Text Recognition: Transformer

第二版

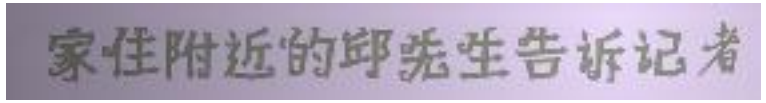
程峰

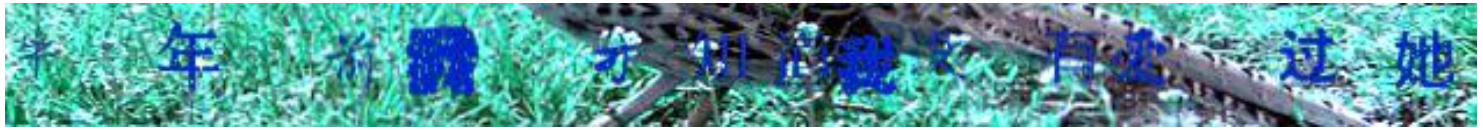
20190510

- 背景介绍
- Sequence to sequence
- Transformer

4Paradigm 第四范式

背景介绍





TEXT-RECOGNITION: 使计算机识别出图片中的文字

背景介绍-相关工作

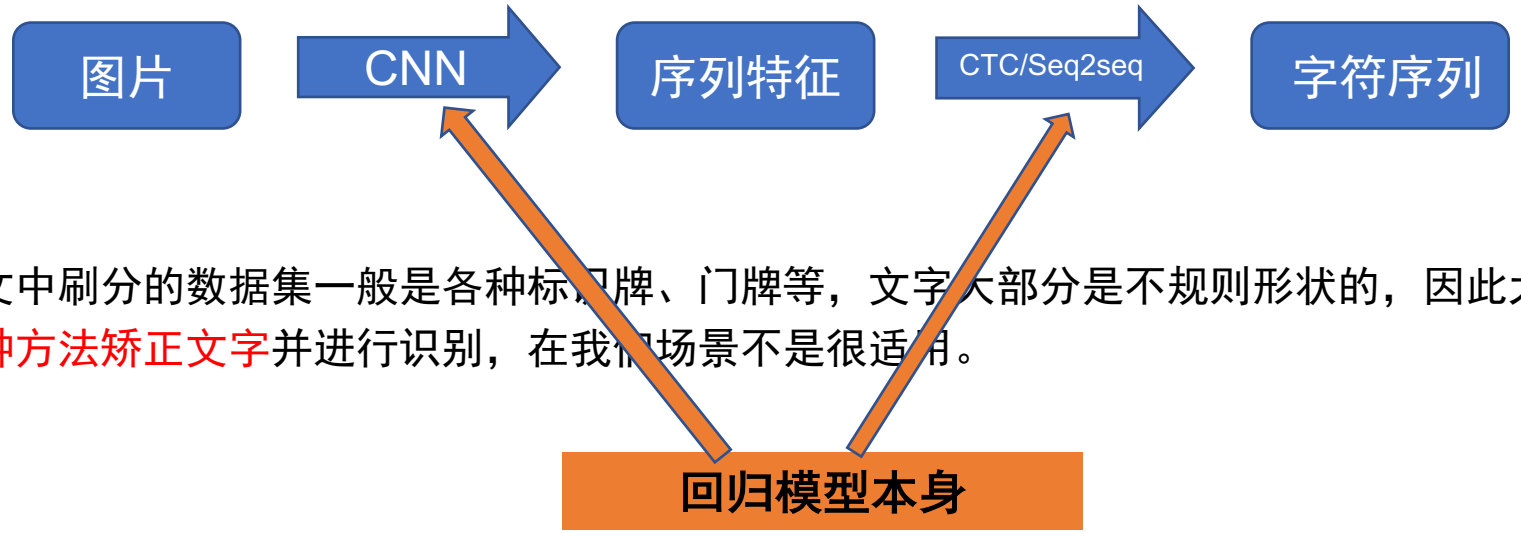
相关论文汇总：

https://github.com/hwalsuklee/awesome-deep-text-detection-recognition

文字识别是一个序列问题，解决方法无外乎两种：

1. CTC: Connectinist Temporal Classification
2. Seq2seq: sequence to sequence.

识别过程:



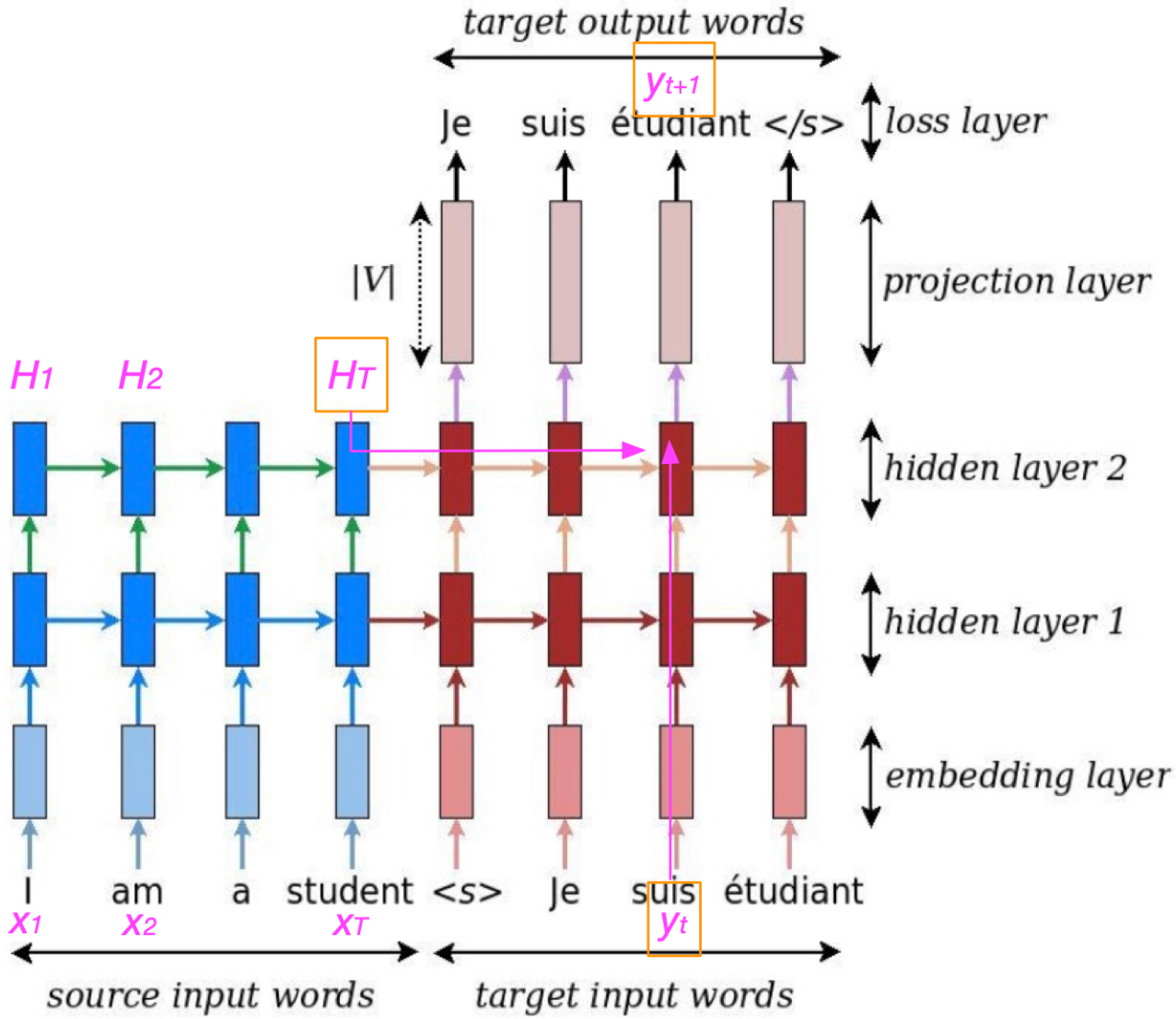
由于论文中刷分的数据集一般是各种标识牌、门牌等，文字大部分是不规则形状的，因此大部分刷分高的论文主要是使用各种方法矫正文字并进行识别，在我们场景不是很适用。

Seq2seq: sequence to sequence

- Encoder
  - 蓝色，对输入特征进行编码
  - 输入X = {$x_1, x_2,\ldots,x_T$}
  - 输出H={$H_1, H_2, \ldots H_T$}
- Decoder
  - 红色，根据上一个预测字符循环解码
  - $y_{t+1} = Decoder([H_T, y_t])$

图中采用多层堆叠的RNN实现Encoder和Decoder.

如何训练？



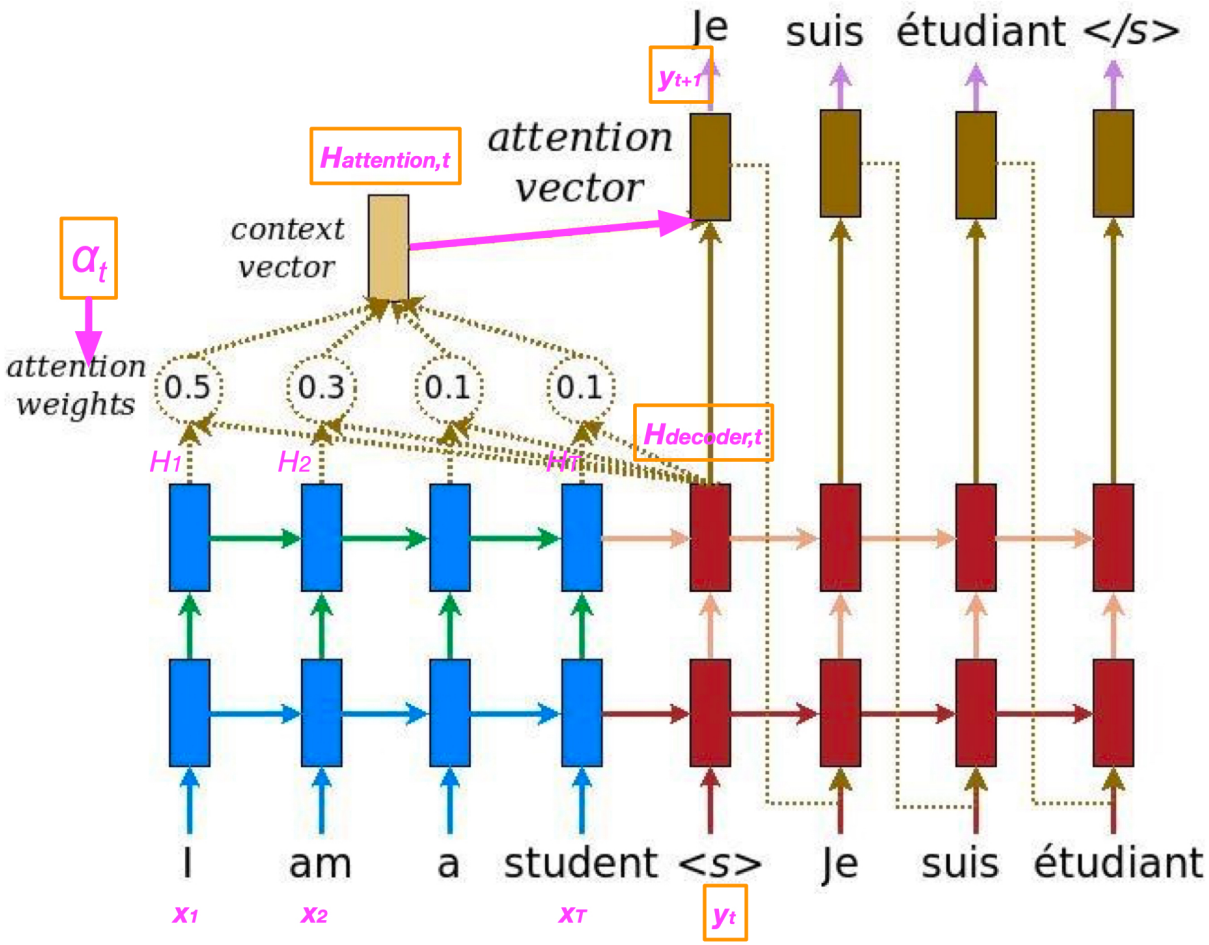
Attention-based Seq2seq

- Encoder
  - 输入X = {$x_1, x_2, \ldots, x_T$}
  - 输出H={$H_1, H_2, \ldots H_T$}

- Decoder
  - 红色，根据上一个预测字符循环解码
  - $y_{t+1} = Decoder([H_{attention,t}, y_t])$

普通Seq2seq，解码时不同时间t都依赖于H_T，随着t的增加前面的信息会失真越来越严重。

引入Attention: 在不同时间t，根据$H_{decoder,t}$与H之间的重要性(相关性)，对H进行重新组合得到新的特征$H_{attention,t}$。



Attention-based Seq2seq

- Decoder
  - 红色，根据上一个预测字符循环解码
  - $y_{t+1} = Decoder([H_{attention,t}, y_t])$

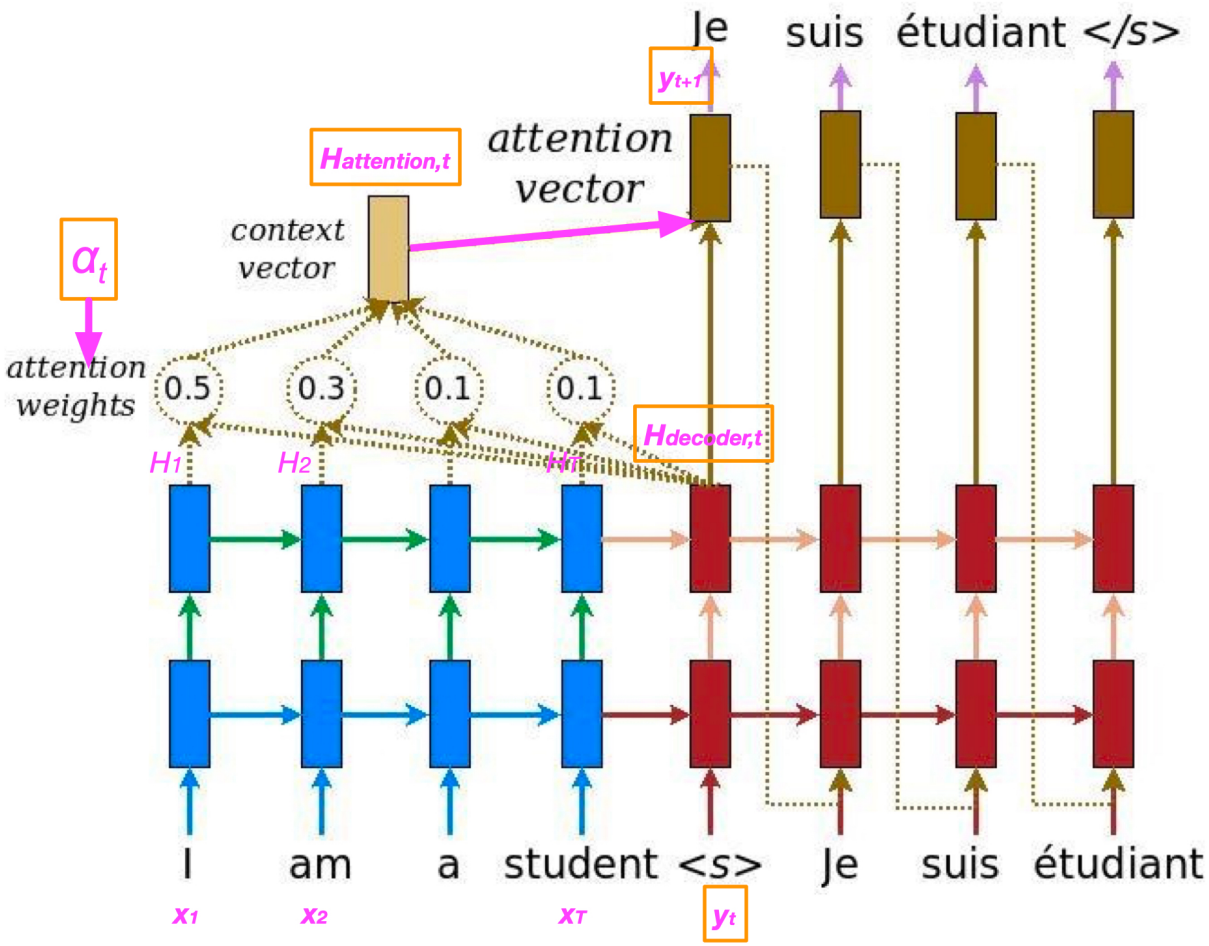
引入Attention: 在不同时间t，根据$H_{decoder,t}$与H之间的重要性(相关性)，对H进行重新组合得到新的特征$H_{attention,t}$。

不同的组合方式就是不同的attention.

如Luong attention：

$$\alpha_t = Softmax((H_{decoder,t})WH)$$

$$H_{attention,t} = reduce\_sum(\alpha_t \odot H)$$

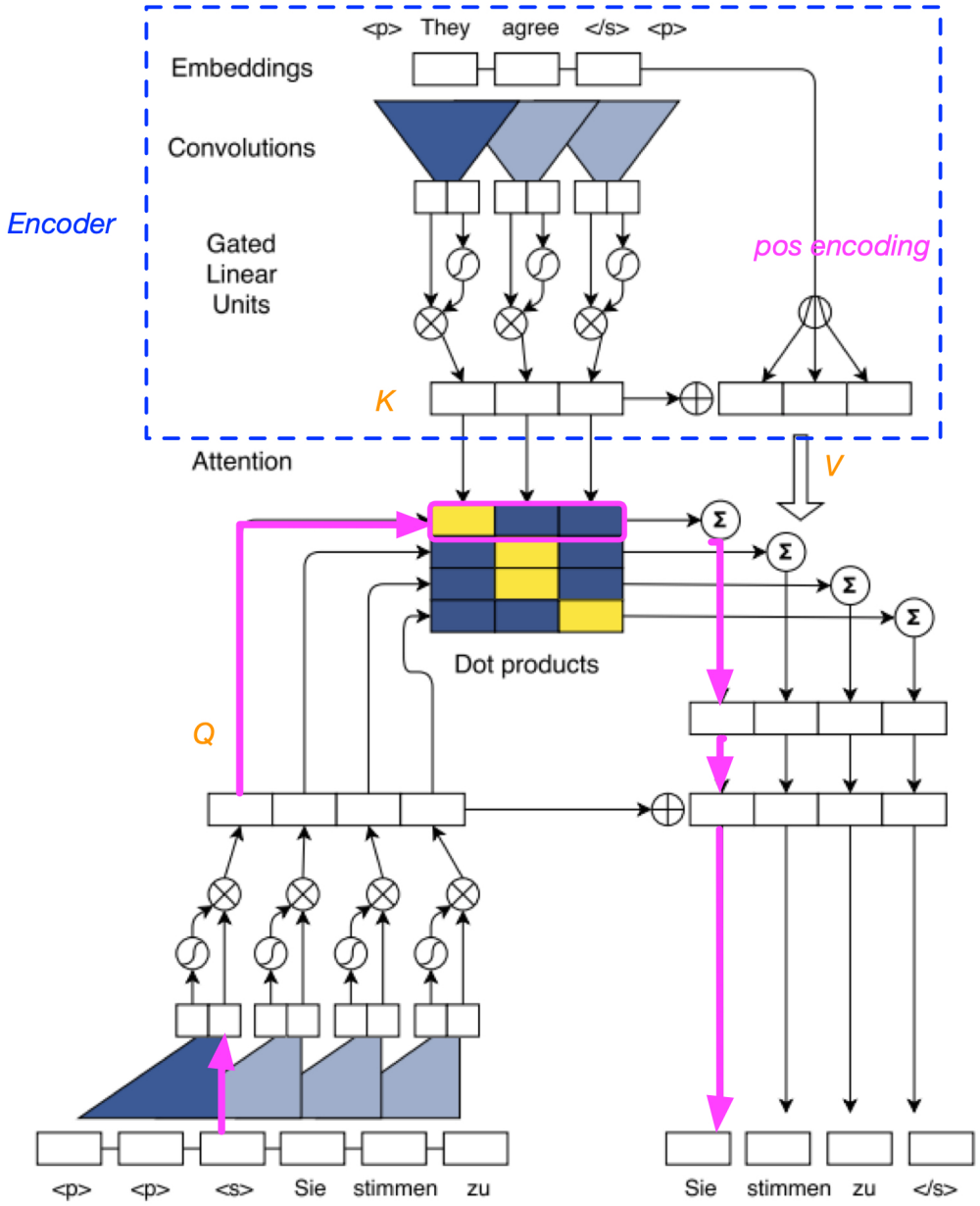


Convolutional Seq2seq

- Paper: Convolutional Sequence to Sequence Learning. Facebook FAIR, ICML2017

  - Q: query
  - K: key
  - V: value

Attention(Q,K,V) = Sum(Coef(Q,K) * V)

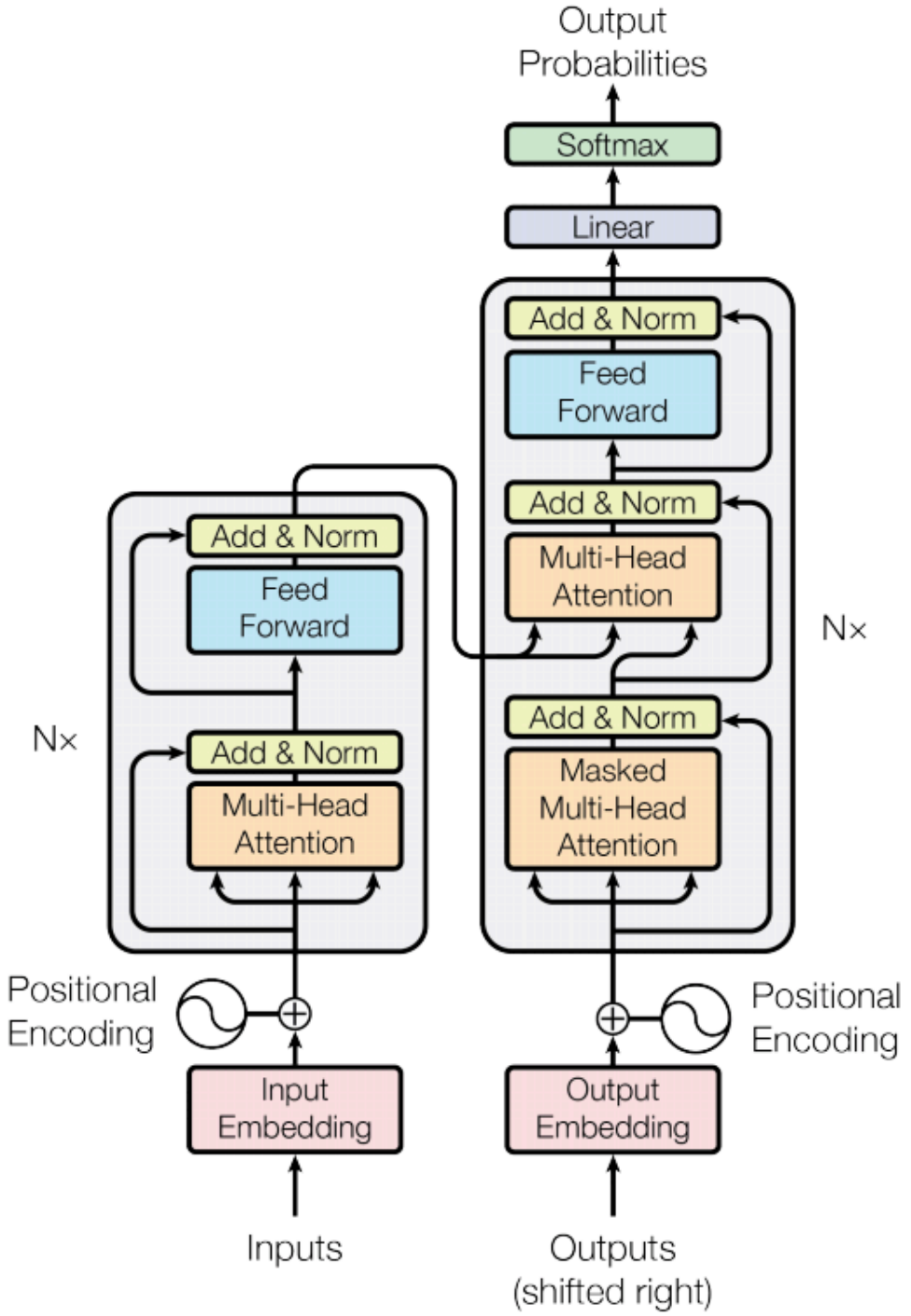


Transformer

Transformer

- Paper: Attention Is All You Need, Google Brain, NIPS 2017

- Key Elements
  - Scaled Dot-Product Attention
  - Multi-Head Attention
  - Feed-Forward
  - Positional Encoding



Transformer

Scaled Dot-Product Attention

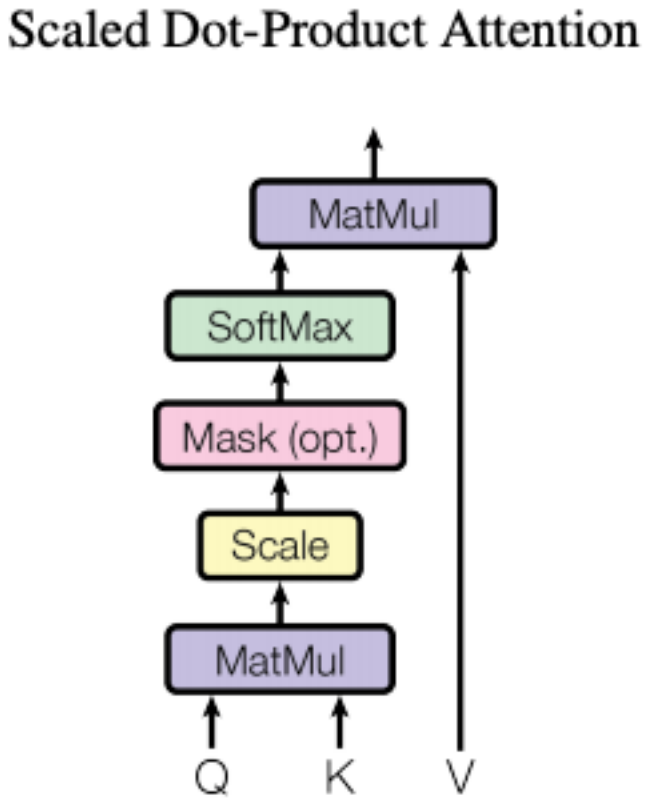
Attention的通用描述:

- 给定一个query(Q), 和key-value对(K, V)
- 计算Q和K之间的相关矩阵
- 利用这个相关矩阵对V进行加权和得到Attention输出

$$Q \in R^{T_o \times C_k}, K \in R^{T_i \times C_k}, V \in R^{T_i \times C_v}$$

$$Score = \frac{QK^T}{\sqrt{C_k}} \in R^{T_o \times T_i}$$

$$Attention(Q, K, V) = Softmax(Score)V \in R^{T_o \times C_v}$$

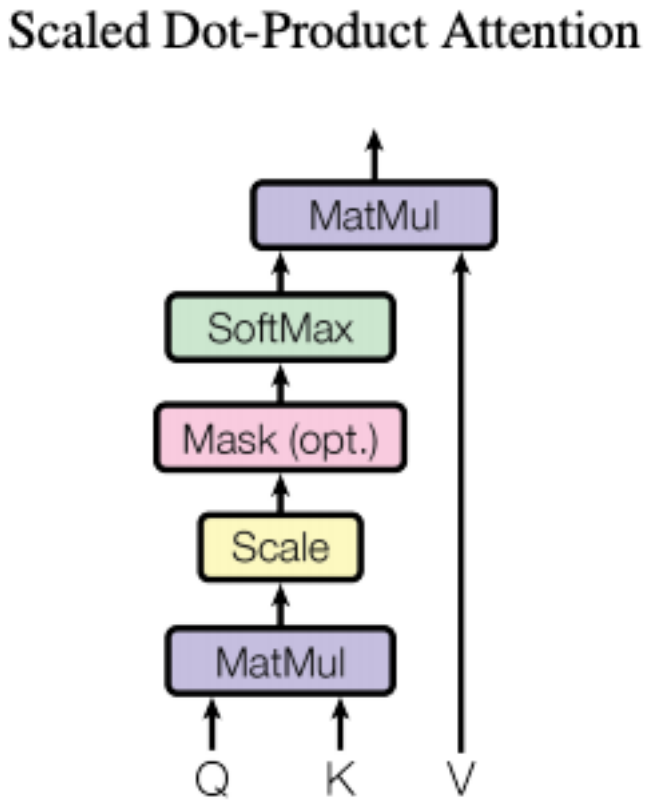


Transformer

Scaled Dot-Product Attention

Attention的通用描述:

- 给定一个query(Q), 和key-value对(K, V)
- 计算Q和K之间的相关矩阵
- 利用这个相关矩阵对V进行加权和得到Attention输出



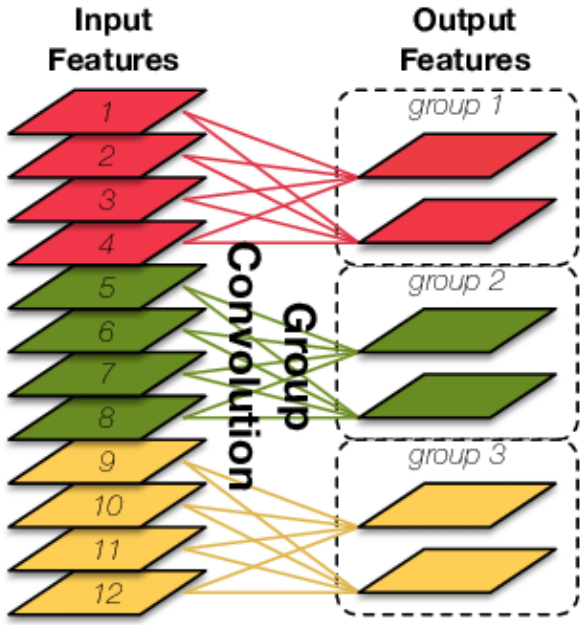
- 当Q=K=V时，这个Attention即为Self-Attention
- 对相关矩阵与下三角矩阵掩码相乘，
  得到Masked Attention (Decoder不能利用未来的信息)

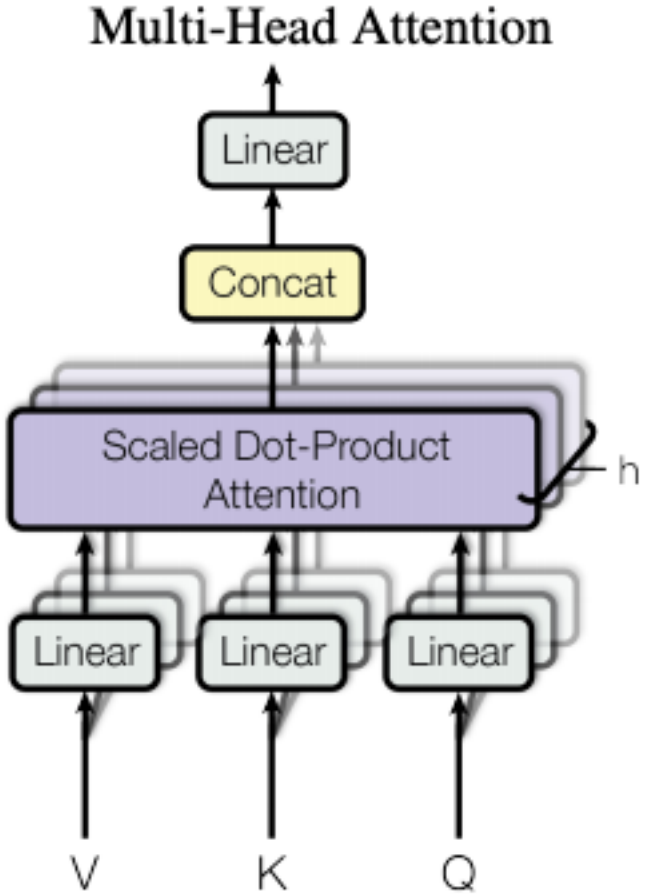
Transformer

Multi-Head Attention

- 将channel维度划分为多个Group
- 每个Group分别进行Attention
- 最后将不同Group计算结果concat

- 类似于Group-Convolution



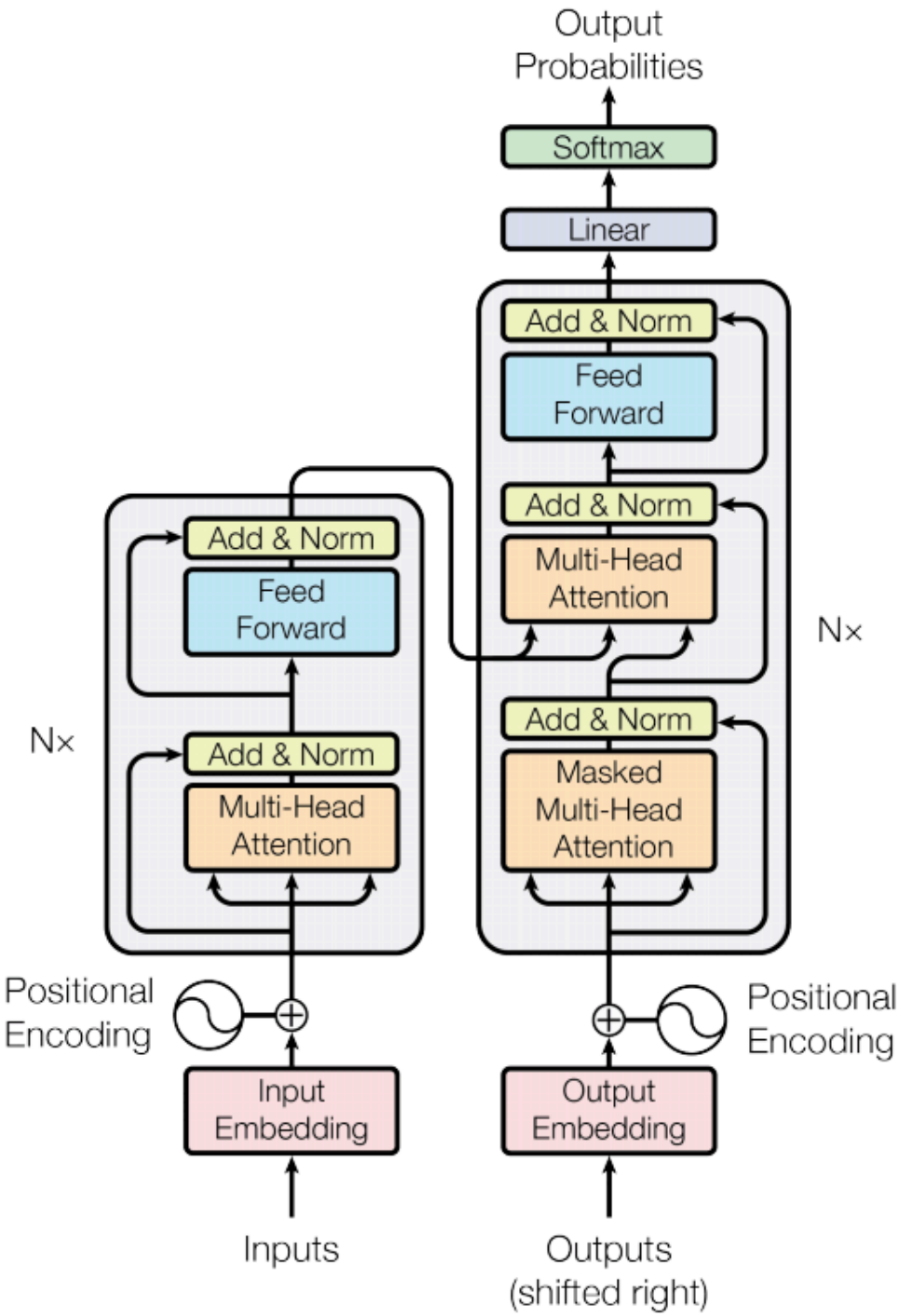


Transformer

Feed-Forward

- 两个全连接层
  - 第一个全连接层将channel维度升高，ReLu激活
  - 第二个全连接层将channel维度还原

如: 512 -> 2048 -> 512
对特征进行非线性变换，相当于进一步提取特征

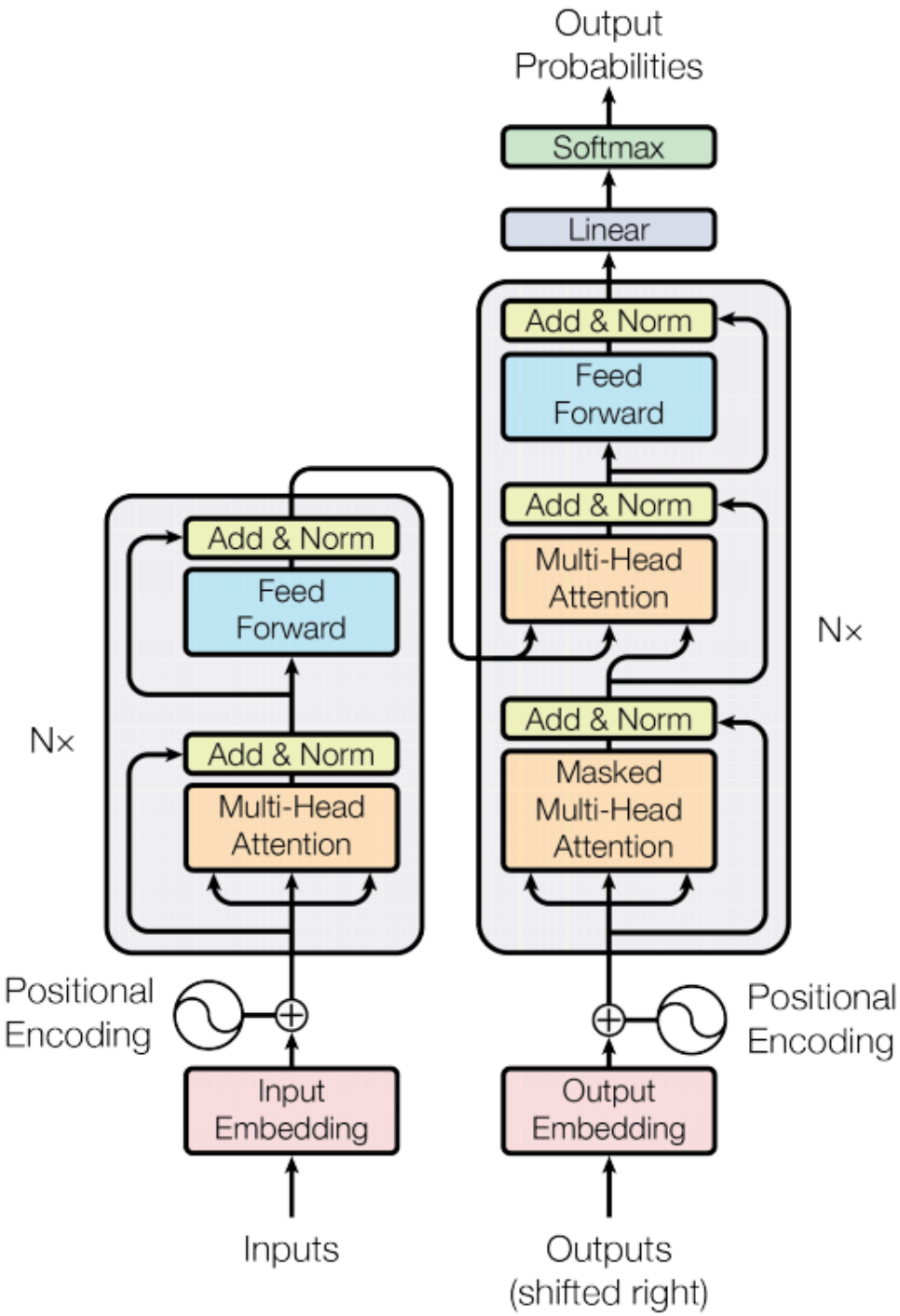


Transformer

Feed-Forward

- 两个全连接层
  - 第一个全连接层将channel维度升高，ReLu激活
  - 第二个全连接层将channel维度还原

如: 512 -> 2048 -> 512
对特征进行非线性变换，相当于进一步提取特征



Transformer

Positional Encoding

- Transformer中特征提取仅仅用了Dense层
- Dense层在时间维度不会有任何交流

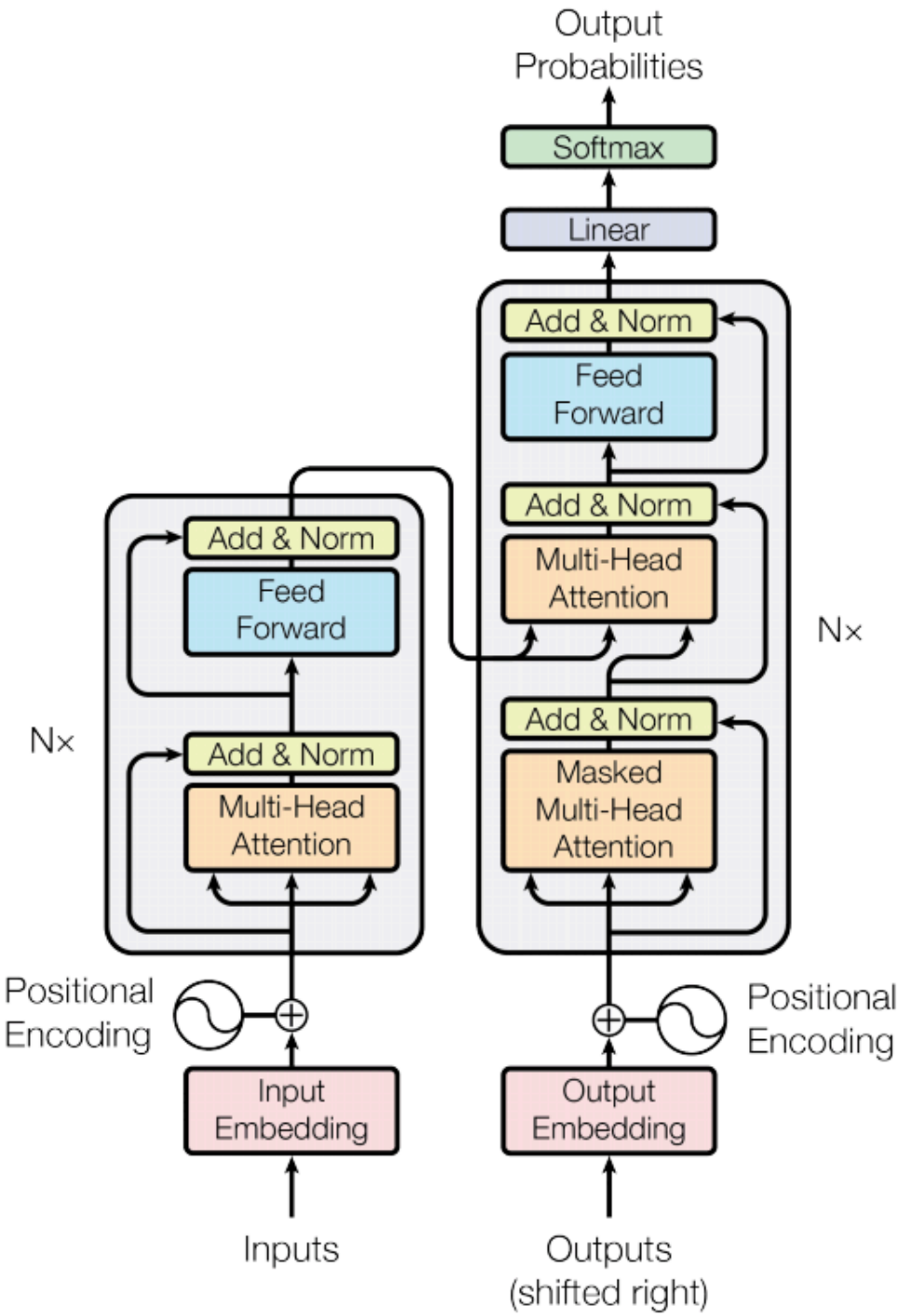
- 加入位置编码之前

$$coef_{ij} = x_i^T x_j = x_j^T x_i$$

  - 即使 x_i 和 x_j 交换位置，两个特征的相关性不会变化，但是句子中单词出现的位置是有重要影响的。

- 加入位置编码之后

$$coef_{ij} = (x_i + pe_i)^T (x_j + pe_j) \neq (x_j + pe_i)^T (x_i + pe_j)$$



Transformer

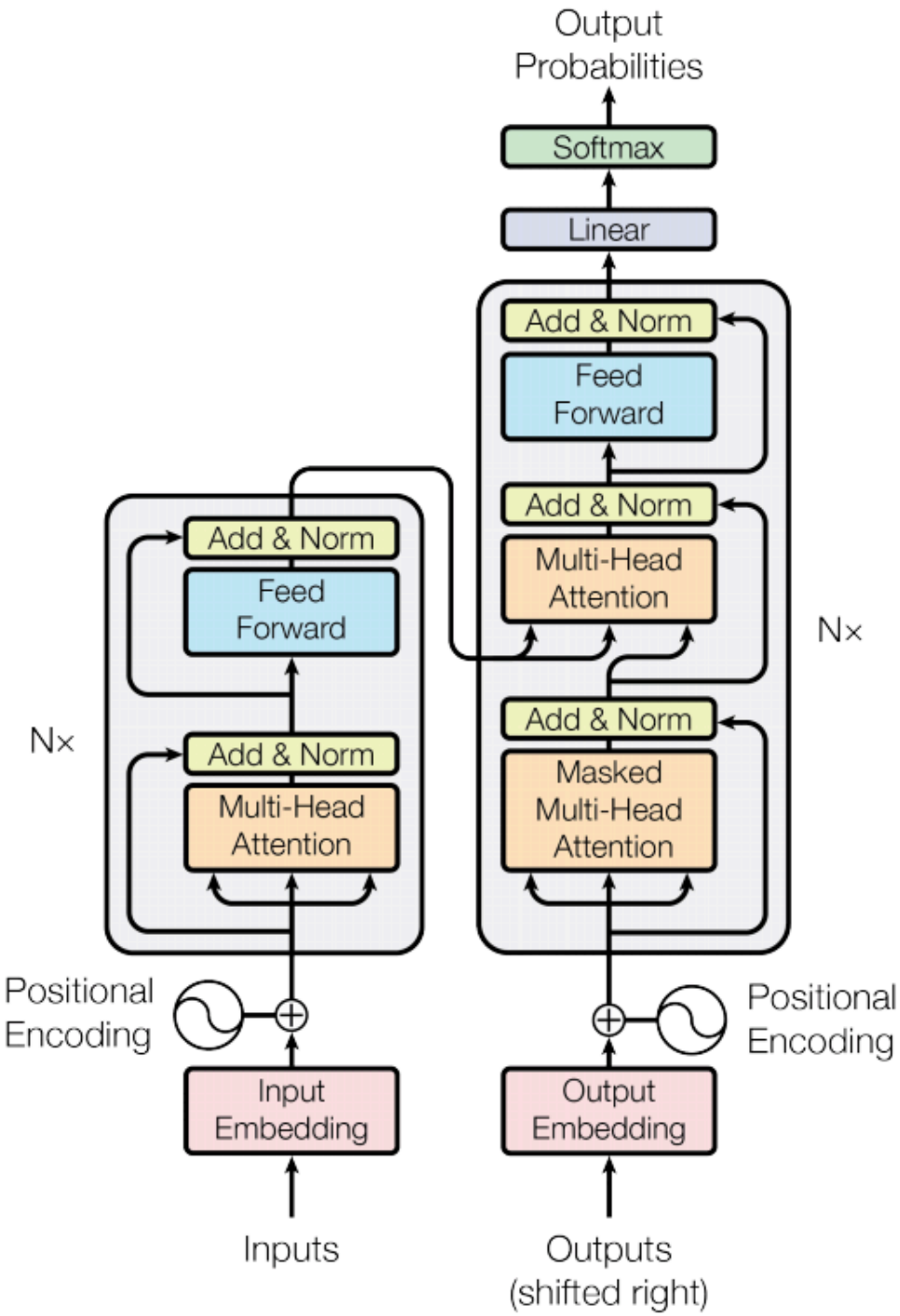
Positional Encoding

- Transformer中特征提取仅仅用了Dense层
- Dense层在时间维度不会有任何交流

位置编码有多种实现方式，有可训练的或者固定的，本文采用：

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

pos为当前的位置，$1 \leq pos \leq T$。$2i, 2i+1$为特征维度($[1, C]$)



Transformer

Why Transformer outperforms others?

↓

Self-Attention?

↓

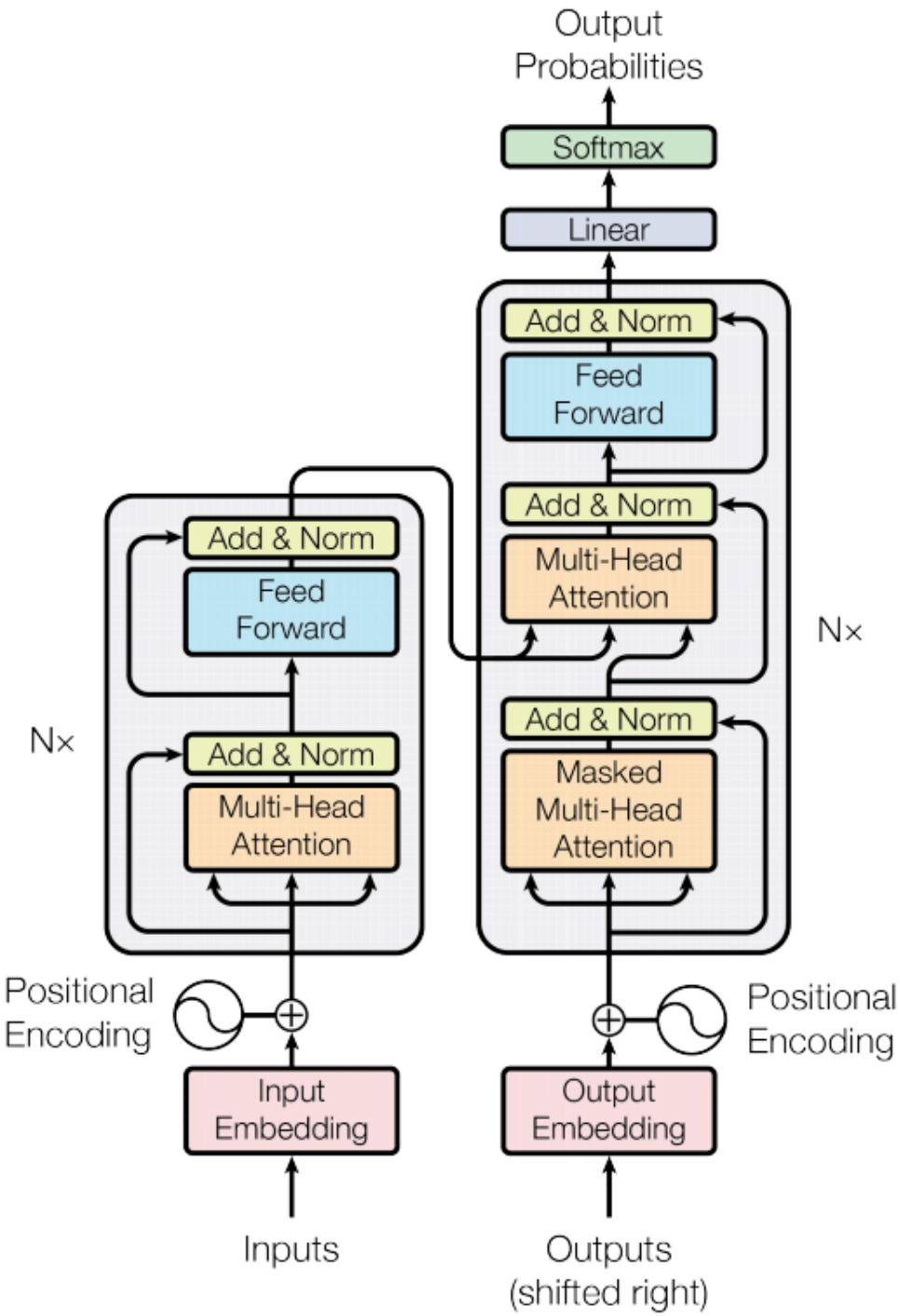
Why Self-Attention?

↓

An Empirical Study of Spatial Attention Mechanisms in Deep Networks, MSRA, Apr 2019.

Self-attention有作用的成份分析

Deformable convolution, Dynamic convolution



Thanks

AI for everyone.

商务咨询
contact@4paradigm.com

TEL
010-8278-0800

媒体合作
pr@4paradigm.com

北京总部
北京市海淀区上地东路35号颐泉汇C座写字楼303室

上海总部
上海市浦东新区浦东南路855号世界广场27楼CD座

深圳总部
深圳市南山区粤海街道高新南九道深圳湾科技生态园二区6栋502A

4Paradigm 第四范式