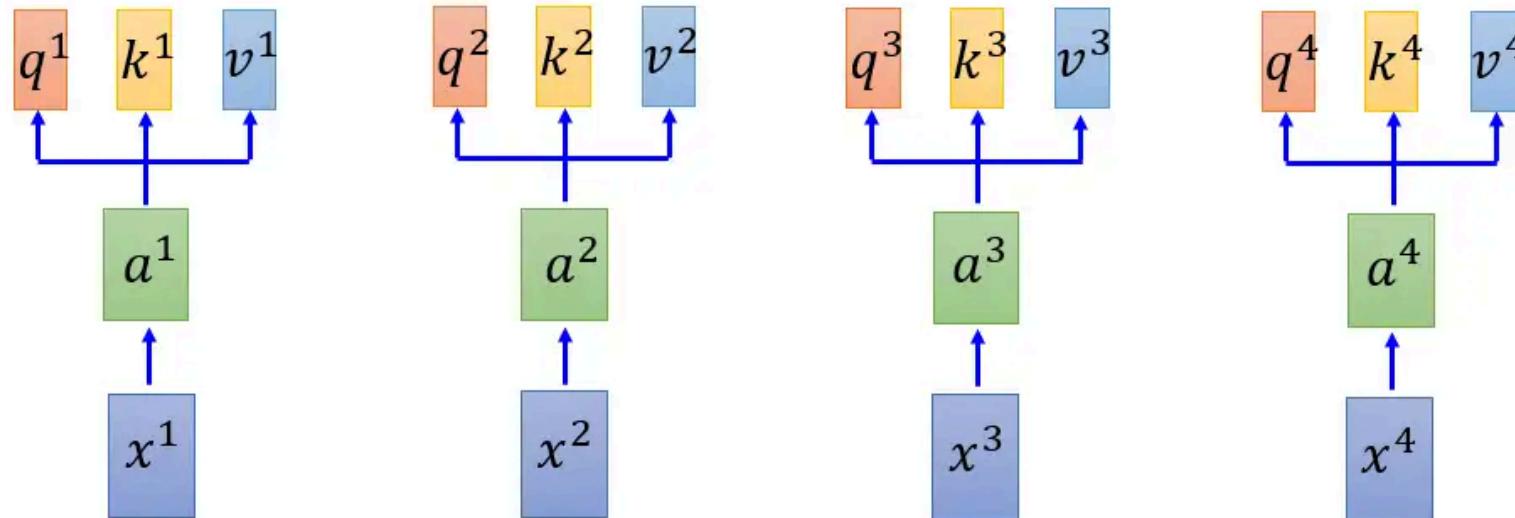
$$q^i = W^q a^i$$

$$k^1 \ k^2 \ k^3 \ k^4 = \begin{matrix} W^k \\ K \end{matrix} \quad a^1 \ a^2 \ a^3 \ a^4 = \begin{matrix} W^k \\ I \end{matrix}$$

$$k^i = W^k a^i$$

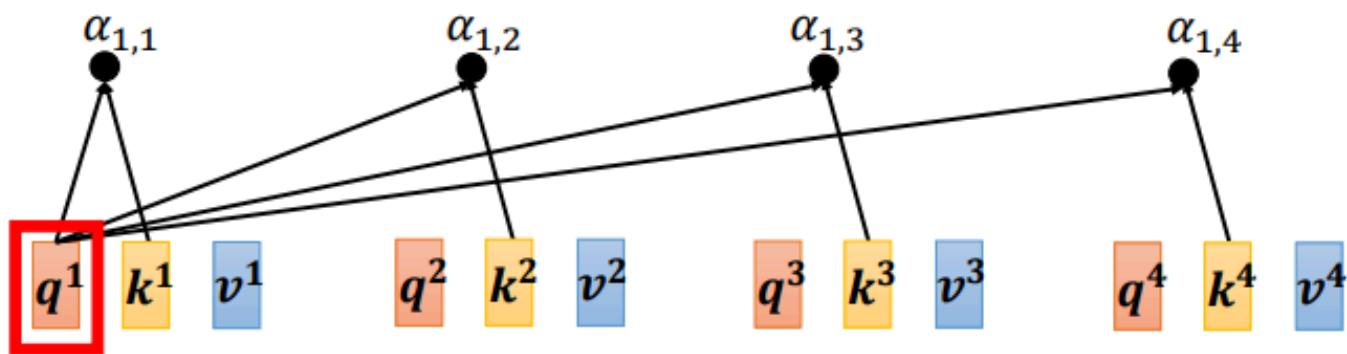
$$v^1 \ v^2 \ v^3 \ v^4 = \begin{matrix} W^v \\ V \end{matrix} \quad a^1 \ a^2 \ a^3 \ a^4 = \begin{matrix} W^v \\ I \end{matrix}$$

$$v^i = W^v a^i$$



2 得到  $Q$ 、 $K$  和  $V$  三个矩阵之后，下一步就是由矩阵  $Q$  和  $K$  计算关联性  $\alpha$ ，先以计算  $q^1$  的关联性为例，将矩阵  $q^1$  分别与矩阵  $[k^1, k^2, k^3, k^4]$  相乘得到  $[\alpha_{1,1}, \alpha_{1,2}, \alpha_{1,3}, \alpha_{1,4}]$ （我们现在是看作矩阵运算，为了方便计算，我们先对矩阵  $k^i$  进行转置，在和矩阵  $q^1$  相乘）：

$$\begin{aligned}\alpha_{1,1} &= \begin{matrix} k^1 \\ q^1 \end{matrix} \quad \alpha_{1,2} = \begin{matrix} k^2 \\ q^1 \end{matrix} \\ \alpha_{1,3} &= \begin{matrix} k^3 \\ q^1 \end{matrix} \quad \alpha_{1,4} = \begin{matrix} k^4 \\ q^1 \end{matrix}\end{aligned}\quad \begin{matrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{matrix} = \begin{matrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{matrix} \begin{matrix} q^1 \end{matrix}$$



3 将第一步中得到的矩阵 $K$ 整体转置，与矩阵 $Q$ 相乘，得到所有的相关性矩阵 $A$ ；然后对attention分数（相关性）做normalization，即每次对 $A$ 中的一列（每列对应着一个 $q^i$ ）做softmax(也可以是其他激活函数)，让每一列的值相加为1，得到矩阵 $A'$ ，如下图所示：

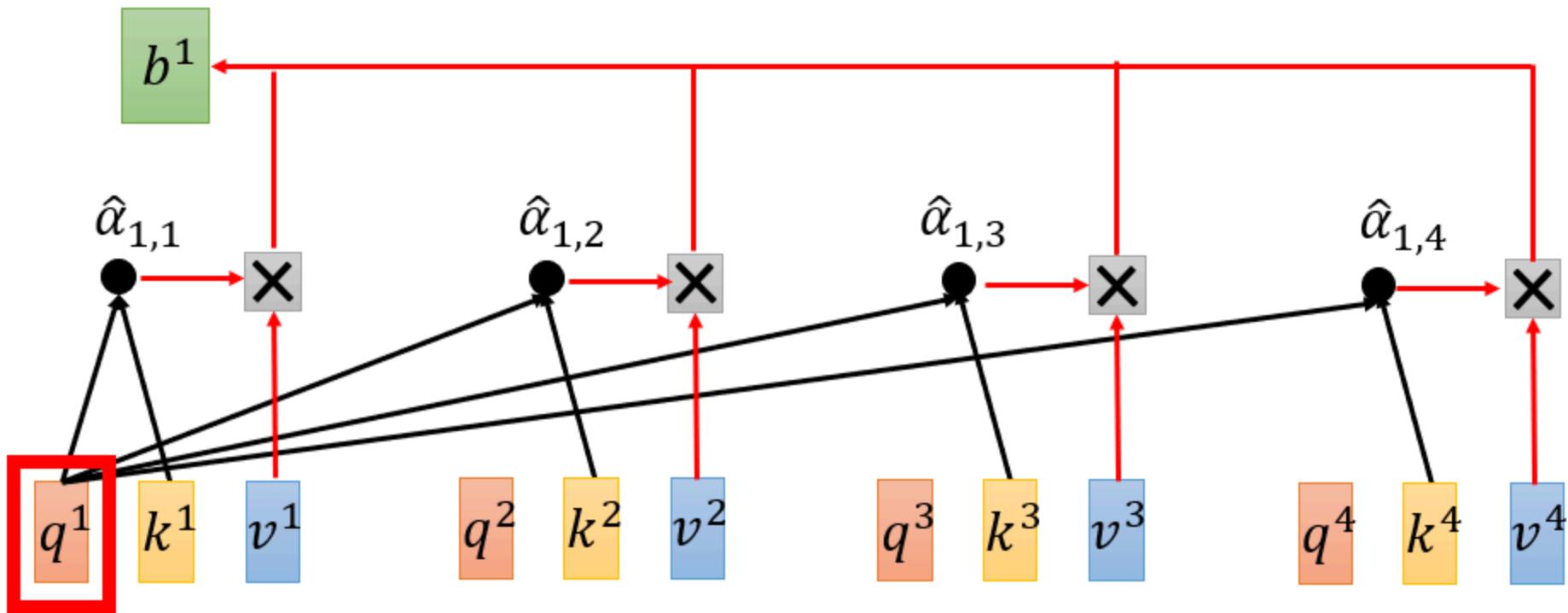
$$\begin{array}{cccc}
 \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\
 \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\
 \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\
 \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4}
 \end{array}
 \xleftarrow{\text{softmax}}
 \begin{array}{cccc}
 \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1} \\
 \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2} \\
 \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3} \\
 \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4}
 \end{array}
 = \begin{matrix}
 k^1 \\
 k^2 \\
 k^3 \\
 k^4
 \end{matrix} \quad Q$$

$A'$       softmax       $A$        $K^T$

4 我们将矩阵 $V$ 依次乘以矩阵 $A'$ 中的每一列得到输出 $[b^1, b^2, b^3, b^4]$ , 如下图所示:

$$\begin{matrix} b^1 & b^2 & b^3 & b^4 \\ 0 \end{matrix} = \begin{matrix} v^1 & v^2 & v^3 & v^4 \\ V \end{matrix} \begin{matrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{matrix} A'$$

# 综上所述



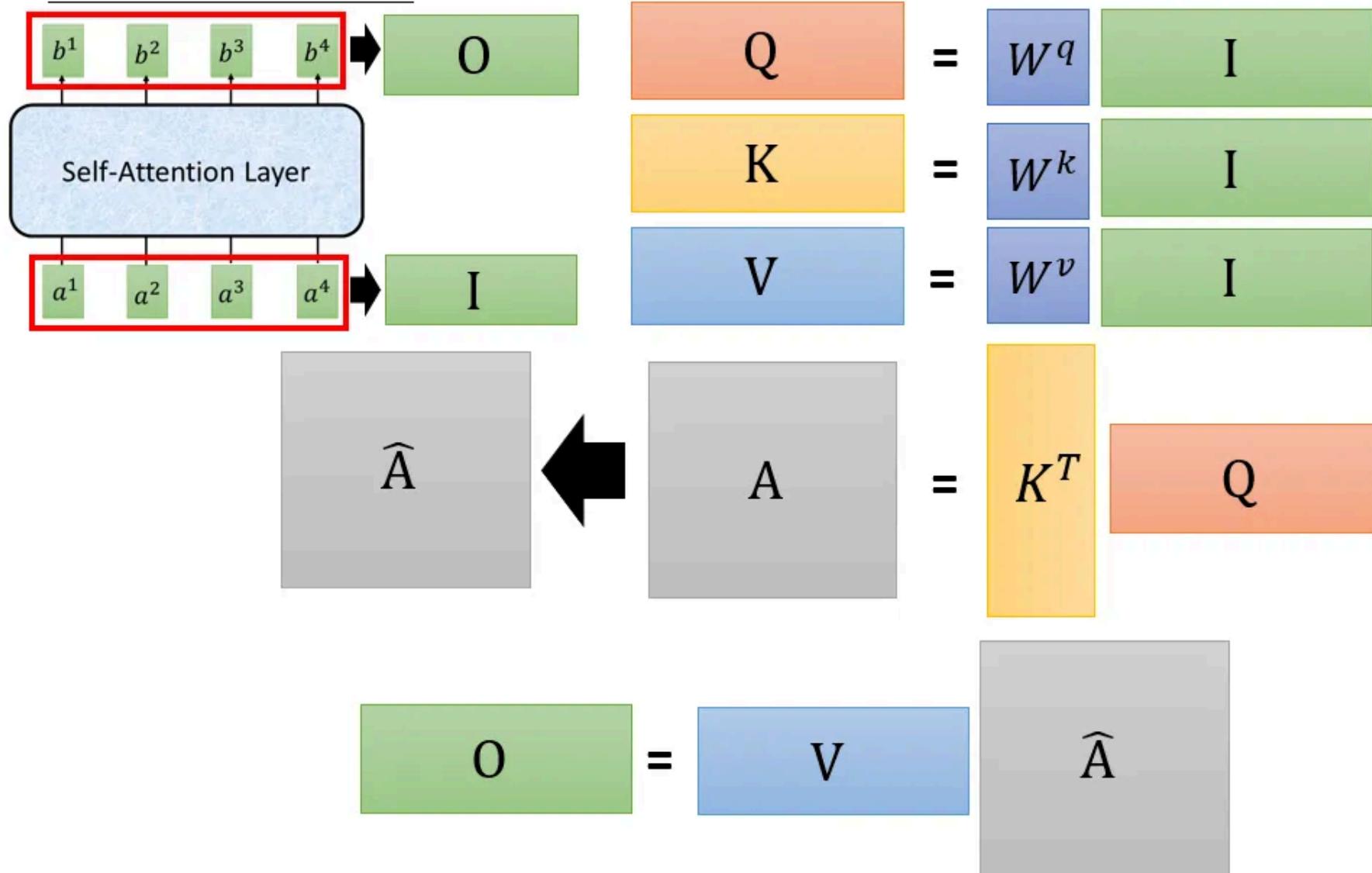
$$\alpha_{1,1} = [k^1 \quad q^1] \quad \alpha_{1,2} = [k^2 \quad q^1]$$

$$\alpha_{1,3} = [k^3 \quad q^1] \quad \alpha_{1,4} = [k^4 \quad q^1]$$

(ignore  $\sqrt{d}$  for simplicity)

$$\begin{matrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{matrix} = \begin{matrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{matrix} \quad \begin{matrix} q^1 \\ q^1 \\ q^1 \\ q^1 \end{matrix}$$

## *Self-attention*



反正就是一堆矩阵乘法，用 GPU 可以加速

综上所述，矩阵计算公式如下：

计算Q、K和V三个矩阵

$$Q = W^q I$$

$$K = W^k I$$

$$V = W^v I$$

计算相关性并使用softmax作normalization：

$$A = K^T Q$$

$$A' = softmax(A)$$

计算最后的输出  $O = V A'$

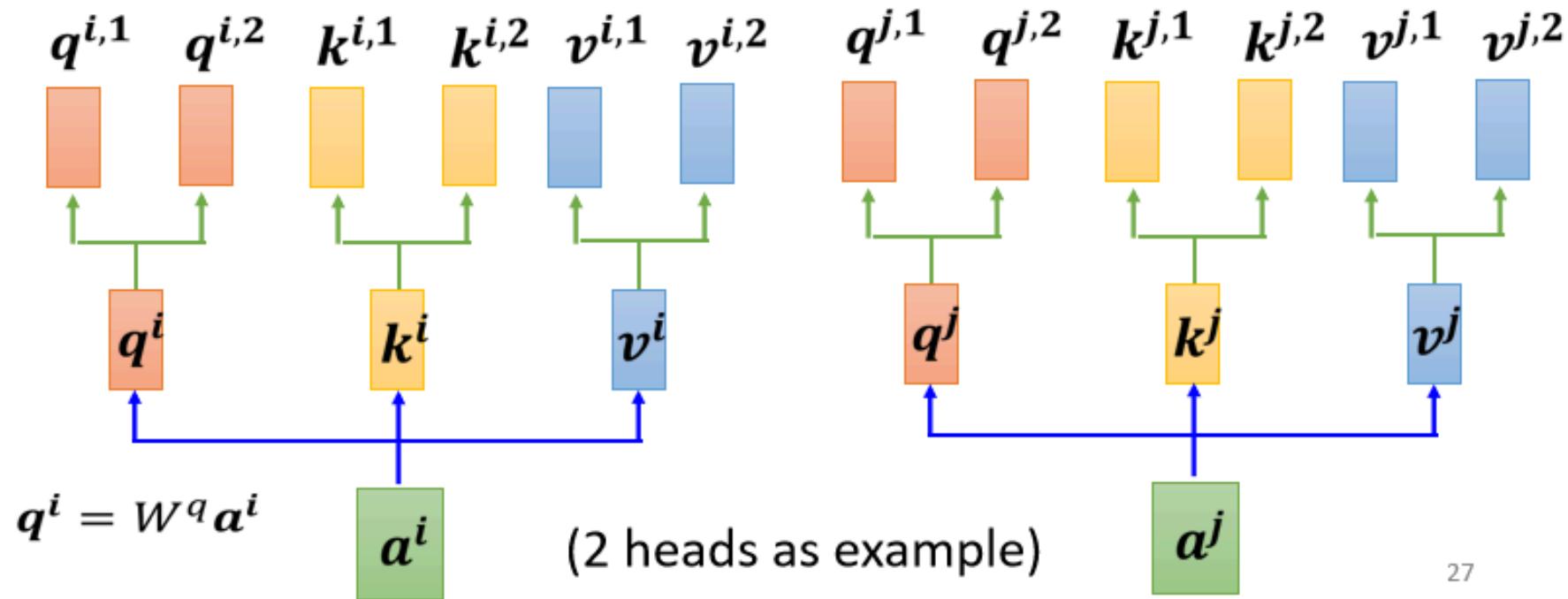
通过上述计算过程，我们可以看出只有  $W^q, W^k, W^v$  三个矩阵是未知的，需要我们通过训练资料将他们学习出来。

## Multi-head Self-Attention

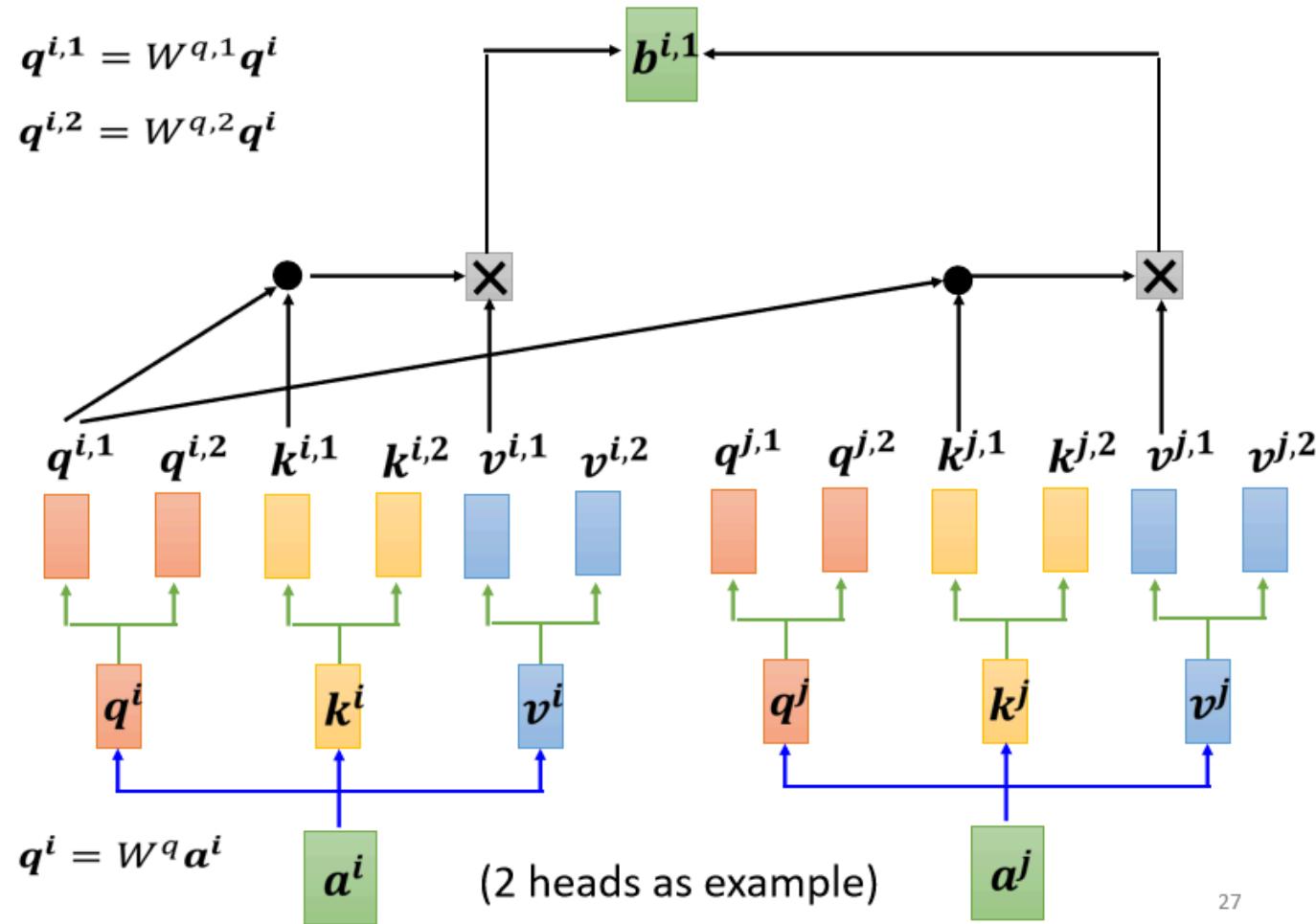
Self-Attention有一个使用非常广泛的进阶版Multi-head Self-Attention，具体使用多少个head，是一个需要我们自己调节的超参数（hyper parameter）。

在Self-Attention中，我们是使用 $q$ 去寻找与之相关的 $k$ ，但是这个相关性并不一定有一种。那多种相关性体现到计算方式上就是有多个矩阵 $q$ ，不同的 $q$ 负责代表不同的相关性。我们以2 heads为例，先使用 $a$ 计算得到 $q$ ，然后让 $q$ 乘以两个矩阵 $W^{q,1}$ 和 $W^{q,2}$ 得到 $q^{i,1}$ 和 $q^{i,2}$ ，代表2 heads，以同样的方式处理 $k$ 和 $v$ ：

## 双头注意力的结构



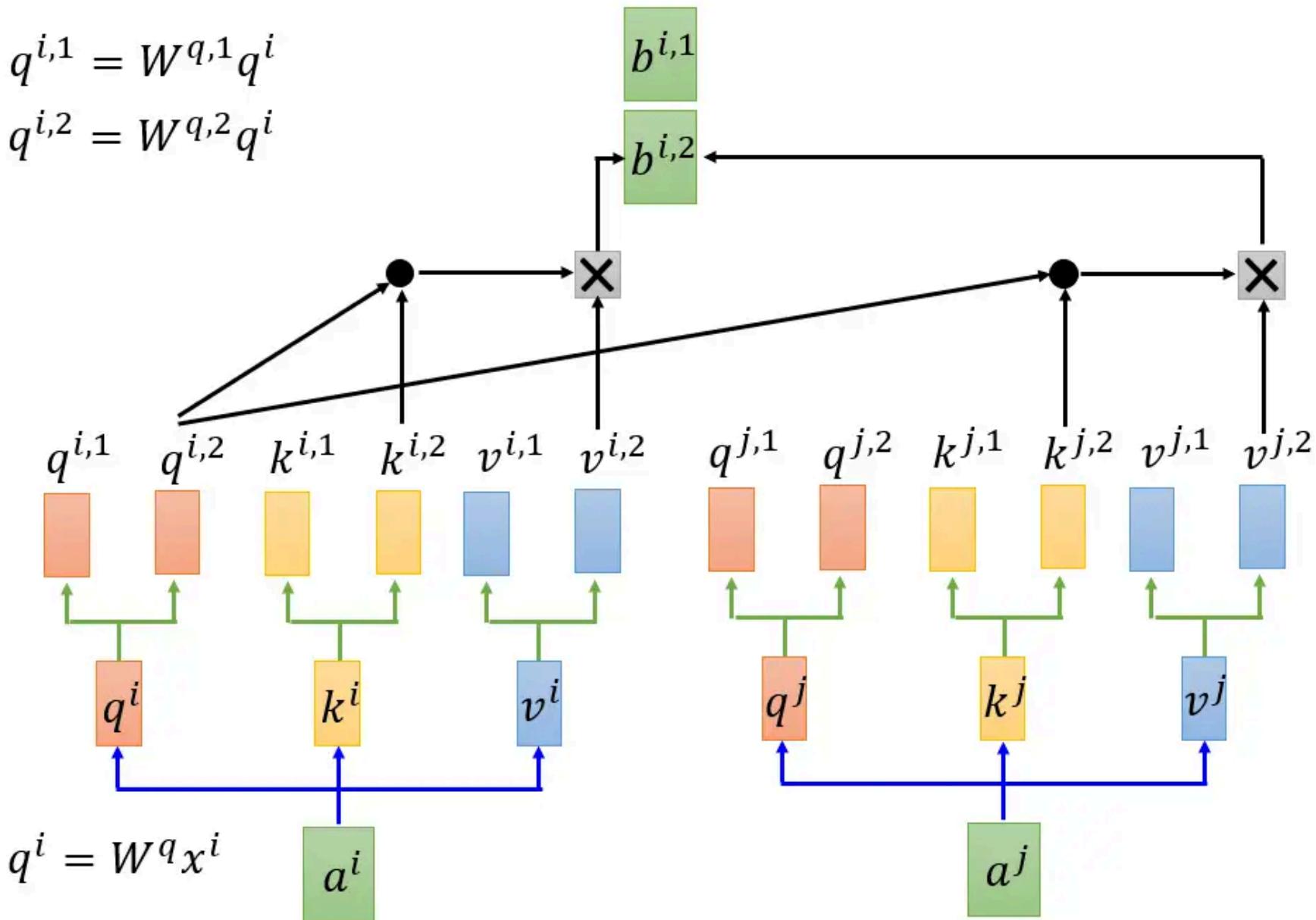
在后续的计算中，我们只将属于相同相关性的矩阵进行运算，如下图所示：



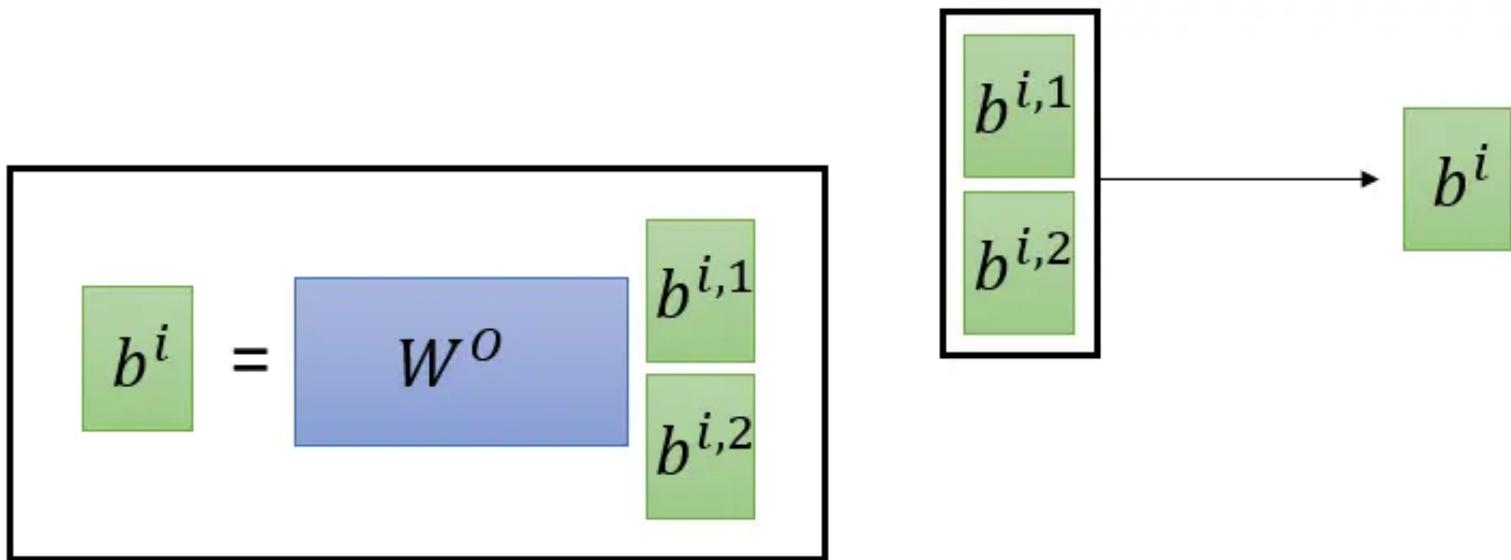
## ***Multi-head Self-attention*** (2 heads as example)

$$q^{i,1} = W^{q,1} q^i$$

$$q^{i,2} = W^{q,2} q^i$$



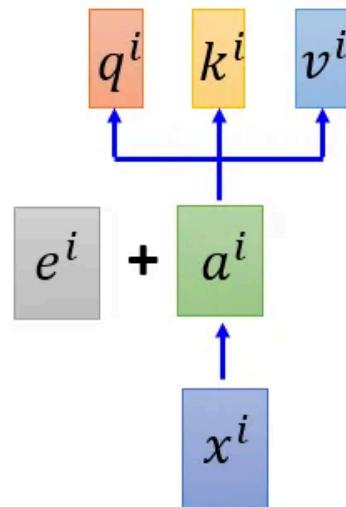
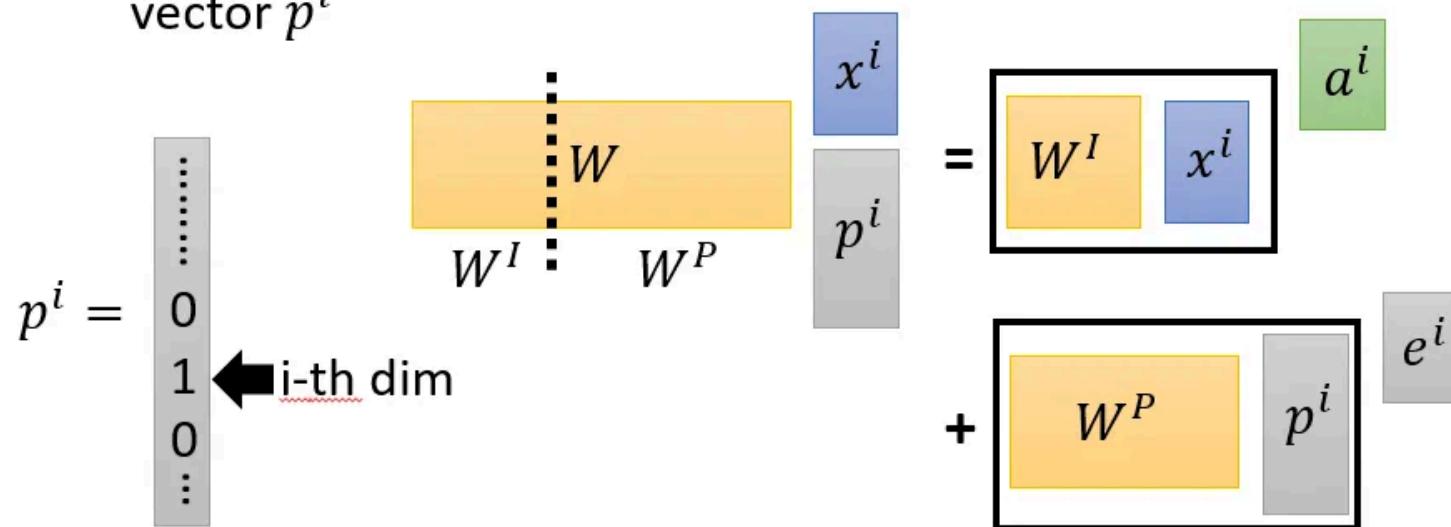
- $q^{i,1}$  分别与  $k^{i,1}$  和  $k^{j,1}$  计算得到  $\alpha_{1,1}^1$  和  $\alpha_{1,2}^1$
- 然后将  $\alpha_{1,1}^1$  和  $\alpha_{1,2}^1$  分别与  $v^{i,1}$  和  $v^{j,1}$  相乘得到  $b^{i,1}$
- 我们以同样的方式，得到矩阵  $b^{i,2}$ ，将  $b^{i,1}$  和  $b^{i,2}$  拼起来乘以一个矩阵  $W^O$  得到最后的输入  $b^i$ 。



# Position Encoding

## Positional Encoding

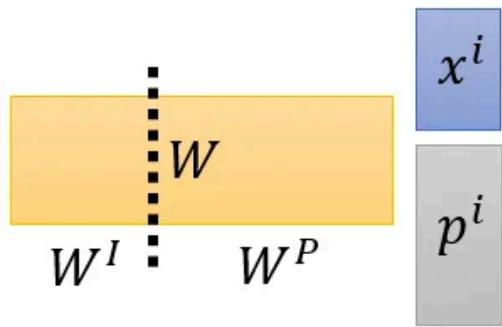
- No position information in self-attention.
- Original paper: each position has a unique positional vector  $e^i$  (not learned from data)
- In other words: each  $x^i$  appends a one-hot vector  $p^i$



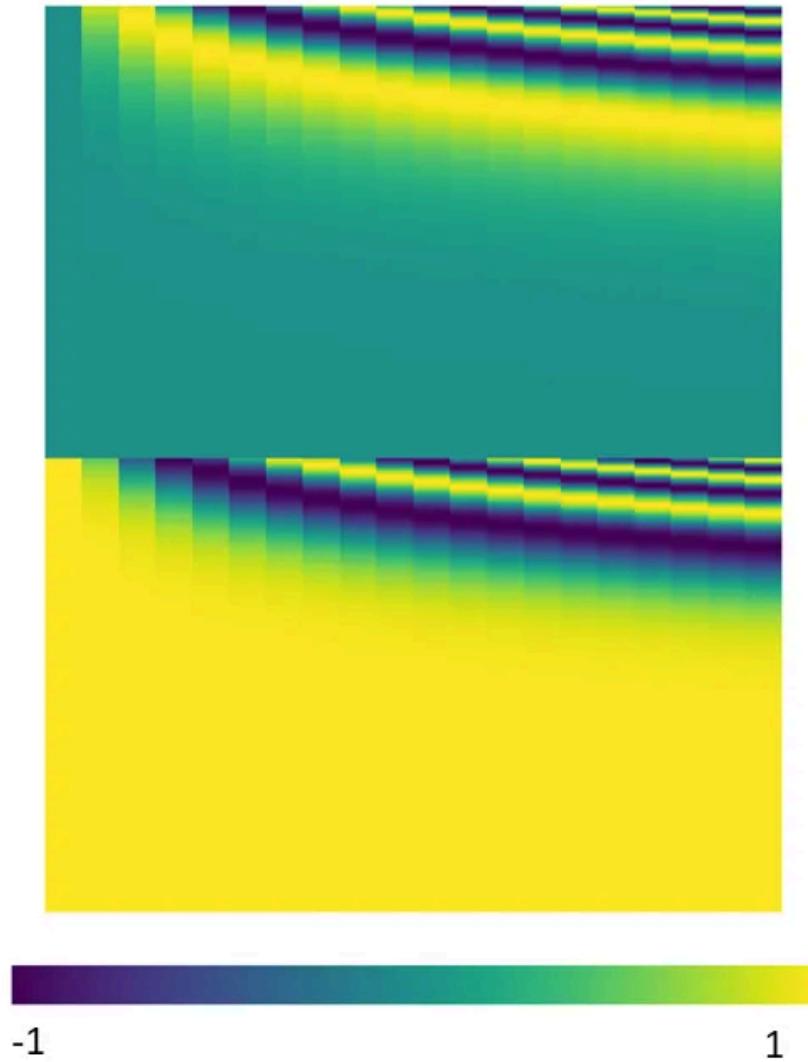
一种解释方式是，将每个与one-hot vector concat一下然后乘以一个matrix W（可以分成与）得到（与 点乘）与（与 点乘），视为position vector（不是网络学出来的，而是事先人为设定好的）。

注：one-hot vector 就是一个向量第*i*个位置为1，其余位置为0。

$W^P$ 长这个样子：



$$= \boxed{W^I \quad x^i} \quad a^i \\ + \boxed{W^P \quad p^i} \quad e^i$$



# 关于位置编码的研究：

[https://arxiv.org/abs/  
2003.09229](https://arxiv.org/abs/2003.09229)

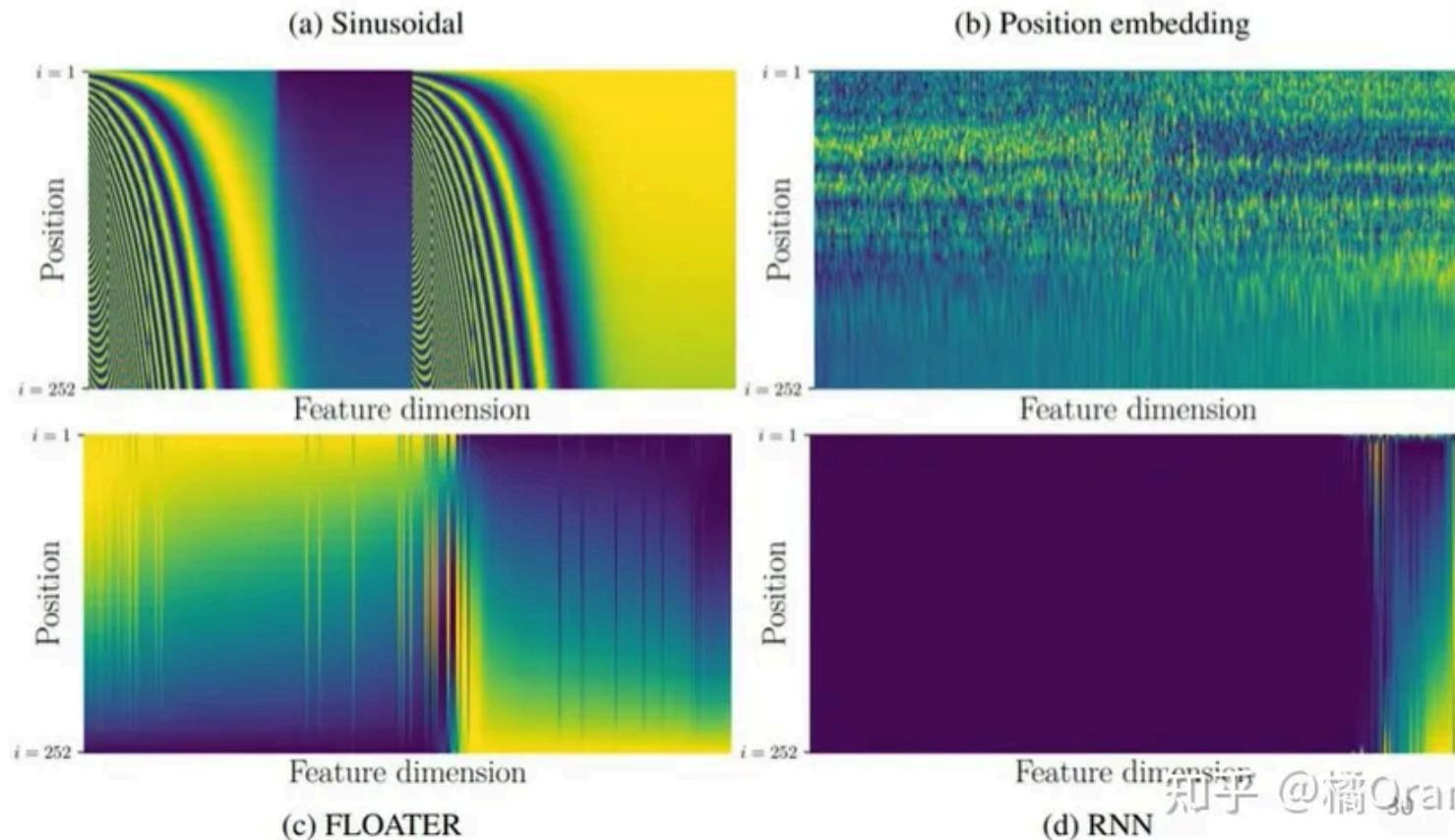


Table 1. Comparing position representation methods

Methods	Inductive	Data-Driven	Parameter Efficient
Sinusoidal (Vaswani et al., 2017)	✓	✗	✓
Embedding (Devlin et al., 2018)	✗	✓	✗
Relative (Shaw et al., 2018)	✗	✓	✓
This paper	✓	✓	✓

## self-attention 和 CNN 的比较：

CNN 可以看作简化版的 self-attention；

或者 self-attention 是复杂版的 CNN。receptive field 不再是人工画出来，而是自动学习出来。

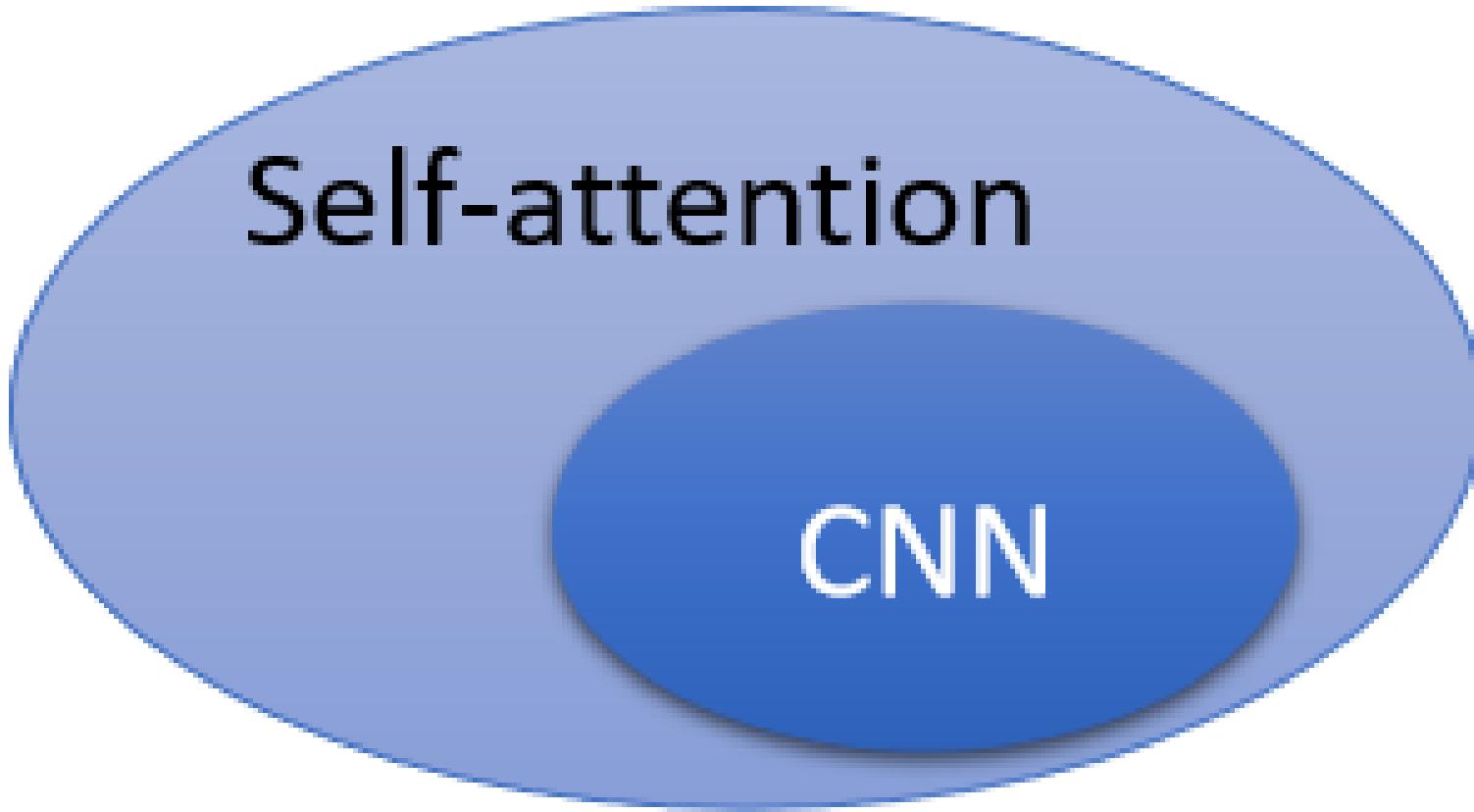
CNN 是 self-attention 的特例

self-attention 更加灵活，若将它进行一定限制，会和 CNN 的结果一模一样。

论证 self-attention 和 CNN 比较的论文

具体的比较：

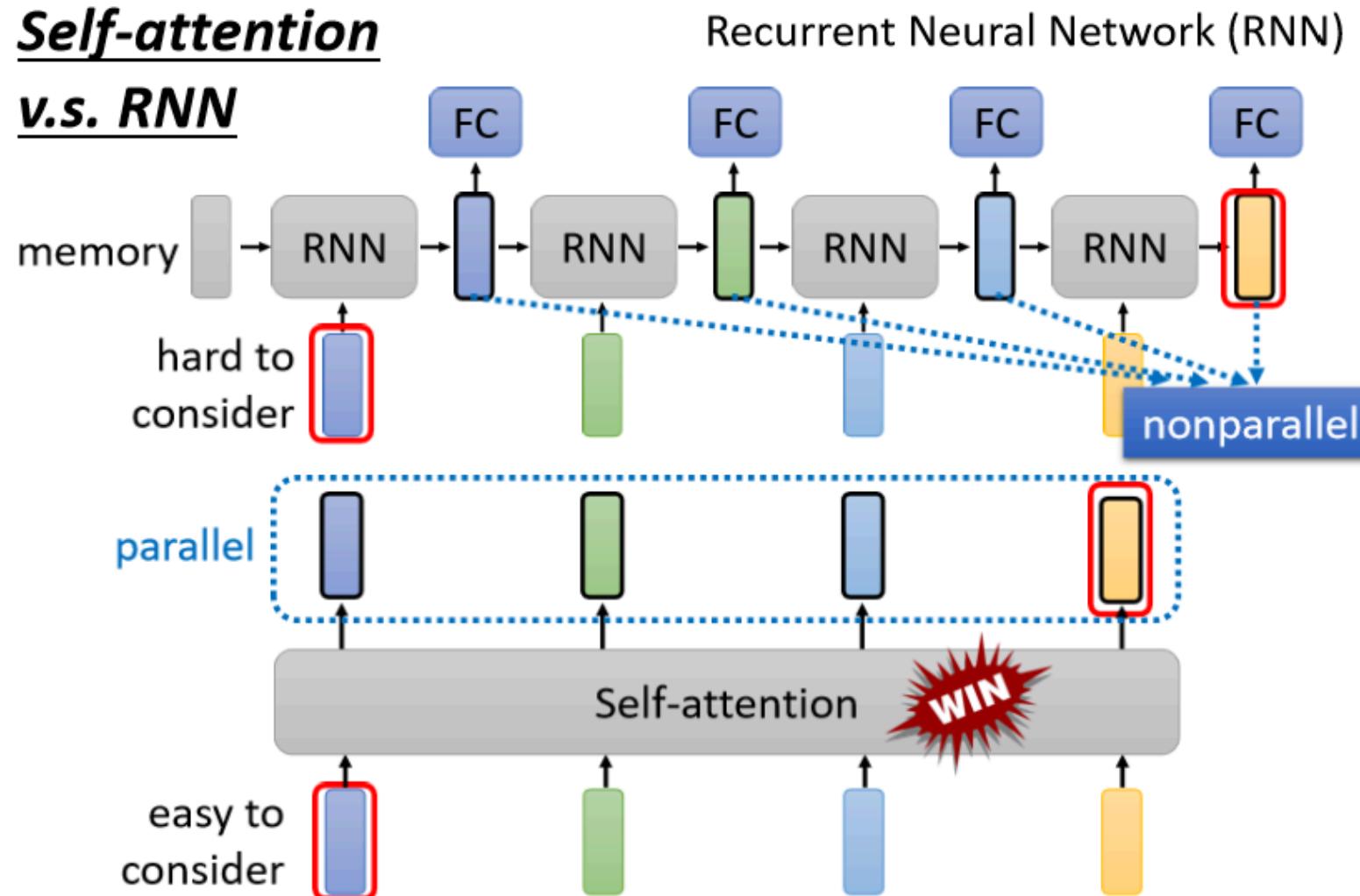
横轴：图片数据量，纵轴为表现。发现当数据量小的时候，CNN 的表现更好，当数据量非常大时，self-attention 的效果（才）会更好。



Self-attention

CNN

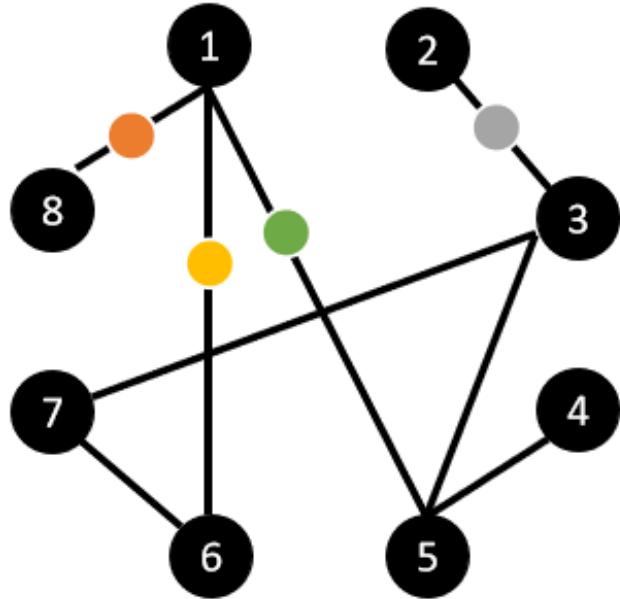
- RNN 必须串行处理，无法并行计算。self-attention 可以并行。
- 如何将 RNN 和 self-attention 结合在一起呢？



**attention + Graph 就是一种 GNN 啦。**

GNN 的课程传送门：

self-attention 的缺点：计算量太大。因此后来各式各样的变形也用于解决运算速度问题。



Consider **edge**: only attention to connected nodes

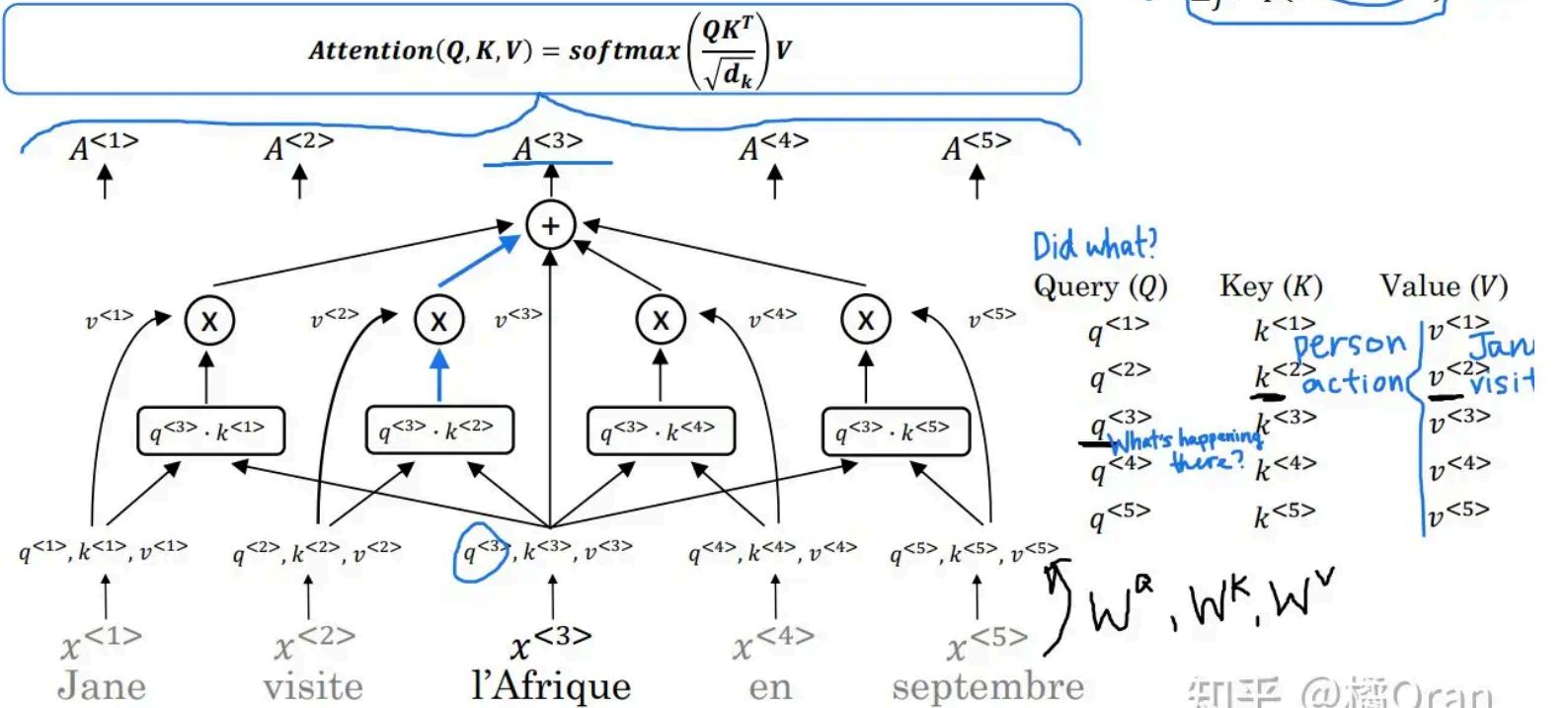
<i>Attention Matrix</i>							
1	2	3	4	5	6	7	8
1							
2							
3							
4							
5							
6							
7							
8							
1							
2							
3							
4							
5							
6							
7							
8							

This is one type of **Graph Neural Network (GNN)**.

各种各样的 Transformers 变形

Andrew Ng 的讲解：

## Self-Attention



[Vaswani et al. 2017, Attention Is All You Need]

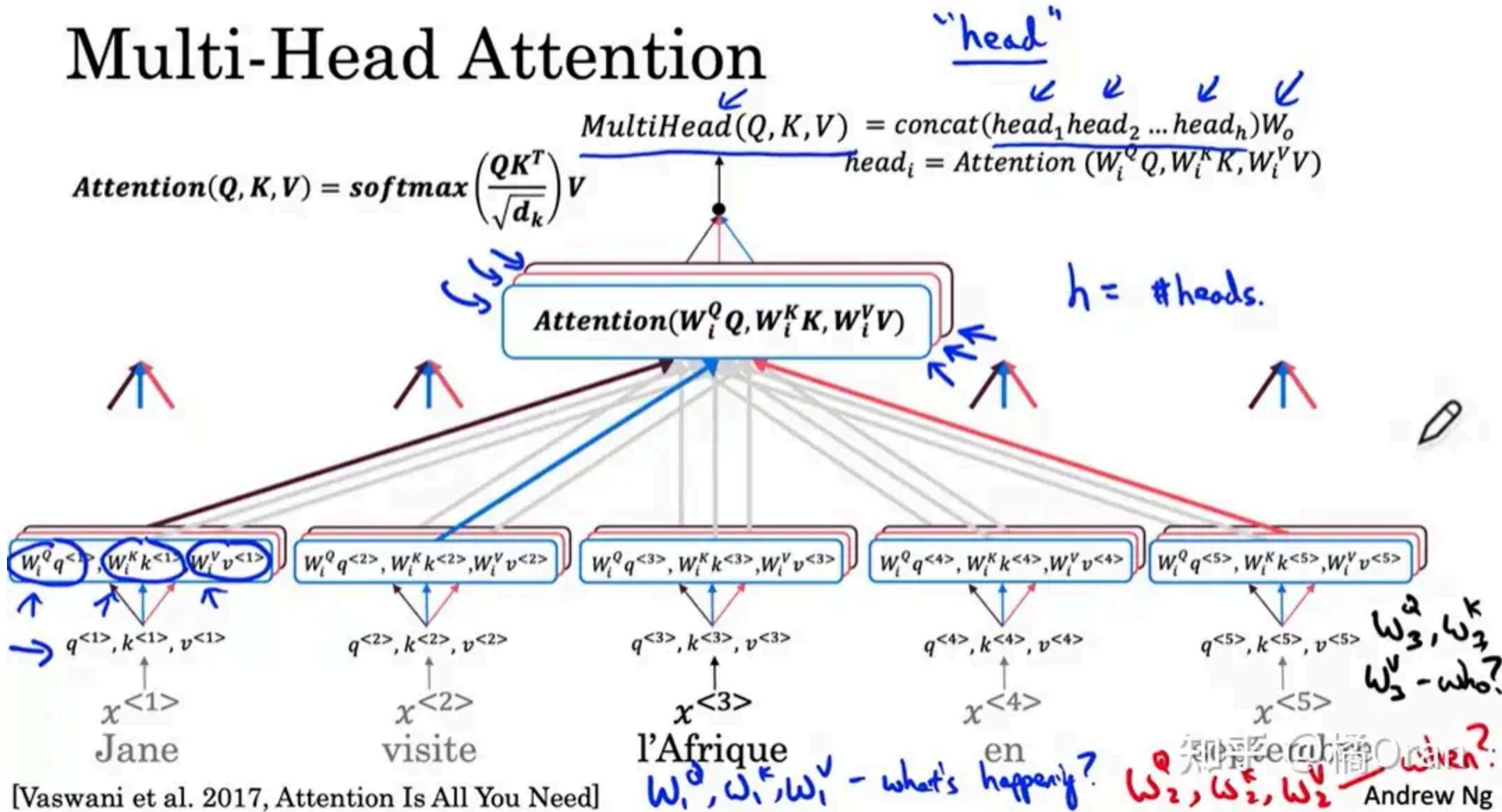
- 参考：<https://zhuanlan.zhihu.com/p/505105707>

### 3. 多头注意力

多头注意力就是上面 self-attention 的 big for-loop。

三个 head 是可以并行训练的，输出的时候 concat 在一起，然后再乘以个权重矩阵，得到输出。

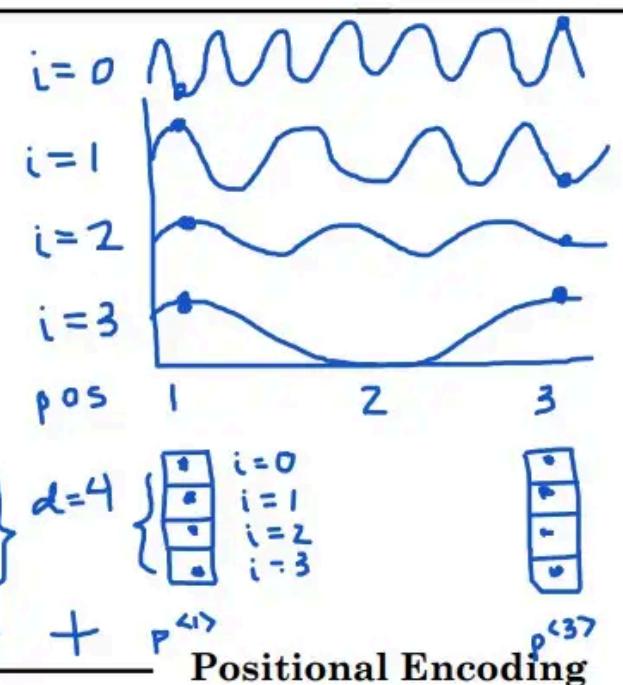
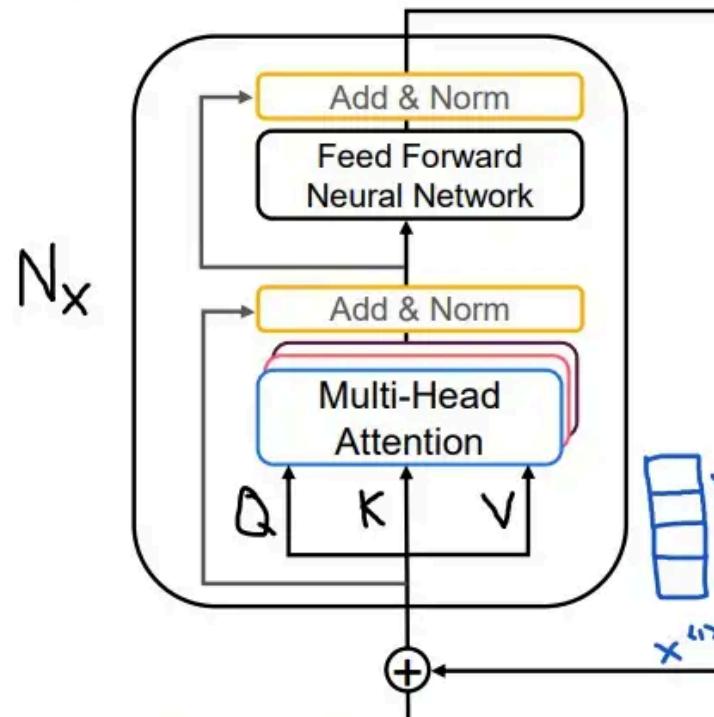
# Multi-Head Attention



[Vaswani et al. 2017, Attention Is All You Need]

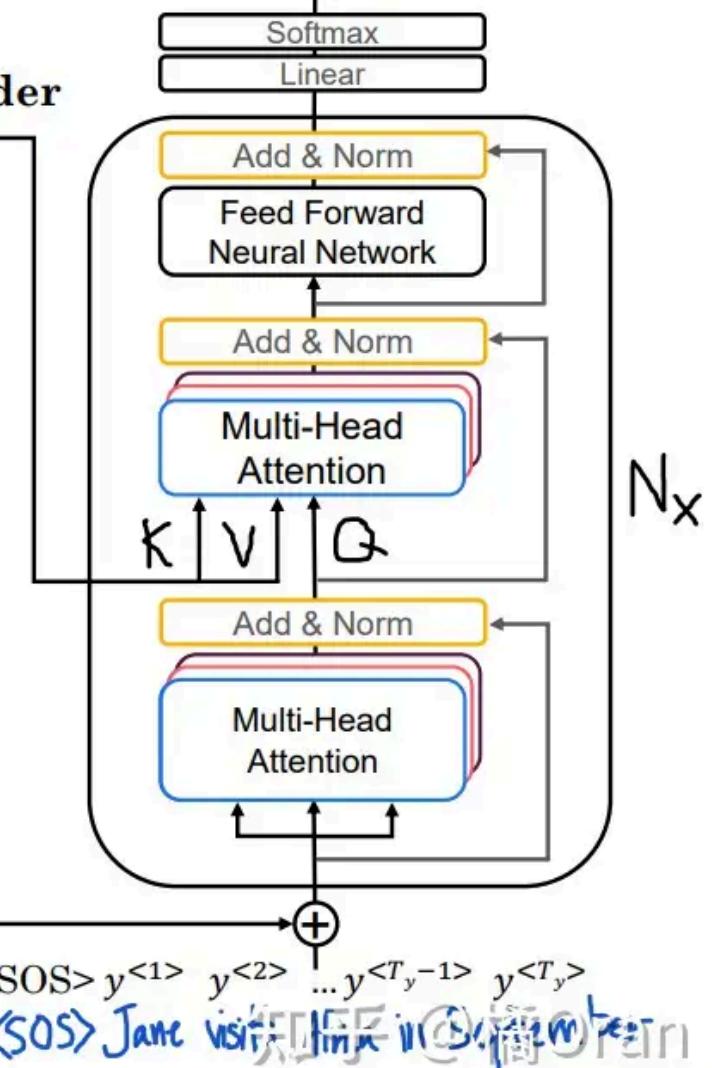
# Transformer Details

Encoder



<SOS> Jane visits Africa in September <EOS>

Decoder



## 总结

- Self-attention 就本质上是依然是一种特殊的 Attention 其实是 Attention 的一个变体，改变了计算相关性权重的计算方式，从输出和输入之间的相关计算，转变成输入和输入自身的相关性计算。
- Attention能够帮我们找到子序列和全局的相关度的关系，也就是找到权重值 $w_i$ 。Self-Attention 对于 Attention 的变化，其实就是寻找权重值  $w_i$  的过程不同。原来，我们计算时使用的是子序列和全局，而现在我们计算 Self-attention 时，用的是自己和自己，这是 Attention 和 Self-attention 从计算上来说最大的区别。

- Self-attention 的运算过程。为了能够产生输出的向量  $y_i$ ，Self-attention 其实是对所有的输入做了一个加权平均的操作，这个公式和上面的 Attention 是一致的。

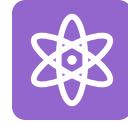
$$y_i = \sum_j w_{ij} x_j$$

$j$  代表整个序列的长度，并且  $j$  个权重的相加之和等于 1。这里的  $w_{ij}$  并不是一个需要神经网络学习的参数，它是来源于  $x_j$  和  $x_i$ （这里  $x_i$  和  $x_j$  就都是自己 self）的之间的计算的结果。而它们之间最简单的一种计算方式，就是使用点积的方式。

$$w'_{ij} = x_i^T x_j$$

这个点积的输出的取值范围在负无穷和正无穷之间，所以要使用一个 Softmax 把它映射到  $[0,1]$  之间，并且要确保它们对于整个序列而言的和为 1。

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

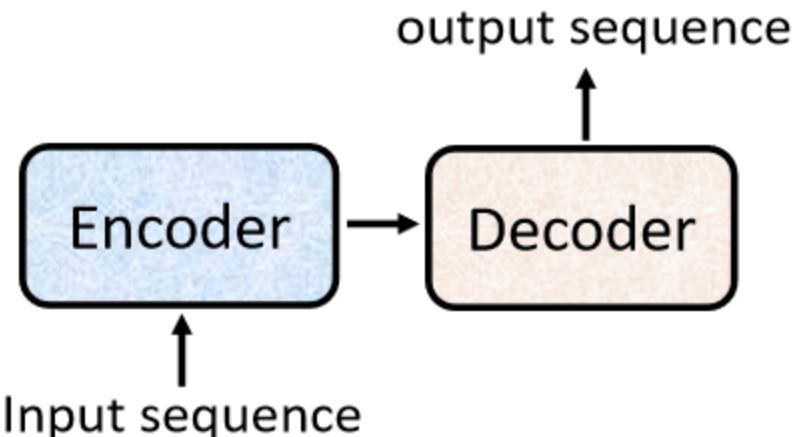


# Transformer

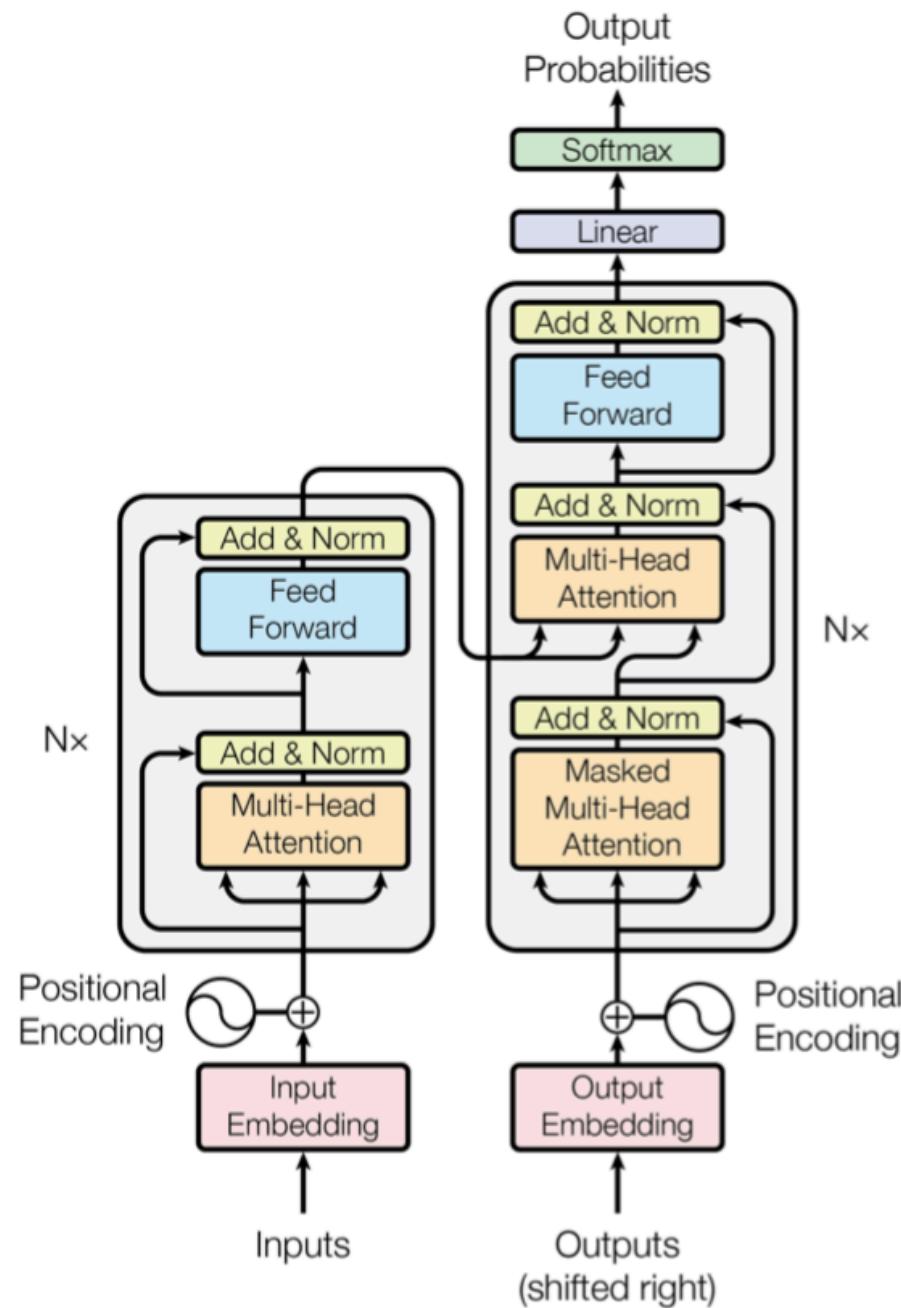
Transformer 模型来源于 Google 发表的一篇论文 "Attention Is All You Need"

- <https://arxiv.org/abs/1706.03762>

Transformer模型是一个基于多头自注意力的序列到序列模型（seq2seq model），整个网络结构可以分为编码器（Encoder）和解码器（Decoder）两部分。



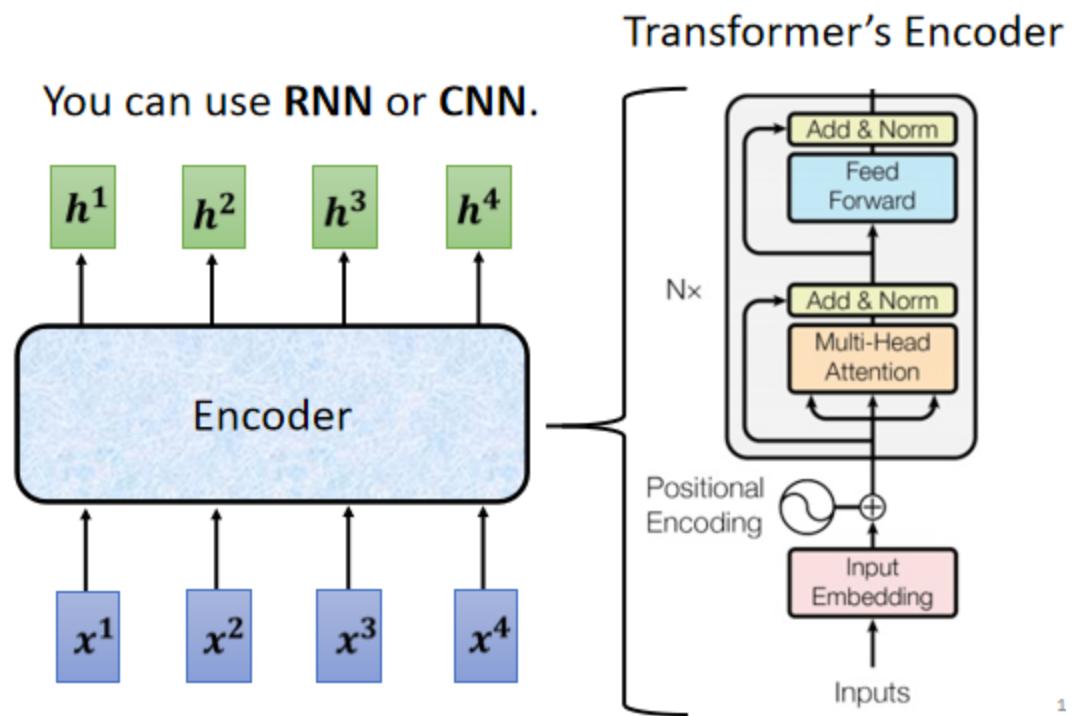
这个seq2seq模型输出序列的长度是不确定的。我们输入一个sequence后，先由Encoder负责处理，再把处理好的结果输入到Decoder中，由Decoder决定最后输出什么样的sequence。Transformer的完整结构如下图所示：



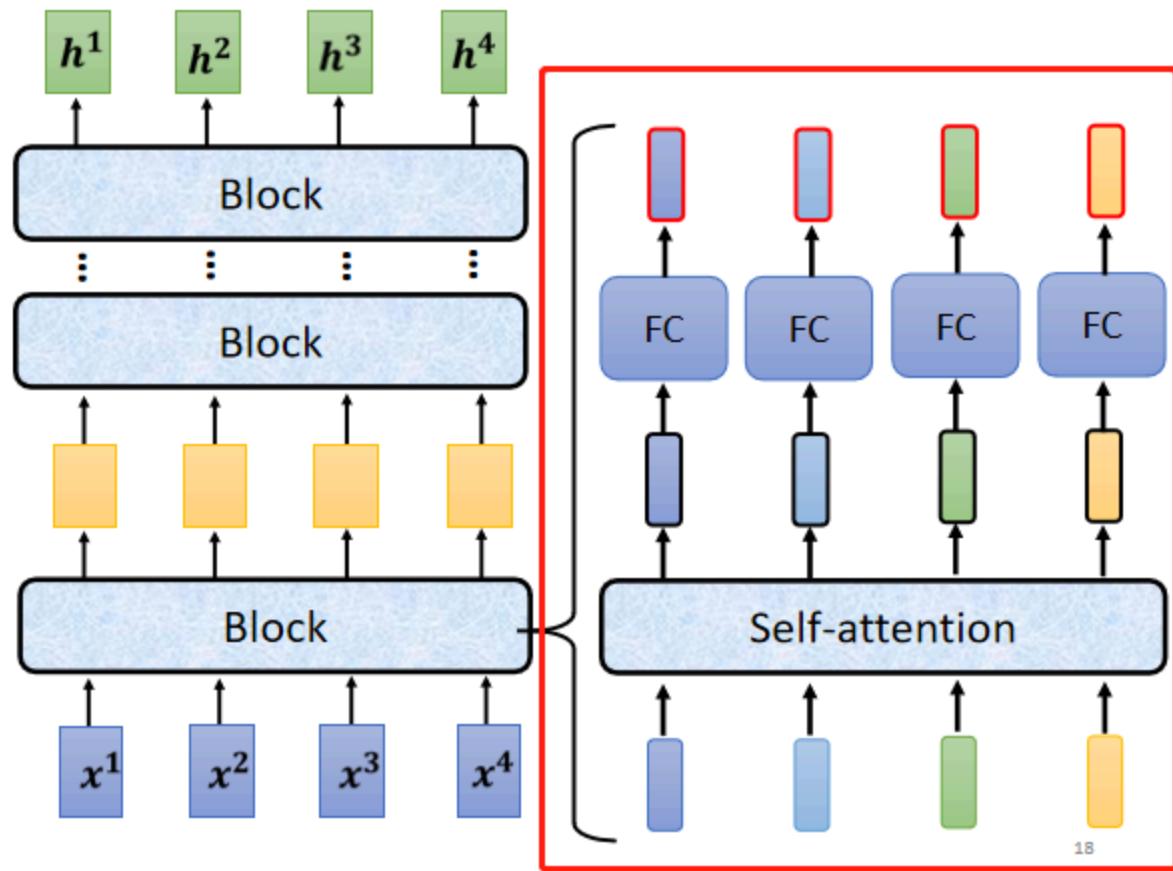
Encoder-Decoder架构中，有三处Multi-head Attention模块，分别是：

- Encoder模块的Self-Attention，在Encoder中，每层的Self-Attention的输入 $Q = K = V$ ，都是上一层的输出。Encoder中的每个position都能够获取到前一层的所有位置的输出。
- Decoder模块的Mask Self-Attention，在Decoder中，每个position只能获取到之前position的信息，因此需要做mask，将其设置为 $-\infty$
- Encoder-Decoder之间的Attention，其中 $Q$ 来自于之前的Decoder层输出， $K, V$ 来自于encoder的输出，这样decoder的每个位置都能够获取到输入序列的所有位置信息。

在seq2seq模型中的Encoder要做的事情就是输入一排向量，输出另一排向量。能实现输入一排向量，输出一排向量功能的模型有Self-attention、CNN和RNN等模型，而Transformer中用得到的则是Multi-Head attention模型。



- 在将输入向量进行self-attention之前，先加上Positional Encoding，也就是输入向量中的位置信息。
- Multi-Head Attention：进行Multi-Head的self-attention处理得到输出向量。
- Add & Norm (residual & layer normalization): 也就是将self-attention的输出加上它对应的输入然后对其进行Layer Normalization。
- Feed Forward：将上一层的输出输入到fully connected中，将得到的输出向量同样经过residual & layer normalization操作后得到该block的最终输出。  
将这个block重复n次。



## Encoder

Encoder有 $N = 6$ 层，每层包括两个sub-layers:

- 第一个sub-layer是multi-head self-attention mechanism，用来计算输入的self-attention
- 第二个sub-layer是简单的全连接网络。
- 在每个sub-layer都模拟了残差网络，每个sub-layer的输出都是 $LayerNorm(x + Sublayer(x))$

其中Sublayer( $x$ ) 表示Sub-layer对输入  $x$  做的映射，为了确保连接，所有的sub-layers和embedding layer输出的维数都相同  $d_{model} = 512$ .

## Decoder

Decoder分为Auto regressive (AT) 和Non-Auto regressive (NAT) 两种。其中AT应用范围更为广泛一些。

### Auto regressive (AT)

以语音辨识为例，假设我们要处理的这个NLP问题，每一个Token都用One-Hot的Vector表示，并假设START和END两个special token，其中START表示开始工作，END表示结束工作。

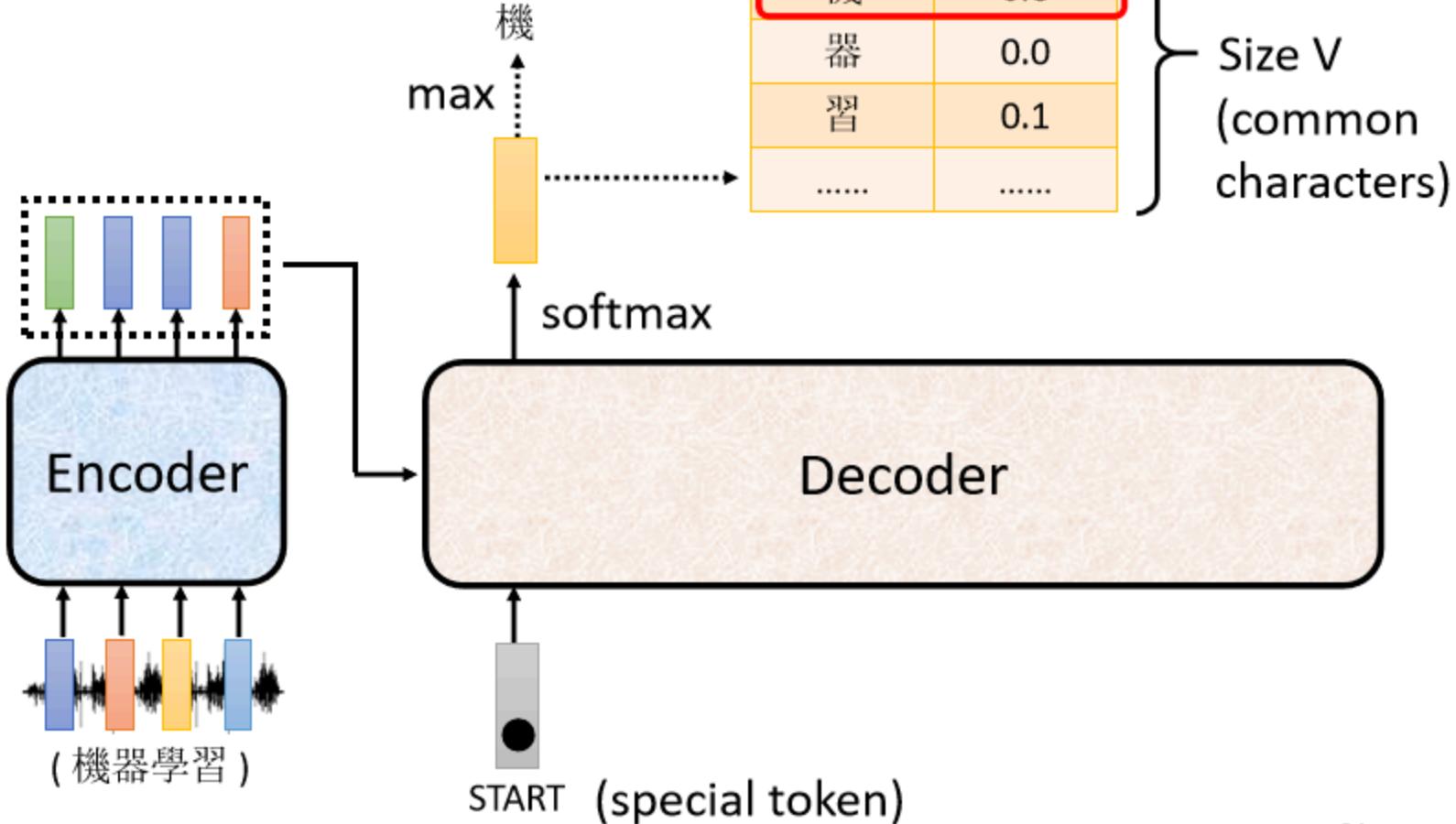
在Encoder完成之后，将其输出作为一个输入喂到Decoder中。

同时，输入一个special token： START表示开始工作。

Decoder结合这两个输入，输出一个经过softmax处理后的长度为Vocabulary Size的输出向量，该向量中每一个中文字都对应一个数值，数值最大的中文字为最终输出的中文字，下图中，输出的结果是“机”。

# Autoregressive

(Speech Recognition as example)



Decoder也是 $N = 6$ 层，每层包括3个sub-layers：

- 第一个是Masked multi-head self-attention，也是计算输入的self-attention，但是因为是生成过程，因此在时刻的时候，大于的时刻都没有结果，只有小于的时刻有结果，因此需要做Mask
- 第二个sub-layer是全连接网络，与Encoder相同
- 第三个sub-layer是对encoder的输入进行attention计算。
- 同时Decoder中的self-attention层需要进行修改，因为只能获取到当前时刻之前的输入，因此只对时刻之前的时刻输入进行attention计算，这也称为Mask操作。

## Attention机制

在Transformer中使用的Attention是Scaled Dot-Product Attention, 是归一化的点乘Attention, 假设输入的query  $q$ 、key维度为  $d_k$ , value维度为  $d_v$ , 那么就计算query和每个key的点乘操作, 并除以  $\sqrt{d_k}$ , 然后应用Softmax函数计算权重。

$$\text{Attention}(Q, K_i, V_i) = \text{softmax}\left(\frac{Q^T K_i}{\sqrt{d_k}}\right) V_i$$

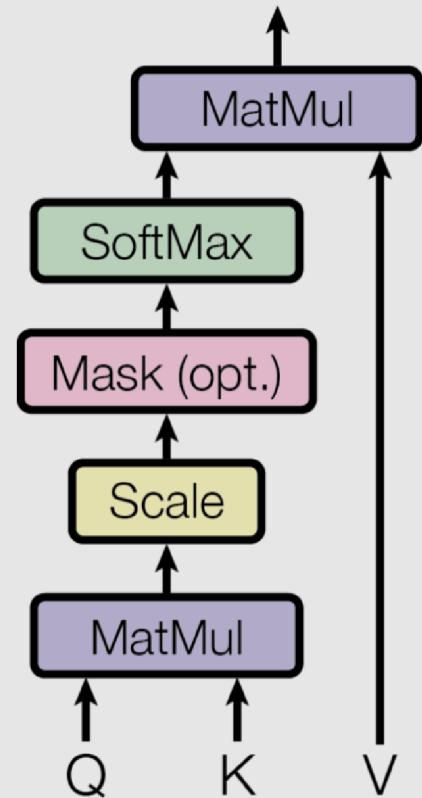
在实践中, 将query和keys、values分别处理为矩阵, 那么计算输出矩阵为:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q^T K}{\sqrt{d_k}}\right) V$$

其中  $Q \in R^{m \times d_s}$ ,  $K \in R^{m \times d_k}$ ,  $V \in R^{m \times d_v}$ , 输出矩阵维度为  $R^{m \times d_v}$ 。

## Scaled Dot-Product Attention

那么Scaled Dot-Product Attention的示意图如图所示，Mask是可选的(opt.)，如果是能够获取到所有时刻的输入( $K, V$ )，那么就不使用Mask；如果是不能获取到，那么就需要使用Mask。使用了Mask的Transformer模型也被称为Transformer Decoder，不使用Mask的Transformer模型也被称为Transformer Encoder。



如果只对  $Q$ 、 $K$ 、 $V$  做一次这样的权重操作是不够的，这里提出了 Multi-Head Attention，操作包括：

- 首先对  $Q$ 、 $K$ 、 $V$  做一次线性映射，将输入维度均为  $d_{model}$  的  $Q, K, V$  矩阵映射到  $Q \in R^{m \times d_s}, K \in R^{m \times d_k}, V \in R^{m \times d_v}$
- 然后在采用 Scaled Dot-Product Attention 计算出结果
- 多次进行上述两步操作，然后将得到的结果进行合并
- 将合并的结果进行线性变换

总结来说公示如下所示

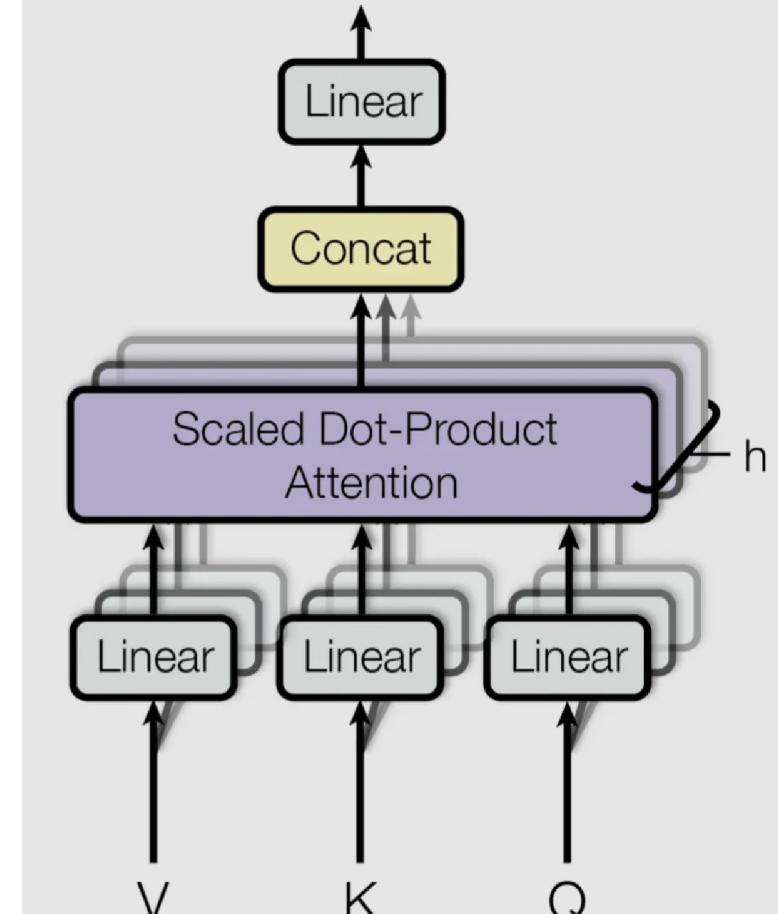
$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

其中第1步的线性变换参数为  $W_i^Q \in R^{d_{model} \times d_k}$ ,  $W_k^K \in R^{d_{model} \times d_k}$ ,  $W_i^V \in R^{d_{model} \times d_v}$   
第4步的线性变化参数为  $W^O \in R^{hd_v \times d_{model}}$ . 而第三步计算的次数是  $h$ 。

在论文中取  $d_{model} = 512$  表示每个时刻的输入维度和输出维度， $h = 8$  表示8次Attention操作， $d_k = d_v = \frac{d_{model}}{h} = 64$  表示经过线性变换之后、进行Attention操作之前的维度。那么进行一次Attention之后输出的矩阵维度是  $R^{m \times d_v} = R^{m \times 64}$ ，然后进行  $h = 8$  次操作合并之后输出的结果是  $R^{m \times (h \times d_v)} = R^{m \times 512}$ ，因此输入和输出的矩阵维度相同。

这样输出的矩阵，每行的向量都是对向量中每一行的加权，示意图如图所示

### Multi-Head Attention





# BERT

BERT的全称为Bidirectional Encoder Representation from Transformers，是一个预训练的语言表征模型。它强调了不再像以往一样采用传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的masked language model (MLM)，以致能生成深度的双向语言表征。BERT论文发表时提及在11个NLP (Natural Language Processing, 自然语言处理) 任务中获得了新的state-of-the-art的结果。

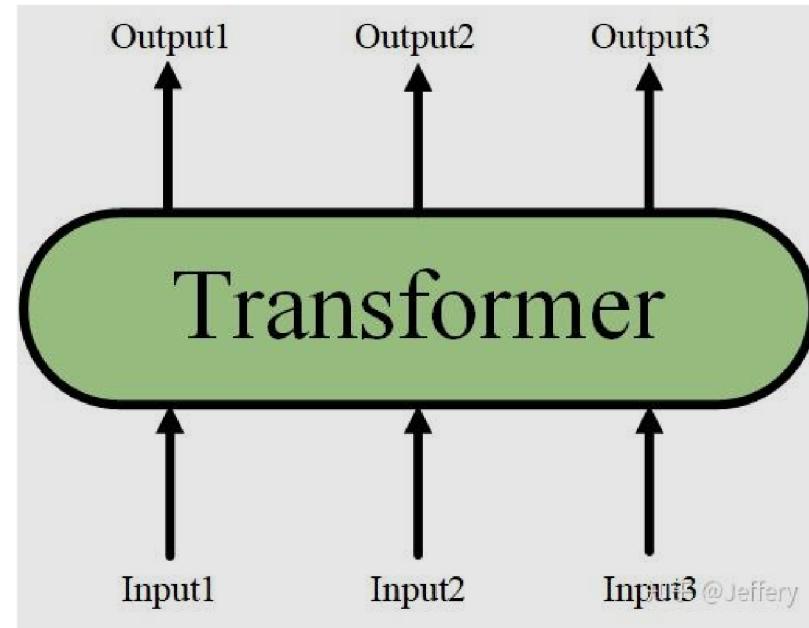
该模型有以下主要优点：

- 采用MLM对双向的Transformers进行预训练，以生成深层的双向语言表征。
- 预训练后，只需要添加一个额外的输出层进行fine-tune，就可以在各种各样的下游任务中取得state-of-the-art的表现。在这过程中并不需要对BERT进行任务特定的结构修改。

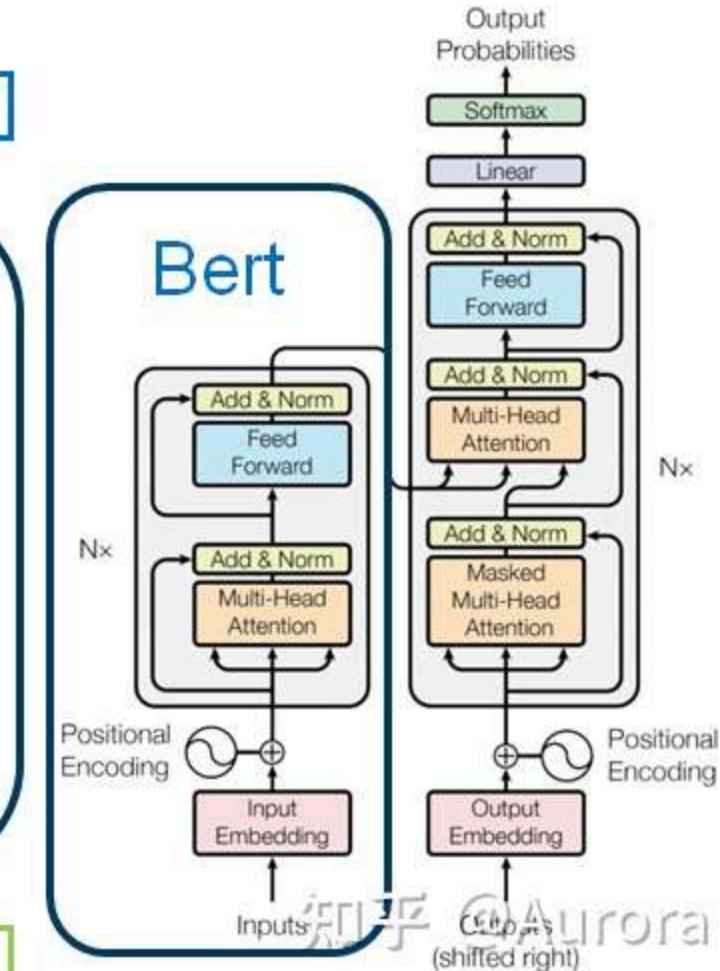
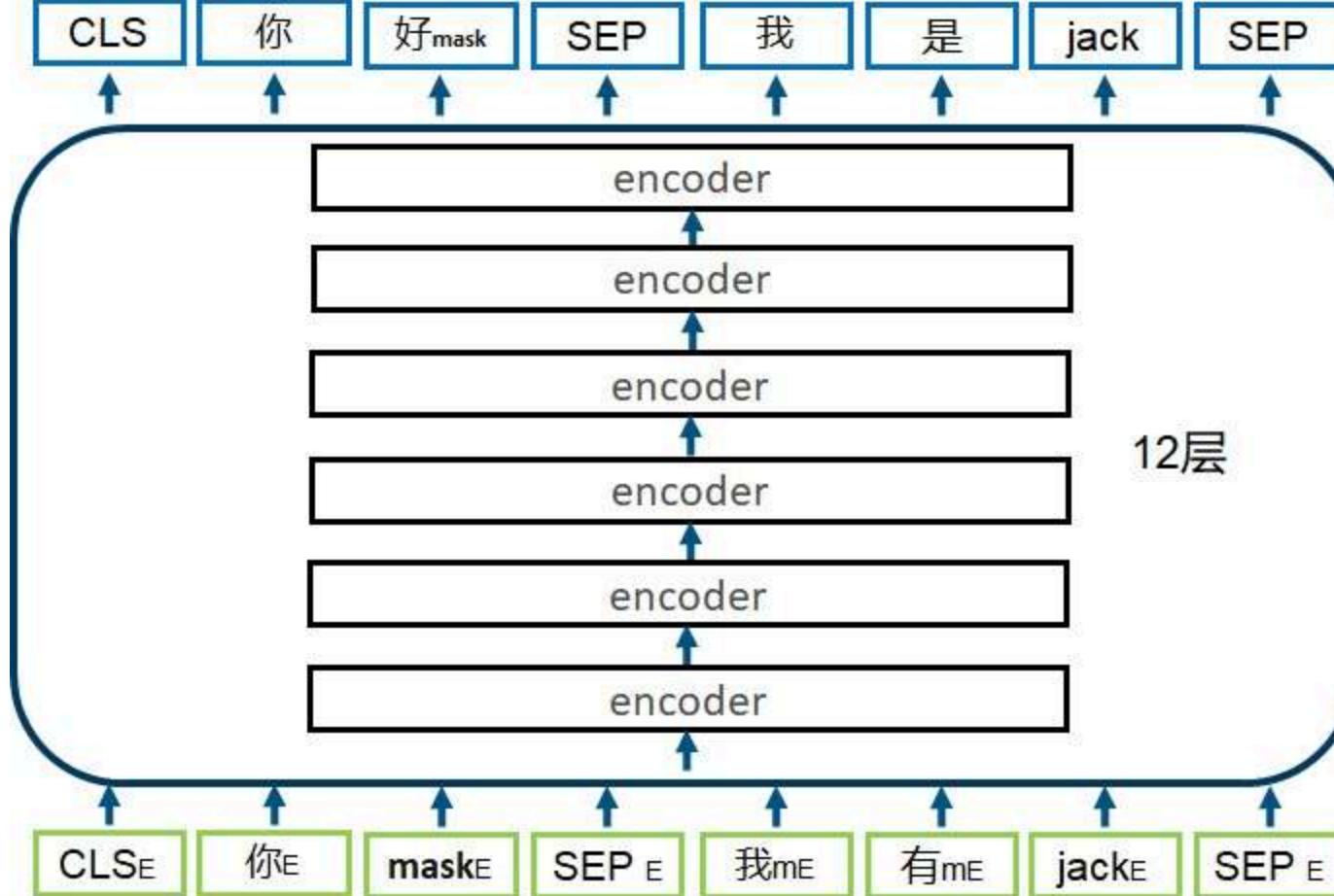
## BERT的结构

以往的预训练模型的结构会受到单向语言模型（从左到右或者从右到左）的限制，因而也限制了模型的表征能力，使其只能获取单方向的上下文信息。而BERT利用MLM进行预训练并且采用深层的双向Transformer组件（单向的Transformer一般被称为Transformer decoder，其每一个token（符号）只会attend到目前往左的token。而双向的Transformer则被称为Transformer encoder，其每一个token会attend到所有的token。）来构建整个模型，因此最终生成能融合左右上下文信息的深层双向语言表征。

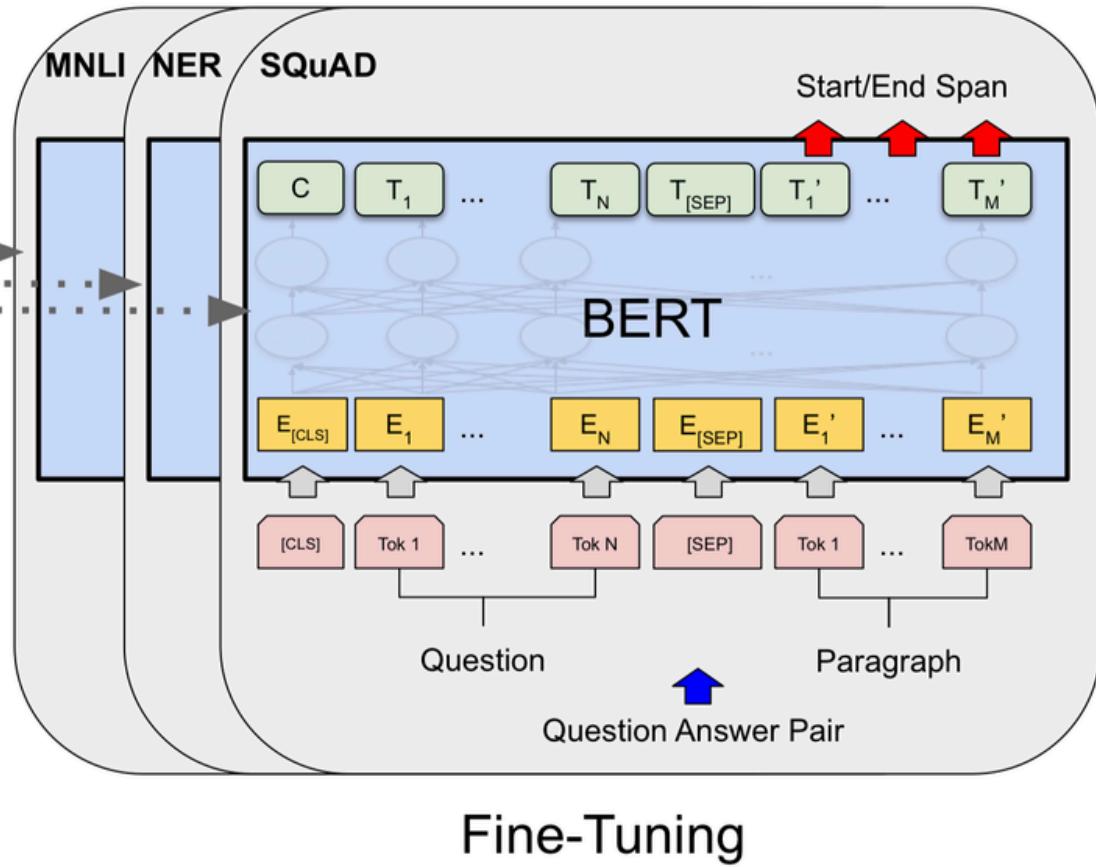
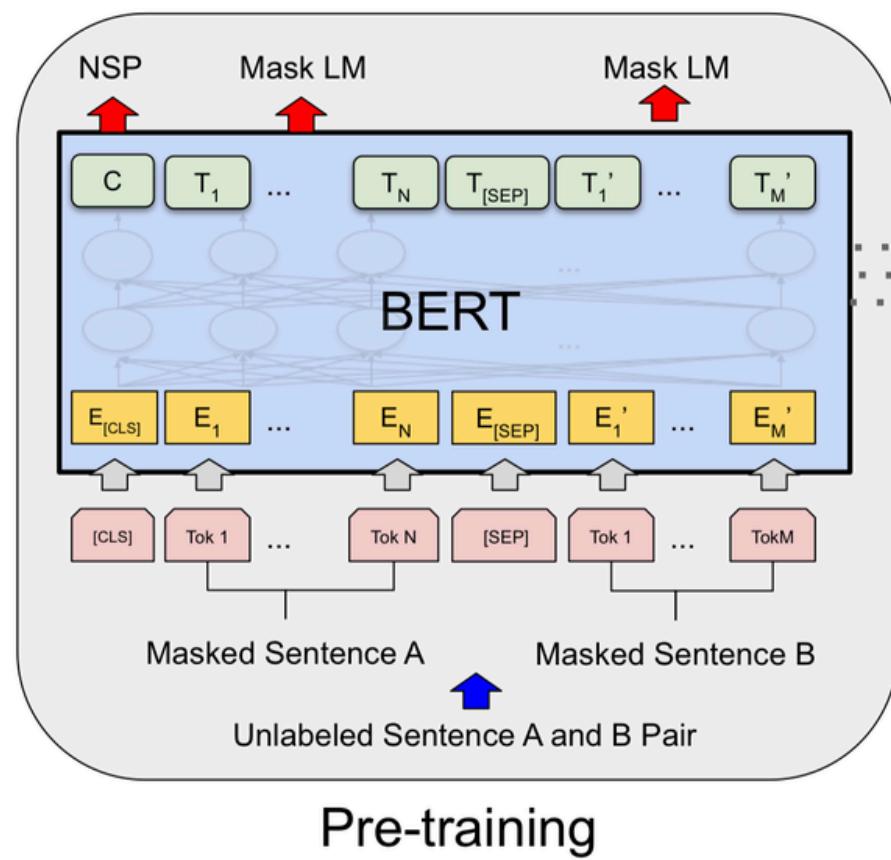
当隐藏了Transformer的详细结构后，可以用一个只有输入和输出的黑盒子来表示它了：



而Transformer结构又可以进行堆叠，形成一个更深的神经网络（这里也可以理解为将Transformer encoder进行堆叠）：



最终，经过多层Transformer结构的堆叠后，形成BERT的主体结构：



在论文中，作者分别用12层和24层Transformer Encoder组装了两套BERT模型，两套模型的参数总数分别为110M和340M。

## 下游任务

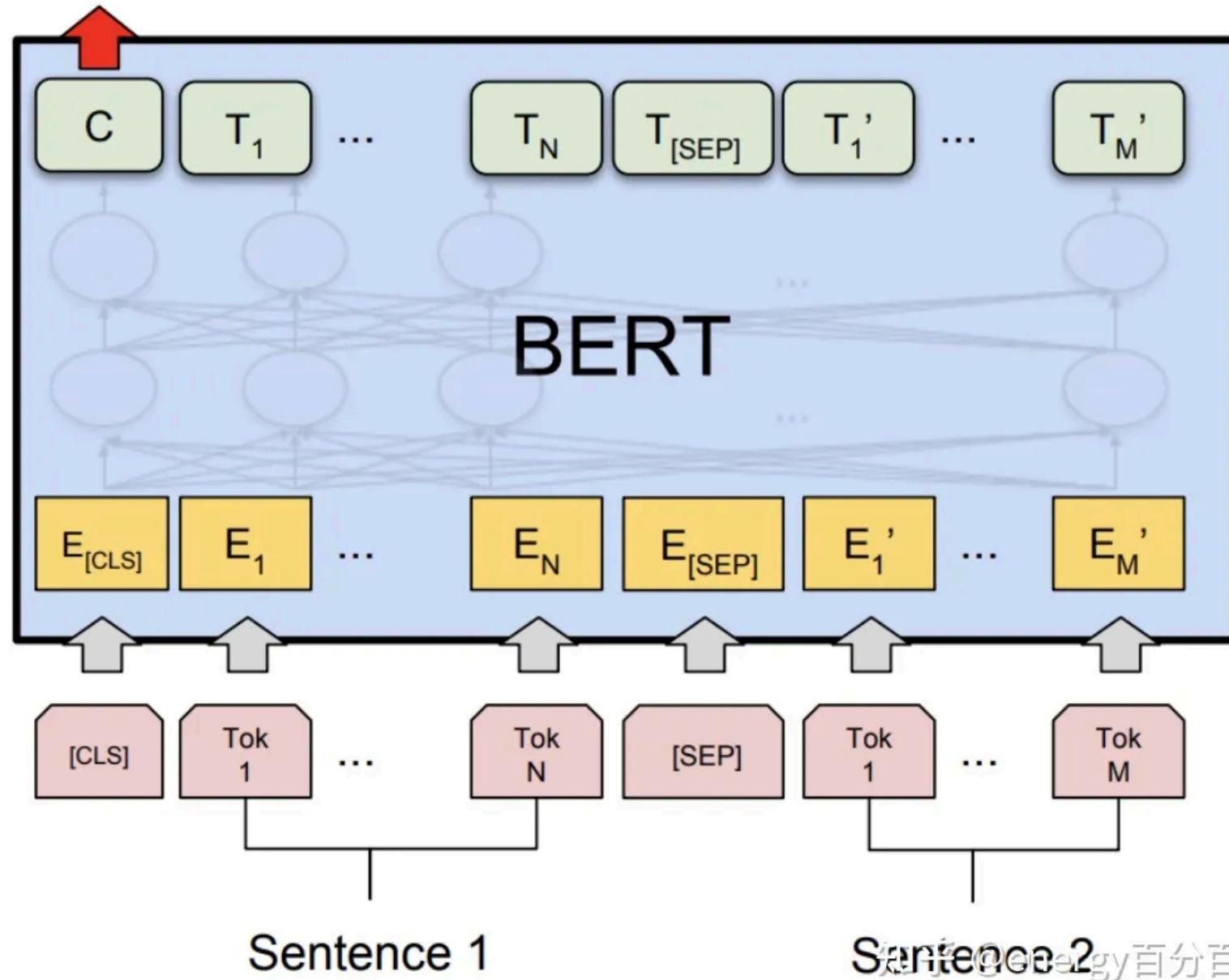
### 分类

句子对分类任务属于序列级任务（对于每个序列只计算 Bert 模型中一个输出的损失），使用 Bert 模型解决句子对分类任务需要对 Bert 模型做如下调整：

对输入的序列添加 [CLS] 和 [SEP] 两个符号，使 Bert 能够接受语句级别的输入； [SEP] 用来分隔序列中的两个句子， [CLS] 对用位置的输出用来进行得到分类结果  
在 [SEP] 位置对应的输出后增加一个分类层（全连接层+softmax层），用于输出最后的分类概率

修改后的模型如下图所示：

Class  
Label



## GLUE

- 自然语言处理（NLP）主要自然语言理解（NLU）和自然语言生成（NLG）。
- 为了让NLU任务发挥最大的作用，来自纽约大学、华盛顿大学等机构创建了一个多任务的自然语言理解基准和分析平台，也就是GLUE（General Language Understanding Evaluation）。
- GLUE包含九项NLU任务，语言均为英语。GLUE九项任务涉及到自然语言推断、文本蕴含、情感分析、语义相似等多个任务。像BERT、XLNet、RoBERTa、ERNIE、T5等知名模型都会在此基准上进行测试。
- 大家要把预测结果上传到官方的网站上，官方会给出测试的结果。
- GLUE的论文为：GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding[1]

## GLUE共有九个任务：

分别是CoLA、SST-2、MRPC、STS-B、QQP、MNLI、QNLI、RTE、WNLI。可以分为三类，分别是单句任务，相似性和释义任务，

### 1 CoLA

CoLA(The Corpus of Linguistic Acceptability, 语言可接受性语料库)，单句子分类任务，语料来自语言理论的书籍和期刊，每个句子被标注为是否合乎语法的单词序列。本任务是一个二分类任务，标签共两个，分别是0和1，其中0表示不合乎语法，1表示合乎语法。

样本个数：训练集8,551个，开发集1,043个，测试集1,063个。

任务：可接受程度，合乎语法与不合乎语法二分类。

评价准则：Matthews correlation coefficient。

标签为1（合乎语法）的样例：

She is proud.

she is the mother.

John thinks Mary left.

Yes, she did.

Will John not go to school?

Mary noticed John's excessive appreciation of himself.

标签为0（不合语法）的样例：

Mary sent.

Yes, she used.

Mary wonders for Bill to come.

They are intense of Bill.

Mary thinks whether Bill will come.

Mary noticed John's excessive appreciation of herself.

注意到，这里面的句子看起来不是很长，有些错误是性别不符，有些是缺词、少词，有些是加s不加s的情况，各种语法错误。但我也注意到，有一些看起来错误并没有那么严重，甚至在某些情况还是可以说的通的。

## 2 SST-2

SST-2(The Stanford Sentiment Treebank, 斯坦福情感树库), 单句子分类任务, 包含电影评论中的句子和它们情感的人类注释。这项任务是给定句子的情感, 类别分为两类正面情感 (positive, 样本标签对应为1) 和负面情感 (negative, 样本标签对应为0) , 并且只用句子级别的标签。也就是, 本任务也是一个二分类任务, 针对句子级别, 分为正面和负面情感。

样本个数: 训练集67,350个, 开发集873个, 测试集1,821个。

任务: 情感分类, 正面情感和负面情感二分类。

评价准则: accuracy。

标签为1（正面情感， positive）的样例：

two central performances  
against shimmering cinematography that lends the setting the ethereal beauty of an asian  
landscape painting  
the situation in a well-balanced fashion  
a better movie  
at achieving the modest , crowd-pleasing goals it sets for itself  
a patient viewer

标签为0（负面情感， negative）的样例：

a transparently hypocritical work that feels as though it 's trying to set the women 's liberation movement back 20 years  
so pat it makes your teeth hurt  
blood work is laughable in the solemnity with which it tries to pump life into overworked elements from eastwood 's dirty harry period .  
faced with the possibility that her life is meaningless , vapid and devoid of substance , in a movie that is definitely meaningless , vapid and devoid of substance monotone  
this new jangle of noise , mayhem and stupidity must be a serious contender for the title .  
注意到，由于句子来源于电影评论，又有它们情感的人类注释，不同于CoLA的整体偏短，有些句子很长，有些句子很短，长短并不整齐划一。

### 3 MRPC

MRPC(The Microsoft Research Paraphrase Corpus, 微软研究院释义语料库), 相似性和释义任务, 是从在线新闻源中自动抽取句子对语料库, 并人工注释句子对中的句子是否在语义上等效。类别并不平衡, 其中68%的正样本, 所以遵循常规的做法, 报告准确率(accuracy) 和F1值。

样本个数: 训练集3,668个, 开发集408个, 测试集1,725个。

任务: 是否释义二分类, 是释义, 不是释义两类。

评价准则: 准确率 (accuracy) 和F1值。

标签为1 (正样本，互为释义) 的样例 (每个样例是两句话，中间用tab隔开) :

The largest gains were seen in prices , new orders , inventories and exports . Sub-indexes measuring prices , new orders , inventories and exports increased .

Trading in Loral was halted yesterday ; the shares closed on Monday at \$ 3.01 . The New York Stock Exchange suspended trading yesterday in Loral , which closed at \$ 3.01 Friday . He plans to have dinner with troops at Kosovo 's U.S. military headquarters , Camp Bondsteel . After that , he plans to have dinner at Camp Bondsteel with U.S. troops stationed there .

Retailers J.C. Penney Co . Inc . ( JCP ) and Walgreen Co . ( WAG ) kick things off on Monday . Retailers J.C. Penney Co . Inc . JCP.N and Walgreen Co . WAG.N kick things off on Monday .

标签为0（负样本，不互为释义）的样例：

Earnings per share from recurring operations will be 13 cents to 14 cents . That beat the company 's April earnings forecast of 8 to 9 cents a share .

He beat testicular cancer that had spread to his lungs and brain . Armstrong , 31 , battled testicular cancer that spread to his brain .

Graves reported from Albuquerque , Villafranca from Austin and Ratcliffe from Laredo . Pete Slover reported from Laredo and Gromer Jeffers from Albuquerque .

The commission must work out the plan 's details , but the average residential customer paying \$ 840 a year would get a savings of about \$ 30 annually . An average residential customer paying \$ 840 a year for electricity could see a savings of \$ 30 annually .

A former teammate , Carlton Dotson , has been charged with the murder . His body was found July 25 , and former teammate Carlton Dotson has been charged in his shooting death .

本任务的数据集，包含两句话，每个样本的句子长度都非常长，且数据不均衡，正样本占比68%，负样本仅占32%。

## 4 STSB

STSB(The Semantic Textual Similarity Benchmark, 语义文本相似性基准测试), 相似性和释义任务, 是从新闻标题、视频标题、图像标题以及自然语言推断数据中提取的句子对的集合, 每对都是由人类注释的, 其相似性评分为0-5(大于等于0且小于等于5的浮点数, 原始paper里写的是1-5, 可能是作者失误)。任务就是预测这些相似性得分, 本质上是一个回归问题, 但是依然可以用分类的方法, 可以归类为句子对的文本五分类任务。

样本个数: 训练集5,749个, 开发集1,379个, 测试集1,377个。

任务: 回归任务, 预测为1-5之间的相似性得分的浮点数。但是依然可以使用分类的方法, 作为五分类。

评价准则: Pearson and Spearman correlation coefficients。

一些训练集中的样例句子对及其得分：

A plane is taking off. An air plane is taking off. 5.000

A man is playing a large flute. A man is playing a flute. 3.800

A dog rides a skateboard. A dog is riding a skateboard. 5.000

A woman is playing the flute. A man is playing the guitar. 1.000

A man is playing the guitar. A man is playing the drums. 1.556

A cat is playing a piano. A man is playing a guitar. 0.600

A group of people dance on a hill. A group of people are dancing. 3.200

A woman is sitting at a desk. A woman is riding a donkey. 0.400

Someone is slicing tortila's. Someone is riding a horse. 0.000

A man is playing the guitar. A man plays an acoustic guitar. 3.750

整体句子长度适中偏短，且均衡。

## 5 QQP

QQP(The Quora Question Pairs, Quora问题对数集), 相似性和释义任务, 是社区问答网站Quora中问题对的集合。任务是确定一对问题在语义上是否等效。与MRPC一样, QQP也是正负样本不均衡的, 不同的是的QQP负样本占63%, 正样本是37%, 所以我们也是报告准确率和F1值。我们使用标准测试集, 为此我们从作者那里获得了专用标签。我们观察到测试集与训练集分布不同。

样本个数: 训练集363,870个, 开发集40,431个, 测试集390,965个。

任务: 判定句子对是否等效, 等效、不等效两种情况, 二分类任务。

评价准则: 准确率 (accuracy) 和F1值。

标签为1（正样本，互为释义，等效）的样例（每个样例是两句话，中间用tab隔开）：

How can I improve my communication and verbal skills? What should we do to improve communication skills?

What has Hillary Clinton done that makes her trustworthy? Why do Democrats consider Hillary Clinton trustworthy?

What are the top countries worth visiting? What are the top ten countries you think are most worth visiting in your lifetime, and why?

hat would happen if all the people in the world didn't need to sleep? Where would humans be if we didn't need sleep?

Why is Russia improving relations with Israel so much in 2016? Why is Russia and Israel improving relations with each other so much as of 2016?

hy does the iPad Mini say not charging? Why is my iPad Mini not charging?

标签为0（负样本，不互为释义，不等效）的样例：

Why are you so sexy? How sexy are you?

Which programming languages are common to develop in the area of gamification? Who is the worst Director in the history of MNIT/MREC?

How do I solve  $3^{1/3}$ ? How do I solve  $(x^2 - 1) / (x - 3) < 0$ ?

Why is the Mophie case charged by a micro-USB cable, and not a lightning cable? Which USB hub should I buy?

Can we do a mobile recharge using the BHIM app? How do I select state bank of Patiala in bhim app?

What is the feeling of love? What it feels to be loved?

类似于MRPC，句子对的释义问题。这里突出的除了样本不均衡、训练集测试集分布不一致外，还有这个训练集、测试集都非常大。这里的测试集比其他训练集都要多好几倍。

## 6 MNLI

MNLI(The Multi-Genre Natural Language Inference Corpus, 多类型自然语言推理数据库), 自然语言推断任务, 是通过众包方式对句子对进行文本蕴含标注的集合。给定前提 (premise) 语句和假设 (hypothesis) 语句, 任务是预测前提语句是否包含假设 (蕴含, entailment), 与假设矛盾 (矛盾, contradiction) 或者两者都不 (中立, neutral)。前提语句是从数十种不同来源收集的, 包括转录的语音, 小说和政府报告。

样本个数: 训练集392,702个, 开发集dev-matched 9,815个, 开发集dev-mismatched 9,832个, 测试集test-matched 9,796个, 测试集test-dismatched 9,847个。因为MNLI是集合了许多不同领域风格的文本, 所以又分为了matched和mismatched两个版本的数据集, matched指的是训练集和测试集的数据来源一致, mismatched指的是训练集和测试集来源不一致。

任务: 句子对, 一个前提, 一个是假设。前提和假设的关系有三种情况: 蕴含 (entailment), 矛盾 (contradiction), 中立 (neutral)。句子对三分类问题。

评价准则: matched accuracy/mismatched accuracy。

标签为蕴含 (entailment) 的句子对示例：

you know during the season and i guess at at your level uh you lose them to the next level if if they decide to recall the the parent team the Braves decide to call to recall a guy from triple A then a double A guy goes up to replace him and a single A guy goes up to replace him You lose the things to the following level if the people recall.

ow do you know? All this is their information again. This information belongs to them.

well you see that on television also You can see that on television, as well.

According to the Office of the Actuary at the Health Care Financing Administration, the estimated net present value of future additional resources needed to fund HI benefits alone over the 75 years is \$4. The net present value of future additional resources for funding HI benefits was \$4.

标签为矛盾 (contradiction) 的句子对示例：

They're made from a secret recipe handed down to the present-day villagers by their Mallorcan ancestors, who came here in the early 17th century as part of an official repopulation scheme. The recipe passed down from Mallorcan ancestors is known to everyone.

Felicia's Journey takes place behind the eyes of its central a young Irish girl, Felicia, who crosses the sea to England in a hopeful quest to find the father of her unborn child; and the fat, middle-aged catering manager, Hiditch, who takes a paternal interest in the lass when it becomes clear that her young man has caddishly given her the slip. The woman did not care where the man was as long as it was far.

Poirot, I exclaimed, with relief, and seizing him by both hands, I dragged him into the room. Poirot was now back and I was sorry that he would take over what I now considered my own investigation.

but that takes too much planning It doesn't take much planning.

标签为中立 (neutral) 的句子对示例：

Conceptually cream skimming has two basic dimensions - product and geography.

Product and geography are what make cream skimming work.

hebes held onto power until the 12th Dynasty, when its first king, Amenemhet I who reigned between 1980 1951 b.c. established a capital near Memphis. The capital near Memphis lasted only half a century before its inhabitants abandoned it for the next capital.

When the trust fund begins running cash deficits in 2016, the government as a whole must come up with the cash to finance Social Security's cash deficit by reducing any projected non-Social Security surpluses, borrowing from the public, raising other taxes, or reducing other government spending. The public would generally prefer to see the government reduce its spending in other areas to finance Social Security.

She smiled back. She was so happy she couldn't stop smiling.

总体训练集很充足， GLUE论文作者使用并推荐SNLI数据集[2]作为辅助训练数据。

## 7 QNLI

QNLI(Qusetion-answering NLI, 问答自然语言推断), 自然语言推断任务。QNLI是从另一个数据集The Stanford Question Answering Dataset(斯坦福问答数据集, SQuAD 1.0)[3]转换而来的。SQuAD 1.0是有一个问题-段落对组成的问答数据集, 其中段落来自维基百科, 段落中的一个句子包含问题的答案。这里可以看到有个要素, 来自维基百科的段落, 问题, 段落中的一个句子包含问题的答案。通过将问题和上下文 (即维基百科段落) 中的每一句话进行组合, 并过滤掉词汇重叠比较低的句子对就得到了QNLI中的句子对。相比原始SQuAD任务, 消除了模型选择准确答案的要求; 也消除了简化的假设, 即答案适中在输入中并且词汇重叠是可靠的提示。

样本个数: 训练集104,743个, 开发集5,463个, 测试集5,461个。

任务: 判断问题 (question) 和句子 (sentence, 维基百科段落中的一句) 是否蕴含, 蕴含和不蕴含, 二分类。

评价准则: 准确率 (accuracy) 。

标签为蕴含 (entailment, 正样本) 的样例 (每个样例是两句话, 中间用tab隔开, 第一句是问题, 第二句是上下文中的一句) :

What did Arsenal consider the yellow and blue colors to be after losing a FA Cup final wearing red and white? Arsenal then competed in three consecutive FA Cup finals between 1978 and 1980 wearing their "lucky" yellow and blue strip, which remained the club's away strip until the release of a green and navy away kit in 1982–83.

Which collection of minor poems are sometimes attributed to Virgil? A number of minor poems, collected in the Appendix Vergiliana, are sometimes attributed to him.

What does confrontational scavenging involve doing to other predators after they've made a kill? Robert Blumenschine proposed the idea of confrontational scavenging, which involves challenging and scaring off other predators after they have made a kill, which he suggests could have been the leading method of obtaining protein-rich meat by early humans.

Why were dogs initially selected? Unlike other domestic species which were primarily selected for production-related traits, dogs were initially selected for their behaviors.

标签为不蕴含 (not\_entailment, 负样本) 的样例 (每个样例是两句话, 中间用tab隔开, 第一句是问题, 第二句是上下文中的一句) :

When did the third Digimon series begin? Unlike the two seasons before it and most of the seasons that followed, Digimon Tamers takes a darker and more realistic approach to its story featuring Digimon who do not reincarnate after their deaths and more complex character development in the original Japanese.

While looking for bugs, what else can testing do? Although testing can determine the correctness of software under the assumption of some specific hypotheses (see hierarchy of testing difficulty below), testing cannot identify all the defects within software.

What was the highest order of species n land? The climate was much more humid than the Triassic, and as a result, the world was very tropical.

In what century was the church established at the location? Construction of the present church began in 1245, on the orders of King Henry III.

总体就是问答句子组成的问答对, 一个是问题, 一个是句子信息, 后者包含前者的答案就是蕴含, 不包含就是不蕴含, 是一个二分类。

## 8 RTE

RTE(The Recognizing Textual Entailment datasets, 识别文本蕴含数据集), 自然语言推断任务, 它是将一系列的年度文本蕴含挑战赛的数据集进行整合合并而来的, 包含 RTE1[4], RTE2, RTE3[5], RTE5等, 这些数据样本都从新闻和维基百科构建而来。将这些所有数据转换为二分类, 对于三分类的数据, 为了保持一致性, 将中立 (neutral) 和矛盾 (contradiction) 转换为不蕴含 (not entailment) 。

样本个数: 训练集2,491个, 开发集277个, 测试集3,000个。

任务: 判断句子对是否蕴含, 句子1和句子2是否互为蕴含, 二分类任务。

评价准则: 准确率 (accuracy) 。

标签为蕴含 (entailment, 正样本) 的样例 (每个样例是两句话, 中间用tab隔开) :

A place of sorrow, after Pope John Paul II died, became a place of celebration, as Roman Catholic faithful gathered in downtown Chicago to mark the installation of new Pope Benedict XVI. Pope Benedict XVI is the new leader of the Roman Catholic Church.

Herceptin was already approved to treat the sickest breast cancer patients, and the company said, Monday, it will discuss with federal regulators the possibility of prescribing the drug for more breast cancer patients. Herceptin can be used to treat breast cancer.

The name for the newest James Bond film has been announced today. The 22nd film, previously known only as "Bond 22", will be called "Quantum of Solace". EON Productions who are producing the film made the announcement today at Pinewood Studios, where production for the film has been under way since last year. The name of the film was inspired by a short story (of the same name) from For Your Eyes Only by Bond creator, Ian Fleming. James Bond was created by Ian Fleming.

The gastric bypass operation, also known as stomach stapling, has become the most common surgical procedure for treating obesity. Obesity is medically treated.

标签为不蕴含 (not\_entailment, 正样本) 的样例 (每个样例是两句话, 中间用tab隔开) :

No Weapons of Mass Destruction Found in Iraq Yet. Weapons of Mass Destruction Found in Iraq.

Sierra is likely to remain in jail at the Hillsborough County jail in her native Tampa until her next hearing on December 20, where she is being held without bail, which would prevent her attending the Washington event on Friday even if she still had permission to perform. Sierra has been in jail since the start of the month after an altercation with police officers outside a Tampa nightclub, which she had been ejected from. She is charged with disorderly intoxication and resisting arrest. Sierra once reached the finals of "American Idol".

## 9 WNLI

WNLI(Winograd NLI, Winograd自然语言推断), 自然语言推断任务, 数据集来自于竞赛数据的转换。Winograd Schema Challenge[6], 该竞赛是一项阅读理解任务, 其中系统必须读一个带有代词的句子, 并从列表中找到代词的指代对象。这些样本都是手动创建的, 以挫败简单的统计方法: 每个样本都取决于句子中单个单词或短语提供的上下文信息。为了将问题转换成句子对分类, 方法是通过用每个可能的列表中的每个可能的指代去替换原始句子中的代词。任务是预测两个句子对是否有关 (蕴含、不蕴含) 。训练集两个类别是均衡的, 测试集是不均衡的, 65%是不蕴含。

样本个数: 训练集635个, 开发集71个, 测试集146个。

任务: 判断句子对是否相关, 蕴含和不蕴含, 二分类任务。

评价准则: 准确率 (accuracy) 。

标签为1 (蕴含, entailment, 正样本) 的样例 (每个样例是两句话, 中间用tab隔开) :

The actress used to be named Terpsichore, but she changed it to Tina a few years ago, because she figured it was too hard to pronounce. Terpsichore was too hard to pronounce. Since Chester was dependent on Uncle Vernon, he couldn't very well marry without his approval He couldn't very well marry without Uncle Vernon's approval

When they had eventually calmed down a bit, and had gotten home, Mr. Farley put the magic pebble in an iron safe. Some day they might want to use it , but really for now, what more could they wish for? Some day they might want to use the magic pebble.

The actress used to be named Terpsichore, but she changed it to Tina a few years ago, because she figured it was easier to pronounce. Tina was easier to pronounce.

Frank was upset with Tom because the toaster he had sold him didn't work. The toaster Tom had sold him didn't work.

My meeting started at 4:00 and I needed to catch the train at 4:30, so there wasn't much time. Luckily, it was delayed, so it worked out. The train was delayed, so it worked out.

标签为0（不蕴含，not\_entailment，正样本）的样例（每个样例是两句话，中间用tab隔开）：

Bill passed the half-empty plate to John because he was hungry. Bill was hungry.

The donkey wished a wart on its hind leg would disappear, and it did. The donkey wished a wart on its hind leg would disappear, and leg did.

The man lifted the boy onto his shoulders. The man lifted the boy onto the boy's shoulders.

The delivery truck zoomed by the school bus because it was going so slow. The delivery truck was going so slow.

We went to the lake, because a shark had been seen at the ocean beach, so it was a safer place to swim. The ocean beach was a safer place to swim.

rchaeologists have concluded that humans lived in Laputa 20,000 years ago. They hunted for evidence on the river banks. Prehistoric humans hunted for evidence on the river banks.

可以看到，这个数据集是数量最少，训练集600多个，测试集才100多个。同时目前GLUE上这个数据集还有些问题。

 参考

<https://zhuanlan.zhihu.com/p/345680792>

<https://zhuanlan.zhihu.com/p/571746996>

<https://zhuanlan.zhihu.com/p/365357615>

<https://zhuanlan.zhihu.com/p/135283598#任务介绍>



# Enjoy your machine learning!

<https://github.com/wjssx/Statistical-Learning-Slides-Code>

E-mail: [csrc\\_dsp@sina.com](mailto:csrc_dsp@sina.com)

Copyright © 2099 [Yjssx](#)

This software released under the [BSD License](#).