# Transformer 在时间序列中的应用

刁姝心、沙漠、陆佳欢

数学科学学院

2022.4.13

# 基本信息


## 刁姝心

Email：1446107067@qq.com　　Tel: 18252587369
QQ: 1446107067

## 沙漠

Email：2300024266@qq.com　　Tel: 18052526508
QQ: 2300024266

## 陆佳欢

Email：207829897@qq.com　　Tel: 17849061238
QQ: 207829897


江苏大学 JIANGSU UNIVERSITY

# 0 目录

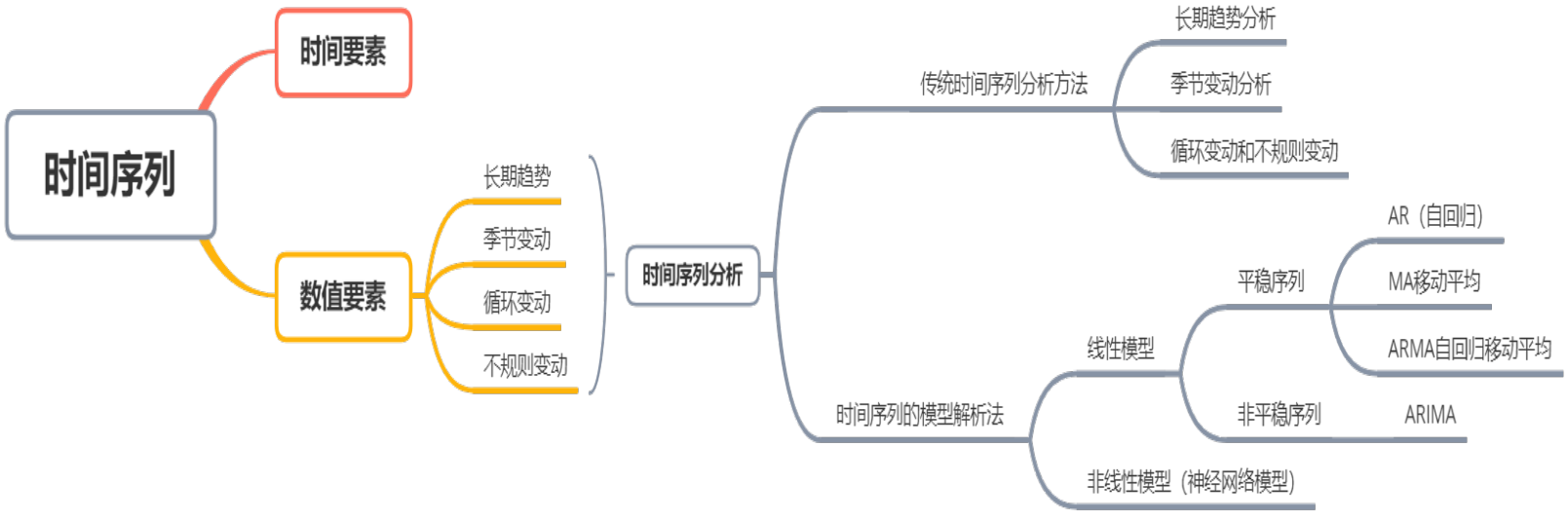




江苏大學 JIANGSU UNIVERSITY

# 1 统计学对时间序列的研究






江苏大学 JIANGSU UNIVERSITY

# 1 时间序列在深度学习上的体现



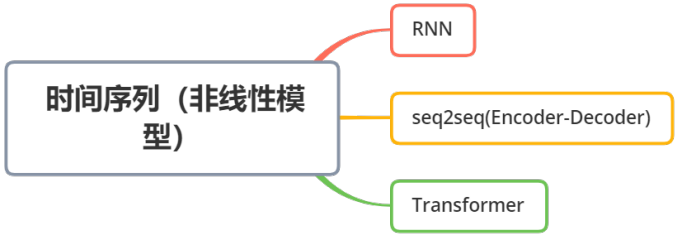



图: 序列模型方法

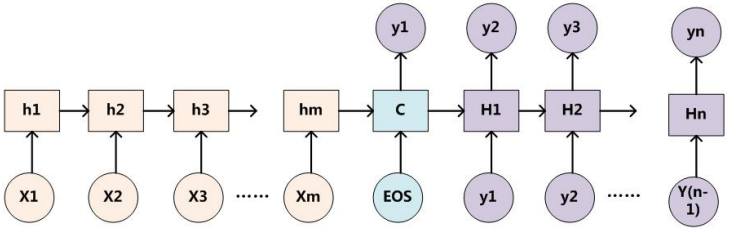


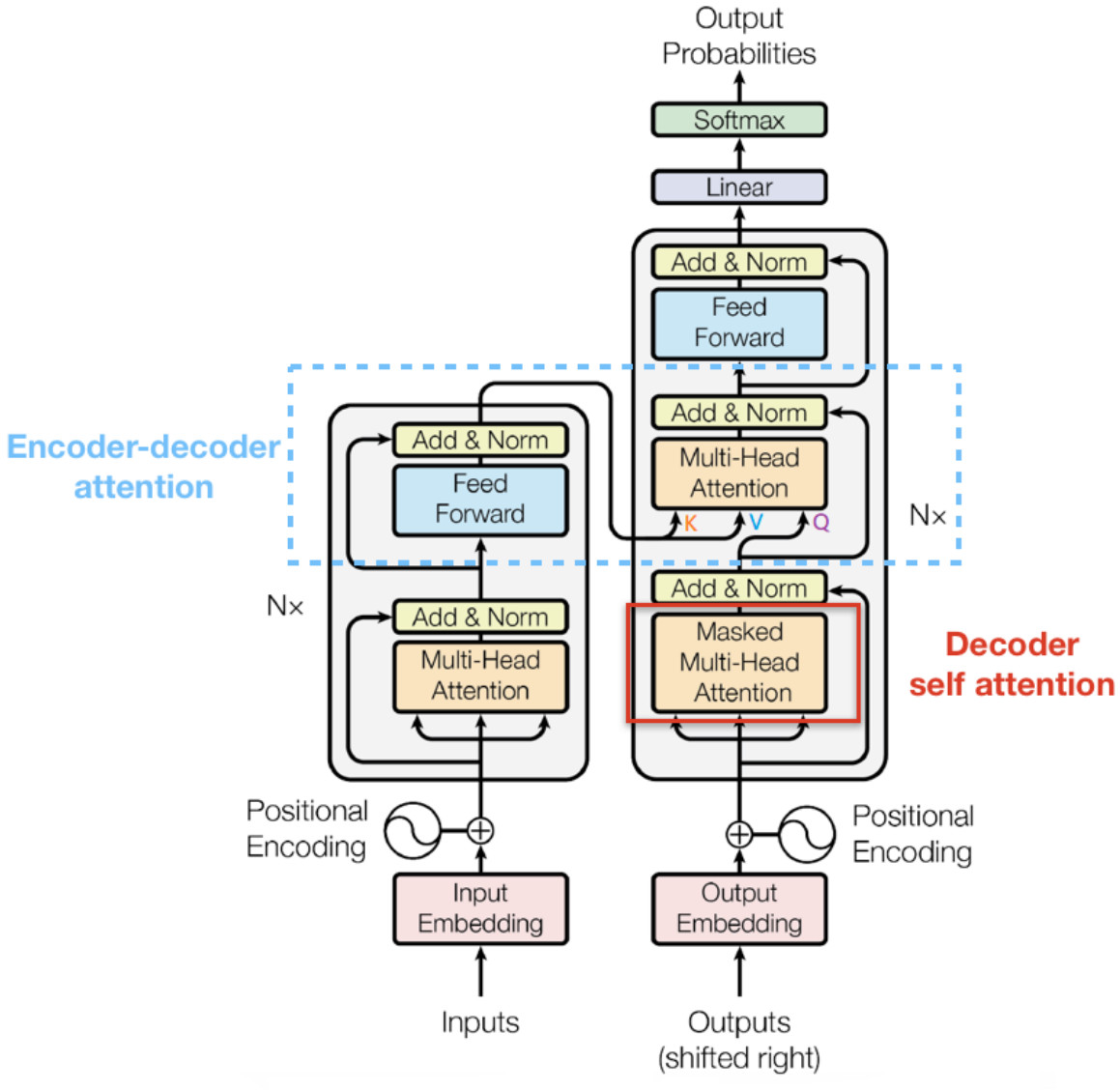

图: RNN Encoder-Decoder 框架

- RNN 由于其顺序性以及时间平移非对称的特性，通常模型会更容易受到输入序列中较后位置的数据的影响。
- seq2seq 是标准的 Enocder-Decoder 模型，加入 attention 后从而能够保存更多的信息。
- Transfromer 不必顺序翻译文本，提高模型的并行能力，解决 LSTM 无法缓解的长期依赖问题。

江苏大学 JIANGSU UNIVERSITY

# 2 Transformer 架构





# 2 Transformer 结构



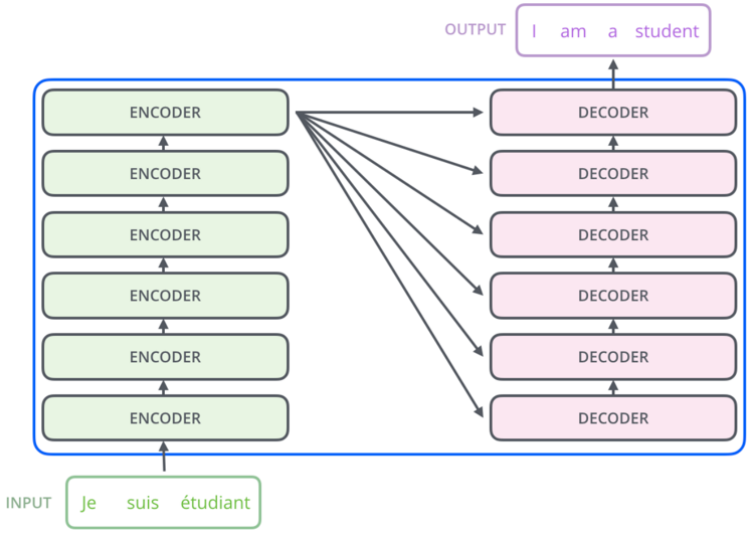



图: Transformer 整体结构

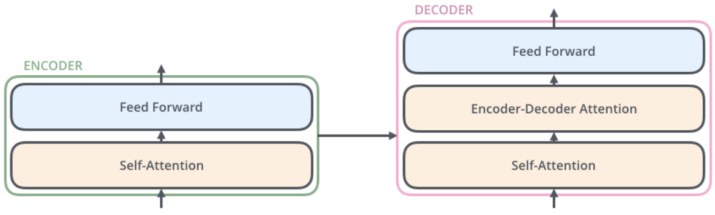



图: encoder-decoder

- 编码部分（6 个编码器）
- 解码部分（6 个解码器）
- 编码器（自注意力 + FFN）
- 解码器（自注意力 + 掩码注意力 + FFN）

江苏大學 JIANGSU UNIVERSITY

# 2 Positional Encoding



Transformer 抛弃了 RNN，而 RNN 最大的优点就是在时间序列上对数据的抽象，所以文章中作者提出两种 Positional Encoding 的方法，将 encoding 后的数据与 embedding 数据求和，加入了相对位置信息。

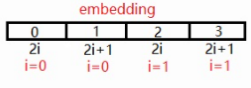
## 基本思路

1. 用不同频率的 sine 和 cosine 函数直接计算
2. 学习出一份 positional embedding

实验发现两种效果一样，作者选择了第一种。

- 对于一个单词 w 位于 $pos \in [0, L-1]$, 假设使用 encoding 的维度为 4, 那么这个单词的 encoding 为:

$$e_w = \left[\sin\left(\frac{\text{pos}}{10000^{\circ}}\right), \cos\left(\frac{\text{pos}}{10000^{\circ}}\right), \sin\left(\frac{\text{pos}}{10000^{2/4}}\right), \cos\left(\frac{\text{pos}}{10000^{2/4}}\right)\right]$$
$$= \left[\sin(\text{pos}), \cos(\text{pos}), \sin\left(\frac{\text{pos}}{100}\right), \cos\left(\frac{\text{pos}}{100}\right)\right]$$

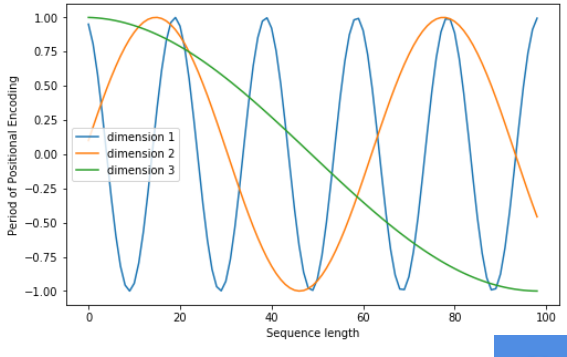



- 根据公式:

$$\text{PE}(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right),$$

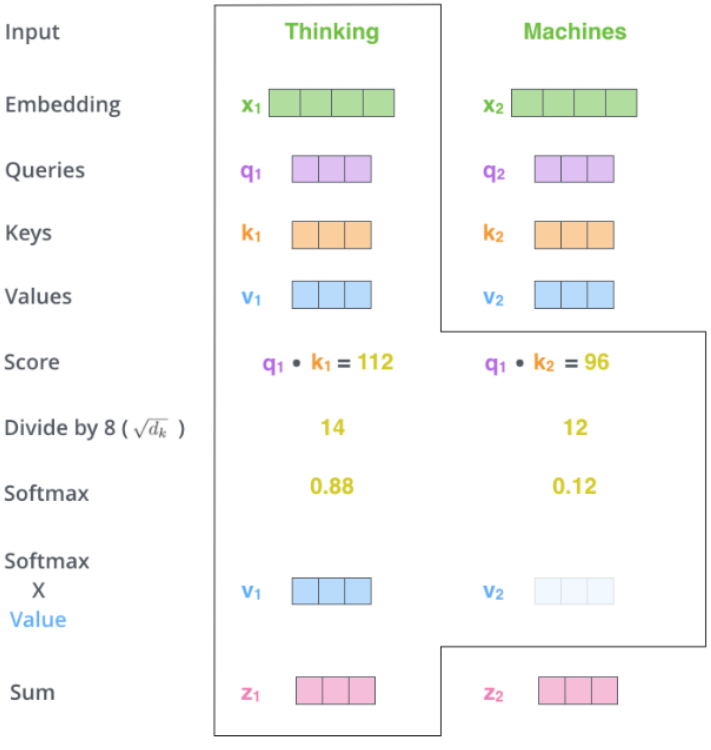
$$\text{PE}(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right).$$

positional-encoding-2




江苏大学 JIANGSU UNIVERSITY

# 向量计算




- 计算 $q_1, k_1, v_1$.

$$q_1 = x_1 W^Q \quad q_2 = x_2 W^Q$$
$$k_1 = x_1 W^K \quad k_2 = x_2 W^K$$
$$v_1 = x_1 W^V \quad v_2 = x_2 W^V$$

- 求权重 $\theta$.

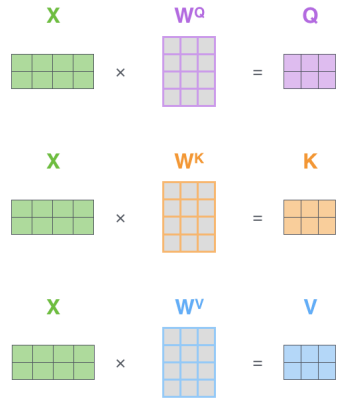
$$[\theta_{11}, \theta_{12}] = \text{softmax}\left(\frac{q_1 k_1^T}{\sqrt{d_k}}, \quad \frac{q_1 k_2^T}{\sqrt{d_k}}\right)$$
$$[\theta_{21}, \theta_{22}] = \text{softmax}\left(\frac{q_2 k_1^T}{\sqrt{d_k}}, \quad \frac{q_2 k_2^T}{\sqrt{d_k}}\right)$$

- 求 $z_1, z_2$

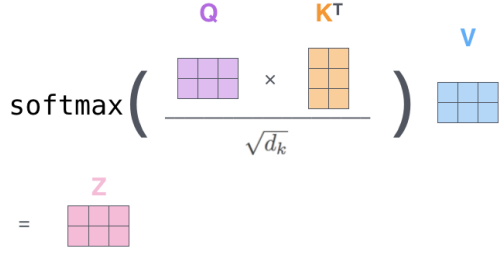
$$z_1 = \theta_{11} v_1 + \theta_{12} v_2$$
$$z_2 = \theta_{21} v_1 + \theta_{22} v_2$$

# 矩阵计算

- 词嵌入的矩阵 $X$ 乘以训练的权值矩阵（$W^Q, W^K, W^V$）

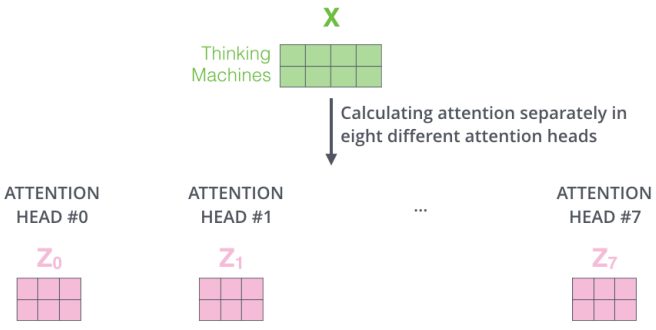


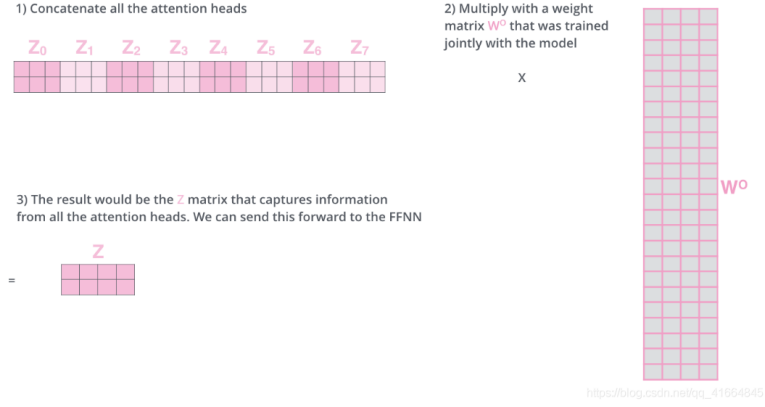

- 使用 $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$. 计算 $Z$



# 2 Multi-head self-attention






- 使用多组 $W^Q, W^K, W^V$，以此来关注不同上下文，求特征矩阵 $Z_i$
- 将 $Z_i$ 按列拼接成一个大特征矩阵并通过一层全连接层得到输出 $Z$.

$$MultiHead(Q,K,V) = Concat(head_1,\ldots,head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

*Add&Norm* 模块 $\rightarrow$LayerNorm $(x + \text{Sublayer}(x))$

**残差连接**

- self-attention 加权之后输出，也就是 Self-Attention $(Q, K, V)$ ，然后把他们加起来做残差连接。

$$X_{\text{embedding}} + \text{MultiHeadAttention}\ (Q, K, V)$$

**LN**

- BN 它是取不同样本的同一个通道的特征做归一化;LN 它取的是同一个样本的不同通道做归一化。

$$\text{Layer Norm}(x) = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

## FFN

- FFN 即 Feed Forward Neural Network。这个全连接有两层，第一层的激活函数是 ReLU，第二层是一个线性激活函数，可以表示为：

$$\mathrm{FFN}(Z) = \max\left(0, ZW_1 + b_1\right) W_2 + b_2$$

- 若选择 ReLU 则，当做完残差和 LN 后得到 X 特征矩阵然后再做如下操作：

$$X_{\text{hidden}} = \mathrm{Linear}\left(\mathrm{ReLU}\left(\ \mathrm{Linear}\ \left(X_{attention}\right)\right)\right)$$

$$X_{\text{hidden}} = X_{\text{attention}} + X_{\text{hidden}}$$

$$X_{hidden} = \mathrm{LayerNorm}\left(X_{\text{hidden}}\ \right)$$

其中 $X_{\text{attention}}$ 是残差层的结果。

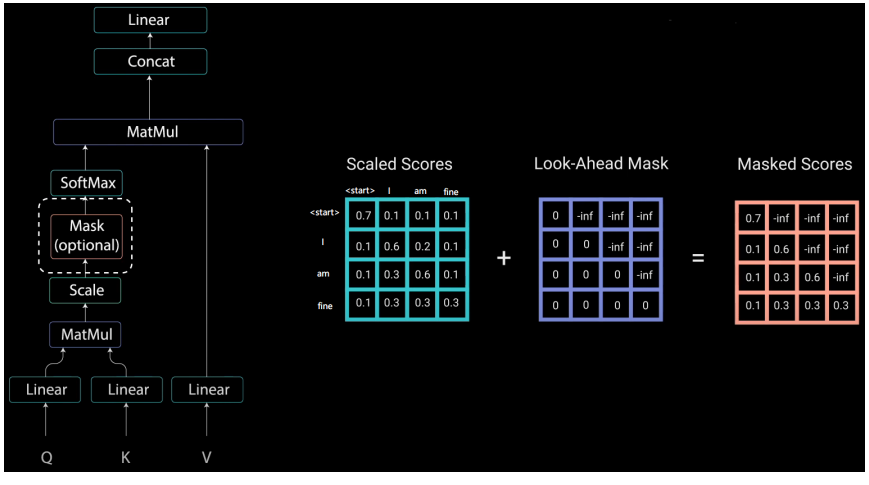
Encoder block 接收输入矩阵 $X_{nxd}$，输出一个矩阵 $O_{nxd}$. 最后一个 block 输出的矩阵就是编码信息矩阵 C，这一矩阵后续会用到 Decoder 中。

江苏大學 JIANGSU UNIVERSITY

# 2 Masked Self-Attention



Seq2Seq 中 Decoder 使用的是 RNN 模型，由于循环神经网络是时间驱动的，只有当时刻运算结束了，才能看到时刻的词。而 Transformer Decoder 抛弃了 RNN，改为 Self-Attention，由此就产生了一个问题，在训练过程中，整个 ground truth 都暴露在 Decoder 中，这显然是不对的，我们需要对 Decoder 的输入进行一些处理，该处理被称为 Mask.



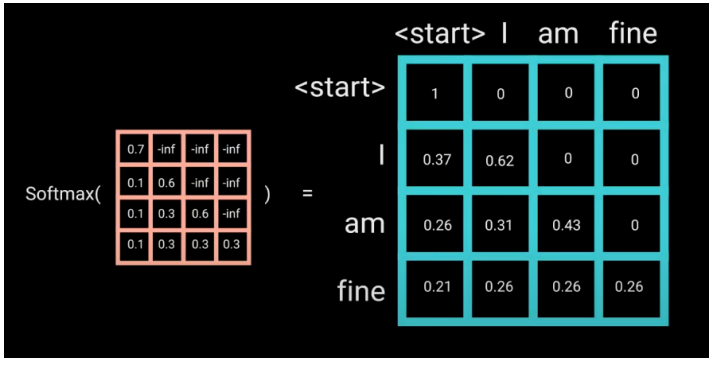




图: Mask $QK^T$

# 流程如下：

## Masked Self-attention 步骤

1. 对于 Decoder 的输入矩阵和 Mask 矩阵，输入矩阵包含所有单词 (翻译完) 的表示向量，Mask 与输入矩阵同维且遮挡 X 的某些单词信息。在 Mask 矩阵中，前面的单词只能用前面的信息。
2. 接下来的操作和之前的 Self-Attention 一样，通过输入矩阵 X 计算得到 Q, K, V 矩阵。然后计算 Q 和 $K^T$ 的乘积 $QK^T$。
3. 在得到 $QK^T$ 之后需要进行 Softmax，计算 attention score，我们在 Softmax 之前需要使用 Mask 矩阵遮挡住每一个单词之后的信息。得到 $QK^T$ 之后在 $QK^T$ 上进行 Softmax，每一行的和都为 1。但是单词 0 在单词 1, 2, 3, 4 上的 attention score 都为 0。
4. 使用 Mask $QK^T$ 与矩阵 V 相乘，得到输出 Z，则 Z 中前面的单词只包含前面的信息。

江苏大学 JIANGSU UNIVERSITY

# 2 Masked Encoder-Decoder Attention



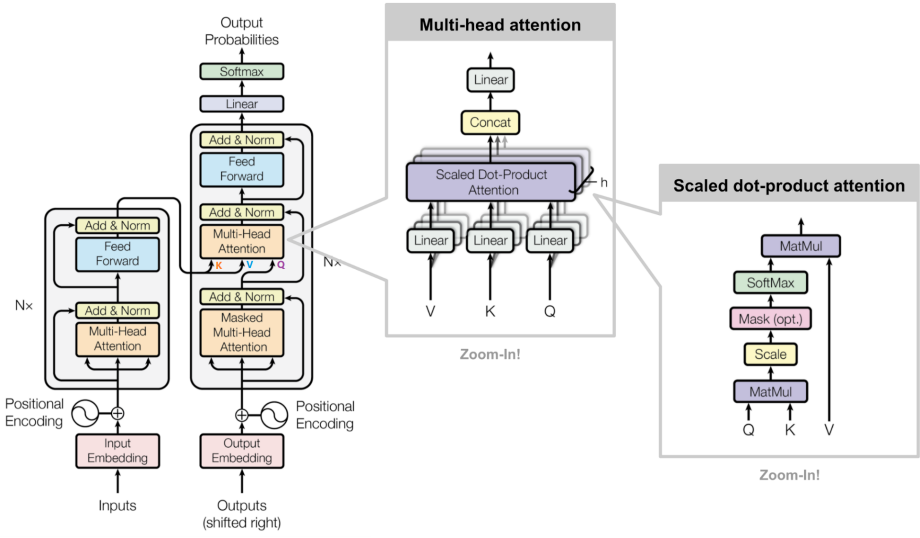


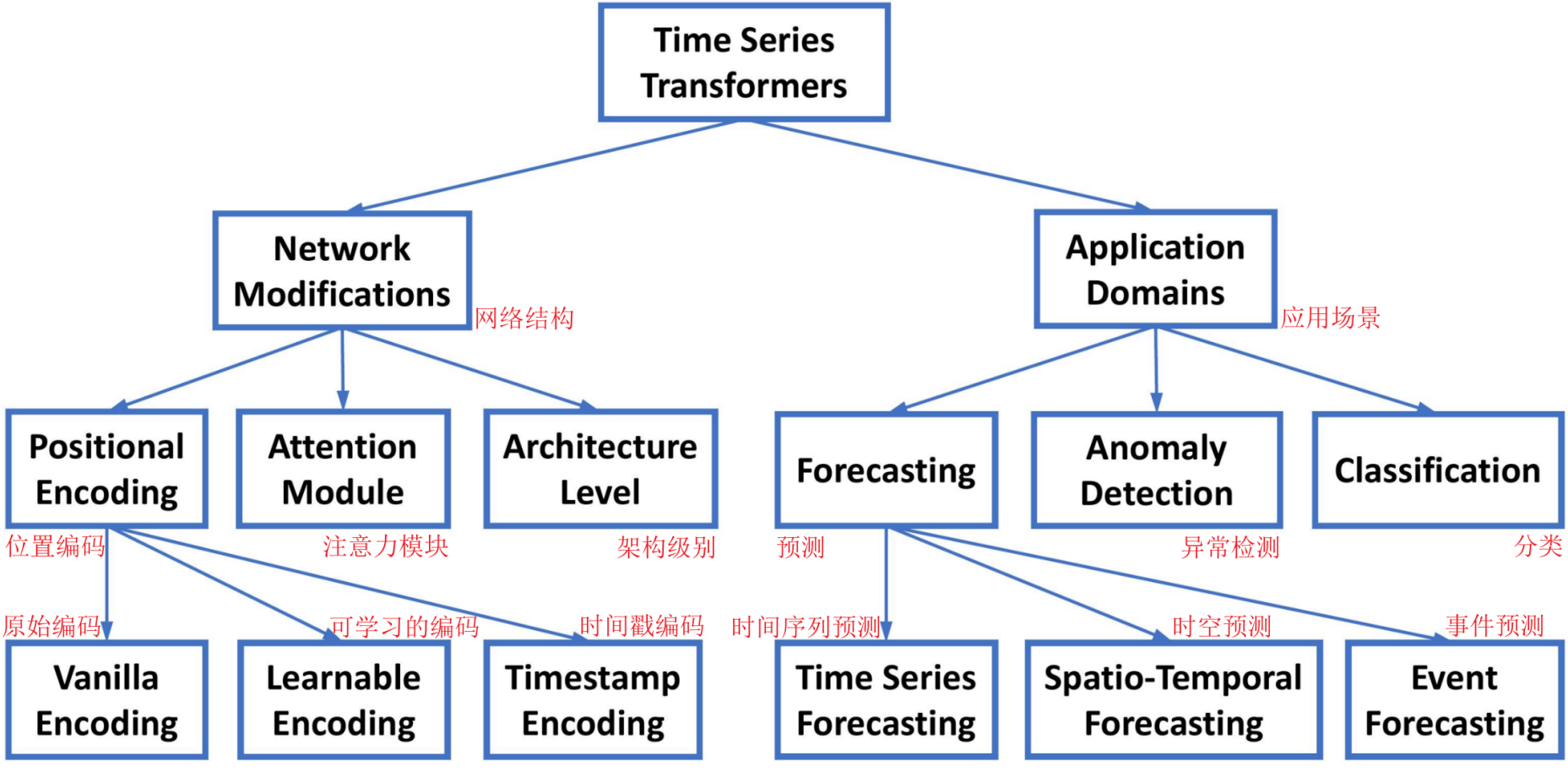

图: 序列模型方法

- K, V 矩阵不是使用上一个 Decoder block 的输出计算的，而是使用 Encoder 的编码信息矩阵 C 计算的。
- 根据 Encoder 的输出 C 计算得到 K, V，根据上一个 Decoder block 的输出 Z 计算 Q (如果是第一个 Decoder block 则使用输入矩阵 X 进行计算)
- Transfromer 不必顺序翻译文本，提高模型的并行能力，解决 LSTM 无法缓解的长期依赖问题。

江苏大学 JIANGSU UNIVERSITY

# 3 时间序列 Transformer 模型





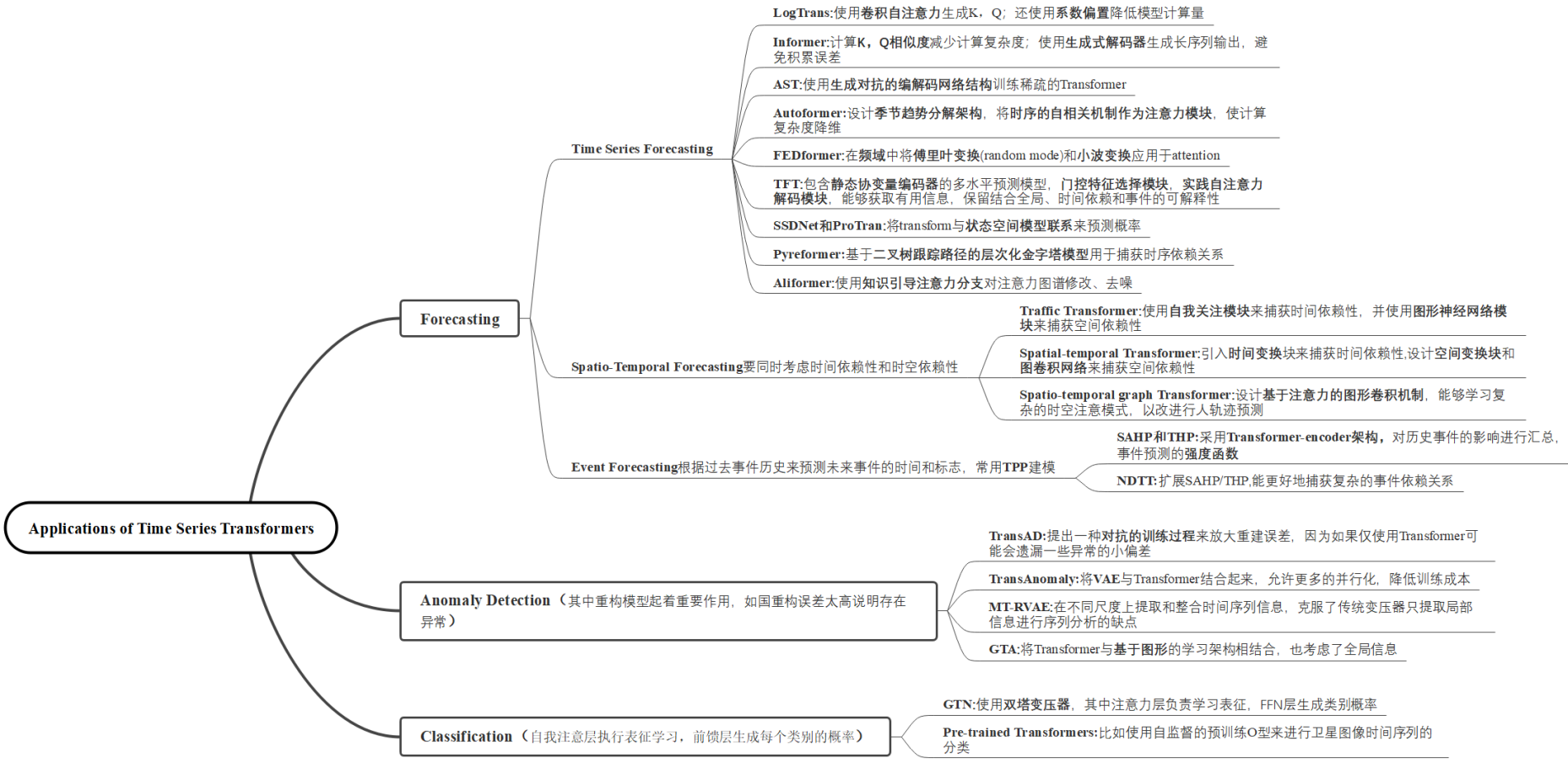

江苏大学 JIANGSU UNIVERSITY

# 时间序列的网络化

- Positional Encoding
  - 原始的位置编码送入 Transformer, 不能充分利用时间序列数据信息，也不能适应不同的数据。
  - Learnable Positional Encoding 通过学习每个位置的一组位置嵌入能表征更丰富的位置信息, 这样学习到的位置嵌入更灵活，能够适应特定的任务。
  - Timestamp Encoding 通常可以获得时间戳信息但在 Transformer 中几乎没有得到利用。
- Attention Module
  - 原始 Transformer 的时间和内存复杂度是 $O(L^2)$ 其中 $L$ 是输入序列长度，许多模型尝试减少这一计算复度，主要依赖稀疏性或者低秩近似的方式。
- Architecture-Level Innovation
  - 可以以较低的计算复杂度处理较长序列；
  - 层次建模可以获得多分辨率表示，对特定任务是有用的。

江苏大學 JIANGSU UNIVERSITY

# 时间序列 Transformer 应用场景




江苏大學 JIANGSU UNIVERSITY

- 鲁棒性分析
  - ”降低注意力计算复杂度一般使用小规模的输入”为了质疑这种观点而做了鲁棒性检验，
  - 当输入序列长度变化是许多算法性能会迅速恶化，这种现象使得许多精心设计的 Transformer 模型在长序列的实际应用中不切实际，因为他们不能有效的利用长序列输入信息，需要进行更多的研究来充分适应长输入序列。

- 模型大小分析
  - 3-6 层的浅层次 Transformer 获得了最优结果，同时也提出一个问题，即如何设定一个合适的深度来提高模型的容量、获得最好的预测性能。

- season-trend 分解分析
  - seasonal-trend decomposition 模块可以大幅度提升算法性能，提升高达 50%-80%, 这一独特的模块值得进一步研究。

# 5 Future Work



- 引入先验知识
  - 将序列的周期性或者频域特点加入 Transformer 结构可以带来性能提升。
  - 对特定时间序列和任务特点进行分析，采用最有效的方式引入先验知识，从而获得更优的 Transformer 架构。

- 引入 GNN
  - 引入图神经网络 GNN 是一种建模空间依赖性和维度间关系的方式。
  - GNN 与 Transformer 结合可以显著提升性能，还可以进行多模态预测、深入了解动态失控特征，因此二者的结合指的进一步研究。

- 预训练模型
  - Transformer 用于 NLP 需要大规模的预训练，但是对于时间序列的预训练 Transformer 模型研究有限，主要集中在时间序列的分类上，仍有很大的研究空间。

- NAS 设计模型
  - NAS 可以有效解放人工，因此如何利用 NAS 自动设计高效的时间序列 Transformer 是值得研究的。

江苏大學 JIANGSU UNIVERSITY

# Thank you for listening!

江蘇大學
JIANGSU UNIVERSITY