时间序列预测与深度学习:文献综述与应用实例

李 文[1] 邓 升[1] 段 妍[1] 杜守国[2*]

[1](上海对外经贸大学统计与信息学院 上海 201620)

[2](上海市人力资源和社会保障局信息中心 上海 200051)

摘 要 随着深度学习与时间序列传统模型的融合发展,通过利用大量可用数据在整个时间序列集合中估计全局模型的参数,使得传统的局部建模方法得到了实质性的改进。介绍近年来提出的与深度学习相结合的时间序列预测方法及三种时间序列预测模型:深度状态空间模型(DSSM),深度自回归模型(DeepAR),Transformer 模型。采用 GluonTS 时间序列预测框架对上海市出口额数据进行预测并给出效果评估。实验结果表明,基于深度学习的时间序列预测效果明显优于传统的 ARIMA 模型的预测。

关键词 时间序列预测 深度状态空间模型 深度自回归模型 Transformer 模型

中图分类号 O211.61 TP183 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2020.10.011

TIME SERIES FORECASTING AND DEEP LEARNING: LITERATURE REVIEW AND EMPIRICAL EXAMPLE

Li Wen[1] Deng Sheng[1] Duan Yan[1] Du Shouguo[2*]

[1](*School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai 201620, China*)

[2](*Shanghai Municipal Human Resources and Social Security Bureau Information Technology Center, Shanghai 200051, China*)

Abstract With the development of the integration of deep learning and traditional time series models, the traditional local modeling methods have been substantially improved by using a large amount of available data to estimate the parameters of the global model in the entire time series set. This paper introduces the time series forecasting methods combined with deep learning proposed in recent years and sort out related literature, as well as three kind of time series forecasting models: the deep state space model(DSSM), the deep autoregressive model(DeepAR), and the Transformer model. We used the GluonTS time series forecasting framework to predict the export data of Shanghai. The experimental results show that the time series forecasting effect based on deep learning is significantly better than that of the traditional ARIMA model.

Keywords Time series forecasting Deep state space model Deep autoregressive model Transformer model

0 引 言

时间序列预测旨在基于给定的时间序列观测值估计未来时间点上的取值或概率分布,这是风险管理和决策的关键任务,它在许多领域中发挥了重要作用,包括统计学、机器学习、数据挖掘、计量经济学、运筹学等。例如,预测产品的供需可用于优化库存管理、车辆调度和拓扑规划,这对于供应链优化的大多数方面都至关重要[1-2]。

传统的时间序列预测模型包括 ARIMA 模型、指数平滑方法和状态空间模型(SSM)等,其中 ARIMA 模型、指数平滑方法都可以表示成状态空间模型的形式,SSM 提供了时间序列建模的通用框架,由状态方程和

收稿日期:2020-02-27。海关总署决策咨询研究课题(HG-YB009);国家社会科学基金项目(17BTJ025);上海市领军人才。
李文,副教授,主研领域:时间序列分析,经济统计学。**邓升**,硕士生。**段妍**,硕士生。**杜守国**,教授级高工。

观察值方程组成[2]。

在现代预测应用中,传统的 SSM 模型无法从相似的时间序列数据集合中推断出共享模式,这就产生了繁重的计算任务和大量人力劳动需求。因此,深度神经网络(Deep Neural Networks,DNN)凭借其提取高阶特质的能力走进了人们的视野。通过深度神经网络,可以识别时间序列内部和时间序列之间的复杂模式,并且能够从原始时间序列的数据集合中进行识别,所需的人力要少得多。然而,由于这些模型所作的结构假设较少,它们通常需要更大的训练数据集来学习得到准确的模型[2]。

为了弥补这两种方法各自的不足,将传统统计模型与深度学习融合,产生了一些新的预测方法。Chung 等[3]和 Fraccaro 等[4]使用循环神经网络(Recurrent Neural Networks,RNN)在 SSMs 和变分自编码器(Variational Auto Encoding,VAE)之间建立联系。Krishnan 等[5]利用深度卡尔曼滤波器(Deep Kalman Filters,DKF)在 SSM 中引入外生变量。在预测方面,Rangapuram 等[2]使用 RNN 在每个时间步上生成线性高斯状态空间模型(LGSSM)的参数,提出了深度状态空间模型(Deep State Space Model,DSSM)。对于非线性 SSM,Eleftheriadis 等[6]提出非线性高斯过程状态空间模型(Gaussian Process State Space Model, GPSSM)。Salinas 等[7]研究了多元时间序列预测问题。Salinas 等[8]的深度自回归模型(DeepAR)是建立在对时间序列数据进行深度学习的基础上,为概率预测问题设计了一个类似的基于 LSTM(Long Short-Term Memory)的自回归 RNN 架构。而 Vaswani 等[9]提出的 Transformer 利用 Attention Mechanism 来处理数据。与基于 RNN 的方法不同,Transformer 允许模型访问历史的任何部分,而不考虑距离,这使得它更适合于捕捉具有长期依赖性的循环结构。

徐超等[10]提出一种集成自回归综合移动平均(ARIMA)模型与自适应过滤法的组合预测模型。该组合强调 ARIMA 模型对时间序列数据特征识别与参数估计的优势,同时引入自适应过滤法的“权数”调整思想,对 ARIMA 模型的参数进行调整,以减少预测误差,提高预测精度。沈旭东[11]对近年来基于深度学习的时间序列分析方法进行讨论,从应用、网络架构、思想等方面总结了最新的时间序列预测、分类、异常检测等任务的深度学习方法,为了解时间序列深度学习解决方案的技术和发展趋势提供了参考。吴双双[12]利用卷积神经网络、循环神经网络、双通道神经网络对数据进行了预测,并取得了不错的预测效果。权钲杰[13]利用长短期记忆网络和卷积神经网络对数据进行预测,并针对深度神经网络模型训练不稳定的问题,研究了将集成学习方法应用于对深度神经网络预测模型的改进,提出了基于噪声扰动集成方法的深度神经网络集成模型。刘峰等[14]提出了一种组合聚类分析和神经网络的预测方法。王慧健等[15]提出一种新的时间序列短期趋势预测方法,通过对时序数据进行离散化,用字符表示各个时间段数据的范围,并利用神经网络语言模型预测得到下一个字符。李洁等[16]基于真实的民航旅客历史出行记录,根据其时序数据的特征建立基于后向传播算法的循环神经网络(RNN)预测模型,对未来时段的日客流量进行预测。在此基础上考虑到时序数据在不同时间尺度呈现不同的变化规律,建立多时间尺度的预测模型对旅客出行的周期性和趋势性进行建模,提升预测精度。蒋倩仪[17]根据震荡盒理论提出一种新的适应于与机器学习相结合的交易边界模型,通过结合基于距离的多核极限学习机(DB-MK-ELM)与交易边界模型,构建基于时间序列预测的股票交易决策建议系统,使得在股票交易中能稳定获得较高的收益率并保持较低的投资风险。

本文旨在介绍近年来提出的与深度学习相结合的时间序列预测方法。本文介绍三种时间序列预测模型:深度状态空间模型(DSSM)、深度自回归模型(DeepAR)、Transformer 模型(Transformer),并运用上海市出口额数据的预测实例说明它们的应用效果。实验结果表明,基于深度学习的时间序列预测效果明显优于传统的 ARIMA 模型。

1　时间序列预测问题

一般的时间序列概率预测问题描述如下,设 N 为一组单变量时间序列 $\{\boldsymbol{z}_{1:T_i}^{(i)}\}_{i=1}^{N}$,其中 $z_t^{(i)}$ 表示时间 t 第 i 个时间序列的值,$\boldsymbol{z}_{1:T_i}^{(i)} = (z_1^{(i)}, z_2^{(i)}, \cdots, z_{T_i}^{(i)})$,进一步令 $\{\boldsymbol{x}_{1:T_i+\tau}^{(i)}\}_{i=1}^{N}$ 为一组相关的、随时间变化的协变量向量,其中 $x_t^{(i)} \in \mathbf{R}^D$。时间序列概率预测的目标是产生一组概率分布,即对于每个 $i = 1,2,\cdots,N$,基于过去的未来 $z_{T_i+1:T_i+\tau}^{(i)}$ 的概率分布:

$$P = (\boldsymbol{z}_{T_i+1:T_i+\tau}^{(i)} \mid \boldsymbol{z}_{1:T_i}^{(i)}, \boldsymbol{x}_{1:T_i+\tau}^{(i)}; \boldsymbol{\Phi}) \tag{1}$$

式中:$\boldsymbol{\Phi}$ 表示模型的一组可学习参数,这些参数在所有 N 个时间序列之间共享并共同学习。对于任何给定的 i,引用时间序列 $\boldsymbol{z}_{1:T_i}^{(i)}$ 作为目标时间序列,时间区间

$\{1,2,\cdots,T_i\}$ 作为训练区间，预测区间是 $\{T_i+1,T_i+2,\cdots,T_i+\tau\}$。时间点 T_i+1 为预测开始时间，$\tau \in N>0$ 为预测层位。假设协变量向量 $x_t^{(i)}$ 在预测范围内可知。

同时假设时间序列在相关协变量 $\boldsymbol{x}_{1:T_i}^{(i)}$ 和参数 $\boldsymbol{\Phi}$ 的条件下相互独立。与许多作出相同假设的相关方法相比，在这种方法中，模型参数 $\boldsymbol{\Phi}$ 在所有时间序列之间是共享的。因此，虽然这种假设无法对时间序列之间的相关性进行建模，但并不意味着所提出的模型无法在不同时间序列之间共享统计强度和学习模式，因为该模型正在从所有时间序列中联合学习参数 $\boldsymbol{\Phi}$。

2 时间序列模型

2.1 深度状态空间模型(DSSM)

传统的 SSM 模型利用时间序列的潜在状态 $\boldsymbol{l}_t \in \mathbf{R}^L$ 对数据结构进行建模，该状态可用于编制时间序列的组成部分(如水平、趋势和季节性构成)，并通常应用于单个时间序列的预测。一般的 SSM 包含了定义潜在状态随时间演变的随机转移概率 $p(\boldsymbol{l}_t \mid \boldsymbol{l}_{t-1})$ 的状态转移模型，以及给定潜在状态的观测条件概率 $p(z_t \mid \boldsymbol{l}_t)$ 的观测模型。

状态转移方程的形式为：

$$\boldsymbol{l}_t = \boldsymbol{F}_t\boldsymbol{l}_{t-1} + g_t\varepsilon_t \tag{2}$$

式中：$\varepsilon_t \sim N(0,1)$；在时间 t 潜在状态 $\boldsymbol{l}_{t-1}$ 代表关于水平、趋势以及季节性因素的信息，通过确定的转移矩阵 $\boldsymbol{F}_t$ 和随机创新 $g_t\varepsilon_t$ 进行递归计算；转移矩阵 $\boldsymbol{F}_t$ 和创新强度 g_t 确定了由潜在状态 $\boldsymbol{l}_t$ 编制的时间序列构成。

概率观测模型则描述了如何由潜在状态 $\boldsymbol{l}_t$ 生成观测值。这里考虑一个单变量高斯观测模型：$z_t = y_t + \sigma_t v_t, y_t = \boldsymbol{a}_t^{\mathrm{T}}\boldsymbol{l}_{t-1} + b_t, v_t \sim N(0,1), \boldsymbol{a}_t \in \mathbf{R}^L, \sigma_t \in \mathbf{R}>0, b_t \in \mathbf{R}$ 是模型的时变参数，并假定初始状态 $\boldsymbol{l}_0$ 遵循同方向的高斯分布 $\boldsymbol{l}_0 \sim N(\mu_0, \mathrm{diag}(\sigma_0^2))$。

状态空间模型完全由参数指定 $\boldsymbol{\Theta}_t = (\mu_0, \boldsymbol{\Sigma}_0, \boldsymbol{F}_t, g_t, \boldsymbol{a}_t, b_t, \sigma_t)$，$\forall t>0$，并假定为时不变的，即 $\boldsymbol{\Theta}_t = \boldsymbol{\Theta}$，$\forall t>0$。通用的估计方法是最大边际似然估计，即：

$$\boldsymbol{\Theta}_{1:T}^* = \underset{\boldsymbol{\Theta}_{1:T}}{\mathrm{argmax}} p_{ss}(z_{1:T} \mid \boldsymbol{\Theta}_{1:T}) \tag{3}$$

式中：$p_{ss}(z_{1:T} \mid \boldsymbol{\Theta}_{1:T}) := p(z_1 \mid \boldsymbol{\Theta}_1)\prod_{t=2}^{T} p(z_t \mid z_{1:t-1}, \boldsymbol{\Theta}_{1:t}) = \int p(\boldsymbol{l}_0)\left[\prod_{t=1}^{T} p(z_t \mid \boldsymbol{l}_t)p(\boldsymbol{l}_t \mid \boldsymbol{l}_{t-1})\right]\mathrm{d}\boldsymbol{l}_{0:T}$ 表示观察值 $z_{1:T}$ 在状态空间 $\boldsymbol{\Theta}_t$ 参数下的边际概率。

在 SSM 中，如果有多个时间序列，则每个时间序列 $\boldsymbol{z}_{1:T_i}^{(i)}$ 独立地学习一组单独的参数 $\boldsymbol{\Theta}^{(i)}$。这就导致不同的时间序列之间无法共享任何信息，所以当此方法应用于历史数据有限或噪声水平较高的时间序列时表现不佳。

深度状态空间模型(DSSM)并非独立地学习每个时间序列的状态空间参数 $\boldsymbol{\Theta}^{(i)}$，而是从与每个时间序列相关的协变量向量 $\boldsymbol{x}_{1:T_i}^{(i)}$ 和目标时间序列 $\boldsymbol{z}_{1:T_i}^{(i)}$ 中学习，全局共享映射到第 i 个时间序列的线性状态空间模型的(时变)参数 $\boldsymbol{\Theta}_t^{(i)}$。$\boldsymbol{\Theta}_t^{(i)}$ 表示为：

$$\boldsymbol{\Theta}_t^{(i)} = \Psi(x_{1:t}^{(i)}, \boldsymbol{\Phi}) \tag{4}$$

式中：$i=1,2,\cdots,N; t=1,2,\cdots,T_i+\tau; \boldsymbol{\Theta}_t^{(i)}$ 是整个协变量时间序列 $x_{1:t}^{(i)}$ 的函数，并含有一组共享参数 $\boldsymbol{\Phi}$；Ψ 是从与每个时间序列相关的协变量向量 $x_{1:T_i}^{(i)}$ 和共享参数 $\boldsymbol{\Phi}$ 到状态空间模型参数 $\boldsymbol{\Theta}_t^{(i)}$ 的映射。在给定的 $x_{1:T_i}^{(i)}$ 和参数 $\boldsymbol{\Phi}$ 下，$z_{1:T_i}^{(i)}$ 的分布如下：

$$p(\boldsymbol{z}_{1:T_i}^{(i)} \mid \boldsymbol{x}_{1:T_i}^{(i)}, \boldsymbol{\Phi}) = p_{ss}(\boldsymbol{z}_{1:T_i}^{(i)} \mid \boldsymbol{\Theta}_{1:T_i}^{(i)}) \quad i=1,2,\cdots,N \tag{5}$$

式中：p_{ss} 表示线性状态空间模型下给定其(时变)参数 $\boldsymbol{\Theta}_t^{(i)}$ 的边际似然。

利用深度递归神经网络(RNN)将协变量到状态空间模型参数的映射 ψ 参数化。图 1 为整个模型结构的框架。给定与时间序列 $\boldsymbol{z}_t^{(i)}$ 相关的协变量 $\boldsymbol{x}_t^{(i)}$，带有 LSTM 单元和参数 $\boldsymbol{\Phi}$ 的多层递归神经网络，通过递归函数 h(无边界条件)计算：

$$\boldsymbol{h}_t^{(i)} = h(\boldsymbol{h}_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}, \boldsymbol{\Phi}) \tag{6}$$

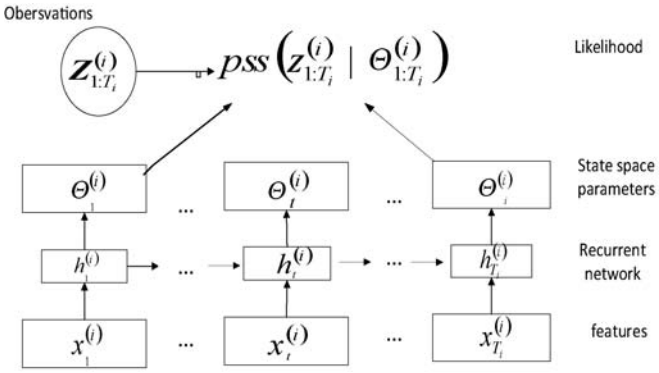

图 1 状态空间模型的框架

最后一层 LSTM(见图 1 中的 Recurrent network 层)的实值输出向量映射到状态空间模型参数 $\boldsymbol{\Theta}_t^{(i)}$ 的方法是应用仿射映射，然后进行适当的元素变换，将参数约束到适当的范围。使用参数 $\boldsymbol{\Theta}_t^{(i)}$ 计算给定观测值 $z_t^{(i)}$ 的似然函数，该似然函数用于学习网络参数 $\boldsymbol{\Phi}$。

训练时将训练区间内观测到数据 $\{\boldsymbol{z}_{1:T_i}^{(i)}\}_{i=1}^N$ 的概率最大化，即通过最大似然函数学习模型参数 $\boldsymbol{\Phi}^* = \underset{\boldsymbol{\Phi}}{\mathrm{argmax}} L(\boldsymbol{\Phi})$，其中：

$$L(\boldsymbol{\Phi}) = \sum_{i=1}^{N} \log p(\boldsymbol{z}_{1:T_i}^{(i)} \mid \boldsymbol{x}_{1:T_i}^{(i)}, \boldsymbol{\Phi}) = \sum_{i=1}^{N} \log p_{ss}(\boldsymbol{z}_{1:T_i}^{(i)} \mid \boldsymbol{\Theta}_{1:T_i}^{(i)}) \tag{7}$$

将式(7)中的每一个总和看作(负)损失函数,它度量给定输入 $\boldsymbol{x}_{1:T_i}^{(i)}$ 时 RNN 产生的状态空间模型参数 $\boldsymbol{\Theta}_{1:T_i}^{(i)}$ 与真实观测值 $z_{1:T_i}^{(i)}$ 之间的相容性。每一项都是线性高斯状态空间模型下的标准似然,可以通过卡尔曼滤波来进行。这主要涉及矩阵-矩阵、矩阵-向量乘法,使用神经网络框架(MXNet)实现整体对数似然计算,并使用自动微分获得关于参数 $\boldsymbol{\Phi}$ 的梯度,用于基于随机梯度下降的优化过程。

通过训练参数 $\boldsymbol{\Phi}$ 得到极大似然估计后,就可以对每个给定的时间序列进行概率预测。给定 $\boldsymbol{\Phi}$ 可以计算每个时间序列在预测范围内的联合分布,该联合分布是多元高斯分布。在实践中用 K 个蒙特卡洛样本来表示预测分布通常更为方便,公式为:

$$\hat{\boldsymbol{z}}_{k,T_i+1:T_i+\tau}^{(i)} \sim p(\boldsymbol{z}_{T_i+1:T_i+\tau}^{(i)} \mid \boldsymbol{z}_{1:T_i}^{(i)}, \boldsymbol{x}_{1:T_i+\tau}^{(i)}, \boldsymbol{\Theta}_{1:T_i+\tau}^{(i)}) \quad k = 1,2,\cdots,K \tag{8}$$

为了从状态空间模型生成预测样本,从样本 $\boldsymbol{l}_T \sim p(\boldsymbol{l}_T \mid z_{1:T})$ 开始,递归地应用:

$$y_{T+t} = \boldsymbol{a}_{T+t}^{\mathrm{T}} \boldsymbol{l}_{T+t-1} + b_{T+t} \qquad t = 1,2,\cdots,\tau$$

$$\hat{z}_{T+t} = y_{T+t} + \sigma_{T+t}\varepsilon_{T+t} \quad \varepsilon_{T+t} \sim N(0,1) \quad t = 1,2,\cdots,\tau$$

$$\boldsymbol{l}_{T+t} \sim \boldsymbol{F}_{T+t}\boldsymbol{l}_{T+t-1} + g_{T+t}\varepsilon_{T+t} \quad \varepsilon_{T+t} \sim N(0,1) \qquad t = 1,2,\cdots,\tau - 1$$

计算每个时间序列 $\boldsymbol{z}_{1:T_i}^{(i)}$ 的后验 $p(\boldsymbol{l}_{T_i}^{(i)} \mid \boldsymbol{z}_{1:T_i}^{(i)})$,在训练区间内展开 RNN 网络,获得 $\boldsymbol{\Theta}_{1:T_i}^{(i)}$,然后使用卡尔曼滤波算法,拓展预测区间 $t = T_i + 1, T_i + 2, \cdots, T_i + \tau$,得到 $\boldsymbol{\Theta}_{T_i+1:T_i+\tau}^{(i)}$,通过递归应用上述方程 K 次,生成预测样本。

在 DSSM 中,与经典的 SSM 和基于深度学习的自回归模型(如 DeepAR)相比,目标值并没有直接用作输入,这就带来了几个优点。首先,目标值只是适当考虑噪声的似然项的合并,故模型对噪声更为鲁棒;然后,简单地删除相应的似然项,就可以很容易地处理丢失的目标值;最后,生成预测样本路径的计算效率也更高,因为整个预测过程中 RNN 只需要展开一次(与样本数无关)。

2.2　深度自回归模型(DeepAR)

DeepAR 由一个 RNN(使用 LSTM 或 GRU 单元)组成,该 RNN 以序列滞后值和协变量作为输入,训练和预测遵循自回归模型的一般方法。

与传统模型不同的是,DeepAR 不仅将最后的目标值作为输入,而且还将一些滞后项作为输入。例如,对于小时数据,滞后可能是 1(前一小时)、1×24(前一天)、2×24(前两天)、7×24(前一周)等。

用 $z_{i,t}$ 表示时间序列 i 在时间 t 的值,在给定过去 $\{z_{i,1}, z_{i,2}, \cdots, z_{i,t_0-2}, z_{i,t_0-1}\} := \boldsymbol{z}_{i,t_0-1}$ 的前提下,建立未来每个时间序列 $\{z_{i,t_0}, z_{i,t_0+1}, \cdots, z_{i,T}\} := \boldsymbol{z}_{i,t_0:T}$ 的条件概率分布:

$$p(\boldsymbol{z}_{i,t_0:T} \mid \boldsymbol{z}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T})$$

式中:t_0 表示预测开始的时间点;$x_{i,1:T}$ 为在所有时间点都已知的协变量。

假设模型分布 $p(\boldsymbol{z}_{i,t_0:T} \mid \boldsymbol{z}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T})$ 由似然因子的乘积组成(无边界条件):

$$p(\boldsymbol{z}_{i,t_0:T} \mid \boldsymbol{z}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T}) = \prod_{t=t_0}^{T} p(z_{i,t} \mid \boldsymbol{z}_{i,1:t-1}, \boldsymbol{x}_{i,1:T}) = \prod_{t=t_0}^{T} \ell(z_{i,t} \mid \theta(\boldsymbol{h}_{i,t}, \boldsymbol{\Theta})) \tag{9}$$

由输出 $\boldsymbol{h}_{i,t}$ 参数化的自回归递归网络,$\boldsymbol{h}_{i,t} = h(\boldsymbol{h}_{i,t-1}, \boldsymbol{z}_{i,t-1}, \boldsymbol{x}_{i,t}, \boldsymbol{\Theta})$,其中 h 是由具有 LSTM 单元的多层递归神经网络实现的函数。该模型是自回归的,最后时刻的观测值 $z_{i,t-1}$ 以及递归网络的先前输出 $\boldsymbol{h}_{i,t-1}$ 都会作为下一时刻的输入。似然函数 $\ell(z_{i,t} \mid \theta(\boldsymbol{h}_{i,t}, \boldsymbol{\Theta}))$ 是一个固定分布,其参数由网络输出 $\boldsymbol{h}_{i,t}$ 的函数 $\theta(\boldsymbol{h}_{i,t}, \boldsymbol{\Theta})$ 给出。

$\boldsymbol{z}_{i,1:t_0-1}$ 中的观测值信息通过初始状态 $\boldsymbol{h}_{i,t_0-1}$ 传递到预测范围。在 sequence-to-sequence 的设置中,此初始状态是编码器网络的输出。一般来说,这个编码器网络可以有不同的结构,在这里选择在条件区间和预测区间(对应于 sequence-to-sequence 模型中的编码器和解码器)中对模型使用相同的结构。此外,它们之间共享权重,以便计算 $t = 1,2,\cdots,t_0 - 1$ 时解码器的初始状态 $\boldsymbol{h}_{i,t_0-1}$。编码器 $\boldsymbol{h}_{i,0}$ 以及 $z_{i,0}$ 的初始状态初始化为零。

给定模型参数 $\boldsymbol{\Theta}$,可以通过祖先采样法直接获得联合样本 $\tilde{\boldsymbol{z}}_{i,t_0:T} \sim p(\boldsymbol{z}_{i,t_0:T} \mid \boldsymbol{z}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T})$。计算当 $t = 1,2,\cdots,t_0$ 时的 $\boldsymbol{h}_{i,t_0-1}$。对于 $t = t_0, t_0+1, 2, \cdots, T$,$\tilde{\boldsymbol{z}}_{i,t} \sim \ell(\cdot \mid \theta(\tilde{\boldsymbol{h}}_{i,t}, \boldsymbol{\Theta}))$ 中 $\tilde{\boldsymbol{h}}_{i,t} = h(\boldsymbol{h}_{i,t-1}, \tilde{\boldsymbol{z}}_{i,t-1}, \boldsymbol{x}_{i,t}, \boldsymbol{\Theta})$ 初始化为 $\tilde{\boldsymbol{h}}_{i,t_0-1} = \boldsymbol{h}_{i,t_0-1}$ 和 $\tilde{\boldsymbol{z}}_{i,t_0-1} = z_{i,t_0-1}$。获得的模型样本可用于计算各种不同的量。

图 2 为 DeepAR 模型的概述。(a)为模型的训练阶段,在每个时间点 t,网络的输入变量为 $x_{i,t}$、$z_{i,t-1}$ 和 $\boldsymbol{h}_{i,t-1}$,然后利用网络输出 $\boldsymbol{h}_{i,t} = h(\boldsymbol{h}_{i,t-1}, z_{i,t-1}, \boldsymbol{x}_{i,t}, \boldsymbol{\Theta})$ 计算似然函数 $\ell(z \mid \theta)$ 的参数 $\theta_{i,t} = \theta(\boldsymbol{h}_{i,t}, \boldsymbol{\Theta})$,用于训练模型参数;(b)为模型的预测阶段,将时间序列 $z_{i,t}$ 中 $t < t_0$ 的历史数据导入,然后在 $t \geqslant t_0$ 随机采集样本

得到 $\tilde{z}_{i,t} \sim \ell(\cdot \mid \theta_{i,t})$ 并反馈到下一个时间点，直到预测范围 $t = t_0 + T$ 结束时生成一个样本轨迹，重复这个预测过程会产生许多表示联合预测分布的轨迹。

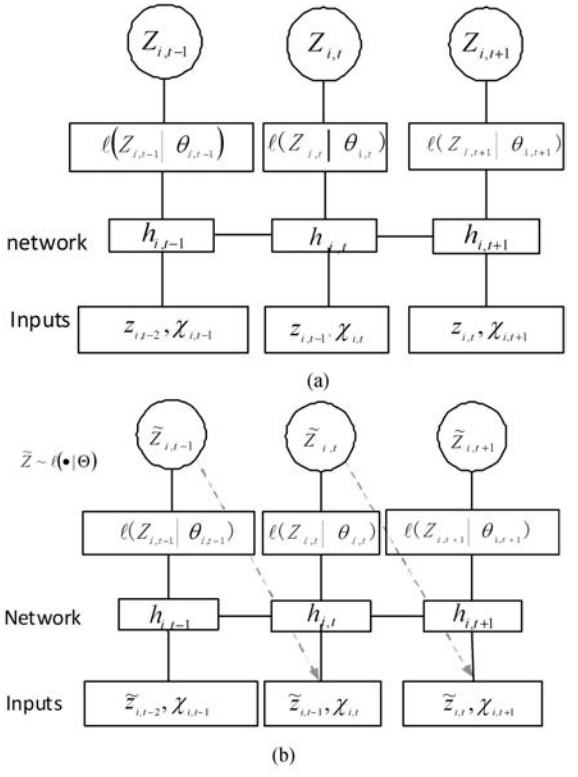

图 2 DeepAR 模型摘要

2.3 Transformer 模型

Transformer 完全基于 Attention Mechanisms，而不需要递归和卷积。递归模型通常沿输入和输出序列的符号位置进行因子计算。将位置与计算的时间步对齐，生成隐藏状态 $\boldsymbol{h}_t$ 的序列，作为先前隐藏状态 $\boldsymbol{h}_{t-1}$ 和位置 t 的输入函数。这种固有的序列性质使得训练无法并行化，而在较长的序列研究中，因为内存约束限制，并行化至关重要。Transformer 完全依赖于 Attention Mechanisms 来绘制输入和输出之间的全局依赖关系，允许更显著的并行化[18]。

时间序列概率预测的目标是建立以下条件概率分布：

$$p(\boldsymbol{z}_{i,t_0+1:t_0+\tau} \mid \boldsymbol{z}_{i,1:t_0}, \boldsymbol{x}_{i,1:t_0+\tau}; \boldsymbol{\Phi}) = \prod_{t=t_0+1}^{t_0+\tau} p(z_{i,t} \mid \boldsymbol{z}_{i,1:t-1}, \boldsymbol{x}_{i,1:t}; \boldsymbol{\Phi}) \quad (10)$$

具体预测过程中将此问题简化为学习一步预测模型 $p(z_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:t}; \boldsymbol{\Phi})$，其中 $\boldsymbol{\Phi}$ 表示由所有时间序列集合共享的可学习参数。为了充分利用观测值和外生变量，将它们连接起来得到一个增广矩阵（无边界条件）$y_t \triangleq [\boldsymbol{z}_{t-1} \circ \boldsymbol{x}_t] \in \mathbf{R}^{d+1}$，$\boldsymbol{Y}_t = [y_1, y_2, \cdots, y_t]^{\mathrm{T}} \in \mathbf{R}^{t \times (d+1)}$ 式中：$\circ$ 代表两个向量的拼接。观测值和外生变量作为整体输入变量，探讨一个合适的 $\boldsymbol{z}_t \sim f(\boldsymbol{Y}_t)$ 模型来预测给定 $\boldsymbol{Y}_t$ 条件下 $\boldsymbol{z}_t$ 的概率分布。

利用 multi-head self-attention 机制，用 Transformer 实例化 f，因为 self-attention 使 Transformer 能够捕获长期和短期依赖，并且不同的 attention heads 学习时间模型的不同方面。这些优点使 Transformer 成为时间序列预测的一个很好的预选方法。

图 3 为 Transformer 模型概述。大多数竞争性神经序列转导模型都具有编码器-解码器结构。这里，编码器将由符号表示的输入序列 $(x_1, x_2, \cdots, x_n)$ 映射到连续表示的序列 $\boldsymbol{z} = (z_1, z_2, \cdots, z_n)$。给定 $\boldsymbol{z}$，解码器一次生成一个符号的输出序列 $(y_1, y_2, \cdots, y_n)$。每一步，模型都是自回归的，在生成下一步时，将先前生成的符号作为附加输入。Transformer 遵循这个总体架构，使用堆叠的 self-attention 和 point-wise 作为编码器和解码器完全连接层，如图 3 所示。

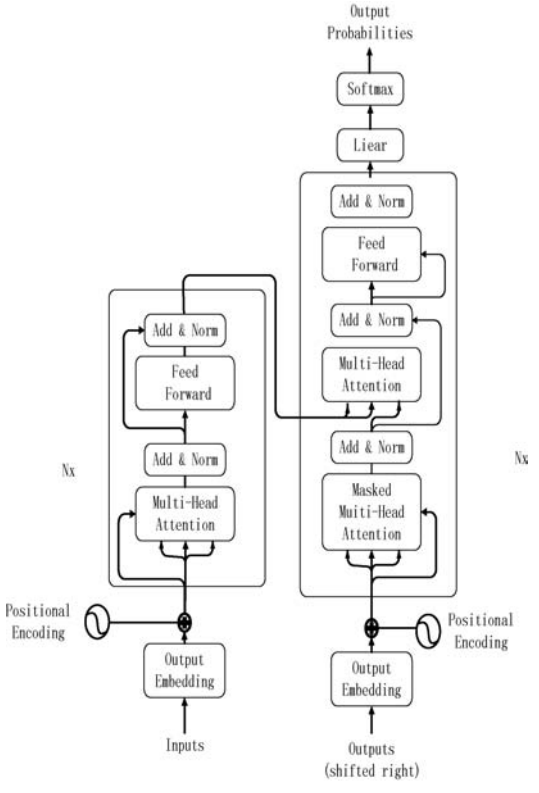

图 3 Transformer 模型摘要

其中 attention 函数可以描述为将 query 和一组键值对映射到输出，query、键、值和输出都是向量。输出是值的加权和，分配给每个值的权重由 query 的兼容函数和相应的键计算得出。

在 self-attention 层中，一个 multi-head self-attention 子层同时将 Y_t 分别转换成 H 个不同的查询矩阵 $\boldsymbol{Q}_h = Y_t \boldsymbol{W}_h^Q$，关键矩阵 $\boldsymbol{K}_h = Y_t \boldsymbol{W}_h^K$ 和价值矩阵 $\boldsymbol{V}_h = Y_t \boldsymbol{W}_h^V$，其中 $h = 1, 2, \cdots, H$。这里 $\boldsymbol{W}_h^Q, \boldsymbol{W}_h^K \in \mathbf{R}^{(d+1) \times d_k}$，$\boldsymbol{W}_h^V \in \mathbf{R}^{(d+1) \times d_v}$ 是可学习的参数。在这些线性投影之后，标度数量积 attention 计算向量输出序列：

$$\boldsymbol{O}_h = \text{Attention}(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \text{soft max}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^{\mathrm{T}}}{\sqrt{d_k}} \cdot \boldsymbol{M}\right)\boldsymbol{V}_h \tag{11}$$

为了避免将来的信息泄漏,使用 mask 矩阵 $\boldsymbol{M}$ 通过将所有上三角元素设置为 $-\infty$ 来过滤 rightward attention。之后,$\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_H$ 被串联起来,再次线性投影。在 attention 输出端,叠加一个位置前馈子层,该子层具有两层完全连接的网络,中间有一个 ReLU 激活[19]。

3　实证分析

3.1　研究背景

2018 年 3 月,美国使用"232 措施"对进口钢、铝产品分别加征 25% 和 10% 关税。2018 年 6 月,美国贸易代表办公室公布修订版的"301"对华加征 25% 关税的产品清单,并在 2018 年 7 月和 8 月分两批对从中国进口的 500 亿美元商品加征 25% 关税,涉及的行业主要有通用设备、电气机械、专用设备、通信电子设备、仪器仪表等 5 个设备制造业,其余为橡胶和塑料制品业、金属制品业等行业。2018 年 9 月又对 2 000 亿中国输美产品征收 10% 的关税(《关于中美经贸摩擦的事实与中方立场》白皮书 2018)。为了了解中美贸易摩擦对上海市出口贸易的影响,利用深度学习方法预测上海市出口额数据,探究在中美贸易摩擦不断升温背景下上海市出口额的变化发展规律。

3.2　数　据

本文使用的数据来自"上海海关数据库"(http://shanghai.customs.gov.cn),数据集是上海市总出口额和上海市对美国市场的出口额数据,该数据集一共有 72 个时间点的数据,涵盖了从 2014 年 1 月份开始到 2019 年 12 月结束的每个月上海市总出口额和上海市对美国市场的出口贸易额信息。每个时间点以月为单位,出口额的单位是亿元人民币。图 4 和图 5 分别为上海市总出口额和上海市对美国市场出口额的时序图。

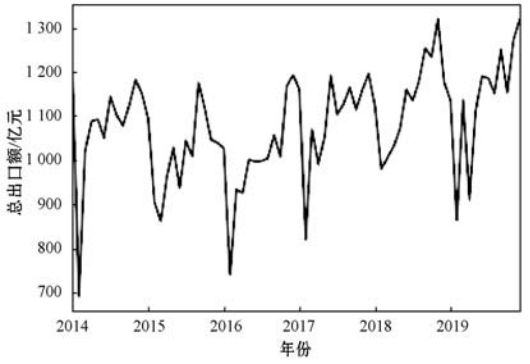

图 4　上海市总出口额

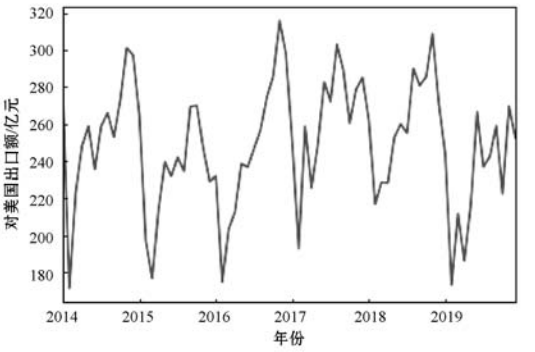

图 5　上海市对美国市场出口额

可以看出,上海市总出口额和上海市对美国市场的出口额两者存在相似的趋势,并且时序图的走势存在一定的周期性规律,在每年年初和年末的时候会出现下降的趋势,而年中大部分时间存在上升的趋势。

3.3　计量方法

本文采用五种模型对上海市总出口额和上海市对美国市场的出口额作预测,分别是自回归求和移动平均模型(ARIMA)[20]、条件时序卷积模型(Wavenet)[21]、深度状态空间模型(DSSM)、深度自回归模型(DeepAR)和 Transformer 模型(Transformer),并对五种模型的预测效果进行比较。用连续分级概率评分(CRPS)对模型的预测效果进行评价。

连续分级概率评分(Continuous Ranked Probability Score,CRPS)或"连续概率排位分数"是一个函数或统计量,可以度量概率分布 F(由分位数函数 F^{-1} 表示)与观测值 z 的相容性[22]。CRPS 可视为平均绝对误差(Mean Absolute Error, MAE)在连续概率分布上的推广。CRPS 可以作为概率模型的损失函数和评价函数,应用于概率天气预报、误差分析、异常值检测(Anomaly Detection)等现实问题。作为评价函数时,按 CRPS 评价概率模型所得的(优劣)结果与按 MAE 评价概率模型的数学期望所得的结果等价。

在分位数水平为 $\alpha \in [0,1]$ 且预测的第 α 分位数为 q 的条件下,pinball 损失(或分位数损失)定义为:

$$\Lambda_\alpha(q,z) = (\alpha - I(z < q))(z - q) \tag{12}$$

式中:z 是观测值;$I(z < q)$ 是示性函数,如果 $z < q$ 为 1,否则为 0。CRPS 具有一个直观的定义,即在所有分位数水平 $\alpha \in [0,1]$ 上对 pinball 损失的积分:

$$CRPS(F^{-1},z) = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha),z)\,\mathrm{d}\alpha \tag{13}$$

CRPS 作为一个适用的评分规则[22],意味着当预测分布等于从实际数据中得出的分布时,CRPS 值最小。CRPS 值越小,说明预测分布与观测值分布相近,预测性能越好。

3.4 实证结果

本文使用 GluonTS 时间序列预测框架进行预测(http://gluon-ts. mxnet. io/index. html),GluonTS 是亚马逊推出的一种使用 Gluon API 的 MXNet 时间序列分析工具包。

利用以上五种模型对上海市每月贸易总出口额(TS1)和上海市每月对美国市场的贸易出口额(TS2)进行预测和预测效果评估,样本数据区间是 2014 年 1 月到 2019 年 12 月,训练期是 2014 年 1 月到 2018 年 12 月,测试期是 2019 年 1 月到 2019 年 12 月。表 1 给出五种预测模型 CRPS 值,可以看出,深度状态空间模型(DSSM)、深度自回归模型(DeepAR)和 Tansformer 模型的预测 CRPS 值相对较小,表明这三种方法预测效果较好,并且预测效果都明显优于传统的自回归求和移动平均模型(ARIMA),其中 Transformer 模型的预测效果是相对最优。图 6 和图 7 是 Transformer 模型的预测效果图。

表 1 五种模型预测 CRPS 值

序列	Arima	DeepAR	DSSM	Transformer	WaveNet
TS1	65.58	41.340	43.922	**39.637**	86.344
TS2	32.36	20.675	21.633	**12.835**	49.098

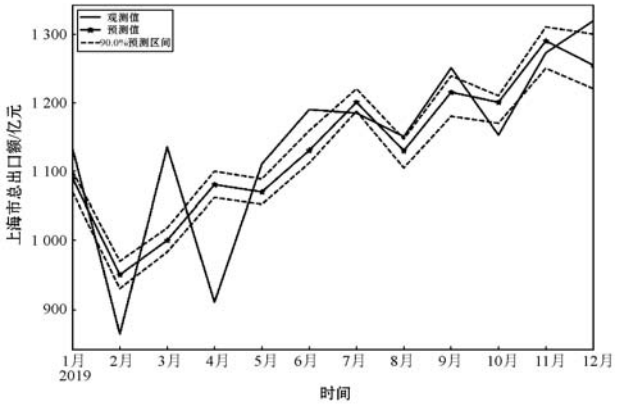

图 6 利用 Transformer 模型预测上海市总出口额

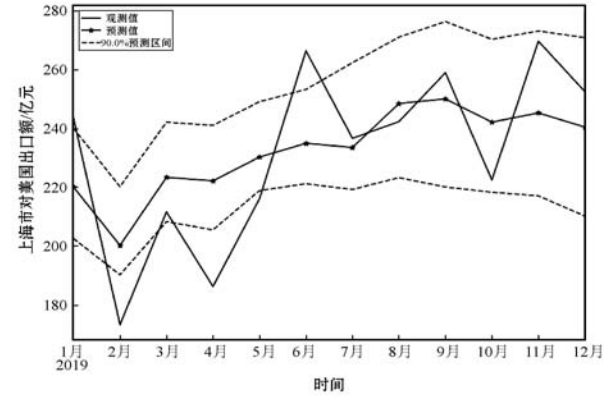

图 7 利用 Transformer 模型预测出口市场为美国的上海市出口额

4 结 语

尽管传统的统计建模方法将结构假设合并到模型中,使得模型易于解释,但是在现代预测应用中,传统统计模型对时间序列单独建模,这就需要大量的劳动和计算成本。深度学习方法恰好可以识别时间序列内部和时间序列之间的复杂模式,所需的人力要少得多,但是这些模型所做的结构假设较少,很难解释,通常需要更大的训练数据集来学习得到准确的模型。由此产生了将传统统计模型与深度学习融合的一些新的预测方法,这些方法较好地克服两方面的不足。它们既允许模型自动提取特征并学习复杂的时间模式,同时也可以实施和利用时间平滑等假设,使模型可解释。本文在综述时间序列预测与深度学习文献的基础上,重点介绍三种与深度学习相结合的时间序列预测模型,并利用这些模型预测中美贸易摩擦背景下的上海市出口额数据。实验结果表明,相比于传统的时间序列预测方法(ARIMA 模型),基于深度学习的时间序列预测方法的预测 CRPS 值显著降低,预测效果更优。

参 考 文 献

[1] Faloutsos C, Flunkert V, Gasthaus J, et al. Forecasting big time series: theory and practice[C]//The Twenty-fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[2] Rangapuram S S, Seeger M, Gasthaus J, et al. Deep state space models for time series forecasting[C]//The Thirty-first Conference on Neural Information Processing Systems, 2018.

[3] Chung J, Kastner K, Dinh L, et al. A recurrent latent variable model for sequential data[C]//28th International Conference on Neural Information Processing Systems, 2015.

[4] Fraccaro M, Sønderby S K, Paquet U, et al. Sequential neural models with stochastic layers[C]//30th International Conference on Neural Information Processing Systems, 2016.

[5] Krishnan R G, Shalit U, Sontag D. Structured inference networks for nonlinear state space models[C]//Thirty-first AAAI Conference on Artificial Intelligence, 2017.

[6] Eleftheriadis S, Nicholson T, Deisenroth M, et al. Identification of gaussian process state space models[C]//31st Conference on Neural Information Processing Systems, 2017.

[7] Salinas D, Bohlke-schneider M, Callot L, et al. High-Dimensional multivariate forecasting with low-rank gaussian copula processes[C]//The Thirty-second Conference on Neural Information Processing Systems, 2019.

(下转第 84 页)

个整体的解决方案,对联合任务卸载和资源分配进行优化,从而最大限度地提高用户的卸载收益。首先,为每个用户的卸载效用建模,将 JTORA 问题转化为 MINLP 问题。然后,采用 Tammer 分解方法将高复杂度的原始问题转化为等效的主问题和一组复杂度较低的子问题。最后,利用本文提出的低复杂度启发式算法,以次优解的方式解决 JTORA 问题,实现共同优化任务卸载决策、用户上行链路传输功率的目标。仿真结果表明,本文提出的优化策略的平均系统效能明显优于其他方案,提出的启发式算法能够很好地实现最优解,显著提高系统的平均卸载效率。

参 考 文 献

[1] 于博文,蒲凌君,谢玉婷,等. 移动边缘计算任务卸载和基站关联协同决策问题研究[J]. 计算机研究与发展,2018,55(3):537 - 550.

[2] Li S, Zhai D, Du P, et al. Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks[J]. Science China Information Sciences, 2019, 62(2): 29307 - 29309.

[3] Tran T X, Hajisami A, Pandey P, et al. Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges[J]. IEEE Communications Magazine, 2017, 55(4): 54 - 61.

[4] Roman R, Lopez J, Mambo M. Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges[J]. Future Generation Computer Systems, 2018, 78: 680 - 698.

[5] Tran T X, Hosseini M P, Pompili D. Mobile edge computing: Recent efforts and five key research directions[J]. IEEE COMSOC MMTC Communications-Frontiers, 2017, 12(4): 29 - 33.

[6] Chen M, Hao Y. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587 - 597.

[7] Mao Y, Zhang J, Letaief K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590 - 3605.

[8] You C, Huang K. Multiuser resource allocation for mobile-edge computation offloading[C]//2016 IEEE Global Communications Conference(GLOBECOM), 2016.

[9] Lyu X, Tian H, Sengul C, et al. Multiuser joint task offloading and resource optimization in proximate clouds[J]. IEEE Transactions on Vehicular Technology, 2016, 66(4): 3435 - 3447.

[10] 徐佳,李学俊,丁瑞苗,等. 移动边缘计算中能耗优化的多重资源计算卸载策略[J]. 计算机集成制造系统,2019,25(4):954 - 961.

[11] Chen X, Pu L, Gao L, et al. Exploiting massive d2d collaboration for energy-efficient mobile edge computing[J]. IEEE Wireless Communications, 2017, 24(4): 64 - 71.

(上接第 70 页)

[8] Salinas D, Flunkert V, Gasthaus J. DeepAR: probabilistic forecasting with autoregressive recurrent networks[J]. International Journal of Forecasting,2019,36(3):1181 - 1191.

[9] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//The Thirtieth Conference on Neural Information Processing Systems,2017.

[10] 徐超,项薇,季孟忠,等. 基于 ARIMA 与自适应过滤法的组合预测模型研究[J]. 计算机应用与软件,2018,35(11):296 - 300,320.

[11] 沈旭东. 基于深度学习的时间序列算法综述[J]. 计算机应用技术,2019(1):71 - 76.

[12] 吴双双. 基于深度神经网络的时间序列预测技术研究[D]. 南京:南京理工大学,2017.

[13] 权钲杰. 基于深度神经网络集成和信息论学习的时间序列预测[D]. 杭州:浙江大学,2019.

[14] 刘峰,瞿俊. 基于聚类分析和神经网络的时间序列预测方法[J]. 微电子学与计算机,2006,23(9):85 - 87.

[15] 王慧健,刘峥,李云,等. 基于神经网络语言模型的时间序列趋势预测方法[J]. 计算机工程,2019(7):13 - 19.

[16] 李洁,林永峰. 基于多时间尺度 RNN 的时序数据预测[J]. 计算机应用与软件,2018,35(7):33 - 37,62.

[17] 蒋倩仪. 基于时间序列预测的股票交易决策建议系统[J]. 计算机应用与软件,2017,34(4):75 - 81,104.

[18] Alexandrov A, Benidis K, Bohlkeschneider M, et al. GluonTS: probabilistic time series models in python [EB]. arXiv:1906.05264, 2019.

[19] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[C]//The Thirty-second Conference on Neural Information Processing Systems,2019.

[20] Ziegel E R, Box G E P, Jenkins G M. Time series analysis, forecasting, and control[J]. Journal of Time,1976,31(2): 238 - 242.

[21] Oord A V D, Dieleman S, Zen H, et al. WaveNet: a generative model for raw audio[EB]. arXiv:1609.03499, 2016.

[22] Gneiting T, Raftery A E. Strictly proper scoring rules, prediction, and estimation[J]. Journal of the American Statistical Association, 2007, 102(477): 359 - 378.