

基于 Conv1d-LSTM 模型的能源分配预测^①



安鹤男², 姜邦彦¹, 管 聪¹, 马 超¹, 邓武才²

¹(深圳大学 微纳光电子学研究院, 深圳 518000)

²(深圳大学 电子与信息工程学院, 深圳 518060)

通信作者: 姜邦彦, E-mail: 423713382@qq.com

摘 要: 能源分配问题往往与其所在区域环境有关, 能源分配的预测可以通过当地环境因素数据来推测之后对该区域的能源分配数值, 最大程度上分配好能源. LSTM 网络预测短期效果良好, 但预测较长时期的数据会导致误差积累, 速度慢且准确性差; Informer 是近期新提出的能源预测算法模型, 速度快但在该任务上预测能力不够. 本文提出 Conv1d-LSTM 模型, 预测结果优于上述两个模型, 具有更低的平均绝对误差和均方根误差.

关键词: 能源分配; 卷积神经网络; 长短期记忆网络 (LSTM); 平均绝对误差; 均方根误差; 深度学习

引用格式: 安鹤男, 姜邦彦, 管聪, 马超, 邓武才. 基于 Conv1d-LSTM 模型的能源分配预测. 计算机系统应用, 2023, 32(1): 206-213. <http://www.c-s-a.org.cn/1003-3254/8937.html>

Energy Distribution Prediction Based on Conv1d-LSTM Model

AN He-Nan², JIANG Bang-Yan¹, GUAN Cong¹, MA Chao¹, DENG Wu-Cai²

¹(Institute of Microscale Optoelectronics, Shenzhen University, Shenzhen 518000, China)

²(College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China)

Abstract: Energy distribution is often related to the local environment. Regarding energy distribution prediction, data on local environmental factors can be availed to predict the value of energy to be distributed to the region, thereby maximizing the extent of proper energy distribution. The long short-term memory (LSTM) network, despite its favorable short-term prediction effect, is weakened by error accumulation, a slow speed, and poor accuracy when it is used for long-term data prediction. As a new algorithmic energy prediction model recently proposed, Informer is fast but not sufficiently capable of prediction in this task. This study proposes a Conv1d-LSTM model that achieves better prediction results than those of the above two models with a smaller mean absolute error (MAE) and root mean square error (RMSE).

Key words: energy distribution; convolutional neural network (CNN); long short-term memory (LSTM); mean absolute error (MAE); root mean square error (RMSE); deep learning

能源分配问题是地区发展需要面临的一个难题, 对于未来一段时间内地区的能源分配数值进行预测, 使得能源被充分利用, 既能保证该地区的发展, 还能减少对环境造成的污染, 所以对于所需能源的具体值进行预测是一项非常有必要的工作^[1-3]. 而能源分配往往需要考虑到该地区的气候环境因素, 同时还要将以往能源分配的数值作为参考值, 这是典型的时序预测问题.

对于传统的时序预测, 有包括卡尔曼滤波预测^[4]、时间序列预测^[5]和差分整合滑动平均自回归模型预测^[6]等方法, 而此类传统方法预测精度低^[7]且预测耗时很长. 利用神经网络预测能更快更好的解决上述问题, 循环神经网络 (recurrent neural network, RNN) 作为最早使用的算法结构容易出现梯度爆炸的问题, 而且其预测的结果基本仅取决于过去超短时期的数据. 后续为

① 收稿时间: 2022-05-29; 修改时间: 2022-06-29, 2022-07-29; 采用时间: 2022-08-15; csa 在线出版时间: 2022-11-16

CNKI 网络首发时间: 2022-11-18

了改进 RNN 遗忘性大这一问题, 引入了传统的长短期记忆网络 (long short-term memory, LSTM) 这一种时间循环神经网络, LSTM 的记忆门与遗忘门可以处理传入的短期数据, 在预测问题上可以减缓梯度爆炸或者梯度消失的问题; 虽然传统的 LSTM 模型在相比 RNN 时有了很大的进步, 但仍然存在 3 个问题: (1) 预测多元问题效果差. LSTM 模型在面对多元预测问题上, 往往都是将原始数据直接传入网络中生成对应的预测结果; (2) 统计窗口小. LSTM 模型进行大量的数据预测时, 很难将过去较长一段时期的数据纳入计算范围, 模型投入过多的数据不但浪费算力甚至会影响结果的准确性; (3) 时间复杂度高. LSTM 模型预测所花费的时间随所需预测的数量的增加而变长. 另一方面, Informer 模型^[8]常被用于预测能源分配问题, 该模型预测速度快且在中长距离预测和多元预测问题上都有较好的表现, 但在此任务上该模型效果不好.

本文针对上述问题, 提出了 Conv1d-LSTM 模型, 引入计算机视觉中的卷积核^[9,10]作为统计窗口, 提取出结合了多元的环境因素与能源分配的数值的特征; 设置较大的计算窗口可以让网络接收到更久远的信息; 采用并联多个 LSTM 组件的方法可进行快速地长序列预测. Conv1d-LSTM 模型面对多元数据集时, 相较于其他经典算法有着更高的精度. 本文将采用平均绝对误差 MAE (mean absolute error)、均方根误差 RMSE (root mean square error)、平均绝对百分比误差 MAPE (mean absolute percentage error) 和运行时长作为评价指标. 通过实验发现, Conv1d-LSTM 模型较其他模型具有更低的 MAE 和 RMSE, 在短距离预测 MAPE 仅为 4.4%, 同时在运行速度上也有着一定优势.

1 算法结构

1.1 输入与输出

1.1.1 输入

原始数据 $D_t = (d_0, d_1, d_2, \dots, d_{m-1})$, D_t 为 t 时刻该地能源分配具体数值 d_0 与当地环境因素相关数值 d_1-d_{m-1} . 算法输入 $X_t = (D_t, D_{t+1}, D_{t+2}, \dots, D_{t+k-1})$, X_t 为用于预测 t 时刻输出 Y_t 的矩阵, 维度为 (k, m) , k 为使用的原始数据 D_t 的数量.

1.1.2 输出

输出是未来 P 个连续时间预测的能源分配数值及环境因素数值, 可以表示为 $\hat{Y} = (Y_0, Y_1, Y_2, \dots, Y_{P-1})$, 其中

每个 Y_t 为 t 时刻模型的输出, 即 $Y_t = (y_0, y_1, y_2, \dots, y_{m-1})$, y_0 为预测的能源分配具体数值, y_1-y_{m-1} 为预测的环境因素数值.

1.2 算法结构

相较于传统的 LSTM 模型, Conv1d-LSTM 模型继承了 LSTM 内部的结构, 仍然保留了遗忘门、输入门、输出门等结构, 但输入序列不再从原始数据中直接获得, 而是将多组能源分配的数值以及该区域的环境因素数值传入卷积神经网络中.

如图 1 所示, 输入序列 X_t 通过多个卷积层和池化层, 最后经过全连接层提取出特征向量 L_t , 将生成的特征向量传入 LSTM 的相关组件中.

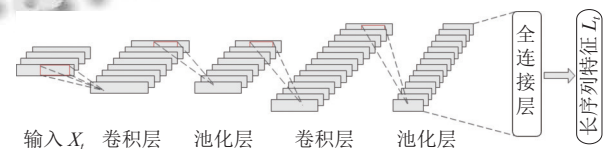


图 1 卷积神经网络 (CNN) 提取数据特征

卷积层 Conv1d 中输入为 $X_{in} (k, L_{in})$, 输出为 $X_{out} (n, L_{out})$, 其计算公式如下:

$$X_{out} = \frac{W_D \cdot D_0 + W_D \cdot D_1 + \dots + W_D \cdot D_{k-1} + b}{n} \quad (1)$$

$$L_{out} = \frac{L_{in} + 2 * p - (ks - 1)}{s} + 1 \quad (2)$$

其中, W_D 为卷积核的权值; D_0-D_{k-1} 为 k 个原始数据; b 为偏置量; L_{in} 为输入的矩阵的第二维度; n 为卷积核数量; p 为填充数量; ks 为卷积核的尺度; s 为卷积核每次移动的步长. X_{out} 为输出数值, L_{out} 为维度值.

卷积层计算方式如图 2, 假设输入为 $X_{in} (3, 6)$, 设 $p=1, ks=3, s=1, n=1$, 则通过卷积层后输出 X_{out} 维度为 $(1, 6)$.

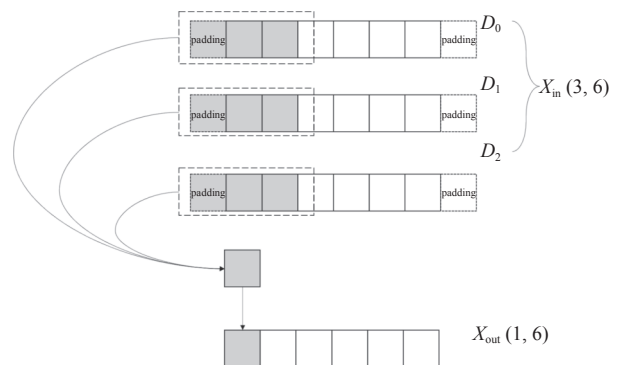


图 2 卷积层结构

平均池化层和激活函数 ReLU 维度计算公式与卷积层同理, 数值计算公式如下:

$$P_{out} = \frac{P'_{in} + P_{in} + P''_{in}}{3} \quad (3)$$

$$R_{out} = \max(R_{in}, 0) \quad (4)$$

其中, P'_{in} , P_{in} , P''_{in} 分别为输入的 3 个相邻的数值; R_{in} 为数据输入 ReLU 激活函数的数值. P_{out} 为平均池化层输出的数值, R_{out} 为数据通过 ReLU 激活函数输出数值.

将输入 X_t 传入通过多层卷积池化以及全连接层的卷积神经网络后获得长序列特征向量 L_t , 记为:

$$L_t = Conv1d(X_t) \quad (5)$$

其中, 计算结果 L_t 为 t 时刻前 k 次能量分配的原始数据通过卷积神经网络 (convolutional neural networks, CNN) 提取出的特征参数, k 为超参数, 用于设定每个特征包含多少组的数据.

LSTM 模型如图 3, 将每个时刻通过卷积神经网络生成的长序列特征 L_t 传入模型计算如下, 激活函数:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (7)$$

遗忘门 f_t , 输入参数为前一时刻的隐藏状态 H_{t-1} 及特征向量 L_t :

$$f_t = \sigma(W_f \cdot [H_{t-1}, L_t] + b_f) \quad (8)$$

输入门 i_t , 输入参数与遗忘门一样:

$$i_t = \sigma(W_i \cdot [H_{t-1}, L_t] + b_i) \quad (9)$$

当前细胞状态 C_t , 输入参数由遗忘门的输出 f_t 、输入门的输出 i_t 、上一时刻细胞状态值 C_{t-1} 及特征向量 L_t 组成:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_C \cdot [H_{t-1}, L_t] + b_C) \quad (10)$$

输出门 o_t , 输入参数为前一时刻隐藏状态 H_{t-1} 、当前细胞状态值 C_t 及特征向量 L_t :

$$o_t = \sigma(W_o \cdot [H_{t-1}, L_t] + b_o) \quad (11)$$

更新后的当前隐藏状态 H_t :

$$H_t = o_t \cdot \tanh(C_t) \quad (12)$$

输出预测出的长序列值 Y_t :

$$Y_t = o_t \cdot \tanh(C_t) \quad (13)$$

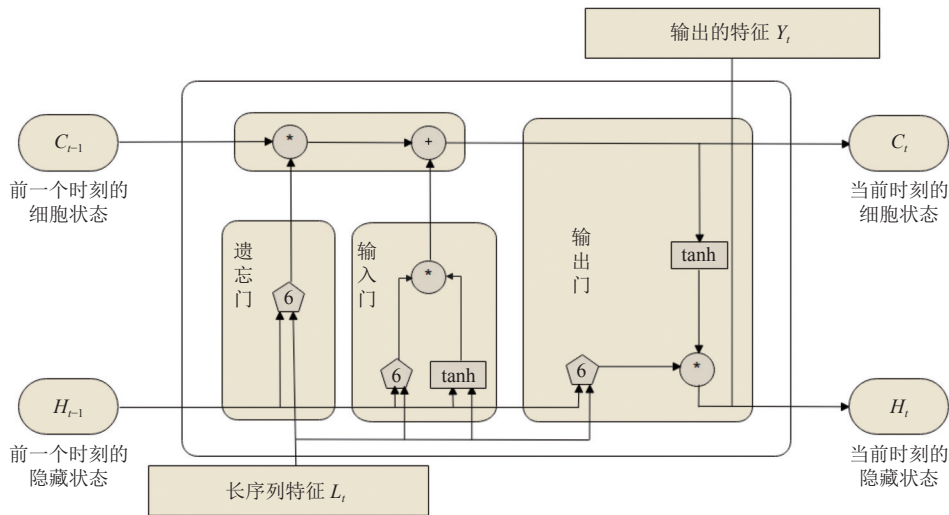


图 3 LSTM 模型

2 计算方法

2.1 单步计算

2.1.1 单步训练

模型单步训练流程如图 4, 假设输入 X_{t0} 由 10 个原始数据 $D_{t0}-D_{t9}$ 组成, 即 $X_{t0} = (D_{t0}, D_{t1}, D_{t2}, \dots, D_{t9})$, 再假设每条原数据 D_t 有 9 条数据, 即 $D_t = (d_0, d_1, d_2, \dots, d_8)$,

由此可知输入 X_{t0} 维度为 (10, 9). 通过卷积神经网络提取 X_{t0} 的特征生成特征向量 L_{t0} , 向量长度为 L , 该值由卷积神经网络最后的全连接层决定. 特征向量 L_{t0} 传入 LSTM 中得到预测值 Y_{t0} , 维度为 (10, 9). Conv1d-LSTM 模型使用的损失函数为 $MSELoss$, 公式如下:

$$MSELoss(p, t) = (p - t)^2 \quad (14)$$

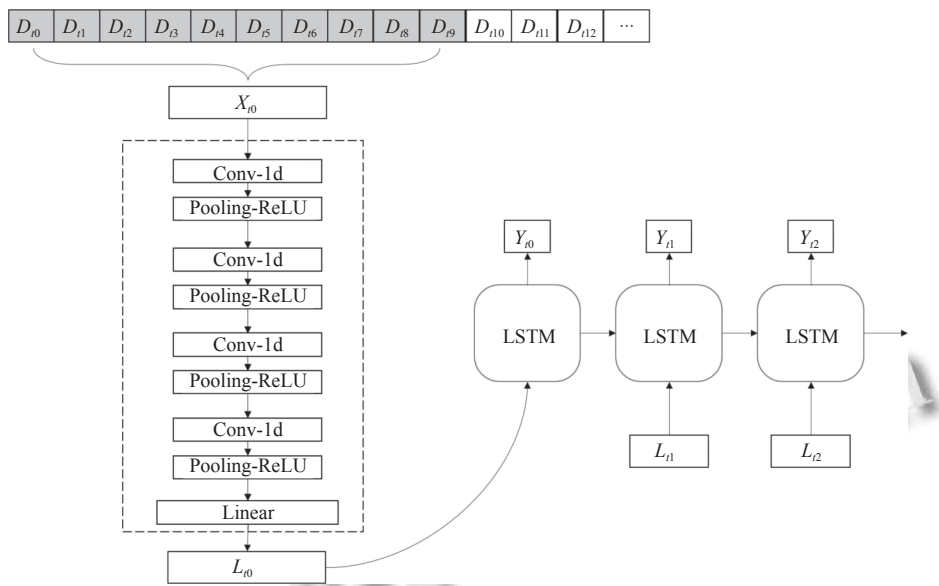


图4 Conv1d-LSTM 单步训练

预测值 Y_{t0} 与真实值 D_{t10} 通过 $MSELoss(Y_{t0}, D_{t10})$ 计算出损失值后通过反向传播训练模型, 后续如 Y_{t1} 与 D_{t11} 、 Y_{t2} 与 D_{t12} 同理, 直至模型收敛。

2.1.2 单步预测

模型单步预测流程如图5, 假设预测第1个未知

值 D_{t10} , 则将 $D_{t0}-D_{t9}$ 组成1个输入 X_{t0} 传入模型, 即可获得预测值 Y_{t0} 。若预测后续值如 D_{t11} , 则将 $D_{t1}-D_{t9}$ 和预测值 Y_{t0} 组成1个输入 X_{t1} 传入模型, 获得预测值 Y_{t1} 。按照此方法反复计算直到获得所有预测值 \hat{Y} 。

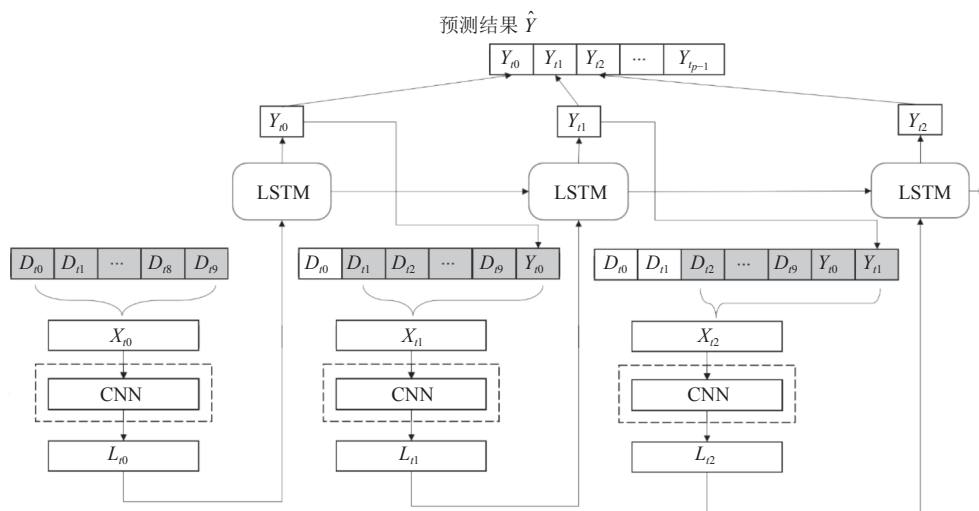


图5 Conv1d-LSTM 单步预测

2.2 多步计算

2.2.1 多步训练

由于单步训练的时间复杂度与 LSTM 一样, 长距离预测时会导致模型性能下降严重, 故设计出一组并联的 Conv1d-LSTM 结构来提升模型性能。模型多步训练

流程如图6, 假设一次性输入 X_{t0} 和 X_{t1} , 分别由3个原始数据 $D_{t0}-D_{t2}$ 和 $D_{t1}-D_{t3}$ 组成, 即 $X_{t0} = (D_{t0}, D_{t1}, D_{t2})$, $X_{t1} = (D_{t1}, D_{t2}, D_{t3})$ 再假设每条原数据 D_t 有9条数据, 即 $D_t = (d_0, d_1, d_2, \dots, d_8)$, 由此可知输入 X_{t0} 和 X_{t1} 维度均为 (3, 9)。通过卷积神经网络提取 X_{t0} 的特征生成特

征向量 L_{t_0} , X_{t_1} 的特征生成特征向量 L_{t_1} , 向量长度均为 L . 特征向量 L_{t_0} 、 L_{t_1} 传入 LSTM 中得到预测值 Y_{t_0} 、 Y_{t_1} , 维度均为 $(10, 9)$. 该模型仍然使用的损失函数 $MSELoss$, 但预测值 Y_{t_0} 所对应的真实值为 D_{t_4} , Y_{t_1} 对应 D_{t_5} , 故使用 $MSELoss(Y_{t_0}, D_{t_4})$ 和 $MSELoss(Y_{t_1}, D_{t_5})$ 计算出损失值, 最后通过反向传播训练模型, 后续如 Y_{t_2} 与 D_{t_6} 、 Y_{t_3} 与 D_{t_7} 同理, 直至模型收敛.

2.2.2 多步预测

模型多步预测流程如图 7, 假设预测两个未知值 D_{t_4} 、 D_{t_5} , 则将输入 $X_{t_0} = (D_{t_0}, D_{t_1}, D_{t_2})$ 和 $X_{t_1} = (D_{t_1}, D_{t_2}, D_{t_3})$ 同时传入模型, 即可获得预测值 Y_{t_0} 、 Y_{t_1} . 若预测后续值如 D_{t_6} 、 D_{t_7} , 则将 $X_{t_2} = (D_{t_2}, D_{t_3}, Y_{t_0})$ 和 $X_{t_3} = (D_{t_3}, Y_{t_0}, Y_{t_1})$ 同时传入模型, 获得预测值 Y_{t_2} 、 Y_{t_3} . 按照此方法反复计算直到获得所有预测值 \hat{Y} .

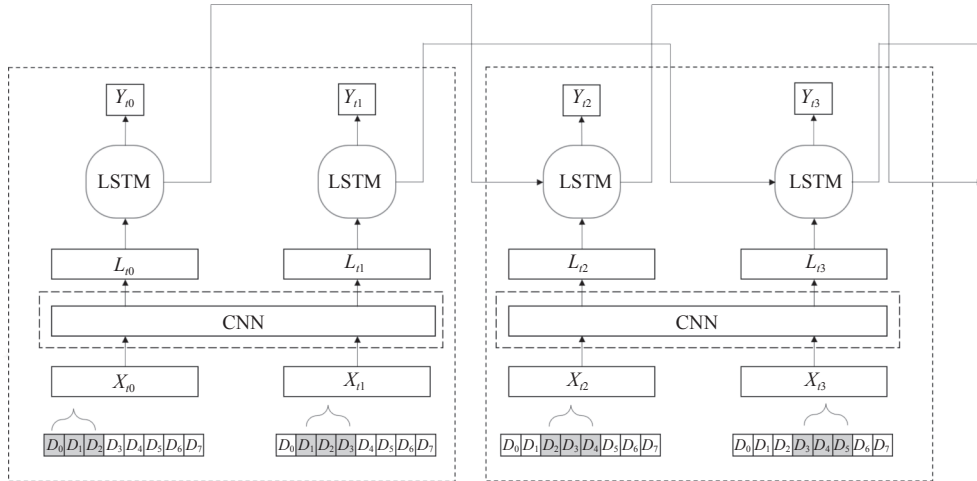


图 6 Conv1d-LSTM 多步训练

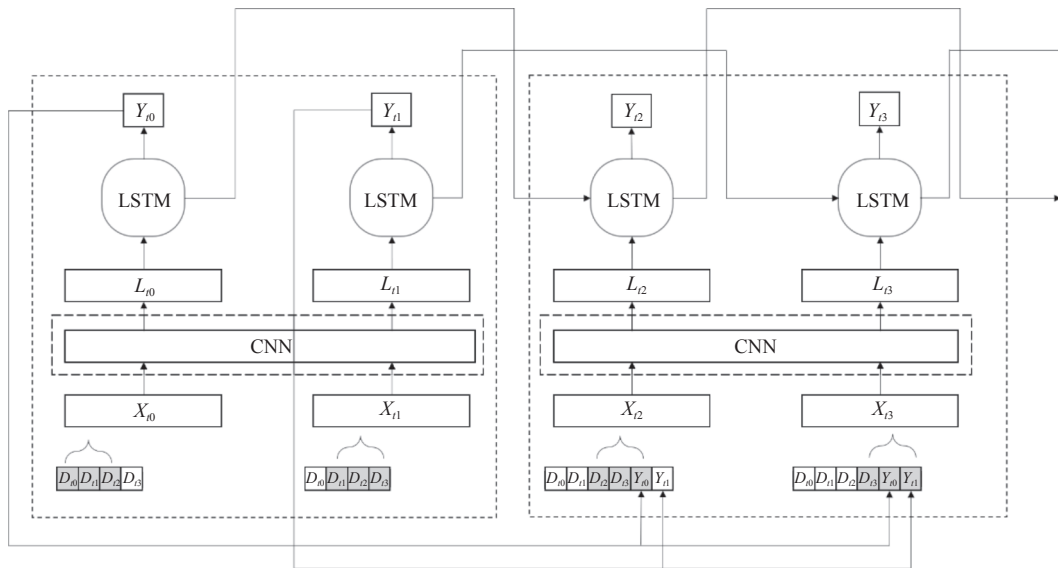


图 7 Conv1d-LSTM 多步预测

3 实验分析

3.1 实验数据集

为了评估 Conv1d-LSTM 模型的性能, 本文使用了从公开数据集 (rp5.ru) 下载的由比利时切维斯机场气

象站收集的当地相关数据. 该数据共 29 列, 其中 1 列时间、1 列能源分配数值和 27 列环境因素, 环境因素包括 1 列光照度、24 列从当地采集到的西北角至东南角的气压和湿度数值以及 2 列匿名环境因素. 其采样

时间间隔为 10 分钟,共 19736 行原始数据。

3.2 实验配置

实验服务器配置 CPU: Intel Xeon E5-2620 v4, GPU: GTX1080Ti×4 (显存 11 GB), 操作系统: CentOS Linux release 7.6.1810 (Core), 训练和测试均在单块显卡上进行,软件使用 Python 3.8、PyTorch 1.7.0、cuda 10.1。见表 1。

表 1 服务器配置和环境

配置	属性
CPU	Intel Xeon E5-2620 v4
GPU	GTX1080Ti×4
操作系统	CentOS Linux release 7.6.1810 (Core)
模型使用框架	PyTorch 1.7.0, Python 3.8, cuda 10.1

3.3 参数选择

3.3.1 普通参数

各个模型内部权值矩阵 W , 其内部数值分布是以均值为 0, 标准差为 1 的正态分布; 损失值 Loss, 由于 L2 损失较 L1 损失对异常值更为敏感且惩罚更重, 故本实验使用 $MSELoss$ 为损失函数。

3.3.2 超参数

统计窗口 $train_window$, 默认值为 92; LSTM 组件内部隐藏层大小 $hidden_size$, 默认值为 100; 优化器选择 Adam^[11], 学习率 $learning_rate$, 初始值为 0.001; 每批次数据量 $batch$, 默认值为 32; 迭代次数 $epoch$, 默认值为 100。

3.4 实验过程

训练模型时, 全部使用超参数的默认值, 一批次输入 124 个原始数据, 即统计窗口 92, 可截出 32 个连续的原始数据片段, 使得传入 CNN 的 $batch$ 数量为 32, 并获得相应的长序列特征。将获得的长序列特征传入 LSTM 结构中, 同时传出对应的预测值 \hat{y} 。通过预测值与真实值计算损失值 Loss, 使用反向传播^[12] 机制训练网络, 损失值 Loss 在模型迭代训练 100 次的变化如图 8 所示。

测试模型时, 以平均绝对误差 MAE (mean absolute error), 均方根误差 $RMSE$ (root mean square error) 和平均绝对百分比误差 $MAPE$ (mean absolute percentage error), 其具体表达式为:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (15)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (16)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right| \quad (17)$$

其中, n 为测试的样本数目; y_i 和 \hat{y}_i 分别为能量分配值预测值和真实值。由上述公式可知, 模型的 MAE 和 $RMSE$ 越低说明预测准确性越高, $MAPE$ 越接近 0 说明模型越适合该任务, 若超过 100% 则说明该模型为劣质模型。

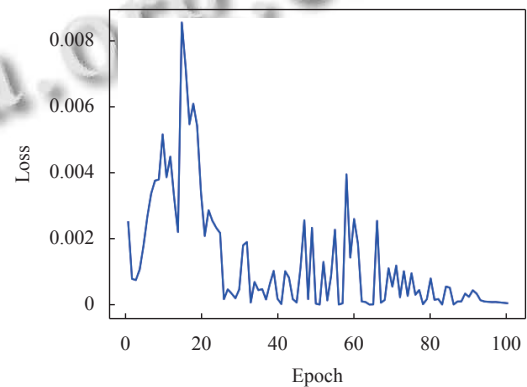


图 8 损失值 Loss 变化

3.5 实验结果

实验选择传统 LSTM 模型, Informer 模型与 Conv1d-LSTM 模型接收上述数据进行学习并预测后续能源分配的具体数值, 分别预测未来间隔为 24 次 (4 h)、96 次 (16 h)、192 次 (32 h) 以及 216 次 (36 h) 的能源分配数值, 比较其 MAE 、 MSE 、 $MAPE$ 以及运行时长, 见表 2。

表 2 不同模型的实验结果

模型	模型指标	间隔24次	间隔96次	间隔192次	间隔216次
LSTM	MAE	0.1146	0.2599	0.4112	0.4939
	$RMSE$	0.1520	0.4691	0.7353	0.8277
	$MAPE$ (%)	15.05	74.15	210.7	283.62
	运行时长 (s)	2.143	5.122	9.610	12.50
Informer	MAE	0.4030	0.4291	0.4407	0.4428
	$RMSE$	0.7503	0.7677	0.7687	0.8002
	$MAPE$ (%)	203.4	231.4	245.6	237.1
	运行时长 (s)	0.0942	0.0943	0.0917	0.0979
Conv1d-LSTM	MAE	0.0338	0.1235	0.2414	0.2939
	$RMSE$	0.0397	0.2206	0.2972	0.3521
	$MAPE$ (%)	4.4	23.09	69.98	82.23
	运行时长 (s)	0.2665	0.2833	0.2747	0.2944

如表 2 所示, 使用 Conv1d-LSTM 模型时, 间隔 24 次时的超短预测间隔平均绝对误差 (MAE) 仅为

0.033 8, 其预测效果远好于 Informer 模型, 较传统 LSTM 模型的效果也有小幅提升; 同样, Conv1d-LSTM 在中长预测间隔中, 如间隔 96 次、间隔 192 次、间隔 216 次时的 *MAE*、*RMSE* 均远低于 Informer 和 LSTM 模型, *MAPE* 也在始终保持在 100% 以下。

运行时长用于记录训练好的模型从加载到最后预测出结果的耗时。传统的 LSTM 模型面对越来越长的预测任务时, 运行时长也会随之变久, 而 Conv1d-LSTM 模型随着任务间隔变长所消耗的时间增多不明显, 相较于传统的 LSTM 模型有着速度上的绝对优势。

3.6 预测波形

为了直观体现 Conv1d-LSTM 模型在预测问题上的优势, 本文采用 Python 3.8 中的第三方库 matplotlib 绘制预测折线图, 数据由模型预测后直接导入该第三方库。图 9、图 10 展示了预测未来 24 次间隔的能量分配值, 图展示了未来 192 次间隔的能量分配值, 横坐标为间隔次数, 纵坐标为能源分配数值, 能源分配数值为原始数据缩放在 $(-1, 1)$ 区间内归一化后的数值。

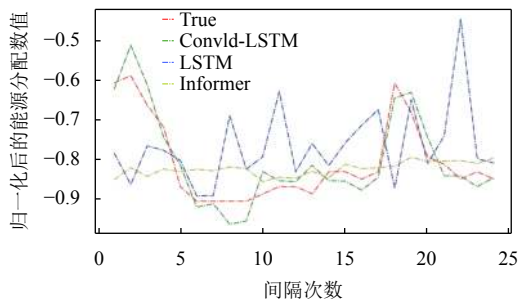


图 9 各模型 24 次间隔预测图

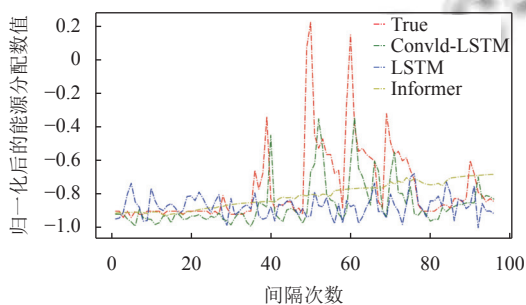


图 10 各模型 96 次间隔预测图

如图 9、图 10 可知, Informer 模型在预测此类能源分配任务时几乎没有波动, 从实验结果的 *MAPE* 远高于 100% 和展示的波形可以看出, Informer 模型不具备对该任务预测能力; LSTM 模型在 24 次间隔即短间

隔时预测能力较好, 但在 96 次中长预测间隔中性能严重下滑, 折线图中 LSTM 的波形只会 $(-1.0, -0.7)$ 之间徘徊, 故在更长间隔的任务中 LSTM 模型也将不具备预测能力; 本文提出的 Conv1d-LSTM 模型, 在 24 次短间隔的预测结果与真实值接近, 且中长距离的预测结果波形与真实数据波形走势接近, 仍然具有很好的预测能力。

4 总结与展望

4.1 当前存在的问题

(1) Conv1d-LSTM 模型中运用了神经网络来进行计算, 属于黑箱模型, 无法从原理上找到问题的突破口, 目前阶段主要是以计算出结果来对比真实值, 再利用对比结果进行反推从而达到调优模型性能的目的。

(2) Conv1d-LSTM 的模型在该能源分配任务上效果显著, 但其泛化能力还有待验证。

(3) 目前关于能源分配的问题还未有成熟的计算体系, 难以从理论上构建更加有效的预测模型, 现阶段仍需要组合搭建不同的网络来尝试解决此类问题。

4.2 发展动向

(1) 随着“双碳”政策的提出, 越来越多的人重视能源的相关问题, 也有更多的科研工作者投身该方向的研究。文中使用的 Informer 模型虽不适合本文的任务, 但该模型也是我国科研工作者提出的有关能源分配预测的新算法模型。

(2) 未来的能源分配任务将会根据能源分配预测的特点, 从数学角度出发, 寻找对能源分配具有高影响力的环境因素, 通过对不同地方采集到的数据进行先筛选归类, 再从算法的角度组合可用的模块或搭建合适的模型。

4.3 总结

能源分配的具体数值具有一定的规律性, 非常适合用深度学习的方法来挖掘其内部各个数据间的关系, 故我们需要搜集大量的当地环境因素数据如温度、湿度及 $PM_{2.5}$ 等。本文提出的 Conv1d-LSTM 用于进行能源分配数值的预测, 拥有传统 LSTM 短距离的高精确度, 同时还能通过扩大感受野来预测长期数值, 预测的速度也不会因任务的长短不同而发生较大变化。测试结果也表明, 通过当地的环境因素数值及能源分配数值来训练 Conv1d-LSTM 模型, 能够准确地预测该地一段时间内能源所需分配的具体数值, 能够达到节能或

是充分利用能源的目的。

参考文献

- 1 施冬梅, 肖锋. 基于改进长短时记忆网络的驾驶行为检测方法研究. 汽车工程, 2021, 43(8): 1203–1209, 1262. [doi: 10.19562/j.chinasae.qcgc.2021.08.011]
- 2 张亚伟, 陈瑞凤, 徐春婕, 等. 基于 LSTM-LightGBM 模型的车站环境温度预测. 计算机测量与控制, 2022, 30(1): 20–25. [doi: 10.16526/j.cnki.11-4762/tp.2022.01.004]
- 3 黄冬梅, 王唱, 胡安铎, 等. 基于 CNN-BiLSTM 的潮汐电站潮位预测. 水力发电, 2021, 47(10): 80–84. [doi: 10.3969/j.issn.0559-9342.2021.10.016]
- 4 陈军, 陶巍, 吕英飞, 等. 基于卡尔曼滤波的短期负荷预测. 电气开关, 2014, 52(2): 43–45. [doi: 10.3969/j.issn.1004-289X.2014.02.013]
- 5 李东东, 覃子珊, 林顺富, 等. 基于混沌时间序列法的微网短期负荷预测. 电力系统及其自动化学报, 2015, 27(5): 14–18. [doi: 10.3969/j.issn.1003-8930.2015.05.03]
- 6 李晨熙. 基于 ARIMA 模型的短期电力负荷预测. 吉林电力, 2015, 43(6): 22–24. [doi: 10.3969/j.issn.1009-5306.2015.06.007]
- 7 马得银, 孙波, 刘澈. 基于天气信息的短期冷热电负荷联合预测方法. 电网技术, 2021, 45(3): 1015–1022. [doi: 10.13335/j.1000-3673.pst.2020.0794]
- 8 Zhou HY, Zhang SH, Peng JQ, *et al.* Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the 35th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2021. 11106–11115.
- 9 甘昕艳, 唐晓年. 基于 CNN 的时序数据关联规则挖掘模型. 计算机仿真, 2021, 38(3): 282–285, 326. [doi: 10.3969/j.issn.1006-9348.2021.03.058]
- 10 陈海鹏, 周越豪, 王趁录, 等. 基于改进的 CNN-LSTM 短期风功率预测的系统旋转备用经济性分析. 高电压技术, 2022, 48(2): 439–446. [doi: 10.13336/j.1003-6520.hve.2020.1850]
- 11 Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.
- 12 Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature, 1986, 323(6088): 533–536. [doi: 10.1038/323533a0]

(校对责编: 牛欣悦)