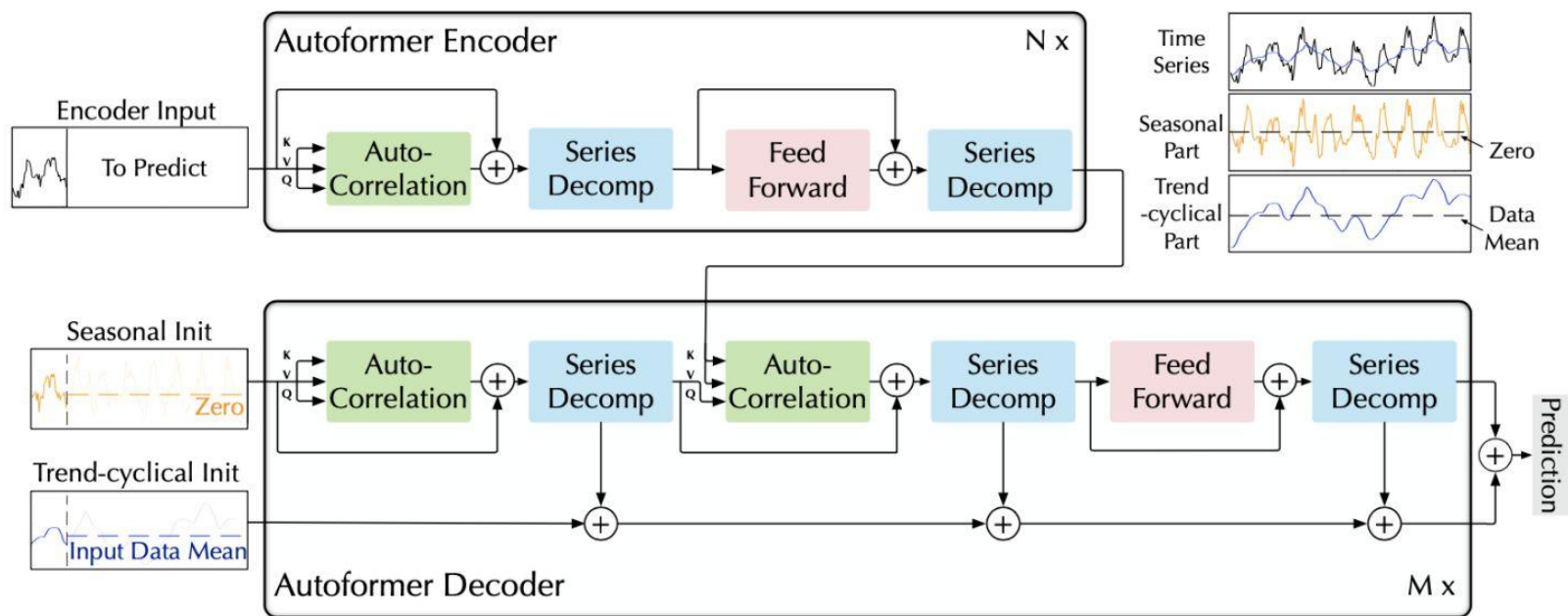


论文分享之 *Autoformer*



• 2022.05.18

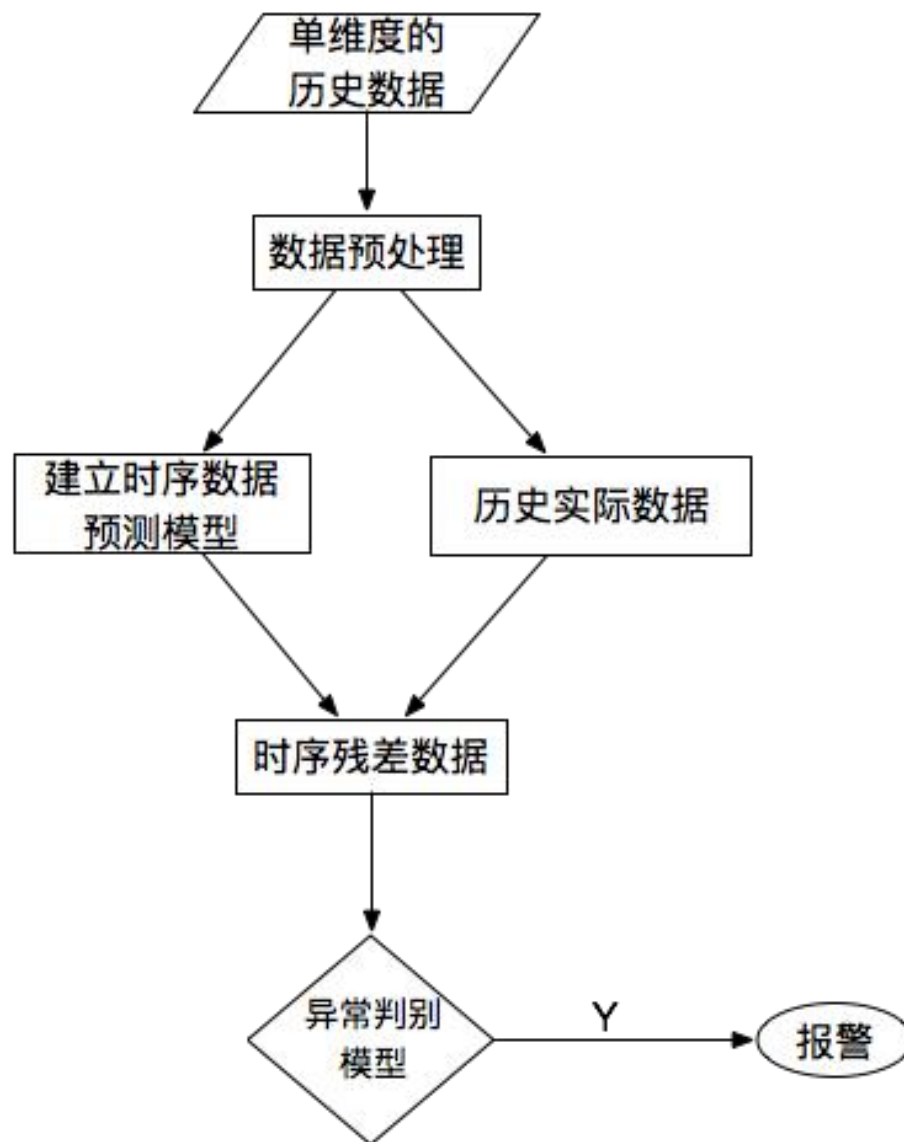
1. 预备知识
2. 模型简介
3. 数据
4. 模型架构及内部组件
5. 代码细节
6. 模型分析
7. 总结

业务关联

数据种类：

1. 日志
2. 时间序列

基于预测的时间序列异常检测



预备知识

- **时间序列 (time series, TS)** 是一组按照时间发生先后顺序进行排列的数据点序列。通常一组时间序列的时间间隔为一恒定值（如 1s, 1min, 1h, 1d），因此时间序列亦可作为离散时间数据进行分析处理。**任何按照时间顺序观察的事物都是时间序列。**
- **时间序列预测：**在预测时间序列数据时，目的是估计观测序列将如何持续到未来。图 [1.1](#) 显示从1992年到2010年第二季度的澳大利亚季度啤酒产量。

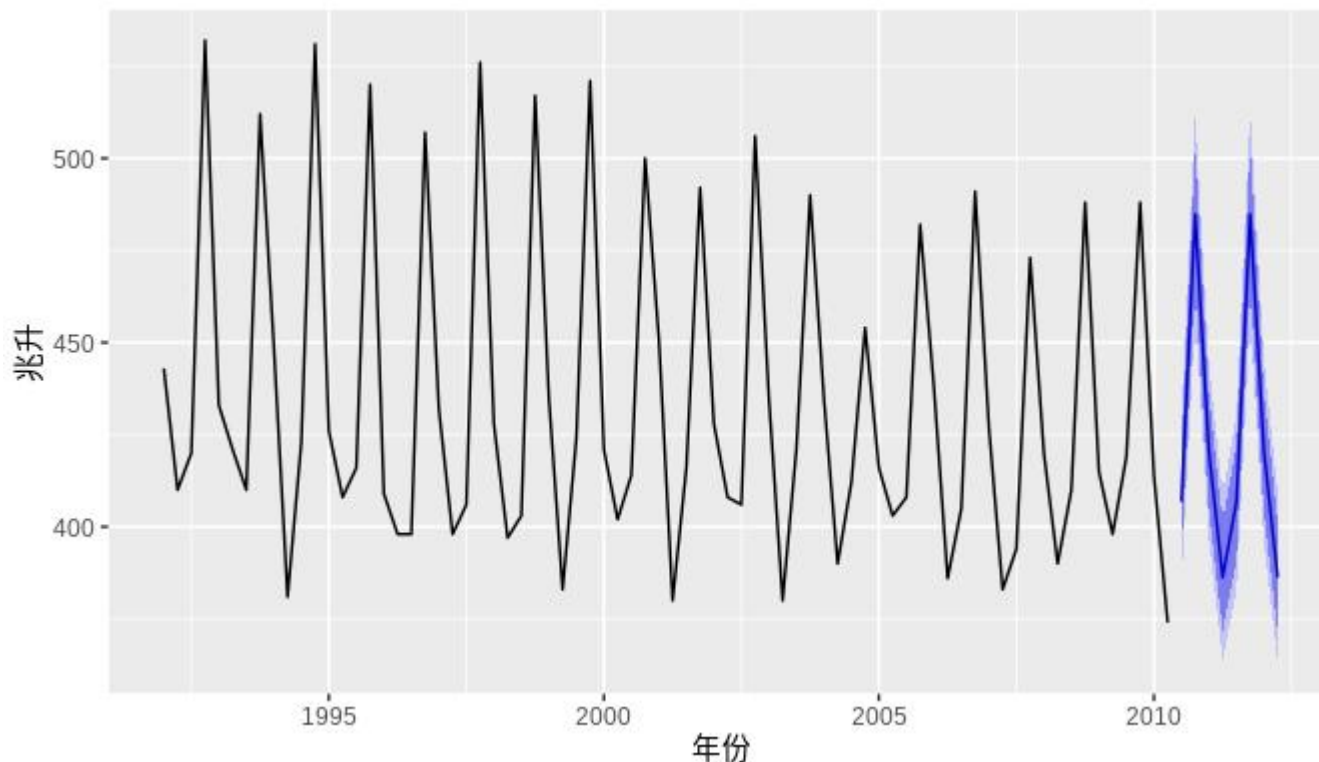


图 1: 澳大利亚季度啤酒产量： 1992年第一季度至2010年第二季度, 及对未来两年的预测

预备知识

时间序列模式：

趋势

当一个时间序列数据长期增长或者长期下降时，表示该序列有趋势。在某些场合，趋势代表着“转换方向”。例如从增长的趋势转换为下降趋势。

季节性

当时间序列中的数据受到季节性因素（例如一年的时间或者一周的时间）的影响时，表示该序列具有季节性。季节性总是一个已知并且固定的频率。

周期性

当时间序列数据存在不固定频率的上升和下降时，表示该序列有周期性。这些波动经常由经济活动引起，并且与“商业周期”有关。周期波动通常至少持续两年。

当数据的波动是无规律时，表示序列存在周期性；如果波动的频率不变并且与固定长度的时间段有关，表示序列存在季节性。一般而言，周期的长度较长，并且周期的波动幅度也更大。

预备知识

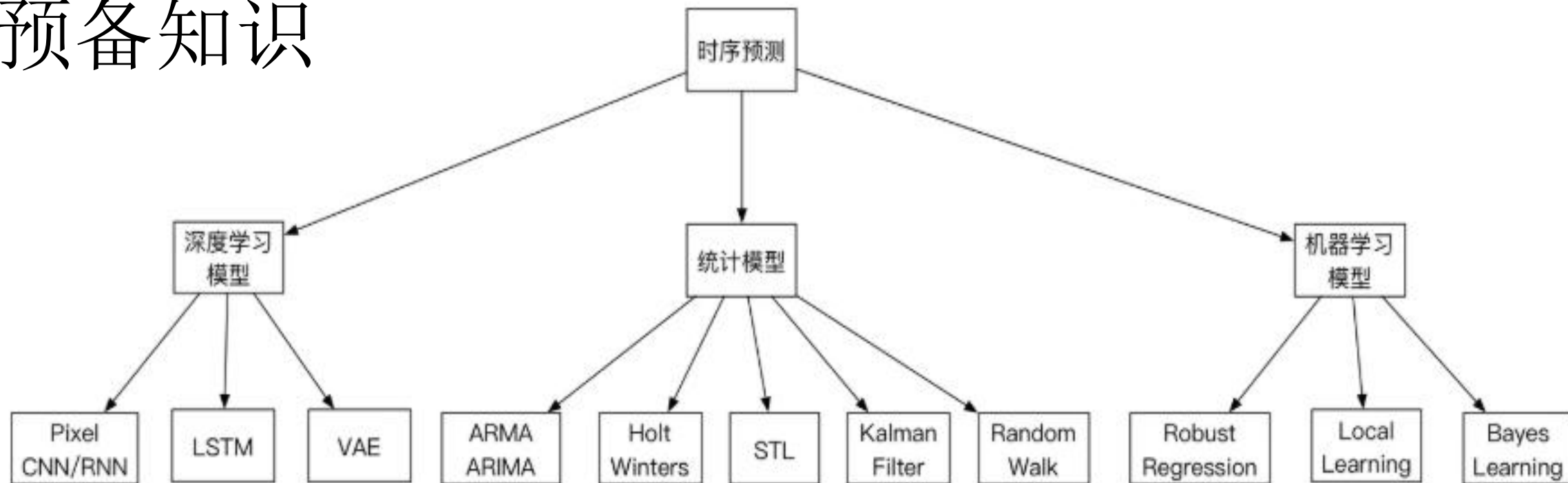
时间序列分解：

假设一条时间序列是由多种成分相加得来，那么它可以写为如下形式： $y_t = S_t + T_t + R_t$,

y_t 表示时间序列数据， S_t 表示季节项， T_t 表示趋势-周期项， R_t 表示残差项。

对于预测任务，在预测未来序列 之前，总是使用分解作为历史序列的预处理。

预备知识



1. 基于RNN的模型问题： 逐个处理，效率慢，无法并行运算，训练速度慢，无法建模长期依赖，容易梯度消失；

2. 基于Transformer 的模型：

优点:并行计算，训练速度快，可以建模长期依赖

缺点: *point-wise* 连接，时间复杂度高 $O(N^2)$

预备知识

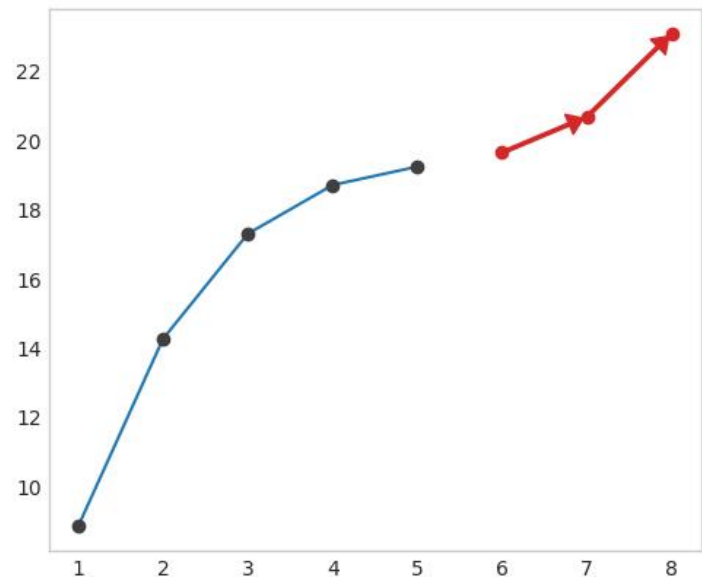
基本架构：编码器-解码器

编码器：将输入 $N \times d$ 表示为向量

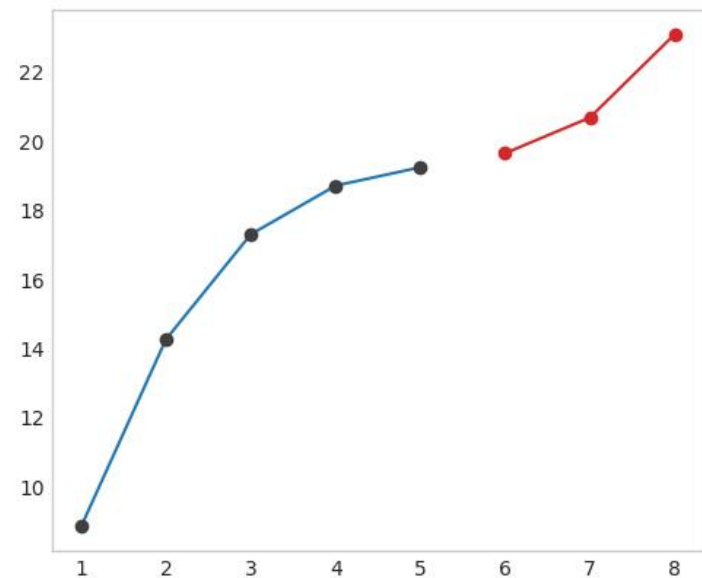
解码器：编码器的输入转为输出 $M \times d$

1. 自回归输出(误差累积)
2. 多输出模型

Recursive Strategy



MultiOutput Model



Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

- 目的：长期序列预测问题：待预测的序列长度远远大于输入长度，即基于有限的信息预测更长远的未来。
- 特点：
 1. 将时间序列分解嵌入到深度学习模型中（以前的方法常用时间序列分解用作预处理），效果：能够从复杂时间模式中分解出可预测性更强的组分。
 2. 更新了Transformer：基于随机过程理论，提出Auto-Correlation机制，代替point-wise连接的self-attention机制。效果：实现序列级（series-wise）连接和降低复杂度，打破信息利用瓶颈。
 3. 效果：在长期预测问题中，Autoformer在能源、交通、经济、气象、疾病五大时序领域大幅超越之前SOTA，实现38%的相对效果提升。

数据(以天气数据集为例)

数据大小：**52696条数据，每条数据22个特征，第一个特征为日期。**如下：

date	p (mbar)	T (degC)	Tpot (K)	Tdew (deg C)	rh (%)	VPmax (mbar)	VPact (mbar)	VPdef (mbar)	sh (g/kg)	H2OC (mmol/ mol)	rho (g/m* *3)	wv (m/s)	max. wv (m/s)	wd (deg)	rain (mm)	raining (s)	SWDR (W/m 锺?)	PAR (锺晒 ol/m 锺?s)	max. PAR (锺晒 ol/m 锺?s)	Tlog (degC)	OT
2020/1/1 0:10	1008.8 9	0.71	273.18	-1.33	86.1	6.43	5.54	0.89	3.42	5.49	1280. 62	1.02	1.6	224.3	0	0	0	0	0	11.45	428.1

模型架构(Encoder-Decoder)

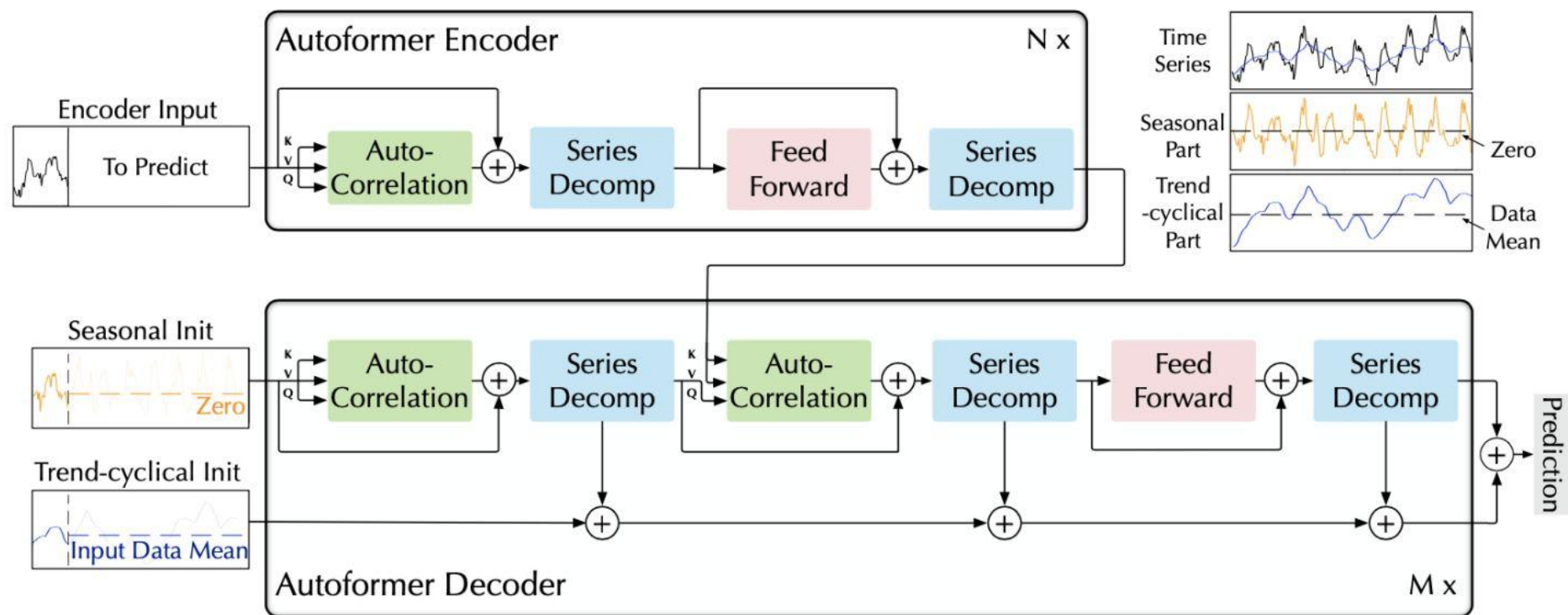
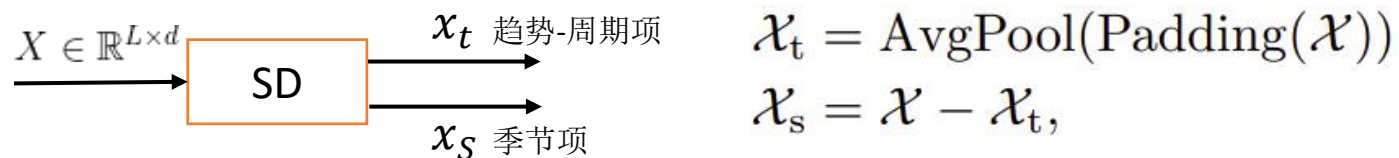


图2: Autoformer 架构。Encoder 通过序列分解模块（蓝色模块）消除了 long-term 趋势-周期项，并专注于季节项建模。Decoder 逐步累加从隐藏变量中提取的趋势-周期项。encoder-decoder Auto-Correlation（Decoder 中的中间绿色块）利用来自 Encoder 的过去季节项信息。

模型组件之序列分解模块(*Series decomposition*)

$length-L$ 的输入序 列 $X \in \mathbb{R}^{L \times d}$ d 表示每个时间步有 d 维数据(天气时21维度(不考虑时间))



采用 *moving average* 以平滑周期性波动并突出 *long-term trends*。 *Padding* 操作用来保持序列长度不变

模型组件之自相关模块(*Auto-Correlation*)

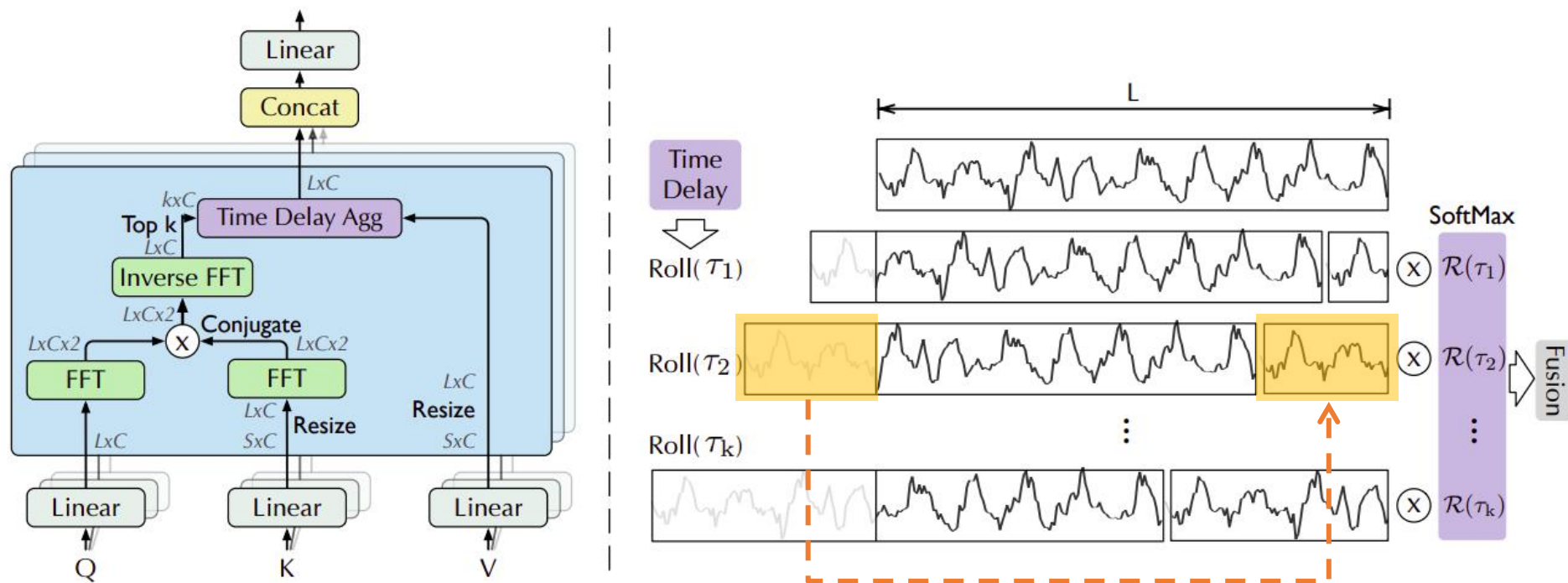
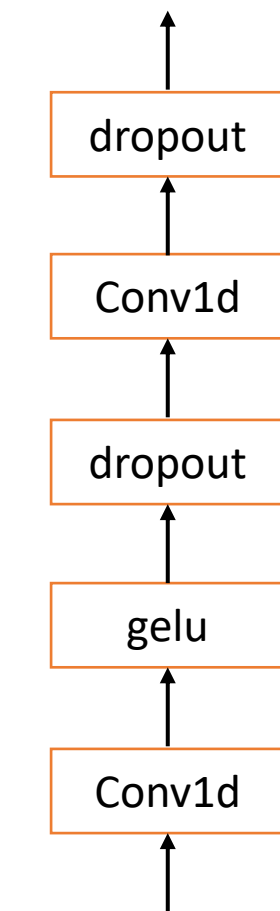
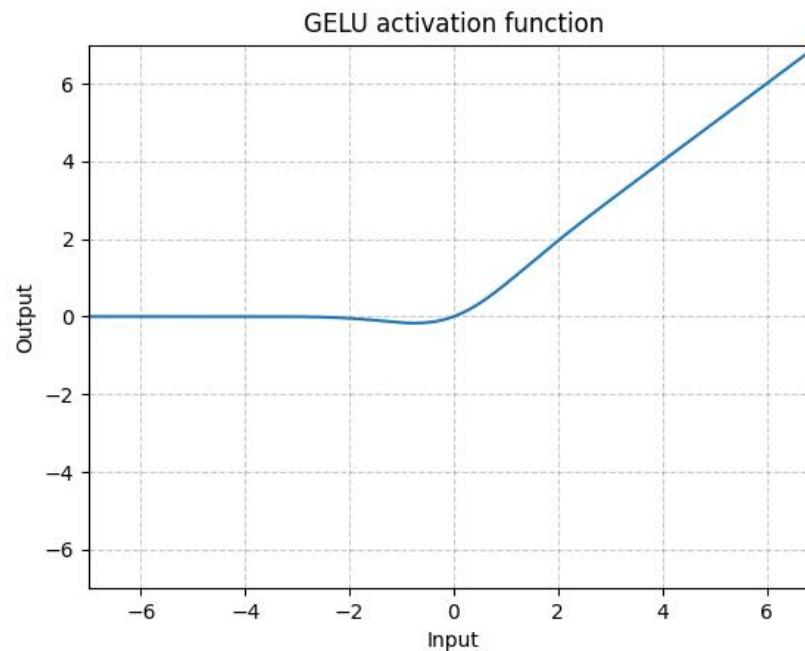


图2: **Auto-Correlation** (左) 和**时间延迟聚合 (Time Delay Aggregation)** (右)。我们利用快速傅里叶变换来计算自相关 $\mathcal{R}(\tau)$ ，它反映了时延相似性。然后基于选定的延迟 τ 将相似的子过程滚动到相同的索引，并通过 $\mathcal{R}(\tau)$ 聚合。

模型组件之Feed Forward



Feed Forward



参考: <https://pytorch.org/docs/stable/generated/torch.nn.GELU.html>
<https://zhuanlan.zhihu.com/p/349492378>

代码细节(<https://github.com/thuml/Autoformer>)

4 表示时间如2020/1/1 0:10 的编码结果

设 1. batch size = 32,

2. 输入序列长度 = 96, 表示96个时刻的数据

3. 待预测序列长度 = 96,

4. 每个时间步特征数目 = 21(天气数据集, 不考虑时间)

5. 嵌入维度大小 = 512

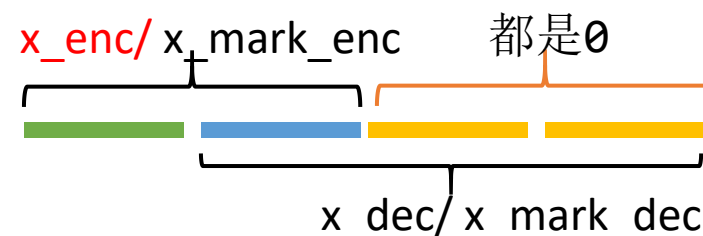
模型输入:

1. 编码器输入数据序列 x_{enc} 维度: (32, 96, 21)

2. 编码器输入时间戳序列 x_{mark_enc} 维度: (32, 96, 4)

3. 解码器输入数据序列 x_{dec} 维度: (32, 48 + 96, 21)

4. 解码器输入时间戳序列 x_{mark_dec} 维度: (32, 48 + 96, 4)

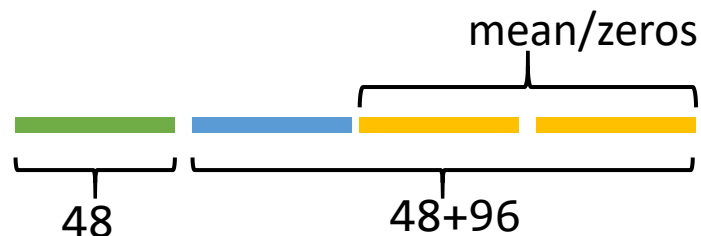


x_{enc} 获取均值 $\text{mean}(32, 96, 21)$, 长度96 为预测序列长度。

设置 zeros 序列(32, 96, 21)

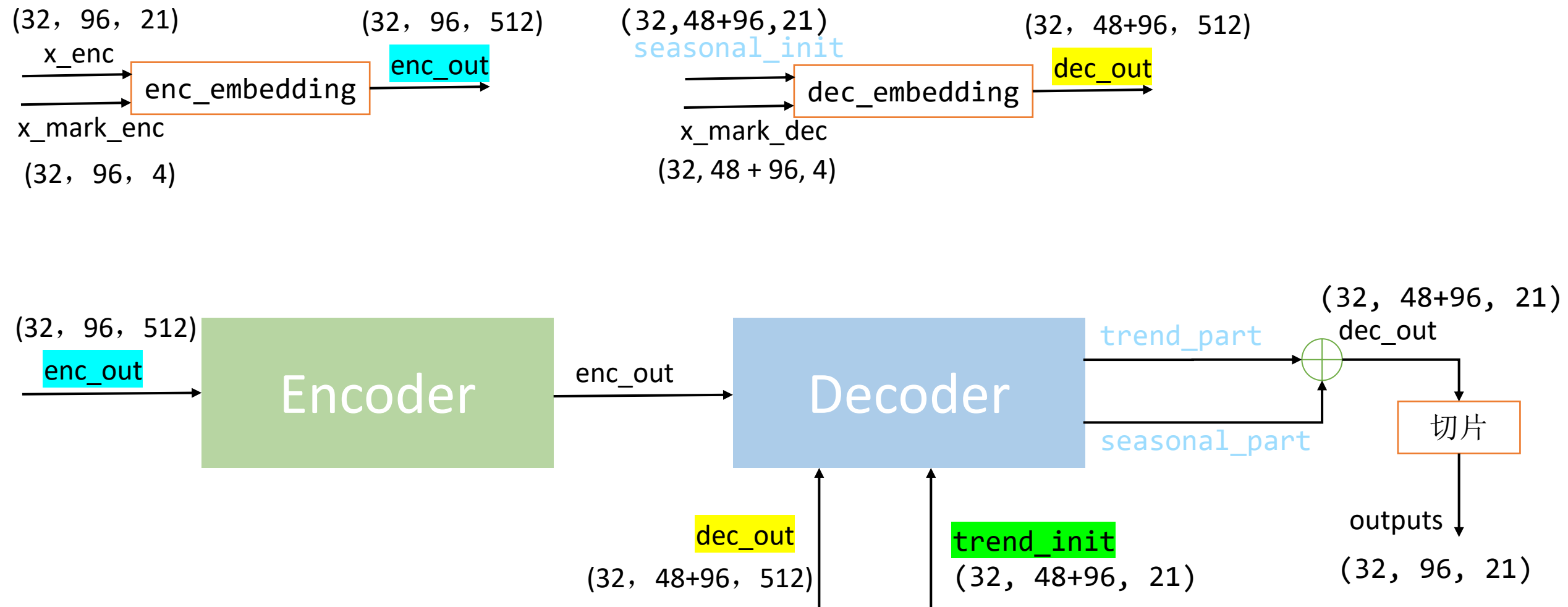
x_{enc} 序列分解为: 季节项初始值 $\text{seasonal_init}(32, 96, 21)$ 与 趋势-周期项初始值 $\text{trend_init}(32, 96, 21)$ 。

解码器输入: 1. 趋势-周期项 $\text{trend_init}(32, 48+96, 21)$ (通过cat 上述 mean 得到)

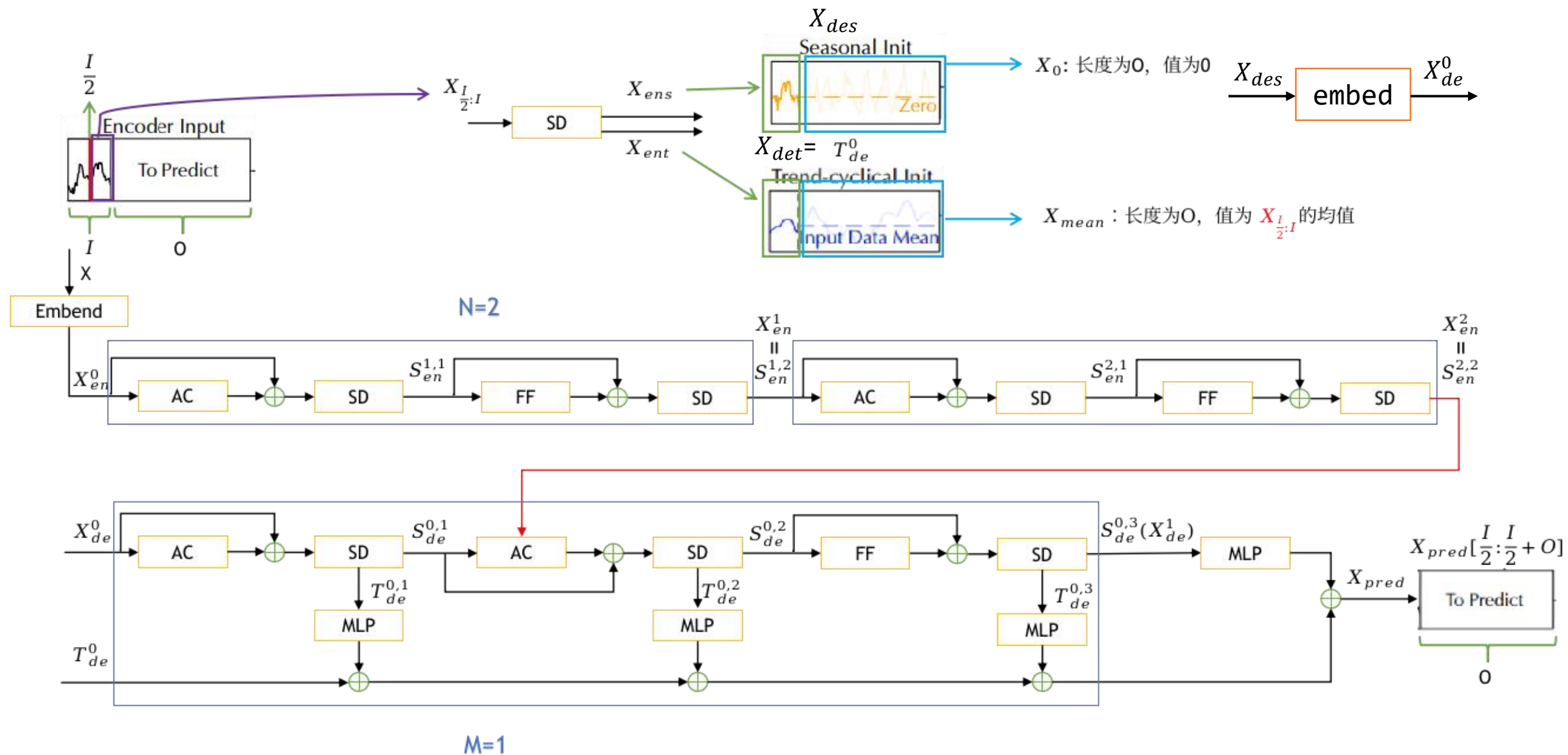


2. 季节项 $\text{seasonal_init}(32, 48+96, 21)$ (通过cat 上述 zeros 得到)

代码细节(<https://github.com/thuml/Autoformer>)

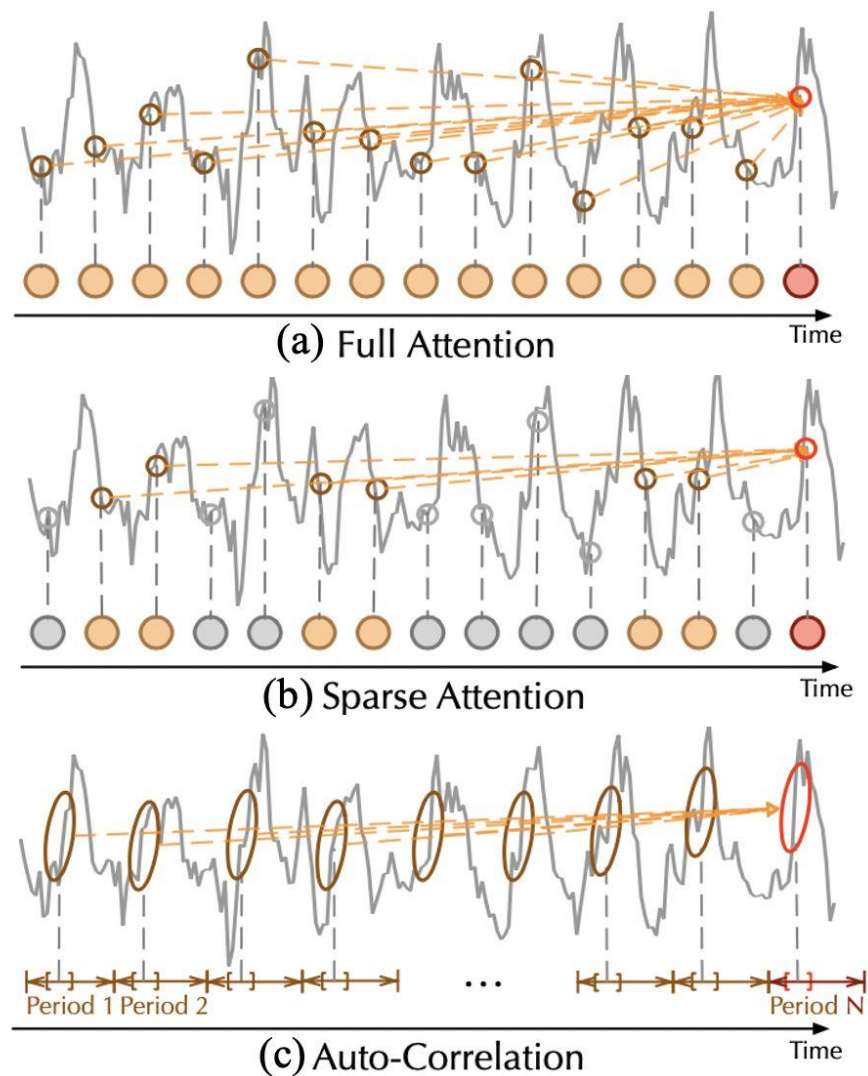


代码细节(<https://github.com/thuml/Autoformer>)



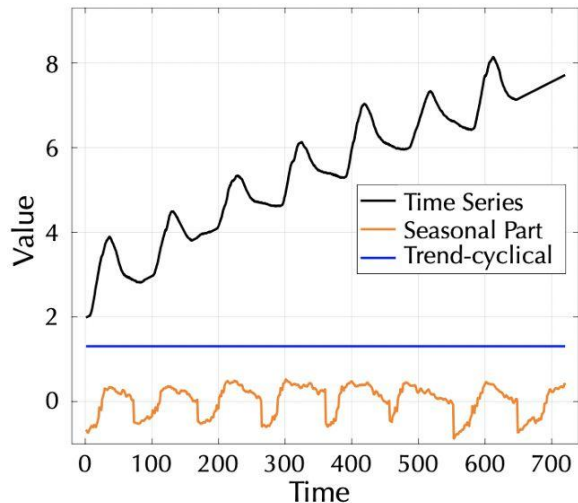
模型分析

对比分析： 相比于之前的注意力机制或者稀疏注意力机制，自相关机制（*Auto-Correlation Mechanism*）实现了序列级的高效连接，从而可以更好的进行信息聚合，打破了信息利用瓶颈

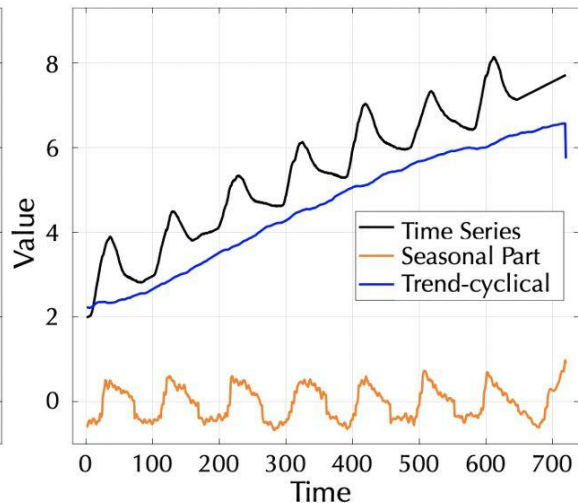


模型分析

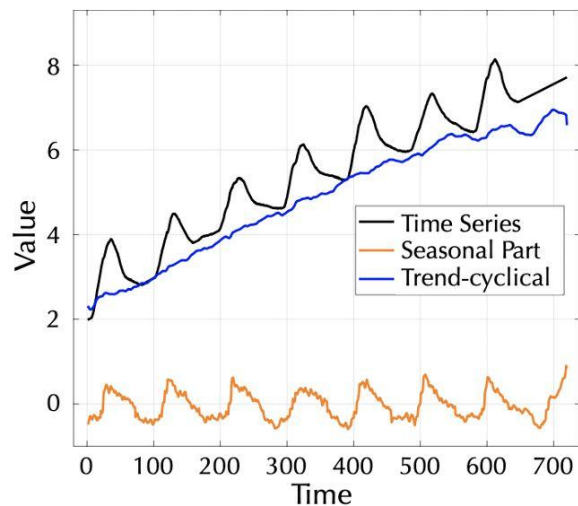
渐进式分解效果： 随着序列分解单元的数量增加，模型的学到的趋势项会越来越接近数据真实结果，周期项可以更好的捕捉序列变化情况，这验证了渐进式分解的作用。



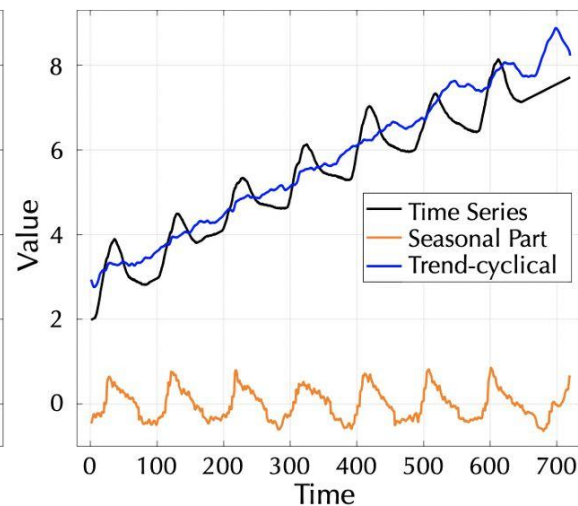
(a) Without decomposition block



(b) One decomposition block



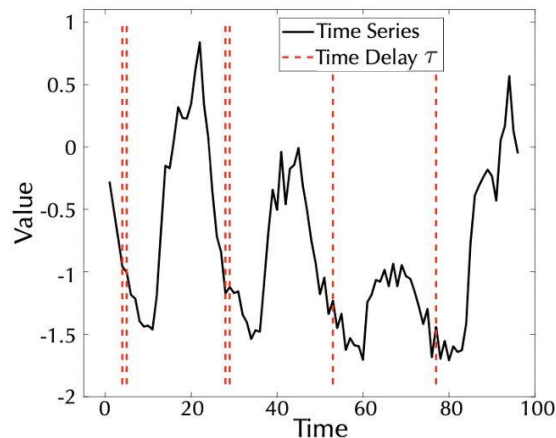
(c) Two decomposition blocks



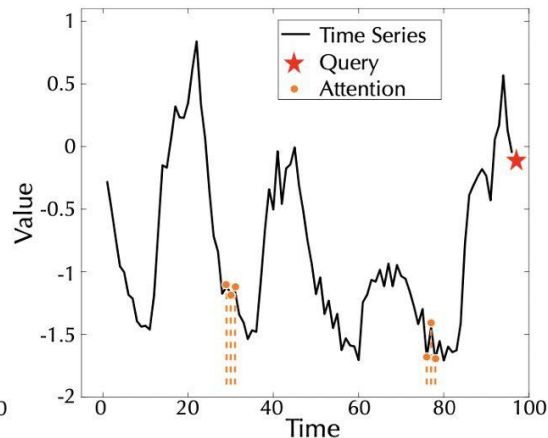
(d) Three decomposition blocks

模型分析

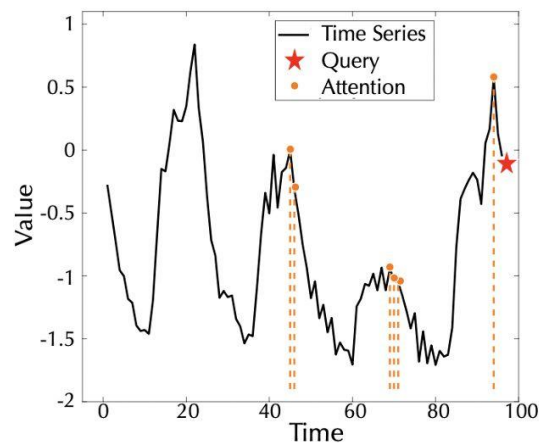
时序依赖可视化：通过对比可以发现，*Autoformer*中自相关机制可以正确发掘出每个周期中的下降过程，并且**没有误识别和漏识别**，而自注意力机制存在错误和缺漏。



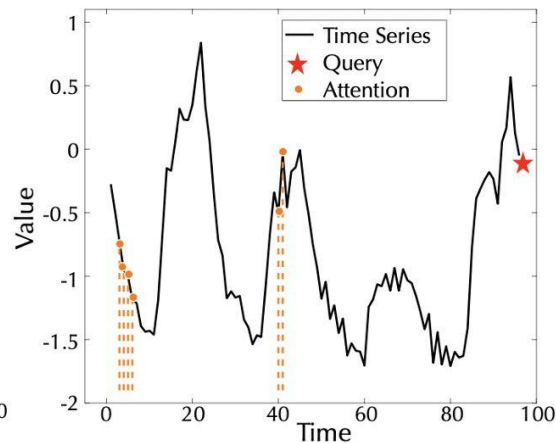
(a) Auto-Correlation



(b) Full Attention



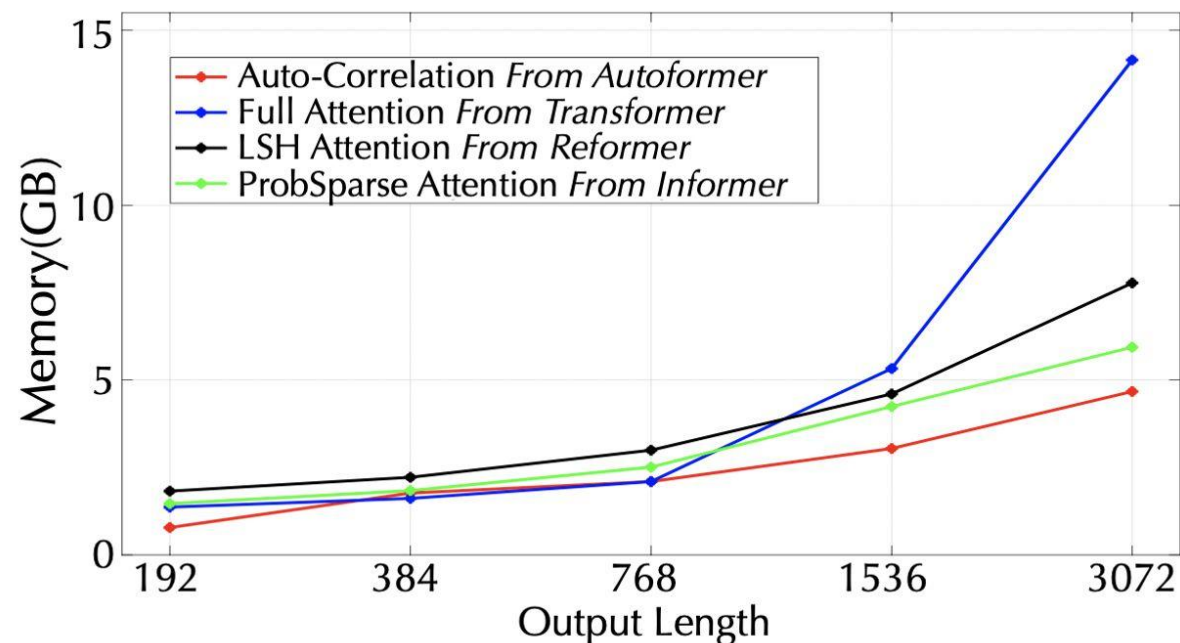
(c) LSH Attention



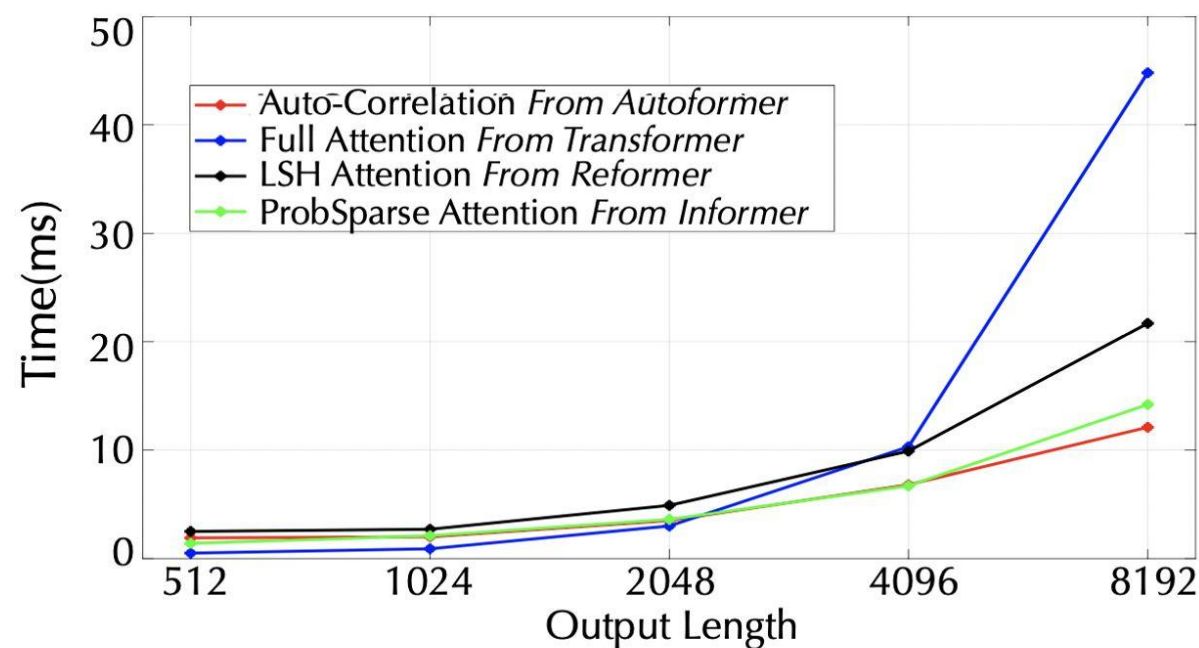
(d) ProbSparse Attention

模型分析

效率分析： 在显存占用和运行时间两个指标上，自相关机制均表现出了优秀的空间时间效率，两个层面均超过自注意力机制，表现出高效的 $\mathcal{O}(L \log L)$ 复杂度。



(a) Memory Efficiency Analysis



(b) Running Time Efficiency Analysis

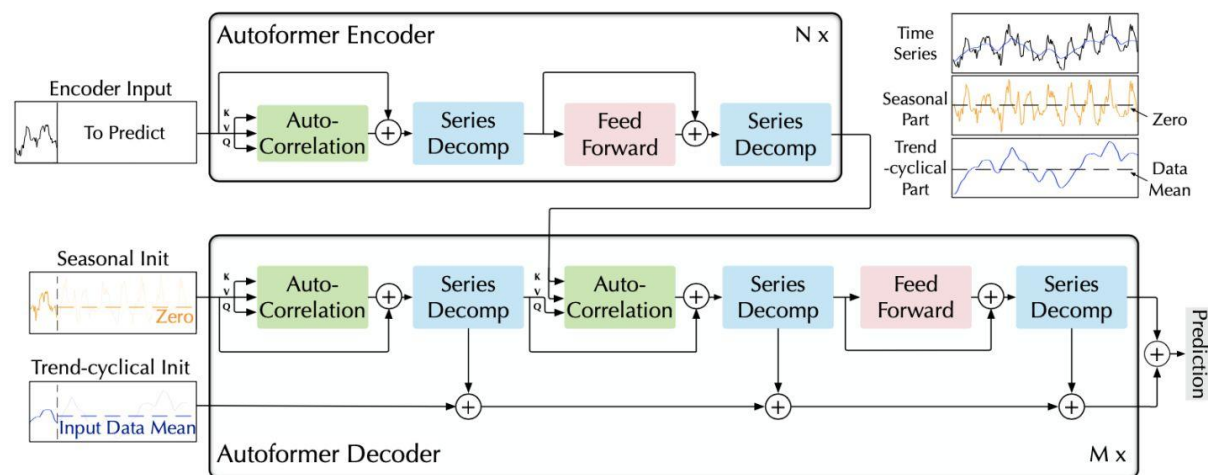
总结

Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

	基于Transformer 的预测模型	Autoformer
应对复杂 时间模式	难以直接发现 可靠的时间依赖 X	深度分解架构 得到可预测的组分 √
长序列 高效处理	稀疏注意力机制带来 信息利用瓶颈 X	自相关机制 高效、序列级连接 √

问题

Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting



1. 解码器输出部分：建模的趋势项和季节项的和 被监督信息约束，因此趋势项往往需要多个分解模块才能建模。
2. 自相关机制是用来降低self-attention的复杂度，因此可以替换为别的模型，如 线性 Transformer。
3. 序列分解模块用的简单的移动平均，可以优化为加权平均。
4.