

基于 SAE 与 CEEMDAN-BiLSTM 组合模型的短期电力负荷预测

黄 炜 陈 田*

(上海电机学院机械学院 上海 200120)

摘要 单一模型在迭代训练过程中由于模型的自身误差,最终会降低预测精度。为了提高预测的准确性,引入完整集成经验模态分解-双向长短期记忆网络(CEEMDAN-BiLSTM)作为误差修正模型,提出一种栈式自编码器(SAE)与 CEEMDAN-BiLSTM 相结合的负荷预测模型。通过 SAE 模型学习气象因素、工作日类型、气温影响下负荷序列的主要特征,预测过程中产生的误差序列则反映了负荷序列的次要特征;使用 CEEMDAN 算法将误差序列分解为数个分量,针对每一项分量建立 BiLSTM 模型学习误差序列的时序特征,将各项分量的预测值累加得到误差的预测结果;将两种模型的预测值求和从而达到修正误差的目的。通过比较几种模型的预测结果表明:SAE 与 CEEMDAN-BiLSTM 组合模型应用在短期电力负荷预测具有更好的准确性与稳定性。

关键词 短期电力系统负荷预测 栈式自编码器 CEEMDAN 双向长短期记忆网络

中图分类号 TP3 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2022.07.009

SHORT-TERM POWER LOAD FORECASTING BASED ON THE COMBINATION MODEL OF SAE AND CEEMDAN-BILSTM

Huang Wei Chen Tian*

(College of Mechanics, Shanghai Dianji University, Shanghai 200120, China)

Abstract In the process of iterative training, the prediction accuracy of a single model will be reduced due to its own errors. In order to improve the accuracy of the prediction, this paper introduced the integrated empirical mode decomposition-bidirectional long short term memory network (CEEMDAN-BiLSTM) as the error correction model, and proposed a load prediction model combining the stack self-encoder (SAE) and CEEMDAN-BiLSTM. The main characteristics of the load sequence under the influence of meteorological factors, working days and air temperature were learned through the SAE model. The error sequences generated in the prediction process reflected the secondary characteristics of the load sequence. CEEMDAN algorithm was used to decompose the error sequence into several components. For each component, the BiLSTM model was established to learn the time sequence characteristics of the error sequence. The predicted values of the two models were summed to correct the errors. By comparing the prediction results of several models, it is shown that the combination model of SAE and CEEMDAN-BiLSTM has better accuracy and stability in the short-term power load prediction.

Keywords Short-term power system load forecasting Stacked autoencoder CEEMDAN Bidirectional long short term memory network

0 引言

近年来,随着我国科技工业的高速发展,电能的需求量不断增大,对于电力负荷预测准确度的要求也在

不断地提高。按照时间跨度的不同,负荷预测主要分为超短期、短期、中期和长期,其中短期负荷预测主要对未来 24 小时或者数天时间内的用电负荷进行预测^[1],它对于发电机的投切、机组的检修安排、电力营销等有着重要意义,精准的负荷预测有利于发电设备

的充分利用,能有效降低电网的运营成本^[2]。

国内外对于负荷预测算法的研究工作有许多,其中具有代表性的算法主要包括:支持向量^[3]、随机森林^[4]、卡尔曼滤波法^[5]与人工神经网络^[6]等。文献[7]使用粒子群算法对反向神经网络进行初始化,并且引入了遗传算法,优化网络初始权值,从而提高了模型的性能。文献[8]将深度信念网络模型(DBN)与支持向量机(SVR)相结合,使得模型具有更高的准确性。文献[9-10]探讨了BP神经网络在负荷预测中的应用,文献[11]使用了模糊神经网络进行负荷预测。文献[12]采用了EMD算法对负荷数据进行分解,然后对于分解得到的各个平稳分量建立预测模型,最终在更加平稳的分量中取得了更高的精度。文献[13]和文献[14]分别通过改进的果蝇算法和粒子群算法优化BP神经网络权值和阈值,从而提高预测精度。但是传统的神经网络没有考虑到负荷在时间序列上的关系,所以在一定程度上不能很好地拟合时间序列上的负荷曲线。

长短期记忆网络(Long Short-term Memory, LSTM)同时兼顾非线性与时序性,被广泛应用于语音识别、机器翻译等领域^[15]。基于电力负荷在时间序列上的特性,文献[16]提出用LSTM神经网络进行负荷预测,最后经实验表明LSTM的预测精度比其他模型的精度高。文献[17]使用CNN对负荷数据进行降维,然后将降维后的数据作为LSTM的输入,这在一定程度上提高了LSTM的预测精度。

鉴于上述模型都无法对预测结果进行误差修正,本文提出SAE与CEEMDAN-BiLSTM组合的短期电力负荷预测模型,算法首先使用SAE模型学习负荷序列的主要特征,误差特征主要体现在SAE模型预测过程中所产生的误差序列中;然后使用CEEMDAN算法将误差序列分解为数个IMF分量与残余分量,建立BiLSTM模型学习每个误差分量的时序特征,将各个分量的BiLSTM模型的预测值相加得到误差的预测值;最后使两部分的预测值相加从而达到修正误差的效果。实验结果表明:SAE与CEEMDAN-BiLSTM的组合模型具有更好的预测精度,能够为电力系统的调度和生产计划提供更加有效的科学依据。

1 SAE与CEEMDAN-BiLSTM的组合模型

1.1 栈式自编码器

自编码器(autoencoder, AE)作为栈式自动编码器

(Stacked AutoEncoder, SAE)的基本组成部分,其结构类似于三层神经网络,是一种无监督学习,主要由输入的编码器与输出的解码器构成,一般用于数据的降维与去噪,功能类似于主成分分析PCA。如果使用适当的维度和稀疏性约束,会得到比PCA更好的数据投影效果。自编码器的输入节点等于输出节点,通过一个恒等的函数,可以对数据进行重构。

对于输入的编码器,其计算公式如下:

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b}_1) \quad (1)$$

式中: \mathbf{y} 表示编码器的输出; \mathbf{x} 表示解码器的输入; f 表示激活函数,一般为Sigmoid函数; \mathbf{W} 和 \mathbf{b}_1 分别表示输入层到隐藏层的权重与偏置。

对于输出的解码器,其计算公式如下:

$$\mathbf{z} = f(\mathbf{W}^T\mathbf{y} + \mathbf{b}_2) \quad (2)$$

式中: \mathbf{z} 表示解码器的输出; \mathbf{W}^T 和 \mathbf{b}_2 表示输入层到隐藏层的权重与偏置。这样通过调整网络的参数,可以使最终输出的 \mathbf{z} 尽可能地接近于自编码器输入的 \mathbf{x} ,从而达到重构输入数据的目的。

自编码器通过反向传播的方式使重构的误差最小,与传统神经网络一样,自编码器也是通过最小化损失函数来调整网络结构中的参数,损失函数的公式如下:

$$C_{sp}(\mathbf{W}, \rho) = \frac{1}{2N} \sum_{i=1}^N \sum_{j=1}^d (r(x_{i,j} - x_{i,j}))^2 + \frac{\lambda}{2} \sum_i \sum_j (\omega_{i,j}(l))^2 + \gamma \sum_{k=1}^n KL(\rho \parallel \hat{\rho}_k) \quad (3)$$

式中: C_{sp} 表示损失函数; d 表示编码器输入节点个数; n 为隐藏节点个数; λ 为权重衰减参数; $KL(\rho \parallel \hat{\rho}_k)$ 为稀疏项惩罚因子; γ 为稀疏项权重; ρ 为稀疏性参数。

栈式自动编码器由多个自编码器堆叠而成,其训练过程一般分为无监督预训练阶段和有监督训练阶段。在无监督预训练阶段,它可以逐层提取数据的高阶特征,逐步降低数据的维度,将复杂的数据转化为简单的特征序列。在有监督训练阶段,栈式自动编码器将无监督训练过程中得到的特征序列输入到一个分离器中进行分类。SAE的训练过程主要分为以下三步,如图1所示。

(1) 首先使用自编码器训练从输入层 X 到隐藏层 H 之间的参数,训练结束后,去除自编码器的输出层 x ,接着将AE1的隐藏层 H 的输出作为AE2的输入,采用相同的办法训练自编码器AE2。

(2) 重复步骤(1),直到初始化所有的自编码器参数。

(3) 将最后一个自编码器隐藏层的输出连接到分离器中,以有监督的方式进行训练。

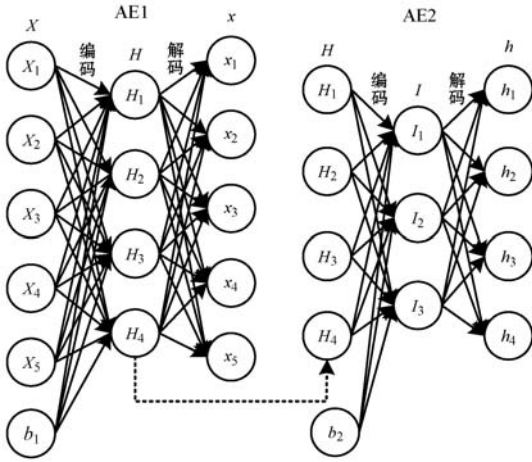


图1 SAE 的训练过程

1.2 CEEMDAN 原理

CEEMDAN 在 EEMD 分解的过程中添加自适应的白噪声并且计算唯一的余量信号以获取 IMF 分量,可以在较少的实验次数中对信号进行序列重构,消除了 EEMD 分解过程产生的残余信号中由于添加自适应白噪声所产生的误差。

定义计算符 $E_i(\cdot)$ 表示使用 EMD 分解产生的第 i 个 IMF 分量,CEEMDAN 算法的计算步骤如下。

(1) 在 k 次实验中,对原始信号 $x_i + \delta_0 \omega_j$ 进行分解,其中: δ_0 表示高斯白噪声的标准差; ω_j 表示高斯白噪声。通过 EMD 分解得到第一个 IMF 分量并获取唯一的余量信号 $r_1(t)$ 。

$$IMF_1(t) = \frac{1}{k} \sum_{i=1}^k IMF_1^i \quad (4)$$

$$r_1(t) = x(t) - IMF_1(t) \quad (5)$$

(2) 继续获取第二个 IMF 分量:

$$IMF_2(t) = \frac{1}{k} \sum_{i=1}^k E_1(r_1(t) + \delta_0 E_1(\omega_j(t))) \quad (6)$$

(3) 重复以上步骤,计算第 n 个余量信号:

$$r_n(t) = r_{n-1}(t) - IMF_n(t) \quad (7)$$

则第 $n+1$ 个 IMF 分量为:

$$IMF_{n+1}(t) = \frac{1}{k} \sum_{i=1}^k E_1(r_n(t) + \delta_n E_n(\omega_j(t))) \quad (8)$$

(4) 重复步骤(3),直到余量信号呈单调趋势,分解停止原始信号 $x(t)$ 被分解为:

$$x(t) = \sum_{i=1}^N IMF_i + r(t) \quad (9)$$

式中: N 为最终模态分量的个数; $r(t)$ 为最终单调的余量信号。

1.3 LSTM 循环网络

长短期记忆网络从 RNN 网络结构上改善了网络,避免了 RNN 神经网络在循环传播的过程中对于较远节点的计算会涉及到雅可比矩阵的累乘,从而避免了

梯度膨胀或者梯度弥散的现象。其结构如图 2 所示, LSTM 神经网络将每个记忆单元交由 3 个控制门控制, 3 个控制门分别是输入门 i_t 、遗忘门 f_t 、输出门 o_t 。网络在 t 时刻的输入为 x_t , 输出为 h_t , 候选状态为 \tilde{c}_t , 记忆状态为 c_t 。

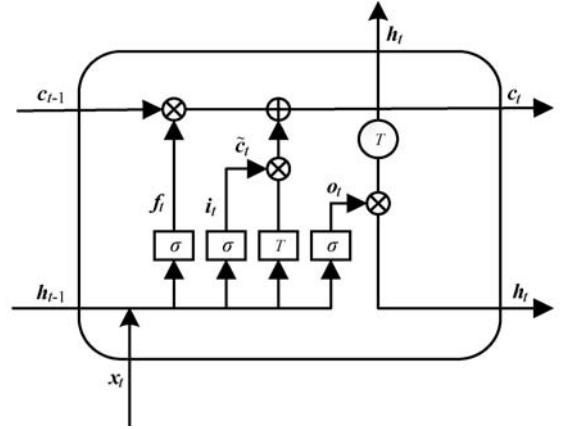


图2 LSTM 结构

LSTM 在每个隐藏层内都接收上一时刻的输出和当前时刻的输入及当前的隐藏层状态,并且通过输入门、遗忘门、输出门来控制 and 更新当前隐藏层的状态,最后将更新结果输出。计算公式如式(10) - 式(15)所示。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (10)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (11)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (12)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (13)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (14)$$

$$h_t = o_t * \tanh(c_t) \quad (15)$$

式中: W_f, W_i, W_c 和 W_o 分别表示遗忘门、输出门、记忆状态和输出门的权重矩阵; b_f, b_i, b_c 和 b_o 分别表示对应的偏置向量; σ 为激活函数 Sigmoid; $*$ 表示点乘。

1.4 BiLSTM 算法

首先使用 EMD 算法将负荷数据分解为数个本征模态函数(IMF)与残余分量 $r(t)$, 然后对于每个分量建立 BiLSTM 模型。BiLSTM 是由前向 LSTM 网络与后向 LSTM 网络构成,其结构如图 3 所示。

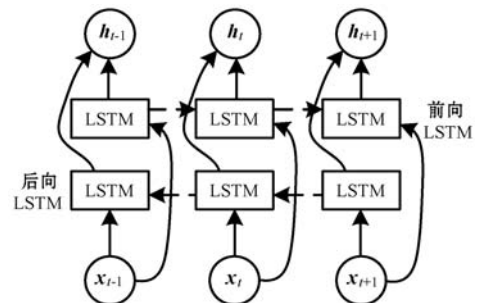


图3 BiLSTM 结构

BiLSTM 神经网络会基于整个时间序列作出预测, 它将隐藏层划分为前向与后向两个对立的部分, 分别

读取过去与未来时刻的信息。第一层前向 LSTM 计算当前时刻序列的顺序信息,第二层后向 LSTM 计算相同时刻序列的逆序信息,最后 BiLSTM 在 t 时刻的隐藏层状态由前向隐藏层状态和后向隐藏层状态加权得到。其计算公式如下:

$$\vec{h}_t = \text{LSTM}(x_t, \vec{h}_{t-1}) \quad (16)$$

$$\overleftarrow{h}_t = \text{LSTM}(x_t, \overleftarrow{h}_{t-1}) \quad (17)$$

$$h_t = W_t \vec{h}_t + V_t \overleftarrow{h}_t + b_t \quad (18)$$

式中: \vec{h}_t 表示 t 时刻前向隐藏层状态; \overleftarrow{h}_t 表示 t 时刻后向隐藏层状态; W_t, V_t 表示 \vec{h}_t 与 \overleftarrow{h}_t 所对应的权重; b_t 表示 t 时刻的隐藏层的偏置。

2 实验数据

现有美国某电力市场 2015 年的样本数据集,数据集的采样周期为 30 min。实验所用特征包括:负荷值、气象因素、节假日类型和气温。实验数据如表 1 所示。

表 1 实验数据

参数	含义
X_1	当前时刻负荷值
$X_2 - X_4$	风速、降雨、湿度
X_5	节假日类型
$X_6 - X_8$	平均气温、最高气温、最低气温

部分数据用虚拟编码模式表示,以 $\{0,1,2\}$ 表示 $\{\text{工作日,休息日,节假日}\}$ 。降雨 $\{有,无\}$ 表示为 $\{0,1\}$ 。

实验采用全年数据集 6 月至 8 月共计 92 天的数据作为训练集,以 9 月 1 日至 9 月 11 日的的数据共计 11 天的数据作为测试集,以历史 48 个时刻点的数据预测未来 48 个时刻点的负荷数据。

将 93 天的训练集转化为的三维矩阵,三维矩阵的维度信息为:

$$\{D_1 D_2 \cdots D_{92}\} \times \{L_1 L_2 \cdots L_{48}\} \times \{X_1 X_2 \cdots X_8\} \quad (19)$$

式中: $\{D_1 D_2 \cdots D_{92}\}$ 表示数据集长度为 92 天; $\{L_1 L_2 \cdots L_{48}\}$ 表示每天所包含的 48 个时刻的数据; $\{X_1 X_2 \cdots X_8\}$ 表示数据集的 8 维特征。

同理,测试集的维度信息为:

$$\{D_1 D_2 \cdots D_{11}\} \times \{L_1 L_2 \cdots L_{48}\} \times \{X_1 X_2 \cdots X_8\} \quad (20)$$

同时 BiLSTM 与其他机器学习预测方法一样,对数据的尺度比较敏感,所以这里采用 MaxMin 的方法对数据进行归一化处理,将所有数据的值域归一化到 $[0,1]$ 范围内,然后将归一化的数据传入到模型中,最后将模型的输出结果进行反归一化,计算的公式如下:

$$x'_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (21)$$

式中: x'_i 与 x_i 分别表示归一化后的值、数据值; x_{\max} 与 x_{\min} 表示最大值与最小值。

3 算例分析

3.1 SAE-CEEMDAN-BiLSTM 模型计算流程

(1) 负荷预测。使用包含气象因素、工作日类型的训练集对 SAE 模型进行有监督训练,待模型收敛后对训练集与测试集进行预测,得到测试集的预测值 P_b ,并且根据下式得到 SAE 训练过程中产生的误差序列 P_e :

$$P_e = P_{\text{btrain}} - T_{\text{train}} \quad (22)$$

式中: P_{btrain} 为训练集的预测值; T_{train} 为训练集的真实值。

(2) 误差预测。使用 CEEMDAN 算法将 P_e 分解为数个 IMF 分量与一个残余分量,并且针对每项分量分别建立 BiLSTM 模型进行学习与预测,最后对预测结果进行累加得到误差的预测值 P'_e 。

(3) 计算组合模型的预测值。SAE-CEEMDAN-BiLSTM 模型的预测结果 P 为测试集的预测值与误差的预测值之和,即:

$$P = P_b + P'_e \quad (23)$$

SAE 与 CEEMDAN-BiLSTM 组合模型的预测流程如图 4 所示。

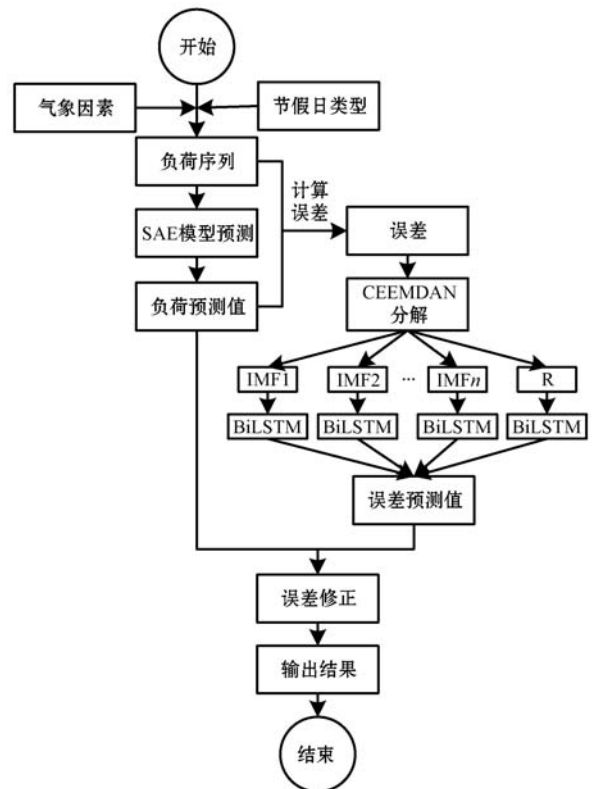


图 4 预测流程

3.2 模型训练方法

模型的训练过程采用 Adam 优化算法。Adam 结合了 RMSprop 善于处理非平稳目标与 Adagrad 善于处理稀疏梯度的优点,其计算公式如下:

$$m_t = \mu * m_{t-1} + (1 - \mu) * g_t \quad (24)$$

$$n_t = v * n_{t-1} + (1 - v) * g_t^2 \quad (25)$$

$$\hat{m}_t = \frac{m_t}{1 - \mu^t} \quad (26)$$

$$\hat{n}_t = \frac{n_t}{1 - v^t} \quad (27)$$

$$\Delta\theta_t = -\frac{\hat{m}_t}{\sqrt{\hat{n}_t + \epsilon}} * \eta \quad (28)$$

式中: g_t 表示梯度; ϵ 为平滑指数,主要用于防止分母为零; μ 与 v 表示动量因子; m_t 与 n_t 是对梯度的一阶矩阵估计,可以看作是对期望 $E|g_t|$ 和 $E|g_t^2|$ 的估计; \hat{m}_t 与 \hat{n}_t 是对 m_t 与 n_t 的修正,这样就可以对期望进行近似的无偏估计。

3.3 模型的评价指标

模型的评价指标通常采用平均绝对百分误差 (Mean Absolute Percentage Error, MAPE),其计算的公式如下:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{x_i} \right| \times 100\% \quad (29)$$

式中: \hat{x}_i 为神经网络在 i 时刻的预测值; x_i 为神经网络在 i 时刻的实际值。

3.4 SAE 模型训练与预测

实验在 Ubuntu 18.04 系统下利用 Pytorch1.10 完成,实验 PC 的处理器为 Inter Core i5-8300H,内存为 16 GB,显卡为 NVIDIA GeForce GTX 1050TI。

在无监督训练阶段,将 SAE 中的自编码器基本单元个数设置为 3,每个自编码器输入节点个数为 48,隐藏节点为 16,激活函数设置为 ReLU,设定学习率 $\eta = 0.1$,最大迭代次数 $n = 40\ 000$ 。在有监督训练阶段,将分离器的输出节点设置为 48,以梯度下降的方法对网络参数进行优化。

SAE 模型收敛后,将训练集中的数据按照时间先后顺序依次输入至训练完毕的模型中,得到训练集的预测结果 P_{train} ,如图 5 所示。同时根据式(22)可以求得 SAE 模型在预测过程中产生的误差序列 P_e ,如图 6 所示。

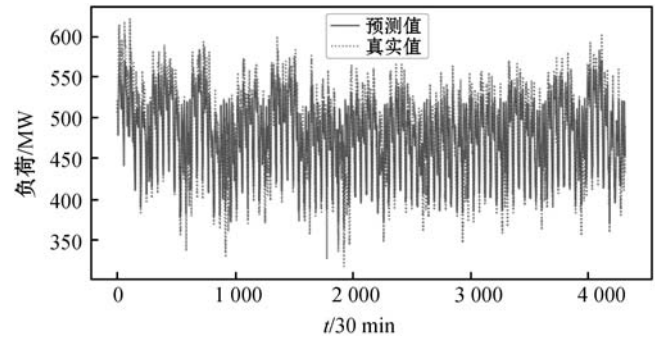


图5 SAE模型的预测结果

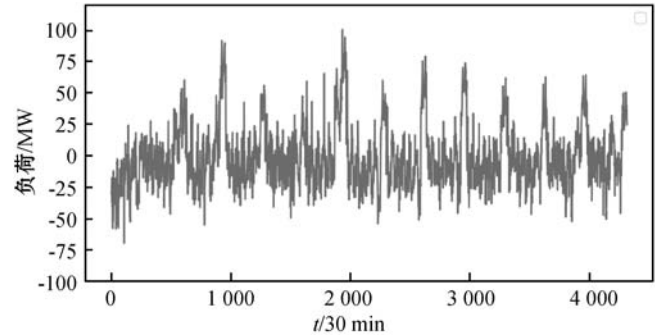


图6 误差序列

3.5 CEEMDAN 误差序列分解

设定 CEEMDAN 算法的总体集成次数为 400,加入白噪声标准差为 0.2,利用 CEEMDAN 算法将 P_e 分为 9 个 IMF 分量与 1 个残余分量,实验仿真结果如图 7 所示。

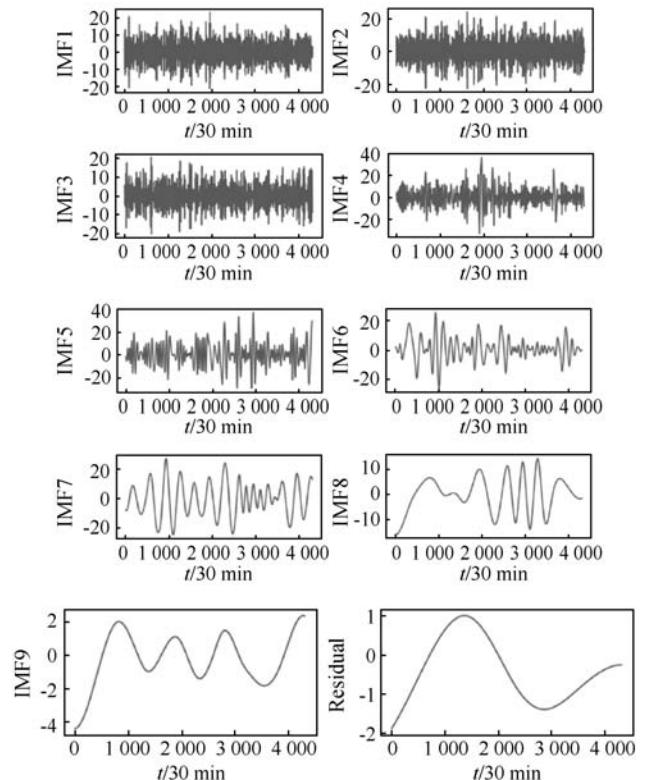


图7 CEEMDAN分解结果

可以看出,IMF1 - IMF4 分量的频率较高且周期性不明显,可以看作负荷序列的高频分量。IMF5 - IMF7

分量的周期性明显,可以看作负荷序列的周期分量。IMF8、IMF9 分量的频率较低且周期性不明显,可以看作负荷序列的低频分量。Residual 为序列的残余分量。

3.6 BiLSTM 网络结构设计

将 CEEMDAN 算法分解得到的 9 个 IMF 分量与 1 个残余分量分别建立 BiLSTM 模型,将前向 LSTM 与后向 LSTM 输入节点设置为 1,表示每次读取一个时刻点的负荷数据,将隐藏层节点设置为 12,分别为当前时间点前后 12 个时刻的误差信息。将输出节点设置为 1,为第 13 个时间点的误差预测值。设置学习率为 0.001,同时为 BiLSTM 网络的输出层再加上一层线性全连接层,激活函数设置为 tanh。

设定 BiLSTM 模型的训练方式为:通过历史 48 个时刻点的误差序列预测未来 48 个时刻点的误差序列,因此以 6 月 1 日至 8 月 30 日的误差序列作为网络的输入,以 6 月 2 日至 8 月 31 日的误差序列作为网络的输出,以 Adam 算法训练网络。BiLSTM 网络参数如图 8 所示,以 BiLSTM 层的输入层为例, $\{91, 48, 1\} = \{ \text{输入数据集长度为 91 天,每日包含的 48 个时刻点,当前时刻的误差值} \}$ 。

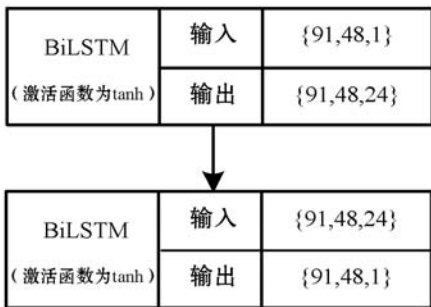


图 8 BiLSTM 网络结构

3.7 预测分量重构

将各个 IMF 分量 $c_i(t)$ 以及残差分量 $r_n(t)$ 的预测值进行累加得到误差的预测值,如式(30)所示。

$$p(t) = \sum_{i=1}^n c_i(t) + r_n(t) \quad (30)$$

式中: $p(t)$ 为重构后的误差数据。

3.8 实验过程及结果分析

实验首先通过 9 月 2 日的 48 个时刻点的数据预测 9 月 3 日 48 个时刻点的负荷数据。

待 SAE 模型与 CEEMDAN-BiLSTM 模型收敛后,通过 SAE 模型预测得到 9 月 3 日负荷序列的主要特征 P_0 ,然后通过 CEEMDAN-BiLSTM 模型预测得到 9 月 3 日的误差特征 P'_0 ,最后根据式(23)求得最终的预测结果。同时使用单独的 SAE 模型与 SAE-BiLSTM 模型作为对比实验。三种模型的预测曲线如图 9 所示。

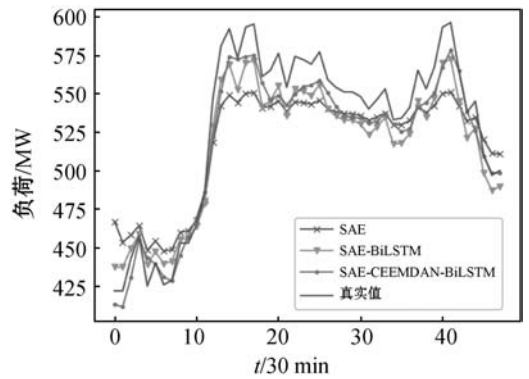


图 9 9 月 3 日预测曲线

为了进一步验证本文组合模型的有效性,以同样的方法对测试集中其余日期的数据进行预测,日均误差曲线如图 10 所示。

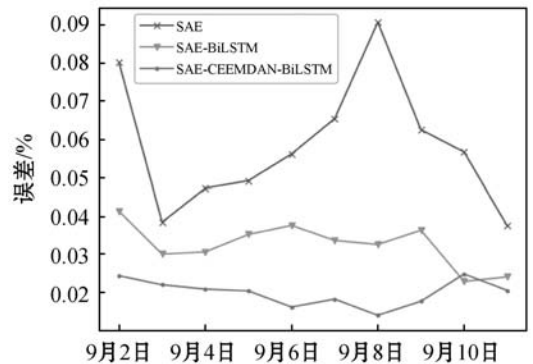


图 10 三种模型日误差对比

图 10 中 9 月 4 日至 5 日为双休日,9 月 7 日为美国劳动节。由图 9 与图 10 可以看出,SAE-CEEMDAN-BiLSTM 模型对负荷序列的拟合程度最高,具有很好的预测精度,即使在双休日、劳动节期间该模型的误差率也是最低;SAE-BiLSTM 模型以 BiLSTM 作为误差修正模型也取得了不错的预测精度,但整体上还是比 SAE-CEEMDAN-BiLSTM 模型的预测精度低;而 SAE 模型由于没有进行误差修正,所以在 3 种模型中表现最差。取平均绝对百分误差 (MAPE)、最大绝对百分误差 (Max) 和最小绝对百分误差 (Min) 三种指标进行比较,结果如表 2 所示。

表 2 性能对比 (%)

模型	MAPE	Max	Min
SAE-CEEMDAN-BiLSTM	2.09	2.63	1.54
SAE-BiLSTM	3.48	4.19	2.51
SAE	6.08	9.06	3.72

综合上述实验结果,可以得出以下结论:

(1) 由图 9、图 10 可知 SAE 模型由于没有进行误差修正,所预测精度相对较低,在测试集中平均准确率为 93.92%。SAE-BiLSTM 模型使用 BiLSTM 作为误差修正模型,所以预测精度高于 SAE 模型,测试集中平

均准确率为 96.52%。而 SAE-CEEMDAN-BiLSTM 模型所使用的误差修正模型可以在不同时间尺度上计算误差序列,提高了预测精度,所以日均准确率在 3 种模型中最优,测试集中平均准确率达到 97.91%。

(2) 由表 2 可知,SAE-CEEMDAN-BiLSTM 模型的三种评价指标均优于其余两种模型,说明该组合模型的预测效果最好;两种组合模型的三种评价指标均优于传统的 SAE 模型,说明引入误差修正模型能提高模型的预测精度。

(3) 由图 10 可知,SAE-CEEMDAN-BiLSTM 模型并不是在每一日的预测精度都达到最优,比如在 9 月 10 日其预测精度低于 SAE-BiLSTM 模型,但从总体上看,SAE-CEEMDAN-BiLSTM 模型相比其他两种模型具有更好的准确性与稳定性。

4 结 语

本文分析了 SAE 模型以及 CEEMDAN-BiLSTM 模型的原理,将两种模型进行组合,提出了 SAE 与 CEEMDAN-BiLSTM 的组合模型,并且将这种组合模型应用到电力系统负荷预测中。经过多次实验以及反复验证,通过对比 SAE-BiLSTM 模型以及单一的 SAE 模型可以得出结论:SAE 与 CEEMDAN-BiLSTM 的组合模型应用在电力系统负荷预测具有更好的准确性与稳定性。但随着预测步长的增加,模型的预测精度可能会随之降低,下一步将考虑更多影响负荷序列的因素以增加数据集的特征维度。

参 考 文 献

[1] 陈剑强,杨俊杰,楼志斌. 基于 XGBoost 算法的新型短期负荷预测模型研究[J]. 电测与仪表,2019,56(21):23-29.

[2] 徐晴,周超,赵双双,等. 基于机器学习的短期电力负荷预测方法研究[J]. 电测与仪表,2019,56(23):70-75.

[3] 张冠英,姜一鸣,葛磊蛟,等. 经济新常态下基于 Verhulst-SVM 的中长期负荷预测模型[J]. 电测与仪表,2019,56(1):102-107.

[4] 吴潇雨,和敬涵,张沛,等. 基于灰色投影改进随机森林算法的电力系统短期负荷预测[J]. 电力系统自动化,2015,39(12):50-55.

[5] 潘志远,韩学山. 电网节点负荷的立体化预测方法[J]. 电力系统自动化,2012,36(21):47-52.

[6] 程宇也. 基于人工神经网络的短期电力负荷预测研究[D]. 杭州:浙江大学,2017.

[7] 唐鉴,从杨,高宏伟. 电力系统供电负荷短期预测模型设计[J]. 计算机仿真,2016,33(2):171-174,179.

[8] 孙海蓉,田瑶,谢碧霞. 基于深度学习的短时热网负荷预

测[J]. 计算机仿真,2019,36(2):79-82,301.

[9] 赵宇红,苏泽光,盛义发,等. BP 神经网络在电力系统短期负荷预测中的应用[J]. 南华大学学报(自然科学版),2005,19(3):71-73.

[10] 赵娟娟. 基于 BP 神经网络的电力系统短期负荷预测[D]. 哈尔滨:哈尔滨理工大学,2013.

[11] 喻圣,邹红波,余凡,等. 模糊神经网络在电力系统短期负荷预测中的应用[J]. 智慧电力,2018,46(11):88-91,97.

[12] 李琦,赵峰. 基于气温与日期类型的改进 BP 网络热负荷预测[J]. 系统仿真学报,2018,30(4):1464-1472.

[13] 王亚琴,王耀力,王力波,等. 一种改进果蝇算法优化神经网络短期负荷预测模型[J]. 电测与仪表,2018,55(22):13-18,24.

[14] 孙景文,常鲜戎. 基于高斯混沌粒子群优化动态前馈神经网络的短期负荷预测[J]. 陕西电力,2015,43(9):67-69,75.

[15] Pichotta K, Mooney R J. Using Sentence-Level LSTM language models for script inference[C]//54th Annual Meeting of the Association for Computational Linguistics,2016.

[16] 张亚宁. 基于 LSTM 模型的短期负荷预测[J]. 科技与创新,2018(12):18-21.

[17] 陆继翔,张琪培,杨志宏,等. 基于 CNN-LSTM 混合神经网络模型的短期负荷预测方法[J]. 电力系统自动化,2019,43(8):131-137.

(上接第 31 页)

[7] 沈凤龙,满永奎,王建辉,等. 一种改进的三电平 PWM 整流器双闭环控制策略[J]. 电力电子技术,2019,53(6):83-86.

[8] 贾嵘,刘伟,徐其惠. 电压不平衡时 PWM 整流器自抗扰控制策略的研究[J]. 水力发电学报,2008,27(5):172-176.

[9] Ma Y, Zhao F, Zhou X, et al. DC side bus voltage control of wind power grid-connected inverter based on second-order linear active disturbance rejection control [J]. Energies, 2019, 12(22):4274.

[10] 高华鑫. 基于变结构自抗扰的车用 PMSM 模型预测控制[J]. 计算机应用与软件,2018,35(6):83-87,137.

[11] 杨阳. 一种双闭环控制 PWM 变流器同相供电系统[D]. 南昌:华东交通大学,2018.

[12] 易星,吴昊. 基于双种群混沌鲸鱼算法的自抗扰控制器参数整定[J]. 计算机应用与软件,2019,36(4):281-285,294.

[13] 刘英培,栗然,梁海平. 基于最小二乘支持向量机优化自抗扰控制器的永磁同步电机直接转矩控制方法[J]. 中国电机工程学报,2014,34(27):4654-4664.

[14] 于希宁,朱丽玲. 自抗扰控制器的动态参数整定及其应用[J]. 华北电力大学学报,2005,32(6):9-13.

[15] 袁东林,邵如平,许文钱,等. 基于新型自抗扰控制的 VIENNA 整流器控制策略[J]. 电力电子技术,2019,53(12):102-105.