初步版本:请勿引用
AAAI 数字图书馆将包含已出版的
会议后一段时间的版本

Informer: Beyond Efficient Transformer for Long Sequence 时间序列预测

周浩一，[1] 张上航，[2] 彭洁琪，[1] 张帅，[1] 李建新，[1] 熊辉，[3] 张万才，[4]
[1] 北京航空航天大学 加州大学伯克利分校[2] 罗格斯大学[3]
[4] 北京国网富达科技发展公司
{zhouhy, pengjq, zhangs, lijx}@act.buaa.edu.cn, shz@eecs.berkeley.edu, xionghui@gmail.com,
zhangwancai@sdee.sgcc.com.cn

抽象的

许多现实世界的应用程序需要长时间的预测
时序时间序列,如用电量计划宁。长序列时间序列预测 (LSTF) 需求

模型的高预测能力,即能力
捕获精确的远程依赖耦合
有效地输出和输入。最近的研究表明,
Transformer 提高预测能力的潜力。
但是,Transformer 存在几个严重的问题
阻止它直接适用于 LSTF,例如
二次时间复杂度,高内存使用,以及编码器-解码器架构的限制。为了解决
这些问题,我们设计了一个高效的基于变压器的

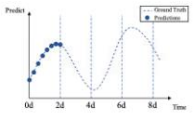
LSTF 模型,名为 Informer,具有三个独特的特征:(i) ProbSparse Self-attention 机制,它
在时间复杂度和内存使用上达到 O(Llog L),
并且在序列依赖性方面具有相当的性能
结盟。 (ii) 自注意力提炼通过将级联层输入减半来突出主导注意力,并且
有效地
处理极长的输入序列。 (iii) 生成
风格解码器,虽然概念上很简单,但可以预测长
一次前向操作的时间序列序列,而不是
循序渐进的方式,极大地改善了推理
长序列预测的速度。广泛的实验
在四个大规模数据集上表明 Informer 显着优于现有方法并提供了新的

LSTF 问题的解决方案。

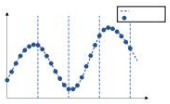
介绍

时间序列预测是许多领域的关键要素
域,例如传感器网络监控（Papadimitriou
和 Yu 2006）、能源和智能电网管理、经济和金融（Zhu 和 Shasha 2002)和疾病
传播分析（Matsubara et al. 2014）。在这些场景中,我们可以利用大量的时间序列
过去行为的数据,以进行长期预测,
即长序列时间序列预测（LSTF）。然而,现有的方法是在有限的问题下设计的

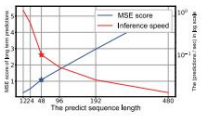
设置,例如预测 48 个点或更少（Hochreiter 和
施米德胡伯 1997;李等人。 2018;于等人。 2017;刘等人。
2019;秦等人。 2017;温等人。 2017）。越来越
长序列使模型的预测能力
有些人认为这种趋势正在保持



(a) 短序列
预测。



(b) 长序列
预测。



(c) 运行 LSTM
长序列。

图 1:(a) 短序列预测仅揭示
不远的将来。 (b) 长序列时间序列预测可以
为更好的政策规划和
投资保护。 (c) 现有方法的预测能力限制了长序列的性能,即

从长度 = 48 开始,MSE 上升到无法接受的高度，
推理速度迅速下降。

LSTF 研究。作为一个经验例子,图（1)显示了
在真实数据集上预测结果,其中 LSTM 网络从短期内预测电力变压器站的每小时温度（12 分,0.5

天)到长期（480点,20天）。这
当预测时,整体性能差距很大
长度大于 48 个点（图（1（c)中的实心星）。
MSE 分数上升到不能令人满意的性能,推理速度急剧下降,LSTM 模型失败。

LSTF 的主要挑战是增强预测能力以满足日益增长的序列需求,这需要 (a)
非凡的远程对齐能力和 (b) 对长序列输入和输出的有效操作。最近，
Transformer 模型在捕获远程依赖方面表现出比

RNN 模型。自注意力机制可以减少
网络信号传播路径进入的最大长度
理论上最短的 O(1) 并避免循环结构，
Transformer 显示出解决 LSTF 问题的巨大潜力。但另一方面,自注意力机制

违反要求 (b),因为它在 L 长度输入/输出上的 L 二次计算和内存消耗。

一些大型的 Transformer 模型倾注了资源和
在 NLP 任务上产生了令人印象深刻的结果（Brown 等人,2020），
但是在数十个 GPU 上的训练和昂贵的部署
成本使这些模型在现实世界的 LSTF 上无法承受
问题。自注意力机制的效率和

Transformer框架成为应用瓶颈
他们解决了 LSTF 问题。因此,在本文中,我们试图回答一个问题:Transformer 模型是否可以改进为
计算、内存和架构的效率也很高
保持较高的预测能力?

Vanilla Transformer (Vaswani et al. 2017) 在求解 LSTF 时存在三个显着限制:

1.self -attention的二次计算。这
自注意力机制的原子操作,即
规范点积,导致时间复杂度和
每层的内存使用量为O(L2)。
2.长时间堆叠层的内存瓶颈
输入。 J编码器/解码器层的堆栈使
总内存使用量为 O(J · L2),这限制了
接收长序列输入的模型可扩展性。
3.预测多头输出的速度骤降。这
vanilla Transformer 的动态解码使得
与基于 RNN 的模型一样慢的逐步推理,
建议在图(1c)中。

有一些关于提高效率的前期工作
自我关注。稀疏变压器(Child et al. 2019),
LogSparse Transformer (Li et al. 2019) 和 Longformer
(Beltagy、Peters 和 Cohan 2020)都使用启发式方法
解决限制 1 并将自我注意机制的复杂性降低到 O(Llog L),其中它们的效率

收益是有限的(Qiu et al. 2019)。改革者(基塔耶夫、凯撒、
和 Levskaya 2019)也通过局部敏感的哈希自注意力实现了 O(Llog L),但它仅适用于极长的序列。最近,Linformer (Wang

等。 2020) 声称线性复杂度 O(L),但该项目
对于实际的长序列,矩阵不能固定,这可能有退化到O(L2)的风险。
Transformer-XL (Dai et al. 2019) 和 Compressive Transformer (Rae et al. 2019) 使用辅助隐藏状态来捕获远程依赖,这可能会放大限制

1、不利打破效率瓶颈。一切
作品主要关注限制1,限制2&3
仍然存在于 LSTF 问题中。为了提高预测能力,我们将解决所有这些问题并实现改进
超出了建议的 Informer 的效率。

为此,我们的工作明确地深入研究了这三个问题。我们研究了自注意力机制的稀疏性,改进了网络组件,并进行了广泛的实验。本文的贡献

总结如下:

- 我们建议 Informer 成功增强 LSTF 问题的预测能力,这验证了
类似 Transformer 的模型在捕获长期之间的个体长期依赖方面的潜在价值
时序时序输出和输入。
- 我们提出 ProbSparse 自注意力机制来
有效地取代了规范的自注意力和
它实现了 O(Llog L) 时间复杂度和
O(Llog L) 内存使用量。
- 我们提出了在J-stacking 层中主导注意力分数的自注意力蒸馏操作
特权
并大幅降低总空间复杂度
O((2 -)Llog L)。
- 我们提出生成式风格解码器来获取长

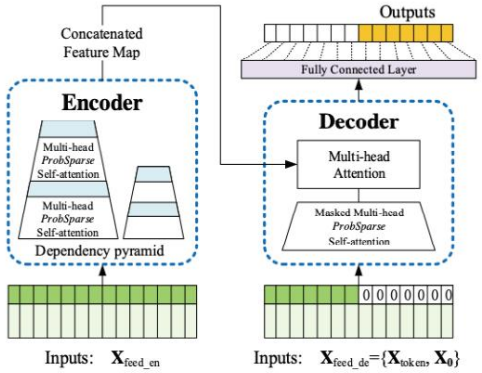


图 2:Informer 模型的整体图。这
左边是Encoder,它接收到海量的长序列
输入(绿色系列)。我们已经替换了规范
使用建议的 ProbSparse 自注意力进行自注意力。
蓝色梯形是自注意力蒸馏操作
提取主导注意力,减少网络规模
尖锐地。层堆叠副本提高了鲁棒性。
对于右侧部分,解码器接收长序列输入,
将目标元素填充为零,测量加权
特征图的注意力组合,并立即以生成方式预测输出元素(橙色系列)。

只需要向前一步的序列输出,
同时避免累积误差扩散
在推理阶段。

初步的

我们首先提供问题定义。滚滚之下
具有固定大小窗口的预测设置,我们有
输入 X Lx = {xt 1,..., xt | xt ∈ Rdx } 在时间 t,
输出是预测对应的序列Yt =
{yt 1,..., yt | yt ∈ Rdy }。 LSTF 问题鼓励一个
比以前的作品更长的输出长度Ly (Cho et al.
2014; Sutskever、Vinyals 和 Le 2014)并且特征维度不限于单变量情况(dy ⩾ 1)。

编码器-解码器架构许多流行的模型是
设计将输入表示 X “编码”为隐藏状态表示Ht并从Ht = {ht 1,..., } “解码”
输出表示Yt 。推论

$$H \lceil h$$

涉及一个名为“动态解码”的逐步过程,
其中解码器计算一个新的隐藏状态ht k+1 从
前一个状态ht 以及来自第 k 个的其他必要输出
step 然后预测第 (k + 1) 个序列yt k+1。

输入表示统一的输入表示是
用于增强时间序列输入的全局位置上下文和局部时间上下文。为避免使描述变得琐碎,我们将详细信息放在附录 B 中。

方法

现有的时间序列预测方法大致可以
分为两类1。经典时间序列模型

[1]限于篇幅,完整的相关工作调查
附录 A 中提供。

els 是时间序列预测的可靠主力（Box et al. 2015; Ray 1990; Seeger et al. 2017; Seeger,
Salinas 和 Flunkert 2016)以及深度学习技术
主要通过以下方式开发编码器-解码器预测范式
使用 RNN 及其变体（Hochreiter 和 Schmidhuber
1997;李等人。 2018;于等人。 2017）。我们提议的告密者
保持编码器 - 解码器架构,同时针对
LSTF 问题。请参阅图（2)的概述和
以下部分了解详细信息。

高效的自注意力机制

(Vaswani et al. 2017) 中的规范自注意力是
在接收元组输入时定义（查询、键、值）
并将缩放的点积执行为 A(Q, K, V) =

Softmax($\frac{QK^{\top}}{\sqrt{d}}$)V,其中 Q ∈ RLQ×d， K ∈ RLK×d，
V ∈ RLV ×d ,d 为输入维度。进一步讨论
自注意力机制,让qi， ki， vi代表第i个
分别排在 Q、K、V 中。按照中的公式
（Tsai et al. 2019），第 i 个查询的注意力被定义为
概率形式的核更平滑：

$$A(qi, K, V) = \sum_j \frac{k(qi, kj)}{\sum_l k(qi, kl)} vj = E_{p(kj|qi)}[vj] \quad (1)$$

其中 $p(kj|qi) = \frac{k(qi,kj)}{\sum_l k(qi,kl)}$ 和 k(qi, kj) 选择
非对称指数核 exp($\frac{qikj}{\sqrt{d}}$ ）。自己
d attention 结合价值并获得输出基于
计算概率 p(kj |qi)。它需要二次方
乘以点积计算和O(LQLK)内存使用时间,这是增强预测的主要缺点

容量。
之前的一些尝试表明,分布
自注意概率具有潜在的稀疏性,并且它们
设计了一些“选择性”计数策略
p(kj |qi)不会显着影响性能。这
Sparse Transformer (Child et al. 2019) 结合了两者
行输出和列输入,其中稀疏性
产生于分离的空间相关性。对数稀疏
Transformer (Li et al. 2019) 注意到了
自我注意并迫使每个细胞注意其先前的
一个指数步长。 Longformer（贝尔塔吉，
Peters 和 Cohan 2020)将前两部作品扩展到更多
复杂的稀疏配置。但是,它们是有限的
从以下启发式方法到理论分析和
用相同的策略处理每个多头自注意力，
这缩小了它的进一步改进。
为了激发我们的方法,我们首先执行定性
对经典自我注意的学习注意模式的评估。 “稀疏”自注意力得分形式

一个长尾分布（详见附录 C），即
很少有点积对引起主要关注,并且
其他的可以忽略。那么,接下来的问题是如何区分它们?

从等式（1)查询稀疏度测量,第 i 个
查询对所有键的注意力被定义为概率 p(kj |qi) ,输出是它与值 v 的组合。

占主导地位的点积对鼓励相应查询的注意力概率分布远离
均匀分布。如果 p(kj |qi)接近于均匀 dis 1
贡献 q(kj |qi) = self-attention 变成了 triv LK ,
值 V 的总和,对于住宅投入来说是多余的。自然地,分布 p 和

q 可用于区分“重要”查询。我们
通过 Kullback-Leibler 散度测量“相似度”
$KL(q||p) = \ln \sum_{l=1}^{LK} e^{qik\,l/\sqrt{d}} \frac{1}{力劲} \sum_{j=1}^{奇克\,j/\sqrt{d}} -$
lnLK。删除常量,我们定义第 i 个查询的
稀疏度测量为

$$M(qi, K) = \ln \sum_{j=1}^{力劲} e^{\frac{哎哎哎}{\sqrt{d}}} \frac{1}{力劲} \sum_{j=1}^{力劲} \frac{奇克_j}{\sqrt{d}}, \quad (2)$$

其中第一项是qi on的 Log-Sum-Exp (LSE)
所有的键,第二项是算术平均值
他们。如果第 i 个查询获得更大的 M(qi, K),则其注意概率 p 更“多样化”,并且有很高的机会
在标题字段中包含主要的点积对
长尾自注意力分布。

ProbSparse Self-attention基于建议的测量,我们通过允许每个键
只关注 u 个主要查询来实现 ProbSparse Self-attention：

$$A(Q, K, V) = Softmax(\frac{\bar{Q}K}{\sqrt{d}})V, \quad (3)$$

其中 $\bar{Q}$ 是与 q 大小相同的稀疏矩阵,它
仅包含稀疏度量 M(q, K) 下的 Top-u 查询。由恒定的采样因子 c 控制，

我们设置 u = c · lnLQ,这使得 ProbSparse self attention 只需要计算 O(lnLQ)点积
每个查询键查找和层内存使用保持 O(LK lnLQ)。

然而,遍历测量 M(qi, K) 的所有查询需要计算每个点积对，

即二次O(LQLK),并且 LSE 操作具有
潜在的数值稳定性问题。受此启发,我们
提出了查询稀疏度测量的近似值。

引理 1.对于每个查询qi ∈ Rd和kj ∈ Rd
键集 K,我们的边界为 lnLK ⩽ M(qi, K) ⩽
$\frac{qik}{\sqrt{d}} - \ln LK \le \max_j\{\frac{1}{力劲}\sum_{j=1}^{力劲}\frac{j}{\sqrt{d}}\} + \ln LK$。当qi ∈ K 时，
立。

根据引理 1（证明在附录 D.1 中给出），我们
建议最大平均测量为

$$\bar{M}(qi, K) = \max_j\{\frac{奇克_j}{\sqrt{d}}\} - \frac{1}{力劲}\sum_{j=1}^{力劲}\frac{奇克_j}{\sqrt{d}}. \quad (4)$$

Top-u 的顺序在边界松弛中成立
与命题 1（参见附录 D.2 中的证明）。在下面
长尾分布,我们只需要随机抽样
$\bar{U} = LQ\,\ln LK$点积对计算 $\bar{M}(qi, K)$，
即用零填充其他对。我们从
它们为 Q。 $\bar{M}(qi, K)$ 中的最大算子不太敏感
为零值并且是数值稳定的。在实践中,查询和键的输入长度通常是等价的,即
LQ = LK = L 这样总的 ProbSparse self-attention
时间复杂度和空间复杂度为 O(LlnL)。

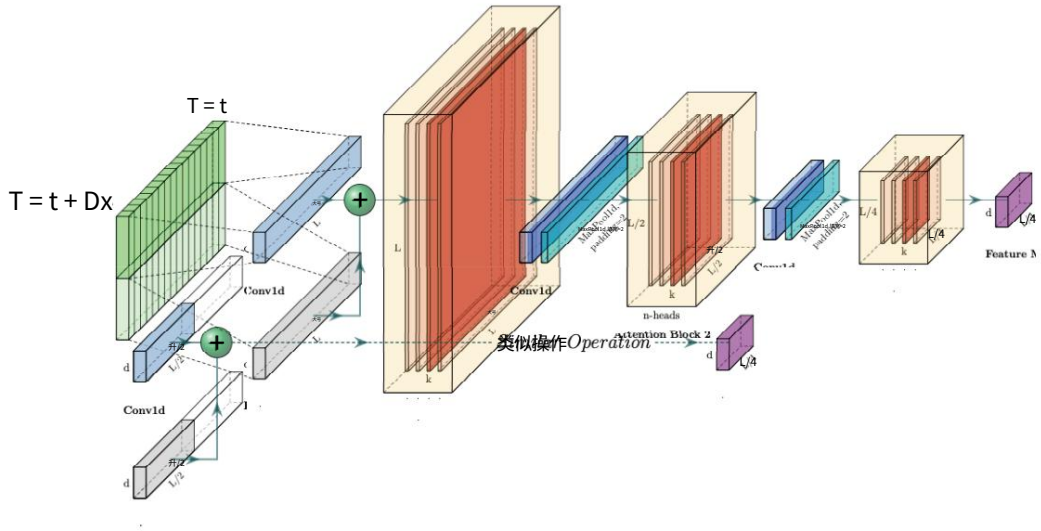


图 3:Informer 编码器的架构。 (1) 每个水平堆栈代表一个单独的编码器副本
在图（2)中。 (2) 上层栈是主栈,接收整个输入序列,而第二层栈占据一半
输入的切片。 (3) 红色层是self-attention机制的点积矩阵,级联减少
在每一层上应用自注意力蒸馏。 (4) 将两个堆栈的特征图连接起来作为编码器的输出。

命题 1.假设kj N (μ, Σ) 并且我们
设 qki表示集合 {(qik)/ √ d | j = 1,...,LK},那么
∀Mm = maxi M(qi, K) 存在 κ > 0 使得：
区间∀q1, q2 ∈ {q|M(q, K) ∈ [Mm, Mm − κ)}，
如果 M(q1, K) > M(q2, K) 且 Var(qk1) > Var(qk2)，
我们很有可能 M(q1, K) > M(q2, K)。到
简化,概率的估计在
证明。

编码器:允许处理更长的顺序
内存使用限制下的输入

编码器旨在提取长序列输入的稳健远程依赖。在输入表示之后,第 t 个
序列输入 X 已经被整形为

矩阵Xt ∈ RLx×dmodel 。我们给出了 en 的草图
为清楚起见,图（3)中的编码器。

Self-attention Distilling作为自然的结果
在 ProbSparse Self-attention 机制中,编码器的特征图具有值 V 的
冗余组合。我们使用
蒸馏操作以赋予上级特权
支配特征并在下一层制作集中的自注意力特征图。它修剪输入的时间维度
急剧地,看到 n 头权重矩阵（重叠红色
图（3)中的注意块的正方形）。受扩张卷积的启发（Yu、Koltun 和
Funkhouser 2017;Gupta
和 Rush 2017）,我们的“蒸馏”程序从
第 j 层变为第 (j + 1) 层为

$$X^{t}_{j+1} = \text{MaxPool}\,\text{ELU}(\text{Conv1d}([X^{t}_{j}]_{AB})), \quad (5)$$

其中 [·]AB包含 Multi-head ProbSparse 自注意力和注意力块中的基
本操作，
Conv1d(·) 使用 ELU(·) 激活函数 (Clevert, Unterthiner, and
Hochreiter 2016) 在时间维度上执行一维卷积滤波器（内核宽度
=3）。

我们添加一个步幅为 2 和下采样的最大池化层
Xt在堆叠一层后进入它的半片,这减少了
整个内存使用量为 O((2 −)Llog L),其中 是一个很小的数字。为了增
强蒸馏操作的鲁棒性,我们构建了主堆栈的减半副本

并通过一次丢弃一层来逐渐减少自注意力蒸馏层的数量,就像金字塔一
样
在图（3)中,它们的输出维度是对齐的。因此，
我们连接所有堆栈的输出并获得编码器的最终隐藏表示。

解码器:生成长序列输出
通过一个前向程序

我们使用标准解码器结构 (Vaswani et al. 2017)
图（2），它由两个相同的多头注意力层组成。然而,生成推理是

用于缓解长期预测中的速度骤降。
我们为解码器提供以下向量：

$$X^{t}_{\text{feed\_de}} = \text{Concat}(X^{t}_{\text{token}}, X^{t}_{0}) \in R^{(\text{Ltoken}+\text{Ly})\times \text{dmodel}}, \quad (6)$$

其中Xt 令牌 ∈ RLtoken×dmodel为起始令牌， Xt0 ∈
RLy×dmodel是目标序列的占位符（设置
标量为 0)。 Masked multi-head attention应用于
通过将掩码点积设置为 -∞ 来进行 ProbSparse 自注意计算。它阻止每
个职位参加
到即将到来的位置,这避免了自回归。一个完全
连接层获取最终输出,其超大dy
取决于我们执行的是单变量预测还是多变量预测。

Generative Inference Start token 是 NLP 的“动态解码”
（Devlin et al. 2018)中的一种有效技术，
我们将其扩展为一种生成方式。我们不是选择一个特定的标志作为令
牌,而是在输入序列中采样一个Ltoken长序列,它是之前的一个更早的
切片

输出序列。以预测 168 个点为例（图 2(b)）中的 7 天温度预测，我们将

将目标序列前 5 天的已知时间作为“开始标记”，并为生成式推理解码器提供
Xfeed_de = {X5d, X0}。 X0包含目标序列的
时间戳，即目标周的上下文。请注意，我们的
建议的解码器通过一个前向预测所有输出
程序并且免于耗时的“动态
解码”事务在普通的编码器-解码器架构中。详细的性能比较在

计算效率部分。

损失函数我们选择 MSE 损失函数对目标序列进行预测，损失被传播

从整个模型的解码器输出返回。

实验

数据集

我们凭经验对四个数据集进行了实验，包括 2 个收集的 LSTF 真实世界数据集和 2 个公共基准数据集。

ETT （电力变压器温度）[2]： ETT 为
电力长期部署的重要指标。我们从两个独立的县收集了 2 年的数据

在中国。为了探索 LSTF 问题的粒度，
我们为 1 小时级别创建单独的数据集，如 {ETTh1, ETTh2}和15 分钟级别的 ETTm1。每个数据点由目标值“油温”和6个功率负载组成

特征。 train/val/test 是 12/4/4 个月。

ECL （Electricity Consuming Load）[3]：收集321个客户的用电量（Kwh）。由于失踪
数据（Li et al. 2019），我们将数据集转换为 2 年的每小时消耗量，并将“MT 320”设置为目标值。
train/val/test 是 15/3/4 个月。

天气[4]：该数据集包含当地气候数据
从 2010 年到 2013 年的 4 年，近 1,600 个美国地点，
每 1 小时收集一次数据点。每个数据点
由目标值“湿球”和 11 个气候特征组成。训练/验证/测试是 28/10/10 个月。

实验细节

我们简要总结了基础知识，有关网络组件和设置的更多信息在附录 E 中给出。

基线：给出了网络组件的详细信息
在附录 E.1 中。我们选择了 5 种时间序列预测方法作为比较，包括 ARIMA (Ariyo,
Adewumi 和 Ayo 2014），先知（Taylor 和 Letham
2018）、LSTMa（Bahdanau、Cho 和 Bengio 2015)和
LSTnet (Lai et al. 2018) 和 DeepAR (Flunkert, Salinas,
和 Gasthaus 2017）。为了更好地探索 ProbSparse 自注意力在我们提出的 Informer 中的性能，我们采用了规范的自注意力变体（Informer†），

高效变体改革者（Kitaev、Kaiser 和 Levskaya
2019)和最相关的工作LogSparse self-attention
(Li et al. 2019) 在实验中。

超参数调优：我们进行网格搜索
超参数和详细范围在附录 E.3 中给出。 Informer 包含一个 3 层堆栈和一个 2 层
编码器中的堆栈（1/4 输入），2 层解码器。我们提出的
方法使用 Adam 优化器进行优化，其学习率从1e－4 开始，每下降 10 倍

2 epochs，总 epochs 为 10。我们按照推荐设置比较方法，批量大小为 32 。
每个数据集的输入都是零均值归一化的。在下面
LSTF 设置，我们延长预测窗口大小Ly
渐进式，即 {ETHh, 中的 {1d, 2d, 7d, 14d, 30d, 40d}
ECL, Weather}, {6h, 12h, 24h, 72h, 168h} 在 ETTm 中。指标：
我们使用了两个评估指标，包括 MSE = $\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$和 MAE = $\frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$在每一个上
预测窗口（多元预测的平均），
并以 stride = 1 滚动整个集合。平台：全部
模型在单个 Nvidia V100 32GB 上进行训练/测试
显卡。源代码位于 https://github.com/
周浩毅/Informer2020。

结果与分析

表 1 和表 2 总结了单变量/多变量

所有方法在 4 个数据集上的评估结果。我们逐渐延长预测范围作为更高的要求
的预测能力。为了公平比较，我们有
精确控制问题设置以使 LSTF 为
每种方法都可以在一个 GPU 上处理。最佳结果以粗体突出显示。

单变量时间序列预测在此设置下，
每种方法都在单个变量中获得预测
时间。从表 1 中，我们观察到：(1)提出的
model Informer 大大提高了推理性能
（最后一列中的获胜次数)在所有数据集中，以及
他们的预测误差在
不断扩大的预测范围。这证明了成功
Informer 在提高 LSTF 中的预测能力
问题。 (2) Informer 击败了它的规范降级
Informer †主要在获胜计数中，即 28>14，它支持查询稀疏假设，以提供可比较的注意力特征图。我们提出的方法也执行了最相关的工作
LogTrans 和 Reformer。我们

请注意，Reformer 保持动态解码并且在 LSTF 中表现不佳，而其他方法则受益于
生成风格解码器作为非自回归预测器。 (3) Informer 模型显示出明显优于循环神经网络 LSTMa 的结果。我们的方法

MSE 下降了 41.5%（在 168）、60.7%（在 336)和
60.7%（在 720 处）。这揭示了一个较短的网络路径
自注意力机制比基于 RNN 的模型获得更好的预测能力。 (4)我们提出的方法
取得比 DeepAR、ARIMA 和 Prophet 更好的结果
在 MSE 上降低 20.9%（在 168）、61.2%（在 336)和
平均为 51.3%（在 720 处）。在 ECL 数据集上，DeepAR 在较短的视野（≤ 336)上表现更好，我们的方法
在更长的视野中超越。我们将此归因于特定的
例如，其中预测能力的有效性是
体现为可扩展性问题。

[2]我们收集了 ETT 数据集并将其发布在 https://github.com/zhouhaoyi/ETDataset。

[3]ECL 数据集在 https://archive.ics.uci.edu/ml/ 获得 datasets/ElectricityLoadDiagrams20112014 天

[4]气数据集在 https://www.ncdc.noaa.gov/ 获得
订单/qclcd/

表1:四个数据集的单变量长序列时间序列预测结果（5个案例）

方法度量	ETH1								ETH2								ETTm1								天气								ECL								数数
	24 48				168 336 720				24 48				168 336 720				24 48				96 288 672				24 48				168 336 720				48				168 336 720				
告密者	MSE	0.062	0.108	0.146	0.208	0.193	0.079	0.103	0.143	0.171	0.184	0.051	0.092	0.119	0.181	0.204	0.107	0.164	0.226	0.241	0.259	0.335	0.408	0.451	0.466	0.470	MAE	0.178	0.245	0.294	0.363	0.365	0.206	0.240	0.296	0.327	0.339	28
		0.153	0.217	0.249	0.320	0.345	0.223	0.282	0.338	0.352	0.367	0.423	0.466	0.488	0.499	0.520																								
告密者†	MSE	0.046	0.129	0.183	0.189	0.201	0.083	0.111	0.154	0.166	0.181	0.054	0.087	0.115	0.182	0.207	0.107	0.167	0.237	0.252	0.263	0.304	0.416	0.479	0.482	0.538	MAE	0.152	0.274	0.337	0.346	0.357	0.213	0.249	0.306	0.323	0.338	14
		0.160	0.210	0.248	0.323	0.353	0.220	0.284	0.352	0.366	0.374	0.404	0.478	0.508	0.515	0.560																								
日志传输	MSE	0.059	0.111	0.155	0.196	0.217	0.080	0.107	0.176	0.175	0.185	0.061	0.156	0.229	0.362	0.450	0.120	0.182	0.267	0.299	0.274	0.360	0.410	0.482	0.522	0.546	MAE	0.191	0.263	0.309	0.370	0.379	0.221	0.262	0.344	0.345	0.349	0
		0.192	0.322	0.397	0.512	0.582	0.247	0.312	0.387	0.416	0.387	0.455	0.481	0.521	0.551	0.563																								
改革者	MSE	0.172	0.228	1.460	1.728	1.948	0.235	0.434	0.961	1.532	1.862	0.055	0.229	0.854	0.962	1.605	0.197	0.268	0.590	1.692	1.887	0.917	1.635	3.448	4.745	6.841	MAE	0.319	0.395	1.089	0.978	1.226	0.369	0.505	0.797	1.060	1.543	0
		0.170	0.340	0.675	1.107	1.312	0.329	0.381	0.552	0.945	1.352	0.840	1.515	2.088	3.913	4.913																								
LSTMa	MSE	0.094	0.175	0.210	0.556	0.635	0.135	0.172	0.359	0.516	0.562	0.099	0.289	0.255	0.480	0.988	0.107	0.166	0.305	0.404	0.784	0.475	0.703	1.186	1.473	1.493	MAE	0.232	0.322	0.352	0.644	0.704	0.275	0.318	0.470	0.548	0.613	1
		0.201	0.371	0.370	0.528	0.805	0.222	0.298	0.404	0.476	0.709	0.509	0.617	0.854	0.910	0.9260																								
深度增强现实	MSE	0.089	0.126	0.213	0.403	0.614	0.080	0.125	0.179	0.568	0.367	0.075	0.197	0.336	0.908	2.371	0.108	0.177	0.259	0.535	0.407	0.188	0.295	0.388	0.471	0.583	MAE	0.242	0.291	0.382	0.496	0.643	0.229	0.283	0.346	0.555	0.488	6
		0.205	0.332	0.450	0.739	1.256	0.242	0.313	0.397	0.580	0.506	0.317	0.398	0.471	0.507	0.583																								
有马	MSE	0.086	0.133	0.364	0.428	0.613	3.538	3.168	2.768	2.717	2.822	0.074	0.157	0.242	0.424	0.565	0.199	0.247	0.471	0.678	0.996	0.861	1.014	1.102	1.213	1.322	MAE	0.190	0.242	0.456	0.537	0.684	0.407	0.440	0.555	0.680	0.952	1
		0.168	0.274	0.357	0.500	0.605	0.321	0.375	0.541	0.666	0.853	0.726	0.797	0.834	0.883	0.908																								
预言家	MSE	0.093	0.150	1.194	1.509	2.685	0.179	0.284	2.113	2.052	3.287	0.102	0.117	0.146	0.414	2.671	0.280	0.421	2.409	1.931	3.759	0.506	2.711	2.220	4.201	6.827	MAE	0.241	0.300	0.721	1.766	3.155	0.345	0.428	1.018	2.487	45.92	0
		0.256	0.273	0.304	0.482	1.112	0.403	0.492	1.092	2.406	1.030	0.557	1.239	3.029	1.363	4.184																								

表2:四个数据集上的多元长序列时间序列预测结果（5个案例）

方法度量	ETH1								ETH2								ETTm1								天气								ECL								数数
	24 48				168 336 720				24 48				168 336 720				24 48				96 288 672				24 48				168 336 720				48				168 336 720				
告密者	MSE	0.509	0.551	0.878	0.884	0.941	0.446	0.934	1.512	1.665	2.340	0.325	0.472	0.642	1.219	1.651	0.353	0.464	0.592	0.623	0.685	0.269	0.300	0.311	0.308	0.328	MAE	0.523	0.563	0.722	0.753	0.768	0.523	0.733	0.996	1.035	1.209	32
		0.440	0.537	0.626	0.871	1.002	0.381	0.455	0.531	0.546	0.575	0.351	0.376	0.385	0.385	0.406																								
告密者†	MSE	0.550	0.602	0.893	0.836	0.981	0.683	0.977	1.873	1.374	2.493	0.324	0.446	0.651	1.342	1.661	0.355	0.471	0.613	0.626	0.680	0.269	0.281	0.309	0.314	0.356	MAE	0.551	0.581	0.733	0.729	0.779	0.637	0.793	1.094	0.935	1.253	12
		0.440	0.508	0.616	0.927	1.001	0.383	0.456	0.544	0.548	0.569	0.351	0.366	0.383	0.388	0.394																								
日志传输	MSE	0.656	0.670	0.888	0.942	1.109	0.726	1.728	3.944	3.711	2.817	0.341	0.495	0.674	1.728	1.865	0.365	0.496	0.649	0.666	0.741	0.267	0.290	0.305	0.311	0.333	MAE	0.600	0.611	0.766	0.766	0.843	0.638	0.944	1.573	1.587	1.356	2
		0.495	0.527	0.674	1.656	1.721	0.405	0.485	0.573	0.584	0.611	0.366	0.382	0.395	0.397	0.413																								
改革者	MSE	0.887	1.159	1.686	1.919	2.177	1.381	1.715	4.484	3.798	5.111	0.598	0.952	1.267	1.632	1.943	0.583	0.633	1.228	1.770	2.548	1.312	1.453	1.507	1.883	1.973	MAE	0.630	0.750	0.996	1.090	1.218	1.475	1.585	1.650	1.508	1.793	0
		0.489	0.645	0.795	0.886	1.006	0.497	0.556	0.763	0.997	1.407	0.911	0.975	0.978	1.002	1.185																								
LSTMa	MSE	0.536	0.616	1.058	1.152	1.682	1.049	1.331	3.987	3.276	3.711	0.511	1.280	1.195	1.598	2.530	0.476	0.763	0.948	1.497	1.314	0.388	0.492	0.778	1.528	1.343	MAE	0.528	0.577	0.725	0.794	1.018	0.689	0.805	1.560	1.375	1.520	0
		0.517	0.819	0.785	0.952	1.259	0.464	0.589	0.713	0.889	0.875	0.444	0.498	0.629	0.945	0.886																								
LSTnet	MSE	1.175	1.344	1.865	2.477	1.925	2.632	3.487	1.442	1.372	2.403	1.856	1.909	2.654	1.009	1.681	0.575	0.622	0.676	0.714	0.773	0.279	0.318	0.357	0.442	0.473	MAE	0.793	0.864	1.092	1.193	1.084	1.337	1.577	2.389	2.429	3.403	4
		1.058	1.085	1.378	1.902	2.701	0.507	0.553	0.585	0.607	0.643	0.337	0.368	0.391	0.433	0.443																								

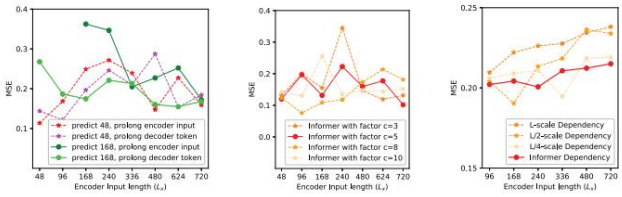
多变量时间序列预测在此设置中,一些单变量方法是不合适的，
LSTnet
是最先进的基线。相反,我们提出的 In 前者很容易通过调整最终的 FCN
层从单变量预测变为多变量预测。从表 2，

我们观察到： (1)提出的模型 Informer 大大
优于其他方法,单变量设置中的结果 1 和 2 仍然适用于多变量时间序
列。 (2)
Informer 模型比基于 RNN 的模型显示出更好的结果
基于 LSTMa 和 CNN 的 LSTnet,MSE 降低
平均为 9.5%（在 168 处）、2.1%（在 336 处）、13.8%（在 720 处）。
与单变量结果相比,压倒性的性能降低了,这种现象可能是由

特征维度预测能力的各向异性。它
超出了本文的范围,我们将在未来的工作中进行探索。
考虑粒度的 LSTF我们执行一个
尝试探索性能的附加比较
具有各种粒度。的序列 {96, 288, 672}
ETTm1 （分钟级别)与 {24, 48, 168} 对齐
ETHh1 （小时级别）。即使序列处于不同的粒度级别,Informer 也优
于其他基线。

参数灵敏度

我们在单变量设置下对 ETTh1 上提出的 In 前模型进行了敏感性分析。
输入
长度:在图（4（a）)中,当预测短序列时（如
48）,最初增加编码器/解码器的输入长度会降低性能,但进一步增加会
导致 MSE
下跌,因为它带来了重复的短期模式。如何-



(a) 输入长度。 (b) 抽样。 (c) 堆叠。

图 4:Informer 参数的敏感度。

曾经,MSE 会随着预测的输入越长而越低
长序列（如 168）。因为更长的编码器输入
可能包含更多的依赖,以及更长的解码器令牌
有丰富的本地信息。抽样因子:抽样
factor控制ProbSparse的信息带宽
等式 (3)中的自我注意。我们从小因素 (=3)开始
到大的,一般性能会增加一点
并最终稳定在图（4（b）)中。它验证了我们的查询稀疏性假设,即在

自注意力机制。我们设置样本因子 c = 5
（红线)在实践中。堆叠层数：
Layers的replica对self-attention是互补的
蒸馏,我们研究图（4（c）)中每个堆栈{L,L / 2,L / 4}的行为。越长的堆
栈越敏感
投入,部分原因是接收到更多的长期信息。
我们方法的选择（红线），即结合 L 和
L/4,是最稳健的策略。

表 3:ProbSparse 机制的消融

预测长度		336			720		
编码器的输入		336	720	1440	720	1440	2880
告密者	MSE 0.243 MAE		0.225	0.212	0.258	0.238	0.224
	0.487 MSE 0.214		0.404	0.381	0.503	0.399	0.387
告密者†	MAE 0.369 MSE		0.205	-	0.235	-	-
	0.256 MAE 0.496		0.364	-	0.401	-	-
日志传输	MSE 1.848 MAE		0.233	-	0.264	-	-
	1.054		0.412	-	0.523	-	-
改革者			1.832	1.817	2.094	2.055	2.032
			1.027	1.010	1.363	1.306	1.334

1 Informer †使用规范的自注意机制。 2
“-”表示内存不足失败。

表 4:自注意力蒸馏的消融

预测长度	336	480
编码器输入	336 480 720 960 1200	336 480 720 960 1200
告密者†	MSE 0.201 0.175 0.215 0.185 0.172 0.136 0.213 0.178 0.146 0.121	
	MAE 0.360 0.335 0.366 0.355 0.321 0.282 0.382 0.345 0.296 0.272	
告密者‡	MSE 0.187 0.182 0.177 0.208 0.182 0.168 MAE 0.330 0.341 0.329 0.384 0.337 - -	
	0.304 - -	- -

1 Informer ‡消除了 Informer † 中的自注意力提炼。
2 “-”表示内存不足失败。

表 5:生成式解码器的消融

预测长度		336				480			
预测偏移		+0	+12	+24	+48	+0	+48	+96	+168
告密者‡	MSE 0.101 0.102 0.103 0.103 0.155 0.158 0.160 0.165								
	MAE 0.215 0.218 0.223 0.227 0.317 0.397 0.399 0.406								
告密者§	MSE 0.152 MAE	-	-	-		0.462	-	-	-
	0.294		-	-	-	0.595	-	-	-

1 Informer §将我们的解码器替换为 Informer ‡ 中的动态解码器。
2 “-”表示不可接受的度量结果失败。

消融研究:Informer 如何工作?

我们还对 ETHh1 进行了额外的实验
消融考虑。

ProbSparse 自注意力机制的性能在整体结果表 1 和表 2 中,我们限制了

问题设置以使内存使用对
规范的自注意力。在本研究中,我们将我们的方法与 LogTrans 和
Reformer 进行比较,并深入探索
他们的极端表现。高效隔离内存
问题,我们首先将设置减少为{batch size=8,heads=8,
dim=64},并在单变量情况下保持其他设置。
在表 3 中,ProbSparse self-attention 表现出比同类产品更好的性
能。 LogTrans 进入 OOM
公开实施的极端案例是
全注意力,仍然有 O(L2)内存使用。我们的
提出的 ProbSparse self-attention 避免了这种情况,因为 Eq.(4) 中
的查询稀疏假设带来了简单性,
参考附录 E.2 中的伪代码,并达到
较小的内存使用量。

self-attention distilling 的表现在此
研究中,我们使用 Informer †作为基准来消除
ProbSparse 自注意力的附加效果。另一个
实验设置与单变量时间序列的设置一致。从表 4 中可以看出,Informer †完成
了所有实验,并在利用长序列输入后取得了更好的性能。比较方法 Informer ‡
去除了蒸馏操作和

以更长的输入 (> 720) 达到 OOM。关于
LSTF 问题中长序列输入的好处,我们

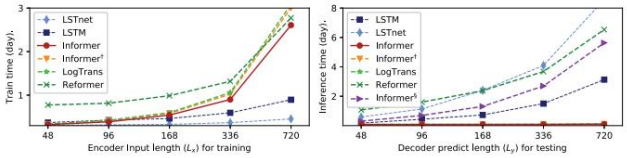


图 5:训练/测试阶段的总运行时间。

表 6:每层的 L 相关计算静态

方法	训练		测试
	时间复杂度	内存使用情况	脚步
告密者	O(L log L)	O(L log L)	1
变压器	O(L2)	O(L2)	大号
日志传输	O(L log L)	O(L2)	1
改革者	O(L log L)	O(L log L)	大号
长短期记忆体	O(L)	O(L)	大号

1 LSTnet 很难有一个封闭的形式。

得出结论,self-attention distilling 值得采用,
特别是当需要更长的预测时。

生成式解码器的性能
研究中,我们证明了我们的解码器在获得“生成”结果方面的潜
在价值。与现有方法不同的是,
标签和输出在训练中被迫对齐,并且
推断,我们提出的解码器的预测完全依赖于
时间戳,可以用偏移量进行预测。从表 5 可以看出,一般预测性能

Informer ‡的抵制随着偏移量的增加而增加,而
对应的动态解码失败。它证明了解码器捕获个体远程依赖的能力

任意输出之间,避免错误累积
推论。

计算效率

使用多变量设置和每种方法的当前
最好的实现,我们在图（5)中进行了严格的运行时比较。在训练阶段,Informer

（红线)达到最好的训练效率
基于变压器的方法。在测试阶段,我们的
方法比其他具有生成风格的方法快得多
解码。理论时间复杂度的比较
和内存使用情况总结在表 6 中,Informer 的性能与运行时实验一致。

请注意,LogTrans 专注于自注意力机制,我们在 LogTrans 中应用我们提出的
解码器
公平比较（表 6 中的）。

结论

在本文中,我们研究了长序列时间序列预测问题,并提出了 Informer 来预测长
序列。具体来说,我们设计了 ProbSparse 自注意力机制和蒸馏操作来处理

vanilla Transformer 中二次时间复杂度和二次内存使用的挑战。此外,精心设
计的生成解码器缓解了传统编码器-解码器架构的局限性。上的实验

真实世界的数据证明了 Informer 的有效性
用于提高 LSTF 问题的预测能力。

致谢

这项工作得到了自然科学基金的支持
中国基金会 (U20B2053, 61872022 和 61421003)
软件开发环境国家重点实验室（SKLSDE-2020ZX-12）。感谢北京大数据与
脑计算先进创新中心提供的计算基础设施。相应的

作者是李建新。

道德声明

提议的 Informer 可以处理长输入并生成
有效的长序列推理,可应用于
解决具有挑战性的长序列时间序列预测 (LSTF) 问题。一些重要的实际应用是
传感器网络监控（Papadimitriou 和 Yu

2006)、能源和智能电网管理、疾病传播分析 (Matsubara et al. 2014)、经济和
金融预测 (Zhu and Shasha 2002)、农业生态系统的演变、气候变化预测和空
气变化

污染。作为一个具体的例子,在线卖家可以预测
每月产品供应,有助于优化长期库存管理。我们的贡献不仅限于 LSTF 问题。除
了获取长序列外,我们的方法还可以为其他人带来实质性的好处

领域,例如文本的长序列生成,音乐,
图片和视频。
在伦理考虑下,任何从历史数据中学习的时间序列预测应用程序都会运行

产生有偏见的预测的风险。它可能给财产/资产的真正所有者造成无法弥补的
损失。领域
专家应该指导我们方法的使用,他们
从长序列预测中具有潜在的好处。
以应用我们的电力变压器温度预测方法为例,经理将检查

结果并决定未来的电力部署。如果长
有足够的预测,将有助于管理者在早期防止不可逆转的失败。除了识别偏差数
据之外,一种有前途的方法是

采用迁移学习。我们捐赠了收集到的
数据（ETT 数据集）,用于进一步研究相关主题
（如供水管理、5G网络部署）。
另一个缺点是我们的方法需要高性能
GPU,这限制了它在欠发达地区的应用。

参考

阿里约,AA;阿德乌米,AO;和 Ayo, CK 2014。股票
使用 ARIMA 模型进行价格预测。在第 16 届计算机建模与仿真国际会议上，
106-112。 IEEE。

巴赫达瑙,D。赵,K。和 Bengio, Y. 2015。联合学习对齐和翻译的神经机器翻译。
在 ICLR 2015 中。

; 贝尔塔吉,我。彼得斯,我;和 Cohan, A. 2020。Longformer：
长文档转换器。 CoRR abs/2004.05150。

通用电气公司;詹金斯,总经理； Reinsel,GC;和 Ljung,总经理
2015.时间序列分析:预测与控制。约翰
威利父子。

布朗,结核病;曼恩,B。 ;莱德,N。苏比亚,M。卡普兰,J。
达里瓦尔,P。尼拉坎坦,A.;希亚姆,P。萨斯特里,G。阿斯克尔，
一种。;阿加瓦尔,S。赫伯特-沃斯,A.;克鲁格,G。黑根汉，
T.;孩子,R。拉梅什,A。齐格勒,DM;吴,J。冬天，
C。;黑森,C。陈,M。西格勒,E。利特温,M。格雷,S。
国际象棋,B。克拉克,J。伯纳,C。麦坎德利什,S。拉德福，
一种。;苏茨克韦尔,我。和 Amodei, D. 2020。语言模型
是少数人的学习者。 CoRR abs/2005.14165。

孩子,R。格雷,S。拉德福德,A。和 Sutskever, I. 2019。
使用稀疏变换器生成长序列。
arXiv:1904.10509 。

赵,K。范梅林博尔,B.;巴赫达瑙,D。和本吉奥，
Y. 2014。关于神经机器翻译的属性:编码器-解码器方法。在诉讼中

SSST@EMNLP 2014,103-111。

聪明的,D。 ;Unterthiner,T。和 Hochreiter, S. 2016。快速
通过指数线性单元 (ELU) 进行准确的深度网络学习。在 ICLR 2016 中。

戴,Z。杨,Z。杨,Y。卡内尔,J。乐,QV;和
Salakhutdinov, R. 2019。Transformer-xl:专注的语言
超出固定长度上下文的模型。 arXiv:1901.02860。

德夫林,J。张,M.-W。韭葱。;和 Toutanova, K. 2018。
Bert:用于语言理解的深度双向转换器的预训练。 arXiv:1810.04805。

弗朗克特,五。萨利纳斯,D。和 Gasthaus, J. 2017。DeepAR：
使用自回归循环网络进行概率预测。 arXiv:1704.04110。

古普塔,A。和 Rush,AM 2017。用于模拟长距离基因组依赖关系的扩张卷积。

arXiv:1710.01278。

霍克赖特,S。和 Schmidhuber, J. 1997。长期短期
记忆。神经计算 9（8）:1735-1780。

基塔耶夫,N。凯撒,L。和 Levskaya, A. 2019。改革者：
高效变压器。在 ICLR 中。

赖,G。常,W.-C.;杨,Y。和 Liu, H. 2018。使用深度神经网络建模长期和短期时间
模式
网络。在 ACM SIGIR 2018 年,95–104。 ACM。

李,S。金,X。 ; 轩,Y。周,X。陈,W。王,Y.-X。
and Yan, X. 2019. 增强地方性,打破传统
时间序列预测中 Transformer 的内存瓶颈。 arXiv:1907.00235。

李,Y。于,R。沙哈比,C。和 Liu, Y. 2018。扩散卷积递归神经网络:数据驱动的流
量
预测。在 ICLR 2018 中。

刘,Y。龚,C。杨,L。和 Chen, Y. 2019。DSTP-RNN：
基于双阶段两阶段注意力的循环神经网络适用于长期和多变量时间序列预测。

CoRR abs/1904.07464。

松原,Y。樱井,Y。 van Panhuis, WG;和 Falout sos, C. 2014 年。漏斗:自动挖
掘空间 coe 流行病。在 ACM SIGKDD 2014,105–114。

帕帕迪米特里乌,S.;和 Yu, P. 2006。时间序列流中的最佳多尺度模式。在 ACM SIGMOD 2006, 647–658. ACM。

秦,Y。宋,D。陈,H。程,W。江,G。和 Cot trell, GW 2017。用于时间序列预测的双阶段基于注意力的递归神经网络。在 IJCAI 2017,2627–2633。

邱,J。马,H。 ;利维,O。易,SW-t。王,S。和 Tang, J. 2019。用于长文档理解的 Blockwise Self-Attention。 arXiv:1911.02972。

雷,JW;波塔彭科,A.;贾古玛,SM;和 Lillicrap,TP 2019。用于远程序列建模的压缩变压器。 arXiv:1911.05507 。

Ray, W. 1990。时间序列:理论和方法。皇家统计学会杂志:A 系列（社会统计学)153（3）:400-400。

西格,M。兰加普拉姆,S.;王,Y。萨利纳斯,D。盖斯豪斯,J。亚努肖夫斯基,T。和 Flunkert, V. 2017。
用于大规模间歇性需求预测的线性状态空间模型中的近似贝叶斯推理。 arXiv:1709.07638 。

西格,兆瓦;萨利纳斯,D。和 Flunkert, V. 2016 年。大库存的贝叶斯间歇性需求预测。在 NIPS 中,4646–4654。

苏茨克韦尔,我。 ;乙烯基,O。和 Le,QV 2014。使用神经网络进行序列到序列学习。在 NIPS,3104-3112。

泰勒,SJ;和 Letham, B. 2018。大规模预测。美国统计学家 72（1）:37-45。

蔡,Y.-HH;白,S。山田,M。莫伦西,L.-P.;和 Salakhutdinov, R. 2019。变压器解剖:通过内核镜头对变压器注意力的统一理解。在 ACL 2019 中,4335–4344。

瓦斯瓦尼,A。沙泽尔,N。帕尔马,N。乌兹科雷特,J。琼斯,L。戈麦斯,AN;凯撒,。和 Polosukhin, I. 2017。注意力就是你所需要的。在 NIPS,5998-6008。

王,S。李,B。哈布萨,M。方,H。和马,H。
2020. Linformer:具有线性复杂性的自我注意。 arXiv:2006.04768。

温,R。托尔科拉,K。纳拉亚纳斯瓦米,B。和 Madeka, D. 2017 年。多水平分位数循环预报器。 arXiv:1711.11053。

于,F。科尔通,V。和 Funkhouser, T. 2017。扩张残差网络。在 CVPR 中,472–480。

于,R。郑,S。阿南德库马尔,A。和 Yue, Y. 2017。使用张量训练 rnns 进行长期预测。 arXiv:1711.00073
.

朱,Y。和 Shasha, DE 2002。StatStream:实时统计监控数以千计的数据流。在 VLDB 2002 中,358–369。