

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



基于Transformer的时间序列分析

李新帅

2023年06月11日

- 背景简介
- 基础概念
- 算法原理
- 应用总结
- 参考文献

- 预期收获
 - 1. 了解时间序列分析任务的基本概念
 - 2. 理解Transformer的基本原理
 - 3. 理解Transformer在时间序列任务中的应用
 - 4. 了解时间序列分析任务的前沿发展

时间序列 为什么?



- 应用场景

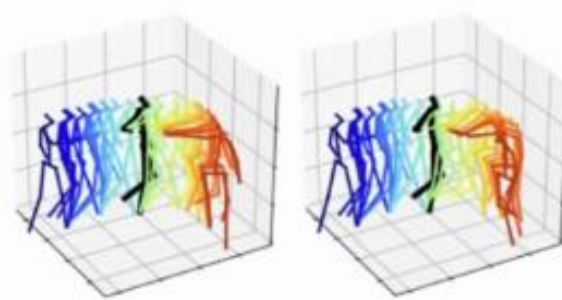
- 工程技术、气象、水文、生物医药、经济管理、智能运维



Stock Market Prediction



Robots Action Prediction



Human Position Prediction



Weather Prediction



Supply Chain Prediction



Society Event Prediction

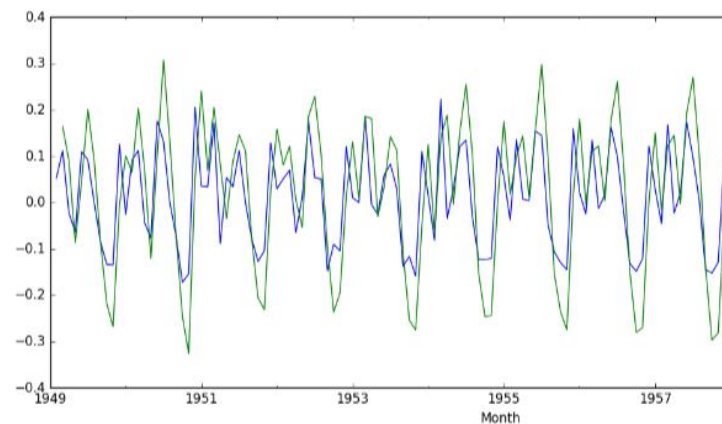
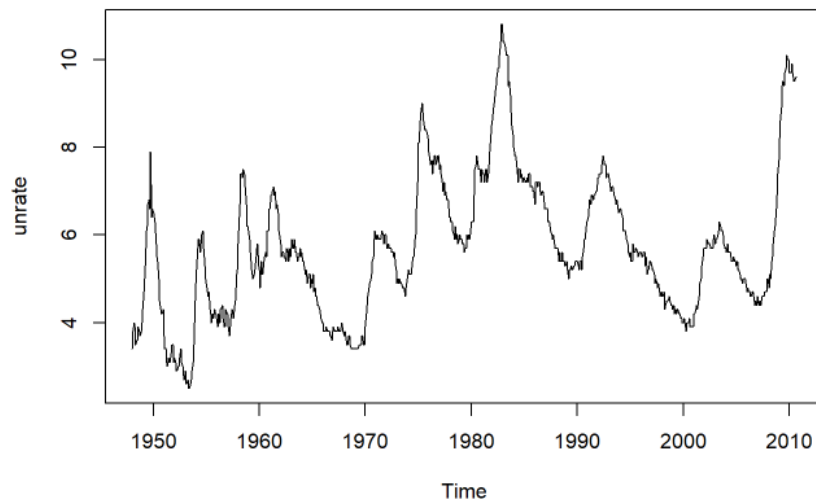
时间序列 是什么？



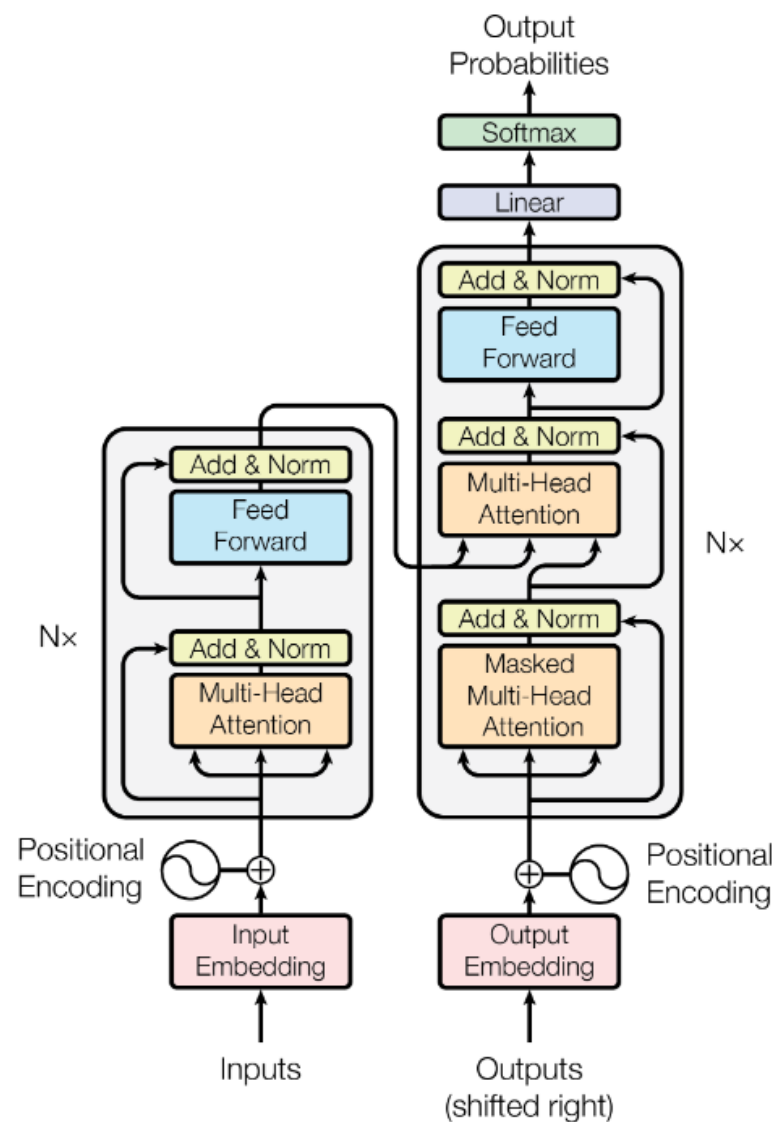
- 时间序列
 - 按照顺序把随机事件**变化发展的过程**记录下来就构成了一个时间序列
- 时间序列分析
 - 对时间序列进行观察、研究，找寻它变化发展的**规律**
- 特征
 - 长期性、周期性、季节性、不规则事件、特殊变化



- 按**数据维度**划分
 - 一元时间序列
 - 多元时间序列：包含多个一元时间序列作为分量，各个一元时间序列的**采样时间点相同**
- 按**时间特点**划分
 - 离散时间序列：采集的时间是离散的，例如气温变化曲线
 - 连续时间序列：采集的时间是连续的，例如信号采集
- 按**任务类型**划分
 - 预测：从当前值预测时间序列的未来值
 - 分类：给定多个时间序列，将它们按照相似性进行分类
 - 异常检测：检测时间序列行为何时发生变化



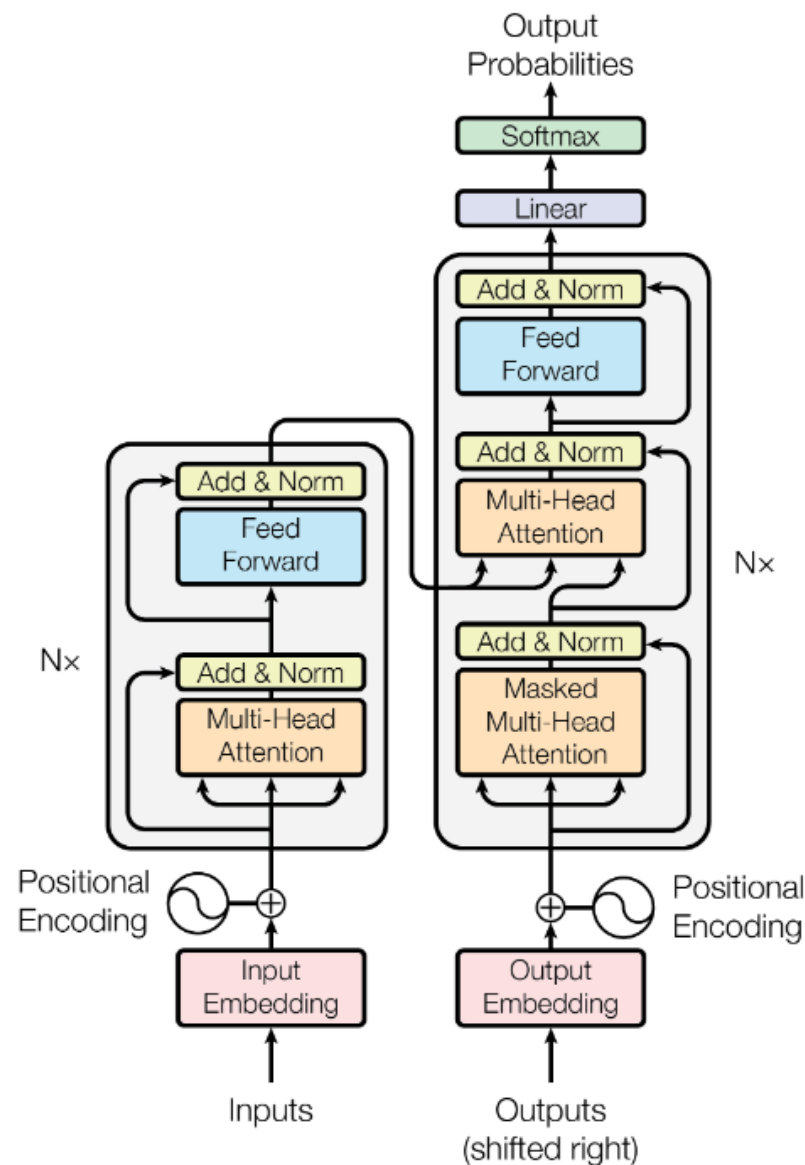
- 传统时间序列预测
 - AR、MA、ARMIA
 - GARCH、VAR
- 基于机器学习的时间序列预测
 - GBRT
 - Random Forest
- 基于深度学习的时间序列预测
 - LSTM
 - **Transformer**
 - 图神经网络



Transformer 为什么?



- 优势:
 - 更强的长期依赖建模能力，在长序列上效果更好
 - 支持并行，训练得更快
- 劣势:
 - 基于序列的编解码结构，自注意力机制，计算时间、空间复杂度高
 - 位置编码的准确性



- 时间复杂度:

- 是指执行当前算法所消耗的时间
- 算法的渐进时间复杂度

$$T(n) = O(f(n))$$

- 其中 $f(n)$ 表示每行代码执行次数之和
- O 表示正比例关系

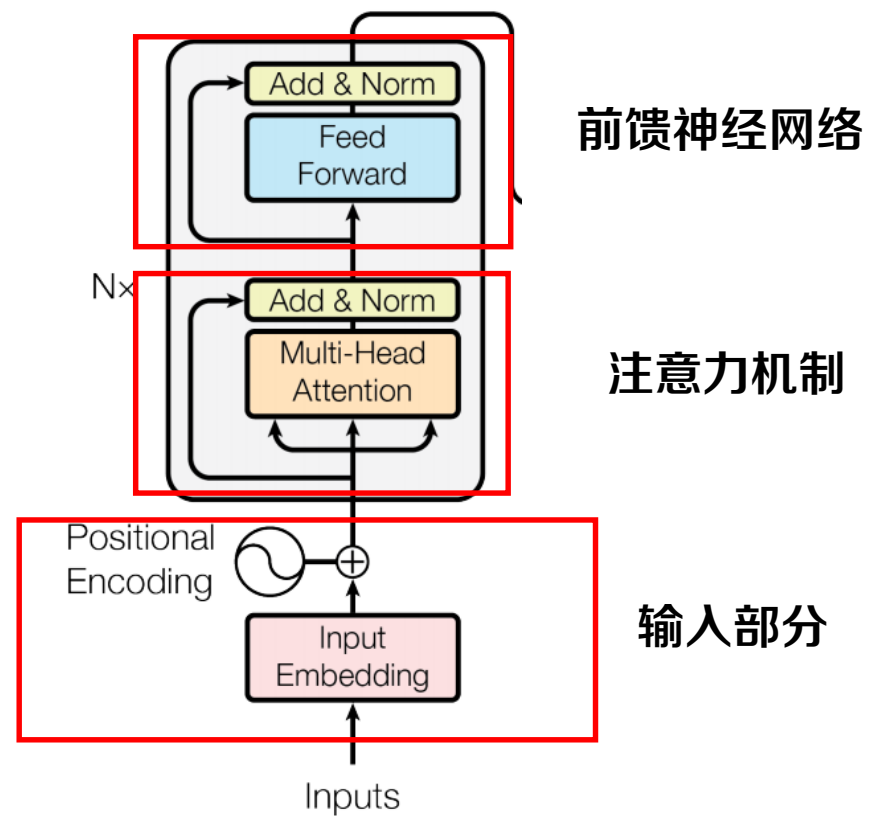
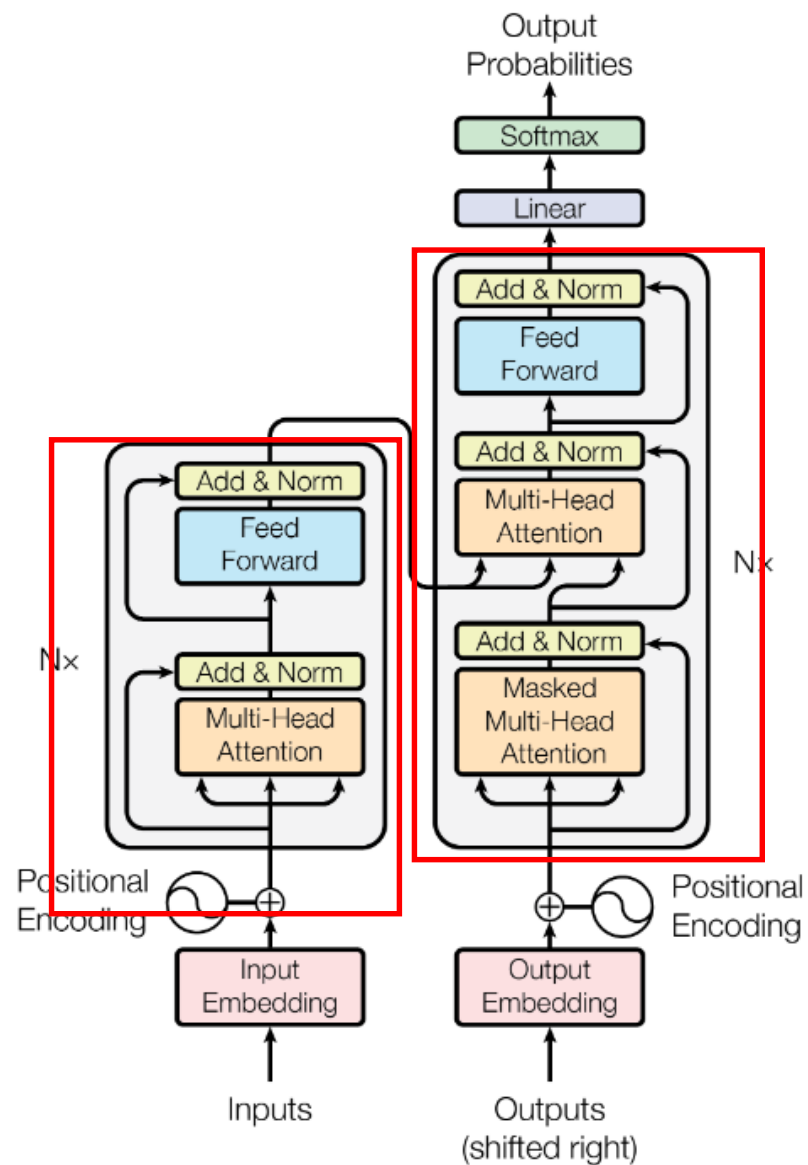
- 空间复杂度:

- 是指执行当前算法需要占用多少内存空间

```
for(i=1; i<=n; ++i)
{
    j = i;
    j++;
}
```

```
public static void init(int n){
    int a = 0;
    int arr[] = new int[n];
    for (int i=0;i<n;i++){
        arr[i]=n;
    }
}
```

Transformer 模型架构

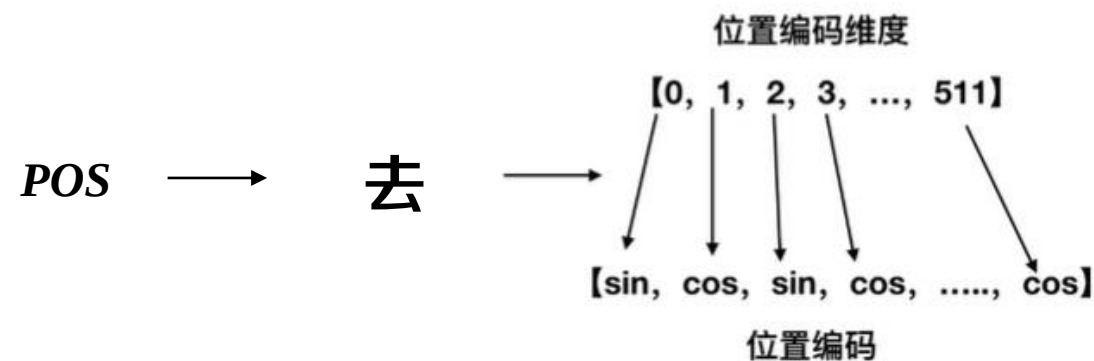
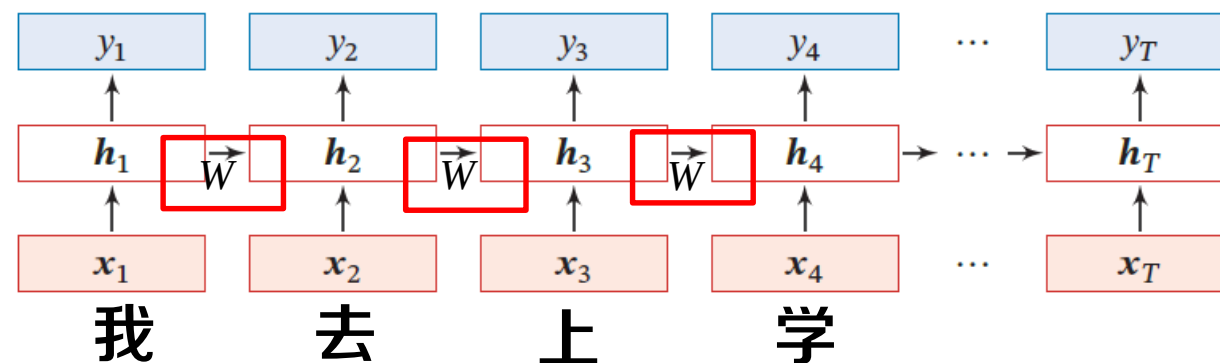
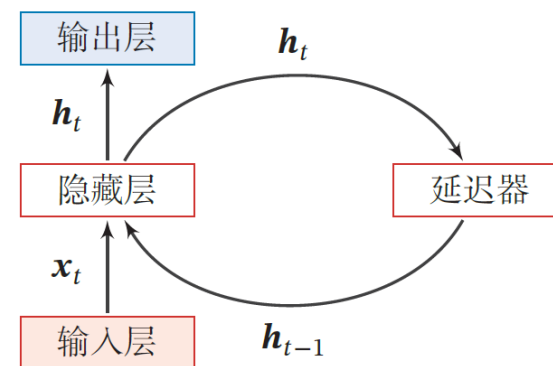


- 为什么需要?
 - 并行处理，提高速度，忽略顺序关系

- 为什么有用?
 - 表示绝对位置信息
 - 可推导相对位置信息

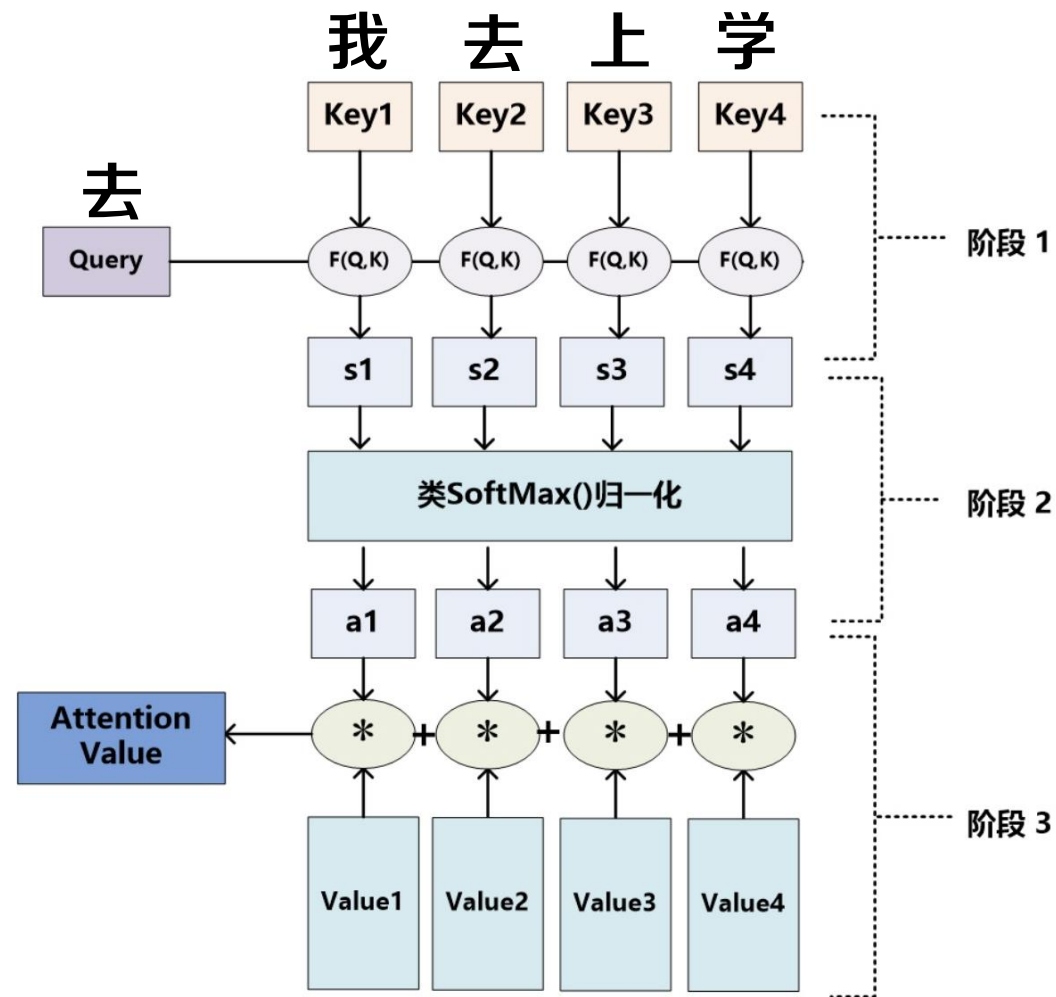
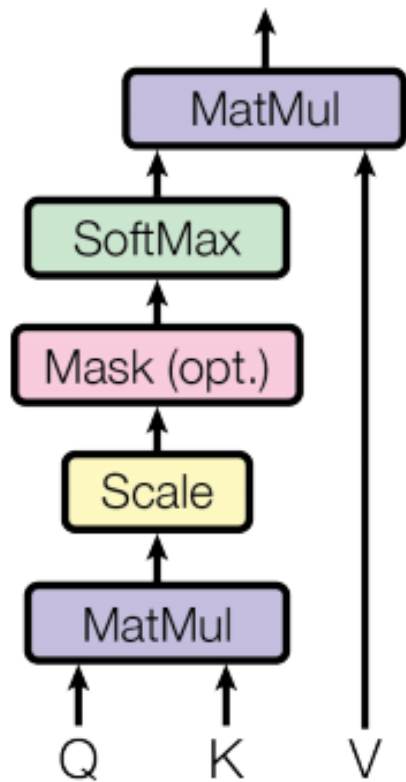
- 位置编码公式

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

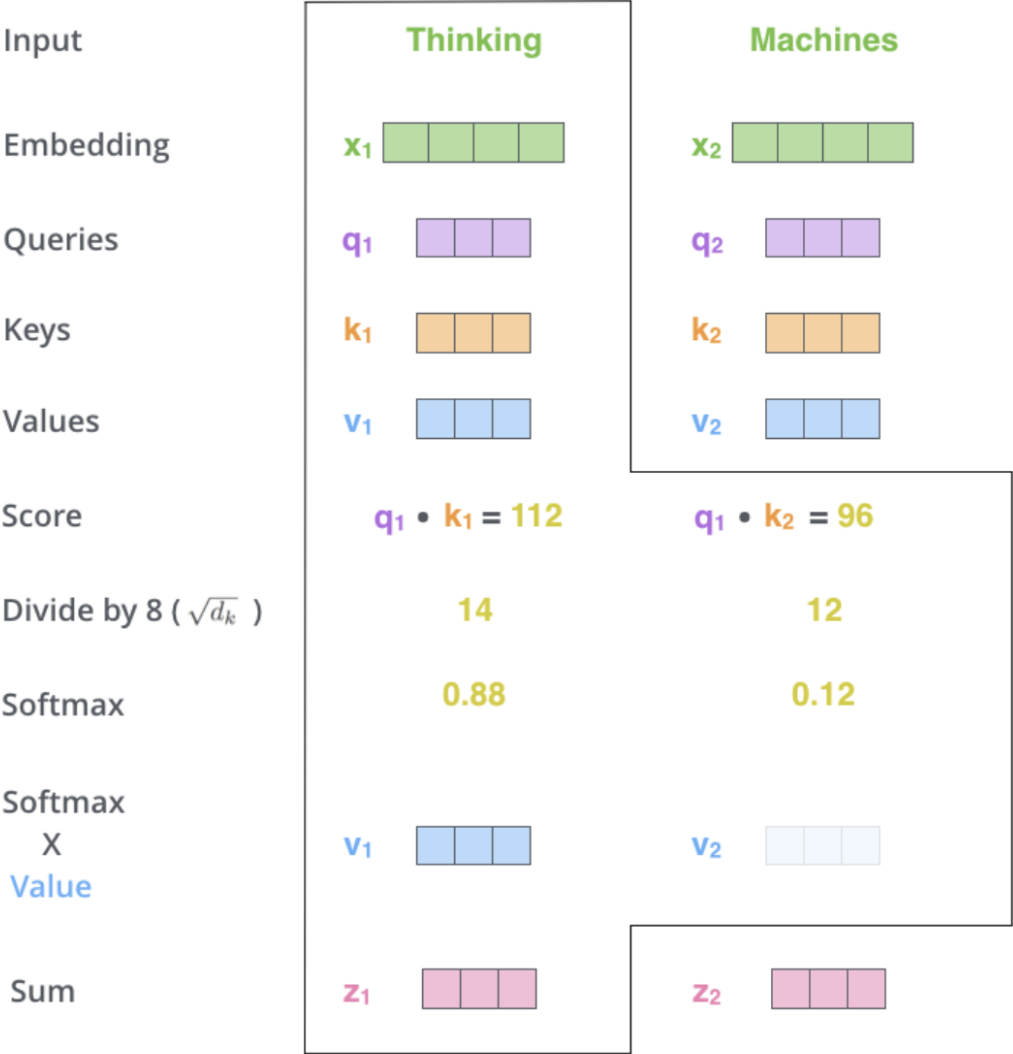


- 注意力机制

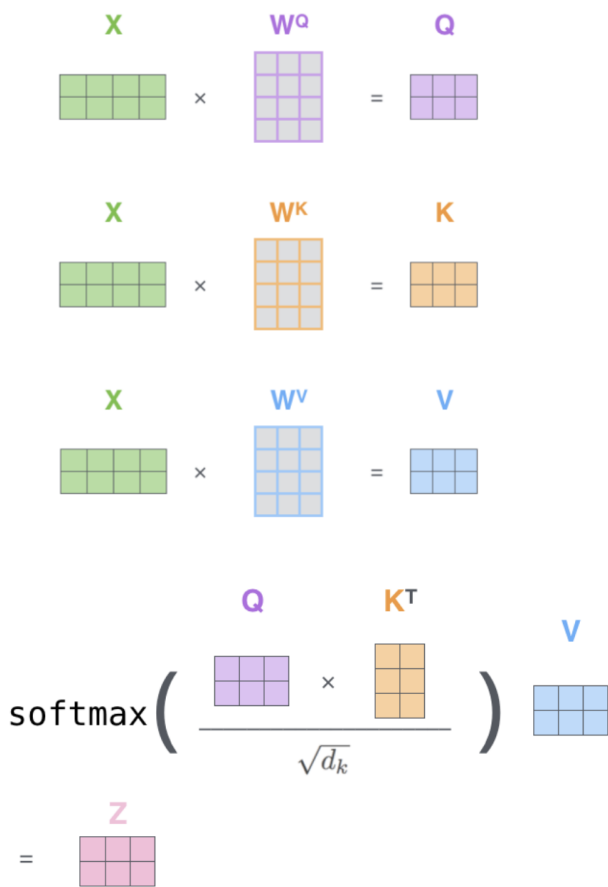
$$\mathcal{A}(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d}}\right)V$$



Transformer Attention



$$\mathcal{A}(Q,K,V) = softmax\left(\frac{QK}{\sqrt{d}}\right)V$$

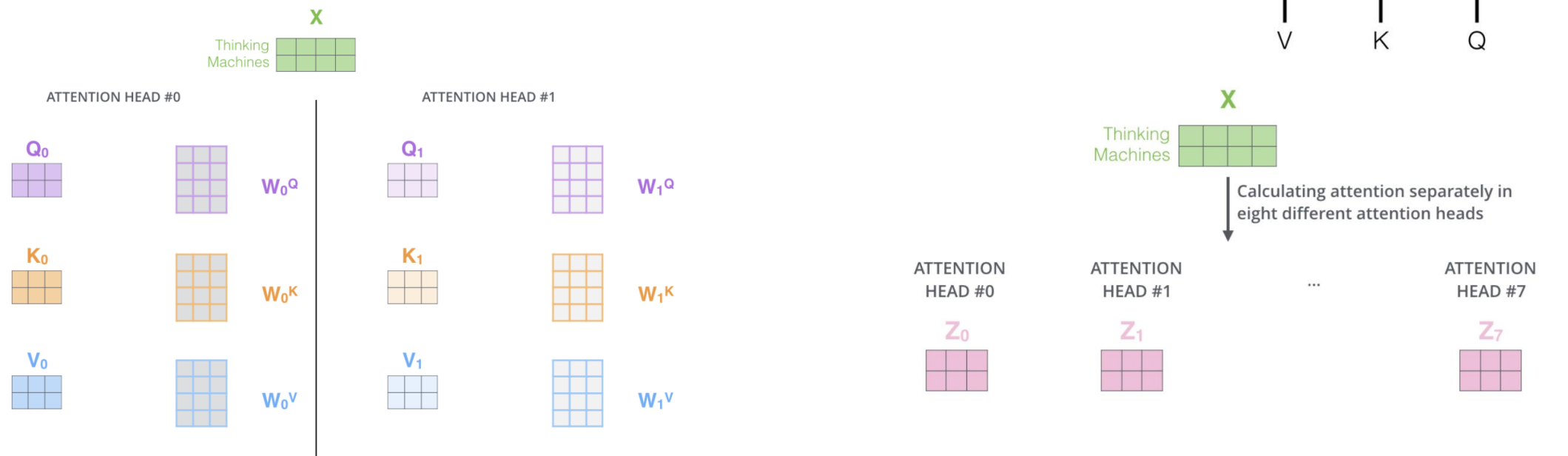


- Multi-Head Attention

- 所谓“多头”（Multi-Head），就是指多做几次同样的事情（参数不共享），然后把结果拼接
- 模型在不同的表示子空间里学习到相关的信息

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

$$MultiHead_i(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

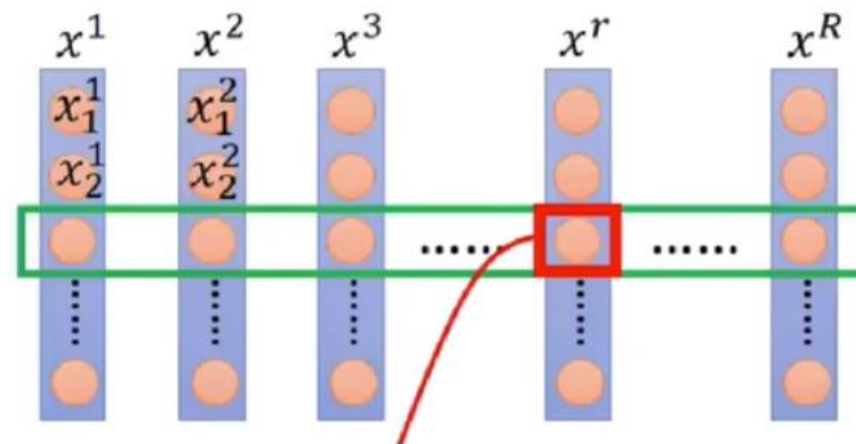
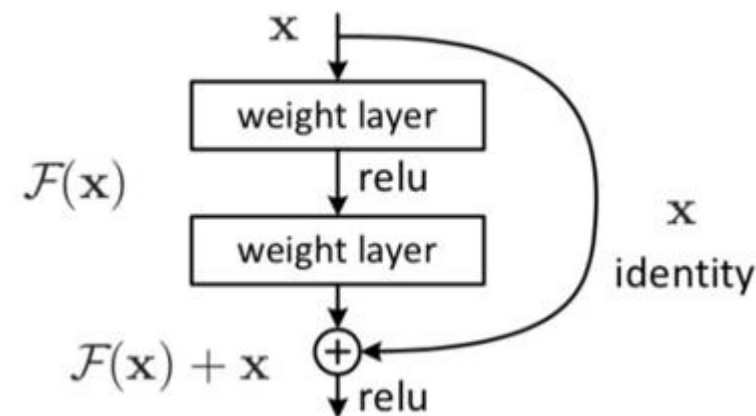
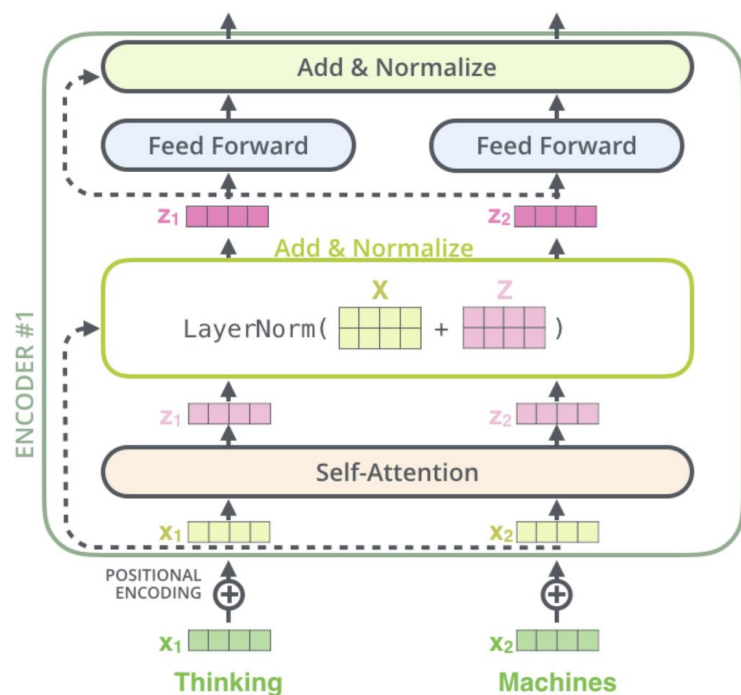


- 残差和LayerNorm

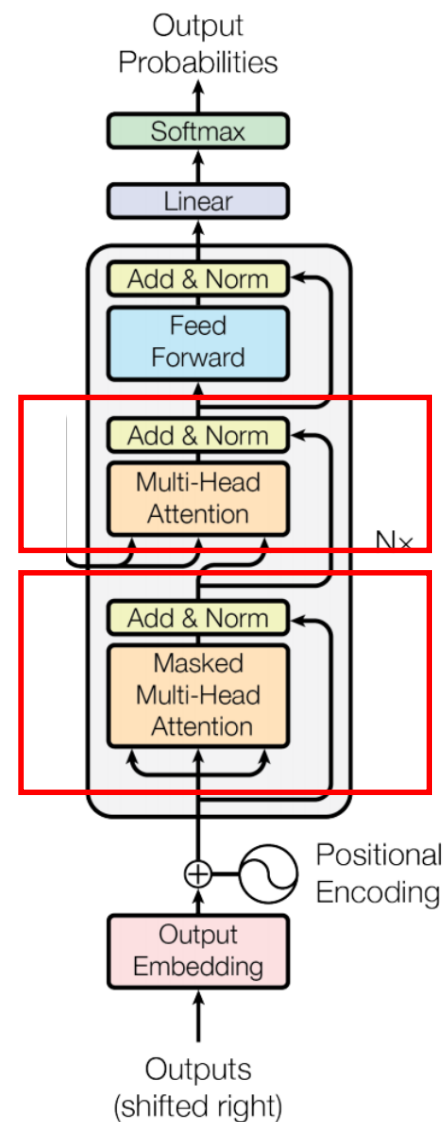
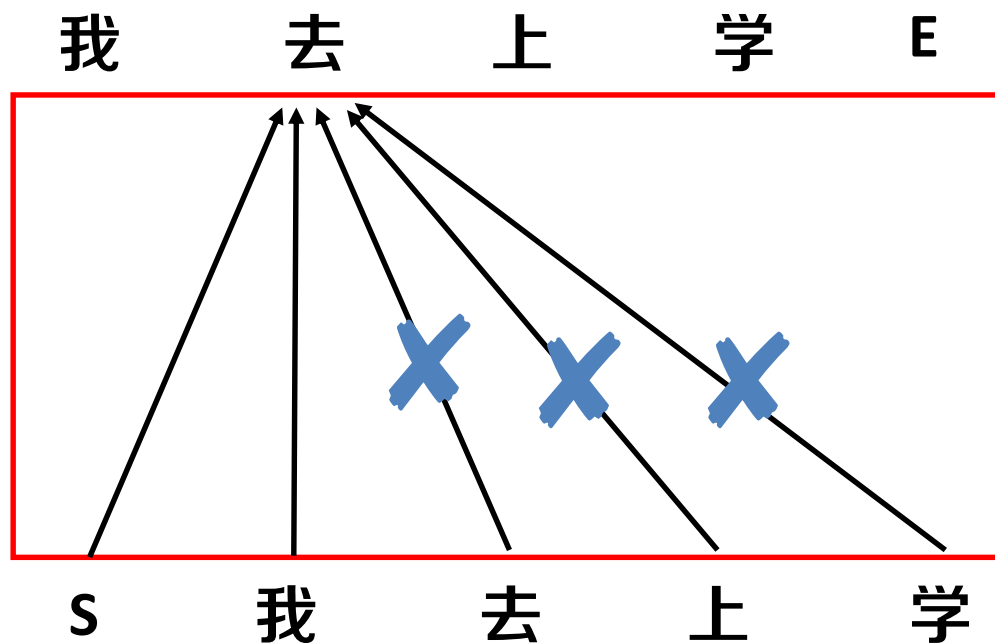
$$\text{LayerNorm}(X + \text{MultiHeadAttention}(X))$$

$$\text{LayerNorm}(X + \text{FeedForward}(X))$$

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

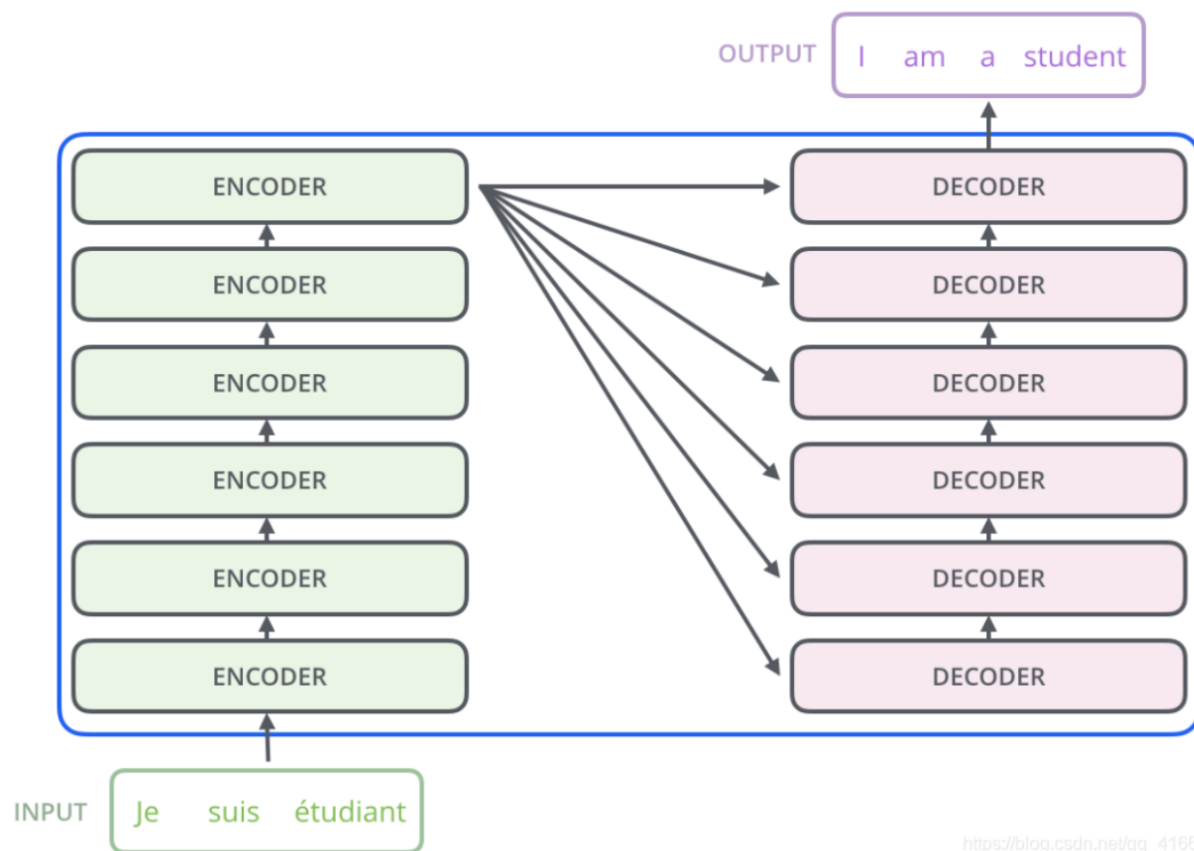


- 多头注意力机制
 - 对当前和之后的单词做mask

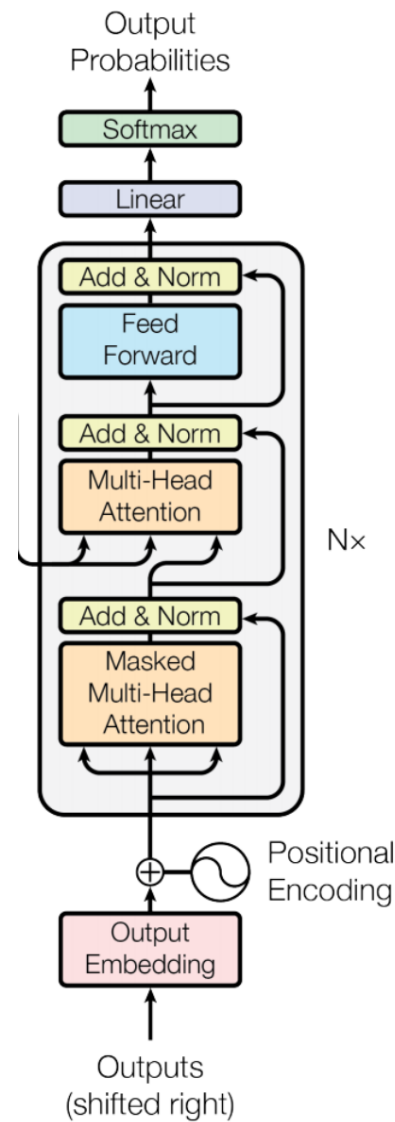


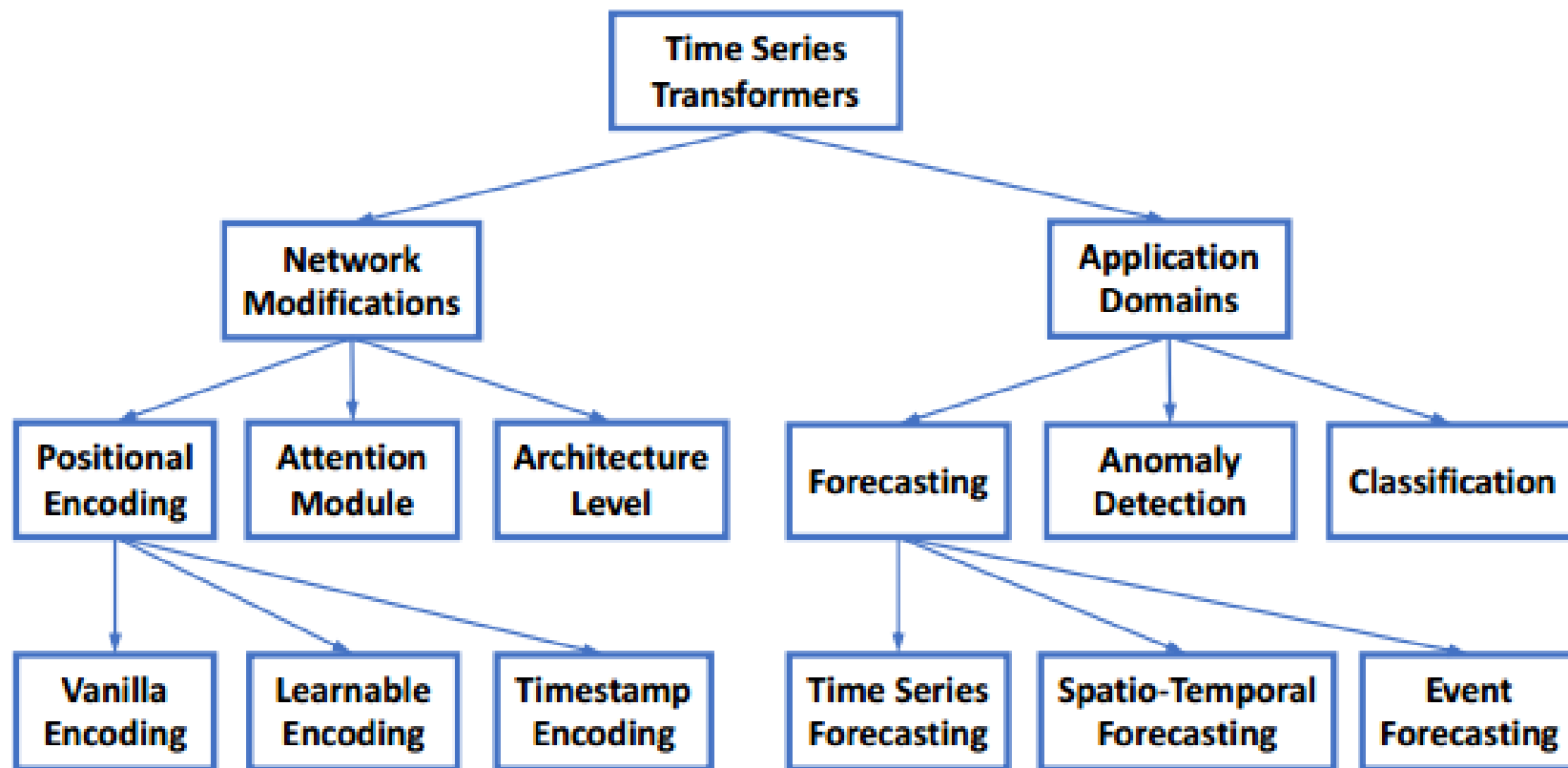
- 交互层

- 将Encoder与Decoder连接



https://blog.csdn.net/qq_41664845





- 对于位置编码的改进

- 可学习位置编码: TST (2021, KDD) $U' = U + W_{pos}$
- 时间戳的编码: Informer (2021, AAAI)、FEDformer (2022, ICML)

- 对于Attention的改进

- 引入稀疏性偏差: LogTrans (2019, NeurIPS)、Reformer (2020, ICLR)
- 研究低秩矩阵: Informer (2021, AAAI)、FEDformer (2022, ICML)

- 对于框架的改进

- Informer (2021, AAAI)、Pyraformer (2022, ICLR)



Informer

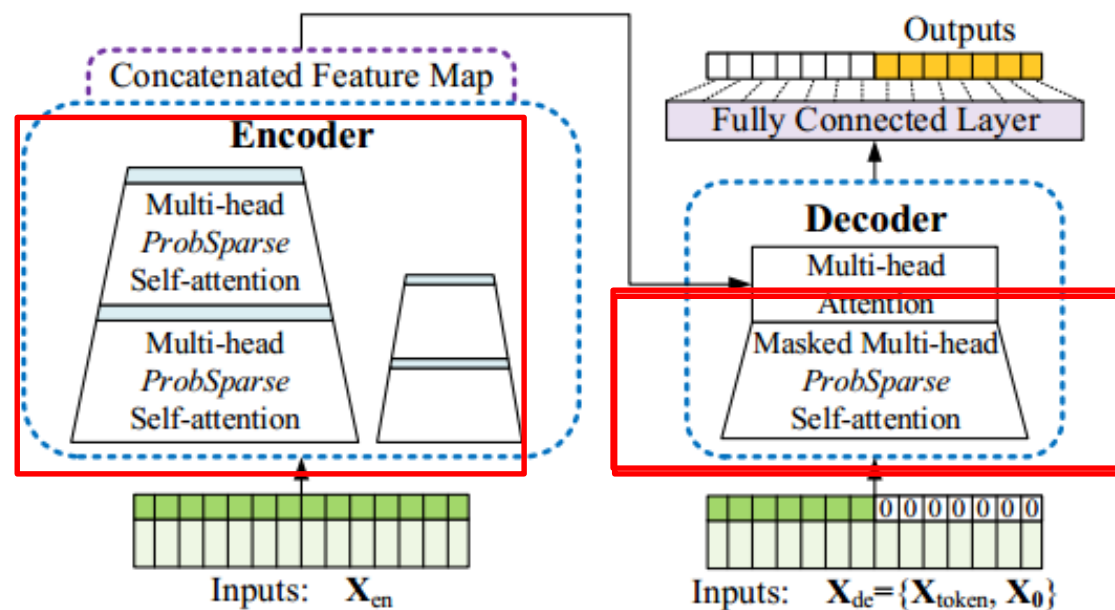
TIPO

T 目标	解决长序列的时间预测
I 输入	多元长时间序列
P 处理	1.Input embedding 2.Eecoder (Multi-head <i>ProbSparse</i> Self-attention) 3.Decoder 4.outputs
O 输出	未来一段时间的预测值

P 问题	时间、空间复杂度高；处理超长时间序列；时间编码的准确性
C 条件	具有完整时间戳的多元长时间序列
D 难点	如何优化Transformer模型，使其更适用于长时间序列预测任务
L 水平	AAAI, 2021

Transformer的问题

- Self-attention机制的点积操作使得时间和空间复杂度是 $\mathcal{O}(L^2)$ ， L 为序列长度。
- 高内存使用量问题。多个Encoder、Decoder层堆叠需要大量memory $\mathcal{O}(NL^2)$ ， N 是Encoder、Decoder数量。
- 预测长序列是速度骤降。Transformer的Decoder是逐步推理的，使得速度变慢。



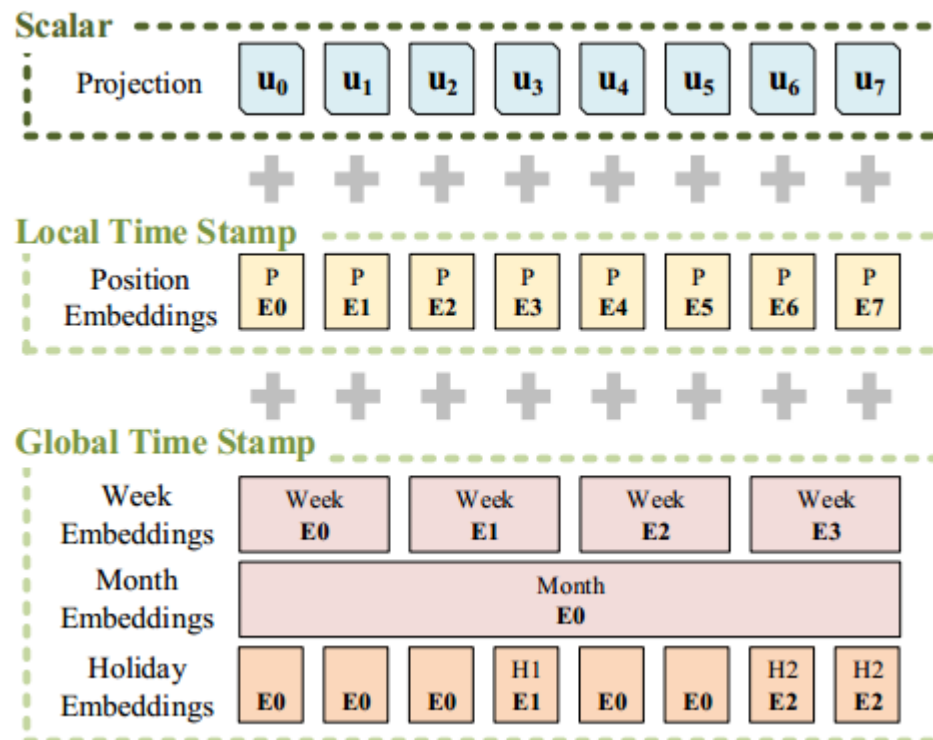
• 解决方案

- 提出ProbSparse self-attention机制来替换self-attention，使得时间和空间复杂度降为 $\mathcal{O}(L \log L)$ ，
- 提出self-attention distilling来突出dominating score，缩短每一层输入的长度，可处理更长的输入序列
- 提出generative decoder来输出预测，此过程仅需要one forward step，时间复杂度降为 $\mathcal{O}(1)$

数据插值

- Input embedding

$$\mathcal{X}_{feed[i]}^t = \alpha u_i^t + PE_{(L_x - (t-1) + i)} + \sum_p [SE_{(L_x - (t-1) + i)}]_p$$



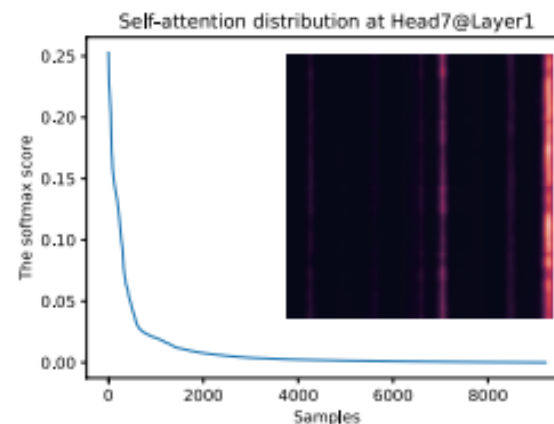
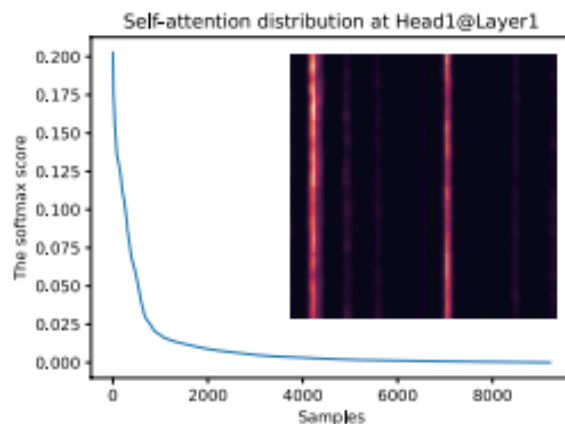
- ProbSparse self-attention

- 存在长尾分布，少量的query和key的点积结果占主导
- 对第*i*个query的Attention定义为核平滑的概率形式

$$\mathcal{A}(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = \mathbb{E}_{p(k_j|q_i)}[v_j]$$

注意力概率

$$p(k_j|q_i) = \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} \quad \Rightarrow \quad \mathcal{O}(L_Q L_K)$$
$$k(q_i, k_j) = \exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right)$$



- ProbSparse self-attention

- 均匀概率分布

$$q(k_j|q_i) = \frac{1}{L_K}$$

- 利用KL散度度量均匀分布和概率分布的距离

$$KL(p||q) = \sum p(x) \log \frac{p(x)}{q(x)}$$

$$KL(q||p) = \ln \sum_{l=1}^{L_K} e^{\frac{q_i k_l^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^L \frac{q_i k_j^T}{\sqrt{d}} - \ln L_K$$

- 第*i*个query的稀疏性度量公式

$$M(q_i, K) = \ln \sum_{l=1}^{L_K} e^{\frac{q_i k_l^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^L \frac{q_i k_j^T}{\sqrt{d}}$$

- 每个key的前u个主导性注意力分数

$$\mathcal{A}(Q, K, V) = \text{softmax} \left(\frac{\bar{Q}K}{\sqrt{d}} \right) V$$

$$u = \text{cln} L$$



$$\mathcal{O}(L \ln L)$$

- self-attention Distilling

- 提取长序列输入的长期依赖性

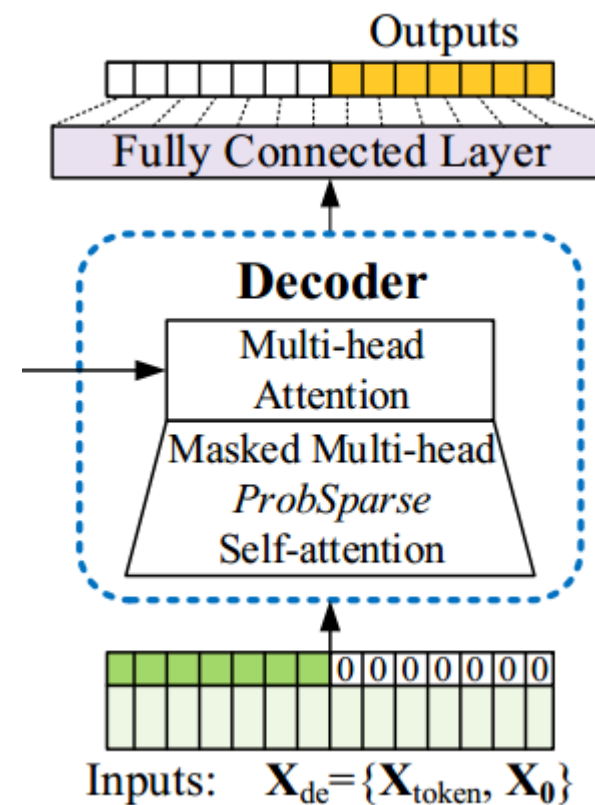
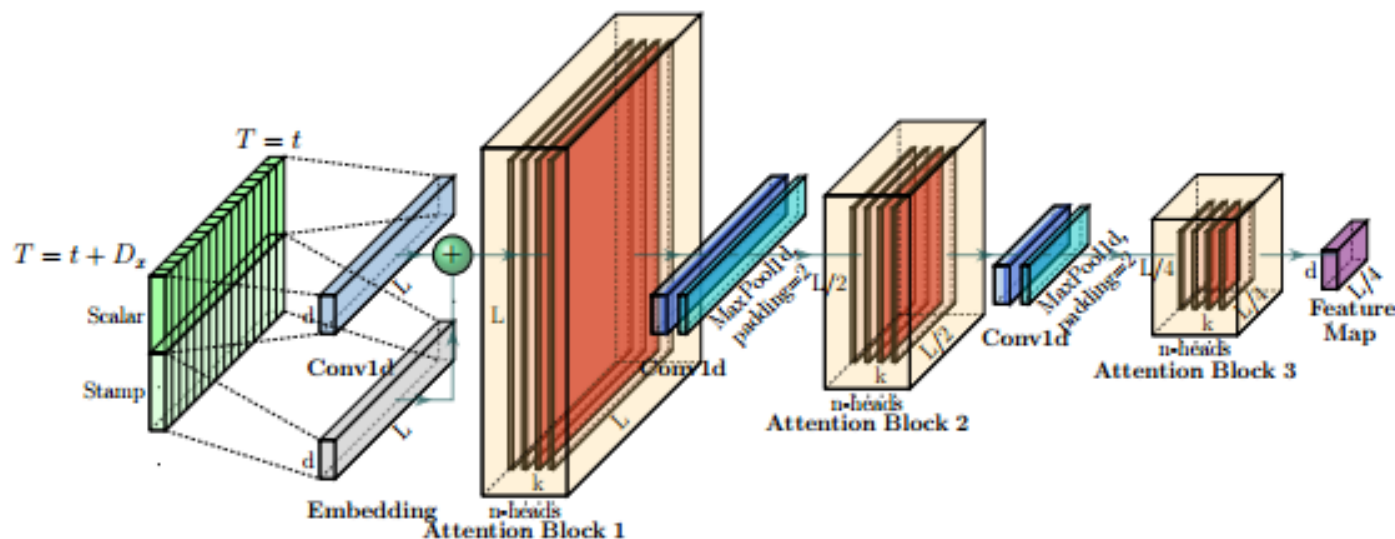
$$X_{j+1}^t = \text{MaxPool} \left(\text{ELU} \left(\text{Conv1d} \left([X_j^t]_{AB} \right) \right) \right)$$

- 生成式解码器

- 用来缓解长时间预测的速度下降问题

$$X_{de}^t = \text{Concat}(X_{Token}^t, X_0^t)$$

$$X_{de} = \{X_{5d}, X_0\}$$



• 实验设计

– 数据集：3个公开数据集 ETT、Weather、ECL 进行实验

– 评价指标：

• MAE（平均绝对误差）：对表达相同现象的成对观察之间的误差的度量

$$\frac{1}{m} \sum_{i=1}^m |Y_i - \hat{Y}_i|$$

• MSE（均方方差）：预测数据和原始数据对应点误差的平方和的均值

$$\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

实验结果

Methods		Informer		Informer [†]		LogTrans		Reformer		LSTMa		DeepAR		ARIMA		Prophet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh ₁	24	0.098	0.247	0.092	0.246	0.103	0.259	0.222	0.389	0.114	0.272	0.107	0.280	0.108	0.284	0.115	0.275
	48	0.158	0.319	0.161	0.322	0.167	0.328	0.284	0.445	0.193	0.358	0.162	0.327	0.175	0.424	0.168	0.330
	168	0.183	0.346	0.187	0.355	0.207	0.375	1.522	1.191	0.236	0.392	0.239	0.422	0.396	0.504	1.224	0.763
	336	0.222	0.387	0.215	0.369	0.230	0.398	1.860	1.124	0.590	0.698	0.445	0.552	0.468	0.593	1.549	1.820
	720	0.269	0.435	0.257	0.421	0.273	0.463	2.112	1.436	0.683	0.768	0.658	0.707	0.659	0.766	2.735	3.253
ETTh ₂	24	0.093	0.240	0.099	0.241	0.102	0.255	0.263	0.437	0.155	0.307	0.098	0.263	3.554	0.445	0.199	0.381
	48	0.155	0.314	0.159	0.317	0.169	0.348	0.458	0.545	0.190	0.348	0.163	0.341	3.190	0.474	0.304	0.462
	168	0.232	0.389	0.235	0.390	0.246	0.422	1.029	0.879	0.385	0.514	0.255	0.414	2.800	0.595	2.145	1.068
	336	0.263	0.417	0.258	0.423	0.267	0.437	1.668	1.228	0.558	0.606	0.604	0.607	2.753	0.738	2.096	2.543
	720	0.277	0.431	0.285	0.442	0.303	0.493	2.030	1.721	0.640	0.681	0.429	0.580	2.878	1.044	3.355	4.664
ETTm ₁	24	0.030	0.137	0.034	0.160	0.065	0.202	0.095	0.228	0.121	0.233	0.091	0.243	0.090	0.206	0.120	0.290
	48	0.069	0.203	0.066	0.194	0.078	0.220	0.249	0.390	0.305	0.411	0.219	0.362	0.179	0.306	0.133	0.305
	96	0.194	0.372	0.187	0.384	0.199	0.386	0.920	0.767	0.287	0.420	0.364	0.496	0.272	0.399	0.194	0.396
	288	0.401	0.554	0.409	0.548	0.411	0.572	1.108	1.245	0.524	0.584	0.948	0.795	0.462	0.558	0.452	0.574
	672	0.512	0.644	0.519	0.665	0.598	0.702	1.793	1.528	1.064	0.873	2.437	1.352	0.639	0.697	2.747	1.174
Weather	24	0.117	0.251	0.119	0.256	0.136	0.279	0.231	0.401	0.131	0.254	0.128	0.274	0.219	0.355	0.302	0.433
	48	0.178	0.318	0.185	0.316	0.206	0.356	0.328	0.423	0.190	0.334	0.203	0.353	0.273	0.409	0.445	0.536
	168	0.266	0.398	0.269	0.404	0.309	0.439	0.654	0.634	0.341	0.448	0.293	0.451	0.503	0.599	2.441	1.142
	336	0.297	0.416	0.310	0.422	0.359	0.484	1.792	1.093	0.456	0.554	0.585	0.644	0.728	0.730	1.987	2.468
	720	0.359	0.466	0.361	0.471	0.388	0.499	2.087	1.534	0.866	0.809	0.499	0.596	1.062	0.943	3.859	1.144
ECL	48	0.239	0.359	0.238	0.368	0.280	0.429	0.971	0.884	0.493	0.539	0.204	0.357	0.879	0.764	0.524	0.595
	168	0.447	0.503	0.442	0.514	0.454	0.529	1.671	1.587	0.723	0.655	0.315	0.436	1.032	0.833	2.725	1.273
	336	0.489	0.528	0.501	0.552	0.514	0.563	3.528	2.196	1.212	0.898	0.414	0.519	1.136	0.876	2.246	3.077
	720	0.540	0.571	0.543	0.578	0.558	0.609	4.891	4.047	1.511	0.966	0.563	0.595	1.251	0.933	4.243	1.415
	960	0.582	0.608	0.594	0.638	0.624	0.645	7.019	5.105	1.545	1.006	0.657	0.683	1.370	0.982	6.901	4.264
Count		32		12		0		0		0		6		0		0	

实验结果

- 与没有稀疏型相比没有势
- 与其他模型较大

Methods		Informer		Informer [†]		LogTrans		Reformer		LSTMa		LSTnet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh ₁	24	0.577	0.549	0.620	0.577	0.686	0.604	0.991	0.754	0.650	0.624	1.293	0.901
	48	0.685	0.625	0.692	0.671	0.766	0.757	1.313	0.906	0.702	0.675	1.456	0.960
	168	0.931	0.752	0.947	0.797	1.002	0.846	1.824	1.138	1.212	0.867	1.997	1.214
	336	1.128	0.873	1.094	0.813	1.362	0.952	2.117	1.280	1.424	0.994	2.655	1.369
	720	1.215	0.896	1.241	0.917	1.397	1.291	2.415	1.520	1.960	1.322	2.143	1.380
ETTh ₂	24	0.720	0.665	0.753	0.727	0.828	0.750	1.531	1.613	1.143	0.813	2.742	1.457
	48	1.457	1.001	1.461	1.077	1.806	1.034	1.871	1.735	1.671	1.221	3.567	1.687
	168	3.489	1.515	3.485	1.612	4.070	1.681	4.660	1.846	4.117	1.674	3.242	2.513
	336	2.723	1.340	2.626	1.285	3.875	1.763	4.028	1.688	3.434	1.549	2.544	2.591
	720	3.467	1.473	3.548	1.495	3.913	1.552	5.381	2.015	3.963	1.788	4.625	3.709
ETTm ₁	24	0.323	0.369	0.306	0.371	0.419	0.412	0.724	0.607	0.621	0.629	1.968	1.170
	48	0.494	0.503	0.465	0.470	0.507	0.583	1.098	0.777	1.392	0.939	1.999	1.215
	96	0.678	0.614	0.681	0.612	0.768	0.792	1.433	0.945	1.339	0.913	2.762	1.542
	288	1.056	0.786	1.162	0.879	1.462	1.320	1.820	1.094	1.740	1.124	1.257	2.076
	672	1.192	0.926	1.231	1.103	1.669	1.461	2.187	1.232	2.736	1.555	1.917	2.941
Weather	24	0.335	0.381	0.349	0.397	0.435	0.477	0.655	0.583	0.546	0.570	0.615	0.545
	48	0.395	0.459	0.386	0.433	0.426	0.495	0.729	0.666	0.829	0.677	0.660	0.589
	168	0.608	0.567	0.613	0.582	0.727	0.671	1.318	0.855	1.038	0.835	0.748	0.647
	336	0.702	0.620	0.707	0.634	0.754	0.670	1.930	1.167	1.657	1.059	0.782	0.683
	720	0.831	0.731	0.834	0.741	0.885	0.773	2.726	1.575	1.536	1.109	0.851	0.757
ECL	48	0.344	0.393	0.334	0.399	0.355	0.418	1.404	0.999	0.486	0.572	0.369	0.445
	168	0.368	0.424	0.353	0.420	0.368	0.432	1.515	1.069	0.574	0.602	0.394	0.476
	336	0.381	0.431	0.381	0.439	0.373	0.439	1.601	1.104	0.886	0.795	0.419	0.477
	720	0.406	0.443	0.391	0.438	0.409	0.454	2.009	1.170	1.676	1.095	0.556	0.565
	960	0.460	0.548	0.492	0.550	0.477	0.589	2.141	1.387	1.591	1.128	0.605	0.599
Count		33		14		1		0		0		2	

实验结果

- 适用于更长的输入
- 比其他模型效果显著提升

Prediction length		336			720		
Encoder's input		336	720	1440	720	1440	2880
Informer	MSE	0.249	0.225	0.216	0.271	0.261	0.257
	MAE	0.393	0.384	0.376	0.435	0.431	0.422
Informer [†]	MSE	0.241	0.214	-	0.259	-	-
	MAE	0.383	0.371	-	0.423	-	-
LogTrans	MSE	0.263	0.231	-	0.273	-	-
	MAE	0.418	0.398	-	0.463	-	-
Reformer	MSE	1.875	1.865	1.861	2.243	2.174	2.113
	MAE	1.144	1.129	1.125	1.536	1.497	1.434

¹ Informer[†] uses the canonical self-attention mechanism.

² The '-' indicates failure for the out-of-memory.

实验结果

– 没有蒸馏

Prediction length		336					480				
Encoder's input		336	480	720	960	1200	336	480	720	960	1200
Informer [†]	MSE	0.249	0.208	0.225	0.199	0.186	0.197	0.243	0.213	0.192	0.174
	MAE	0.393	0.385	0.384	0.371	0.365	0.388	0.392	0.383	0.377	0.362
Informer [‡]	MSE	0.229	0.215	0.204	-	-	0.224	0.208	0.197	-	-
	MAE	0.391	0.387	0.377	-	-	0.381	0.376	0.370	-	-

¹ Informer[‡] removes the self-attention distilling from Informer[†].

² The '-' indicates failure for the out-of-memory.

– 没有生成式编码

Prediction length		336					480				
Prediction offset		+0	+12	+24	+48	+72	+0	+48	+96	+144	+168
Informer [‡]	MSE	0.207	0.209	0.211	0.211	0.216	0.198	0.203	0.203	0.208	0.208
	MAE	0.385	0.387	0.391	0.393	0.397	0.390	0.392	0.393	0.401	0.403
Informer [§]	MSE	0.201	-	-	-	-	0.392	-	-	-	-
	MAE	0.393	-	-	-	-	0.484	-	-	-	-

¹ Informer[§] replaces our decoder with dynamic decoding one in Informer[‡].

² The '-' indicates failure for the unacceptable metric results.

- 优势
 - Informer 算法在Transformer的基础上，从三个方面均进行了改进。
 - 同时可应用于单变量时间序列和多变量时间序列预测任务
 - 在长时间序列预测问题中表现更优
- 劣势
 - 在多元时间序列预测任务中，对稀疏自注意力模块的效果应用不明显
 - 缺乏对多元时间序列内相关性的考虑



研究方向

- 研究方向

- 对数据模式和特征的处理，如考虑季节性、周期性等趋势模式变化
- 引入图神经网络，考虑序列间的互相关性（时空序列）
- 预训练大模型，借鉴NLP中的应用
- 高维、长时间的工业规模时间序列数据，自动发现具有存储效率和计算效率的Transformer架构具有重要的现实意义

- [1] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In ICLR, 2022.
- [2] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time series forecasting. In AAAI, 2021
- [3] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. arXiv preprint arXiv:2201.12740, 2022
- [4] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, Liang Sun. Transformers in Time Series: A Survey. arXiv:2202.07125v3

谢谢！

大成若缺，其用不弊。大盈
若冲，其用不穷。大直若屈。
大巧若拙。大辩若讷。静胜
躁，寒胜热。清静为天下正。

