上海海事大学
SHANGHAI MARITIME UNIVERSITY

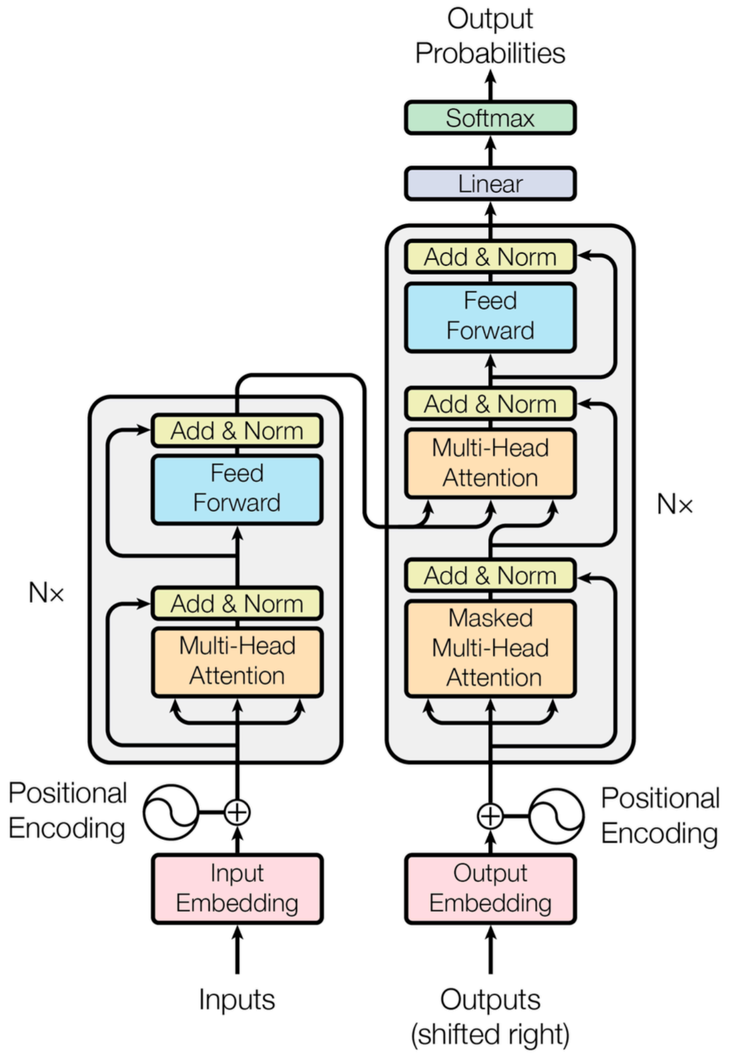
自然语言处理

2024–2025 学年第 2 学期

信息工程学院　谢雨波

Transformer

- 预训练大语言模型的基础
- 基于**自注意力机制（Self-Attention Mechanism）**
- 编码器-解码器结构
- 相较于 RNN，可以并行化运算

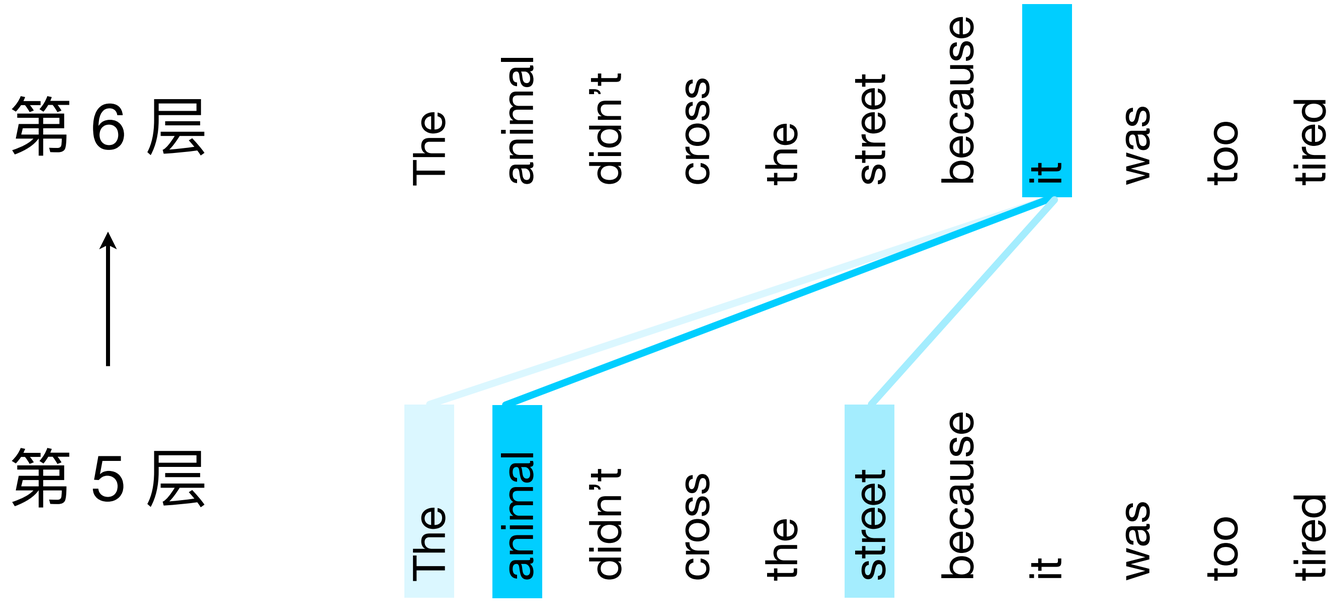


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. NeurIPS 2017.

自注意力机制

自注意力机制

- 通过网络层数的增加，建立输入单词的**上下文表示（Contextualized Representation）**
- 对于每一个单词，基于下层网络的输出，融合其他所有单词位置的信息



自注意力机制

- 将输入的词向量 $x_i \in \mathbb{R}^{1\times d}$ 表示为三个新的向量：
  - **查询（Query）**向量：$q_i = x_i W^Q$
  - **键（Key）**向量：$k_i = x_i W^K$
  - **值（Value）**向量：$v_i = x_i W^V$
- 其中权重矩阵

$$W^Q \in \mathbb{R}^{d\times d_k} \quad W^K \in \mathbb{R}^{d\times d_k} \quad W^V \in \mathbb{R}^{d\times d_v}$$

自注意力机制

- 计算当前词 x_i 与上下文词 x_j 的分数

$$\text{score}\left(x_i, x_j\right) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$

较大的 d_k 将导致 softmax 函数输出 0 或 1，进而导致梯度为极小值

- 对分数进行归一化

$$\alpha_{ij} = \text{softmax}\left(\text{score}\left(x_i, x_j\right)\right) = \frac{\exp\left(\text{score}\left(x_i, x_j\right)\right)}{\sum_k \exp\left(\text{score}\left(x_i, x_k\right)\right)}$$

- 输出的上下文表示

$$a_i = \sum_j \alpha_{ij} v_j$$

自注意力机制

6. 求和

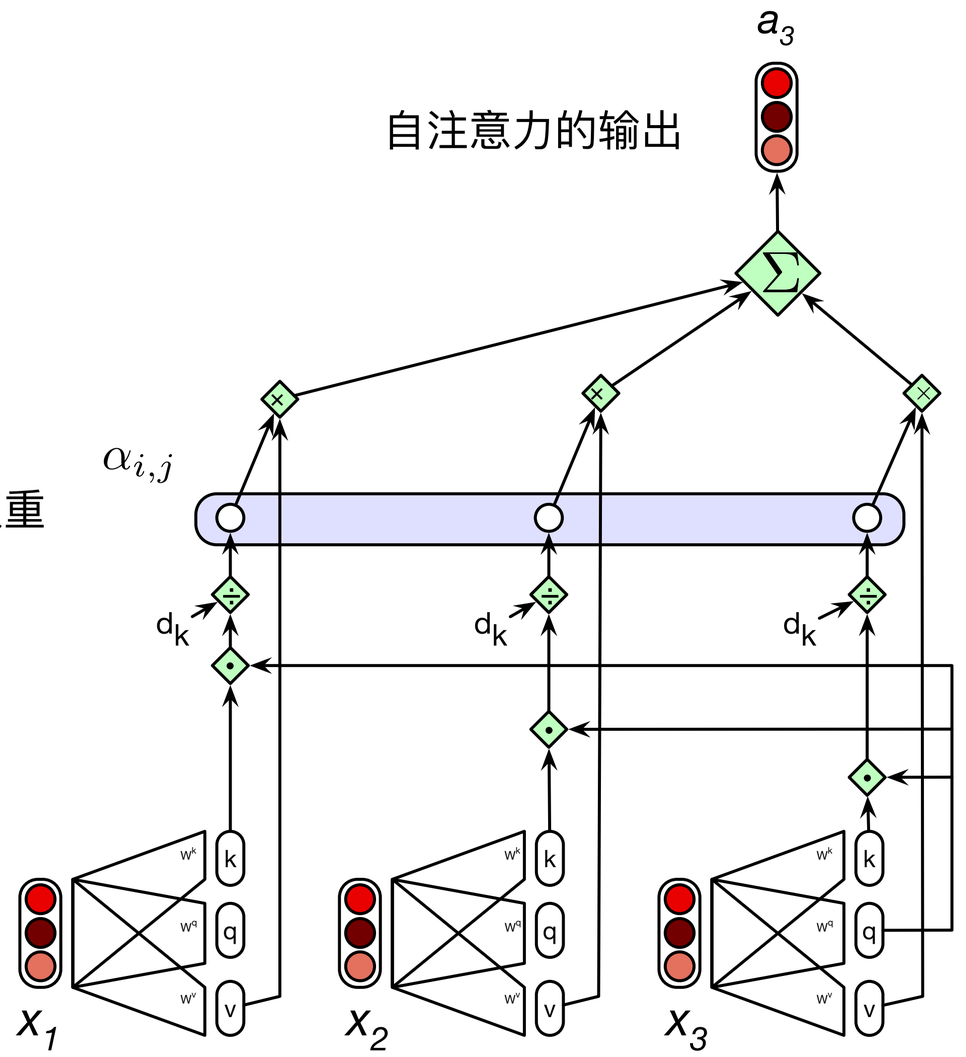
5. 对 v_1, v_2, v_3 进行加权

4. 使用 softmax 转换为权重

3. 分数除以 $\sqrt{d_k}$

2. 将 q_3 与 k_1, k_2, k_3 比较

1. 生成 q, k, v 向量



自注意力机制

- 输入序列为 N 个词：令 $X \in \mathbb{R}^{N \times d}$
- 乘上权重矩阵

$$Q = XW^Q \quad K = XW^K \quad V = XW^V$$

$$Q \in \mathbb{R}^{N \times d_k} \quad K \in \mathbb{R}^{N \times d_k} \quad V \in \mathbb{R}^{N \times d_v}$$

- 自注意力的输出

$$A = \text{SelfAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad \in \mathbb{R}^{N \times d_v}$$

多头注意力机制

- 词与词之间的关系是多样的：语法、语义、语句上的关系同时存在
- 使用**多头注意力机制（Multi-Head Attention Mechanism）**在多个表示子空间（Representation Subspace）中并行处理信息
- 对每一个头（Head）i，有独立权重

$$W_i^Q \in \mathbb{R}^{d \times d_k} \quad W_i^K \in \mathbb{R}^{d \times d_k} \quad W_i^V \in \mathbb{R}^{d \times d_v}$$

多头注意力机制

- 对每一个头（Head）i

$$Q = XW_i^Q \quad K = XW_i^K \quad V = XW_i^V$$

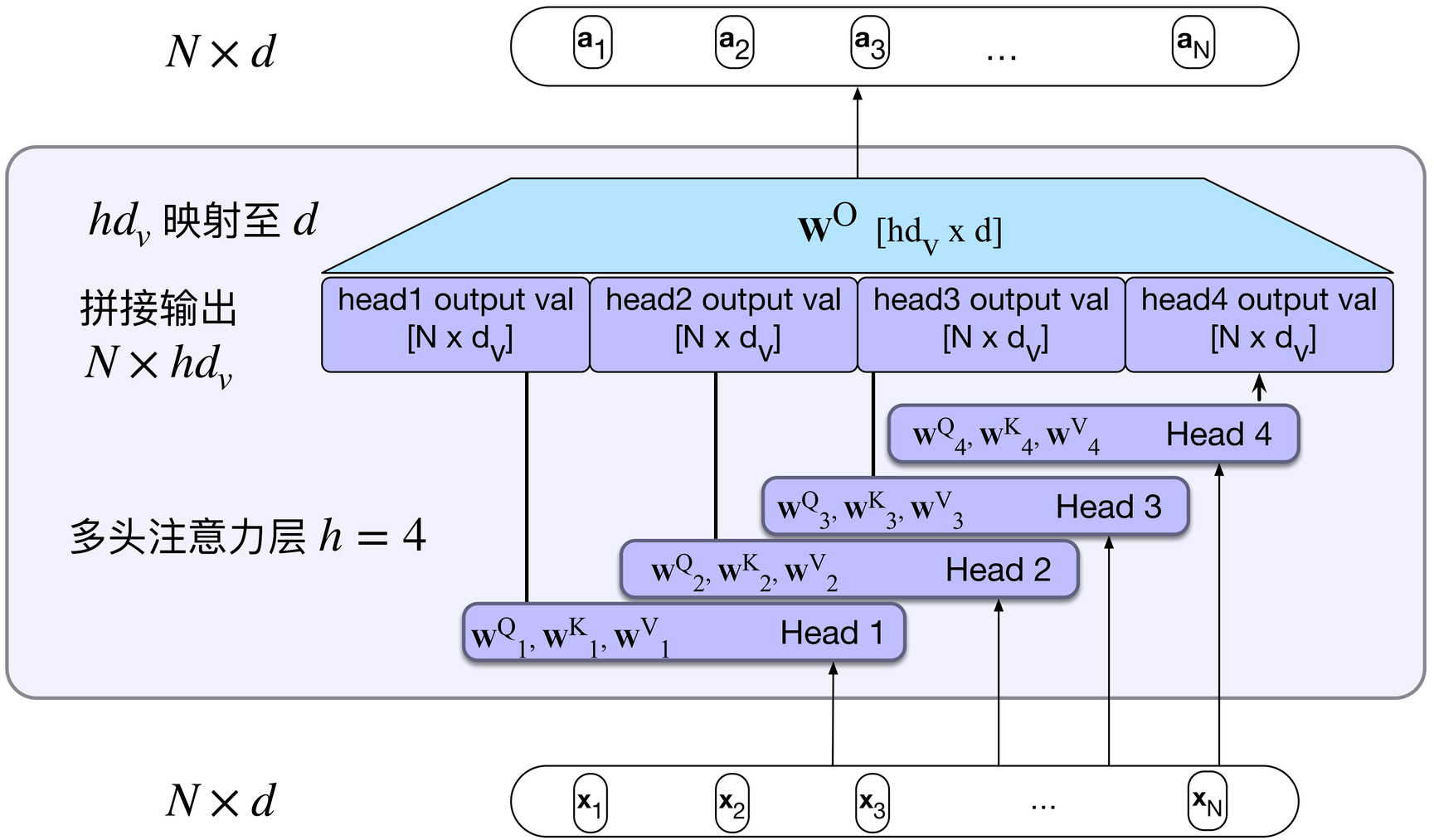
$$\text{head}_i = \text{SelfAttention}(Q, K, V) \quad \in \mathbb{R}^{N \times d_v}$$

- 将所有 h 个 head_i **拼接**起来，并且通过线性变换 $W^O \in \mathbb{R}^{hd_v \times d}$

$$A = \text{MultiHeadAttention}(X)$$

$$= \left(\text{head}_1 \oplus \text{head}_2 \oplus \cdots \oplus \text{head}_h\right) W^O \quad \in \mathbb{R}^{N \times d}$$

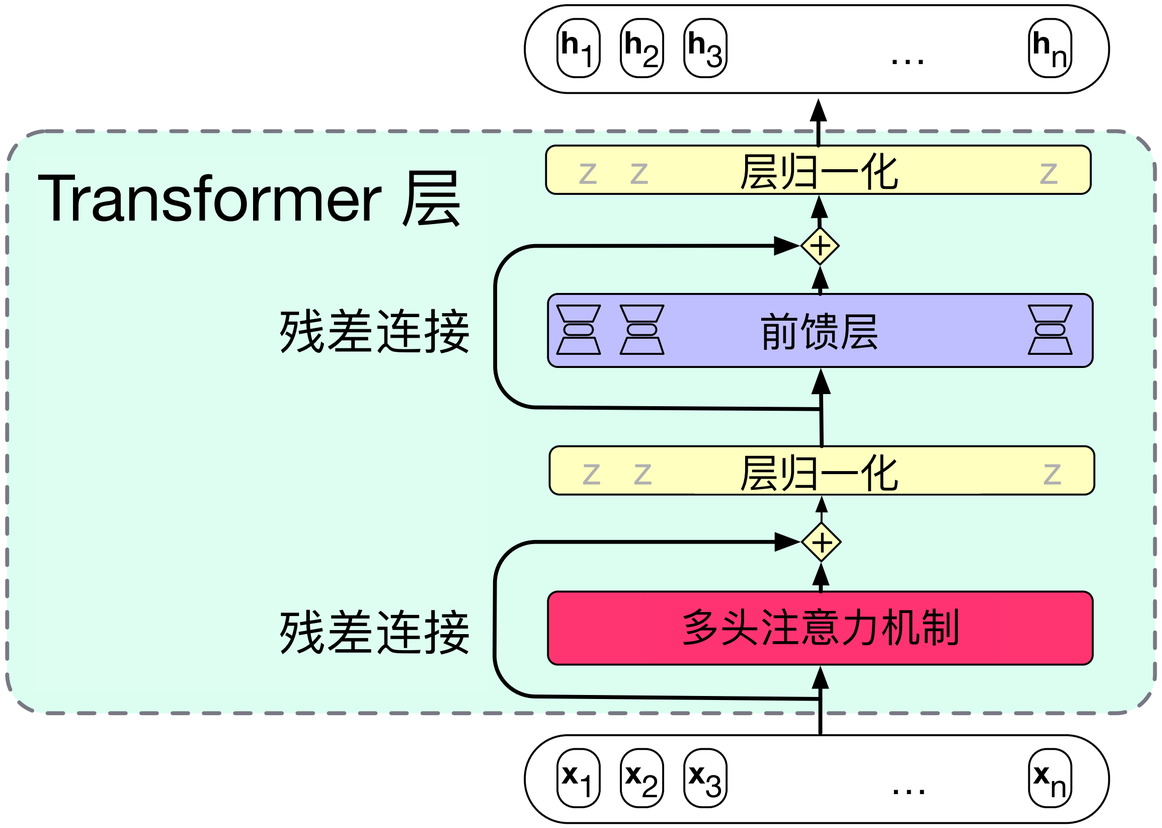
多头注意力机制


$N \times d$
a1 a2 a3 … aN
hd_v 映射至 d
$\mathbf{W}^O$ [hdV x d]
拼接输出
$N \times hd_v$
head1 output val [N x dV]
head2 output val [N x dV]
head3 output val [N x dV]
head4 output val [N x dV]
$\mathbf{w}^Q_4, \mathbf{w}^K_4, \mathbf{w}^V_4$ Head 4
$\mathbf{w}^Q_3, \mathbf{w}^K_3, \mathbf{w}^V_3$ Head 3
$\mathbf{w}^Q_2, \mathbf{w}^K_2, \mathbf{w}^V_2$ Head 2
$\mathbf{w}^Q_1, \mathbf{w}^K_1, \mathbf{w}^V_1$ Head 1
多头注意力层 $h = 4$
$N \times d$
x1 x2 x3 … xN

Transformer 层（编码器）

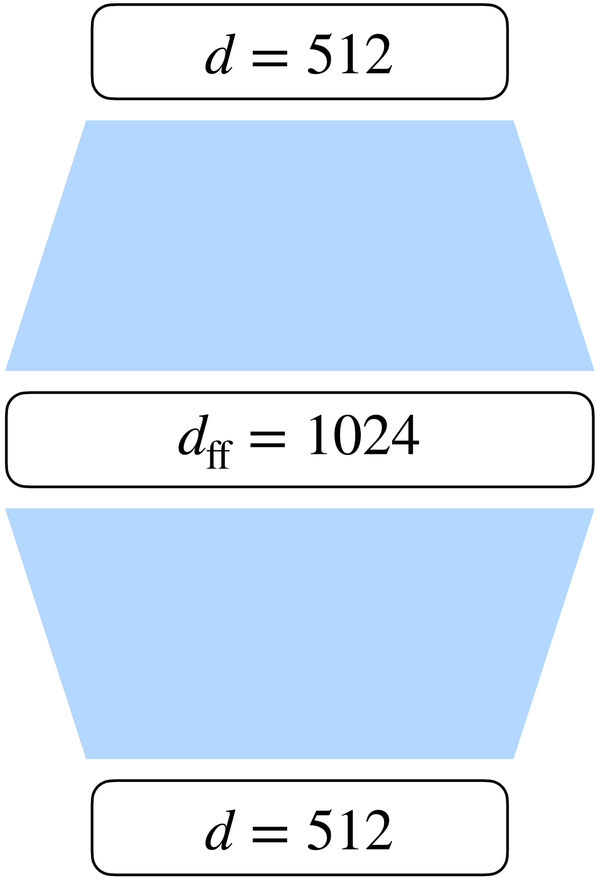
Transformer 层（编码器）

- 编码器中，一个 Transformer 层由四部分组成：
  - 多头注意力机制
  - 前馈层（Feedforward Layer）
  - 残差连接（Residual Connection）
  - 层归一化（Layer Normalization）



前馈层（Feedforward Layer）

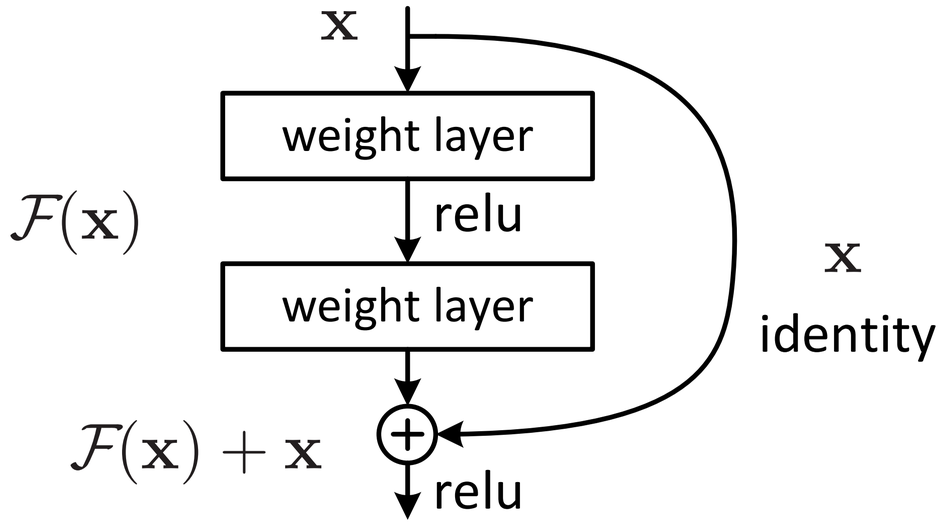
- N 个独立的 2 层前馈神经网络
  - 应用于每个输入位置（Position-wise）
- 同一 Transformer 层的 N 个网络参数共享
- 不同 Transformer 层的前馈层参数不同



残差连接（Residual Connection）

- 解决网络层数增加但训练误差也增大的反直觉问题
- 增加一个残差连接
  - 学习残差表示 $F(x) := H(x) - x$
  - 相比较直接学习 $H(x)$ 要更容易

例如，假设最优的 $H(x)$ 为恒等映射，即 $H(x) = x$，直接通过多层前馈网络学习较为困难，但是通过残差连接，可以容易地学习到 $F(x) = 0$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. CVPR 2016.

层归一化（Layer Normalization）

Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton. Layer Normalization. NeurIPS Workshop 2016.

- 将隐藏层的值保持在某一范围内，使得梯度更新更加稳定，从而提高深度神经网络的训练效率
- 首先，计算输入向量 x 元素的均值和标准差：

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i$$

$$\sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2}$$

层归一化（Layer Normalization）

Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton.
Layer Normalization. NeurIPS Workshop 2016.

- 然后，对 x 进行归一化（得到均值为 0 标准差为 1 的向量）：

$$\hat{x} = \frac{x - \mu}{\sigma}$$

- 最后，结合可学习的参数 γ 和 β 得到归一化结果：

$$\text{LayerNorm} = \gamma\hat{x} + \beta$$

Gain Offset

增加模型的**灵活性**：
允许模型在归一化后调整激活值的缩放和偏移，
有助于恢复归一化可能损失的表达能力

Transformer 层（编码器）

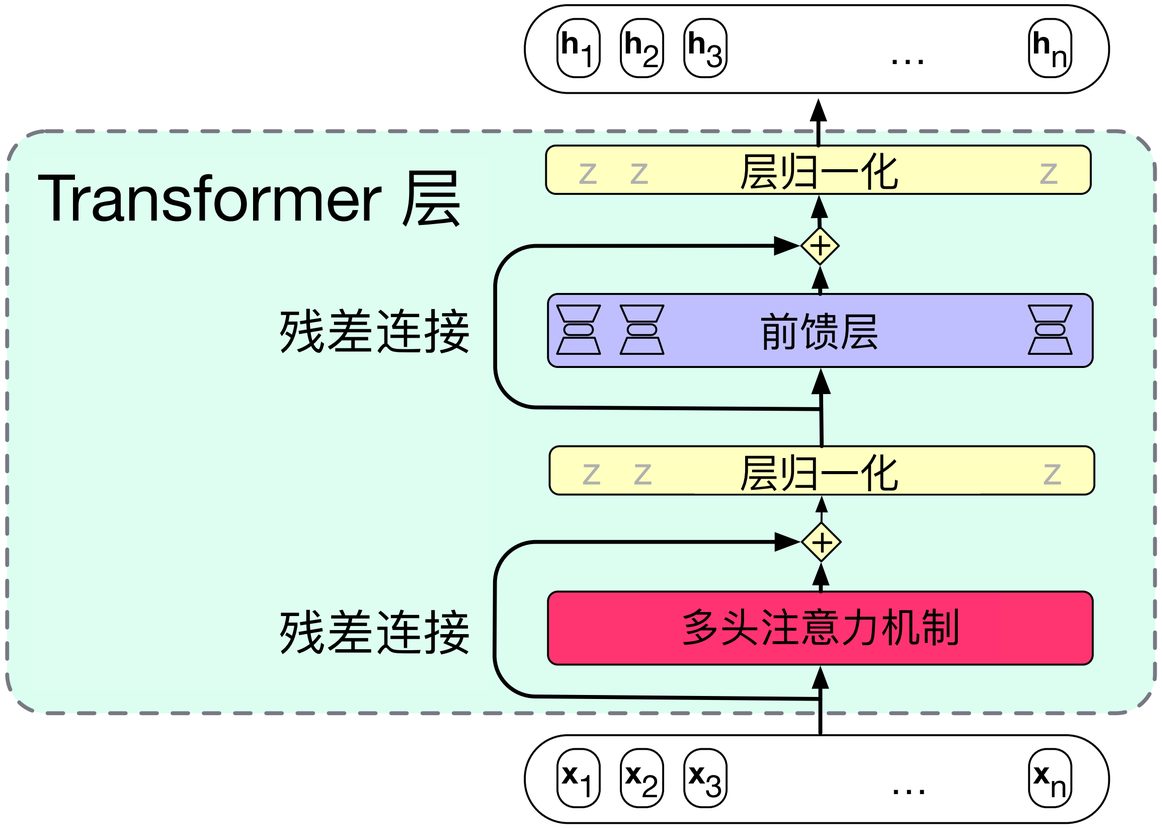
$$O = \mathrm{LN}(X + \mathrm{MHA}(X))$$

$$H = \mathrm{LN}(O + \mathrm{FFN}(O))$$

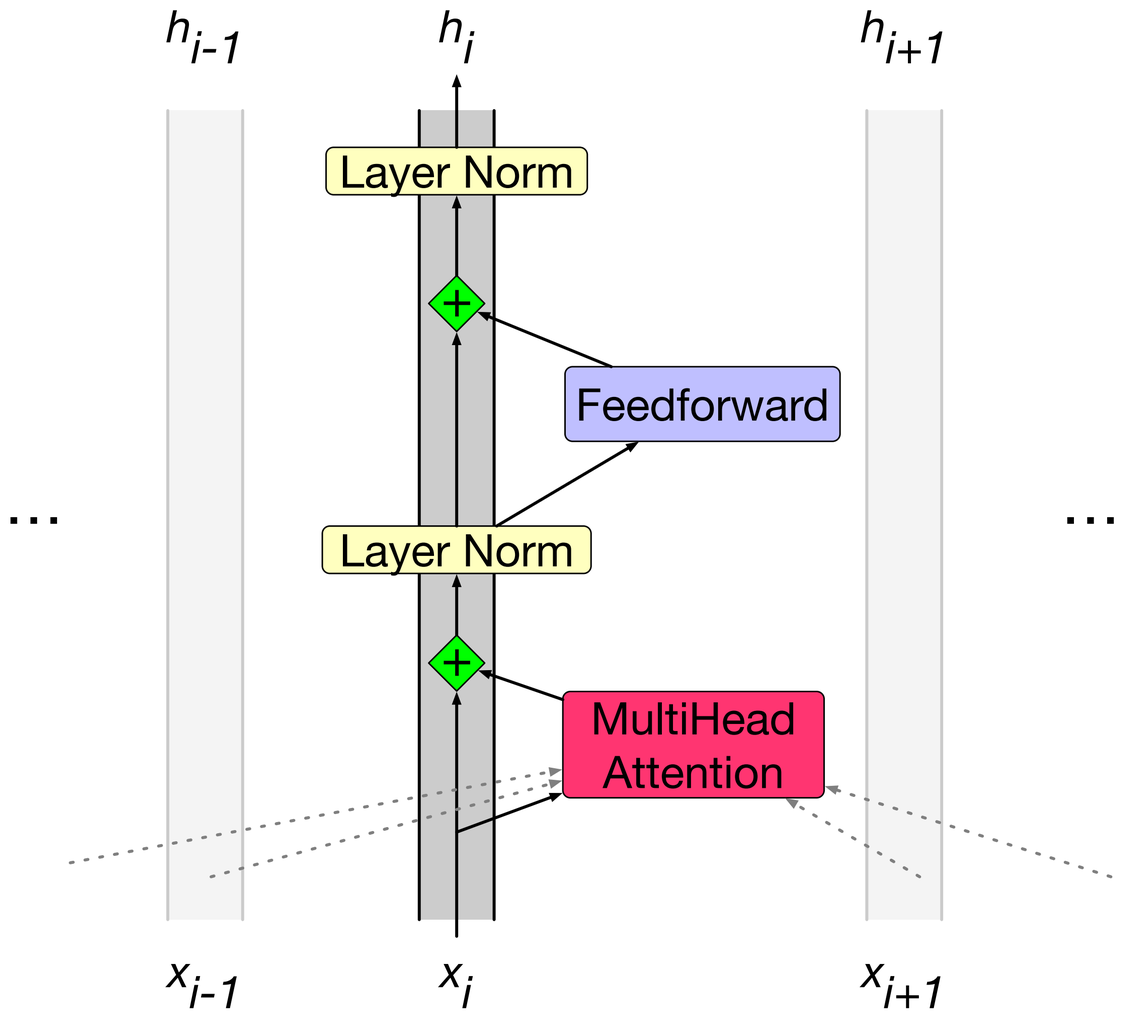
LN: Layer Normalization

MHA: Multi-Head Attention

FFN: Feedforward Network



层归一化的位置

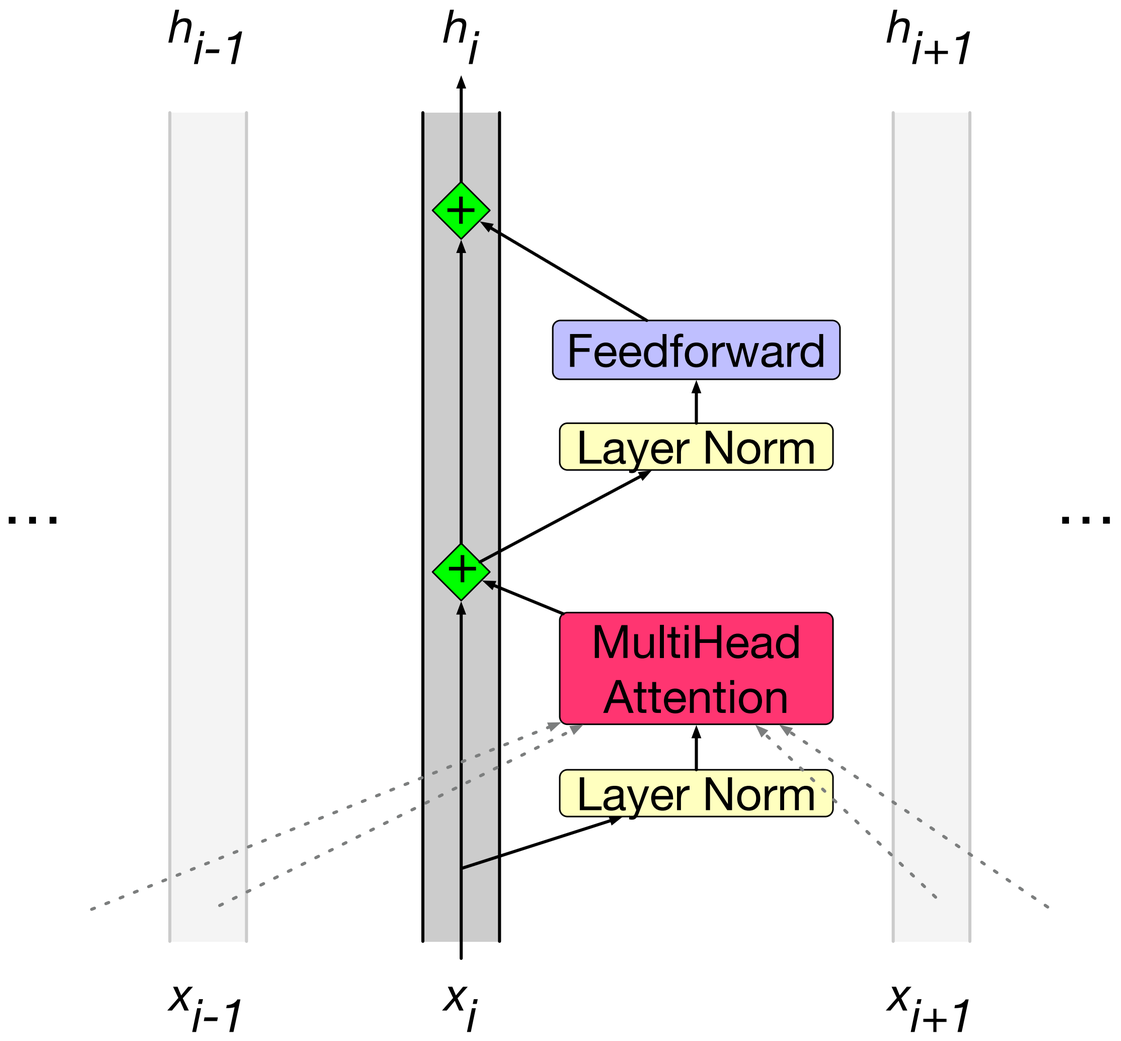


Post-Norm：层归一化在残差连接之后

Transformer 原论文

（Attention Is All You Need）

层归一化的位置



Pre-Norm：层归一化在注意力和前馈层之前

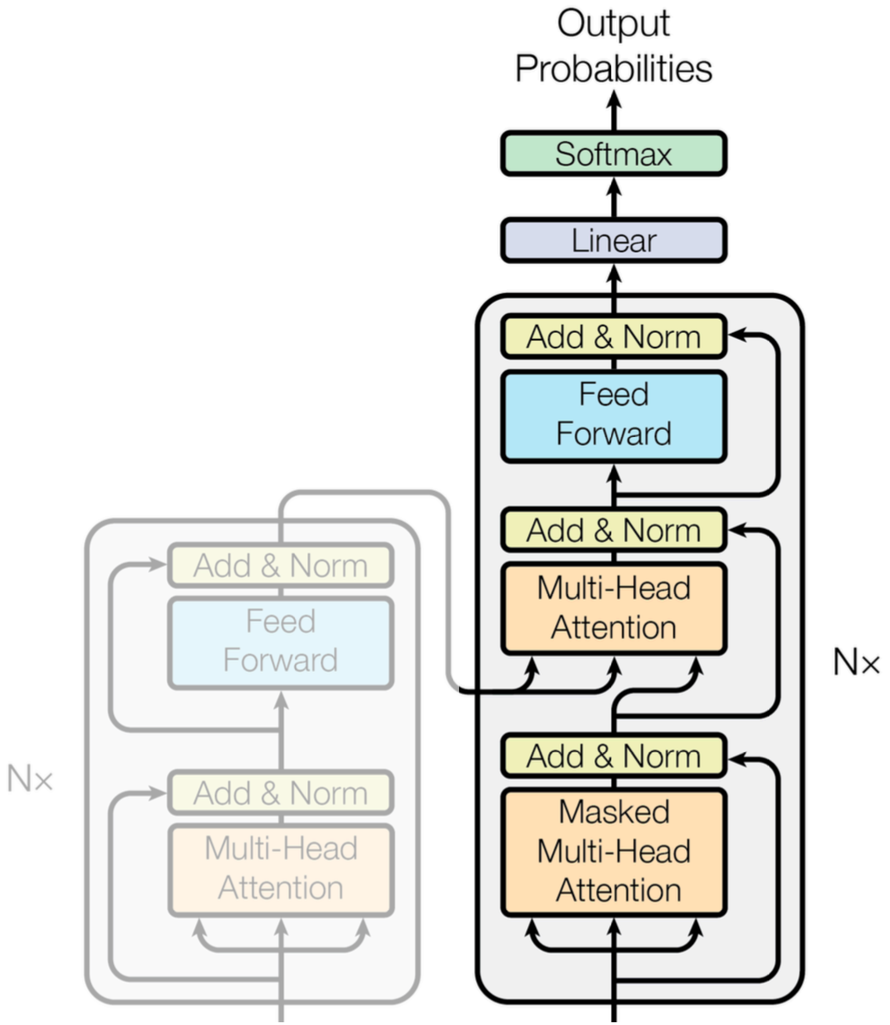
更容易训练，主流大模型以 Pre-Norm 居多

需要在最后一个 Transformer 层的最后加上一个额外的层归一化

Transformer 层（解码器）

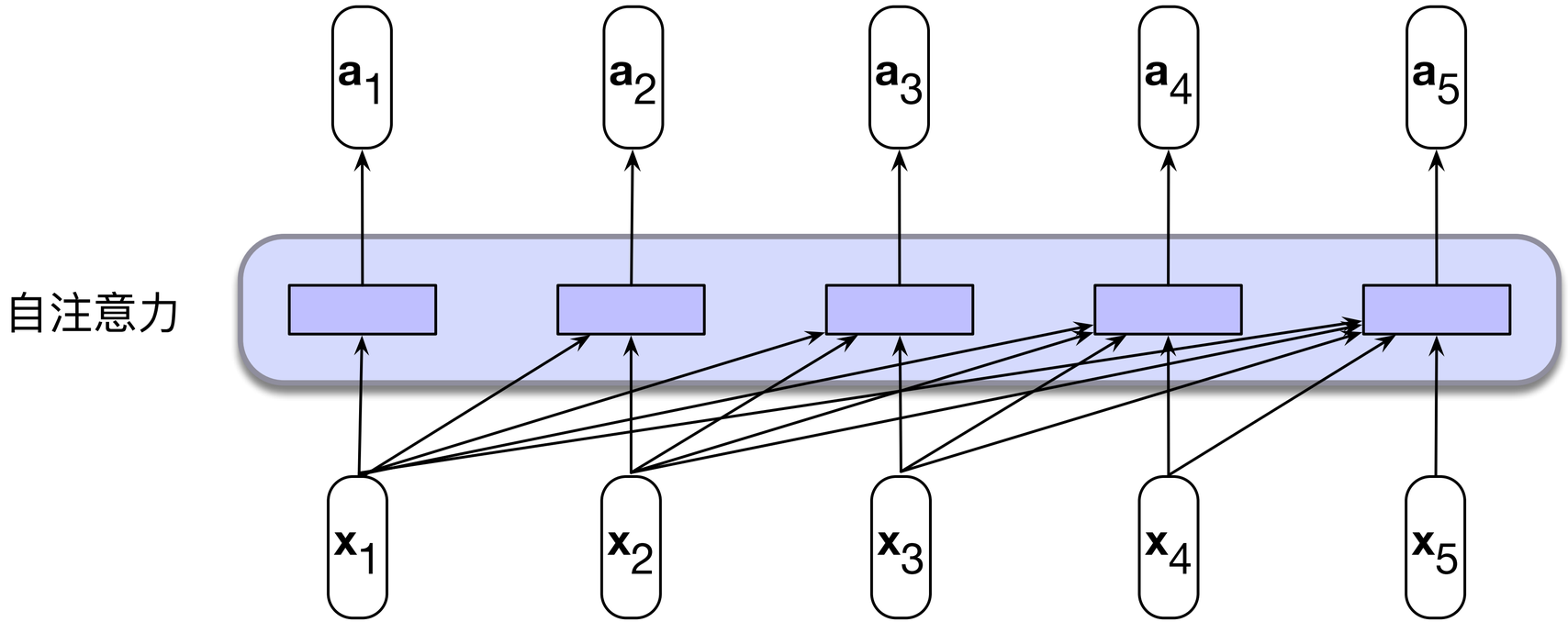
Transformer 层（解码器）

- 解码器中，一个 Transformer 层由五部分组成：
  - **掩码多头注意力（Masked Multi-Head Attention）**
  - **编码器-解码器多头注意力（Encoder-Decoder Multi-Head Attention）**
  - 前馈层（Feedforward Layer）
  - 残差连接（Residual Connection）
  - 层归一化（Layer Normalization）



掩码多头注意力

- 解码器的自注意力机制中，每一个单词只能获取它前面单词的信息，而不能获取它后面单词的信息



掩码多头注意力

- 计算完分数后，直接令分数矩阵的右上三角部分为 $-\infty$

N

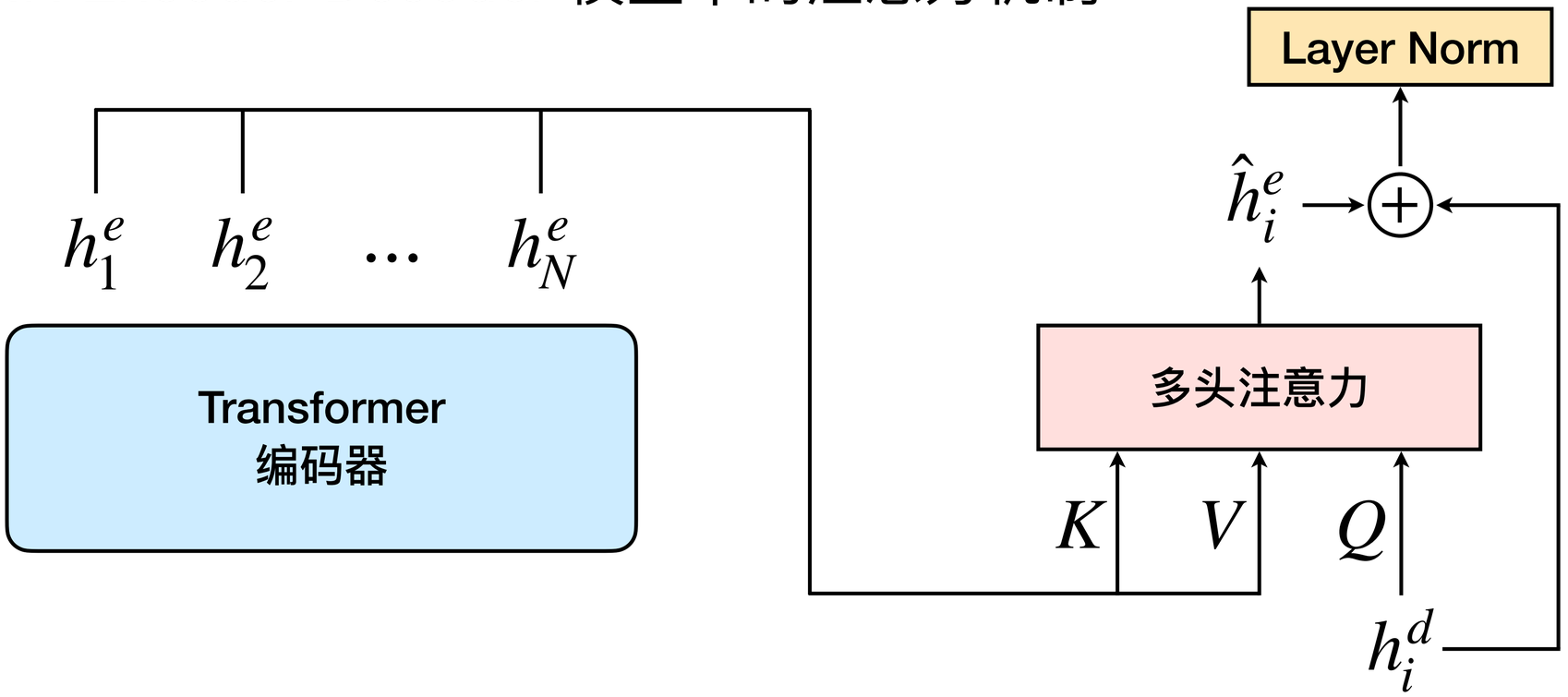
$q_1 \cdot k_1$	$q_1 \cdot k_2$	$q_1 \cdot k_3$	$q_1 \cdot k_4$	$q_1 \cdot k_5$
$q_2 \cdot k_1$	$q_2 \cdot k_2$	$q_2 \cdot k_3$	$q_2 \cdot k_4$	$q_2 \cdot k_5$
$q_3 \cdot k_1$	$q_3 \cdot k_2$	$q_3 \cdot k_3$	$q_3 \cdot k_4$	$q_3 \cdot k_5$
$q_4 \cdot k_1$	$q_4 \cdot k_2$	$q_4 \cdot k_3$	$q_4 \cdot k_4$	$q_4 \cdot k_5$
$q_5 \cdot k_1$	$q_5 \cdot k_2$	$q_5 \cdot k_3$	$q_5 \cdot k_4$	$q_5 \cdot k_5$

N

$q_1 \cdot k_1$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$q_2 \cdot k_1$	$q_2 \cdot k_2$	$-\infty$	$-\infty$	$-\infty$
$q_3 \cdot k_1$	$q_3 \cdot k_2$	$q_3 \cdot k_3$	$-\infty$	$-\infty$
$q_4 \cdot k_1$	$q_4 \cdot k_2$	$q_4 \cdot k_3$	$q_4 \cdot k_4$	$-\infty$
$q_5 \cdot k_1$	$q_5 \cdot k_2$	$q_5 \cdot k_3$	$q_5 \cdot k_4$	$q_5 \cdot k_5$

编码器-解码器多头注意力

- 同时，解码器还需要从编码器的输出中获取信息
- 类似于 RNN Encoder-Decoder 模型中的注意力机制



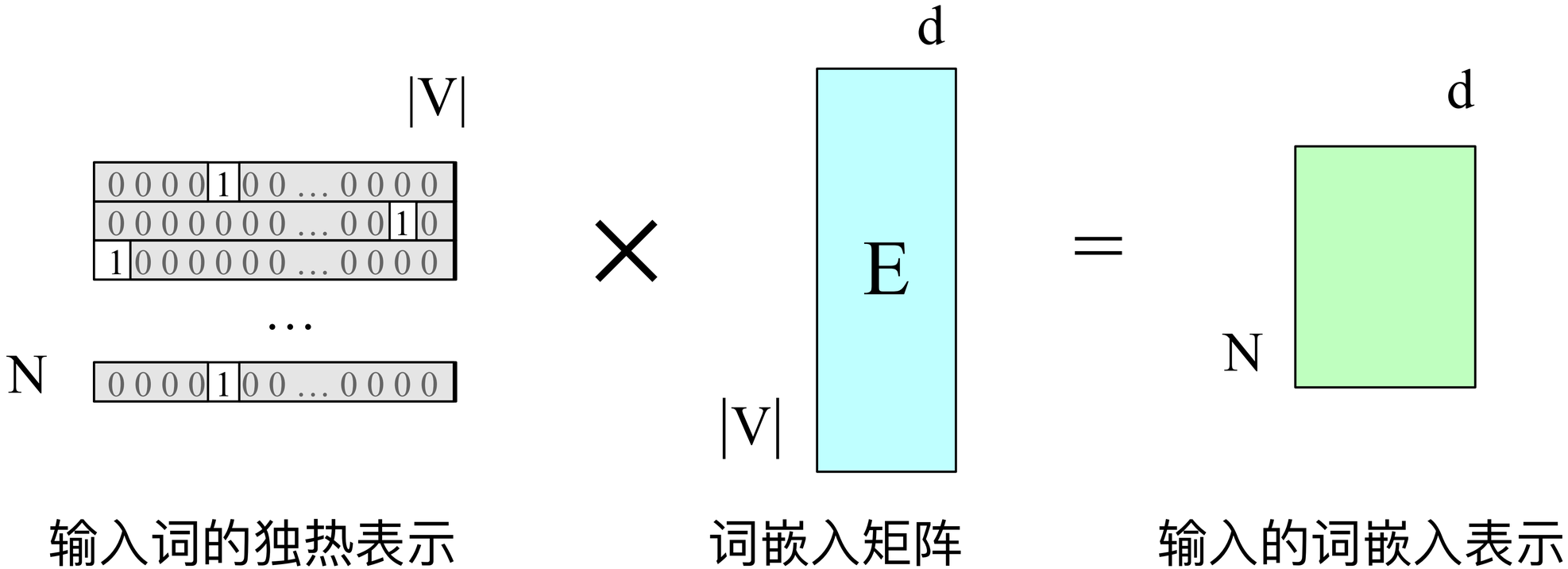
嵌入层

嵌入层

- Transformer 的嵌入层分为两个部分：
  - **词嵌入（Word Embedding）**
  - **位置嵌入（Positional Embedding）**
    - 不同于 RNN 的顺序读入，Transformer 一次性读入所有输入
    - 因此需要加入额外的位置信息

词嵌入

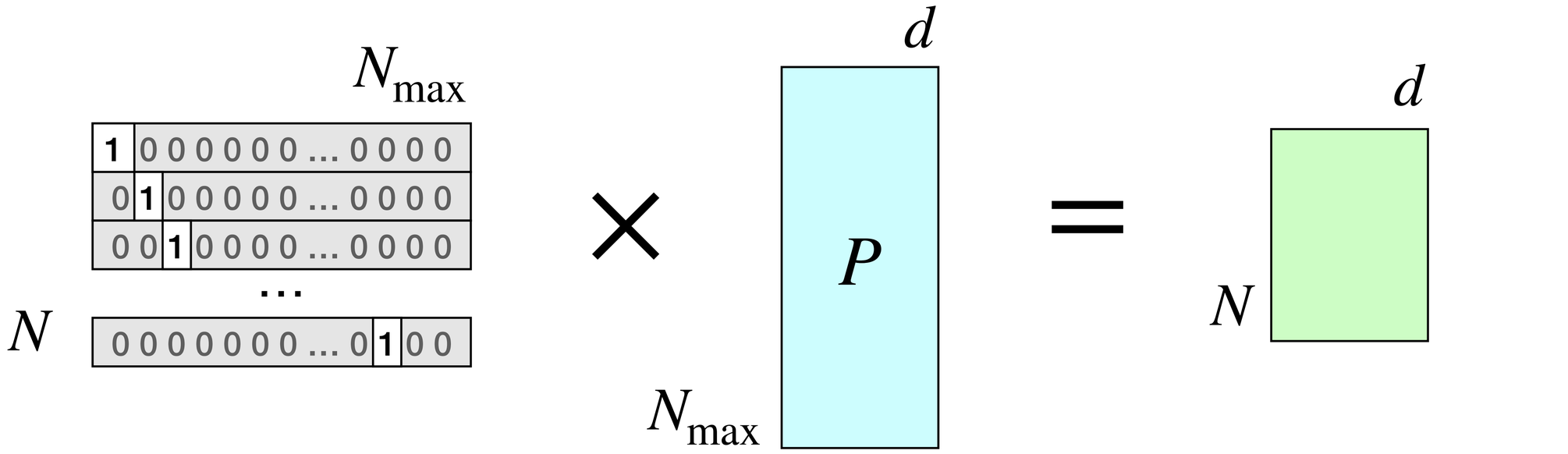
- 和 RNN 模型相同，将代表词的**独热向量（One-hot Vector）**与**词嵌入矩阵** E（模型参数的一部分）相乘



输入词的独热表示　　词嵌入矩阵　　输入的词嵌入表示

位置嵌入

- **绝对位置嵌入（Absolute Positional Embedding）**
  - **训练式**：和词嵌入类似，定义一个**位置嵌入矩阵** P（模型参数的一部分）



输入位置的独热表示　　位置嵌入矩阵　　输入的位置嵌入表示

位置嵌入

- 训练式位置嵌入：嵌入矩阵靠近底部的行可能训练不充分（相较于顶部的行）
- **静态位置嵌入（Static Positional Embedding）**
  - 手动挑选固定的位置嵌入（不可训练）
  - 例如，Transformer 原论文提出的三角式（Sinusoidal）位置嵌入：

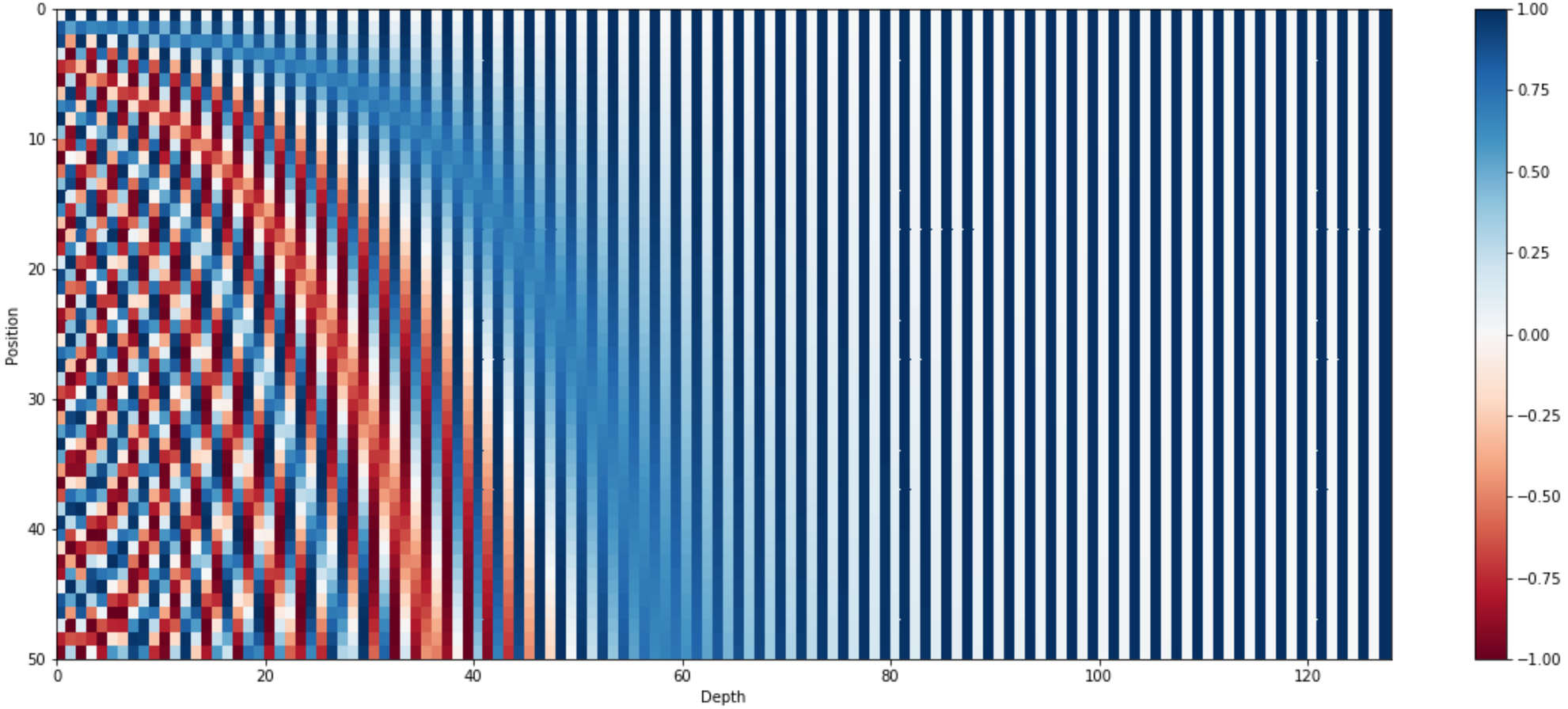
$$PE(\text{pos}, 2i) = \sin\left(\text{pos} \Big/ 10000^{2i/d}\right)$$

$$PE(\text{pos}, 2i+1) = \cos\left(\text{pos} \Big/ 10000^{2i/d}\right)$$

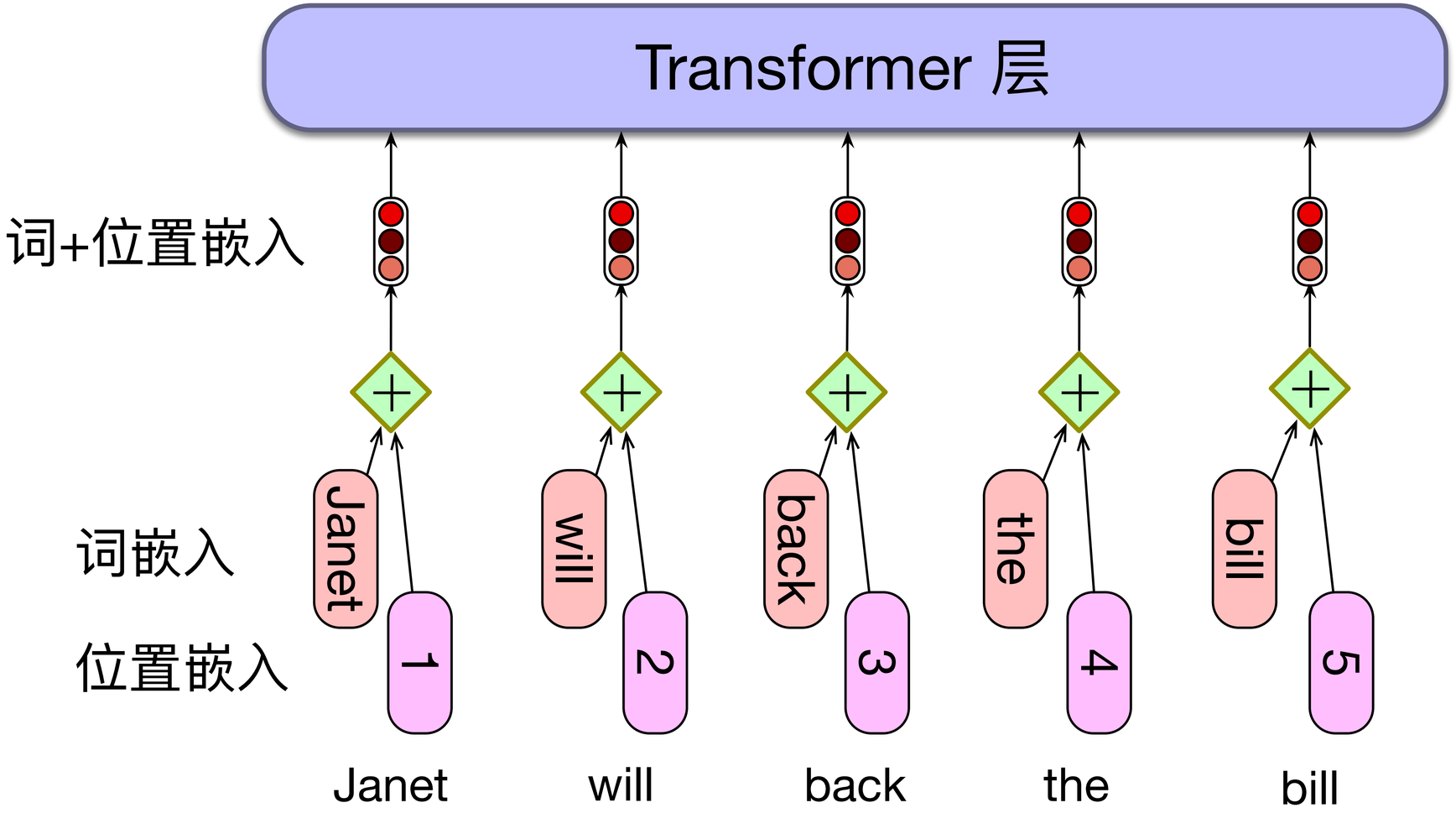
  - 其中，pos 为位置，i 为维度

位置嵌入

- 静态三角式位置嵌入（每一行对应输入的一个位置）：



嵌入层


Transformer 层
词+位置嵌入
词嵌入
位置嵌入
Janet
will
back
the
bill
1
2
3
4
5
Janet
will
back
the
bill

其他位置嵌入

- 绝对位置嵌入的缺点：**外推能力差**（训练上下文长度短，但推理上下文长度长）
- **相对位置嵌入（Relative Positional Embedding）**
  - 在每一层的注意力机制内实现相对位置信息的编码
  - 例如，DeBERTa 中的 Disentangled Attention
    - Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. ICLR 2021.
- 绝对位置嵌入与相对位置嵌入的结合：
  - 几乎所有大语言模型都在用的**旋转位置编码（Rotary Position Embedding, RoPE）**

旋转位置编码（RoPE）

Su et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. Zhuiyi 2021.

- 旋转位置编码（RoPE）直接在注意力层对位置信息进行编码
- 假设查询向量 $\boldsymbol{q}$ 的维度为 2，乘上一个旋转矩阵 R_m 可将其旋转 $m\theta$ 角度，其中 m 是对应词元的位置

$$R_m \boldsymbol{q} = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$$

- 类似地，键向量 $\boldsymbol{k}$ 可以旋转 $n\theta$ 角度，其中 n 是对应词元的位置

旋转位置编码（RoPE）

Su et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. Zhuiyi 2021.

- 实际情况中，$\boldsymbol{q}$ 和 $\boldsymbol{k}$ 的维度 d 远远不止 2
- 因此，可以将 d 维空间切分为 $d/2$ 个子空间（维度两两一组）

$$R_m\boldsymbol{q} = \begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix}$$

其中 $\theta_i = 10000^{-2i/d}$（与三角式位置嵌入相同）

旋转位置编码（RoPE）

Su et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. Zhuiyi 2021.

- 在计算注意力时：

$$\left(R_m \boldsymbol{q}\right)^T R_n \boldsymbol{k} = \boldsymbol{q}^T R_m^T R_n \boldsymbol{k}$$
$$= \boldsymbol{q}^T R_{n-m} \boldsymbol{k}$$

- 因此，注意力层编码了**相对位置信息**
- 旋转位置编码具有**远程衰减**性：
  - 随着相对距离 $n - m$ 变大，内积结果有衰减趋势

Transformer 训练

Transformer 训练

- Transformer 的训练与 RNN 类似，不同的是解码器一次性读入所有输入