NEURAL BASIS EXPANSION ANALYSIS FORINTERPRETABLE TIME SERIES FORECASTING-CSDN博客

成就一亿技术人!

20-25分钟

ICLR 2020

0 摘要

本文重点研究了利用深度学习解决单变量时间序列点预测问题。

我们提出了一种**基于后向和前向残留链路和一个非常深的全连接层堆栈的深度神经结构。**

该体系结构具有许多令人满意的特性，这些特性是可解释的，适用于广泛的目标领域而无

需修改，并且可以快速训练。

我们在几个著名的数据集上测试了提出的体系结构，包括M3、M4和旅游竞赛数据集，这些数据集包含了来自不同领域的时间序列。

我们展示了两个配置下N-BEATS模型 在所有数据集 中的最先进的性能：相比于统计基准，提高了预测准确性11%，相比于去年M4比赛的冠军，提升了3%。

我们模型的第一个配置不使用任何特定于时间序列的组件，它在异构数据集上的表现强烈地表明，与普遍接受的智慧相反，深度学习原语(如残差块)本身就足以解决广泛的预测问题。

最后，我们演示了如何将所提议的架构加以扩充，以提供可解释的输出，而不会造成大的准确性损失。

1 简介

时间序列预测问题，不像计算机视觉或自然语言处理等领域【在这两个领域，深度学习(DL)技术现在已经牢牢占据优势】，仍有证据表明，深度学习和DL相比于超越经典的统计方

法，没有特别多的优势 （Statistical and machine learning forecasting methods: Concerns and ways forward）

本篇论文旨在探索纯deep learning架构在时间序列预测问题中的潜力。

此外，在可解释DL架构设计的背景下，我们有兴趣回答以下问题:我们能否在模型中注入合适的归纳偏差，使其内部操作更易于解释（即提取一些可解释的驱动因素，结合产生一个给定的预测）？

1.1 本文贡献

1.1.1 深度神经网络架构

我们所知,这是第一篇论文用实验结论证明：**没有使用时间序列特定组件、只依靠纯deep learning的模型**，在 M3, M4和旅游数据集上的表现 优于广泛使用的统计方法。

在我们看来，这为纯ML在时间序列预测中的应用提供了一个长期缺失的概念证明，并增强了继续推进该领域研究的动力。

1.1.2 时间序列问题中的可解释性深度学习

除了准确性方面的好处外，我们还表明设计一个具有可解释输出的架构是可行的，从业者可以以与传统分解技术(如“季节性-趋势-级别”方法)非常相同的方式使用该架构

2 问题定义

考虑离散时间条件下的单变量点预测问题。

给定一个长度为h的预测视界，一个长度为t的观测序列历史$[y_1,\dots,y_T]\in\mathbb{R}^T$，我们的任务是预测未来$\mathbf{y}\in\mathbb{R}^H=[y_{T+1},y_{T+2},\dots,y_{T+H}]$.

出于简化考虑，我们将一个长度为t≤t，以最后一个观测值yT结束的回溯窗口作为模型输入$\mathbf{x}\in\mathbb{R}^t=[y_{T-t+1},\dots,y_T]$.

我们记$\hat{y}$为y的预测值

下面的几个是用来衡量模型的预测效果的

RMSE、MAE等误差指标整理_UQI-LIUWJ的博客-CSDN博客

$$\text{sMAPE} = \frac{200}{H}\sum_{i=1}^{H}\frac{|y_{T+i} - \hat{y}_{T+i}|}{|y_{T+i}| + |\hat{y}_{T+i}|},$$

$$\text{MAPE} = \frac{100}{H}\sum_{i=1}^{H}\frac{|y_{T+i} - \hat{y}_{T+i}|}{|y_{T+i}|},$$

$$\text{MASE} = \frac{1}{H}\sum_{i=1}^{H}\frac{|y_{T+i} - \hat{y}_{T+i}|}{\frac{1}{T+H-m}\sum_{j=m+1}^{T+H}|y_j - y_{j-m}|},$$

$$\text{OWA} = \frac{1}{2}\left[\frac{\text{sMAPE}}{\text{sMAPE}_{\text{Naïve2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naïve2}}}\right].$$

` 这里m是数据的周期。

MAPE (Mean Absolute Percentage Error)，sMAPE (symmetric MAPE)和MASE (Mean Absolute scaling Error)是预测实践中的标准无标度度量。

sMAPE是通过预测和真实之间的平均值来缩放误差。

MASE根据naïve预测器的平均误差进行缩放，该预测器简单地复制了过去测量的倒数第m个周期的观测结果，因此考虑了周期性。

OWA(总体加权平均)是一个m4特定的指标，用于对参赛作品进行排名，其中sMAPE和MASE指标被标准化，因此经季节性调整的

naïve预测得到OWA = 1.0。

3 N-Beats

我们的架构设计方法依赖于几个关键原则。

- 首先，基本架构应该是简单simple和通用generic的，但具有表现力expressive。
- 其次，架构不应该依赖于时间序列特定的特性工程或输入缩放。这些先决条件让我们探索纯deep learning构在时间序列预测中的潜力。
- 最后，作为探索可解释性的先决条件，架构应该是可扩展的，以使其输出具有可解释性。

现在，我们将讨论这些原则如何与提议的体系结构结合起来

3.1 基本单元

> 我们提出所提议的基本单元模块具有分叉结构，如图1(左)所示。
>
> 在这一节中，我们将重点描述第l个块的操作(注意，为了简洁起见，在图1中删除了块索引l)。

第l个块接受输入xl并输出两个向量$\hat{x}_l$和$\hat{y}_l$。

对于模型中的第一个块，它的输入xl是整个模型的输入——一个一定长度的历史回溯窗口。

我们设置输入窗口的长度（回溯窗口的窗口长度）为预测未来窗口长度H的倍数，在我们的设置中，回溯窗口的窗口长度从2H到7H。

对于其余的块，它们的输入xl是前面单元块的残差输出。（后面会说明）

每个块有两个输出$\hat{x}_l$和$\hat{y}_l$：长H的前向预测$\hat{y}_l$、xl的最佳估计$\hat{x}_l$（被称为“backcast”。

基本单元模块内部由两部分组成。

这里的线性层就是一个简单的线性投影层

$$i.e.\ \theta_\ell^f = \mathbf{W}_\ell^f \mathbf{h}_{\ell,4}.$$

FC层是一个标准的全连接层，具有RELU非线性激活函数

$$\mathbf{h}_{\ell,1} = \mathrm{ReLU}(\mathbf{W}_{\ell,1}\mathbf{x}_\ell + \mathbf{b}_{\ell,1}).$$

第二部分由后向g_l^b和前向g_l^f基层组成，它们接受各自的前向θ_l^f和后向θ_l^b预测系数，生成 前向预测$\hat{y}_l$ 和后向预测$\hat{x}_l$。

$$\widehat{\mathbf{y}}_\ell = g_\ell^f(\theta_\ell^f) \text{ and } \widehat{\mathbf{x}}_\ell = g_\ell^b(\theta_\ell^b).$$

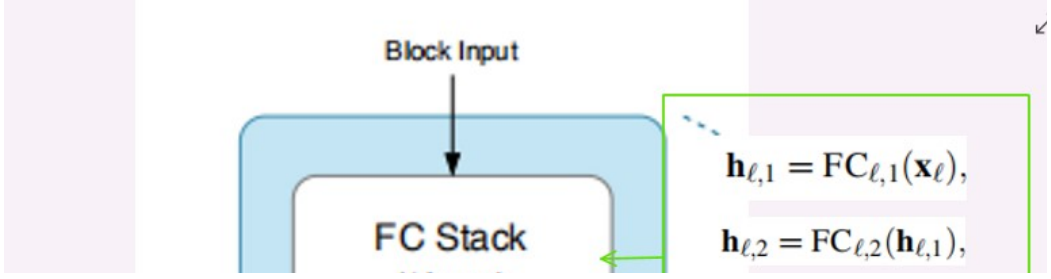
$$\widehat{\mathbf{y}}_\ell = \sum_{i=1}^{\dim(\theta_\ell^f)} \theta_{\ell,i}^f \mathbf{v}_i^f, \quad \widehat{\mathbf{x}}_\ell = \sum_{i=1}^{\dim(\theta_\ell^b)} \theta_{\ell,i}^b \mathbf{v}_i^b.$$

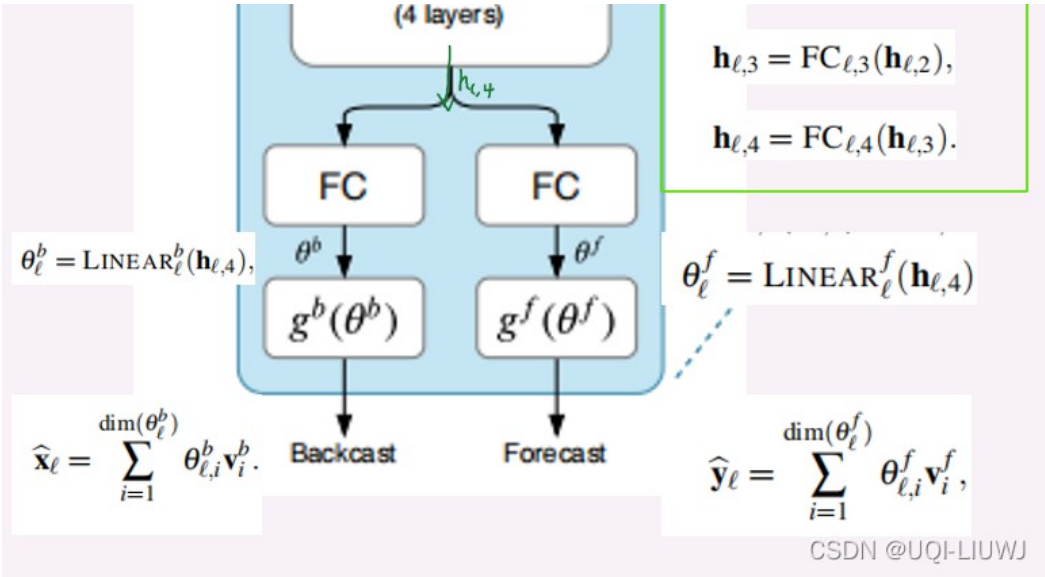
这v_i^f 和v_i^b分别是前向和后向基向量

g_l^b和g_l^f的作用是提供足够丰富的集合 $\{\mathbf{v}_i^f\}_{i=1}^{\dim(\theta_\ell^f)}$ and $\{\mathbf{v}_i^b\}_{i=1}^{\dim(\theta_\ell^b)}$，他们可以是可学习的；也可以设置为特定的功能形式，以反映特定问题的归纳偏差，以适当地限制输出结构。

这部分结构的一个任务是预测正向预测θ_l^f（每一个dim 一个θ值），最终目标优化 前向预测$\hat{y}_l$ 的准确性。

此外，该子网络预测xl的最佳估计 $\hat{x}_l$，其最终目标是通过移除输入数据中对于预测任务没有帮助的成分，来帮助下游模块更好地进行预测。





3.2 DOUBLY RESIDUAL STACKING 双重残差叠加

> 经典的残差网络架构在将结果传递给下一个模块之前，将此模块的输入添加到其输中。
>
> 机器学习笔记：ResNet及残差连接_UQI-LIUWJ的博客-CSDN博客
>
> Huang等人(2017)提出的DenseNet架构扩展了这一原则，从每个模块的输出到其后的每个模块的输入之间引入了额外的连接。
>
> NTU 课程 7454 （5） CNN进阶_UQI-LIUWJ的博客-CSDN博客
>
> 这些方法在提高深层架构的可训练性方面具有明显的优势。

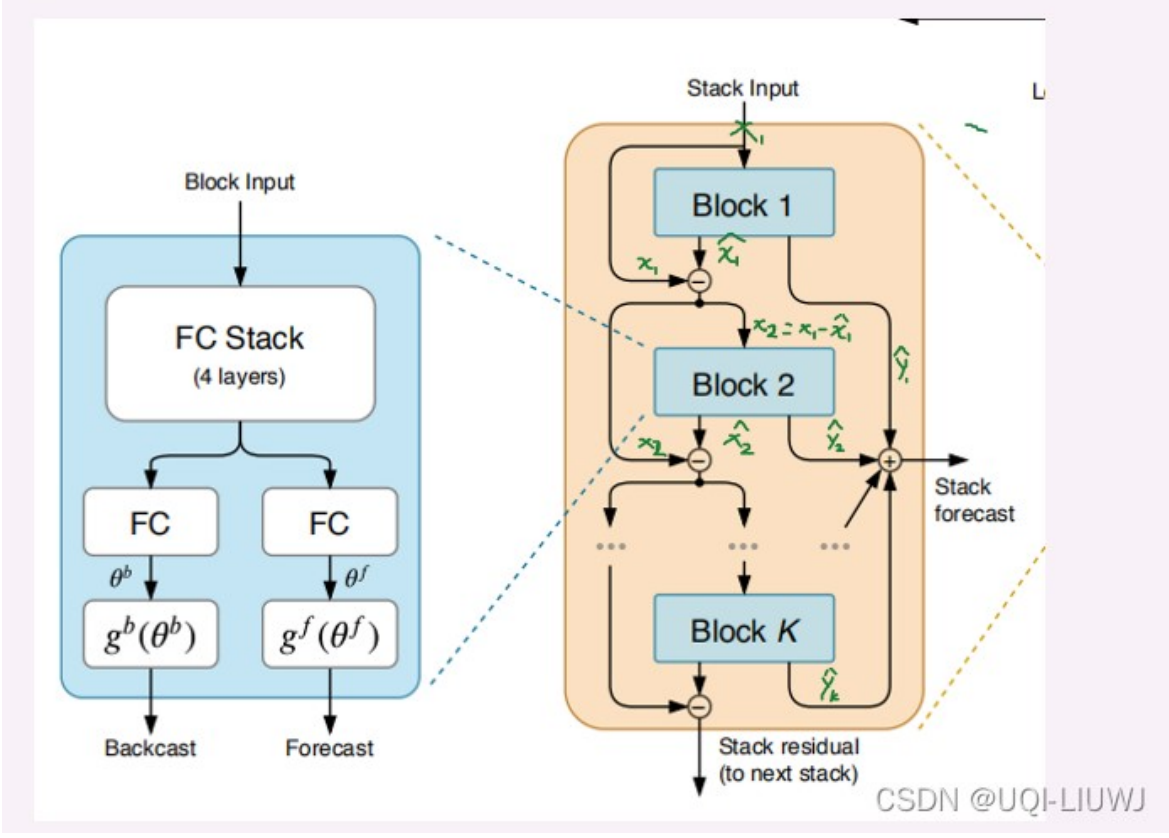
在这项工作的背景下，它们的缺点是它们导致了难以解释的网络结构。

我们提出了一种新的分层双残差拓扑结构，如图1(中间和右边)所示。

提出的体系结构有两个剩余分支，一个分支运行在每一层的backcast预测上，另一个分支运行在每一层的预测分支上。

它的运行由以下方程描述:

$$\mathbf{x}_\ell = \mathbf{x}_{\ell-1} - \hat{\mathbf{x}}_{\ell-1}, \quad \hat{\mathbf{y}} = \sum_\ell \hat{\mathbf{y}}_\ell.$$



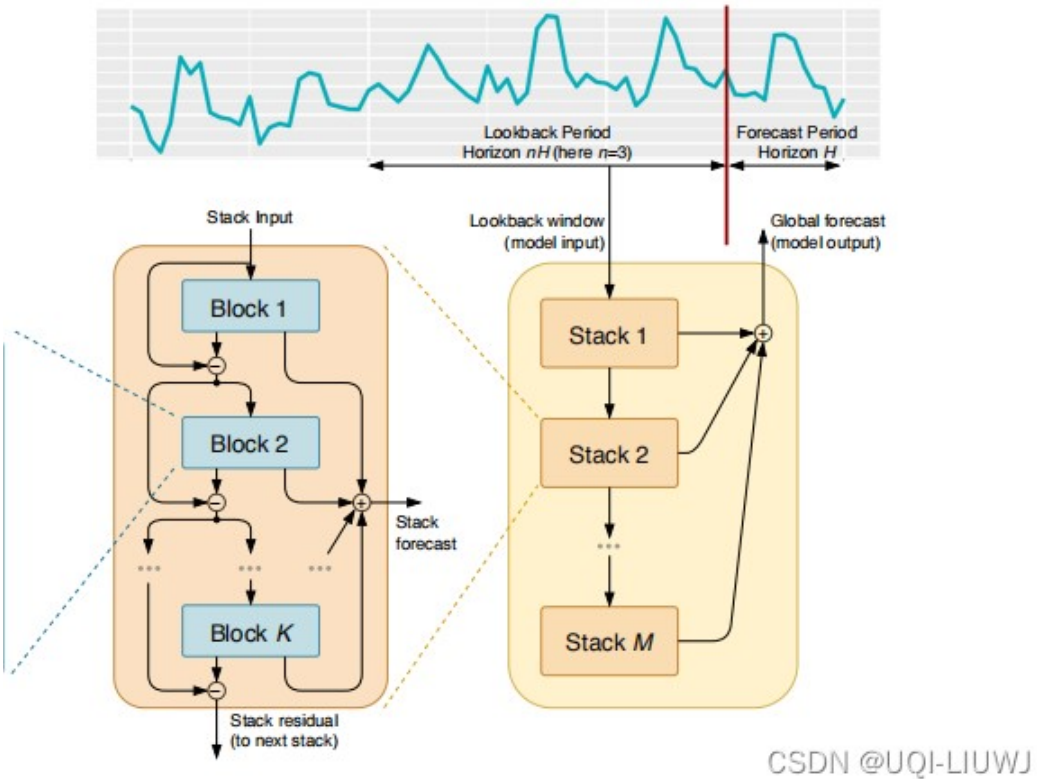
如前所述，在第一个块的特殊情况下，它的

输入是模型的全部输入x, x1≡x。

对于所有其他块，backcast残差分支xl可以被认为是对输入信号进行序列分析。

前一个基本模块去掉了信号 $\hat{x_{l-1}}$ 中可以很好近似的部分，使得下游区块的预测工作更加容易。(残差链接的好处)

这种结构还促进了梯度反向传播。(残差链接的好处)



更重要的是，每个基本模块输出它本身可以预测的部分前向预测 $\hat{y}_l$ ，这些前向预测提供了层级分解。

最后的预测是所有部分预测的总和。

当允许为每一个基本模块有自己的后向g_l^b和前向g_l^f时，这使得网络对梯度流更加透明。

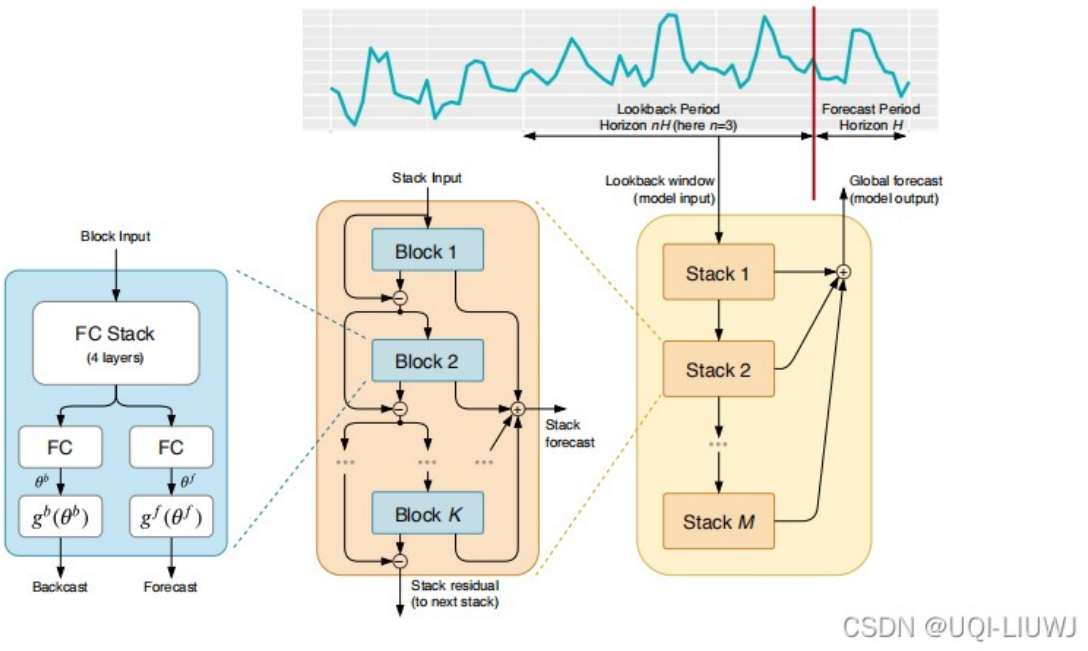
如果 后向g_l^b和前向g_l^f 共享一个基，此时通过聚合有意义的部分，对实现可解释性至关重要。（3.3将会涉及）

3.3 可解释性

基于g_l^b和g_l^f'的选择，我们提出了两种架构配置。

其中一种是通用深度学习，另一种增加了某些归纳偏差以使其可解释。

换句话说，整体的架构就是3.1和3.2描述的内容



通用架构不依赖于时间序列特定的知识。我们将g_l^b和g_l^f设置为前一层输出的线性投影。

在这种情况下，block l的输出被描述为:（b在3.1中式没有的）

$$\hat{\mathbf{y}}_\ell = \mathbf{V}_\ell^f \theta_\ell^f + \mathbf{b}_\ell^f, \quad \hat{\mathbf{x}}_\ell = \mathbf{V}_\ell^b \theta_\ell^b + \mathbf{b}_\ell^b.$$

其中V_l^f 的维度是$H \times \dim(\theta_\ell^f)$。第一个维度我们可以想成 预测区域 离散时刻的数量。因而V_l^f的每一列可以看成是一个时域波形。

因为V_l^f没有额外的限制，所以V_l^f的各个列（各个时域波形）没有内部的结构，因而V_l^f是不可解释的

3.3.1 可解释性结构

可解释的体系结构可以通过重用图1中的总体体系结构方法，以及在stack级别（图1中间的结构）向基层添加结构来构建。

预测人员经常使用如X13-ARIMA模型等将时间序列分解成趋势和季节性。

我们建议在模型中设计趋势和季节性分解，以使stack级输出更容易解释。

注意，对于通用模型来说，栈的概念是不必要的，并且为了清晰起见省略了栈级索引。

现在我们将同时考虑堆栈级和块级索引。例如，
$\hat{y}_{s,l}$
表示栈s中块l的局部预测'。

3.3.2 趋势模型

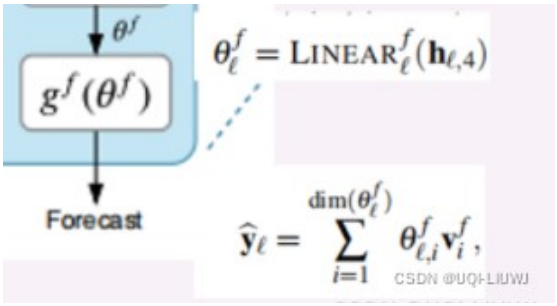
趋势的一个典型特征是，大多数时候它是一个单调的或者至少是一个缓慢变化的函数。

为了模拟这种行为，我们建议将$g^b_{s,l}$和$g^f_{s,l}$约束为一个p的多项式（p是一个小的degree值），一个在预测窗口中缓慢变化的函数:

$$\hat{\mathbf{y}}_{s,\ell} = \sum_{i=0}^{p} \boldsymbol{\theta}^f_{s,\ell,i} t^i.$$

其中向量t为
$\mathbf{t} = [0, 1, 2, \ldots, H-2, H-1]^T / H$
（H是预测窗口的长度）

换句话说，我们只是把基本模块里面的$v^f_{s,i}$替换成了t^i



用矩阵的形式表示，有：

$$\hat{\mathbf{y}}_{s,\ell}^{tr} = \mathbf{T}\theta_{s,\ell}^f,$$

其中

$\theta_{s,l}^f$

就是第s个stack，第l层的前向传播系数

$$\mathbf{T} = [\mathbf{1}, \mathbf{t}, \ldots, \mathbf{t}^p]$$

3.3.3 周期性模型

周期性的典型特征是它是一个有规律的，循环的，反复出现的波动。

为了模拟周期性，我们限制$g_{s,l}^b$和$g_{s,l}^f$属于一类周期函数，即$y_t = y_{t-\Delta}$,其中Δ是周期。

建立周期函数模型的基础自然选择是傅里叶级数:

$$\hat{\mathbf{y}}_{s,\ell} = \sum_{i=0}^{\lfloor H/2-1 \rfloor} \theta_{s,\ell,i}^f \cos(2\pi i t) + \theta_{s,\ell,i+\lfloor H/2 \rfloor}^f \sin(2\pi i t), \quad (3)$$

$i=0$

用矩阵形式表述，有：

$$\hat{\mathbf{y}}_{s,\ell}^{seas} = \mathbf{S}\theta_{s,\ell}^{f},$$

$$\mathbf{S} = [\mathbf{1}, \cos(2\pi\mathbf{t}), \ldots \cos(2\pi\lfloor H/2-1\rfloor\mathbf{t})), \sin(2\pi\mathbf{t}), \ldots, \sin(2\pi\lfloor H/2-1\rfloor\mathbf{t}))]$$

是正弦波矩阵

3.3.4 整体可解释结构

整个可解释架构由两个stack组成:趋势stack，周期性stack。

双重残差叠加结合预测/倒推原理可以使得

(i)趋势成分在x被输入到周期性stack之前被移除了【个人猜测是因为相同基的系数互相抵消】

(ii)趋势和周期性的预测作为单独的可解释输出。

从结构上讲，每个栈由图1所示的用残余连接连接的几个块组成，每个块共享其各自的、不可学习的$g_{s,l}^{b}$和$g_{s,l}^{f}$。

趋势和季节性的区块数都是3。

我们发现，在共享$g_{s,l}^{b}$和$g_{s,l}^{f}$的基础上，在堆栈中跨块共享所有权重可以获得更好的验证性

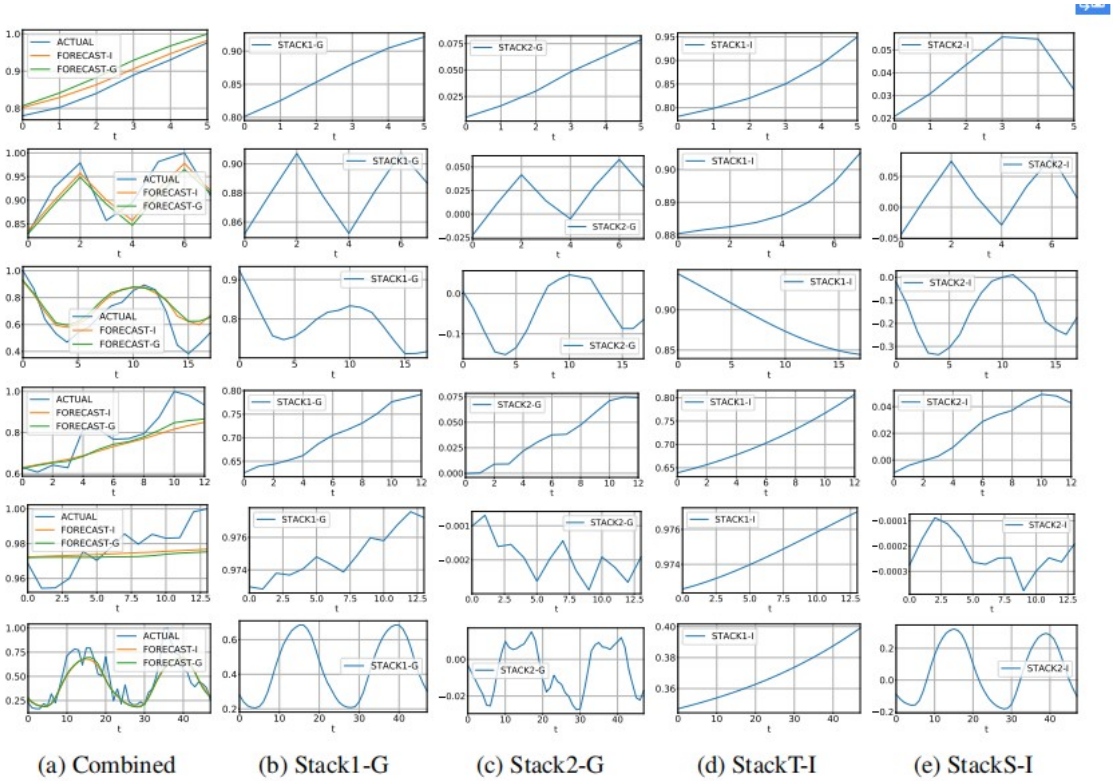
能。

*********集成学习部分暂时略去，后补**********

5 实验部分

论文中说的两个配置被记为：generic (N-BEATS-G)；interpretable (N-BEATS-I)

M4 Average (100,000)	sMAPE	OWA	M3 Average (3,003)	sMAPE	TOURISM Average (1,311)	MAPE
Pure ML	12.894	0.915	Comb S-H-D	13.52	ETS	20.88
Statistical	11.986	0.861	ForecastPro	13.19	Theta	20.88
ProLogistica	11.845	0.841	Theta	13.01	ForePro	19.84
ML/TS combination	11.720	0.838	DOTM	12.90	Stratometrics	19.52
DL/TS hybrid	11.374	0.821	EXP	12.71	LeeCBaker	19.35
N-BEATS-G	11.168	0.797		12.47		**18.47**
N-BEATS-I	11.174	0.798		12.43		18.97
N-BEATS-I+G	**11.135**	**0.795**		**12.37**		18.52



(a) Combined (b) Stack1-G (c) Stack2-G (d) StackT-I (e) StackS-I

Figure 2: The outputs of generic and the interpretable configurations, M4 dataset. Each row is one time series example per data frequency, top to bottom (Yearly: id Y3974, Quarterly: id Q11588, Monthly: id M19006, Weekly: id W246, Daily: id D404, Hourly: id H344). The magnitudes in a row

are normalized by the maximal value of the actual time series for convenience. Column (a) shows the actual values (ACTUAL), the generic model forecast (FORECAST-G) and the interpretable model forecast (FORECAST-I). Columns (b) and (c) show the outputs of stacks 1 and 2 of the generic model, respectively; FORECAST-G is their summation. Columns (d) and (e) show the output of the Trend and the Seasonality stacks of the interpretable model, respectively; FORECAST-I is their summation.

图2研究了提出的模型在通用和可解释配置下的输出。

我们以N-BEATS的通用配置输出作为对照组(将含有30个残差块的通用模型分为两个栈)。

在图2中将通用(后缀“G”)和可解释(后缀“-I”)栈的输出并排绘制出来。

通用模型的输出是任意的和不可解释的:无论是从趋势还是从抽周期的角度看，或者两者都出现在两个stack的输出中。

在第二个堆栈的输出处，输出的大小(峰对峰)通常较小。

可解释模型的输出表现出不同的特性:趋势输出是单调的、缓慢移动的；周期性输出是有规律的、周期性的、反复波动的。

如果在时间序列中存在显著的季节性，那么季节性产出的峰值比趋势产出的峰值要大得多。

同样地，当真实信号中没有明显的趋势时，趋势输出的峰值到峰值的幅度往往较小。

> 因此，提出的可解释架构将其预测分解为两个不同的组成部分。

我们的结论是，DL模型的输出可以通过在体系结构中编码一个合理的归纳偏差来实现。表1确认了这不会导致性能下降。

****

后注：可以想象成，我输入一个单变量时间序列，经过基本块后，得到前向传播（y）以及backcast的x 这两个的基的系数。然后将基加权求和。（两个配置里面，一个的基是可学习的，另一个的基是指定的【解释性配置】）