

杭州电子科技大学

本科毕业设计

(2023 届)

题 目	基于深度学习的光伏发电量预测方法的实现与分析
学 院	通信工程学院
专 业	通信工程
班 级	19083413
学 号	19081312
学生姓名	高海涛
指导教师	刘春山
完成日期	2023 年 6 月

## 诚信承诺

我谨在此承诺：本人所写的毕业论文《基于深度学习的光伏发电量预测方法的实现与分析》均系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释，若有不实，后果由本人承担。

承诺人（签名）：

2023 年      月      日

## 摘 要

本研究提出了一种基于深度学习算法的光伏发电量预测方法，用来解决长短期预测问题。该方法基于长短记忆网络（LSTM）。在数据处理方面，我们对数据集进行了多项处理，包括数据集融合、缺失值处理、数据重采样、数据集划分、数据归一化和获取数据标签。为了增强模型对于时序信息（如周期性、尖峰、高峰、低谷等特征）的学习，我们采用了碎片化（patch）的方法，将连续的长序列按天切片，并按小时对齐。在短期预测方面，我们将 Patch-LSTM 与 ARIMA 模型进行了对比；在长期预测方面，我们与 MLP 和 DLinear 模型进行了对比。在模型训练方面，我们采用了 Adam 方法进行梯度下降，并使用均方误差（MSE）作为损失函数。为了避免过拟合，我们使用了早停方法训练模型。模型的输入包括前若干小时的发电量、电流和逆变器温度等特征信息，用于预测未来 1 小时和 24 小时的发电量。在模型评估阶段，我们使用了多个指标：均方误差（MSE）、平均绝对误差（MAE）、均方根误差（RMSE）和平均绝对百分比误差（MAPE）。这些指标能够更公平、更综合、更全面地评价和比较不同模型的性能。最后，我们将本文的长短期预测模型与传统时序预测模型 ARIMA、深度学习模型多层感知机（MLP）和 DLinear 进行了对比。对比结果表明，本文的预测方法具有更高的精确性和更强的稳定性。

**关键词：**深度学习、光伏发电、时间序列、数据处理、MLP、LSTM

## ABSTRACT

This study proposes a deep learning-based method for photovoltaic (PV) power generation forecasting to address both short-term and long-term prediction challenges. The method is built upon the long short-term memory (LSTM) network. In terms of data processing, various techniques were employed including data fusion, missing value handling, data resampling, dataset partitioning, data normalization, and label acquisition. To capture temporal patterns such as periodicity, peaks, and valleys, the study adopts a patch-based approach by dividing continuous long sequences into daily slices and aligning them at an hourly level. For short-term forecasting, the Patch-LSTM model is compared against the ARIMA model, while for long-term forecasting, comparisons are made with MLP and DLinear models. During the model training phase, the Adam optimization method is used for gradient descent, and mean squared error (MSE) is employed as the loss function. To prevent overfitting, early stopping is implemented. The model takes inputs such as the previous 24 hours' power generation, current, and inverter temperature as features, and predicts the power generation for the next 1 hour (short-term) and 24 hours (long-term). For model evaluation, a range of metrics including MSE, mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) are utilized, enabling a fair, comprehensive, and holistic assessment and comparison of different models. Finally, the proposed short-term and long-term forecasting models are compared against traditional time series forecasting model ARIMA, as well as deep learning models MLP and DLinear. The comparative results demonstrate that the proposed approach achieves higher accuracy and greater stability in power generation forecasting.

**Keywords:** Deep learning, photovoltaic power generation, time series, data processing, LSTM, MLP, ARIMA.

## 目 录

1	绪论.....	1
1.1	研究背景及意义.....	1
1.2	国内外研究现状.....	1
1.3	研究内容.....	2
1.4	论文组织结构.....	2
2	基本原理.....	4
2.1	时间序列.....	4
2.2	时间序列预测.....	4
2.3	深度学习.....	5
2.4	神经网络.....	6
2.5	批标准化和反批标准化.....	7
2.6	模型评价指标.....	7
2.7	本章小结.....	9
3	光伏发电量预测相关技术.....	10
3.1	差分自回归移动平均（ARIMA）模型.....	10
3.2	多层感知机（MLP）.....	10
3.3	长短期记忆单元反馈神经网络（LSTM）.....	12
3.3	本章小结.....	16
4	实验过程.....	17
4.1	台州光伏数据集.....	17
4.2	数据预处理.....	17
4.3	发电量短期预测.....	20
4.4	发电量长期预测.....	27
4.5	本章小结.....	35
5	总结.....	36
	致谢.....	38
	参考文献.....	39

# 1 绪论

## 1.1 研究背景及意义

近年来,随着全球变暖问题的不断加剧,发展可再生能源成为了各国共同努力的目标。太阳能作为一种广泛可用且环保的可再生能源,已成为许多国家日益重要的能源来源之一<sup>[1]</sup>。光伏发电作为太阳能的一种主要形式,具有趋势和周期性,其发电量的准确预测对于电力系统的稳定性和普及推广具有重要意义。然而,由于光伏发电的间歇性特性以及预测的不确定性,准确地预测光伏发电量成为了一项重要的研究课题。基于深度学习技术的光伏发电量预测方法可以有效地提高预测的准确性和可靠性,有望为工业应用和可再生能源的普及做出重要贡献<sup>[2]</sup>。

时间序列分析是一种用于处理时间相关数据的统计分析方法,具有广泛的应用领域<sup>[3]</sup>。在经济、物流、医疗等领域,时间序列分析已经得到了广泛的应用。随着5G时代和物联网的到来,大量传感器设备被部署在各种设施和设备上,产生了大量的时间序列数据。对这些数据进行分析和预测,可以帮助我们更好地理解自然现象和社会现象,提高生产力和生活质量。季节性分析、趋势分析、周期性分析和异常检测等是时间序列分析的常用方法<sup>[4]</sup>。在本研究中,我们将应用这些方法和深度学习技术来预测光伏发电量,提高预测的准确性和可靠性。

总之,本研究旨在探究基于深度学习技术的光伏发电量预测方法,以提高光伏发电量预测的准确性和可靠性,为工业应用和可再生能源的普及推广做出贡献。

## 1.2 国内外研究现状

在文献中,目前预测光伏系统发电量的方法大致可分为三类:物理方法、统计方法和深度学习方法。物理方法通过利用大气参数如温度、压力和风速等,构建复杂的数学模型和大量参数,以估算日辐照度,进而预测光伏系统的发电量<sup>[5]</sup>。但是该方法需要专业知识和复杂计算设备才能实现。统计方法包括时间序列传统算法和机器学习模型<sup>[6][7]</sup>。其中时间序列预测经典方法 ARIMA 只能进行短期预测,而在长期预测领域的误差累计使得其表现并不卓越。基于人工神经网络的算法属于深度学习算法<sup>[2][8]</sup>。随着深度学习和大数据的发展,深度学习成为最为流行的预测方法,同时它也提供更好的预测效果。据研究,人工神经网络最适合预测短期光伏功率预测和在线时序极佳的自适应网络极限学习机;而 Bootstrap 技术最适合估计不确定。此外,<sup>[10,9]</sup>发现卷积神经网络(CNN)在引发模型的深层基础方面表现出色非线性输入输出关系。<sup>[11]</sup>循环神经网络(RNN)的出现使得模型可以容易捕捉时序特征。<sup>[12]</sup>LSTM 的出现则解决了长时间序列训练中梯度消失和梯度爆炸的问题。<sup>[13]</sup>Transformer 架构中注意力机制则使得模型可以捕捉整个时间序列的特征。由此得出结论,深度学习模型为光伏发电预测计划提供了新的希望,尤其是在

混合人工神经网络和进化算法的使用方面。此外，光伏发电功率预测受很多关键因素的影响，尤其是光伏发电预测模型的输入，经过研究光辐照度、天气、湿度等均对光伏发电产生影响。基于深度学习的预测有望在应对现代电网系统面临的挑战所需的光伏功率预测中发挥关键作用。

从方法论的角度来看，可再生能源预测被认为是一种大数据应用，因此决定输入和输出的数据质量至关重要。

### 1.3 研究内容

光伏发电的间歇性和波动性是电力系统中需要克服的难题，因此准确预测光伏发电量对于保持电力系统的稳定性至关重要。然而，预测光伏发电量存在以下几个问题：

- 光伏发电系统具有间接性，并且其波动性不规律。
- 传感器是建造成本中非常重要的一部分，大多数基站缺乏采集温度、湿度、风速、日辐照度等指标的传感器。因此，可利用的外部数据十分稀少。
- 尽管传统的短期预测模型已经具有良好的精度，但是仍有巨大的提升空间。
- 对于一个较长时间的预测（如 24 小时、48 小时、一周、一个月等），目前并没有模型做出较好的预测。传统的统计方法存在累计误差，一次的不准确预测将会造成巨大的灾难。

本研究旨在基于深度学习的方法解决以上问题，实现光伏发电量的短期和长期预测。具体来说，本研究将回顾前人的研究成果，并介绍基于统计学习和深度学习的预测方法。在此基础上，我们将建立适合光伏发电系统的深度学习模型，针对选取的数据进行建模和研究，并进行预测。同时，我们还将通过评价指标改进模型，构建出适合用于光伏发电的预测方法。

本研究的创新点在于解决了利用少量特征构建强大的数据特征提取器的问题，同时改进了深度学习模型在波动较为剧烈的时间序列上的长期预测效果。通过本研究，我们希望能够为解决光伏发电预测中存在的问题提供一定的帮助。

### 1.4 论文组织结构

本文的内容一共包括五个章节，各章节的内容安排如下：

第一章：绪论。本章介绍了光伏发电对于有效使用太阳能清洁能源，对于可再生能源开发和利用的重大意义，其次简要介绍课题中时间序列的概念和包含相关阐明了本文所选课题考虑的应用场景，对光伏发电预测的国内外研究现状进行了整理，最后给出了本文各章节的内容安排。

第二章：基本原理。本章介绍了时间序列、时间序列预测和深度学习神经网络的一些基本组件的定义，数学表达和发展历史，还介绍了在深度学习训练过程中经

常使用的技巧批标准化和反标准化的定义以及数据表达，最后介绍了一些常用的模型评价指标并确定了本文选择使用的模型评价指标。

第三章：光伏发电量预测相关技术。本章介绍了关于时间序列预测的四种常用方法——基于传统统计学习的时间序列预测方法 ARIMA 以及基于深度学习的时间序列预测方法 MLP, LSTM 的神经元的结构和数学形式，以及在神经网络发挥的作用，最后介绍了神经网络中的重要算法：前向传播和反向求导。

第四章：预测实验。本章首先对数据进行预处理操作，在短期预测实验中对本文提出的 Patch-LSTM 与 ARIMA 进行了对比；在长期预测实验中将 Patch-LSTM 与经典的神经网络模型 MLP 和先进时序预测模型 DLinear 设计了对比实验得到了相同基站的不同模型的预测精度。

第五章：总结。对全文所做工作进行总结。



## 2 基本原理

### 2.1 时间序列

时间序列是一种数据类型，它按照时间的顺序记录了某个现象的变化过程。时间序列的特点是数据点之间存在一定的时间间隔，这些数据点可能在等间隔的时间点上采集，也可能在不等间隔的时间点上采集。

假设有 $N$ 个固定长度为 $T$ 的时间序列，其中第 $i$ 个序列的时间序列数据可以表示为式 (2.1)：

$$(X_{1:T}^{(i)}, Y_{1:T}^{(i)}) = \left[ (x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), \dots, (x_T^{(i)}, y_T^{(i)}) \right] \quad (2.1)$$

在表达式中， $x_t^{(i)}$  是时刻 $t$ 的输入， $y_t^{(i)}$  是时刻 $t$ 的输出。在时间序列的预测中，应该有  $x_t^{(i)}, y_t^{(i)} \in R$  以及  $y_t^{(i)} = x_{t+1}^{(i)}$ 。为了方便起见，我们将用  $X_{1:T} = [X_1, \dots, X_T]$  和  $Y_{1:T} = [y_1, \dots, y_T]$  来表示一般的时间序列。

时间序列通常被用来描述各种现象的变化，例如气温、股票价格、销售额、心电图等等。

时间序列数据具有一些特殊的性质。首先，时间序列数据的趋势通常是非线性的，因为趋势可能会随时间变化而变化。其次，时间序列数据通常存在季节性和周期性的变化，因为许多现象会受到季节、周期等因素的影响。此外，时间序列数据还可能存在着噪声，噪声是指随机误差或者非系统性误差。

在实际应用中，时间序列数据的分析和预测被广泛应用于金融、医疗、天气预报、工业生产等领域。对于预测性应用，时间序列的分析和预测可以帮助人们更好地理解各种现象的变化规律，并能够提前进行相应的决策和调整，以减少风险和提高效率。

### 2.2 时间序列预测

时间序列预测是指对时间上连续的数据进行分析和建模，从而对未来的时间点或时间段进行预测的一种方法。它的应用广泛，包括金融市场、气象学、医疗保健等领域。

时间序列通常由时间点和相应的数值组成。通过对历史数据的分析，可以得到时间序列的一些特征，例如趋势、季节性和周期性等。然后，可以利用这些特征来建立模型，预测未来时间点或时间段的数值。如果进行公式化，即给定的观测值  $(X_{1:T}, Y_{1:T})$  和未来的输入  $X_{T:T+K}$ ，来预测未来的输出  $Y_{T:T+K}$ 。在时刻  $t$ ，给定一个模型的输入为  $x_t$ ，此时的输出为  $o_t$ ，那么时间序列预测的优化公式就表示为式 (2.2)：

$$\min \sum_{t=1}^T \|o_t - y_t\|^2 \quad (2.2)$$

时间序列预测的方法包括传统的统计学方法和机器学习方法。在传统的统计

学方法中,常见的方法包括平滑方法、指数平滑方法、自回归移动平均模型(ARMA)和自回归积分移动平均模型(ARIMA)等。这些方法通常需要对数据的特征进行假设,并且需要对模型参数进行估计

机器学习方法则不需要对数据特征做出假设,而是通过训练模型来自动学习数据的特征。常见的机器学习方法包括支持向量回归(SVR)、随机森林回归(RFR)和长短时记忆网络(LSTM)等。

近年来,深度学习已经成为时间序列预测中的一种重要方法,其主要优势在于可以自动学习时间序列中的特征,无需手工设计特征。常见的深度学习模型包括循环神经网络(RNN)、长短时记忆网络(LSTM)、门控循环单元网络(GRU)等。

## 2.3 深度学习

深度学习是人工智能领域的一种重要技术,具有极高的应用价值。随着计算机技术的飞速发展,深度学习得以快速发展,从最初的人工神经网络,到卷积神经网络、循环神经网络和生成对抗网络等,其发展历程呈现出不断迭代和创新的特点。

20 世纪 50 年代,人工神经网络(Artificial Neural Networks,简称 ANN)被提出。最初的神经网络主要由感知机(Perceptron)组成,能够实现二分类问题的解决,但是其深度有限,无法处理复杂的非线性问题。

20 世纪 80 年代,通过引入多层神经网络的方法,实现了更深的网络结构,但是训练难度较大,存在梯度消失或梯度爆炸等问题。

20 世纪 90 年代,支持向量机(Support Vector Machine,简称 SVM)和决策树(Decision Tree)等机器学习算法逐渐成为主流,神经网络的发展受到了抑制。

2006 年,Geoffrey Hinton 等人提出了深度置信网络(Deep Belief Networks,简称 DBN),并发明了有效的训练算法——贪婪逐层训练(Greedy Layer-Wise Training),这标志着深度学习的重要突破,使得神经网络能够实现更深层次的结构,同时也解决了传统神经网络训练难的问题。

2012 年,Alex Krizhevsky 等人提出了基于卷积神经网络(Convolutional Neural Network,简称 CNN)的深度学习模型——AlexNet,获得了当年的 ImageNet 挑战赛冠军,引起了全球的关注。CNN 在计算机视觉领域得到广泛应用,并取得了很大的成功。

2014 年,Ian Goodfellow 等人提出了<sup>[14]</sup>生成对抗网络(Generative Adversarial Networks,简称 GAN),GAN 可以通过生成器和判别器的博弈学习,生成高质量的样本数据,被广泛应用于图像生成、语音合成和自然语言处理等领域。

2015 年,循环神经网络(Recurrent Neural Network,简称 RNN)的变种——长短时记忆网络(Long Short-Term Memory,简称 LSTM)被提出,解决了传统 RNN 存在的梯度消失或梯度爆炸的问题,成为处理序列数据的重要工具,被广泛应用于语音识别。

2016 年, AlphaGo 以 4:1 的成绩战胜了世界围棋冠军李世石, 这标志着深度学习在复杂博弈领域取得了重大突破, 同时也引发了人工智能领域的革命。

2017 年, 深度学习在自然语言处理领域得到了广泛的应用, Google 提出的 Transformer 模型大大提高了机器翻译的效果, 成为了该领域的标杆模型。同时, BERT 模型也成为了处理自然语言处理任务的重要模型。

2018 年, 深度强化学习取得了重要进展, DeepMind 提出的 AlphaZero 模型在围棋、国际象棋和将棋等游戏中取得了惊人的成绩, 表明深度学习在强化学习领域的应用前景广阔。

2019 年至今, 深度学习在诸多领域都取得了突破性进展。在自然语言处理领域, BERT 模型被进一步优化, 成为了处理文本分类、命名实体识别等任务的重要工具。在计算机视觉领域, YOLOv4 模型和 EfficientDet 模型在目标检测和实例分割任务上取得了领先的成绩。在自动驾驶领域, 深度学习被广泛应用于目标检测、车道线检测、跟车等任务中, 成为了自动驾驶技术的重要支撑。

## 2.4 神经网络

神经元是神经系统的基本单位, 也是神经网络的基本组成部分。它是一个接收输入信号, 并将其转换成输出信号的计算单元。

神经元的输入输出关系可以用以下公式表示:

$$y = f(\sum_{i=1}^n w_i x_i + b) \quad (2.3)$$

其中,  $x_i$  是输入信号的第  $i$  个分量,  $w_i$  是第  $i$  个输入信号对应的权重,  $b$  是神经元的偏置 (也被称为阈值),  $f$  是神经元的激活函数,  $y$  是神经元的输出信号。

神经网络是一种模仿生物神经网络工作方式的计算模型, 由一系列神经元组成。如图 2-1 所示:

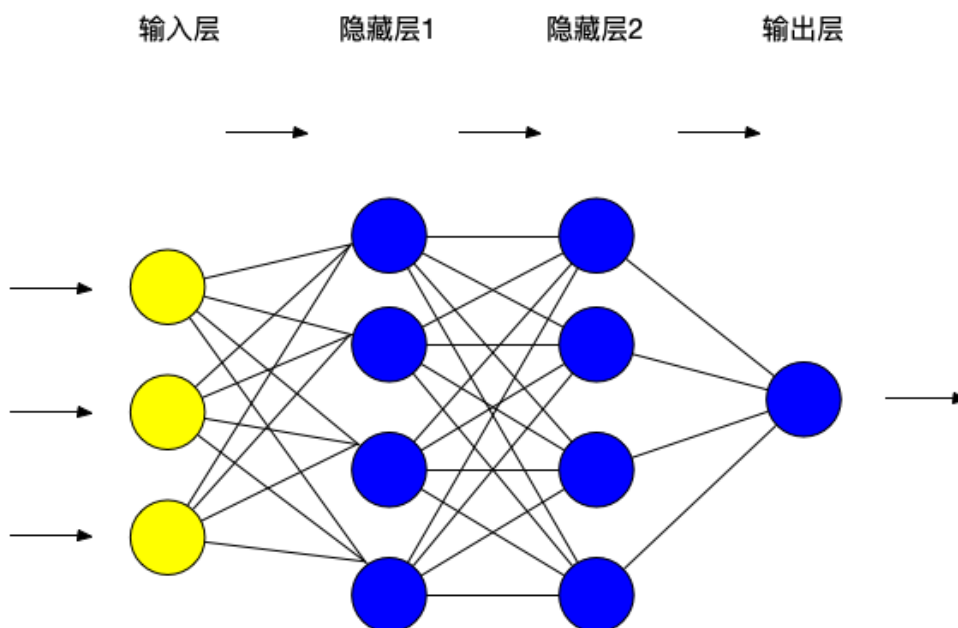


图 2-1 神经网络结构图

神经网络的主要任务是对输入数据进行处理，通过对数据的加工，将其转换为输出结果。神经网络由输入层、隐藏层和输出层组成，其中每一层都包含若干个神经元。输入层接收原始数据，隐藏层对数据进行加工处理，输出层则产生最终的结果。

## 2.5 批标准化和反批标准化

Batch Normalization (BatchNorm)和 De-Batch Normalization (De-BatchNorm)是神经网络中常用的正则化方法，用于加速模型的训练并提高其性能<sup>[15]</sup>。

BatchNorm 是在每个 mini-batch 数据上对输入数据进行归一化，即使得其均值为 0，标准差为 1，使得数据分布更加稳定。BatchNorm 的公式如下：

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2.4)$$

其中， $x$ 是输入数据， $\mu_B$ 和 $\sigma_B^2$ 分别是当前 mini-batch 中的均值和方差， $\epsilon$ 是为了避免分母为 0 而加的小常数。然后，对于每个特征通道，使用可学习的缩放因子 $\gamma$ 和偏移因子 $\beta$ 对数据进行缩放和平移：

$$y = \gamma \hat{x} + \beta \quad (2.5)$$

这个缩放和平移操作可以使得网络能够自适应地学习到每个特征通道的重要性，从而提高网络的表现力和泛化能力。

De-BatchNorm 是 BatchNorm 的逆操作，可以用于网络的 fine-tuning 或特定任务下的前向传播。它的公式如下：

$$x = \frac{y - \beta}{\gamma} \sqrt{\sigma_B^2 + \epsilon} + \mu_B \quad (2.6)$$

其中， $y$ 是经过 BatchNorm 后的输出， $\gamma$ 和 $\beta$ 是 BatchNorm 中学习到的参数， $\mu_B$

和 $\sigma_B^2$ 是 BatchNorm 中当前 mini-batch 的均值和方差。de-BatchNorm 的作用是将经过 BatchNorm 的输出重新变为未经过 BatchNorm 的原始数据，使得可以在该数据上进行进一步的操作，例如进行可视化或计算特征的统计信息等。

BatchNorm 和 De-BatchNorm 是神经网络中非常有用的正则化方法，可以使网络更加稳定、泛化能力更强、训练更加高效。

## 2.6 模型评价指标

构建和训练好深度学习模型后需要使用模型评价指标来对模型的能力进行量化的评价。深度学习模型评价指标是用于衡量机器学习模型性能的度量标准，<sup>[16]</sup>用于评估模型在给定数据集上的预测精度。常见的模型评价指标有：平均绝对误差（Mean Absolute Error, MAE）、均方误差（Mean Square Error, MSE）和均方根误差（Root Mean Square Error, RMSE），平均绝对百分比误差（Mean Absolute

Percentage Error, MAPE) 和均方根百分比误差 (Root Mean Square Percentage Error, RMSPE)。一般评价指标输出的值越小, 意味着预测精度越高。假设在一段时间内, 数据测试集的真实值为  $y_i$ , 对应的模型预测值为  $\hat{y}_i$ ,  $N$  为数据测试集的样本数量。

#### (1) 平均绝对误差 (MAE)

平均绝对误差 (MAE) 的计算公式如式 (2.7) 所示:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.7)$$

对于长度为  $N$  的时间序列数据, 将其中每一个原始数据和预测数据一一相减取绝对值, 再对所有绝对值求和计算这个差值的平均值, 就得到了平均绝对误差。它反映的是预测数据相对于原始数据的偏移平均值。

#### (2) 均方误差 (MSE)

均方误差 (MSE) 的计算公式如式 (2.8) 所示:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.8)$$

均方误差和平均绝对误差的计算方法类似, 但是它是将模型预测值和真实值之间的绝对值求了平方和后再取平均值, 若存在异常值导致模型预测值和真实值相差较大, 则求出的均方误差就会比较大。

#### (3) 均方根误差 (RMSE)

均方根误差 (RMSE) 的计算公式如式 (2.9) 所示:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (2.9)$$

均方根误差是在均方误差的计算基础上, 对均方误差取数学平方根。和均方误差相同, 均方根误差对于异常值也比较敏感。

#### (4) 平均绝对百分比误差 (MAPE)

平均绝对百分比误差 (MAPE) 的计算公式如式 (2.10) 所示:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (2.10)$$

对于长度为  $N$  的时间序列数据, 将其中每一个原始数据和预测数据一一相减取绝对值, 再将绝对值除以原始数据, 求和计算这个值的平均值, 就得到了平均绝对百分比误差。它反映的是预测数据相对于原始数据的偏移百分比。

#### (5) 均方根百分比误差 (RMSPE)

均方根百分比误差 (RMSPE) 的计算公式如式 (2.11) 所示:

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{y}_i}{y_i} \times 100 \right)^2} \quad (2.11)$$

绝对误差和相对误差都是用于衡量预测值和真实值之间的误差的指标。它们的优劣会因具体应用场景和任务而不同。相对误差可以用来衡量预测误差与真实值的比例，具有更强的可解释性，可以反映出误差随着真实值的变化而变化的趋势。而绝对误差则可以更好地反映出预测误差的绝对大小，更直观地表示出预测的准确性。

在应用中，相对误差通常用于需要比较不同规模或数量级的预测值的任务，如销售额预测、股票价格预测等。而绝对误差则更适用于需要精准预测的任务，如机器人导航、自动驾驶等。在不同场景下，选择合适的误差指标可以更好地评估模型的性能并进行参数调整。

本文中为了全面体现模型之间的差异，使用了绝对误差：MSE，MAE。和相对误差 RMSE，MAPE。

## 2.7 本章小结

本章首先介绍了时间序列的定义和公式化表达，接着说明了时间序列预测定义和在实际生活中的应用场景。其次，紧扣本文主题介绍了时间序列预测方面的常见方法 ARIMA 和深度学习方法，其中重点介绍了深度学习的发展历史，神经网络的定义和数学表达。另外，介绍了构建深度模型时经常使用的技巧批标准化和反批标准化，以便提高模型的预测能力。最后，介绍了模型评价指标并且指出了本文使用的四个评价指标

### 3 光伏发电量预测相关技术

#### 3.1 差分自回归移动平均（ARIMA）模型

##### 3.1.1 自回归（AR）模型

自回归模型是基于线性回归的一种预测方法，它通过分析一个变量过去的数值来预测该变量自身的未来值，因此只适用于与该变量密切相关的数据。要使用 AR 模型进行预测的时间序列必须满足平稳时间序列的要求。自回归模型的计算公式如式（3.1）所示：

$$y_t = \mu + \sum_{i=1}^p \gamma_i y_{t-i} + \varepsilon_t \quad (3.1)$$

其中， $\mu$  为常数项， $p$  为阶数， $\gamma_i$  为自相关系数， $\varepsilon_t$  表示误差， $y_t$  为当前时刻数据值。

##### 3.1.2 移动平均（MA）模型

移动平均模型是对自回归模型中的误差加权求和，计算公式如式（3.2）所示：

$$y_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (3.2)$$

##### 3.1.3 自回归移动平均（ARMA）模型

<sup>[17]</sup>自回归模型和移动平均模型结合，得到了自回归移动平均模型，计算公式如式（3.3）所示：

$$y_t = \mu + \sum_{i=1}^p \gamma_i y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (3.3)$$

其中， $p$  和  $q$  都是自回归移动平均模型的阶数。当  $p=0$  时就等于移动平均模型，当  $q=0$  时就等于自回归模型。

##### 3.1.4 差分自回归移动平均（ARIMA）模型

差分自回归移动平均（ARIMA）它结合了自回归模型（AR）和移动平均模型（MA），并加入了差分（I）操作，可以应对一定程度的非平稳性时间序列数据。具体地，ARIMA 模型包含三个参数： $p$ 、 $d$ 、 $q$ 。其中  $p$  表示自回归模型的阶数， $d$  表示时间序列需要差分的阶数， $q$  表示移动平均模型的阶数。在实际应用中，可以通过观察时间序列的自相关图和偏自相关图来确定  $p$  和  $q$  的值，而  $d$  的值则需要根据时间序列数据的平稳性情况来确定。

#### 3.2 多层感知机（MLP）

多层感知机（Multilayer Perceptron, MLP）是一种前向反馈神经网络，由至少三层节点组成（输入层、隐藏层和输出层），常用于分类和回归问题的解决<sup>[18]</sup>。它可以通过学习一组训练数据来建立非线性的模型，实现对未知数据的预测。模型结构如图 3-1 所示：

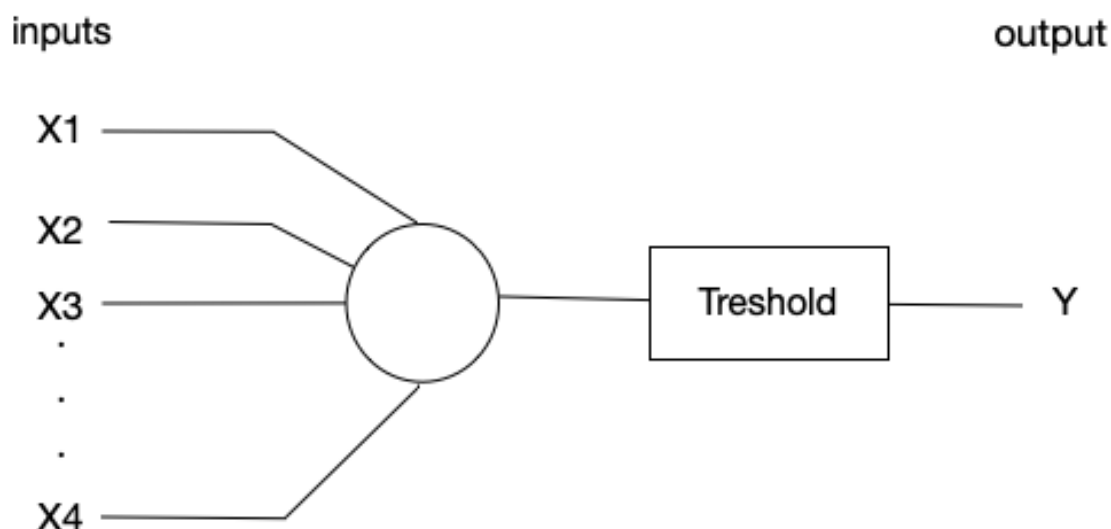


图 0-1 MLP 神经元示意图

MLP 的基本单元是人工神经元，它可以接受多个输入，并生成一个输出。每个神经元都有一个或多个输入权重，以及一个偏置项，这些权重和偏置项是 MLP 学习的参数。每个神经元还使用一个激活函数将加权输入求和的结果转换为输出。一个 MLP 神经网络如图 3-2 所示：

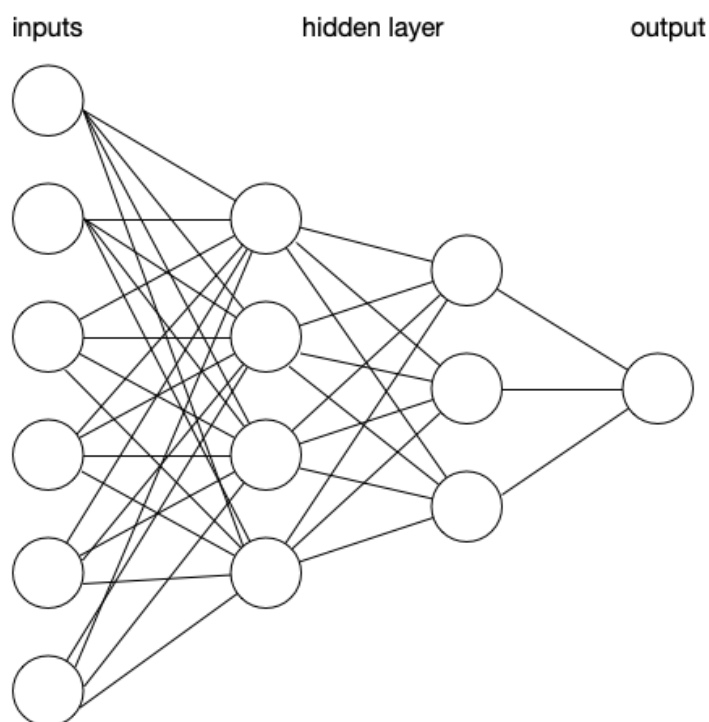


图 0-2 MLP 神经网络示意图

如果使用公式化的语言描述 MLP 网络：假设我们有一个包含  $n$  个样本的训练集  $D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，其中  $x_i$  是输入向量， $y_i$  是输出标签。

对于一个有  $L$  层的 MLP，第  $l$  层 ( $l \in [1, L]$ ) 有  $N^{(l)}$  个神经元，每个神经元都



有一个激活函数 $f^{(l)}$ 。对于第 $l$ 层第 $i$ 个神经元，其输入为 $z_i^{(l)}$ ，输出为 $a_i^{(l)}$ 。第一层是输入层，因此 $L \geq 3$ 。

输入层接受输入向量 $x$ ，输出 $a^{(1)} = x$ 。

对于每个 $l \in [2, L]$ ，通过以下公式计算该层的输入 $z^{(l)}$ 和输出 $a^{(l)}$ ：

$$z^{(l)} = w^{(l)} a^{(l-1)} + b^{(l)} \quad (3.4)$$

$$a^{(l)} = f^{(l)} z^{(l)} \quad (3.5)$$

其中， $w^{(l)}$ 是第 $l$ 层的权重矩阵， $b^{(l)}$ 是第 $l$ 层的偏置向量， $f^{(l)}$ 是激活函数。

对于一个具有 $K$ 个输出的MLP，输出层的激活函数通常为恒等函数，因此输出 $a^{(L)}$ 即为预测值。

MLP通过最小化损失函数来学习权重和偏置项的值，常用的损失函数包括均方误差（Mean Squared Error, MSE）和交叉熵（Cross-Entropy）等。通常使用反向传播算法来计算梯度，并使用梯度下降等优化算法来更新权重和偏置项的值。

### 3.3 长短期记忆单元反馈神经网络（LSTM）

#### 3.2.1 循环神经网络（RNN）

RNN（Recurrent Neural Network，循环神经网络）是一类特殊的神经网络，主要用于处理序列数据，例如文本、音频、视频等等。相比于传统神经网络，RNN最大的特点是引入了循环结构，使得网络可以记忆之前的信息，并将之作为输入用于当前的计算。这种记忆能力使得RNN能够更好地处理序列数据，例如文本数据中的语义、音频数据中的语音识别等等。

RNN的结构可以看作是对某个时间步的输入数据进行处理，同时将上一个时间步的状态作为额外的输入进行计算。这种状态称为“隐藏状态”，并通过循环连接使得它可以在时间上持续传递。RNN的训练通常采用反向传播算法（Backpropagation Through Time, BPTT），通过对隐藏状态和输出进行误差反向传播来更新网络的权重参数。

RNN的简化结构如图3-3所示：

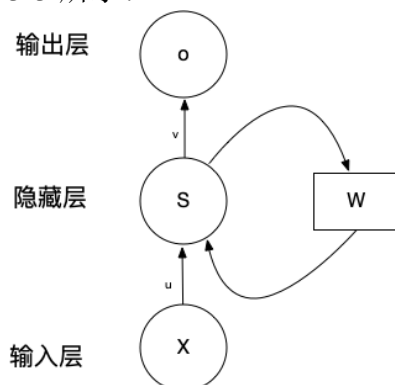


图 0-3 RNN 简化结构图

其中， $X$  表示输入， $S$  表示隐藏状态输出， $O$  表示输出。 $W$ 、 $V$  和  $U$  都表示权值，分别表示隐藏层自身各节点之间的、隐藏层到输出层的、输入层到隐藏层的权值。这个网络在  $t$  时刻接收到输入  $x_t$  之后，隐藏层的值是  $s_t$ ，输出值是  $o_t$ 。关键一点是， $s_t$  的值不仅仅取决于  $x_t$ ，还取决于  $s_{t-1}$ 。我们可以用下面的公式来表示循环神经网络的计算方法：

从输入层到隐藏层的计算如式 (3.6) 所示：

$$S_{j(\tau)} = \tanh(Z_{j(\tau)}) = g \left( \sum_{k(\tau)} U_{j(\tau),k(\tau)} X_{k(\tau)} + \sum_{j(\tau-1)} U_{j(\tau),j(\tau-1)} X_{j(\tau-1)} + b_{j(\tau)} \right) \quad (3.6)$$

该式子可以简化为式 (3.7)：

$$S_{(\tau)} = \tanh(Z_{(\tau)}) = g(UX_{(\tau)} + WS_{(\tau-1)} + b) \quad (3.7)$$

某一时刻的输出  $Y$  的计算公式如式 (3.8) 所示：

$$Y_{(\tau)} = \text{softmax}(Z_{(\tau)}) = f(VS_{(\tau)} + c) \quad (3.8)$$

式中， $b$  和  $c$  都为偏置项。

为了优化模型性能，我们需要最小化损失函数，即预测值与真实值之间的差距。为此，我们可以使用梯度下降法对损失函数进行反向传播，从最终损失值开始，逐层计算每个参数对损失函数的梯度。这个过程被称为误差反向传播。设  $E_\tau$  为时间  $\tau$  时的梯度损失， $E$  为最终的损失，则有式 (3.9)：

$$E = \sum_{\tau} E_{\tau} \quad (3.9)$$

总损失关于  $U$  的偏导数如式 (3.10) 所示：

$$\frac{\partial E}{\partial U} = \sum_{\tau=1}^t \delta_{j(\tau)} X_{k(\tau)} \quad (3.10)$$

总损失关于  $V$  的偏导数如式 (3.11) 所示：

$$\frac{\partial E}{\partial V} = \sum_{\tau} \delta_{i(\tau)} S_{j(\tau)} \quad (3.11)$$

总损失关于  $W$  的偏导数如式 (3.12) 所示：

$$\frac{\partial E}{\partial W} = \sum_{\tau=1}^t \delta_{j(\tau)} S_{j(\tau)j'(\tau-1)} \quad (3.12)$$

从理论上来说，RNN 具备随着时间传递下去并保留过去信息的特性，因此应该在处理长时间序列时表现良好。然而，在实际训练过程中，RNN 并不能很好地实现长时间序列的记忆，存在梯度消失和梯度爆炸等问题。

### 3.2.2 长短期记忆单元反馈神经网络 (LSTM)

LSTM (Long Short-Term Memory, 长短时记忆网络) 是一种常用的循环神经网络变体，旨在解决 RNN 在处理长序列时遇到的梯度消失问题。LSTM 通过引入三个门 (输入门、遗忘门和输出门)，可以更好地控制信息的流动和遗忘，使得网络可以在长序列上保持更好的记忆和预测能力。

与传统的 RNN 相比，LSTM 引入了一个新的状态向量  $h_t$ ，称为 LSTM 的记忆

状态。该状态可以被看作是 RNN 中的隐藏状态的扩展，用于记录过去的信息，并在当前时间步上更新和传递。

LSTM 包括三个门，分别是输入门、遗忘门和输出门，这些门通过对输入和隐藏状态的加权控制，决定了信息是否进入、保留或输出。每一个 LSTM 都包含着三个门结构，如图 3-4 所示：

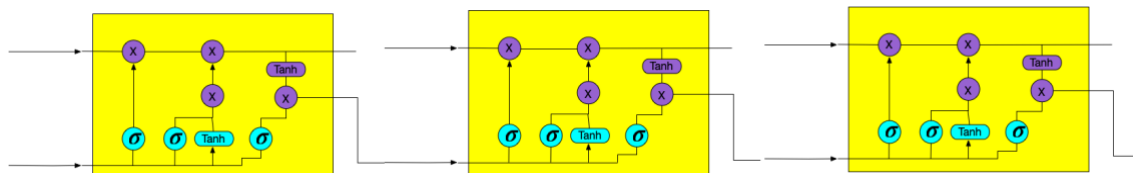


图 0-4 LSTM 的网络结构图

其中， $x_t$  为当前时刻的输入， $h_{t-1}$  为上一时刻网络输出的状态。

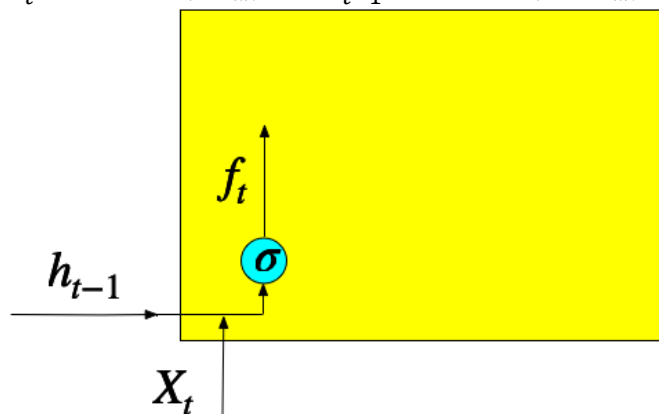


图 0-5 LSTM 前向计算遗忘门

如图 3-5 所示，LSTM 模型的前向计算过程可以分为多个步骤，其中第一步是决定上一时刻存储元中的哪些信息可以被保留或遗忘。这个决策由“遗忘门”来控制，通过将上一时刻的隐藏状态  $h_{t-1}$  和当前时刻的输入  $x_t$  输入到 sigmoid 激活函数中，得到一个 0 到 1 的输出，表示应该保留或遗忘的信息的比例。然后将该输出与上一时刻存储元中的信息  $C_{t-1}$  按位相乘，从而得到更新后的存储元信息  $C_t$ 。遗忘门的计算公式如式 (3.13) 所示：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.13)$$

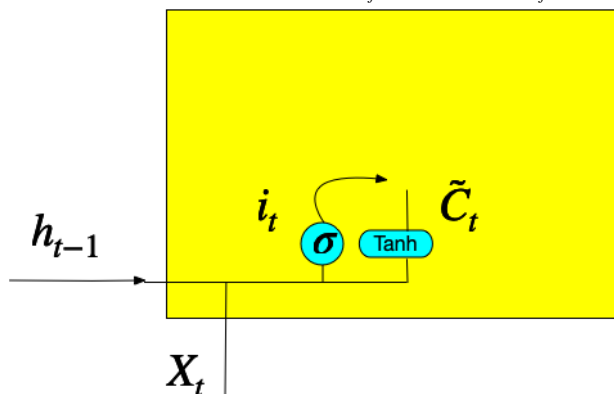


图 0-6 LSTM 前向计算输入门和隐层输出信息

如图 3-6 所示, LSTM 前向计算的第二步是产生新的存储单元信息。该步骤包含两部分, 首先是 $h_{t-1}$ 和 $x_t$ 通过一个“输入门”, 门结构前的激活函数为 **sigmoid** 函数, 在激活函数的作用下产生 0 到 1 的值, 来决定隐层输出的信息的保留或者丢弃; 第二个部分是通过一个 **tanh** 函数用来生成隐层输出信息 $C_t$ , 它是当前层产生的输出信息, 在下一步时有被添加到下一个隐层单元中的概率。输入门的计算公式如式 (3.14) 所示, 隐层输出信息的计算公式如式 (3.15) 所示:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.14)$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3.15)$$

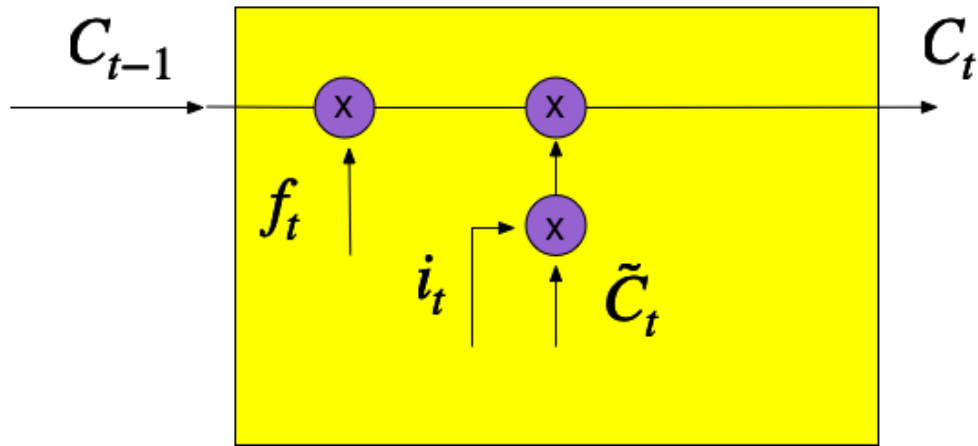


图 0-7 LSTM 前向计算过程存储元状态更新

如图 3-7 所示, 现在我们对隐层单元存储状态进行更新。将上一时刻的存储元中的信息 $C_{t-1}$ 与遗忘门输入 $f_t$ 相乘, 然后将隐层输出信息 $C_t$ 与输入门结果 $i_t$ 相乘, 最后将两者相加就得到了新的存储元状态, 计算公式如式 (3.16) 所示:

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (3.16)$$

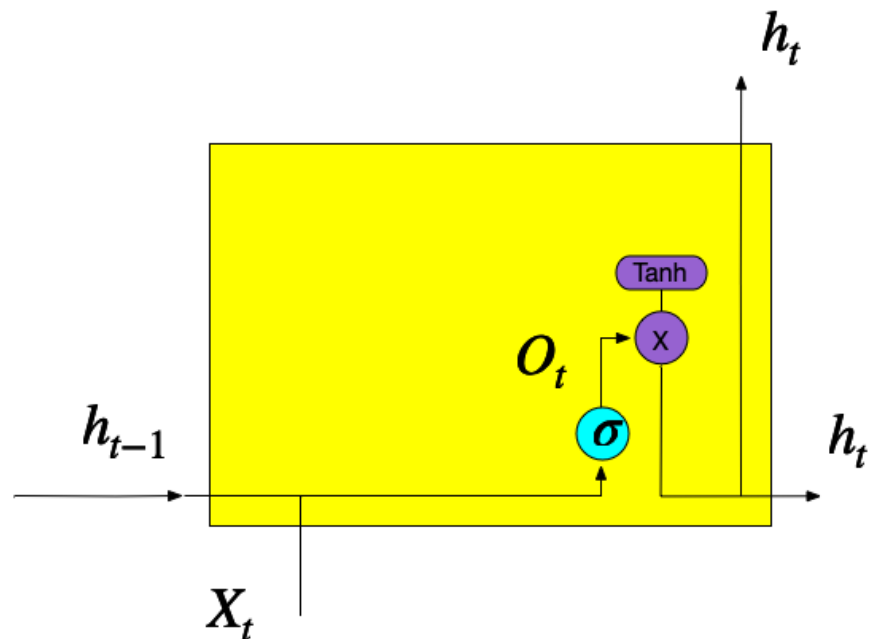


图 0-8 LSTM 前向计算输出

如图 3-8 所示，LSTM 前向计算的最后一步是计算模型的输出。 $h_{t-1}$ 和当前时刻的输入 $x_t$ 经过“输出门”，门结构前的输出函数的为 sigmoid 函数，由激活函数来得到一个 0 到 1 之间的值，然后将更新后的存储单元信息经过 tanh 函数输出一个-1 到 1 之间的值，两个输出值按位相乘，从而得到隐层的输出。输出门计算公式如式（3.17）所示，隐层输出计算公式如式（3.18）所示：

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.17)$$

$$h_t = o_t * \tanh(C_t) \quad (3.18)$$

同样的，LSTM 也具有反向误差传播过程，下游误差通过遗忘门、输入门、隐层单元和输出门反向传播到上一时刻的计算公式如式（3.19）所示：

$$\frac{\partial E_{\tau+1}}{\partial h_\tau} = \delta_{(\tau+1)}^o W^o + \delta_{(\tau+1)}^f W^f + \delta_{(\tau+1)}^i W^i + \delta_{(\tau+1)}^c W^c \quad (3.19)$$

### 3.3 本章小结

本章首先介绍了差分自回归移动平均（ARIMA）模型的演变过程，它是将差分法和自回归（AR）模型与移动平均（MA）模型相结合得到的模型。接着介绍了长短期记忆单元反馈神经网络（LSTM）的演变过程，它是由循环神经网络（RNN）加入门控单元演变而来的，最后介绍了长短期记忆单元反馈网络的前向计算流程、误差反向传播和计算公式。

## 4 实验过程

### 4.1 台州光伏数据集

本文实验的数据集来自深圳一家能源公司设立在浙江省台州市光伏基站传感器的数据，原始数据一共包含 14 个基站，30 个特征。记录了从 2021 年 4 月 14 日开始到 2022 年 12 月 31 日结束共约 20 个月的数据，数据值间隔不定，我选择了其中一个基站的数据进行实验。

### 4.2 数据预处理

#### 4.2.1 数据聚合

由于原始数据中不同月份，不同天的数据都存放在不同的 Excel 文件中。所以我首先使用 python 工具实现了一个自动将不同天，不同月份的数据按顺序聚合到同一个 Excel 文件中的脚本程序。最后使用 python 工具绘制出完整数据集的图像。如图 4-1 所示：

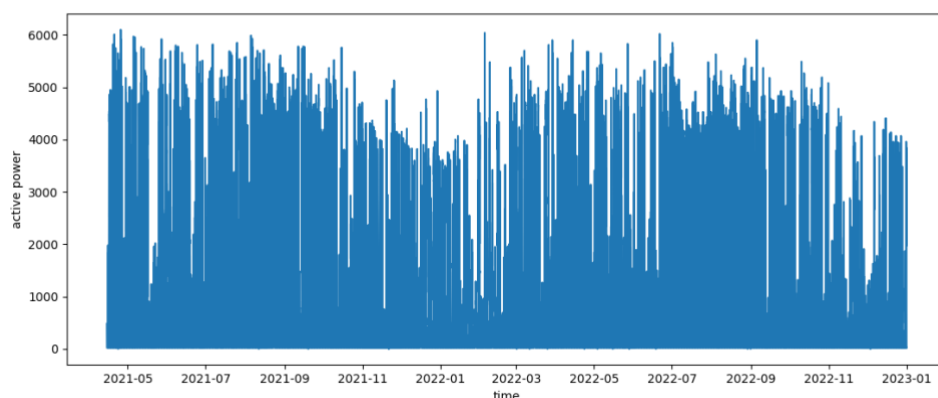


图 0-1 未经处理的原数据

由图 4-1 可知，数据最低值为 0，最高值为 6000，数据起始于 2021 年 4 月结束于 2022 年 12 月。

#### 4.2.2 数据重采样

每天数据采集的间隔不定，绘制出 2021 年 4 月 15 日的有功功率的时序图，如图 4-2 所示：

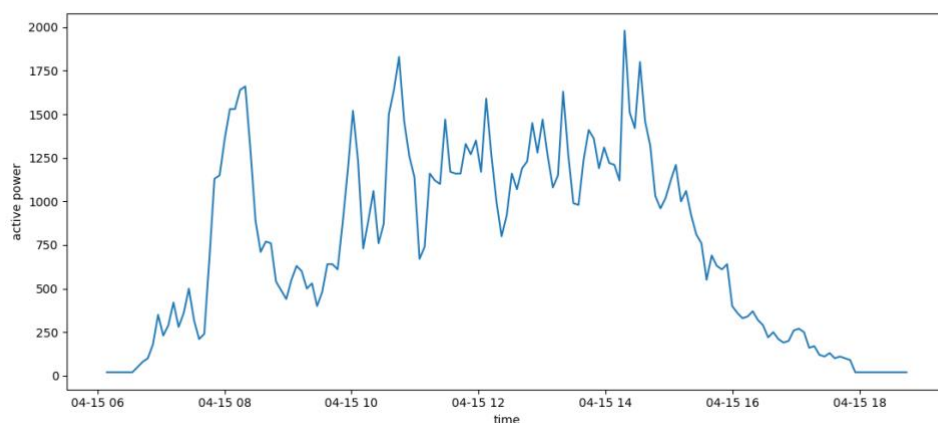


图 0-2 2021 年 4 月 15 日发电量原数据

由图 4-2 可知：传感器采集数据不是按一个固定的频率进行采集，这不方便后续模型学习。所以我进行了重采样，在数据采样频率调整为 1 个小时，每天的采集时间为 0 点到 23 点。重采样方式为将包含在这段时间内的数据求和，生成一条新的数据。将重采样后的数据集绘制，如图 4-3 所示：

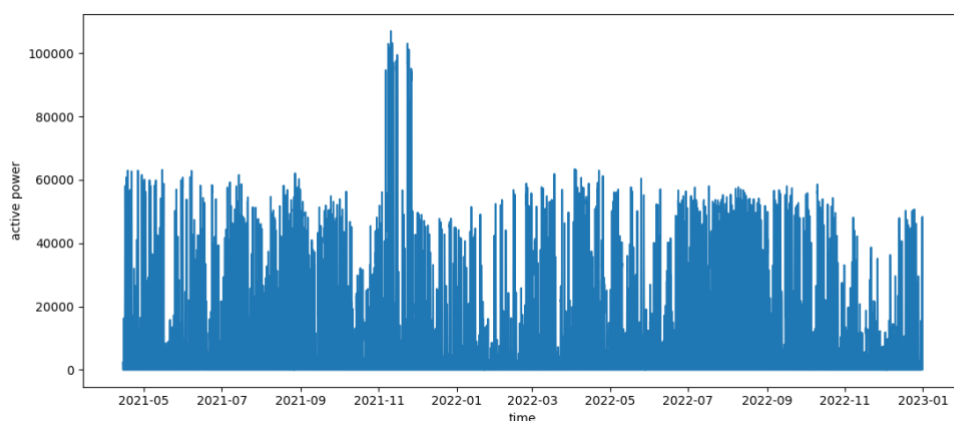


图 0-3 重采样后的数据

如图 4-3 所示，数据仍起始于 2021 年 4 月结束于 2022 年 12 月。但是由于重采样过程中的求和操作，数据的最大值变为 100000。与原始数据存在差异。每天的数据采集间隔相同，绘制 2021 年 6 月 23 日的发电量数据如图 4-4 所示：

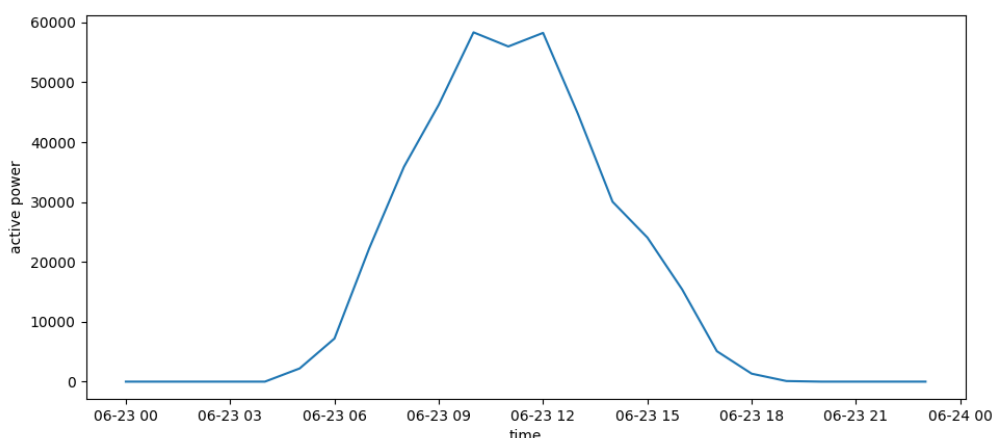


图 0-4 重采样后 2021 年 6 月 23 日的数据

如图 4-4 所示，重采样的数据变得光滑且平稳，这样可以有利于模型对于时序信息的学习。

#### 4.2.3 数据归一化

在深度学习中，数据归一化是指将原始数据通过某种方式缩放到相同的范围或分布中。数据归一化的作用主要有以下几个方面：

**提高模型收敛速度和效果：**不同特征的数值范围可能相差很大，这会导致模型在更新参数时对不同特征的梯度有差异，从而导致训练效果较差，收敛速度较慢。通过数据归一化，可以将不同特征的数值范围缩放到相同的区间内，从而减少特征间的差异，提高模型的训练效果和收敛速度。

**减少过拟合：**过拟合是深度学习中常见的问题之一，通过数据归一化可以减少不同特征之间的相关性，从而降低过拟合的风险。

**提高模型的鲁棒性：**当输入数据发生一定变化时，模型的输出也会发生相应的变化。通过数据归一化，可以使得输入数据的变化对模型的输出影响更加稳定，提高模型的鲁棒性。

**优化算法：**部分深度学习算法（如梯度下降）对数据的范围敏感，通过数据归一化可以避免算法收敛速度受到数据范围的影响，提高算法的优化效果。

根据这个数据集最大值，最小值的特点，我将数据放缩到 0-1 的范围，因为符合下限为 0 这个重要的特征，如果放缩到 -1 - 1 的范围，模型可能会学习到负数，不利于模型预测。放缩的数学公式 4-1 为：

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4-1)$$

其中， $x$  是原始数据， $x_{min}$  和  $x_{max}$  分别是原始数据的最小值和最大值， $x_{norm}$  是经过归一化后的数据。绘制归一化后的发电量数据如图 4-5 所示：



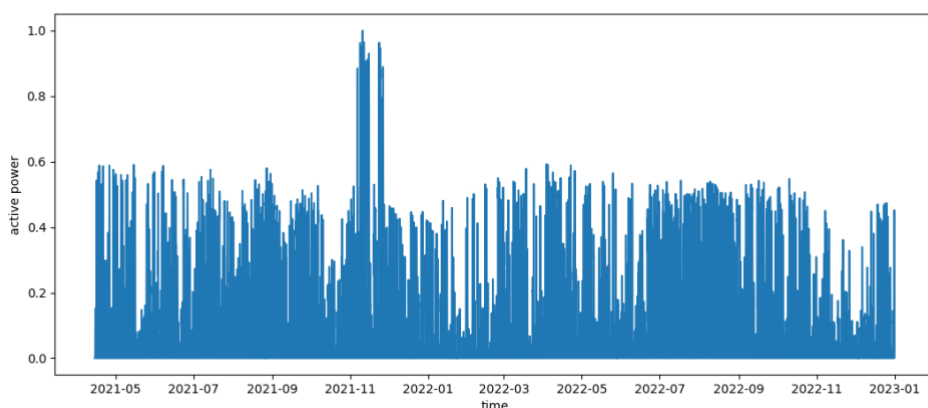


图 0-5 归一化后的发电量数据

如图 4-5 可知：数据除数值范围发生变化外，相对高低，频率，总时间均相同，保证了数据处理前后特征的一致性。

### 4.3 发电量短期预测

#### 4.3.1 实验过程

传统方法 ARIMA 模型适合进行单变量的时间序列预测，在发电量短期预测中，我会把本文提出的方法 Patch-LSTM 与 ARIMA 模型进行对比。

#### 4.3.2 时间序列分析

面对一个时间序列任务，要做的第一件事就是判断时间序列是否是噪声序列以及时间序列是否平稳。因为噪声序列不具有可预测性。而传统方法如：ARIMA 要求序列是平稳序列。

##### 4.3.2.1 白噪声检验

白噪声检验是检验时间序列是否具有随机性的一种方法。在白噪声序列中，任何两个观察值之间都不存在自相关性和异方差性，而且它们的平均值和方差都是恒定的。因此，如果时间序列是白噪声序列，则它是具有随机性的，不具备可预测性。

Ljung-Box Q 检验是一种广泛应用的统计检验方法，用于检测时间序列中是否存在自相关性。在 Ljung-Box Q 检验中，我们计算出时间序列的自相关函数值，并计算出一个统计量  $Q$ ，该统计量表示自相关函数值是否随机分布。如果统计量  $Q$  的值越大，则表明自相关函数值越可能不是随机分布，即时间序列存在自相关性。

在 Ljung-Box Q 检验中，我们通常会计算出一个  $p$  值，该  $p$  值表示统计量  $Q$  在假设时间序列不存在自相关性的情况下得到的概率。如果  $p$  值小于某个显著性水平（通常是 0.05），则可以拒绝假设时间序列不存在自相关性的假设。而本次实验中，多次  $Q$ -state 的值在发生变化，而  $P$  值始终保持 0 值。说明该序列说明该时间序列不是白噪声序列，具有可预测性。下表 4-1 是 python 中 `acorr_ljungbox` 的返回参数值：

Q-stat	P-value
12595.169974	0.0
21309.949616	0.0
26042.163925	0.0
27811.040719	0.0
28082.622761	0.0

表 0-1 ljumgbox 的返回值

如表 4-1 所示：我设置了返回五个滞后时间步的 Q 统计量和 P 值，五个滞后时间步的 P 值全为 0。

#### 4.3.2.2 平稳性检验

在时间序列分析中，平稳性检验是一个非常重要的步骤，用于确定一个时间序列是否平稳。平稳性是指时间序列在不同时间段内的统计特性如：均值、方差、自相关等不随时间变化，因此平稳性是进行时间序列分析的基础。

平稳性检验的方法有很多种，如：绘制时序图，绘制自相关图和偏自相关图，单位根检验等，单位根检验通过检验时间序列中单位根是否存在，判断时间序列是否平稳。常见的单位根检验方法有 ADF（Augmented Dickey-Fuller）检验、PP（Phillips-Perron）检验等。本次实验选择了 ADF 方法。

ADF 方法是一种常用的单位根检验方法，用于检验一个时间序列是否是平稳的。该方法基于单位根模型（unit root model），假设时间序列存在单位根（即不平稳），并通过检验是否存在单位根来判断时间序列是否是平稳的。通过 python 中 adfuller 工具包求出时间序列的统计量和 P 值，如表 4-2 所示：

ADF-stat	P-value
-11.369291	0.000000

表 0-2 adfuller 的返回值

由表 4-2 可知：P 值为 0，所以该时间序列是平稳的，可以使用 ARIMA 进行预测。

#### 4.3.3 ARIMA

经典的时间序列预测一般使用 ARIMA 方法，在短期预测中，ARIMA 方法具有精度较高、适用范围广、可解释性强、相对简单、可自适应性强的优点，所以我们将 ARIMA 方法作为短期预测（1 小时预测）的基线模型。

##### 4.3.3.1 时间序列平稳性检测

因为 ARIMA 模型对于非平稳时间序列的预测效果较差，所以使用模型前要进行时间序列平稳性检测。

在非平稳时间序列中，时间序列的均值、方差等统计特征可能随时间而变化，因此无法使用 ARIMA 模型进行准确的预测。平稳性是指时间序列在统计学意义下

具有稳定的统计特征，包括均值、方差和协方差等。

由 4.3.2 中时序平稳性分析得，该数据的发电量是平稳数据，可以使用直接使用 ARIMA 模型进行预测。

#### 4.3.3.2 确定 ARIMA 模型参数

要使用 ARIMA 模型进行预测，需要确定两个关键的参数  $p$  和  $q$ ，分别表示自回归项和移动平均项的阶数。自回归项用来捕捉时间序列自身的历史趋势对当前值的影响， $p$  的取值通常根据样本的自相关函数（ACF）和偏自相关函数（PACF）确定。移动平均项用来捕捉时间序列中的噪声对当前值的影响， $q$  的取值通常根据样本的 ACF 和 PACF 确定。所以绘制出发电量时序数据的 ACF 和 PACF 图，如 4-6 所示：

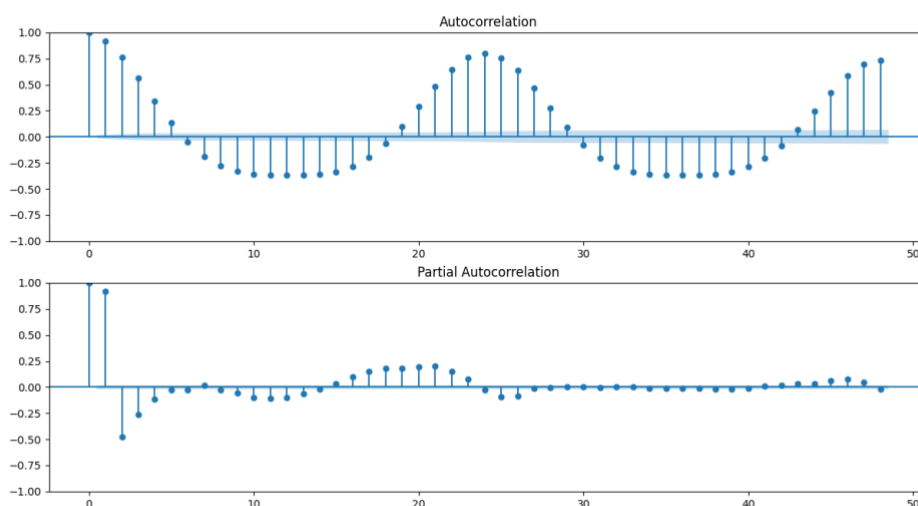


图 0-6 ACF 和 PACF

因为数据未经过差分，所以参数  $d=0$ ，由图 4-6 可知： $p=1$ ， $q=4$ ，即参数  $order=(1,0,4)$ 。

#### 4.3.3.3 ARIMA 模型预测

因为，ARIMA 模型对于时间序列的记忆能力十分有限，所以训练集输入大量数据没有任何意义，所以我截取了整个数据集的最后一个月数据进行实验，将数据集分为训练集和验证集，训练集起始于 2022 年 12 月 1 日，终止于 2022 年 12 月 29 日。测试集起始于 2022 年 12 月 30 日，终止于 2022 年 12 月 30 日。训练集共 29 天，共 696 小时，验证集共 1 天，共 24 小时。如图 4-7 所示：

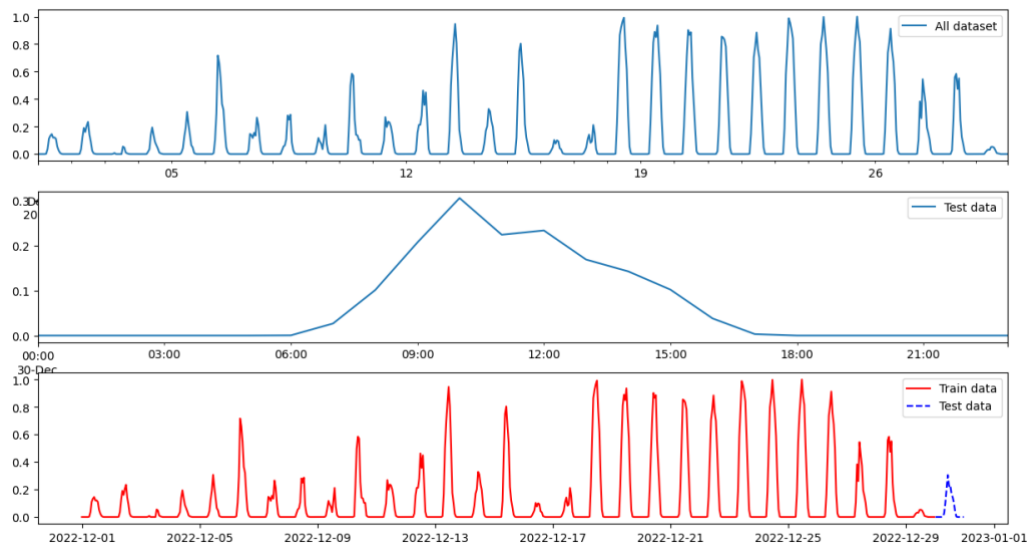


图 0-7 训练集与测试集

由图 4-7 所示：第一幅子图是整个十二月份的数据，第二幅子图是进行预测那天的真实数据，第三幅子图和训练集和测试集的分布情况。

预测过程的数据和标签由名为”滑动时间窗口”的数据分割方式产生,有一个固定的历史数据时间窗口  $T$ ，和预测时间步  $P$ ，和滑动长度  $S$ 。从下标为 0 的数据开始，预测模型中的输入  $X=T$ ，输出  $y=T:T+P$ ；然后时间窗口向前滑动，产生下一个  $X$  和  $y$ 。以此类推，形成历史数据  $X$  矩阵和真实值  $y$  向量。

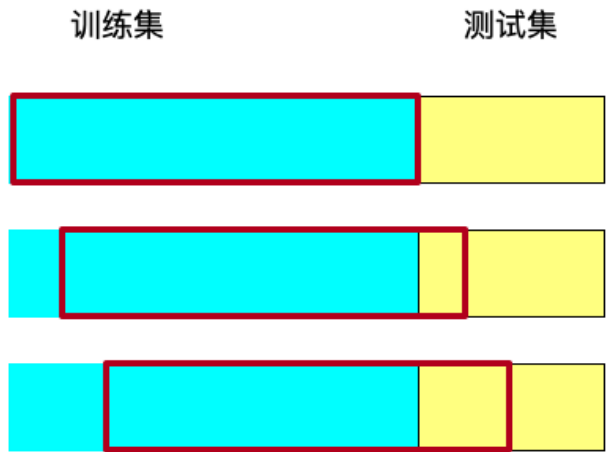


图 0-7 ARIMA 预测更新训练集过程示意图

ARIMA 模型预测的流程图 4-8 为：

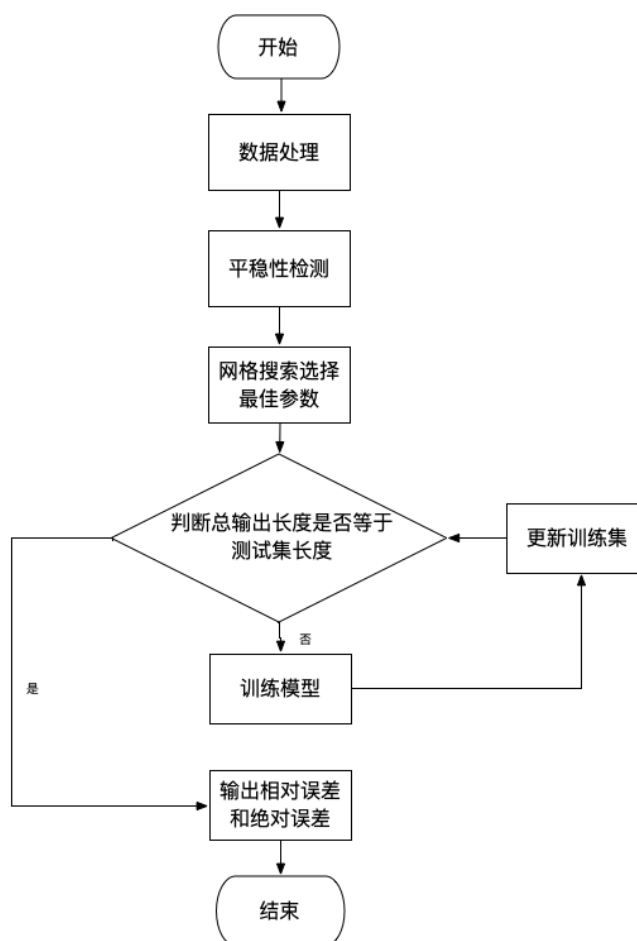


图 0-8 ARIMA 预测过程流程图

#### 4.3.3.4 网格搜索确定 ARIMA 参数

<sup>[17]</sup>在 ARIMA 模型中 AIC (Akaike Information Criterion) 是一种用于评估模型拟合质量的统计指标。它基于信息论原理, 考虑了模型的拟合能力和参数数量之间的平衡。

AIC 的计算公式为:

$$AIC = -2\log(L) + 2k \quad (4-2)$$

其中,  $L$  是模型的似然函数值,  $k$  是模型的参数数量。AIC 的值越小, 表示模型的拟合能力越好, 同时考虑到参数数量的限制, 更适合用于预测。

在 ARIMA 模型中, AIC 用于选择最优的 ARIMA 模型。通常, 会针对不同的 ARIMA 模型, 计算其 AIC 值, 然后选择具有最小 AIC 值的模型作为最优模型。这可以通过在 ARIMA 模型中尝试不同的阶数 ( $p$ 、 $d$ 、 $q$ ) 来实现, 以找到最小 AIC 值对应的参数组合。

由于 ACF 和 PACF 确定的参数就有的一定的主观性, 所以需要使用网格搜索法穷举所有可能的参数, 比较 AIC 的值, 并确定最优选择。由于未进行差分运算, 所以  $d=0$ , 在实验中  $q$  的范围是 1-6,  $p$  的范围是 1-6, 使用网格搜索法确定最优的

阶数是  $q=2, d=0, p=2$ ，最小的 AIC 是 -1723.755163705643。

#### 4.3.3.5 预测结果

ARIMA 预测结果如图 4-9 所示：

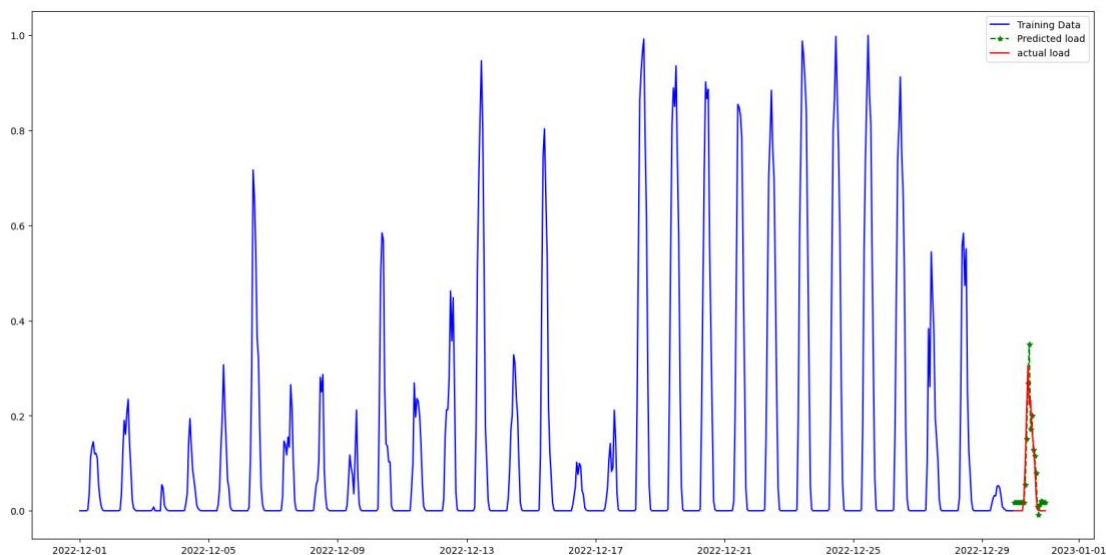


图 0-9 ARIMA 模型短期预测结果

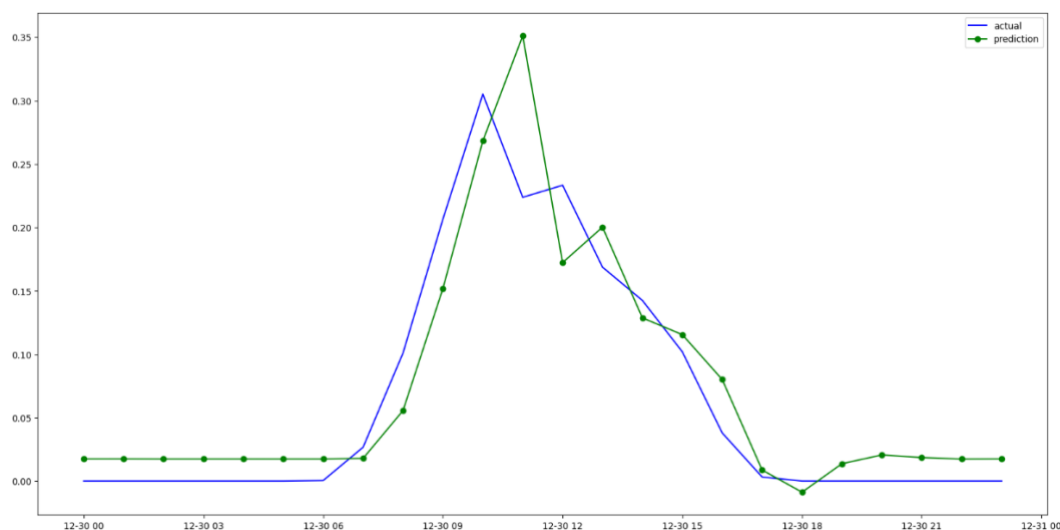


图 0-10 ARIMA 模型短期预测结果局部放大

图中蓝色线表示真实值，绿色线表示预测值，由图可知，ARIMA 模型捕捉到了时序的基本结构，但是预测结果出现负值，并且峰值预测滞后。模型误差统计如表 4-3 所示：

MSE	0.0013888212092888917
RMSE	0.03726689159681676
MAE	0.02743285985117187

表 0-3 ARIMA 预测结果误差统计

#### 4.3.4 Patch-LSTM

##### 4.3.4.1 模型超参数

本文提出的模型 Patch-LSTM 结构如 4-11 所示,在数据输入前经过一个 Batch-Norm 的处理层,可以加速模型训练,<sup>[19]</sup>接着经过 Patch 处理单元,将连续的时间序列进行切片,从而通过模型时,增加模型对于时间序列结构特征的学习能力,然后通过 LSTM 进行训练,最后经过反标准化恢复数据得到预测值。在本次实验中,使用的历史数据为前 24 小时,预测数据为后 1 小时,patch 的大小为 24,输入特征维度为 5: 当天最高温,明天最高温,当天最低温,明天最低温,和历史发电量。

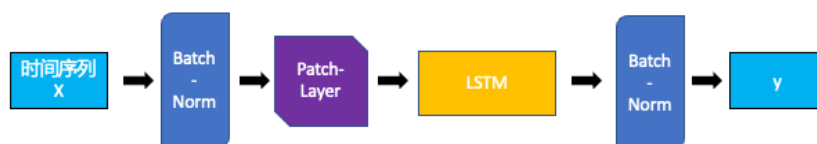


图 0-11 Patch-LSTM 模型结构图

隐藏层神经元大小为 512,一个批次 batchsize 为 128,使用的优化器为 Adam,损失函数是 MSE。

##### 4.3.4.2 模型预测结果

预测结果图如图 4-12 所示:

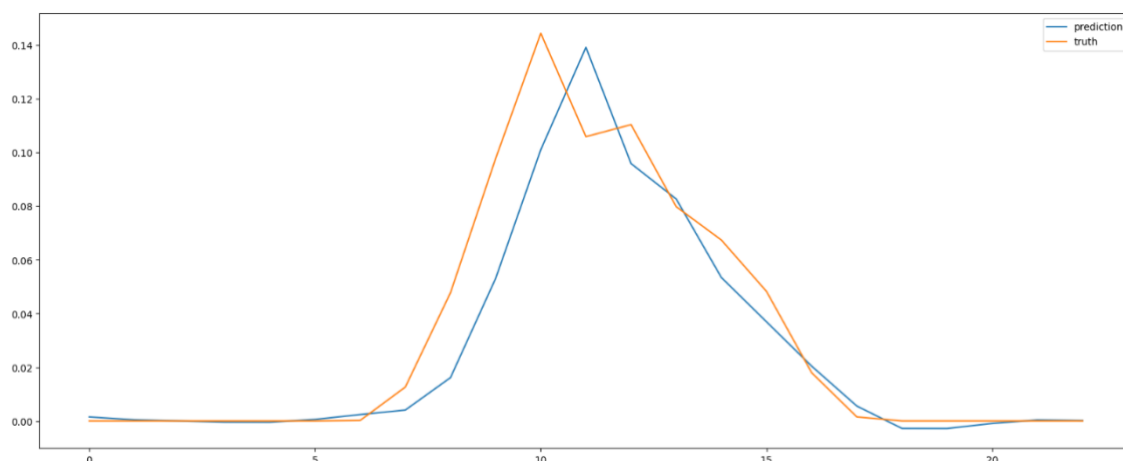


图 0-12 Patch-LSTM 模型短期预测结果局部放大

如图 4-12 所示：橘色线为真实值，蓝色线为预测值。由图可知预测滞后性减弱，同时在凌晨和夜晚的跟随较好，基本符合实际。其中误差统计如下表 4-4 所示：

MSE	0.00028879829915240407
RMSE	0.016994066586676775
MAE	0.00970529392361641

表 0-4 Patch-LSTM 预测结果误差统计

指标对比图为：

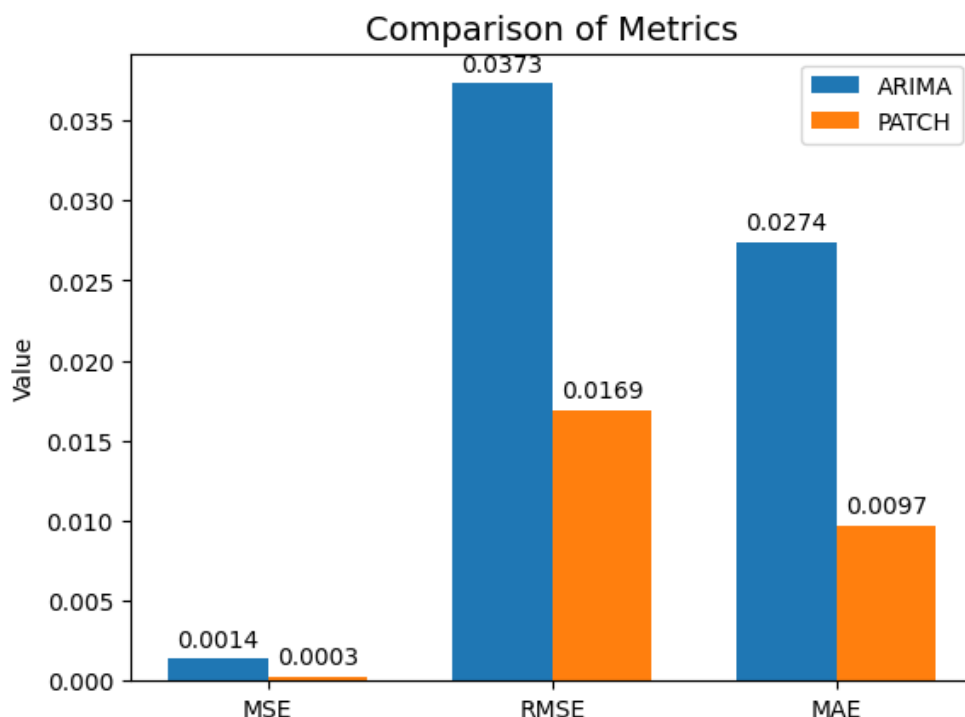


图 0-13 Patch-LSTM 和 ARIMA 模型短期预测指标对比图

由图 4-13 可知：

对于 MSE 和 MAE 指标，Patch-LSTM 模型的表现优于 ARIMA 模型。具体而言，PATCH 模型的 MSE 值为 0.00028，远小于 ARIMA 模型的 0.00139；Patch-LSTM 模型的 MAE 值为 0.0097，也小于 ARIMA 模型的 0.0274。在 RMSE 指标上，Patch-LSTM 模型的表现也好于 ARIMA 模型，ARIMA 模型的 RMSE 值为 0.0373，稍微小于 PATCH 模型的 0.0169。

## 4.4 发电量长期预测

### 4.4.1 实验介绍

与短期预测不同，ARIMA 模型进行长期预测时，由于会产生误差累计，进行长期预测时，结果会十分不准确。为了证明本文提出的 Patch-LSTM 方法在长期预



测方面仍具有有效性，在进行长期预测时与主流时序预测模型模型进行对比，分别是：MLP，DLinear 模型。

#### 4.4.2 数据集划分

在进行长期预测时，与 ARIMA 不同的是，为了使神经网络有效学习时序信息，需要扩大训练数据的数量，而不是像 ARIMA 方法那样仅适用预测前近期的数据。所以在预测时，对数据集进行了新的划分，将数据集分为训练集，验证集，测试集，比例为 6:3:1，划分结果如图 4-14 所示：

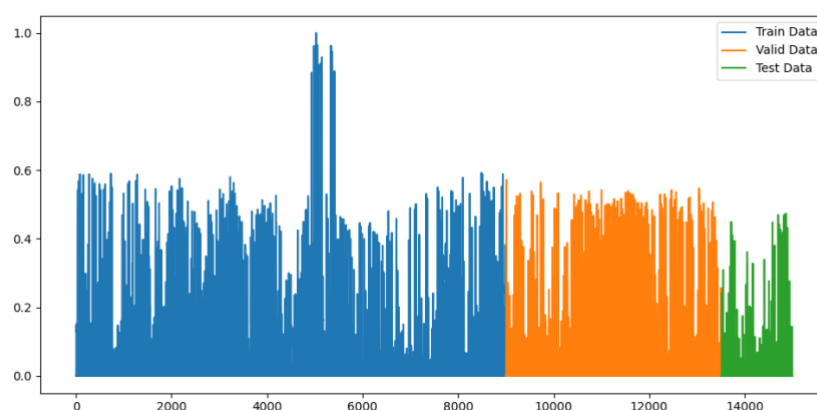


图 0-14 发电量长期预测数据集分割图

如图 4-14 所示：数据集被以 6:3:1 的比例分为训练集，验证集和测试集，分别由蓝色，橘色和绿色部分表示。

此外，在长期预测中，选择使用前两天的历史数据（48 小时）来预测后一天（24 小时）。

#### 4.4.3 MLP

<sup>[18]</sup>MLP（多层感知器）是一种前馈神经网络，由一个或多个中间层（隐藏层）和一个输出层组成。每个隐藏层和输出层都由多个神经元组成，每个神经元都与前一层的所有神经元连接，并具有可学习的权重和偏差。结构如下图 4-15：

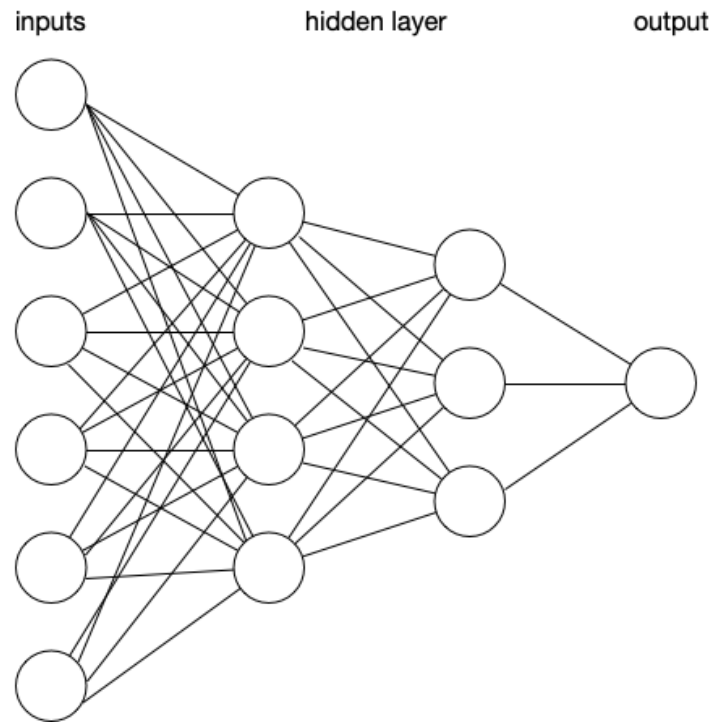


图 0-15 MLP 神经网络结构图

MLP 是一种简单且容易训练的神经网络，在本次实验中，共有两层 Linear 单元，神经元数为 256，优化器为 Adam，损失函数为 MSE。

使用 MLP 预测未来 24 小时结果图 4-16，4-17 如下：

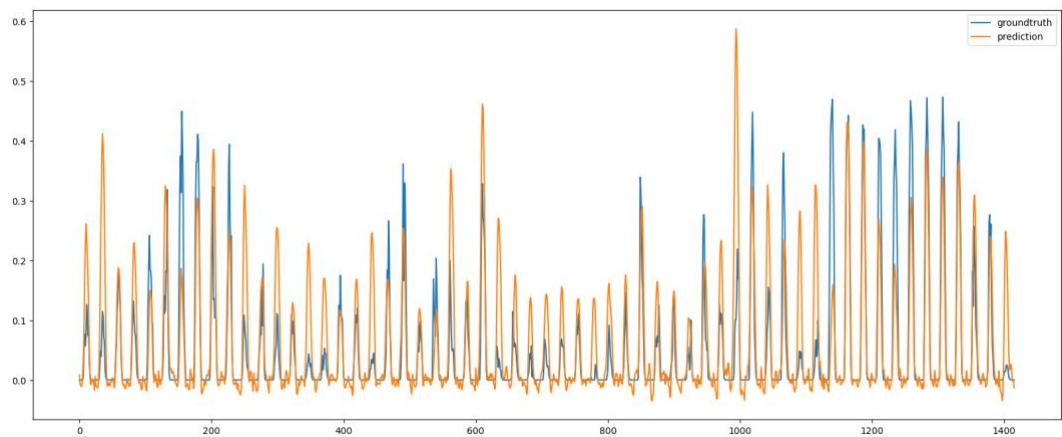


图 0-16 MLP 神经网络 24 小时预测图

局部放大图 4-17 为：

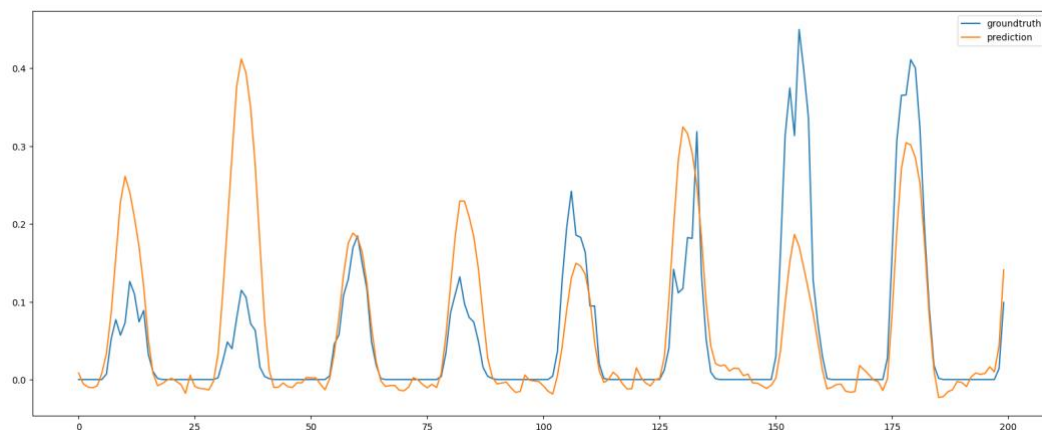


图 0-17 MLP 神经网络 24 小时预测局部放大图

如图所示：MLP 模型学到了每天发电量的时序特征，但是预测的不够精细和准确，在凌晨和深夜附近波动较大，峰值预测不够准确。具体数置指标为表 4-5：

MSE	0.005282088648527861
RMSE	0.07267797911697779
MAE	0.03936387971043587
MAPE	1.4119446

表 0-5 MLP 预测结果误差统计

#### 4.4.4 DLinear

随着神经网络在时间序列的进一步研究和 Transformer 模型的爆火，越来越多的研究开始集中在使用类 Transformer 模型和 Attention 机制来进行时间序列预测，比较卓越的工作有<sup>[20]</sup>Informer, <sup>[21]</sup>Autoformer, <sup>[22]</sup>ETSformer, <sup>[23]</sup>FEDformer。Patch-LSTM 模型，<sup>[24]</sup>但是最新研究表明，Transformer 模型或者 Attention 机制，对于时间序列的信息学习能力不足，甚至不如最简单的 MLP 模型，作者根据简单的时序知识：由于时间序列可以分解为趋势项，季节项和残差项；可以通过分解趋势项和季节项并同时用 Linear 层预测，最后进行合并得到输出。网络结构图 4-18 为：

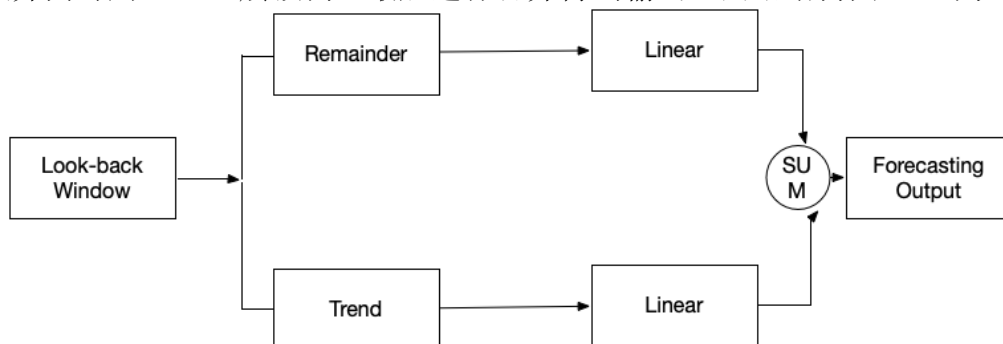


图 0-18 DLinear 神经网络结构图

DLinear 预测未来 24 小时的预测图 4-19 为：

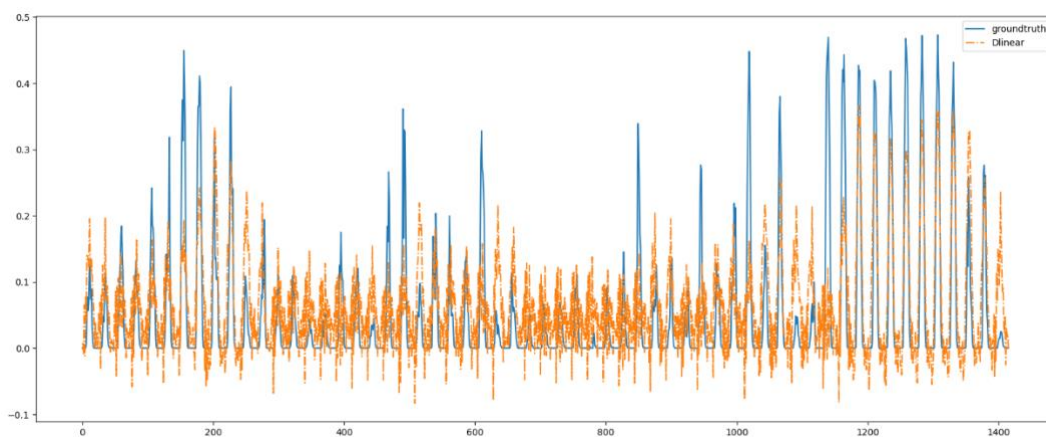


图 0-19 DLinear 神经网络 24 小时预测图

局部放大图为：

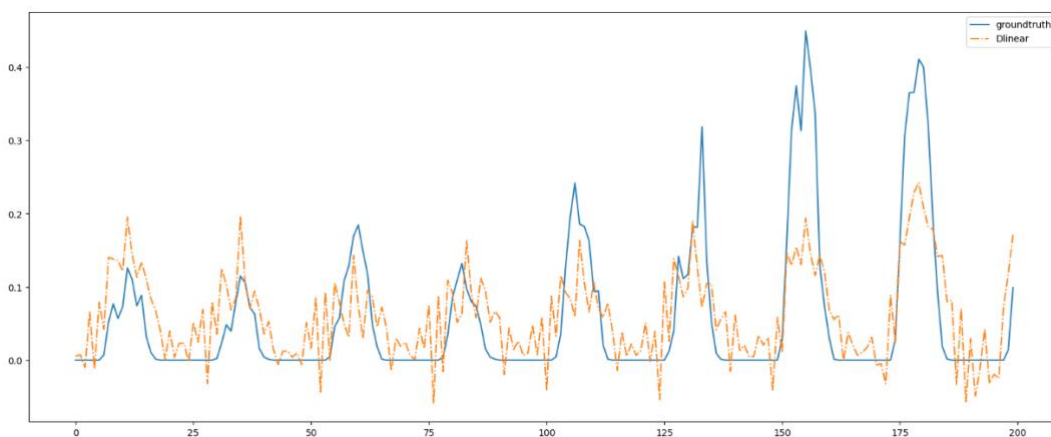


图 0-20 DLinear 神经网络 24 小时预测局部放大图

由图 4-20 可知：DLinear 学习到了每天发电量的基本时序结构，但是预测不平滑，毛刺过多，峰值基本稳定，不能快速预测每天的峰值涨幅。具体数字指标如表 4-6 所示：

MSE	0.005549801047891378
RMSE	0.07449698683766598
MAE	0.0542387031018734
MAPE	1.5561291

表 0-6 DLinear 预测结果误差统计

#### 4.4.5 Patch-LSTM

<sup>[25]</sup>LSTM 相比于 MLP，在时间序列预测中：首先善于处理长期依赖性，LSTM 的主要特点是可以处理长期依赖性，即在序列中较远的时间步长的信息仍然可以影响当前的预测结果。这是由 LSTM 的内部结构所决定的，它具有三个门控单元，能够选择性地记住或忘记一些历史信息，从而使网络可以适应长期依赖性的问题。其次对缺失数据具有鲁棒性，在时间序列预测中，经常会遇到缺失数据的情况，这时候传统的 MLP 往往无法处理。LSTM 可以通过控制门控单元的输出，来过滤掉

缺失数据，从而保持模型的连续性和稳定性。并且具有更好的建模能力：LSTM 通过自适应地学习时间序列的模式，能够更好地拟合非线性、非平稳的时间序列数据，从而获得更高的预测精度。而传统的 MLP 往往无法捕捉时间序列的这些特点，容易产生过拟合或欠拟合的问题。

Patch 的思想来自<sup>[19]</sup>提出的 Patch，这篇文章中讲一段长的时间序列进行切片，分别使用不同的注意力来获取时间信息。但是在<sup>[24]</sup>中提到，Transformer 类模型对于时间序列的信息抓取能力并不强，甚至弱于 MLP。在此基础上，结合本数据集的特点，我将一段长的时间序列，进行了按 24 小时长度的切片，并且按照相同的时间点进行重新排列，示意图 4-21 如下：

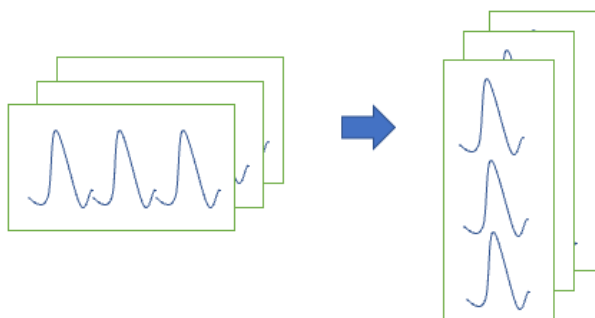


图 0-21 Patch-Layer 示意图

这样当数据输入到 LSTM 模型中，模型可以从不同天的相同时刻学习到数据的时序模式，从而做出精确的预测。

使用 Patch-LSTM 预测未来 24 小时的发电量结果图如下图 4-22，4-23 所示：

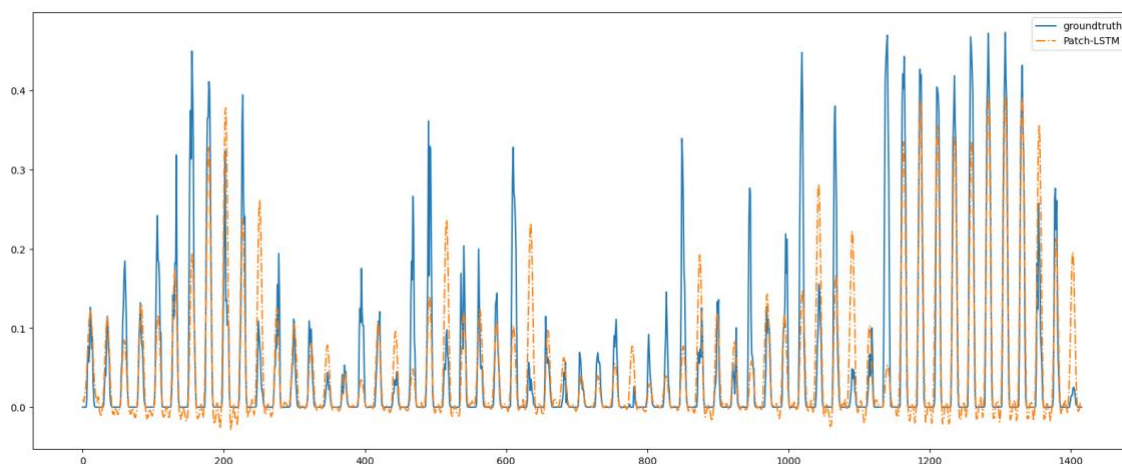


图 0-22 Patch-LSTM 发电量长期预测图

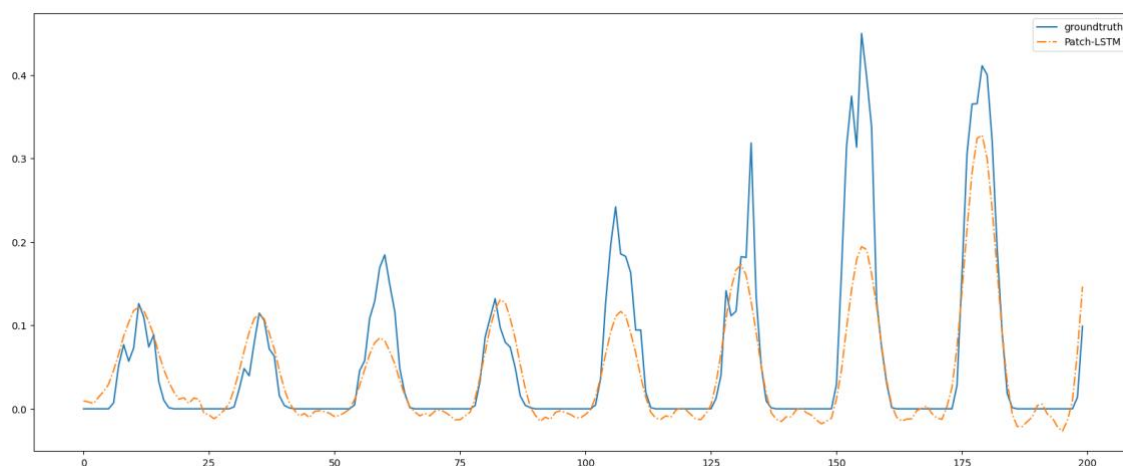


图 0-23 Patch-LSTM 发电量长期预测局部放大图

由图可知：Patch-LSTM 很好的学习到了发电量时序数据的特征信息，零点出波动不大，且峰值预测比起 MLP，DLinear 模型具有很大的改善。数字指标指标为表 4-7：

MSE	0.003447531256824732
RMSE	0.05871568152397392
MAE	0.0294288732111454
MAPE	0.8023116

表 0-7 Patch-LSTM 预测结果误差统计

#### 4.4.6 长期预测结果分析

比较 MLP，DLinear，Patch-LSTM 模型的 MSE，RMSE，MAE，MAPE，如下图 4-24，4-25 所示：

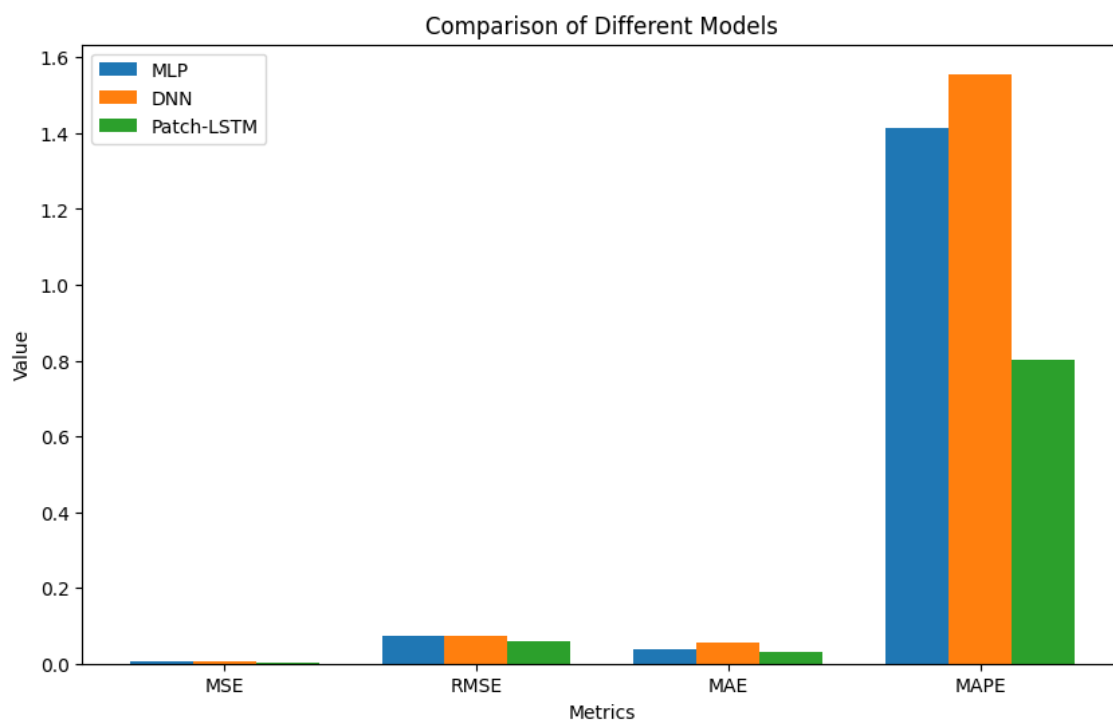


图 0-24 模型指标对比图

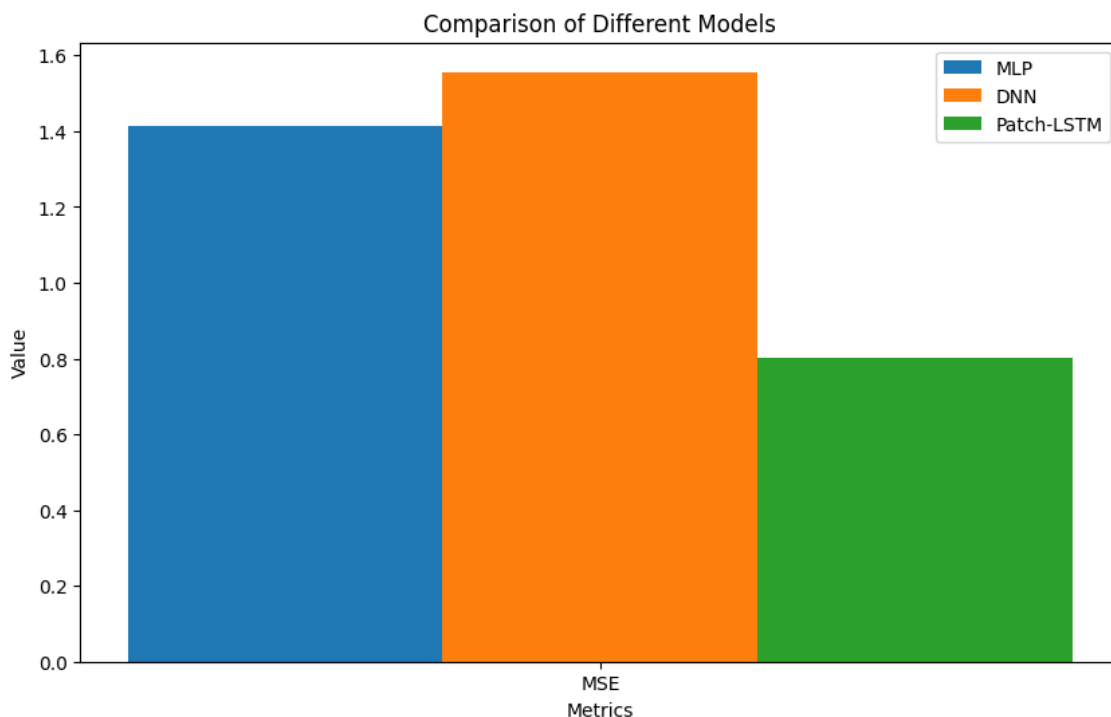


图 0-25 模型 MSE 对比图

由图 4-24,图 4-25 可知,根据给出的数据,分别对 MLP、DLinear 和 Patch LSTM 模型在 MSE、RMSE、MAE 和 MAPE 四个指标下进行了比较。结果如下:

**MSE (均方误差):** Patch LSTM 模型表现最好,为 0.0034,其次是 MLP 和 DLinear 模型,分别为 0.0053 和 0.0055。说明 Patch LSTM 在预测值与实际值之间的误差方差最小。

**RMSE (均方根误差):** Patch LSTM 模型表现最好,为 0.0587,其次是 MLP 和 DLinear 模型,分别为 0.0727 和 0.0745。说明 Patch LSTM 在预测值与实际值之间的误差方差开根号后最小。

**MAE (平均绝对误差):** Patch LSTM 模型表现最好,为 0.0294,其次是 DLinear 模型和 MLP,分别为 0.0542 和 0.0394。说明 Patch LSTM 在预测值与实际值之间的误差绝对值的平均值最小。

**MAPE (平均绝对误差百分比):** Patch LSTM 模型表现最好,为 0.8023,其次是 MLP 和 DLinear 模型,分别为 1.4119 和 1.5561。说明 Patch LSTM 在预测值与实际值之间的误差绝对值的平均百分比最小。

综上所述, Patch LSTM 模型在时间序列预测中表现最好,其次是 MLP 和 DLinear 模型。其中 Patch LSTM 模型在 MAPE 指标下表现最好,其 MAPE 值最小,说明其对于预测值与实际值之间的误差百分比控制能力最强,对于实际应用中的预测具有更高的可靠性。

## 4.5 本章小结

本章重点讲述的是整个实验的流程，本文提出了 Patch-LSTM 深度学习模型，他融合了 LSTM 在时间序列预测方面的优秀特性和 Patch 技术在增强模型对于时间序列学习能力的提升，并与其他模型进行对比来突出它的有效性。在开始说明了本次实验使用的数据集和数据处理的方法。正式的预测实验分为两大类：短期预测（1 小时预测）和长期预测（24 小时预测）。在短期预测中，将 Patch-LSTM 与传统预测模型 ARIMA 进行对比，在预测准确性和稳定性方面都有了较大的提高，在相对误差和绝对误差都有下降。在长期预测中，将 Patch-LSTM 与经典的深度学习模型 MLP，和最新时序研究成果 DLinear 进行了对比，在长期预测方面体现出较大的提升，对于实际应用中的预测具有更高的可靠性。



## 5 总结

本研究旨在通过应用时间序列分析和深度学习技术，提高光伏发电量预测的准确性和可靠性。随着全球变暖问题的不断加剧，各国对可再生能源的发展越来越重视。太阳能作为一种环保且广泛可用的可再生能源，已经成为许多国家的重要能源来源之一。光伏发电作为太阳能的一种主要形式，由于其趋势和周期性特征，其发电量的准确预测对于电力系统的稳定性和普及推广具有重要意义。本研究旨在探索时间序列分析和深度学习技术在光伏发电量预测中的应用，以提高预测的准确性和可靠性，为可再生能源的普及和工业应用做出重要贡献。

首先，本文介绍了时间序列，时间序列预测的定义和相关数学表达，对后续的研究实验提供坚实的理论的基础。接着，从历史发展的角度，引出了深度学习和神经网络，介绍了神经元的数学表达，和经典神经网络训练技巧批正则化和反批正则化。最后介绍了两类模型评价指标，绝对误差：MSE，MAE；相对误差：RMSE，MAPE，使得后续实验评价多样，有效，可靠。

接着，本文对后续实验中的一些组件如 AR，MA 模型，MLP，LSTM 神经元进行了详细的介绍和数学推导。介绍了组件的定义，作用，和在时间序列预测中的作用。

然后，本文在实际数据集上进行了预测实验，首先介绍了数据集，保证其真实可靠性。和数据预处理方法，数据聚合使基站每天的分散数据能够整合成可供深度学习训练的数据，数据重采样保证了数据颗粒度一致，使得数据更加平滑，降低了噪声。数据归一化使得模型更容易训练和收敛。接下来进行了短期预测实验来对比 Patch-LSTM 和 ARIMA 模型。对于 ARIMA 模型，使用时间滑动窗口技术，依次取出训练数据和预测数据，采用网格搜索法，确定了最佳的 ARIMA 参数，最后进行预测，并对预测结果做出量化评价。与 ARIMA 模型不同，深度学习模型需要大量的训练数据和更多特征来学习其中的模式和规律，绘制了 Patch-LSTM 的短期预测图，并进行指标计算。随后，进行了长期预测，将本文提出的 Patch-LSTM 模型与经典模型 MLP 和最新时序研究成果 DLinear 进行对比。在长期预测中，首先将数据集按照训练集，验证集和测试集划分，比例为 6: 3: 1。然后详细介绍了每一种模型的网络架构，和超参数设置，经历了长期预测实验，分别绘制和全局预测图和局部预测图，更详细展示了每一种的模型的预测精度。最后对比分析了三种模型的指标对比图，与短期预测不同，根据实际应用场景，在长期预测中新增了 24 小时总和的 MAPE，更直观准确的反应了模型的预测能力。

综上，在短期预测中，Patch-LSTM 相比于 ARIMA 模型，在数据层面上，一是有着更大的数据吞吐量，二是具有结合外部驱动特征辅助预测的能力。在预测层

面，Patch-LSTM 预测更加准确，尤其在凌晨，深夜和峰值的滞后性上有了有大的改善；在长期预测中，Patch-LSTM 相比 MLP 和 DLinear 模型，在 MSE，MAE，RMSE，MAPE 所有指标均是最佳，并且模型十分稳定，在 0 值，峰值的精确度，以及不同天的涨落的适应速度都有明显的提升。说明其对于预测值与实际值之间的误差百分比控制能力最强，对于实际应用中的预测具有更高的可靠性。但是预测图中也反映出模型对于峰值的预测仍不够准确，对于变化的跟随预测不够灵敏，需要后续更为强大的时序模型解决。总之，本文为了后续关于如何使用少量特征进行较为准确的预测做了大量有效的研究。

## 致谢

在这篇论文的完成过程中，我要感谢许多人的支持和帮助。

首先，我要感谢我的导师刘春山老师，无论是平时的科研指导还是毕业论文的选题，开题答辩，中期检测，最后的论文撰写都给予了无私的指导和建议，刘老师的睿智，博学，谦和使得无论研究中遇到什么困难，都能帮助我冷静应对，找到最佳的解决办法。激励和鼓励着我一步步向前。

我还要感谢我的家人，在我完成这篇论文的过程中给予了我无限的鼓励和支持。他们的爱和支持是我不断前进的动力和源泉。

此外，我还要感谢实验室的师兄师姐，在宏观层面，论文阅读和写作，在细节方面，代码的实现和改进，他们都给予了细致的指导，使得我的科研写作和实验能力有了巨大的进步，没有他们的帮助和合作，我的研究无法顺利完成。

最后，我要感谢所有曾经为这篇论文提供过帮助的人，无论是通过提供数据、文献、建议还是其他形式的支持，他们的贡献都是不可或缺的。

在这里，我要向以上所有人表达我最诚挚的感谢和敬意。

## 参考文献

- [1] 张玉, 张振涛, 莫寒. 光伏发电量的预测综述[J]. 制造业自动化, 2015, 37(21): 27–30.
- [2] Ahmed R, Sreeram V, Mishra Y, 等. A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization[J]. Renewable and Sustainable Energy Reviews, 2020, 124: 109792.
- [3] 罗芳琼, 吴春梅. 时间序列分析的理论与应用综述[J]. 柳州师专学报, 2009, 24(3): 113–117.
- [4] 郑元庄. 非线性系统时间序列分析方法的研究[D]. 集美大学, 2014.
- [5] 卢静, 翟海清, 冯双磊, 等. 光伏发电功率预测方法的探索[J]. 华东电力, 2013, 41(2): 380–384.
- [6] 仇琦, 杨兰, 丁旭, 等. 基于改进 EMD-ARIMA 的光伏发电系统短期功率预测[J]. 电力科学与工程, 2020, 36(8): 42–50.
- [7] Elsayed S, Thyssens D, Rashed A, 等. Do We Really Need Deep Learning Models for Time Series Forecasting?[J]. arXiv, 2021.
- [8] Wu Y-K, Huang C-L, Phan Q-T, 等. Completed Review of Various Solar Power Forecasting Techniques Considering Different Viewpoints[J]. Energies, 2022, 15(9): 3320.
- [9] Convolutional Neural Network | SpringerLink[EB/OL]. /2023-02-24. [https://link.springer.com/chapter/10.1007/978-1-4842-2845-6\\_6](https://link.springer.com/chapter/10.1007/978-1-4842-2845-6_6).
- [10] 余潇智, 杨静, 李兴超, 等. 基于卷积神经网络的目标检测算法综述[J]. 电子技术与软件工程, 2022(24): 171–176.
- [11] Survey on Research of RNN-Based Spatio-Temporal Sequence Prediction Algorithms - ProQuest[EB/OL]. /2023-05-07. <https://www.proquest.com/openview/9ebe553918e3e43d67209a82d3243534/1?pq-origsite=gscholar&cbl=4585453>.
- [12] Yu Y, Si X, Hu C, 等. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures[J]. Neural Computation, 2019, 31(7): 1235–1270.
- [13] Vaswani A, Shazeer N, Parmar N, 等. Attention Is All You Need[J]. arXiv, 2017.
- [14] 牛宇童, 冯天波, 李庆, 等. 基于生成对抗网络的小样本光伏发电短期预测[J]. 电源技术, 2022, 46(11): 1325–1329.
- [15] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[A]. Proceedings of the 32nd International Conference on Machine Learning[C]. PMLR, 2015: 448–456.
- [16] Hu F, Chen A A, Horng H, 等. Image harmonization: A review of statistical and deep learning methods for removing batch effects and evaluation metrics for effective harmonization[J]. NeuroImage, 2023, 274: 120125.
- [17] Newbold P. ARIMA model building and the time series analysis approach to forecasting[J]. Journal of Forecasting, 1983, 2(1): 23–35.
- [18] Peng Z. Multilayer Perceptron Algebra[J]. arXiv, 2017.
- [19] Nie Y, Nguyen N H, Sinthong P, 等. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers[J]. arXiv, 2022.
- [20] Zhou H, Zhang S, Peng J, 等. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[J]. arXiv, 2021.
- [21] Wu H, Xu J, Wang J, 等. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[J]. arXiv, 2022.

- [22] Woo G, Liu C, Sahoo D, 等. ETSformer: Exponential Smoothing Transformers for Time-series Forecasting[J]. arXiv, 2022.
- [23] Zhou T, Ma Z, Wen Q, 等. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting[J]. .
- [24] Zeng A, Chen M, Zhang L, 等. Are Transformers Effective for Time Series Forecasting?[J]. arXiv, 2022.
- [25] Photovoltaic power forecasting based LSTM-Convolutional Network - ScienceDirect[EB/OL]. /2023-02-13.  
<https://www.sciencedirect.com/science/article/pii/S0360544219319206>.