南京大学

研究生毕业论文
(申请硕士学位)

论文题目 基于前馈神经网络的
时间序列预测问题研究

作者姓名 郝鸿延

学科、专业名称 计算机技术

研究方向 人工智能

指导教师 申富饶 教授，宋方敏 教授

2021 年 5 月 29 日

学　　号：MF1833023

论文答辩日期：2021 年 5 月 20 日

指 导 教 师：  　　**（签字）**

Time Series Forecasting based on Feed-forward Neural Networks

by

Hao Hong-Yan

Supervised by

Professor Shen Fu-Rao, Professor Song Fang-Min

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Technology

Department of Computer Science and Technology
Nanjing University

May 20, 2021

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于前馈神经网络的时间序列预测问题研究

计算机技术 专业 2018 级硕士生姓名：郝鸿延

指导教师(姓名、职称)：申富饶 教授，宋方敏 教授

摘　要

时间序列预测问题的研究至关重要，从电商销量预测到股票价格预测等，随处可见其应用场景。时间序列预测任务也是学术界长久以来的研究重点，随着数据形式的复杂化，模型也从最初基于序列稳定性假设的传统统计学方法发展到可以处理非平稳数据的机器学习方法，然而序列特征表示和损失函数的设计等问题仍然是时间序列预测面临的挑战。

本文针对时间序列预测中的特征表示问题，结合现在神经网络方法的优势，利用时序卷积模型、注意力机制和残差结构设计了一种能够具有循环网络建模序列相关性能力的网络结构，即前馈序列模型（**F**eed-forward **S**equential **N**etwork，FSN），此模型旨在模拟循环网络对序列化属性的建模方式，同时克服循环网络梯度消失、训练效率受限和长时记忆衰减的缺点，我们利用时序卷积网络和时序注意力机制达到了对序列化属性的表征，利用残差结构让网络能够加深的同时防止网络退化。我们从信息流动和在标准数据集实验两方面对模型的有效性进行验证，实验结果表明，FSN 能够提升时间序列预测准确度，对不同长度输入序列的敏感性分析实验还证明了 FSN 能够有效控制训练效率，防止出现循环网络中随着输入长度增加而训练效率降低的问题。

另外，本文还从损失函数的角度对时间序列预测模型进行优化，现有常用损失函数未能在训练过程中重点关注序列的形态学习和延时性问题，这导致模型的预测结果出现未能有效预测波动或未能准确预测波动产生时间的现象，而这对于很多实际场景是至关重要的。我们从形态学习和延时性两方面出发，结合 DTW 度量方法设计了多尺度 DTW 和 TDI 结合的损失函数 MS-DTWI （**M**ulti-**S**cale **DTW** with Temporal Distortion **I**ndex），我们将 MS-DTWI 与其他时间序列预测损失函数做对比，用实验验证了此损失函数的有效性和更好的学习效果，同

时为了验证损失函数中超参数对训练的影响，我们设计了敏感性分析实验，给出对 MS-DTWI 更全面的分析。

最后，为解决大型连锁企业对商品的管理和销量预测问题，我们设计并实现了“金牛（Taurus）系统”，系统不仅能够进行库存管理、商店管理和交易管理，还可以对商品做进一步的关联分析，从流动的商品销售数据中挖掘更多有价值的信息，同时利用历史商品的销售数据和已知可用的时间日期特征对未来商品销售数量做出预测。其中预测环节不仅集成了传统统计学算法，还采用了上述提出的基于前馈序列模型的时间序列预测算法，让系统能够适应不同数据量场景的需求，系统对关联分析和销量预测能力都做了可视化展示，让用户更直观地感受系统智能。

关键词： 时间序列预测；时序卷积网络；注意力机制；DTW；销量预测

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Time Series Forecasting based on Feed-forward Neural Networks

SPECIALIZATION: Computer Technology

POSTGRADUATE: Hao Hong-Yan

MENTOR: Professor Shen Fu-Rao, Professor Song Fang-Min

ABSTRACT

The research on time series forecasting is very important, and its application scenarios can be seen everywhere from e-commerce sales forecasting to stock price forecasting. Time series prediction tasks have also been a long-term research focus in the academic community. With the increasing of complexity of the data format, the model has evolved from a statistical methods based on stable series assumptions to machine learning methods that can handle non-stationary data. However, the challenges like feature extraction and the design of loss functions still exist.

We aim at resolving feature extraction problem in time series forecasting by combining with the advantages of neural networks. We use temporal convolutional network, attention mechanism and residual structure to design a network structure called **F**eed-forward **S**equential **N**etwork (**FSN**) capable of simulating recurrent neural network modeling sequential correlation. This model is designed to combine the recurrent network's ability to model the sequential characteristic, and overcoming the shortcomings of the recurrent neural network such as gradient vanishing, limited training efficiency, and long-term memory decay. We use sequential convolutional network and the sequential attention mechanism to achieve the representation of sequential characteristic, and use residual structure to prevent the model from network degradation. We verify the effectiveness of the model from theories and experiments on standard data sets. The experimental results show that FSN can improve the accuracy of time series forecasting. The sensitivity analysis experiments on input sequences of different lengths prove that FSN can effectively control training efficiency and prevent the problem of reduced

training efficiency as the input length increases in the recurrent neural network.

In addition, we also optimize the time series prediction model from the perspective of the loss function. The existing commonly used loss functions fail to focus on the morphological learning and delay between the predicted sequence with the real one, which leads to the deviations of the model's prediction results. The fluctuations and its occurrence time are crucial for many practical scenarios. For these two aspects, we combine DTW and TDI metrics to design a new loss function called **M**ulti-**S**cale **DTW** with Temporal Distortion **I**ndex (**MS-DTWI**), we compare MS-DTWI with the commonly used time series prediction loss functions, and use experiments to verify the effectiveness of MS-DTWI. At the same time, in order to verify the effects of hyper-parameters in the loss function on training, we design sensitivity analysis experiments, which makes the analysis of MS-DTWI more comprehensive.

In order to solve the problem of commodity management and sales forecast for large-scale chain enterprises, we design and implement the "Taurus system". The system can not only carry out inventory management, store management and transaction management, but also perform further correlation analysis of commodities. It can excavate more valuable information from product sales data and use historical product sales data and known available time and date features to predict future product sales. The prediction module not only integrates traditional statistical algorithms, but also the above-mentioned time series forecasting algorithm based on the FSN, they are used to enable the system to adapt to the needs of different data volumes. The system's visualization module displays the correlation analysis and sales forecast capabilities, allowing users to have more intuitively experience on system's intelligence.

KEYWORDS: Time Series Forecasting; Temporal Convolutional Network; Attention Mechanism, DTW, Sales Forecast

目　　录

中文摘要 ····· **i**

英文摘要 ····· **iii**

目　　录 ····· **v**

插图清单 ····· **vii**

附表清单 ····· **ix**

1　绪论 ····· **1**

1.1　研究背景及意义 ····· 1

1.2　研究现状及挑战 ····· 3

1.3　研究内容 ····· 6

1.4　本文组织结构 ····· 7

2　相关工作 ····· **9**

2.1　基于统计学的方法 ····· 12

2.1.1　整合自回归移动平均法 ····· 12

2.1.2　指数平滑法 ····· 14

2.2　基于机器学习的方法 ····· 15

2.2.1　支持向量回归 ····· 15

2.2.2　XGBoost ····· 17

2.3　基于神经网络的方法 ····· 18

2.3.1　传统神经网络 ····· 19

2.3.2　深度神经网络 ····· 20

2.4　本章小结 ····· 22

3　基于卷积的前馈序列网络 FSN ····· **23**

3.1　模型设计思路 ····· 23

3.2　前馈序列网络结构 ····· 26

3.2.1　时序卷积 ····· 27

3.2.2　注意力机制 ····· 30

3.2.3　前馈序列网络 ····· 31

3.3　实验与分析 ····· 36

3.3.1 对比实验 …… 38
3.3.2 消融实验 …… 40
3.3.3 敏感性分析 …… 42
3.4 本章小结 …… 43

4 基于 DTW 的 MS-DTWI 损失函数 …… 45
4.1 现有损失函数的局限性 …… 45
4.2 MS-DTWI 损失函数 …… 47
4.2.1 DTW …… 47
4.2.2 MS-DTW …… 50
4.2.3 TDI …… 53
4.2.4 MS-DTWI 损失函数 …… 54
4.3 实验与分析 …… 55
4.3.1 对比试验 …… 55
4.3.2 敏感性分析 …… 57
4.4 本章小结 …… 60

5 商品管理和销量预测系统 …… 61
5.1 相关背景 …… 61
5.2 系统目标 …… 62
5.3 系统功能 …… 62
5.4 系统架构 …… 63
5.5 系统实现 …… 65
5.5.1 商品管理 …… 65
5.5.2 分析和预测 …… 66
5.6 本章小结 …… 68

6 总结与展望 …… 69

参考文献 …… 71

简历与科研成果 …… 79

致 谢 …… 81

学位论文出版授权书 …… 83

插图清单

1-1 时间序列预测应用场景 ····· 3

2-1 时间序列预测任务分析 ····· 9
2-2 两种多步预测方法 ····· 11
2-3 支持向量回归 ····· 16
2-4 感知机图示 ····· 19
2-5 单层和多层前馈网络 ····· 20
2-6 RNN 网络结构 ····· 21

3-1 因果卷积网络[1] ····· 27
3-2 扩展卷积网络[1] ····· 28
3-3 时序卷积网络 ····· 29
3-4 注意力机制示意图[2] ····· 30
3-5 Self-Attention 结构 ····· 31
3-6 前馈序列网络结构 ····· 32
3-7 网络结构单元 ····· 32
3-8 时序注意力模块结构 ····· 34
3-9 强化残差模块结构 ····· 36
3-10 各种网络结构信息流向 ····· 37
3-11 FSN, LSTM 和 TCN 的 ND 指标变化 ····· 40
3-12 消融实验结果序列图 ····· 41
3-13 不同输入长度模型耗时比较 ····· 42

4-1 真实序列 A 与预测序列 B、C 的曲线 ····· 46
4-2 真实序列 D 与预测序列 E、F 和 G 的曲线 ····· 46
4-3 D 和 F 的序列点对应关系 ····· 47
4-4 D 和 F 的序列点对应矩阵 ····· 48
4-5 不同规整窗口下序列对应关系 ····· 50
4-6 序列 D 与序列 E、F 和 G 的对应关系矩阵 ····· 53
4-7 不同损失函数效果对比 ····· 56
4-8 不同 γ 时平滑最小化函数计算结果 ····· 57
4-9 不同 γ 时模型效果 ····· 58
4-10 不同尺度系数 n 时效果对比 ····· 59
4-11 不同 α 时效果对比 ····· 60

5-1 整体系统架构图 …… 64
5-2 系统主界面 …… 65
5-3 库存管理 …… 66
5-4 关联分析 …… 66

附表清单

2-1 ARIMA(p,d,q) 法和其他经典模型的关系 ····· 13

3-1 数据集统计信息 ····· 38
3-2 FSN 网络和其他方法对比结果 ····· 39
3-3 消融实验结果 ····· 41

4-1 序列 D 与 E、F 和 G 的三种度量指标 ····· 54
4-2 不同损失函数实验对比结果 ····· 55
4-3 不同尺度系数 n 时效果对比值 ····· 58
4-4 不同 α 时效果对比值 ····· 59

5-1 不同时序预测方法对比结果 ····· 67

第一章　绪论

1.1　研究背景及意义

日常生活中，时间序列数据每时每刻都在产生，大到整个地球的温度变化，小到我们每个人的心跳频次。时间序列数据记录着各个场景的动态属性变化过程，随着时间的推移，我们可以获取到历史时间序列，原理上时间无限延长，我们将获得无穷无尽的数据。在气象监测场景中，对某地区过去几年的温度，空气指数等指标进行记录，同时参考历史指数变化，预测未来一段时间的气候变化情况，就是我们日常使用的天气预报功能；在金融领域，股票、期货、数字货币等证券及衍生品时刻产生着交易行为，交易员可以利用历史交易数据，分析预测市场中各种指数的走势，从中抽取有效信息辅助投资决策，赚取收益。不同场景下的历史数据都可以为未来的事件发生提供有用的信息，使模型能够预测此场景下未来的某些数值的变化。

可以看出，时间序列预测在实际生活中发挥了重要作用。我们定义时间序列数据为在某维度上等间隔采样的数据点，时序数据是一种常见的数据形式，学术界有着很广泛的研究。统计学发展过程中，产生了多种时间序列预测的模型，但随着数据量的大幅增加，数据的分布越来越多样化，使具有严格假设条件的早期统计学模型的局限性暴露出来，不适用于解决当下大数据场景下的问题。

当下时间序列预测任务面临众多挑战。不同场景下的时间序列规律不同，有的序列具有明显的周期性，这种时间序列数据较易分析，强周期性大幅降低建模难度，然而更多的时间序列数据缺少直观的周期性，需要专家结合数据场景，对时序进行分解，剥离对序列的影响因素和趋势因素等，才可以观察到时间序列的周期性，分解的方法多种多样，分解后与分解前的处理方法各不相同。另外，伴随时间序列产生的各种协变量数据对时间序列的变化有着不可忽视的影响，这些问题导致对时间序列的分析和预测难度大大提高，如何建立有效的模型分析历史数据，如何充分利用协变量，以及如何设计策略函数引导机器学习模型训练。这些问题在不同发展阶段困扰着研究者，特别是当下大数据时代，传

统统计学方法逐渐失去解决大数据问题的能力，但机器学习模型和深度学习模型的出现让人们找到新的突破口。本文对传统统计方法进行分析，并总结当下机器学习和深度学习算法的设计优缺点，提出一种新的网络结构和策略函数，从分析和实验的角度验证模型的有效性。

时间序列预测问题的有效解决能够不仅能够为企业生产带来效益，还会让日常生活更加便利。因为时间序列预测问题涉及到我们工作生活的方方面面。例如运动手环产生的健康监测数据，每隔一段时间就对脉搏等指标进行一次采样；电厂对城市供应的电能也需要不断调整，供电系统利用历史耗电数据，预测各地区的电力消耗，从而优化供电方案；金融经济数据每时每刻都在产生，金融从业者通过对金融产品的历史走势判断不同市场的资金博弈情况，从而制定策略以在博弈中赚取收益。下面列举一些时间序列预测在实际场景中的应用案例。

企业成本的构成中，库存占比通常较大，如何有效的利用产能，减少库存带来的成本是很多企业面临的共同问题。准确的预测市场需求，使生产线按照需求进行生产，可以从根本上解决问题，这就需要利用市场的历史销售数据来预测未来一段时间的总销售量。近些年很多竞赛平台反映出很多实体产业公司有这种预测需求，比如 Kaggle 至今已经成功举办第五届的 M5 竞赛，每一次的赛题都是利用企业的真实数据，让全世界范围的参赛者设计时间序列预测模型，试图寻求最佳算法解决方案。实际的销量预测中不仅要利用历史销售量数据，还要解决不同区域的实体店销售能力差异，不同季节，不同种类的商品等因素带来的问题，这些挑战对于建立实际有效的模型至关重要[3]。

在银行或者金融科技企业中对资金流动性的预测需求很多，系统每时每刻都产生大量交易数据，历史不同季节，不同节假日的数据走势都有其规律可循，如果能够有效预测未来一段时间的资金需求，亏损或收益，便能够给决策者提供重要的决策依据，保障金融系统安全运转，实现长远利益最大化。比如在面向用户的申赎业务场景下，要准备足够的准备金供用户赎回，如果准备金不足，会造成兑付风险，用户无法赎回资金，对用户和公司都是很严重的损失。除此之外还有很多资金流动性场景，准确地预测资金量能够大幅降低成本，增加收益[4]。

建筑能源的有效利用已经被联合国能源署列为五大脱碳举措之一，“碳中和”也是国家的发展大计，能源的高效利用不仅有利于环境保护，而且会降低建筑成本，带来很大的经济效益。建筑能源的合理分配是提高能源使用效率的重

(a) 商品销量预测

(b) 资金流动性预测

(c) 能源消耗预测

图 1-1: 时间序列预测应用场景

要举措，但建筑往往是一个复杂的整体，很多因素影响着能源消耗量，例如天气条件、建筑材料属性和复杂的能源交互系统（例如照明），这些因素的存在给能源消耗量的有效预测带来极大困难。实际上述提到的很多变量都属于时间序列的协变量范畴，如今基于机器学习的预测方法可以有效解决此类问题，提高预测准确度，预测能力的提升直接为降低能源消耗带来帮助，产生积极的社会效益。[5]。

1.2 研究现状及挑战

时间序列预测是从实际生产生活中抽象出来的研究课题，对于预测的需求也多种多样，本节对时间序列预测问题的子问题进行梳理，明确本文的研究目标。

时间序列数据分为一维时间序列和多维时间序列，一维时间序列可以简单描述为在一维空间上连续等间隔采样的数值集合，多维时间序列为在多个维度上，连续等间隔采样的数值集合，在每个维度上，有多个采样数值。例如在测量室内温度的传感器上，每秒记录一个温度数值，这样每秒就会得到一个一维数据，对这个维度的数据做预测就是一维时间序列预测；如果某个集成传感器在同一时间点上能测量温度、湿度和气压三个环境变量，则每秒会得到三个数值，即一个三维数据，大于一维的时间序列我们称为多维时间序列，对多个维度的数值同时预测则是多维时间序列预测。

在预测过程中，是否借助历史时间点上其他相关数据会对模型产生关键影响。这里我们引入协变量的概念，协变量是指待预测时间序列值以外，随时间的推移，在每个时间点同时产生的数据，比如预测未来的温度时，同时产生的湿度、压强等变量，不同于上述多维预测，此时湿度和压强并非我们预测的目标，

而是作为辅助预测的特征，此时湿度和压强就是对温度做预测时的协变量。对于预测时考虑协变量的情况，我们称为协变量预测，反之只利用时间信息和待预测维度历史数据则称为自回归预测。当待预测的未来时间长度不同时，也分为两种不同的预测方法，即只预测接下来一个时间点的值称为单步预测，预测未来多个时间点的值称为多步预测。多步预测通常有两种策略，一种为一次性预测未来多个时间点数据，另一种为迭代预测，即递归地利用上一次预测值来预测下一个时间点的值，迭代多次。对于预测得到的结果，如果预测值为确切的数值，则属于点估计问题，称之为点预测，如果预测得到的是一个概率区间，即在不同的分位点上会有不同的预测值，则属于概率估计问题，称之为概率预测。以上给出了时间序列预测下多个子问题，此处明确本文研究内容为一维，基于协变量的，多步点预测。

解决时间序列预测问题一直是学术界研究的重点，起初主要是基于统计学的方法，这些方法通常认为一条时间序列可以分解为三种属性，即趋势性（Trend）、季节性（Seasonality）和噪音（Noise）。趋势性指长期来看序列是上升还是下降的的走向，季节性指序列中固定序列长度内的重复走势，噪音指序列中除去以上两部分外不规则的随机部分。对分解后的序列，再用统计学模型建模，比如经典的差分自回归移动平均法（Auto-Regressive Integrated Moving Average，ARIMA），它主要由三部分组成，即表征历史值对当前值影响的自回归项，使序列平稳化的差分项，和表征序列不同时间噪音的周期依赖性的移动平均项[6]。指数平滑（Exponential Smoothing，ETS）也是常用的建模时间序列统计学算法，它的主要思想为距离当前时间点越近的数据，对预测影响越大，越远的数据影响越小，随着序列点距离越远，参数指数项越小，从而表示影响程度逐渐变小，另外指数平滑通过单独对趋势和季节进行参数化建模，整合并得到最后的预测结果[7]。

随着数据维度增加，对协变量的利用成为时间序列预测不可忽视的挑战。然而统计学方法通常用于建模一维时间序列，虽然有一些方法能够利用协变量[8-9]，但它们在预测能力和可解释性上的成本比较大，相比之下，基于机器学习的方法可解释性和预测准确率都更高。基于机器学习的时间序列预测能够对包含协变量的时间序列进行建模，在建模中的特征工程环节可以整合协变量，用于提升机器学习模型的预测能力。常用的算法包含支持向量回归算法[10-11]、梯度提

升算法，以及如今流行的基于神经网络方法，其中研究较多的为神经网络中基于深度学习的一系列方法，本文将这些算法单独归为一类。

传统统计学算法侧重对时间序列的趋势性、季节性和噪音建模，机器学习算法侧重在特征工程环节挖掘样本特征对预测的作用，而基于深度学习的算法不仅具有上述两者的优势，同时还能整合样本不同维度的特征，以及序列数据特有的序列位置属性。时间序列数据与文本和音频数据有一些共同的性质，比如时序性、因果关系属性和高维性等。对于时序类数据的建模，在深度学习领域首先想到的是 RNN（Recurrent Neural Network）以及它的变种模型，比如 LSTM（Long Short-Term Memory）[12] 和 GRU（Gated Recurrent Unit）[13]，或者近几年提出的基于注意力机制的 Transformer[14] 模型，还有基于卷积神经网络（Convolutional Neural Network, CNN）的 TCN（Temporal Convolutional Network）[15] 的模型等，这些模型都是深度学习在发展过程中，研究者为解决不同问题时不断优化创新的结果。

近年来，基于机器学习和深度学习的预测算法逐渐成为业界应用的首选方案。计算机算力的提升和分布式计算的发展也给这些算法提供了实现和落地的土壤。这些算法相比传统算法有很多优势，比如可以利用大规模数据来建模、有效整合时序中协变量特征和实现对历史信息的记忆等，但是仍然面临一些有待解决或有优化空间的困难与挑战。

概念漂移问题存在于很多深度学习解决的任务中，在时间序列预测任务中它是指时间序列数据随着时间的推移，其分布会发生改变。如对于销售数据，不仅受季节、节假日和促销影响，还可能受宏观经济形势影响，所以其变化的规律很难学习，传统统计学算法更是无法解决，基于深度学习的算法可以设计相应模型整合复杂的协变量信息，但不同序列模型的特征提取效果各有不同。

对时序属性的建模也是难点之一，时间序列数据的形式和图片不同，图片中各个像素位置间相互独立，而时间序列数据则不同，时间序列中后产生的数据点受在它先前产生的数据点影响，即 t_i 和 t_{i-n} 是条件相关的，具有 n 阶马尔可夫性，阶数越高，模型复杂度越高，一些深度学习算法虽然考虑到这点，但对历史信息的记忆效果并不理想，存在优化的空间；

要想将模型落地，不得不考虑模型的性能问题，评价模型优劣的指标不只是预测的精度，还要考虑到训练和预测时消耗的时间和计算资源因素，典型的

RNN 类模型随着历史时间步的增加，训练所消耗的时间大幅增加，这是递归模型无法避免的缺陷，而前馈模型相比之下优势更加明显；

损失函数作为引导模型训练的指挥棒，一定程度上决定着模型的泛化性能。当前在时间序列预测任务上广泛应用的损失函数无法有效处理预测序列与真实序列的波动形态偏差和延时性偏差，造成预测结果准确度降低，对于一些对波动和延时性要求高的场景，无法有效应用。

以上提到的困难与挑战是很多模型的设计需要考虑的因素，基于统计学和机器学习的算法不能有效解决，当下各种基于深度学习的算法在其中一些问题上有所突破，但缺少一种综合有效解决上述问题的算法，本文分析和总结现有特征提取网络结构的优势和局限性，利用前馈网络结构设计一种神经网络模型，解决上述提到的挑战，另外分析现有时间序列预测中常用的损失函数，设计一种新的损失函数解决对形态和延时预测的偏差问题。

1.3 研究内容

本文对当下时间序列预测方法进行分析，从机器学习的三要素：模型、策略和解法三个角度[16] 刨析现有算法的局限性，参考相关理论方法克服挑战。这并不是一个由无到有的设计过程，而是站在前人肩膀上迭代优化的过程。RNN 是用于解决时序问题的常用模型，由于 RNN 的递归属性，使得它可以对不同长度的输入样本进行建模，RNN 的变体 LSTM 和 GRU 因其门结构一定程度上克服了 RNN 的梯度消失问题，但递归网络对于历史信息的建模效果并不理想，而且训练模型的时间也会随着历史时间步的增加而大幅增加，限制了模型在实际中的应用。对于损失函数，人们通常会选择 HuberLoss 或 MSELoss，但存在一定的局限性，我们针对上述问题设计新的时间序列预测模型，本文的贡献总结为以下三点：

1. 本文分析循环神经网络的优势并总结其理论和应用方面的局限性，利用时序卷积网络和注意力机制等前馈结构设计一种既能有效建模历史信息，还具有更高训练性能的前馈序列网络（Feed-forward Sequential Network，FSN）。我们用相关实验验证了模型相比 RNN 类模型更好的训练性能和预测准确度；

2. 我们分析现有用于时间序列预测模型的损失函数，发现常用的 MSELoss 和 HuberLoss 未能关注到时间序列数据的波动形态和延时性，而这两者对于实际预测场景又极其重要，为此我们结合动态时间规整（Dynamic Time Warping, DTW）算法设计了多尺度 DTW 损失函数（**M**ulti-**S**cale **DTW** with Temporal Distortion **I**ndex, MS-DTWI），并用相关实验和分析验证其有效性；
3. 为解决实际场景中对商品管理和销量预测的需求，我们设计和实现了“金牛（Taurus）系统”，并整合了传统时间序列预测算法和上述前馈序列网络算法，使系统能够处理不同数据量下时间序列预测任务。系统连接了科研创新和企业需求，实现时间序列预测算法的社会意义。

1.4 本文组织结构

本文围绕基于协变量的一维多步点预测问题进行讨论，首先提出解决通用序列建模问题的 FSN 模型，在应用模型的前提下，从训练损失函数角度分析现有损失函数的局限性，设计了 MS-DTWI 损失函数，并通过实验验证上述算法的有效性，最后将算法应用在商品销量预测系统中。本文分为六章，第一章为绪论部分，介绍时序预测问题的研究背景及意义，列举了一些经典的和当下主流的时序预测方法，总结出当下此问题遇到的困难与挑战，列举本文的主要贡献点；第二章为相关工作部分，展开介绍基于统计学、基于机器学习方法和基于当下主流的神经网络方法的时序预测算法；第三章为基于时序卷积网络和注意力机制的前馈序列网络方法介绍，在介绍相关基础算法后，给出了此方法的详细设计方案和实验结果；第四章为时间序列预测模型损失函数的分析和设计，总结现有损失函数的局限性，结合 DTW 算法设计能够克服局限性的损失函数，并通过实验验证结论；第五章为上述方法在商品管理和销量预测系统中的应用，以实际运行效果图和相关数据的形式给出系统展示；第六章对全文做出总结，并对未来工作进行展望。

第二章　相关工作

对时间序列预测问题的研究一直是学术界关注的重点，因为实际应用场景很多，但随着数据量不断增大，数据形式越来越复杂，基于统计学的参数化方法无法有效建模预测函数，参数化限制了预测模型的形式，非参数化的机器学习模型能够利用灵活的形式和强大的拟合能力建模复杂的时间序列预测函数，近几年迅速发展的深度学习方法不仅具有上述特点，还具有能够端到端训练的特点，提高了模型训练和预测的效率。

我们先用形式化方法明确基于协变量的一维多步点预测任务，再介绍解决此任务的相关方法。设时间 t 为当前时刻，我们利用 t 之前的时间序列 $X_{1:t}$ 和协变量来预测未来 T 个时间点的时间序列值 $X_{t+1:t+T}$ 。

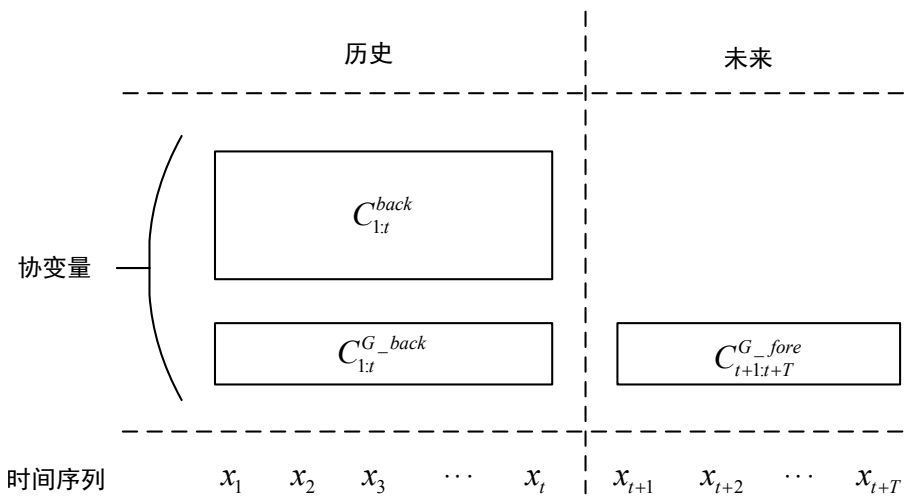


图 2-1: 时间序列预测任务分析

不同协变量与时间序列本身有着不同程度的相关性，比如实际中对于饮料销量预测，面包等食品与其相关性至关重要，而日用品的销量就与其几乎没有相关性，同时注意一点，上面提到的协变量是在当前时间点预测时未知，而对历史时间点已知的变量，此处称为历史协变量，我们用 $C_{1:t}^{back}$ 表示，图 2-1 表示了各种协变量和时间序列的关系。在模型进行预测时无法利用当前时间点信息，只能将历史协变量编码成记忆特征，而后利用记忆特征结合其他特征来预测。另外，有一些协变量是可以提前预知的，比如时间、节假日特征等，对于饮料的销

量同样有影响，这些时间特征不仅能够用在历史特征中，还能编码到未来特征中用于预测，这种特征有学者称为全局特征，我们将这种全局特征分为两部分，一部分为历史全局协变量（$C_{1:t}^{G\_back}$），另一部分为未来全局协变量（$C_{t+1:t+T}^{G\_fore}$）。

不论包含哪种协变量，我们的目标都是为了精准的预测 $X_{t+1:t+T}$，不同协变量的区别在于模型要对它们进行编码时，如何整合协变量特征用于预测。编码模块的设计是影响模型效果的重要因素。

当输入的协变量不同时，时序预测任务的框架各不相同。如果只包含历史协变量，此时的时间序列数据一般是连续等间隔采样，而间隔的单位并不是分钟、小时或者天数等常见日期时间单位：

$$X_{t+1:t+T} = f\left(X_{1:t}, C_{1:t}^{back}\right) \tag{2-1}$$

当包含历史全局协变量和未来全局协变量时（最常见的情况为只利用日期特征进行预测），其形式如下：

$$X_{t+1:t+T} = f\left(X_{1:t}, C_{1:t}^{G\_back}, C_{t+1:t+T}^{G\_fore}\right) \tag{2-2}$$

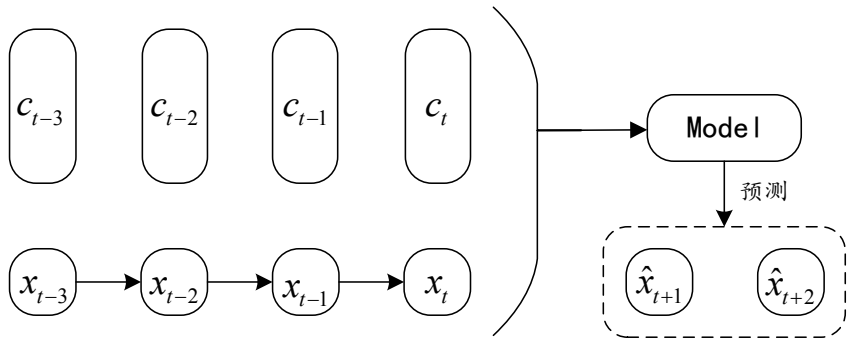
当包含历史全局协变量、未来全局协变量和历史协变量时（协变量一般为时间日期特征和其他相关序列特征），其形式如下：

$$X_{t+1:t+T} = f\left(X_{1:t}, C_{1:t}^{G\_back}, C_{t+1:t+T}^{G\_fore}, C_{1:t}^{back}\right) \tag{2-3}$$

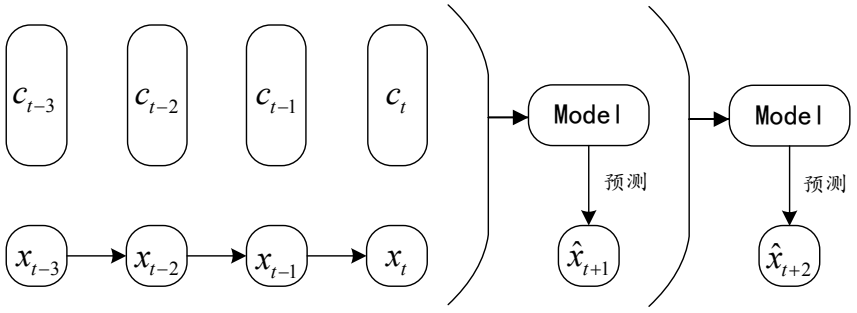
本文用于验证模型性能的框架为式 2-3 所示的函数结构，时间序列预测算法的目标为建模函数 f，基于统计学的经典算法无法利用协变量，而只能利用历史序列值 $X_{1:t}$，基于机器学习的方法开始利用协变量来设计模型，而基于神经网络的模型能够有效利用各种形式的协变量，对于神经网络模型而言，对不同协变量的处理问题难度在于如何对它们进行编码，以及如何将编码后的特征与历史序列值信息整合。

除了对协变量的分类外，时间序列预测任务还需关注序列的马尔可夫性，即当前时间点的时序值只与前 m 个历史时序值相关，称作 m 阶马尔可夫性。实际上历史的任何时序值都是和当前时序值相关的，但是时序模型的复杂度随着 m

的变大而变大，所以我们通常假设时间序列满足 m 阶马尔可夫性，此时我们称 m 为窗口大小，即我们在模型的训练和预测时，只利用 t 时间点之前 m 个时间步数据和协变量来预测未来 T 个时间步的序列值。



(a) 直接预测法



(b) 迭代预测法

图 2-2: 两种多步预测方法

对于预测 $X_{t+1:t+T}$ 的 T 个时间步序列值，通常分为两种方法，即直接预测法和迭代预测法。前者利用已知历史信息（历史时序值和协变量）直接预测得到 T 个时间步序列值，后者则需要迭代预测，即先利用已知历史信息预测得到 x_{t+1} 的值，再利用历史信息和预测得到的 x_{t+1} 预测得到 x_{t+2}，以此类推，迭代 T 次即可得到 T 个待预测的时序值。两种方法的运行逻辑见图 2-2，他们各有优劣，前者预测效率高，一次性输出所有预测值，但预测结果的时序性相对较弱，从而对预测精度造成影响；后者通过迭代法使上一个时间步的预测值能够用于下一个时间步的预测，在每一个时间步的预测阶段都能利用历史信息，但迭代的形式带来误差累积和效率低的弊端。对于不同预测场景，需要比较两种方法的实际预测效果，进而做出选择。

时间序列预测的模型有很多种，基于统计学和机器学习的方法普遍采用迭代预测法，而基于神经网络的方法直接预测和迭代预测都可行，需要做进一步

对比选择。近些年的一些研究认为，直接预测法的要优于迭代预测法[17-18]，因此本文提出的方法都使用直接预测法。而预测环节是时间序列预测任务最后一环，预测效果很大程度上取决于对时间序列预测函数 f 的建模，下面我们针对不同时间序列模型给出相关介绍和分析，总结基于统计学、机器学习和现在流行的神经网络方法解决时间序列预测任务的特点，最后总结对比各模型的优劣，从而说明本文模型设计的出发点。

2.1 基于统计学的方法

基于统计学的方法是时间序列预测早期的解决方案，统计学理论用于时间序列这种容易发生概念漂移的数据时，难免会遇到一些复杂的情况，所以早期的统计学模型都以相应的假设为前提。这些方法包括随机游走法（Random Walk）、自回归法（Autoregressive, AR）、移动平均法（Moving Average, MA）、整合自回归移动平均法（Autoregressive Integrated Moving Average, ARIMA）[19] 和指数平滑法（Exponential Smoothing, ETS）[20]。本小节主要介绍整合自回归移动平均法和指数平滑法。

2.1.1 整合自回归移动平均法

整合自回归移动平均法（ARIMA）是 AR 和 MA 的整合，其中 Integrated 通常指差分操作，其目标在于将非平稳的时间序列通过差分的方式转化为平稳序列，因为 AR，MA 都建立在时序平稳性假设上，所以为了将方法应用在更多的场景，同时整合自回归法与移动平均法的优势，统计学家设计了这样一种应用极为广泛的时间序列预测算法，算法设计简单、理论基础强、易于实现，而且效率很高。

上面说到 ARIMA 是由 AR 和 MA 的结合，所以为了介绍 ARIMA 法的原理，先要对 AR 和 MA 两个算法进行说明。自回归法的核心思想在于描述当前时序值与历史时序值之间的关系，用时间序列自身的历史值对未来时序值进行预测，利用历史 p 个时序值来预测下一个时序值的公式如下：

$$\mathrm{AR(p)}: \quad x_t = \mu + \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t$$

其中 x_t 是当前值，μ 是常数项，p 是阶数，ϕ_i 是自相关系数，ϵ_t 是误差项。p 越大表示当前值受历史越多个时间序列值得影响。自相关系数 ϕ_i 的变化表示时序的不同特征，误差项 ϵ_t 的方差表示序列的数值范围，而不影响序列特征。

移动平均法的核心思想在于使用历史预测的误差来建立类似回归的模型，对于利用 q 个历史预测误差进行建模的表达式如下：

$$\mathrm{MA(q)}: \quad x_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t$$

其中 x_t 是当前值，μ 是常数项，q 是阶数，θ_i 是移动平均系数，ϵ_t 是误差项。移动平均法表示每一个值都可以被认为是一系列历史预测误差的加权移动平均值。

自回归法和移动平均法在 $-1 < \phi_i < 1, -1 < \theta_i < 1$ 条件下是可逆的，任何一个 AR(p) 模型都可以用一个 MA(∞) 表示，同时 MA(q) 也可以用 AR(∞) 来表示，这种可逆的属性不仅让 AR 和 MA 可以互相转化，还给后面模型的优化提供了很好的数学基础。

将 p 阶自回归和 q 阶移动平均结合起来，建模经过 d 阶差分后具有平稳性的时间序列，就得到了 ARIMA(p,d,q) 法，两者整合后的模型其实包含了很多经典的模型，表 2-1总结了 ARIMA(p,d,q) 与多种经典模型的关系。

表 2-1: ARIMA(p,d,q) 法和其他经典模型的关系

白噪声模型	ARIMA$(0,0,0)$
随机游走模型	ARIMA$(0,1,0)$
自回归模型	ARIMA$(\mathrm{p},0,0)$
移动平均模型	ARIMA$(0,0,\mathrm{q})$

对于 ARIMA$(\mathrm{p},\mathrm{d},\mathrm{q})$ 中的三个超参数 p, d, q，我们在建模的时候需要参考自相关系数图（Autocorrelation Function, ACF）和偏自相关系数图（Partial Autocorrelation Function, PACF）的截尾和拖尾情况来决定。建模的过程中往往需要对比不同超参数的拟合效果，最终选取一个相对最优的模型。单纯的 ARIMA 模型在逐渐复杂的时间序列数据下不断体现出其局限性，之后的学者在此基础上不断优化补充，提出了考虑季节性和支持外生变量的 SARIMAX 模型及其向量化模型 VSARIMAX 等[21]。

2.1.2 指数平滑法

指数平滑法（Exponential Smoothing, ETS）经历了由简单指数平滑法到考虑各种因素的复杂指数平滑法过程，起初最简单的指数平滑法只考虑历史几个序列值，利用距离最近的几个序列值作为预测的输入，例如一阶指数平滑法只利用过去一个值作为参考，以加权平均，或者单一分量的形式（如下面公式所示）作为预测值，不考虑趋势或季节因素，形式化表示如下：

$$\begin{aligned} &\text{预测项：} && \hat{x}_{t+1} = \ell_t \\ &\text{平滑项：} && \ell_t = \alpha x_t + (1-\alpha)\ell_{t-1}, \end{aligned}$$

其中 $\hat{x}_{t+1}$ 为预测值，l_t 为用于预测的单一分量，$\alpha \in [0,1]$ 为平滑参数，表示历史时序值对于预测的影响程度。

单一分量的指数平滑法无法建模趋势和季节因素，后来提出的 Holt 线性趋势法[20] 和阻尼趋势法[22] 解决了对趋势项的建模，Holt 和 Winters 将 Holt 线性趋势法进行拓展，设计了可以捕获季节因素的 Holt-Winters 季节性方法[23]。他们对趋势因素和季节因素的引入主要通过添加相应的趋势分量和季节分量。

如下为考虑趋势和季节因素的 Holt-Winters 季节性指数平滑法的形式化表示：

$$\begin{aligned} &\text{预测项：} && \hat{x}_{t+1} = \ell_t + b_t + s_{t-m+1} \\ &\text{平滑项：} && \ell_t = \alpha\left(x_t - s_{t-m}\right) + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right) \\ &\text{趋势项：} && b_t = \beta^*\left(\ell_t - \ell_{t-1}\right) + \left(1-\beta^*\right) b_{t-1} \\ &\text{季节项：} && s_t = \gamma\left(x_t - \ell_{t-1} - b_{t-1}\right) + (1-\gamma) s_{t-m} \end{aligned}$$

其中 l_t 为平滑值，b_t 为趋势项，β^* 为趋势参数，s_t 为季节项，m 表示季节时间步长，即 s_t 和 s_{t-m} 为相同季节，γ 为季节参数。

指数平滑法在应用时都需要定义一些初始值和超参数，只要知道这些值，预测值就可以用公式计算出来，上面各种指数平滑法通常有多个初始值要定义，在某些情况下，平滑参数可以根据预测者以前的经验来设置，但从已有的时序数据来观测得到显然更合理可靠，实现这个设想的方法为最小化残差平方和法，通常简称为误差平方和法。

指数平滑法有很多种同类衍生模型，上面只是简单介绍其中的两种，Pegels (1969) 通过总结趋势项和季节项的不同组合变化，将指数平滑法分成九种。这些方法的发展都在某些方面解决了其他方法受到的限制，使得指数平滑法能够适用于更多场景。关于指数平滑更多的算法细节可以参考 Hyndman et al. (2008)。

ARIMA 法和指数平滑法是基于统计学的时间序列预测算法中具有代表性的两种算法，在传统的小规模、一维时间序列数据上表现良好，效率高，而且具有很强的可解释性。但在当下大数据时代，面对多维时间序列或带有协变量的时间序列建模时，这些算法仅能对线性关系建模，而无法拟合非线性关系，更无法利用协变量做预测。另外这些方法往往需要人工设定一些超参数，这些参数的设定往往需要人为干预，使得建模效率低下，而且参数一旦选定就固定下来，无法随着数据分布的变化而及时进行调整，即无法解决概念漂移问题。

2.2 基于机器学习的方法

近些年机器学习算法已经成为工业界解决大数据问题的重要方案，针对时间序列预测也被提出了很多优秀的模型，他们有着比传统统计学算法更好的泛化能力，预测准确度也大幅提高，能够突破统计学在稳定性和高维度等方面的限制。应用比较多的算法包括支持向量回归法（Support Vector Regression, SVR）[26]，基于 L-BFGS 的 Prophet[27]，基于集成学习的 XGBoost[28]，不同场景下，这些算法有着独特的优势，本小节简单介绍支持向量回归法和 XGBoost 算法。

2.2.1 支持向量回归

支持向量机是基于神经网络算法流行前解决分类问题最常用的算法之一，支持向量回归与支持向量机原理类似，区别在于支持向量机为解决分类问题，支持向量回归用于解决时序问题。时间序列预测为回归问题，将时间序列数据按照一定窗口大小 w，来构造训练样本，可以将训练样本表示为

$$D = \{(X_{1:w}, x_{w+1}), (X_{2:w+1}, x_{w+2}), \cdots, (X_{t-w:t}, x_{t+1})\}$$

为方便描述，用 $\boldsymbol{a_t}$ 表示 $X_{t-w:t}$，b_t 表示 x_t。我们的目标是利用训练数据得到一个回归模型 f，使得 $f(\boldsymbol{a_t})$ 与 b_t 尽可能接近，支持向量回归算法的核心在于对损失函数的设计，传统回归模型通常基于模型预测结果 $f(\boldsymbol{a_t})$ 与真实输出 b_t 之间的差别来计算损失，当且仅当模型预测结果与输出完全相同时，损失为零。SVR 则能容忍 $f(\boldsymbol{a_t})$ 与 b_t 最多有 ϵ 的偏差，即当预测值与真实值之间的差别大于 ϵ 时才计算损失。如图 2-3所示，此时相当于设计了一个宽度为 2ϵ 的间隔带，若训练样本落入此间隔带，就算作是正确预测。

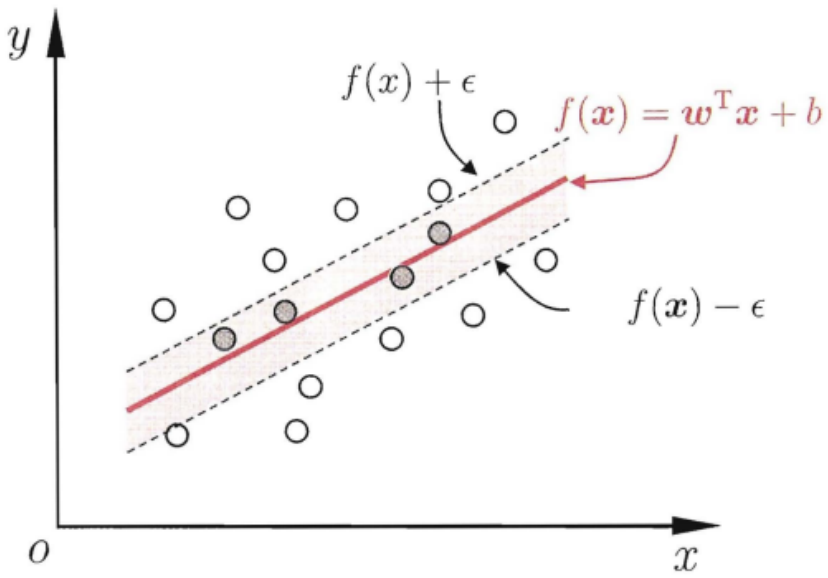


图 2-3: 支持向量回归

于是支持向量回归问题相比于支持向量机模型，在优化目标上多了一个正则化项：

$$C \sum_{i=1}^{m} \ell_c \left(f\left(\boldsymbol{a}_t\right) - b_t \right)$$

其中 C 为正则化系数，ℓ_c 为 ϵ-不敏感损失函数：

$$\ell_\epsilon(z) = \begin{cases} 0, & \text{if } |z| \leqslant \epsilon \\ |z| - \epsilon, & \text{otherwise} \end{cases}$$

支持向量回归法和支持向量机类似，都可以通过结合拉格朗日乘子法、对偶方法和核方法对模型做参数求解和进一步优化。其中核方法是一种功能强大的技巧，他是一系列核函数方法的综合，通过引入核函数，可以将线性学习器拓展为

非线性学习器[29]。

2.2.2 XGBoost

XGBoost（Extreme Gradient Boosting）是梯度提升决策树算法GBDT（Gradient Boosting Decision Tree）的一种高效实现，同时添加了很多新技巧和方法。所以本小节主要介绍其本质的梯度提升决策树算法（GBDT）。

GBDT 的核心思想在于 Boosting，即通过将表现一般的多个 CART 决策树模型组和起来，集成一个表现好的模型。模型优化的目标仍然是最小化预测值 $f(\boldsymbol{a}_t)$ 和 b_t 之间的误差，而训练过程实际上是对任意一个可导的目标函数的优化过程。

GBDT 用于回归问题时可以选择平方损失、绝对值损失或 Huber 损失（前两者的结合），对于损失函数的拟合问题，Freidman 提出利用损失函数的负梯度来拟合，进而拟合一个 CART 回归树。

假设输入训练样本为 $D = \{(\boldsymbol{a}_{1:w}, b_w), (\boldsymbol{a}_{2:w+1}, b_{w+1}), \cdots, (\boldsymbol{a}_{t-w:m}, b_m)\}$，一共 m 个训练样本，其思想可以形式化表达如下：

1. 初始化弱学习器：

$$f_0(\boldsymbol{a}) = \underbrace{\arg\min}_{f} \sum_{i=1}^{m} L(b_i, f)$$

其中 $f_0(\boldsymbol{a})$ 表示第一个弱学习器，$L(b_i, f)$ 为损失函数；

2. 迭代 $y = 1, 2, \cdots, T$ 轮，更新强学习器：

(a) 对于样本 $i = 1, 2, \cdots, m$，负梯度如下：

$$r_{ti} = -\left[\frac{\partial L(b_i, f(\boldsymbol{a}_i)))}{\partial f(\boldsymbol{a}_i)}\right]_{f(\boldsymbol{a})=f_{t-1}(\boldsymbol{a})}$$

利用 $(\boldsymbol{a}_i, r_{ti})$，拟合 CART 回归树，得到第 t 棵回归树，得到叶子节点区域 $R_{tj}, j = 1, 2, \cdots, J$，其中 J 为回归树叶子节点区域的个数；

(b) 对叶子区域 $j = 1, 2, \cdots, J$，计算最佳拟合值：

$$c_{tj} = \underbrace{\arg\min}_{c} \sum_{\boldsymbol{a}_i \in R_{tj}} L(b_i, f_{t-1}(\boldsymbol{a}_i) + c)$$

(c) 更新强学习器：

$$f_t(\boldsymbol{a}) = f_{t-1}(\boldsymbol{a}) + \sum_{j=1}^{J} c_{tj} I\left(\boldsymbol{a} \in R_{tj}\right)$$

3. 得到强学习器 $f(\boldsymbol{a})$ 的结果：

$$f(\boldsymbol{a}) = f_T(\boldsymbol{a}) = f_0(\boldsymbol{a}) + \sum_{t=1}^{T} \sum_{j=1}^{J} c_{tj} I\left(\boldsymbol{a} \in R_{tj}\right)$$

以上为 GBDT 算法的训练过程，为防止过拟合，通常需要添加正则化项，常用的有通过调整子采样比例或者直接对 CART 回归树进行剪枝。基于机器学习的时间序列预测算法具有很强的可解释性，预测准确率也相比基于统计学的方法高很多，主要在于能够利用除时间序列数据本身外的协变量数据，SVR 和 XGBoost 等算法都能够处理离散和连续的高维数据，其中 XGBoost 是基于机器学习的时序预测算法中应用最多的，相比 SVR，它不需要调整太多参数就能够取得相对较高的准确率，另外它对于异常值的鲁棒性也要强很多。但是，XGBoost 存在训练效率低的情况，因为弱学习器之间存在依赖关系，使得难以并行训练数据，而当下流行的基于神经网络的模型通常为端到端模型，能够很好的利用算力提升带来的优势，在效率和准确率上都有所提升。

2.3 基于神经网络的方法

神经网络本是机器学习的一个分支，可以结合相应的模型、算法和解法来得到实现式 2-3映射的函数。之所以单独列出基于神经网络的方法，是因为此类方法在近几年产生很多模型上的创新，展现出优于其他机器学习方法的优势。实际上近几年的神经网络方法和早期的神经网络有所不同，早期的方法结构简单，复杂度低，而近几年的神经网络方法的层数和每层的神经元数量都大幅增加，使得模型参数量增加，复杂度提升，此类神经网络模型我们通常称为深度神经网络（Deep Neural Network，DNN）模型，此处的“深”是层数和神经元数量多的形象化表达。本小节分别介绍传统神经网络和近几年神经网络在时间序列预测上的研究情况。

2.3.1 传统神经网络

针对时间序列预测问题，基于神经网络的方法早已有相关的成果，只是主要集中在传统的浅层神经网络上，比如多层感知机（Multilayer Perception，MLP），时延神经网络（Time-Delay Neural Network, TDNN）等，这些方法的特点在于只有少量的隐藏层、模型复杂度低，激活函数的引入使得神经网络可以拟合非线性函数，参数求解普遍采用梯度反向传播（Back Propagation，BP）的方式。MLP

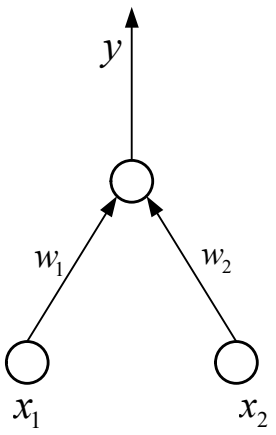
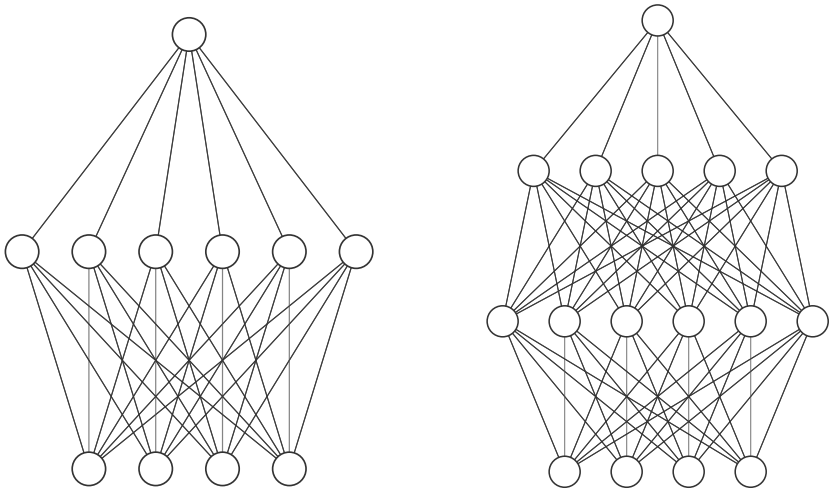


图 2-4: 感知机图示

由简单的感知机单元（如图 2-4[29]）组成，单纯的感知机单元甚至无法解决非线性可分问题，但多层感知机结合，即每层神经元与下一层神经元全互联，神经元之间无同层连接，也无跨界连接，这样多个感知机的结合便得到了“多层感知机”，当神经元在横向增加数量，即仍然有一个输出层、一个输入层和一个隐藏层，隐藏层神经元节点较多时，称为“单隐层前馈网络”，随着隐藏层数量的增加，模型拟合能力越来越强。

早期 MLP 的研究在实际场景中已经有所应用，并将 MLP 与基于统计学的 ARIMA 模型进行对比或结合。比如 Celik et al. 2007 利用 MLP 评估和预测土耳其银行的危机，Zhang et al. 2005 从时间序列分解的角度，将趋势信息和季节信息去除，研究对 MLP 的影响，Sahoo et al. 2006 用 MLP 预测夏威夷河流的流量，Zhang 2003 将 MLP 与 ARIMA 结合，使得两者在线性预测和非线性预测上的能力互补，Tang et al. 1991 从长短期预测的角度对比 ARIMA 和 MLP，发现对于长期预测两者预测准确度相当，而短期时则 MLP 预测效果更好。由相关文献可以看出，MLP 相比统计学算法已经表现出一定的优势，而近几年将网络层数加深，



(a) 单隐层前馈网络 (b) 双隐层前馈网络

图 2-5: 单层和多层前馈网络

则是将神经网络的能力真正地开发出来。图 2-5 表示单层和多层前馈网络。

2.3.2 深度神经网络

随着数据量的增大、数据维度的提高和不同维度间的相关性增加，使得式 2-3表示的映射函数的构建变得困难，对于输入数据的特征提取是最重要的一块，传统统计学算法和其他机器学习算法无法有效利用不同维度之间、不同时间点间的数据特征，“特征工程的效果决定了模型能力的上限”，所以对于特征的提取至关重要。

从 2012 年开始，深度神经网络逐渐受到大家重视，什么叫深度神经网络? 他和以往传统的神经网络有什么区别? 在上一小节我们介绍了神经网络的基本构成，它是由多层感知机发展而来，在层数和宽度上有所扩充，嵌套迭代而成的模型，而深度神经网络的特点在于层数加深再加深，使得网络有很多个隐层。卷积网络发展初期被用于 ImageNet 竞赛[35]，冠军方案就是用了 8 层的神经网络，2015 年时增加到了 152 层，而到了 2016 年就已经多达 1207 层，模型逐渐加深，逐渐由传统的神经网络演变成当下的深度神经网络。

对于时间序列预测问题，它本质上是对序列数据的建模，因为时间序列数据本身是序列型数据，和自然语言和音频数据等有很多共同的特点，比如都存在高维性、马尔可夫性和维度间相关性。所以不断发展的序列模型都会被用于时间序列预测任务，其中循环神经网络（Recurrent Neural Network, RNN）因其

迭代建模的性质通常被认为是解决序列建模的首选方案。循环神经网络的结构如图 2-6 所示：

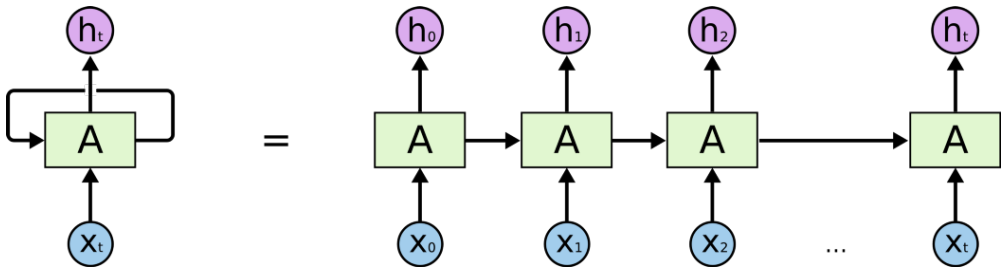


图 2-6: RNN 网络结构

图中左侧为循环单元，方块部分为循环网络主体，右侧为按照时间维度铺展开得到的结构图，实际为循环单元在每一个时间点上进行相似计算过程的重复。对于每个时间点的输入 X_t 来说，在循环单元里，进行着如下计算：

$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}X_t\right) \tag{2-4}$$

其中 h_t 为隐藏层输出，W_{hh} 和 W_{xh} 为隐藏层权重，tanh 为激活函数。式 2-4 的递归计算就是循环神经网络的核心思想，其中 W 作为隐层权重，承载着从开始到结束的记忆，这种长时记忆能能力用来建模序列数据的马尔可夫性，很好的模拟了人类记忆的过程。

但是简单 RNN 模型存在梯度消失的问题，后来提出的长短时记忆模型（Long Short-Term Memory，LSTM）[12] 一定程度上克服了 RNN 的梯度消失现象，常被作为 RNN 类模型的代表，很多时间序列预测模型都是基于 LSTM 设计的[36]，比如亚马逊提出的时间序列预测模型 DeepAR[37]。实际上对于 RNN 类模型（基于 RNN 思想的一类模型）和其他深度学习模型而言，梯度消失现象都是存在的，但是由于 RNN 的递归属性，在利用反向传播求导的过程中，链式法则会将梯度小于 1 的数值随着时间步的增长，即链的增长，以指数形式缩小，导致由损失函数求导得到的梯度在传导过程中逐渐消失，从而将使得参数调整陷入僵局，当梯度大于 1 时，甚至还会存在相反的情况，即梯度过大，造成梯度爆炸。

上面讲到的 RNN 模型是深度神经网络中的一种，其他深度神经网络模型也可以用于解决时间序列预测问题，比如卷积神经网络（Convolutioal Neural Net-

work，CNN）和基于注意力机制的 Transformer 等，这两者属于前馈模型，克服了循环模型更容易梯度消失的缺点，很多序列建模算法是基于以上三个基本模型设计，比如基于 CNN 的 WaveNet[1] 等[38]，基于 CNN 和 RNN 的 LSTNet[39] 和基于 Transformer 的模型 LogSparse Transformer[40] 等。

本小节简单介绍基于神经网络的时间序列预测方法，在几十年的研究过程中，学者们不断推陈出新，优化现有模型，使得利用神经网络进行预测的准确度不断提升。同时如今强大的计算设施使得算力相比上个世纪大幅提升，这也和深度神经网络的发展起到相互促进的作用。深度神经网络成为当下的主流，其实离不开机器学习的基本原理，即模型的复杂度和它的容量相关，而容量又和学习能力息息相关。所以说它强大的学习能力是和模型的复杂度紧密相关的，深度神经网络的复杂度是随着神经网络变深或变宽而增加的，而学者们发现，相对于变宽，深度的增加要更关键，深度增加意味着增加了嵌入程度，这是深度神经网络特征提取能力强的直接原因。对于时间序列数据，我们需要对深度模型做更多的适配和优化，使得其能够更好地提取时间序列特征，有效利用序列数据和协变量，这也是本文后续算法努力的方向。

2.4 本章小结

本章对解决时间序列预测任务的统计学算法、机器学习算法和基于神经网络的算法做了梳理，统计学算法原理简单、可解释性强，适合解决小数据量的一维时间序列预测，而对于包含协变量的时间序列多步预测，基于机器学习和神经网络的方法准确度更高，因为他们可以有效利用协变量提供的特征信息帮助预测，但机器学习类算法缺少对离散特征和连续特征之间关系的建模，基于神经网络的算法，确切地说是基于深度学习的算法，可以有效提取特征间的相关性，同时有效建模序列数据的马尔可夫性，深度网络架构中的多种记忆机制可以保留历史信息，更符合人类的记忆模式，预测准确度也更高。

第三章　基于卷积的前馈序列网络 FSN

前文分析总结了当前时间序列预测中面临的问题和挑战，在领域的不断发展过程中，学者们提出很多优秀的算法解决不断出现的问题，但是目前已有工作对于提到的挑战解决程度还有提高的空间。

本章先对基于神经网络的时间序列预测任务进行拆解分析，明确本章设计的网络结构在预测任务中所处的环节。整理用于解决时序预测问题的神经网络算法在发展过程中所面临的挑战和优化思路，总结出一个用于时序预测的模型应具备的特点属性，在这些特点属性和当下面临的挑战性问题的引导下，我们设计了基于卷积的前馈序列网络模型（Feed-forward Sequential Network，FSN），此网络结构借鉴循环网络对序列数据的表达能力，同时利用前馈网络结构克服了循环网络面临的各种挑战。

3.1　模型设计思路

第二章明确了我们解决的时间序列预测任务为式 3-1 所表示的结构：

$$X_{t+1:t+T} = f\left(X_{1:t}, C_{1:t}^{G\_back}, C_{t+1:t+T}^{G\_fore}, C_{1:t}^{back}\right) \tag{3-1}$$

函数 f 的建模中协变量的特征表示直接影响着模型的预测效果。机器学习算法已经实现对协变量的利用，比如梯度提升算法利用离散协变量构造 CART 树，但这些算法无法建模不同协变量之间和协变量不同序列位置点之间的相关性和序列先后属性，而这正是时间序列数据特有的属性，对这些特性的建模显然很重要。当前基于神经网络的算法能够有效提取数据特征，2.3.2 介绍了深度学习算法出色的建模能力，我们有必要利用其强大的特征表示优势，解决时间序列预测任务面临的协变量特征表示问题。

式 3-1 中包含三种协变量 $C_{1:t}^{G\_back}$, $C_{1:t}^{G\_fore}$, $C_{1:t}^{back}$，对于这三种协变量中的离散变量，我们采取神经网络中的嵌入网络层编码或者 one-hot 编码两种方式，具体选择哪一种要依据输入的变量类型，比如“月内第 n 天”的变量，我们采取

2 维的嵌入编码方式，而对于“是否为节假日”变量，我们用 one-hot 编码方式，因为历史全局协变量和未来全局协变量为同样的变量在不同时段的数据，对其中相同的变量采用相同的编码方式，三种协变量经过对应的编码器处理过后用如下方式表示：

$$E_{1:t}^{back} = \text{Encoder}_{back}\left(C_{1:t}^{back}\right) \tag{3-2}$$

$$E_{1:t}^{G\_back} = \text{Encoder}_{G\_back}\left(C_{1:t}^{G\_back}\right) \tag{3-3}$$

$$E_{t+1:t+T}^{G\_fore} = \text{Encoder}_{G\_fore}\left(C_{t+1:t+T}^{G\_fore}\right) \tag{3-4}$$

其中三个编码器（Encoder）都是可训练的嵌入网络层（Embedding Layer）。最后得到三个编码向量 $E_{1:t}^{back} \in \mathbb{R}^{t\times b}$，$E_{1:t}^{G\_back} \in \mathbb{R}^{t\times g}$，$E_{t+1:t+T}^{G\_fore} \in {}^{T\times g}$，连同输入时间序列数据 $X_{1:t}$ 作为预测模型的输入，训练模型使其预测输出结果与 $X_{t+1:t+T}$ 的差值最小。

下面介绍利用输入 $X_{1:t}$、$E_{1:t}^{back}$、$E_{1:t}^{G\_back}$ 和 $E_{t+1:t+T}^{G\_fore}$ 得到输出 $X_{t+1:t+T}$ 的过程。因为输入中的前三者都是历史信息，其中 $X_{1:t}$ 是不可缺少的输入，另外两个协变量和 $X_{1:t}$ 拼接作为输入。通过在编码维度上进行向量拼接，我们得到历史特征输入：

$$X_t^{back} = X_{1:t} \,\big|\, E_{1:t}^{back} \,\big|\, E_{1:t}^{G\_back} \tag{3-5}$$

为便于描述，设 $X_t^{fore} = E_{t+1:t+T}^{G\_fore}$，本文设计的网络单元作用就在于对 X_t^{back} 进行序列建模，形式化表示如下：

$$H_t^{back} = \text{SeqMod}\left(X_t^{back}\right) \tag{3-6}$$

其中 SeqMod 为序列建模函数，也即后文将详细介绍的网络结构，H_t^{back} 为整合历史信息后得到的隐向量，将得到的隐向量与 X_t^{fore} 结合，再通过解码模块（Decoder）进行预测，便得到整体对未来 T 个时间步的点预测结果 $\hat{X}_{t+1:t+T}$。

时间序列预测模型的目标为使预测结果 $\hat{X}_{t+1:t+T}$ 尽可能接近 $X_{t+1:t+T}$，通过哪些损失函数来实现这一目标决定了模型收敛到函数空间中哪个极小值，这部分也是本文下一章要探讨的重点内容，此章节的重点在于序列建模，故使用最常

用的均方差损失：

$$Loss_{mse} = \sum_{i=1}^{N} \left(X^{i}_{t+1:t+T} - \hat{X}^{i}_{t+1:t+T} \right)^2 \tag{3-7}$$

预测模型的目标在于最小化均方差损失，在利用训练数据学习的过程中模型参数不断调整，最后收敛稳定。此时的模型便是我们可以用于时间序列预测的神经网络模型。

从前文分析可知，我们要想有效解决时间序列特征表示问题，当下最好的选择是利用神经网络方法，它们具有强大的特征提取能力和非线性函数空间里对预测函数的拟合能力，对于利用神经网络解决时间序列预测问题，从上个世纪就有学者进行了尝试，并总结了一些对于今天建模仍然很有参考价值的理论，下面我们分析这些理论如何引导时间序列预测模型的发展，并找到对于解决当下面临的挑战所需的核心指导思想。

对于时间序列数据模型而言，它比对图像的建模更注重“参数共享的程度影响模型的拟合和泛化能力[41]”这个理论，因为时间序列数据具有 N 阶马尔可夫性，当前时间步的数据与历史时间步数据条件相关，阶数越高相关的历史距离越远，就越需要将这种历史数据对模型的影响进行记忆，从而达到利用全部时间序列做预测的目的。

我们利用将 4 个时间步作为输入进行建模举例，对比分析常见网络结构的参数共享情况。下面为全连接网络、卷积网络和循环网络的计算公式（其中 f 为非线性激活函数，X 为输入序列）：

$$a_{FC} = f\left(X W^{T}_{fc} \right) \tag{3-8}$$

$$a_{CNN} = f\left([X_{1:2} W^{T}_{cnn}, X_{2:3} W^{T}_{cnn}, X_{3:4} W^{T}_{cnn}] \right) \tag{3-9}$$

$$a_{RNN} = f\left(h W^{T}_{hh} + X W^{T}_{hx} \right) \tag{3-10}$$

对于全连接网络，如式 3-8 所示，参数 $W_{fc} \in \mathbb{R}^{3\times4}$ 中对于每个时间步分别对应 3 个参数，不同时间步的参数相互独立，参数之间互不共享；卷积网络参数共享程度强于全连接网络，如式 3-9 所示，卷积核 $W_{cnn} \in \mathbb{R}^{1\times2}$ 在 3 个窗口上共享同样的参数，共享参数的数量取决于核的大小，时间步维度上的共享情况

同样取决于核在此维度上的平移时间间隔；循环网络的参数共享程度比卷积网络更强，如式 3-10 所示，网络中隐向量对应的隐参数 $W_{hh} \in \mathbb{R}^{1\times3}, W_{hx} \in \mathbb{R}^{1\times3}$ 对于每个时间步都是共享的，随着时间步向后推动，共享的权重会与每个时间步输入计算，将记忆存储在隐向量 h 中。

循环网络这种递归计算的结构使其具有能够处理变长输入、历史信息记忆和建模序列相对位置等优点。通过隐向量记忆历史信息，甚至在原理上可以记忆无限长的历史信息，递归计算结构也很契合对 N 阶马尔可夫性的建模要求，从这些特点看来，RNN 可以很好地解决上节中对协变量的建模问题，因此这种具有很强表示能力的循环网络成为很多人解决时间序列建模问题的首选网络结构。

但是，循环网络存在着一些缺陷，一定程度上限制了它的应用，我们总结了主要的三点。(1) 循环网络通过沿时间的后向传播方式调整网络参数，但是通过链式法则可知，多个时间步的累乘，增加了梯度消失和梯度爆炸的风险；(2) 在训练和预测时，当前时间步的计算必须要等待上一个时间步计算完成，这种递归结构限制了并行计算的实现，降低了训练和预测效率；(3) 研究表明，循环网络对于临近时间步的序列顺序比较敏感，而对较远时间步的顺序则很少关注[42]。这表明前面提到的无限记忆长度实际是没有必要的。

最初的循环网络具有比较明显的缺陷，但是研究者们提出的循环网络变体一定程度上缓解了上述缺陷，优化的循环网络变体促进了它的广泛应用，但上述提到的缺点是循环网络固有的，而相比之下前馈网络的计算过程简单直接，不需要递归计算，避免了循环网络第 2，3 点的问题，从上述分析也可以看到，卷积网络作为前馈网络的一种，在参数共享程度上也具有建模序列数据的潜力，近几年学者提出的注意力机制也可以实现参数共享，我们相信结合以上两者的优势，可以设计出不仅具有循环网络的优点，同时克服上述提到的缺陷的网络结构。

3.2 前馈序列网络结构

上节总结出一个优秀的神经网络结构应该具有的特点，这些“前车之鉴”作为本文设计的网络结构的指导思想，我们充分发挥循环网络建模序列数据的优势，并结合前馈网络结构设计得到了前馈序列网络（Feed-forward Sequential

Nework, FSN），即式 3-6 中的序列建模函数 SeqMod。为了清晰地理解前馈序列模型，本节对模型设计过程中参考使用的时序卷积网络和注意力机制做介绍，最后结合介绍过程中提到的模型构建思路引出前馈序列网络结构。

3.2.1 时序卷积

常见的前馈网络包括全连接网络和卷积网络，但从上节对参数共享程度的分析，我们选择主要利用卷积网络结构。时序卷积是由多层卷积堆叠而成，利用卷积核的巧妙设计与向量填补技巧整合成的结构。卷积网络不需要递归计算，能够从本质上避免循环网络的缺陷，简单的卷积网络结构是通过增大滤波器的感受野来获取更长跨度的输入特征。为了模拟循网络对序列相对位置属性和长时记忆属性的建模过程，时序卷积结合了因果卷积和扩展卷积两种子结构。

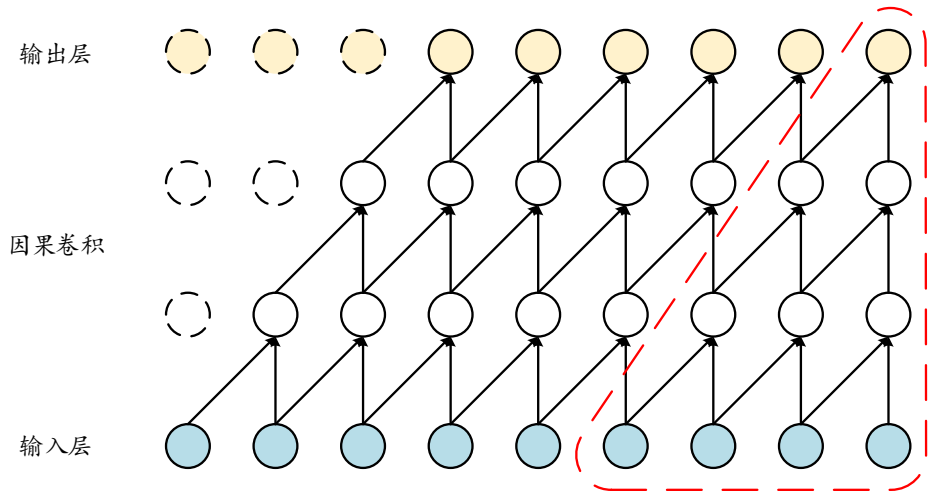


图 3-1: 因果卷积网络[1]

图 3-1 展示了因果卷积的结构，此处我们以核大小为 2 的一维卷积层举例说明，3 个卷积层自下向上堆叠，随着堆叠的层数增加，最下层感受野逐渐变大，但是由于序列点的先后性，站在当前序列点预测下一个序列点时，模型不可以利用未来信息，即未来序列点的特征或序列值，而只能利用历史序列点的特征。为了达到这种防止未来信息泄露的效果，需要在每个序列点都只整合当前序列点以前的特征，所以因果卷积最右上侧的特征类似于一个“直角三角形”的顶点，同时要注意，由于卷积计算的原因，上层卷积特征序列长度会逐渐变短，此时我们需要填补相应数量的序列点特征来维持每层的序列特征长度相同，因果

卷积要求只在左侧进行填充（图 3-1中左侧虚线圆圈），从而使特征向右侧集中，达到特征“收拢”的效果，实际中填充的元素可以有很多选择，我们的方案是简单填充 0 元素。

以上就是因果卷积的主要思想，这种结构巧妙地利用堆叠的卷积模拟了循环网络对序列相对位置属性的建模过程，同时能够利用卷积网络前馈计算和并行化计算的优势。但因果卷积对于达到建模循环网络的效果还完全不够，因为它的记忆长度的增加需要堆叠很多卷积层，这种堆叠会增加大量参数，从而增加模型复杂度，提高过拟合风险。扩展卷积的出现可以有效解决这个问题。

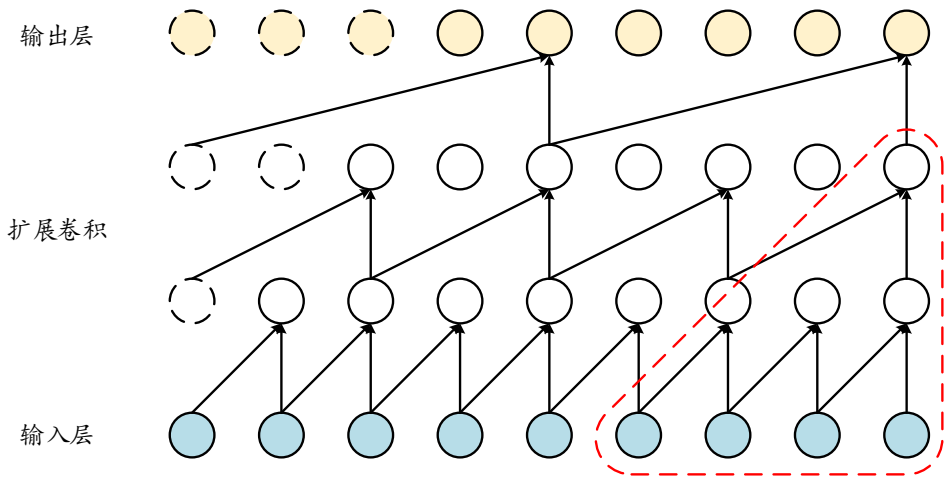


图 3-2: 扩展卷积网络[1]

如图 3-2 所示，假设每层卷积的核大小为 2，由下到上一共 3 层卷积，扩展系数分别为 1、2、4，假设第二层的序列特征为 $a_1, a_2, \cdots, a_t$，则第二层的第一个卷积核的输入为 a_1, a_3，同理假设此时扩展系数为 4，则第一个卷积核的输入为 a_1, a_5，通常情况下，随着层数增加，扩展系数呈指数变大，此处举例就是以 2 的指数倍设置扩展系数。

这种设计的结果使得感受野呈指数倍变大，这样就可以通过少量的层数堆叠获取很长的历史序列点特征，如图 3-2 右侧所示，2 层扩展卷积的感受野就达到了 3 层因果卷积的感受野大小。而且由于卷积的前馈结构，不会存在对临近时间步更重视，而忽略远处时间步的问题，避免了循环网络中历史时间点特征在当前“打折”的现象，可以看出，当扩展系数为 1 时，扩展卷积就退化为普通卷积。这就是扩展卷积的主要思想和结构。

扩展卷积有效模拟了循环网络的长时记忆能力，而且克服了循环网络对远

处时间点特征淡化的缺点，其在序列建模[43-44]和图像处理[45-46]都有着广泛的应用，它避免了单纯卷积网络的大量参数，有效精简模型，避免随着参数量大增导致复杂度增加的风险。

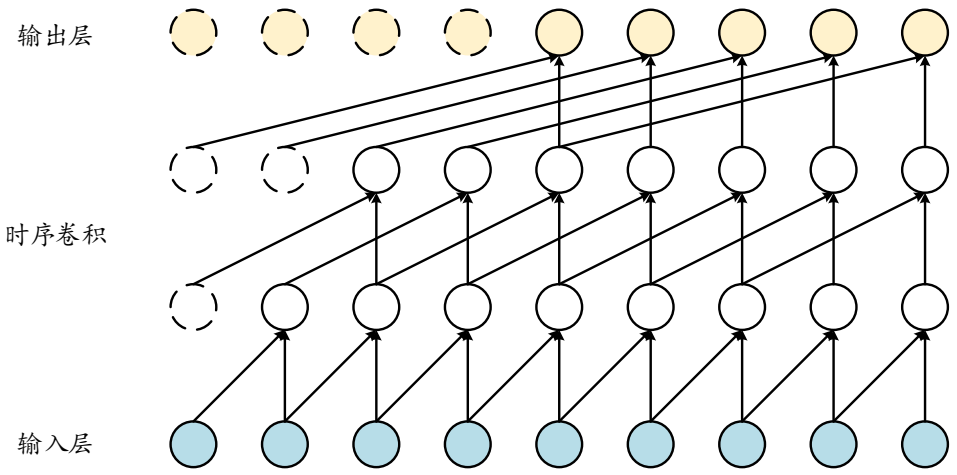


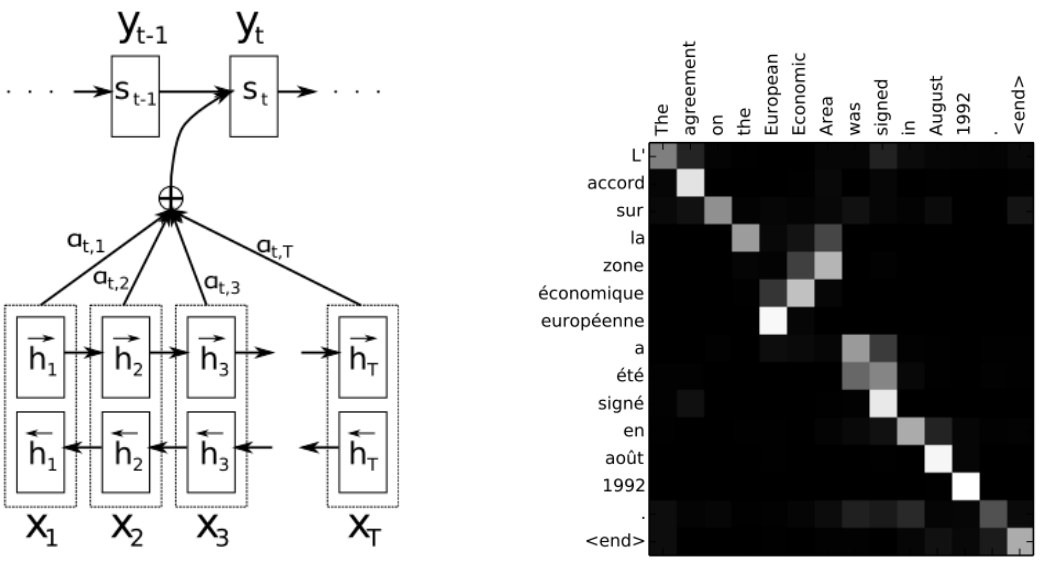
图 3-3: 时序卷积网络

因果卷积和扩展卷积的结合便得到时序卷积，其结构如图 3-3所示，Bai et al. (2018, 2019) 通过理论分析和大量实验对比，证明了时序卷积可以有效替代循环网络，并且在性能上具有前馈网路的可并行计算能力，在项目部署应用上更高效。

时序卷积网络有很大的发展潜力，越来越多的模型使用时序卷积网络代替循环网络，比如语音合成领域，对于语音数据先前人们默认选择循环网络对其进行编码，但是 van den Oord et al. (2016) 提出的 WaveNet 利用时序卷积网络模型解决语音合成任务，为语音合成领域提供了很多新的思路，对于的文本处理也是一样，前馈的时序卷积网络和 Transformer 也逐渐被学者们改良创新，迭代出更多优秀的模型。

通过图 3-2 可以发现，最上层最后一个编码特征整合了前面所有序列点信息，上面讲到扩展卷积的作用是利用较少的层数来获得较大的感受野，但是，感受野的大小仍然受扩展系数和卷积核大小限制，另外，从人类记忆和预测的模式来分析，并不是所有历史信息都同样重要，比如对于下一个节假日的销售预测，显然前几个节假日的数据更具有参考性，而工作日的作用就会次之，为了建模这种不同序列点的重要程度，我们结合注意力机制思想。

3.2.2 注意力机制



(a) 注意力机制 (b) 权重可视化

图 3-4: 注意力机制示意图[2]

注意力机制源于对机器翻译任务的模型优化[2]，翻译的对象为自然语言文本，也属于序列数据。其思想动机在于模拟人类对于输入信息分配不同的重视程度，比如对于文本翻译任务，对于目标语种的某个词而言，不是所有的输入文本都同样重要，肯定会有一些特定的、与其对应的词起着对这部分翻译的主要作用，注意力机制旨在给这部分文本特征分配相比其他部分更高的权值，使得输入到解码端的特征包含更多重要信息。

如图 3-4a 所示，对于预测 Y_t 而言，它需要利用所有的输入 $X_1, X_2, X_3, \cdots, X_T$，但为了达到给重要特征分配更多注意力的目的，注意力机制为每个输入对应着分配了权值，即 $\alpha_{t,1}, \alpha_{t,2}, \alpha_{t,3}, \cdots, \alpha_{t,T}$，这些权值是可学习的，即随着训练的进行，模型参数和权值在反向传播过程中不断调整，最后在模型收敛时，得到目标的权重。通常应用注意力机制的模型会将权重通过可视化的方式展示出来，如图 3-4b 所示，可以看出，对于单词‘Area’的预测，第 4，5 个输入单词重要性要远高于其他单词。

在此思想基础上，Cheng et al. (2016) 提出 Self-Attention 结构，如图 3-5 所示，这种结构的特点在于由输入特征本身得到注意力权值，再将权值与输入相乘，得到加权后的输出向量，上图中最后的加权向量颜色越深表示权值总和越

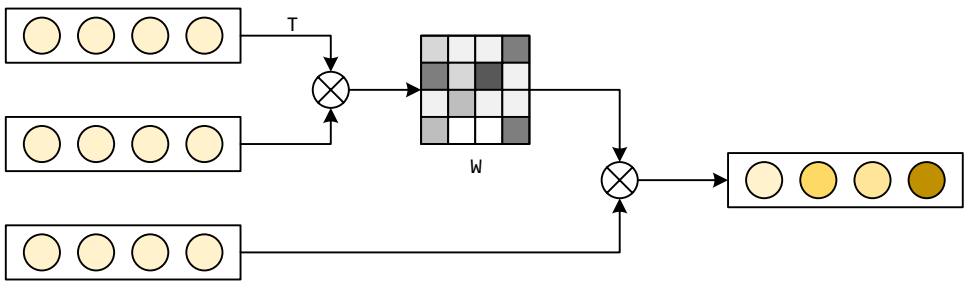


图 3-5: Self-Attention 结构

大，权重矩阵 W 中不同颜色表示不同大小的权值，当模型收敛时，权重矩阵中第 i 行第 j 列表征序列中第 i 个时序点特征对第 j 个时序点特征的影响程度。

这种结构为序列模型提供了新的思路，研究者们在此基础上设计出很多优秀的模型，其中最成功的模型是如今流行的 Transformer[14]，它将多个 self-attention 结构嵌套，并结合了 ResNet[49] 和 BatchNorm[50] 等技巧，当下很多优秀的模型都以 Transformer 为基础单元，比如对文本建模的 Bert[51] 和 GPT 系列[52-54] 模型。Transformer 中应用了多头注意力机制（Multi-head Attention），即多个 Self-Attention 结构的结合，每个 head 表示一个包含 Self-Attention 的单元，每个 head 学习到的特征表示侧重点不同，这种模式使得模型从不同角度对输入做表征，有利于输入特征的挖掘。

3.2.3 前馈序列网络

上文从时序卷积到注意力机制的介绍，实际上就是我们模型结构的设计过程，本节将详细介绍我们如何将注意力机制的思想应用在时序卷积中，从而达到本章开始时讲到的目标，即在吸取循环网络优点的前提下，利用前馈网络克服循环网络的缺点。

如图 3-6 所示，中间部分为前馈序列网络结构。设输入序列为 $X_t^{back} = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t$，其中粗体的 $\mathbf{x}_i$ 表示第 i 个时间步位置的输入特征向量，3 层的时序卷积作为序列模型的外框架，$S_i^{(l)}$ 表示第 l 层中第 i 个时序位置的特征，$H_t^{back} = \{h_1, h_2, \cdots, h_t\}$ 表示 FSN 输出的隐向量集，为方便描述，此处 $t = 5$，即假设有 5 个时间步特征作为输入。每一个序列点的特征都是它前面时间步的特征信息整合，这是为了满足对时间序列数据的建模要求，即不能使用未来数据，对于时序卷积和后面引入的注意力机制都考虑到这一点，图 3-6 中绿色部分在

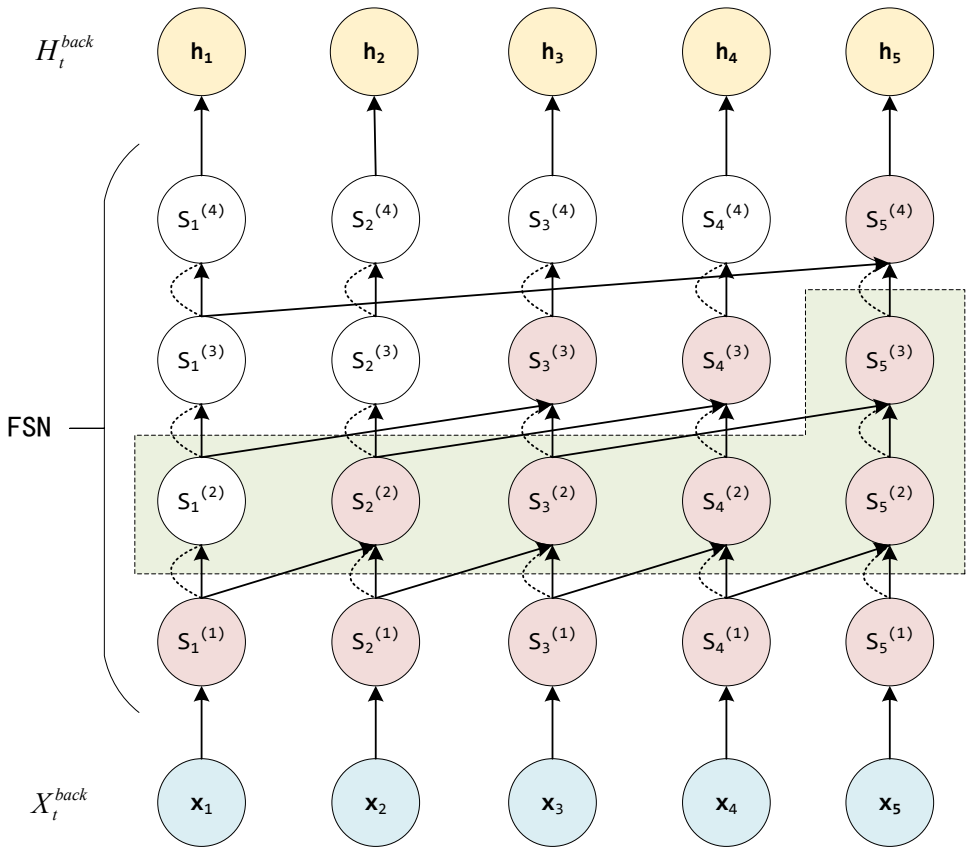


图 3-6: 前馈序列网络结构

图 3-7 中详细展示:

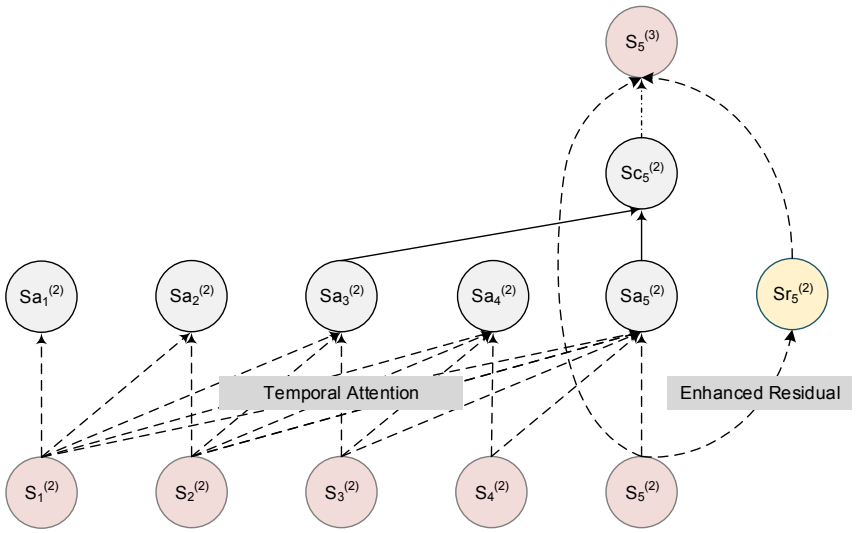


图 3-7: 网络结构单元

其中由 $S_1^{(2)}, S_2^{(2)}, \cdots, S_5^{(2)}$ 到 $S_5^{(3)}$ 的计算分为时序注意力层计算、时序卷积层计算、残差层计算和整合计算四步，下面结合上述符号进行介绍:

1. 时序注意力（Temporal Attention，TA）层是结合注意力机制的关键，本层应用具有序列属性的 Self-Attention，这里的序列属性是通过对图 3-5 中的权重矩阵取下三角矩阵，进而再做归一化得到的，对于下三角矩阵如何屏蔽未来特征，下文会做进一步解释，这里我们设时序注意力层为 **TA**，经过此层的计算我们得到：

$$Sa_5^{(2)} = \mathbf{TA}\left(S_{1:5}^{(2)}\right) \tag{3-11}$$

其中 $S_{1:5}^{(2)} = \{S_1^{(2)}, S_2^{(2)}, S_3^{(2)}, S_4^{(2)}, S_4^{(2)}\}$，$Sa_5^{(2)}$ 作为中间变量，包含了当前第 2 层第 5 个序列点前的加权序列特征，同时在计算过程中，我们得到权重矩阵 $Wa_5^{(2)}$。此层在图 3-7 中用下半部分的虚线和灰色圆圈表示；

2. 时序卷积（Temporal Convolution，TC）层的作用在于对加权后的特征做进一步整合，这里的卷积操作即为前面 3.2.1 讲到的时序卷积计算过程，此处为卷积核与时序注意力层输出的计算：

$$Sc_5^{(2)} = \mathbf{TC}\left(Sa_3^{(2)}, Sa_5^{(2)}\right) \tag{3-12}$$

此时核大小为 $k = 2$，扩展系数为 $d = 2$，则此层的感受野大小为 $(k-1) \times 2^{l-1} = (2-1) \times 2^{2-1} = 2$，但因为时序卷积层的叠加，使得感受野大小逐层加大。此层在图 3-7 中用上半部分的实线和灰色圆圈表示；

3. 强化残差（Enhanced Residual，ER）层利用 **TA** 得到的权重，对权重加和降维后与输入特征相乘，得到加权的特征，其具体操作在下文结合图示介绍，由此我们有：

$$Sr_5^{(2)} = \mathbf{ER}\left(S_5^{(2)}, Wa_5^{(2)}\right) \tag{3-13}$$

$Sr_5^{(2)}$ 的作用在于强化不同序列特征点之间的重要性差异。此计算操作在图 3-7 中用右侧黄色圆圈表示；

4. 上述三步分别得到了时序注意力层、时序卷积层和强化残差层的计算结果 $Sa_5^{(2)}, Sc_5^{(2)}, Sr_5^{(2)}$，这些结果的维度是相同的，所以这里我们将三个计算结果采取直接加和的方式进行整合：

$$S_5^{(3)} = Sa_5^{(2)} + Sc_5^{(2)} + Sr_5^{(2)} \tag{3-14}$$

$S_5^{(3)}$ 即为下一层的特征向量，此步骤的计算表示前馈序列网络的一个单元完整计算结束，整个网络结构就是由此计算单元堆叠组成，最后一层的输出便是包含时序信息的隐向量 h。

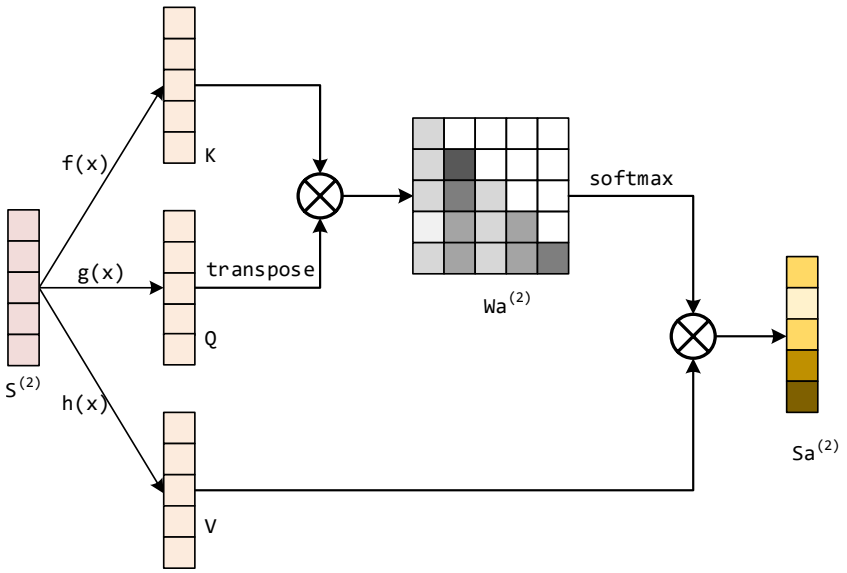


图 3-8: 时序注意力模块结构

其中时序注意力 **TA** 将历史时间步特征通过注意力思想做加权融合，其思想如图 3-8 所示，此模块与 Self-Attention 的区别在于对权重矩阵的处理。为解释两者权重处理方式对加权序列特征的影响，我们假设输入为 $S = [s_1, s_2, s_3]^T, s_i \in \mathbb{R}^p$，经过三个非线性函数 $f(x), g(x), h(x)$ 的后变换为 $K = [k_1, k_2, k_3]^T$，$Q = [q_1, q_2, q_3]^T$ 和 $V = [v_1, v_2, v_3]^T$，则有：

$$W = K \cdot Q^T = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \cdot \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} = \begin{bmatrix} k_1q_1 & k_1q_2 & k_1q_3 \\ k_2q_1 & k_2q_2 & k_2q_3 \\ k_3q_1 & k_3q_2 & k_3q_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \tag{3-15}$$

当此权重矩阵归一化并与 V 相乘后我们得到：

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} w_{11}v_1 + w_{12}v_2 + w_{13}v_3 \\ w_{21}v_1 + w_{22}v_2 + w_{23}v_3 \\ w_{31}v_1 + w_{32}v_2 + w_{33}v_3 \end{bmatrix} = \begin{bmatrix} k_1q_1v_1 + k_1q_2v_2 + k_1q_3v_3 \\ k_2q_1v_1 + k_2q_2v_2 + k_2q_3v_3 \\ k_3q_1v_1 + k_3q_2v_2 + k_3q_3v_3 \end{bmatrix} \tag{3-16}$$

从式 3-16（为简化演示过程，此处省略了归一化的 softmax 操作）可以看出，第一个时间步的结果 $[k_1q_1v_1 + k_1q_2v_2 + k_1q_3v_3]$ 包含了后面两个时间步的特征信息，而前文提到，时间序列建模时不可以利用未来时间步特征，所以为满足序列建模要求，需要把 t 之后的数据屏蔽，此处的操作为对权重矩阵的上三角做掩码处理，公式化表示为：

$$\begin{bmatrix} w_{11} & 0 & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} k_1q_1v_1 \\ k_2q_1v_1 + k_2q_2v_2 \\ k_3q_1v_1 + k_3q_2v_2 + k_3q_3v_3 \end{bmatrix} \tag{3-17}$$

上式 3-17 可以看出，这种方式解决了未来时间特征向当前时间步泄露的问题。通常对于此时得到的矩阵还要除以维度的平方根，进而使用 softmax 函数做归一化处理，即：

$$Wl_{i,j} = \begin{cases} (W)_{i,j}, & \text{if i} \geq \text{j} \\ 0, & \text{if i} < \text{j}, \end{cases} \tag{3-18}$$

$$Wa_{i,j} = \frac{e^{-\frac{Wl_{i,j}}{\sqrt{p}}}}{\sum_{j=1}^{t} e^{-\frac{Wl_{i,j}}{\sqrt{p}}}} \text{ for } j = 1, 2, \ldots, t \tag{3-19}$$

其作用在于原始权值方差较大的，而归一化后的权值会让反向传播过程中的梯度更稳定。最后求得的权重矩阵为下三角矩阵，权重矩阵与 V 相乘后得到时序注意力层的输出：

$$Sa_i = \sum_{j=1}^{t} Wa_{i,j} \cdot V_j \tag{3-20}$$

其中 $i = 1, 2, \cdots, t, j = 1, 2, \cdots, t$。

强化残差的作用在于两方面，一方面结合残差网络的优势，避免因层数加深时浅层信息的淡化消失，另一方面利用时序注意力中得到的权值增大输入序列特征中不同序列点的差异性，其模块结构如图 3-9所示。计算方法为首先对权值 Wa 我们在列的方向上求和，得到 t 维权值向量 w：

$$w_i = \sum_{j=1}^{t} Wa_{i,j} \tag{3-21}$$

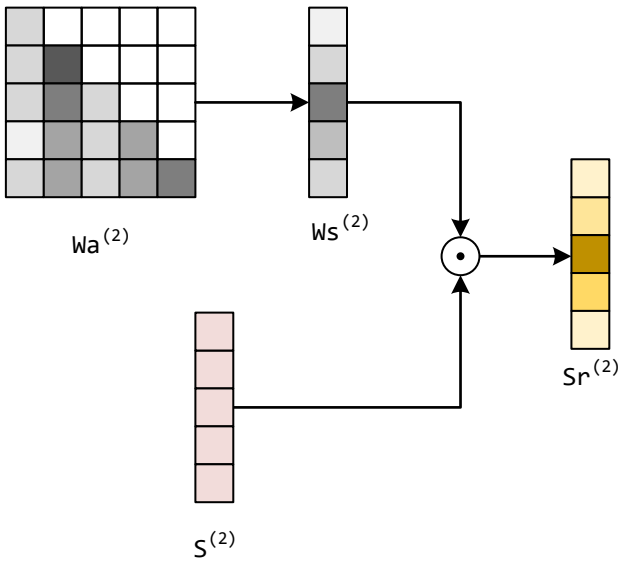


图 3-9: 强化残差模块结构

其中 $i = 1, 2, \cdots, t, j = 1, 2, \cdots, t$，再将权值与 V 相乘，得到强化残差：

$$Sr_i = S_i \times w_i \tag{3-22}$$

其中向量乘法应用到代码框架中的广播功能[①]。

以上就是前馈序列网络的具体网络结构，时序卷积和注意力机制以及残差网络等的结合使得网络在序列特征提取和模型训练上都取得较好的效果，在解决序列预测问题上也具有一定优势。

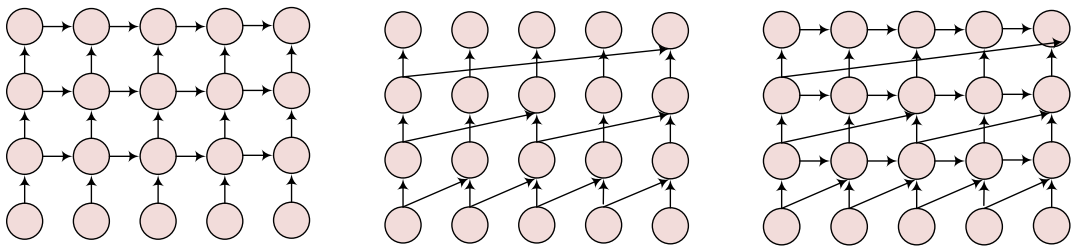
对于模型的有效性，我们从特征信息的流向上给出一种直观的解释，图 3-10 展示了 RNN、TCN 和 FSN 的信息流向，RNN 的特征信息主要在层内向后流动，而 TCN 主要利用卷积结构向上流动，FSN 结合了以上两者的流动方式，让信息向后和向上同时流动，这使得深层的隐向量包含更丰富的历史特征信息。

3.3 实验与分析

为验证前馈序列网络整体的效果和各个模块的作用，以及各种超参数对于变化的敏感程度，本节结合大量实验结果，给出相关分析。

为了量化的比较模型的效果，我们需要选择一些度量指标，常用的指标包

[①]https://numpy.org/doc/stable/user/basics.broadcasting.html



(a) RNN (b) TCN (c) FSN

图 3-10: 各种网络结构信息流向

括均方差（Mean Squared Error, MSE）、平均绝对误差（Mean Absolute Error, MAE）、标准化误差（Normalized Deviation，ND）、对称百分比误差（Symmetric Mean Absolute Percent Error，SMAPE）和标准化均方误差（Normalized Root Mean Squared Error，NRMSE）。它们的计算公式如下：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3-23}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3-24}$$

$$ND = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i|} \tag{3-25}$$

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{2\,(y_i - \hat{y}_i)}{y_i + \hat{y}_i}\right| \tag{3-26}$$

$$NRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\frac{1}{n}\sum_{i=1}^{n}|y_i|} \tag{3-27}$$

其中 MSE 和 MAE 都直观反映预测值与真实值的误差，当 MAE 为 0 时，得到完全拟合的模型，而 MAE 越大，表示误差越大，反之则误差小，但 MAE 和 MSE 有一定的局限性，就是它受极端时序值影响较大，同时还受度量大小的影响，不同度量的时序值无法相互比较。ND 相比 MAE 而言添加了真实值的求和项，相当于对 MAE 去除不同度量引起的差异，减弱了度量对评价指标的影响，更加关注预测值与真实值的百分比差异。RMSE 为对 MSE 开方，和 MSE 的大小有很强的相关性，通过求 RMSE 与真实值均值的比值得到 NRMSE，它同样能够去除度量带来的影响，SMAPE 侧重反映相对于真实值和预测值两者的误差。下面的实验使用 MAE、ND、NRMSE 和 SMAPE 四个评价指标，这些指标值越小表示模型效果越好。

3.3.1 对比实验

本实验所用数据集为四个实际场景中采样得到的数据，分别来自于自行车租赁、电力消耗、网络流量和空气质量监测，下面简单介绍这四个数据集的数据分布情况：

1. **Share Bike**：此数据集为华盛顿州 2011 年到 2012 年自行车租赁系统的注册量，采样间隔为 1 小时，持续记录一年，总序列长度为 17380，我们利用过去 48 小时数据预测未来 24 小时的自行车租赁数量；
2. **TAS2016**：此数据集为澳大利亚能源市场运营商（Australian Energy Market Operator，AEMO）在塔斯玛拉雅州（TSA）州的用电量需求数据，采样间隔为 30 分钟，总共记录 2016 年 1 年时间，总序列长度为 17521，我们利用过去 24 小时数据预测未来 12 小时的用电量；
3. **Traffic Bits**：此数据集为欧洲服务器供应商（ISP）监测的网络流量数据，采样间隔为 5 分钟，持续记录 1 个月，总序列长度为 14772，我们利用过去 2 小时数据预测未来 1 小时的网络流量；
4. **PRSA**：此数据集为北京市环境监测中心在万柳地区的空气质量监测站的空气质量数据，采样间隔为 1 小时，记录了从 2013 年到 2017 年共 4 年的数据，总序列长度为 35064，我们利用过去 48 小时数据预测未来 24 小时的 PM2.5 值。

数据集统计信息见表 3-1。

表 3-1: 数据集统计信息

数据集	序列长度	采样间隔	历史步长	预测步长
Share Bike	17380	1 小时	48	24
TSA2016	17521	30 分钟	48	24
Traffic Bits	14772	5 分钟	24	12
PRSA	35064	1 小时	48	24

为使训练不受不同数据集度量影响，稳定训练过程，我们对所有时间序列

数据进行最大最小归一化，使序列数据归一化到 0 ~ 1 范围内，对预测结果再进行反归一化处理，得到预测结果，但评价指标均为已归一化时计算得到。

本节实验设计目标在于对比本文设计的FSN网络结构与机器学习算法SVR、RNN 类网络结构和原始 TCN 网络结构的效果，其中 SVR 算法采用 sklearn[①] 框架中的实现，RNN 类选择使用最多的 LSTM 算法，原始 TCN 算法为 FSM 模型实现的原始框架，TCN 为 FSN 去除时序注意力层和强化残差层后的框架部分。FSN 模型设计的初衷是利用前馈网络实现对循环网络结构的替代，因此对比结果能够验证前馈序列网络提取的特征在预测上的效果比循环网络用于预测的效果好，就说明我们模型的设计达到了目标。

表 3-2: FSN 网络和其他方法对比结果

Share Bike					TSA2016				
Model	MAE	ND	NRMSE	SMAPE	Model	MAE	ND	NRMSE	SMAPE
SVR	0.140	0.725	0.974	0.855	SVR	0.111	0.212	0.266	0.216
LSTM	0.065	0.542	0.901	0.997	LSTM	0.099	0.206	0.261	0.194
TCN	0.066	0.596	0.837	0.743	TCN	0.083	0.172	0.211	0.171
FSN	**0.050**	**0.449**	**0.677**	**0.738**	FSN	**0.049**	**0.103**	**0.132**	**0.102**
Traffic Bits					PRSA				
Model	MAE	ND	NRMSE	SMAPE	Model	MAE	ND	NRMSE	SMAPE
SVR	0.211	0.494	0.671	0.513	SVR	0.070	0.828	1.054	0.821
LSTM	**0.048**	**0.116**	**0.157**	**0.152**	LSTM	0.052	0.551	0.778	0.634
TCN	0.100	0.242	0.288	0.359	TCN	0.050	0.529	0.780	0.745
FSN	0.063	0.152	0.190	0.199	FSN	**0.021**	**0.223**	**0.331**	**0.327**

实验结果如表 3-2 所示，其中粗体标出了某数据集下最优的预测结果。通过多种方法的实验结果对比可以看出，TCN 已经在多数情况下达到了循环网络的预测效果，TCN 作为前馈网络结构已经展现出其对序列数据的建模优势，但 LSTM 利用强大的序列特征提取能力，仍然在几个指标上优于前馈模型，在 Traffic Bits 数据机上的效果优于其他模型，相比较而言，传统算法的建模能力黯然失色，因为本文选取的数据集都带有协变量，但是传统算法和一般机器学习算法无法有效利用协变量提供的辅助信息，神经网络方法的优势相对明显。本文提出的 FSN 在多数任务上的预测效果有所提升，大部分情况下它具有相对较好的效果，它的目标在于避免循环网络的缺点，但从效果上看，在某些数据分布

[①]https://scikit-learn.org/

下，它的预测效果相较 LSTM 仍然有所欠缺，但 FSN 的前馈结构克服了循环网络的梯度消失等问题，而且训练成本不会随着序列的长度增加而大幅增加，在敏感性分析部分我们设计实验对 FSN 和 LSTM 在随着输入序列增加的情况下训练耗时进行对比，探究 FSN 在长序列输入下的效果。

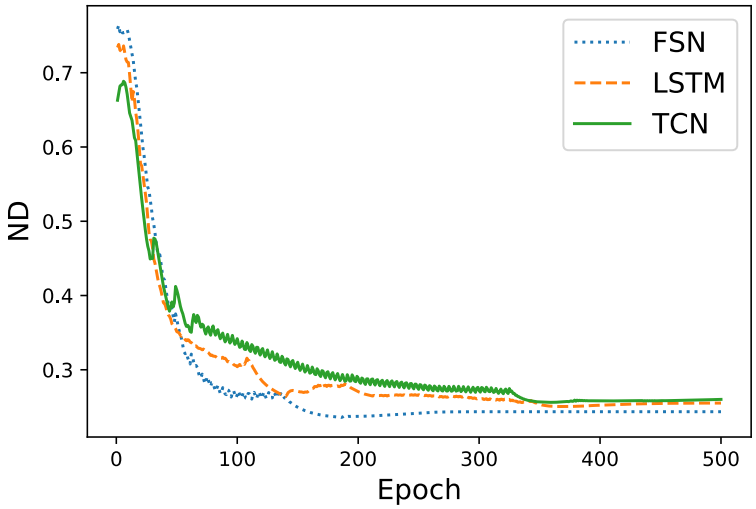


图 3-11: FSN, LSTM 和 TCN 的 ND 指标变化

另外我们对 FSN，LSTM 和 TCN 的收敛速度进行比较，图 3-11 为训练到 500 轮时验证集的 ND 指标变化情况，可以看出三个模型都趋向于收敛，但 FSN 的 ND 在收敛时是最低的，我们用学习率递减的方式控制模型收敛，当验证集的损失在 50 轮训练内不减小，我们就对原学习率减小一倍。图中可以看出，FSN 的收敛速度较快，在第 200 轮后基本开始收敛，而 LSTM 和 TCN 在 350 轮后才开始收敛稳定，由此验证了 FSN 在训练的收敛速度上要快于循环网络，这也是我们设计 FSN 的目标之一。

3.3.2 消融实验

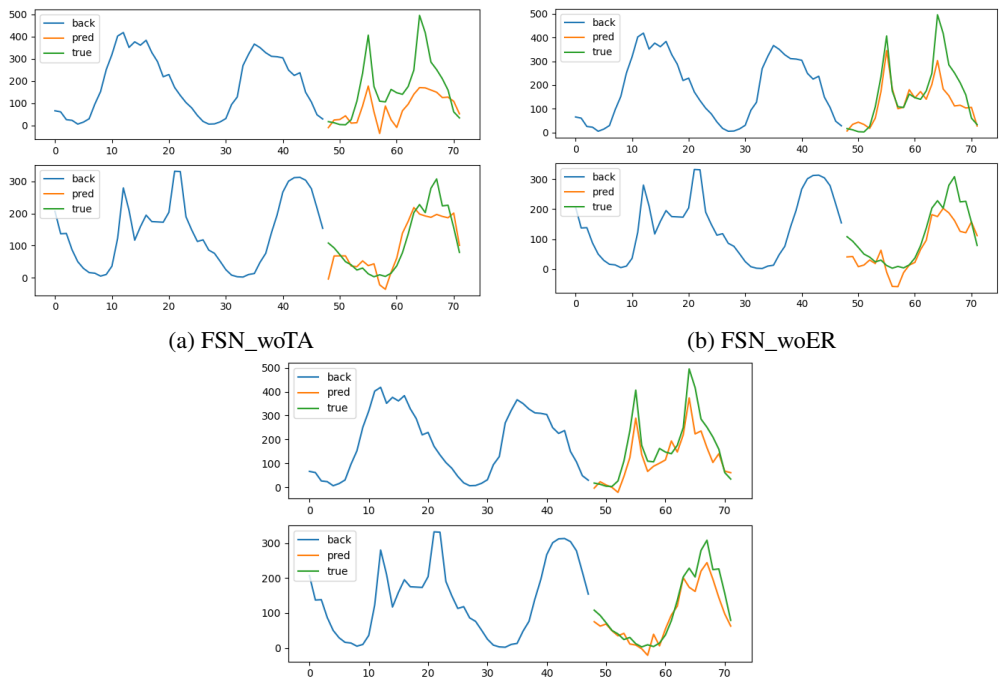
为验证模型中时序注意力层和强化残差层的效果，我们设计以下消融实验。因为强化残差层依赖于时序注意力层的权重矩阵，所以当没有时序卷积层时，强化残差层也不可用，我们定义完整的网络为 FSN，无强化残差层的网络为 FSN_woER，无时序注意力层的则为普通的时序卷积结构 TCN，我们表示为 FSN_woTA。实验数据集为 Share Bike。

消融实验结果如表 3-3 所示，可以看出，在没有时序卷积层和强化残差层

表 3-3: 消融实验结果

Models	ND	NRMSE	MAE
FSN_woTA	0.5959	0.8373	0.0660
FSN_woER	0.5586	0.7783	0.0618
FSN	0.4485	0.6765	0.0496

时，模型效果明显下降，此时的 FSN 网络相当于原始的 TCN 网络，因为缺少注意力机制对于不同序列点的特征加权，使得其对不同序列点的特征利用程度不够高，从上面对比试验的结果也可以看出，原始 TCN 结构的预测效果较 FSN 有些逊色。但时序注意力机制的加持使得网络效果明显提升，对历史时间步特征的加权发挥了注意力机制的特长，让历史时间步的特征得到充分利用，但多种网络结构容易造成梯度在传导过程中变小，不利于模型训练和收敛，FSN_woER 和 FSN 的对比证明了残差结构的有效性，另外强化残差还能够加强不同时间步的特征重要性差异，这也是提升模型预测效果的重要方面。



(a) FSN_woTA (b) FSN_woER

(c) FSN

图 3-12: 消融实验结果序列图

图 3-12 为三种情况下的预测结果和真实序列值曲线。可以直观看出模型对于时间序列的拟合效果有所提升。两个序列都有突变情况，前两者对突变的强

度未有效预测，第一个对于序列趋势预测仍然欠佳，基于 FSN 模型不仅预测到了波动形态，其对波动的时间和强度也比前者预测的好。

3.3.3 敏感性分析

本节目的在于设计 FSN 对于输入长度的敏感性实验，RNN 类算法的痛点之一在于当输入长度增加时，训练时间会变长，而且长度的增加会增加反向传播时的梯度消失和梯度爆炸风险，我们与 LSTM 进行对比，它通过门结构一定程度上解决了梯度消失的问题，我们比较 FSN 和 LSTM 在输入序列长度增加时，两者的训练时间变化情况，以此验证 FSN 模型对长序列输入的训练性能。本节实验所用数据集为 Share Bike。

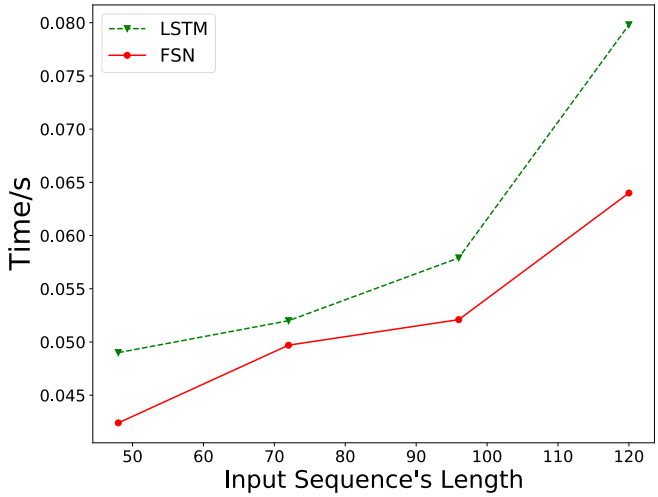


图 3-13: 不同输入长度模型耗时比较

由图 3-13 可以看出，随着输入长度的增加，循环网络 LSTM 的训练耗时相比 FSN 要多，这是由于循环网络本身的递归计算过程，时间步的长度增加就会使得循环网络单元的循环计算次数增加，而且必须要上一个循环单元计算完才能计算下一个单元，无法并行计算。虽然现在已有改进版的循环网络针对此问题做出优化[55]，但前馈网络对循环网络的替代则从本质上消除了这个问题，对于多个时间步的输入，前馈序列网络可以一次性对所有时间步的特征进行计算，现在的深度学习框架 tensorflow① 和 pytorch② 都实现了卷积等操作的加速，有利于模型并行训练和预测，也充分发挥了强大的算力优势。此实验验证了 FSN 网

①https://www.tensorflow.org/
②https://pytorch.org/

络可以在输入长度变长的情况下保持训练效率，表现出比循环网络更优的性能。

3.4 本章小结

本章主要提出了一种基于时序卷积、注意力机制和残差网络思想的前馈序列网络模型 FSN。此模型的目的在于模拟循环网络所特有的长时记忆和序列属性建模能力，试图使人们对于序列建模的基本模块有新的选择，FSN 结合时序卷积层可以增大感受野的能力，利用注意力机制克服循环网络对于历史序列点的临近倾向性，同时强化残差的设计增加了输入序列不同序列点的差异性，对于模型的设计充分提取了序列数据特有的属性，使模型相对于循环网络在训练效率和预测效果上都有所提升。

第四章　基于 DTW 的 MS-DTWI 损失函数

监督学习方法的构成分为模型、策略和算法三个要素[16]，前文讲到的网络结构设计属于模型部分，而对于时间序列预测任务而言，我们还可以从策略的角度来提升预测准确度。模型定义好了假设空间，对于时间序列预测任务，我们应该结合时间序列特有的属性来设计策略，即通常说的损失函数，在假设空间中找到最优模型。本章先分析现有时间序列预测损失函数的优缺点，从而引出我们设计的基于 MS-DTW 的损失函数 MS-DTWI，最后设计实验验证模型效果。

4.1　现有损失函数的局限性

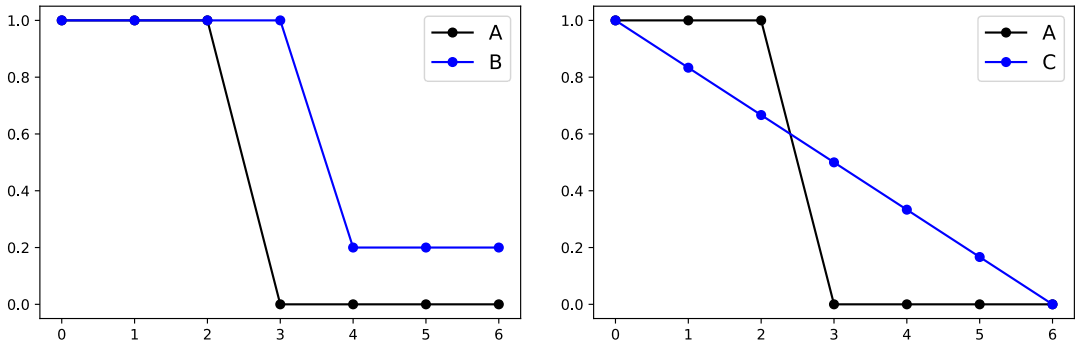
对于时间序列预测任务，在模型从传统统计学到现在基于神经网络的发展过程中，预测的目标函数通常都是最小化均方误差（式 3-23）或平均绝对误差（式 3-24），对应的损失函数即均方误差损失（Mean Squared Error Loss, MSELoss）和平均绝对误差损失（Mean Absolute Error Loss, MAELoss）。MSELoss 的函数公式中平方计算决定它在 y_i 和 $\hat{y}_i$ 的差值大于 1 时，会增大误差，而小于 1 时会减小误差，这使得它更倾向于惩罚误差较大的序列点，给予更多的关注，但是这也使得它对离群点比较敏感，从而牺牲模型整体性能。对于这一缺陷，MAELoss 可以避免，因为它对损失的计算采用绝对值的方法，对任何误差的惩罚力度相同，不会倾向于离群点，从而更具包容性，但是 MAELoss 缺少梯度的变化，不利于模型的学习和收敛。

Huber 损失函数（HuberLoss）对 MSELoss 和 MAELoss 进行融合，其公式如下：

$$L_\delta(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & |y_i - \hat{y}_i| \le \delta \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2, & |y_i - \hat{y}_i| > \delta \end{cases} \tag{4-1}$$

它包含一个超参数 δ，它的大小表示 HuberLoss 对 MSELoss 和 MAELoss 的侧重，当 $|y_i - \hat{y}_i| \le \delta$ 时，HuberLoss 等于 MSELoss，当 $|y_i - \hat{y}_i| > \delta$ 时 HuberLoss

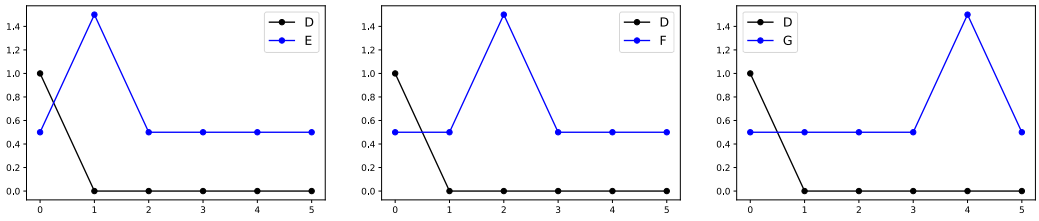
等于 MAELoss，它消除前两者的缺点并结合了二者的优点，使得在误差为 0 时损失函数也是可导的。



(a) A 和 B (b) A 和 C

图 4-1: 真实序列 A 与预测序列 B、C 的曲线

但是 HuberLoss 存在一定局限性，假设图 4-1 为真实时间序列 A 与两个预测结果时间序列 B 和 C 的曲线图，其中 A 与 B 的 HuberLoss 为 0.23，A 与 C 的 HuberLoss 为 0.21，由此损失推断 C 的预测效果比 B 好，然而对比图 4-1a 和图 4-1b 显然可以看出，前者的预测结果更好，因为它拟合了曲线 A 的弯折形态，而 C 的预测结果为一条斜线，无法体现序列的波动属性，设想在股票价格预测中，如果未能预测到近期的价格波动，很可能错失买卖时机，造成亏损失误。



(a) D 和 E (b) D 和 F (c) D 和 G

图 4-2: 真实序列 D 与预测序列 E、F 和 G 的曲线

可以看出 HuberLoss 在时间序列的曲线形态学习上存在短板，针对这一问题，一种有效的解决方案是引入动态时间规整（Dynamic Time Warping，DTW[56]），它作为时间序列相似度的度量指标，能够一定程度上表征序列的预测效果，DTW 越小表示序列相似度越高。对于上述案例，A 与 B 的 DTW 为 1.40，而 A 与 C 的 DTW 为 1.83，可以看出它更倾向与对序列形态的关注，符合我们的预期要求。DTW 给我们提供了使模型关注序列形态的思路，但在序列延时性的关注上，DTW 却不尽人意。假设真实时间序列 D 与其三个预测结果序列 E、F 和 G 如

图 4-2 所示，三个预测结果的 HuberLoss 均为 0.67，DTW 均为 5.5，但是显然可以看出，序列 E、F 和 G 的预测结果延时越来越大，此时 DTW 和 HuberLoss 均无法给出侧重。延时性对于实际场景也极其重要，在上述提到的股票价格预测场景下，即使模型成功预测到了波动会出现，如图 4-2 中一样，均预测到了一个波动，但未能成功预测到波动的时间点，最后还是会错过买卖的时机。

综合上述分析我们发现，对于时间序列预测的策略选择，我们不仅要考虑序列的形态（波动性），还要考虑到时延（延时性）的重要性。本文针对序列预测的上述两个关键问题设计了一种基于多尺度 DTW 的损失函数，同时解决了波动性和延时性的问题。

4.2 MS-DTWI 损失函数

基于多尺度 DTW 的损失函数（Multi-scale DTW with Temporal Distortion Index, MS-DTWI）的核心是多个规整窗口尺寸 DTW 的结合，为更好地对模型进行阐述，本小节从 DTW 的基本思想、不同规整窗口尺寸所指含义，到多尺度 DTW 结合的顺序递进式地讲解，阐明 MS-DTWI 的设计思想。

4.2.1 DTW

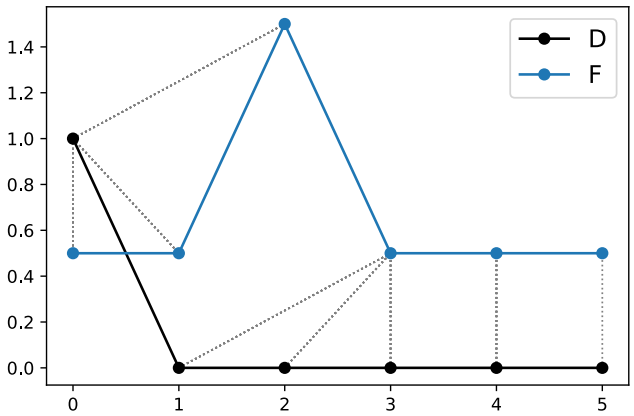
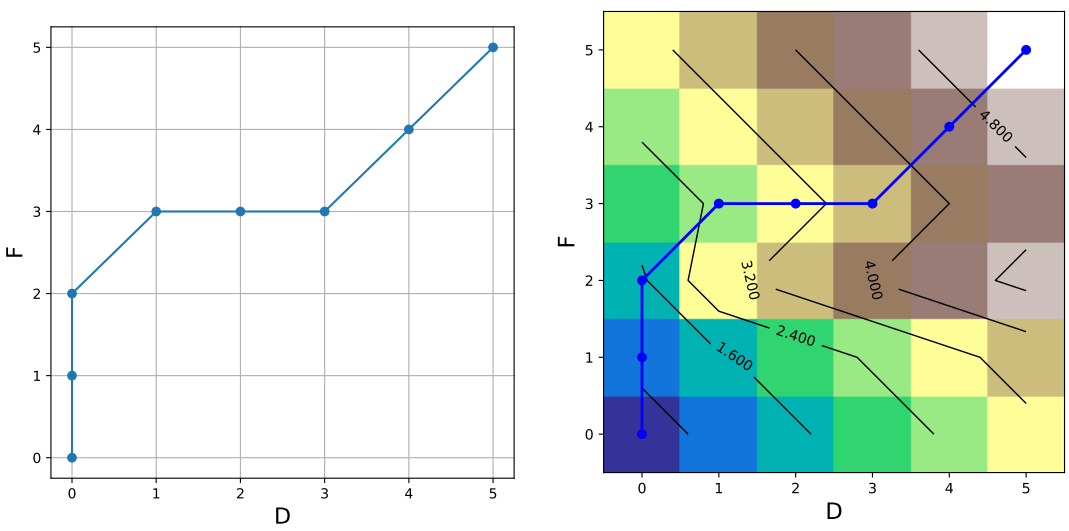


图 4-3: D 和 F 的序列点对应关系

上节提到 DTW 是用于度量时间序列相似度的指标，它通过动态规划算法先找到两个时间序列中序列点之间的对应关系，把两个序列规整对齐，此处所指的对应关系不再是同一时间点上两个序列数值的一一对应，而是会出现“一

对多”或者“多对一”的情况。图 4-3 为序列 $D = \{1, 0, 0, 0, 0, 0\}$ 和序列 $F = \{0.5, 0.5, 1.5, 0.5, 0.5, 0.5\}$ 的对应关系，序列 D 的第一个点对应了序列 F 的前三个点，通过这种对应关系，可以体现出两个序列形态上的相似性，比如图中两个序列的波峰和波峰后的下行位置成功对应。



(a) (b)

图 4-4: D 和 F 的序列点对应矩阵

上述对应关系用矩阵图表示为图 4-4a，在矩阵图中，这种对应关系可以看作图中从 (d_0, f_0) 到 (d_5, f_5) 的一条路径，想要得到这条路径，要求这条路径 P $(P = \{p_0, p_1, \cdots, p_k\}, p_l = (i_s, j_s))$ 须是众多路径中距离和最小的一条，这里距离和为序列 D 和序列 F 对应点距离的累加和。各序列点的距离度量为欧式距离，序列间所有点距离组成点距离矩阵 $D, D_{i,j} = |d_i - f_j|, i, j \in \{0, 1, 2, 3, 4, 5\}$，本例中 D 和 F 的点距离矩阵 D 如下：

$$D = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \tag{4-2}$$

为了和图 4-4 中点对应，矩阵中左下角为 $D_{0,0}$，右上角为 $D_{5,5}$，由 $D_{0,0}$ 到 $D_{5,5}$ 的路径有无数条，为了找出表示对应关系的路径，需要路径上经过的点满

足如下条件：

- 单调性：$i_{s-1} \leq i_s$ 且 $j_{s-1} \leq j_s$。单调性保证了路径向着终点前进，不会出现后退的情况；
- 连续性：$i_{s-1} - i_s \leq 1$ 且 $j_{s-1} - j_s \leq 1$。连续性保证路径连续，不会出现跳跃前进情况；
- 边界条件：$i_0 = 0, i_5 = 5$ 且 $j_0 = 0, j_5 = 5$。此条件表示路径从左下角开始，到右上角结束；
- 规整窗口：$|i_s - j_s| \leq w, w \geq 0$ 其中 w 为规整窗口大小。用于限制路径中的点 (i_s, j_s) 与从 $(0,0)$ 到 $5,5$ 对角线的水平或垂直距离不超过 w。

由于单调性和连续性的限制，使得下一个点只能是上一点的右方一点 $\rightarrow$、上方一点 $\uparrow$ 或右上方一点 $\nearrow$。结合其他条件可知，求解最佳路径是一个动态规划问题，其状态转移方程如下：

$$L_{\min}(i,j) = \min\{L_{\min}(i,j-1), L_{\min}(i-1,j), L_{\min}(i-1,j-1)\} + D(i,j) \quad (4\text{-}3)$$

其中 $D(i,j)$ 为矩阵 D 中的点 $D_{i,j}$，$L_{\min}(i,j)$ 表示从起点 $D_{0,0}$ 到 $D_{i,j}$ 的最小距离和。经过计算可以得到图 4-4b 所示的距离和密度图，简化后的序列点间对应关系矩阵为图 4-4a。但是，由于式 4-3 中最小值函数 min 的存在，使得动态规划的求解方程不可导，为了利用 DTW 度量作为策略函数来引导模型学习，需要改进 DTW 从而具有可导属性，从而可以应用在神经网络的端到端模型中，实现梯度的反向传播。Cuturi et al. (2017) 利用平滑函数的方式，用平滑最小化函数（如式 4-4）替代 min 函数，提出 γ-soft-DTW 实现了这一目标。

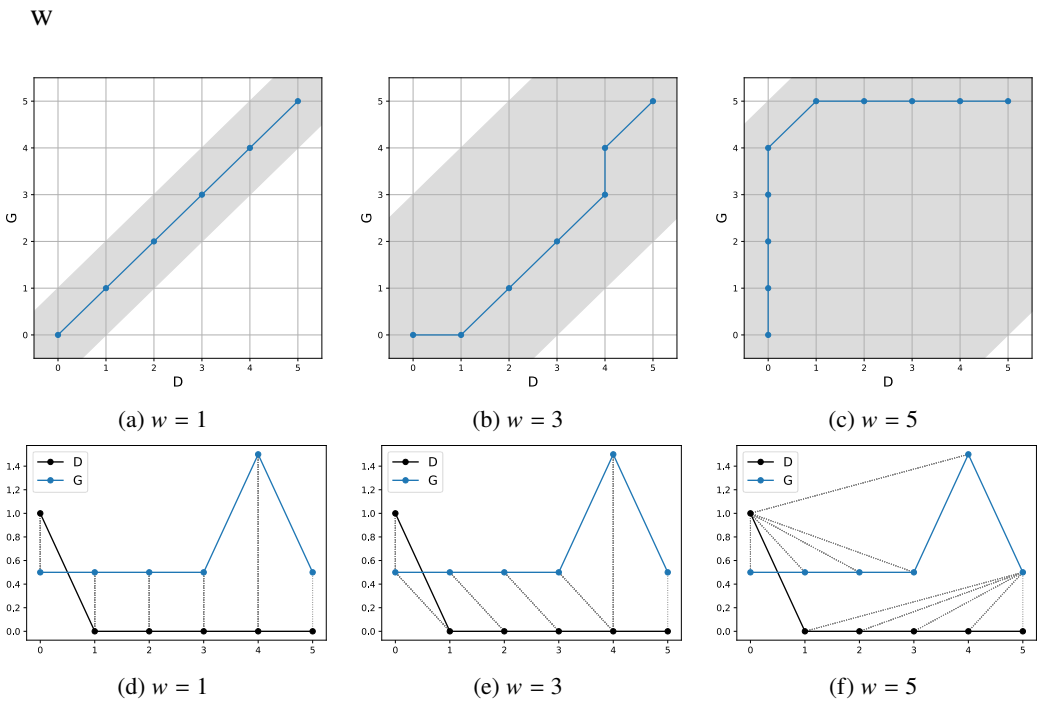
$$\min^{\gamma}\{a_1, \ldots, a_n\} := \begin{cases} \min_{i \leq n} a_i, & \gamma = 0 \\ -\gamma \log \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0 \end{cases} \quad (4\text{-}4)$$

其中 γ 为平滑系数，当 $\gamma = 0$ 时，γ-soft-DTW 等于原始 DTW，当 $\gamma > 0$ 时，它越小则结果越接近最小值。

目前已经解决了 DTW 作为策略函数应用的问题，但正如本章开篇所说，原始 DTW 的应用时具有一定局限性的，下面我们利用上述铺垫的内容作为基础，

阐述 Multi-scale DTW 的原理。

4.2.2 MS-DTW



(a) $w = 1$ (b) $w = 3$ (c) $w = 5$

(d) $w = 1$ (e) $w = 3$ (f) $w = 5$

图 4-5: 不同规整窗口下序列对应关系

在求解 DTW 寻找最短路径过程中涉及到“规整窗口”的概念，对于这个概念的直观理解如图 4-5 所示，图中所示为序列 D 和序列 G 在不同规整窗口大小下的对应关系。图 4-5a 和图 4-5d 表示当窗口大小为 1 时，DTW 退化为两条序列的欧式距离和，随着窗口增大，序列点的对应关系发生变化，常用的 DTW 度量通常取最大规整窗口，即原理上序列中第一个点可以最远可以对应到另一条序列的最后一个点。本文将不同规整窗口大小称为不同尺度，后文尺度的概念即指窗口大小，尺度为 u 的 DTW 表示为 $\mathrm{DTW}^u, u \in \mathbb{Z}^+$。

可以看出，欧式距离是 DTW 的特殊情况，即等价于 DTW^1。不同尺度的 DTW 表示序列不同程度的对应关系，而将这种对应关系进行融合可以用于解决 4.1 提到的 DTW 无法表征序列 E、F 和 G 的延时差异问题。由此我们定义多尺度 DTW （Multi-scale DTW, MS-DTW）形式如下：

$$\mathrm{MS\text{-}DTW}^n = \frac{1}{n}\sum_{i=1}^{n} DTW^{u_i}, u_i \in \mathbb{U} \tag{4-5}$$

其中 $1 \leq |\mathbb{U}| \leq \max\{len(S_1), len(S_2)\}$，$len(S_1)$ 表示序列 S_1 的长度，$\mathbb{U}$ 为不同尺度值的集合，通常我们指定不同尺度的个数 n，然后在 $[1, \max\{len(S_1), len(S_2)\}]$ 内等间距取 n 个尺度构成尺度值集合 $\mathbb{U}$，我们称 n 为尺度系数。

算法 1 计算多尺度 DTW 指标值

输入:

y 表示真实序列，长度为 k
$\hat{y}$ 表示预测序列，长度为 k
γ 表示平滑系数
u 表示规整窗口

输出:

R 表示最短路径和矩阵，$R_{i,j}$ 为 $DTW\left(y_{1:i}, \hat{y}_{1:j}\right)$
γ-soft-DTW$(y, \hat{y})$，即 $R_{k,k}$

function DistanceCal$(y, \hat{y})$
  Initialize $D \leftarrow [0]_{i,j}, 0 \leq i, j < k$ // 记录序列点间距离
  for $i = 0 \rightarrow k-1$ **do**
    for $j = 0 \rightarrow k-1$ **do**
      $D_{i,j} = \sqrt{\left(y_i - \hat{y}_j\right)^2}$
    end for
  end for
  return D
end function

function DTW(D, u, γ)
  $R \leftarrow [\infty]_{i,j}, 0 \leq i, j \leq k+1$
  $R_{0,0} = 0$
  for $j = 1 \rightarrow k$ **do**
    for $i = 1 \rightarrow k$ **do**
      $r_1 = \frac{-R_{i-1,j-1}}{\gamma}$
      $r_2 = \frac{-R_{i-1,j}}{\gamma}$
      $r_3 = \frac{-R_{i,j-1}}{\gamma}$
      $r_{max} = max\left(r_1, r_2, r_3\right)$
      if $|i-j| \leq u$ **then**
        $r_{sum} = e^{r_1 - r_{max}} + e^{r_2 - r_{max}} + e^{r_3 - r_{max}}$
        $softmin = -\gamma * \left(\log\left(r_{sum}\right) + r_{max}\right)$
        $R_{i,j} = D_{i-1,j-1} + softmin$
      else
        $R_{i,j} = 0$
      end if
    end for
  end for
  return $R, R_{k,k}$
end function

由于式 4–5 的计算由 DTW 组成，原始的 DTW 是不可导的，所以无法求得

反向传播的梯度，这对于端到端的深度学习模型来说是不可以作为损失函数的，我们使用上文提到的 γ-soft-DTW 作为 DTW 的替代，γ-soft-DTW^u 表示规整窗口为 u 时的 DTW 值，我们简写为 DTW_γ^u，对应的多尺度 DTW 简写为 $\mathrm{MS\text{-}DTW}_\gamma^n$。

假设模型预测的结果序列为 $\hat{y}$，真实序列为 y，则 $\mathrm{DTW}_\gamma^u(y,\hat{y})$ 的计算过程如算法 1 所示。我们先通过自定义的距离度量方法计算序列 y 和 $\hat{y}$ 的序列点间欧式距离，在得到距离矩阵 D 后，将其作为求解 $\mathrm{DTW}_\gamma^u(y,\hat{y})$ 过程中的距离矩阵，利用动态规划思想，结合平滑最小化函数实现状态转移方程：

$$R_{i,j}=\begin{cases} D(i,j)+\min^{\gamma}\left\{R_{i-1,j-1},R_{i-1,j},R_{i,j-1}\right\} & |i-j|\le u \\ 0, & |i-j|>u \end{cases} \tag{4-6}$$

从而求得最短距离和矩阵 R，矩阵中 $R_{i,j}$ 表示序列 $y_{1:i}$ 和序列 $\hat{y}_{1:j}$ 的 DTW 值，多个规整窗口的 DTW 值取平均后即得到 MS-DTW 。

算法 2 计算多尺度 DTW 指标的导数

输入:

y 表示真实序列，长度为 k
$\hat{y}$ 表示预测序列，长度为 k
γ 表示平滑系数
u 表示规整窗口

输出:

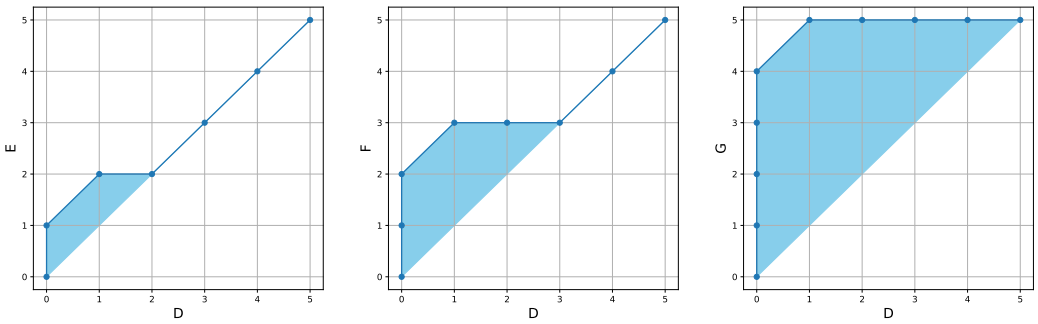
$\mathrm{DTW}_\gamma^u(y,\hat{y})$ 对 $\hat{y}$ 的梯度：$\nabla_{\hat{y}}\mathrm{DTW}_\gamma^u(y,\hat{y})$

$D \leftarrow DistnaceCal(y,\hat{y})$ // 计算得到序列点间距离矩阵
$R, R_{k,k} \leftarrow DTW(D,u,\gamma)$ // 计算得到 DTW 值
$D' \leftarrow [0]_{i,j}, 0 \le i,j \le k+1$
$E' \leftarrow [0]_{i,j}, 0 \le i,j \le k+1$
$D'_{i,j} \leftarrow D_{i,j}, 1 \le i,j \le k$
$E'_{k+1,k+1} \leftarrow 1$
$R_{\cdot,k+1} \leftarrow -\infty$
$R_{k+1,\cdot} \leftarrow -\infty$
$R_{k+1,k+1} \leftarrow R_{k,k}$
for $j = k \to 1$ **do**
    for $i = k \to 1$ **do**
        $a = \exp\frac{1}{\gamma}\left(R_{i+1,j} - R_{i,j} - D'_{i+1,j}\right)$
        $b = \exp\frac{1}{\gamma}\left(R_{i,j+1} - R_{i,j} - D'_{i,j+1}\right)$
        $c = \exp\frac{1}{\gamma}\left(R_{i+1,j+1} - R_{i,j} - D'_{i+1,j+1}\right)$
        $E_{i,j} = E_{i+1,j}\cdot a + E_{i,j+1}\cdot b + E_{i+1,j+1}\cdot c$
    end for
end for
return E //$\nabla_{\hat{y}}\mathrm{DTW}_\gamma^u(y,\hat{y})$

对序列 MS-DTW 的计算是端到端模型结构中的前向算法过程，但端到端网络需要通过反向传播来调整模型参数，因此我们需要计算 MS-DTW 对预测序列 $\hat{y}$ 的梯度值 $\nabla_{\hat{y}}\text{MS-DTW}_\gamma(y,\hat{y})$，计算过程如算法 2 所示。

$\text{DTW}^u_\gamma(y,\hat{y})$ 对 $\hat{y}$ 梯度的求解利用了算法 1 对序列距离矩阵和 DTW 的求解函数。因为平滑最小化函数的引入，$\text{DTW}^u_\gamma(y,\hat{y})$ 对 $\hat{y}$ 的求解原理上可以通过深度学习框架的自动求导得到，但是时间复杂度较高，于是我们采用类似于 Blondel et al. (2016) 中对 ANOVA 核求导的方法，对算法 1 中的递归采用逆序的方式，一次遍历就可以得到最终的梯度。计算过程中还利用了前向算法中的距离和矩阵 R，将求导过程的时间复杂度控制在 $O(k^2)$。算法 2 中对中间变量 E 的推导过程参考 Cuturi et al. (2017)。

4.2.3 TDI



(a) D 和 E (b) D 和 F (c) D 和 G

图 4-6: 序列 D 与序列 E、F 和 G 的对应关系矩阵

对于图 4-2 中三个预测序列 E、F 和 G，他们的与序列 D 的对应关系矩阵如图 4-6 所示，从矩阵中标注的阴影区域可以看出，阴影区域越来越大，而从图 4-2 可知它们的时延逐渐变大，可以发现两者成正比关系，这正是另外一个基于 DTW 的度量指标 TDI（Temporal Distortion Index[59]）的核心思想，它同样可以用来度量序列间时延的大小。TDI 的计算核心部分在于计算阴影区域面积占整个路径空间面积的比例，如式 4-7 所示：

$$\begin{aligned} S_l &= \int_{i_l}^{i_{l+1}} \left(x - \frac{(x-i_l)\,(j_{l+1}-j_l)}{(i_{l+1}-i_l)} + j_l \right) dx \\ TDI &= \frac{2\sum_{l=1}^{k-1} |S_l|}{N^2} \end{aligned} \tag{4-7}$$

其中 i_l, j_l 为 DTW 中最小距离和对应路径上的点坐标 $p_l = (i_s, j_s)$，k 为路径上点的个数，N 为序列长度，N^2 为路径空间面积。

由上述对 TDI 的计算过程可以看出，TDI 也是不可导函数的，但 Guen et al. (2019) 给出了在端到端模型中的前向算法和反向传播中梯度求解的实现，本算法的实现参考其开源实现完成。

4.2.4 MS-DTWI 损失函数

计算序列 D 与序列 E、F 和 G 的 MS-DTW 和 TDI，并对比指标 MSE 和 DTW，可以得到表 4-1，从中可以看出，MS-DTW 和 TDI 能够对不同程度的时延进行区分，对时延大的预测给予更大的指标值。加之 DTW 对序列形态的关注，我们设计基于 MS-DTW 和 TDI 的损失函数，引导模型给序列形态情况和时延问题更多关注，惩罚时延大的预测结果，提升预测效果。

表 4-1: 序列 D 与 E、F 和 G 的三种度量指标

度量指标	D 与 E	D 与 F	D 与 G
MSE	0.67	0.67	0.67
DTW	5.50	5.50	5.50
TDI	0.10	0.37	1.67
MS-DTW(3)	0.99	1.08	1.25
MS-DTW(6)	1.03	1.10	1.25

本文采用加权的方式将 MS-DTW 和 TDI 结合起来，得到损失函数 MS-DTWI（**M**ulti-**S**cale **DTW** with Temporal Distortion **I**ndex），其计算方法如下：

$$\text{MS-DTWI} = \alpha \cdot \text{MS-DTW} + (1 - \alpha) \cdot \text{TDI} \tag{4-8}$$

其中 α 为侧重系数，表示对于 MS-DTW 的侧重程度，默认设为 $\alpha = 0.5$，其中 MS-DTW^n 中的 n 默认取 5。

MS-DTWI 损失函数的主要部分 MS-DTW 和 TDI 均在上文中给出详细介绍，对于 MS-DTWI 损失函数在端到端的深度学习模型中的应用，其前向计算和反

向传播过程为上述两者的结合，但对两者梯度的侧重程度由侧重系数 α 决定，同时在损失函数中的多个超参数也是影响损失函数效果的重要因素，下文通过实验和相关理论分析给出更多解释和验证。

4.3 实验与分析

为验证 MS-DTWI 损失函数的有效性，本节设计了与其他时间序列预测任务中常用的损失函数的对比试验和消融实验，实验所用数据集为上一章中提到的 Share Bike 数据集，评价指标为 MAE、MSE、ND、NRMSE 和 MS-DTW。其中 MS-DTW 用于表征损失函数对序列的形态和延时情况的约束效果。

4.3.1 对比试验

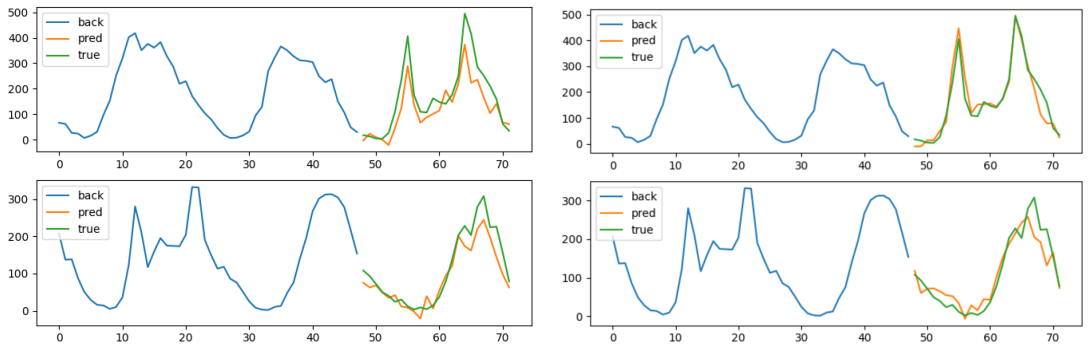
对于时间序列预测任务，人们常用的损失函数为 MSELoss 和 HuberLoss，从 4.1 的分析我们知道，后者因结合了前者的优势，相比之下应该具有更优的性能，本文提出的 MS-DTWI 损失函数在特殊的参数设置下，也近似于 HuberLoss，所以理论上我们的损失函数应该具有更优的策略约束能力。为验证上述结论，本节设计对比实验，其中网络框架部分选取本文在上一章中提出的 FSN 网络，另外我们与目前效果最好的时间序列预测损失函数 DILATE[60] 进行比较，对比学习的效果。神经网络的实现使用 Pytorch 深度学习框架，框架内集成了 MSELoss 和 HuberLoss（又称 SmoothL1Loss），DILATE 的实现使用其开源代码 ①，最终实验结果如表 4-2 所示。

表 4-2: 不同损失函数实验对比结果

损失函数	验证集			测试集			
	MAE	ND	NRMSE	MAE	ND	NRMSE	MS-DTW
MSELoss	0.063	0.334	0.473	0.051	0.461	0.632	0.060
HuberLoss	0.060	0.315	0.440	0.055	0.496	0.701	0.078
DILATE	0.066	0.350	0.500	0.058	0.523	0.772	0.086
MS-DTWI	**0.055**	**0.291**	**0.412**	**0.042**	**0.379**	**0.544**	**0.044**

① https://github.com/vincent-leguen/DILATE

从对比试验结果可以看出，在验证集上 HuberLoss 优于 MSELoss，而测试集上效果相反，即 HuberLoss 的经验风险相对较小，但结构风险较大，发生一定程度的过拟合，MSELoss 测试集的优秀表现说明其训练得到的模型泛化性能更优，而泛化性能的提升才是实际应用中我们更想看到的。DILATE 的预测效果差强人意，其主要超参数参考 DILATE 原文的实验设置，其原文中通过对其与其他损失函数的 Student t-test 判断预测效果，也表现出与 MSELoss 相近的预测准确度。相比之下，MS-DTWI 的预测效果优于以上两者，不仅在常用的 MAE 等指标上精度更高，在我们设计的可以反映形态和延时性情况的 MS-DTW 指标上也可以看出，MS-DTWI 对序列形态和延时性的关注对模型的学习和预测都起到了积极作用，泛化性能也有明显提升，为了更直观地看到拟合效果的差异，我们选取了其中两条预测曲线，如下图所示：



(a) MSELoss (b) MS-DTWI

图 4-7: 不同损失函数效果对比

图 4-7 展示了基于 MSELoss（图 4-7a）和基于 MS-DTWI（图 4-7b）的预测效果对比，可以看出，在使用 MS-DTWI 损失训练得到的模型预测两条曲线时，预测曲线相比 MSELoss 训练得到的模型预测曲线更少会出现时延情况，从第一条预测曲线可以看出，MS-DTWI 对两次波峰的出现时间预测更为精准，而 MSELoss 的预测对第一个波峰稍有延后，而对第二个波峰稍有提前。但从预测效果可以看出，损失函数仍然有待优化的点，图 4-7b 中对于波动的程度反应稍有不足，这说明模型对于序列的尺度范围未能有效预测，这也是当前时序预测面临的难点之一，我们为了聚焦形态和延时性问题，对输入数据做归一化处理，而图中曲线为反归一化后的结果，更大的量级放大了预测值的波动程度，因此图中看到抖动较多，这一问题将作为我们后续优化模型的主要关注点。

4.3.2 敏感性分析

MS-DTWI 损失函数中包含了三个主要的超参数，即 DTW_γ 中的参数 γ、侧重系数 α 和尺度系数 n，本节以实验为依据，分析三个超参数对模型的影响情况。

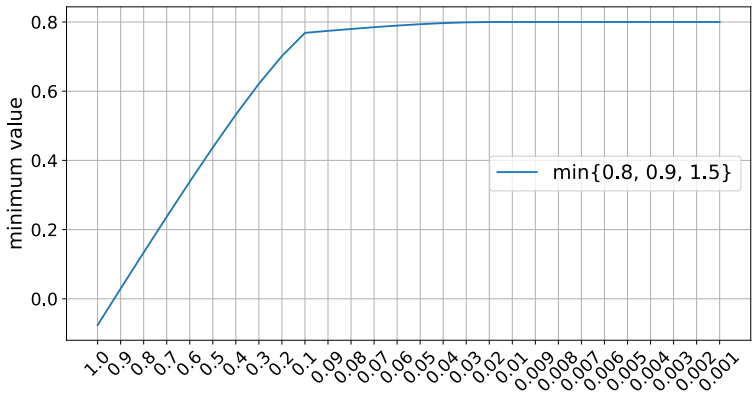


图 4-8: 不同 γ 时平滑最小化函数计算结果

DTW_γ 中的超参数 γ 由于平滑最小化函数的引入而产生，γ 的大小决定了其对最小值函数的近似程度，我们在大量的实验中总结发现，γ 的大小应该随着待比较的数字量级变化，下面的实验中待比较数字普遍在 1 附近变动，于是我们通过简单的样例测试检验对 γ 的选取原则。我们假设待比较的三个数字分别为 $\{0.8, 0.9, 1.5\}$，$\gamma \in \{1.0, 0.9, 0.8, \cdots, 0.1, 0.09, 0.08, \cdots, 0.01, 0.009, \cdots, 0.001\}$，28 个 γ 值分布在从 1.0 到 0.001 之间，图 4-8 为不同 γ 值时平滑最小化函数的计算结果，可以看出，对于此时的量级大小，选择 $\gamma = 0.1$ 就能够达到对最小化函数足够的近似了，为了验证此结论，我们设计了 γ 的消融实验，用 $\gamma \in \{0.0001, 0.001, 0.01, 0.1, 1.0, 5.0\}$ 训练模型。

图 4-9 为不同 γ 时模型的预测效果，为了清楚地展示对比效果，图中对评价指标做了最大最小归一化。结合图 4-9 和图 4-8 可以看出，虽然 $\gamma = 1.0$ 时，度量值甚至为负值，但对于模型的训练仍然有效（度量值为负的问题仍待解决[61]）。当 γ 位于 1.0 到 0.001 之间时，模型预测效果都较好，但当 $\gamma < 0.001$ 时，模型效果开始变差，所以 γ 的选取不能过小，在合适的度量位置就好，本实验中 $\gamma = 0.1$ 时相对更合适。此实验也验证了上述对 γ 选取的准则。因为反向传播算法中的梯度中包含 γ，因此我们认为其值过小会影响梯度的量级，从而造成波动，影响

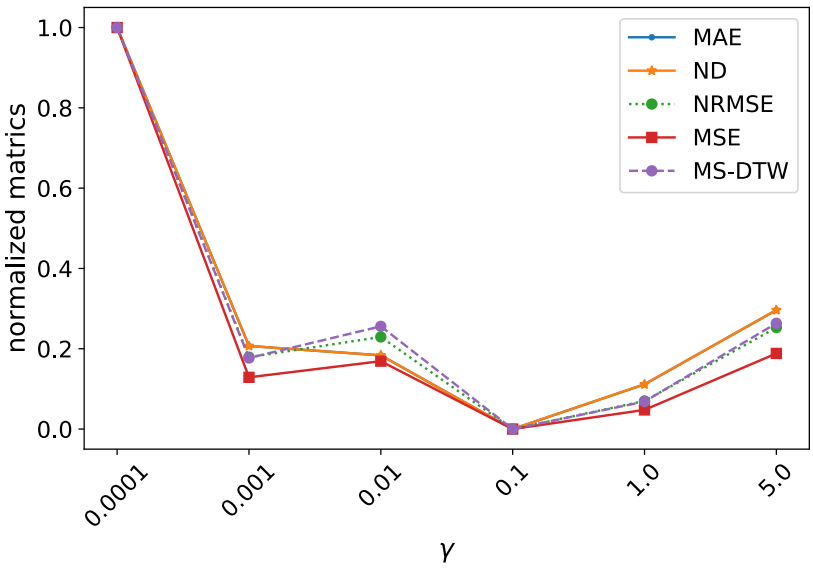


图 4-9: 不同 γ 时模型效果

训练效果，具体原因仍待继续深入研究。

侧重系数 α 在 $[0,1]$ 中取值，当 $\alpha = 0$ 时，表示 MS-DTWI 等价于单纯利用 TDI 指标作为损失函数，$\alpha = 1$ 时表示 MS-DTWI 只利用 MS-DTW 作为策略函数，两者都可以对模型做出引导，让模型更关注时间序列形态和延时性，我们设置侧重系数 α 的取值为 $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$。尺度系数 n 的大小表示 MS-DTW 所结合的不同尺度的数量，说明能够包容更多的时延情况，我们设置尺度系数 n 的取值为 $\{2, 4, 6, 8, 10\}$，通过实验找到尺度系数对模型的影响效果。

表 4-3: 不同尺度系数 n 时效果对比值

n	MAE	ND	NRMSE	MSE	MS-DTW
1	0.0503	0.4549	0.6649	0.0054	0.0697
2	0.0475	0.4293	0.5903	0.0043	0.0511
4	0.0420	0.3794	0.5436	0.0036	0.0445
6	0.0469	0.4236	0.6065	0.0045	0.0542
8	0.0485	0.4380	0.6079	0.0045	0.0512
10	0.0568	0.5132	0.6767	0.0056	0.0711

图 4-10 展示了不同尺度系数 n 下的模型预测效果（此时 $\alpha = 0.8$，评价指标同样采用最大最小归一化方式处理），当 $n = 1$ 时，表示损失函数为平方和损失与 TDI 度量损失的结合，预测效果并不理想，但随着尺度系数的增大，预测准

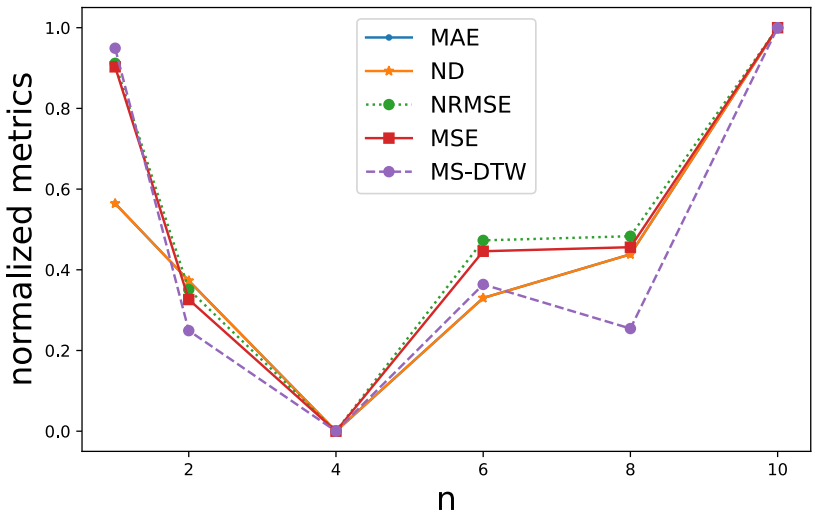


图 4-10: 不同尺度系数 n 时效果对比

确度开始提高，这说明 MS-DTW 要想达到对形态和时延较好的关注度，需要考虑多个尺度的 DTW。通过表 4-1 可以看到 MS-DTW 对不同尺度的指标值都具有相似的趋势，反映了随着尺度系数增大，MS-DTWI 损失函数对训练效果的提升，但我们同时发现，尺度系数不宜过大，可以看到当 $n > 4$ 时，训练效果开始下降，说明 n 值过大对模型训练并不会起到积极效果，因此对于不同时间序列分布情况，尺度系数还需要进行尝试选取。

尺度系数的选择和 α 值的选取都需要参考时间序列数据本身的分布情况，下面对于 α 的敏感性分析给出在不同 α 值下损失函数的效果差异。

表 4-4: 不同 α 时效果对比值

α	MAE	ND	NRMSE	MSE	MS-DTW
0.0	3.2749	29.5909	35.5991	15.5224	372.3373
0.2	0.0536	0.4839	0.6553	0.0053	0.0647
0.4	0.0486	0.4390	0.6374	0.0050	0.0633
0.6	0.0463	0.4182	0.6092	0.0045	0.0561
0.8	0.0420	0.3794	0.5436	0.0036	0.0445
1.0	0.0604	0.5455	0.7297	0.0065	0.0822

图 4-11 为 n=4 时不同 α 值的模型效果，可以看出，当 $\alpha = 0$ 时，即 MS-DTWI 只包含 TDI 时，效果大打折扣，随着 α 变大，即 MS-DTW 的权重逐渐增加，MD-DTWI 的效果越来越好，在 $\alpha = 0.8$ 时效果相对较好，但当 $\alpha = 1.0$

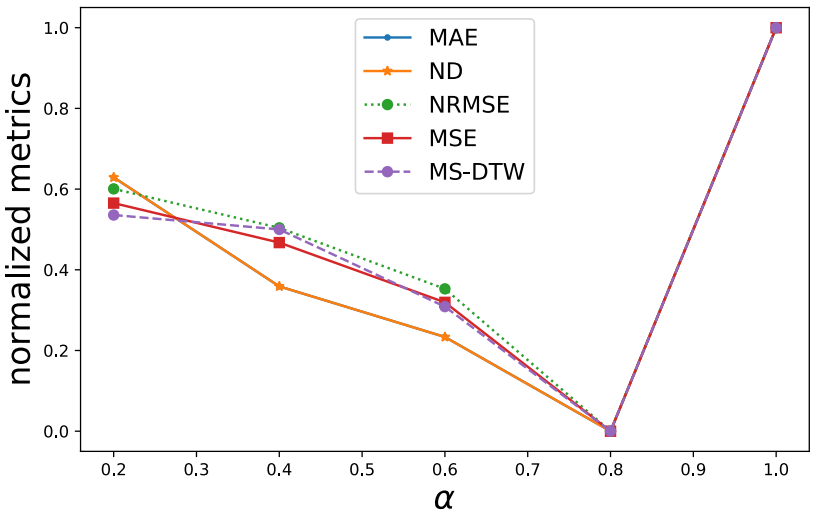


图 4-11: 不同 α 时效果对比

，即只有 MS-DTW 时，由表 4-4 和图 4-11 可以看出，各项指标的增大表示预测效果有所下降，所以单纯使用 MS-DTW 作为损失函数并不可行。从前文中的表 4-1 我们看出，在某些情况下，TDI 指标比 MS-DTW 方差更大，具有更好的差异化表现，辨识度更高，而图 4-11 和表 4-4 所体现出的结果让我们看出，此时 MS-DTW 对模型的影响更大，因此可以看出，α 值得选取也需要对不同值进行尝试，但是 α 值选为 0 和 1.0 对模型效果都未达到最好，可以看出两者缺一不可，通常我们选取 0.5 作为基准，此时对 MS-DTW 和 TDI 的结合比例相同，综合以上对结果的分析可知，MS-DTW 和 TDI 的结合才是最优方案。

4.4 本章小结

本章分析了现有时间序列预测损失函数的局限性，发现常用的 MSELoss 和 HuberLoss 未能对序列的形态和延时性有较好的关注，而 DTW 作为序列相似度度量标准在衡量两个时间序列的形态上具有更好效果，但是仍然存在对序列延时性表征不充分的缺点，我们从 DTW 的方法本质出发，结合多种不同尺度的 DTW 度量，设计了 MS-DTW，同时结合 TDI 度量指标，发挥两者的优势，设计了 MS-DTWI 损失函数，对于 DTW 计算过程不可导的问题，我们采用了 soft-DTW 的平滑化方法，使得损失函数顺利应用在基于深度学习的时间序列预测模型训练中，最后设计了对比实验验证 MS-DTWI 对时间序列预测性能的提升和超参数对损失函数的影响。

第五章　商品管理和销量预测系统

在实际生产生活中，有很多时间序列预测的应用场景，我们将本文提出的FSN 网络结构和 MS-DTWI 损失函数结合，用于解决时间序列预测问题，并应用于我们设计并实现的商品管理和销量预测系统。

5.1　相关背景

随着云计算的普及，不论是互联网企业还是传统制造业，都逐渐进入云时代。信息化程度已经成为决定企业竞争力的重要因素，集成的信息化平台为高效的内部管理、产品质量控制、生产流程安排和销售策略制定等方面提供了有力的信息支撑。

互联网行业和传统制造业还有所区别，互联网行业本就依托网络传输数据，大部分信息都在计算机中留有记录数据，所以问题主要在于分析挖掘数据的内在联系，得到有价值的信息。而制造业的商品数量通常较大，小到机械设备商品住房，大到食品饮料电子元件，不同数量级的产品，都面临相同的问题，即商品管理。人力成本的提高和云服务产业链的成熟催生了信息管理系统的诞生和发展，有效的管理企业产品，跟踪产能、库存和销售数据能够为企业生产带来诸多益处，如果没有一个集成化全面信息管理系统，很有可能造成库存不合理或供应关系混乱等现象。

大型连锁商超是典型的分散化经营企业，连锁型的属性决定了它要面临商品种类数量繁多、分店管理分散和库存销售数量难以平衡的问题，我们从这三方面入手，设计和实现了“金牛（Taurus）系统”，为企业提供一个可靠的信息管理系统，方便企业将商品库存数据、上下游商品流通数据和动态销售数据进行记录，及时同步到各个环节，为管理决策者提供参考。同时为了给企业管理者提供更有效的信息，我们不仅对已有商品信息进行管理，还利用历史销售数据和商品关联性等信息，对商品进行未来一段时间的销量预测，辅助管理者做出具有前瞻性的决策，降低运营成本，增加企业收益[62]。

5.2 系统目标

金牛（Taurus）系统的应用场景主要为大型连锁商超，其设计目标主要包括两方面内容，一方面是进行现有商品的管理；另一方面是商品数据的分析和预测。

商品管理是仓库管理者、代理经销商和线下门店都要面临的问题，不同环节都有着大量的数据，有效的信息管理可以给相关岗位的管理者提供极大便利，比如对于仓库管理者，产品采购到位或生产结束，货品运输到库时及时将商品信息记录到系统中，商品分发需求下达时，将商品出库信息同步系统中，不仅能够便利库存管理者，还能对上下游管理人员有所帮助，销售策略制定者也可以利用买卖流水数据结合系统的分析与预测能力来指定更有效的销售方案。

分析和预测是利用历史销售数据和商品之间的关联关系等，应用时间序列预测模型，做出未来一段时间的数据预测，辅助采购人员、代理经销商和相关管理者制定面向未来的销售决策。分析功能是对数据的挖掘，探索商品间的关联性；预测功能更多是对未来一段时间未知数据的合理估计，给出有利于决策的参考信息。由于系统尚未上线应用，为展示系统的主要功能，我们利用数据挖掘竞赛的公开数据集，将其导入到系统中，其在帮助完善系统流程的同时，还用于有验证上述时间序列预测算法在实际场景中的应用效果。

5.3 系统功能

本系统从上述两个目标出发，结合实际场景的业务逻辑进行功能设计，将商品管理相关功能分为库存管理、商店管理和交易管理三个模块，分析与预测功能分为关联分析和预测系统两个模块。

商品的流通过程中涉及到库存、分销和交易三个主要环节，库存管理用于登记商品信息，对入库商品的品类、数量、价格和预计库存时间做记录，为下游任务中的分销提供信息，供应链上及时的数量信息流动是保证公司稳定经营的必要条件，商品入库是供应链开始的地方，为分销周转提供商品实物基础，出库也同样需要管理者及时登记，维护库存信息及时同步更新。连锁店作为总公司的下游环节，从总库存中调取货物，大量连锁店的管理也是不可忽视的一步，

哪些连锁分店代理人员从总公司订了哪些货品，何时发货以及数量是多少，都需要相关管理人员在商店管理系统中更新，更新的数据为上游库存和下游商店销售提供信息基础，他们依靠相关商品信息指定销售策略，保障货物顺利送到商超门店。当商品摆放到货架上时，对于连锁企业总部来说，这基本是最后一环，但最终需要想让整体数据实现加减同步，即销售商品、计算收益并且继续采购实现闭环，就需要门店对每个商品的交易数据进行详细的记录，销售时间、销售门店和销售额等都是企业要严格把握的信息，这些信息的处理决定了整个闭环是否能实现，这些商品的详细信息也为企业分析商品关联关系和预测各个环节的数量信息提供可靠的基础。

在对商品信息准确管理的基础上，我们利用深入挖掘商品之间的相关关系，利用常用的关联分析方法给出商品关联指数，相关信息可以用于管理者理解商品之间的联动关系，制定合理的销售组合策略，也可以作为商品推荐、销量预测和因果关系推断的特征。另外，商品在各个环节的动态变化数据反映了不同环节的供需变化，例如疫情期间对口罩的需求暴增，相关生产企业若能及时接收到销售端的大量产品缺口信息，及时补充相关原材料，调整产能和销售策略，一方面能够解决前线医护人员和人们对口罩的需求，另一方面也会给企业带来大量的利润，这种信息的高效传导少不了信息化系统的辅助，企业可以利用历史商品流动信息和现有可以预见的特征（日期、节假日等）对未来商品的销售量，库存流动量等数据进行预测，及时调整各个环节的流动策略，将商品信息的作用最大化。对于销量预测算法，我们为用户提供了多个选择，当数据量较小时，用户可选择传统自回归算法，数据量大时可选择利用深度学习算法，即上述我们提出的 FSN 和 MS-DTWI 结合的时间序列预测方法。

5.4 系统架构

本系统采用 B/S（浏览器/服务器，Browser/Server）架构实现，架构分为前端和后端两部分，前端为用户操作的交互界面，为用户展示各种功能；后端部分为实现上述功能设计相关逻辑，将用户输入数据进行处理，存入数据库，并返回给用户。这种实现方式有利于跨平台操作，用户既可以在 PC 端浏览器上使用，也可以在移动设备上登陆使用，避免下载客户端等繁琐操作和对设备的要求。

前端实现借鉴现有系统架构，重点在于逻辑的设计和实现。前端对商品信息、库存信息、关联性分析和销量预测等模块单独列出，直观友好的界面展示能有效降低学习成本，提高使用效率，我们还给出多种可视化方法，让用户直观地感受到数据的变化。

服务端的实现采用 Django 框架，它采用 MVT 的设计模式，即模型（Model）、视图（View）和模板（Template），Django 框架加快了数据库驱动的网站开发过程，注重组件的重用性和低耦合性，为网站的后端开发者提供足够的灵活性来自定义网站模块。本系统在此框架下设计上述功能的逻辑，将数据存储在 MySQL 数据库中，利用框架和数据库的交互来实现相应逻辑。

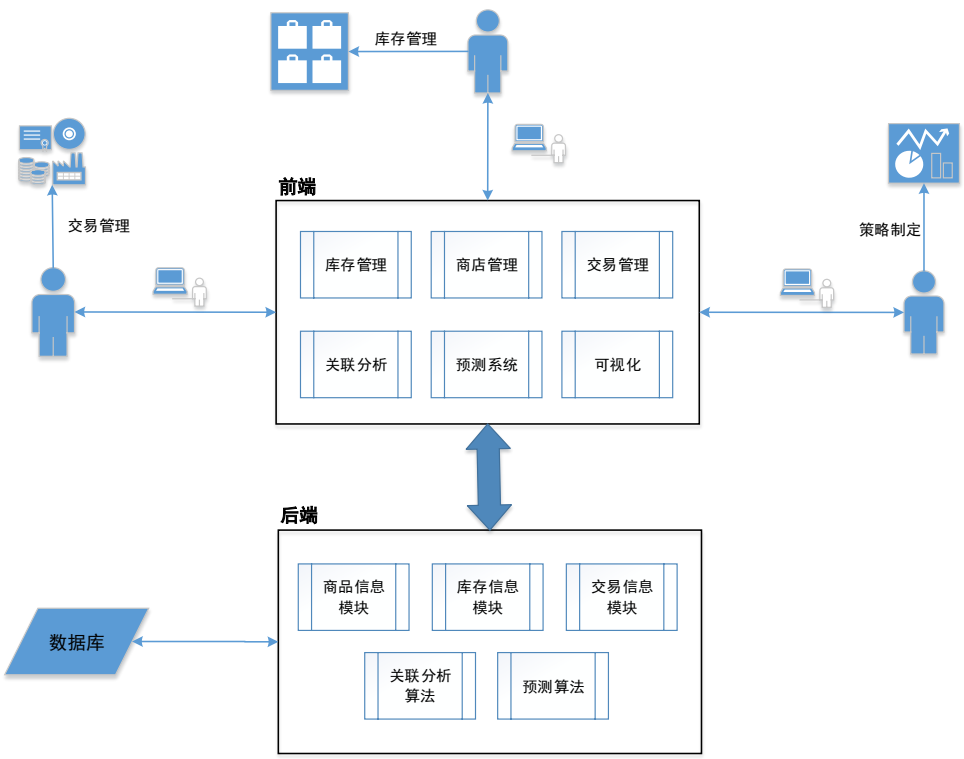


图 5-1: 整体系统架构图

系统的前后端交互逻辑如图 5-1所示，前端界面中包含库存管理、商店管理、交易管理、预测系统和可视化系统六个模块，不同环节的管理者通过前端与系统交互，实现库存管理和策略制定等任务，最后将决策传达给相关部门。前端将商品数据和用户请求发送到后端，我们在后端依据业务逻辑和算法逻辑实现相关函数，用于处理前端发送的数据，最后将处理结果返回给前端，最终反馈到用户界面。此过程中将增加、删除和修改的数据存储在 MySQL 数据库中。

5.5 系统实现

在上述系统架构的指导下，我们编码实现相关逻辑功能，得到“金牛（Taurus）系统”，系统界面的设计参考现有系统设计方案，力求界面简洁明了，降低学习使用成本，提高办公效率。主界面的效果如图 5-2所示，左侧为上述提到的各个功能模块入口，右侧为常用模块的快捷进入方式。本系统目前尚未商用，用于演示的数据为公开数据集中的实际数据或生成数据，本章所述实现主要展示了系统的界面设计、逻辑实现和可视化效果。

图 5-2: 系统主界面

5.5.1 商品管理

商品管理相关的模块注重文字和数字的输入输出，对商品的筛选和展示让用户能够找到想要的信息，提高使用效率。库存管理包含库存信息查询、商品入库和出库操作，库存信息的修改都会在到后端数据库中进行保存，用户可以按照商店和商品来检索和筛选库存数量。商店管理让公司管理者对下游连锁分店的销售情况和近期的利润变动一目了然。交易管理让管理者掌握每天的交易动态，及时观察交易量变化，补充缺少的货品，动态调整影响策略。

图 5-3 为库存管理界面，库存管理板块实现了入库，出库和库存信息管理功能。图中展示了“洗衣液”查询结果。

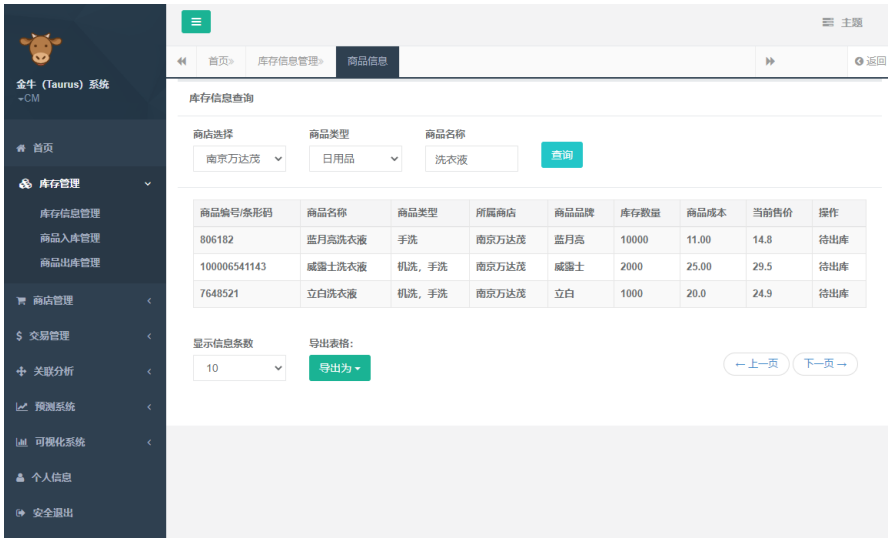


图 5-3: 库存管理

5.5.2 分析和预测

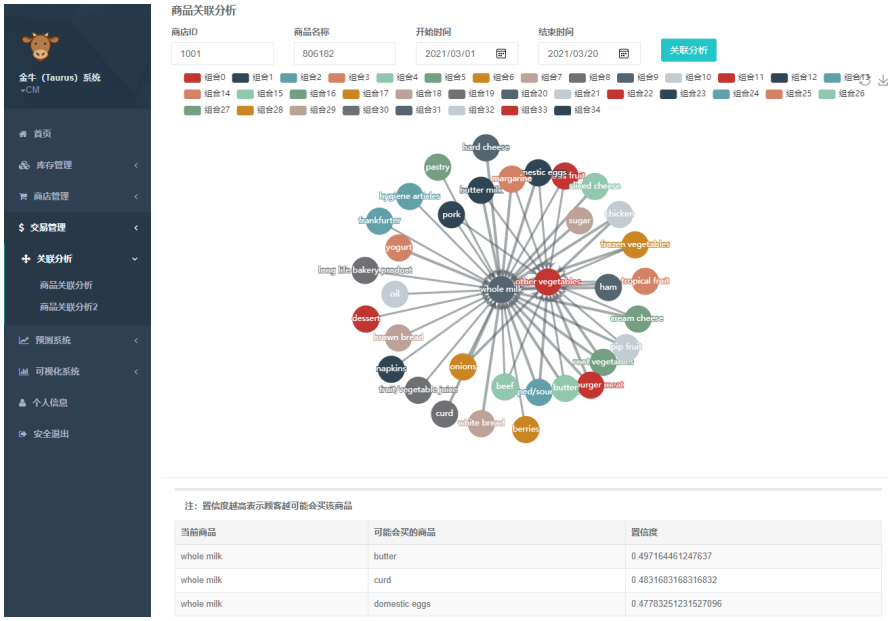


图 5-4: 关联分析

分析和预测是本系统的核心功能。当管理者制定营销策略时，可以参考商品之间的关联关系，将关联性强的商品组合到一起，绑定销售。还可以利用销量预测结果及时将未来可能会销量下跌的商品提前降价促销，将预测到销售会火爆的商品提前增加库存，防止出现断货情况。

对于关联分析模块，系统集成了 FP-growth 关联分析方法，利用商品交易和销量等信息计算关联程度，给出商品关联的置信度，由高到低进行排列，用户可

以挑选理想的数量的关联产品，为销售策略制定者提供参考，或者为下游分店的商品推荐提供有价值的信息。图 5-4 展示了关联分析结果的可视化效果图。

销量预测部分提供了多种预测方法共用户选择，用户可以结合多种预测结果综合考虑，系统还集成了本文提出的时间序列预测算法，不同预测方法对不同类型的数据会有预测效果差异，当数据量较小时，建议选择 AR 等传统统计学方法，因为基于机器学习的方法通常需要大量的训练样本训练模型，而初期通常无法采集到大量数据，当数据量初具规模时，建议采用基于机器学习的方法进行预测。系统会继续添加更多预测算法，解决多样化特征数据的预测问题。对于 FSN 方法，系统将其分为 2 部分，即 FSN 训练和 FSN 预测。使用时需要先进行模型训练，训练结束才可以进行预测。

为验证系统对时间序列预测的实际应用效果，我们导入了 Rossmann 商店销售数据集，此数据集包含了分布在欧洲 7 个国家的 1115 个零售商店商品销售数据，我们对其中 300 个分店进行预测。商品的销售量受许多因素影响，比如促销、市场竞争、节假日、季节和地理位置等。系统中的数据和通常用于学术研究的标准数据集不同，系统中数据更分散，需要在预测前做一些数据聚合工作，还需要手工构造相关特征作为协变量，我们的系统中在训练环节集成了数据聚合操作，另外，对于模型的超参数设置，可以改动算法的内部设定，此处给出的预测结果为系统默认参数设置。

如下表 5-1为我们利用上述提出的 FSN+MS-DTWI 做销量预测与其他模型的预测效果对比结果：

表 5-1: 不同时序预测方法对比结果

预测方法	对比结果		
	MAE	ND	NRMSE
AR	0.0828	0.5902	0.6219
SVR	0.0661	0.3603	0.4826
FSN+MS-DTWI	**0.0208**	**0.1295**	**0.1769**

可以看出针对此数据的预测，我们提出的基于神经网络的前馈序列网络 FSN 和 MS-DTWI 损失函数的结合有明显的效果优势，因为此时数据量较大，并且时间日期等协变量的存在，使得传统统计学方法和通常的机器学习算法不适合应

用，因此可以看到对比结果逊色一些。

5.6 本章小结

本章介绍了我们从商品管理和销量预测需求出发，设计“金牛（Taurus）系统”的目标、功能、架构和实现效果，系统的设计和实现是一个不断优化调整的过程，后续对于系统的优化也会继续进行，使系统更适合企业应用。将本文提出的时间序列预测算法应用于实际的系统中也是科研工作与社会价值相结合的重要一步，时间序列预测是应用范围相当广泛的研究课题，随着企业界的需求逐渐多样化，希望能够我们研究能够与时俱进，设计更强大的时间序列预测算法，提高生产生活质量，为社会创造更多价值。

第六章　总结与展望

本文重点解决时间序列预测任务中的问题与挑战，在分析现有算法的优点和缺点基础上，总结出一个优良的时序预测模型应该具有的特点，在此指导下，FSN 借鉴了循环网络对序列相对位置属性的建模等特点，同时抛开循环神经网络的限制，从卷积神经网络和注意力机制入手，设计一种既能有效建模历史信息，还具有更高训练性能的前馈序列网络模型 FSN，本文从序列信息流传播和真实数据集两个角度对模型有效性进行解释和验证。另外，我们从模型策略角度考虑，分析现有模型的损失函数的局限性，利用 DTW 度量指标设计了 MS-DTWI 损失函数，使模型更关注时间序列预测曲线的形态和时延情况，让时序预测结果更加精准，满足实际应用场景中的预测及时性和准确性的需求。最后我们将上述时间序列预测算法应用在自主设计开发的“金牛（Taurus）系统”中，并通过真实数据的对比实验验证其在实际应用中模型的优势。系统满足了企业对商品管理和分析预测的需求，让研究内容落地到实际场景，为社会创造价值。

对于时间序列预测问题的研究还有很多优化空间，我们会继续优化现有模型框架，提高预测精度，同时为了使算法的应用更安全可靠，我们也着力于解决模型可解释性的研究，期望让模型的使用者能够理解预测结果的产生原因和内部运行机理。其次，我们在对于模型的适用范围做进一步研究，因为机器学习算法的应用场景限制，对于数据量较少的情况，我们试图找到更有效的方法，让模型能够发挥更好的作用，目前计划的研究方向在于预训练模型的使用和小样本学习的结合，我们会继续探索，希望给出一个合理有效的答案。

参考文献

[1] VAN DEN OORD A, DIELEMAN S, ZEN H, et al. Wavenet: A generative model for raw audio[C/OL]//The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016. ISCA, 2016: 125. http://www.isca-speech.org/archive/SSW_2016/abstracts/ssw9_DS-4_van_den_Oord.html.

[2] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C/OL]//BENGIO Y, LECUN Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1409.0473.

[3] MAKRIDAKIS S, SPILIOTIS E, ASSIMAKOPOULOS V. The M4 Competition: 100,000 time series and 61 forecasting methods[J/OL]. International Journal of Forecasting, 2020, 36(1):54-74. https://doi.org/10.1016/j.ijforecast.2019.04.014.

[4] SEZER O B, Ugur Gudelek M, OZBAYOGLU A M. Financial time series forecasting with deep learning: A systematic literature review: 2005-2019[R]//arXiv. [S.l.: s.n.], 2019.

[5] DEB C, ZHANG F, YANG J, et al. A review on time series forecasting techniques for building energy consumption[M/OL]//Renewable and Sustainable Energy Reviews: volume 74. Elsevier Ltd, 2017: 902-924. http://dx.doi.org/10.1016/j.rser.2017.02.085.

[6] MAKRIDAKIS S, SPILIOTIS E, ASSIMAKOPOULOS V. Statistical and machine learning forecasting methods: Concerns and ways forward[J]. PloS one, 2018, 13(3):e0194889.

[7] GARDNER JR E S. Exponential smoothing: The state of the art—part ii[J]. International journal of forecasting, 2006, 22(4):637-666.

[8] ATHANASOPOULOS G, HYNDMAN R J. Modelling and forecasting australian domestic tourism[J]. Tourism Management, 2008, 29(1):19-31.

[9] OSMAN A F, KING M L, et al. A new approach to forecasting based on exponential smoothing with independent regressors[J]. Monash Econometrics & Business Statistics Working Papers, 2015.

[10] KIM K. Financial time series forecasting using support vector machines[J/OL]. Neurocomputing, 2003, 55(1-2):307-319. https://doi.org/10.1016/S0925-2312(03)00372-2.

[11] RÜPING S, MORIK K. Support vector machines and learning about time[C/OL]// 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, Hong Kong, April 6-10, 2003. IEEE, 2003: 864-867. https://doi.org/10.1109/ICASSP.2003.1202780.

[12] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J/OL]. Neural Comput., 1997, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[13] CHO K, VAN MERRIENBOER B, GÜLÇEHRE Ç, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C/OL]// MOSCHITTI A, PANG B, DAELEMANS W. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, 2014: 1724-1734. https://doi.org/10.3115/v1/d14-1179.

[14] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C/OL]// GUYON I, VON LUXBURG U, BENGIO S, et al. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. 2017: 5998-6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[15] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J/OL]. CoRR, 2018, abs/1803.01271. http://arxiv.org/abs/1803.01271.

[16] 李航, 等. 统计学习方法[M]. [出版地不详]: Qing hua da xue chu ban she, 2012.

[17] WEN R, TORKKOLA K, NARAYANASWAMY B, et al. A multi-horizon quantile recurrent forecaster[J]. arXiv preprint arXiv:1711.11053, 2017.

[18] GU J, BRADBURY J, XIONG C, et al. Non-autoregressive neural machine translation[J/OL]. CoRR, 2017, abs/1711.02281. http://arxiv.org/abs/1711.02281.

[19] BOX G E, JENKINS G M, REINSEL G C, et al. Time series analysis: forecasting and control[M]. [S.l.]: John Wiley & Sons, 2015.

[20] HOLT C C. Forecasting seasonals and trends by exponentially weighted moving averages[J]. International journal of forecasting, 2004, 20(1):5-10.

[21] DURBIN J, KOOPMAN S J. Time series analysis by state space methods[M]. [S.l.]: Oxford university press, 2012.

[22] GARDNER JR E S, MCKENZIE E. Forecasting trends in time series[J]. Management science, 1985, 31(10):1237-1246.

[23] WINTERS P R. Forecasting sales by exponentially weighted moving averages[J]. Management science, 1960, 6(3):324-342.

[24] PEGELS C C. Exponential forecasting: some new variations[J]. Management Science, 1969:311-315.

[25] HYNDMAN R, KOEHLER A B, ORD J K, et al. Forecasting with exponential smoothing: the state space approach[M]. [S.l.]: Springer Science & Business Media, 2008.

[26] CORTES C, VAPNIK V. Support-vector networks[J/OL]. Mach. Learn., 1995, 20(3):273-297. https://doi.org/10.1007/BF00994018.

[27] TAYLOR S J, LETHAM B. Forecasting at scale[J/OL]. PeerJ Preprints, 2017, 5: e3190v2. https://doi.org/10.7287/peerj.preprints.3190v2.

[28] CHEN T, GUESTRIN C. Xgboost: A scalable tree boosting system[C/OL]// KRISHNAPURAM B, SHAH M, SMOLA A J, et al. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM, 2016: 785-794. https://doi.org/10.1145/2939672.2939785.

[29] 周志华. 机器学习[M/OL]. 清华大学出版社, 2016. https://books.google.com/books?id=j0G8nQAACAAJ.

[30] CELIK A E, KARATEPE Y. Evaluating and forecasting banking crises through neural network models: An application for turkish banking sector[J/OL]. Expert Syst. Appl., 2007, 33(4):809-815. https://doi.org/10.1016/j.eswa.2006.07.005.

[31] ZHANG G P, QI M. Neural network forecasting for seasonal and trend time series [J/OL]. Eur. J. Oper. Res., 2005, 160(2):501-514. https://doi.org/10.1016/j.ejor.2003.08.037.

[32] SAHOO G, RAY C. Flow forecasting for a hawaii stream using rating curves and neural networks[J]. Journal of hydrology, 2006, 317(1-2):63-80.

[33] ZHANG G P. Time series forecasting using a hybrid ARIMA and neural network model[J/OL]. Neurocomputing, 2003, 50:159-175. https://doi.org/10.1016/S0925-2312(01)00702-0.

[34] TANG Z, DE ALMEIDA C, FISHWICK P A. Time series forecasting using neural networks vs. box- jenkins methodology[J/OL]. Simul., 1991, 57(5):303-310. https://doi.org/10.1177/003754979105700508.

[35] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet Large Scale Visual Recognition Challenge[J/OL]. International Journal of Computer Vision (IJCV), 2015, 115(3):211-252. DOI: 10.1007/s11263-015-0816-y.

[36] BIANCHI F M, MAIORINO E, KAMPFFMEYER M C, et al. An overview and comparative analysis of recurrent neural networks for short term load forecasting [J/OL]. CoRR, 2017, abs/1705.04378. http://arxiv.org/abs/1705.04378.

[37] FLUNKERT V, SALINAS D, GASTHAUS J. Deepar: Probabilistic forecasting with autoregressive recurrent networks[J/OL]. CoRR, 2017, abs/1704.04110. http://arxiv.org/abs/1704.04110.

[38] BOROVYKH A, BOHTE S, OOSTERLEE C W. Conditional time series forecasting with convolutional neural networks[J]. arXiv preprint arXiv:1703.04691, 2017.

[39] LAI G, CHANG W, YANG Y, et al. Modeling long- and short-term temporal patterns with deep neural networks[C/OL]//COLLINS-THOMPSON K, MEI Q, DAVISON B D, et al. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018. ACM, 2018: 95-104. https://doi.org/10.1145/3209978.3210006.

[40] LI S, JIN X, XUAN Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[C/OL]//WALLACH H M, LAROCHELLE H, BEYGELZIMER A, et al. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. 2019: 5244-5254. https://proceedings.neurips.cc/paper/2019/hash/6775a0635c302542da2c32aa19d86be0-Abstract.html.

[41] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. [S.l.]: MIT press, 2016.

[42] KHANDELWAL U, HE H, QI P, et al. Sharp nearby, fuzzy far away: How neural language models use context[C/OL]//GUREVYCH I, MIYAO Y. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers. Association for Computational Linguistics, 2018: 284-294. https://www.aclweb.org/anthology/P18-1027/. DOI: 10.18653/v1/P18-1027.

[43] HOLSCHNEIDER M, KRONLAND-MARTINET R, MORLET J, et al. A real-time algorithm for signal analysis with the help of the wavelet transform[M]// Wavelets. [S.l.]: Springer, 1990: 286-297.

[44] DUTILLEUX P. An implementation of the "algorithme à trous" to compute the wavelet transform[M]//Wavelets. [S.l.]: Springer, 1990: 298-304.

[45] CHEN L, PAPANDREOU G, KOKKINOS I, et al. Semantic image segmentation with deep convolutional nets and fully connected crfs[C/OL]//BENGIO Y, LECUN Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1412.7062.

[46] YU F, KOLTUN V. Multi-scale context aggregation by dilated convolutions [C/OL]//BENGIO Y, LECUN Y. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1511.07122.

[47] BAI S, KOLTER J Z, KOLTUN V. Trellis networks for sequence modeling [C/OL]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. https://openreview.net/forum?id=HyeVtoRqtQ.

[48] CHENG J, DONG L, LAPATA M. Long short-term memory-networks for machine reading[C/OL]//SU J, CARRERAS X, DUH K. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. The Association for Computational Linguistics, 2016: 551-561. https://doi.org/10.18653/v1/d16-1053.

[49] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C/OL]//2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016: 770-778. https://doi.org/10.1109/CVPR.2016.90.

[50] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C/OL]//BACH F R, BLEI D M. JMLR Workshop and Conference Proceedings: volume 37 Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. JMLR.org, 2015: 448-456. http://proceedings.mlr.press/v37/ioffe15.html.

[51] DEVLIN J, CHANG M, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[C/OL]//BURSTEIN J, DORAN C, SOLORIO T. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, 2019: 4171-4186. https://doi.org/10.18653/v1/n19-1423.

[52] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[J]. 2018.

[53] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8):9.

[54] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners [J]. arXiv preprint arXiv:2005.14165, 2020.

[55] LI S, LI W, COOK C, et al. Independently recurrent neural network (indrnn): Building a longer and deeper RNN[C/OL]//2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, 2018: 5457-5466. http://openaccess.thecvf.com/content_cvpr_2018/html/Li_Independently_Recurrent_Neural_CVPR_2018_paper.html. DOI: 10.1109/CVPR.2018.00572.

[56] SAKOE H, CHIBA S. Dynamic programming algorithm optimization for spoken word recognition[J]. IEEE transactions on acoustics, speech, and signal processing, 1978, 26(1):43-49.

[57] CUTURI M, BLONDEL M. Soft-dtw: a differentiable loss function for time-series [C/OL]//PRECUP D, TEH Y W. Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. PMLR, 2017: 894-903. http://proceedings.mlr.press/v70/cuturi17a.html.

[58] BLONDEL M, FUJINO A, UEDA N, et al. Higher-order factorization machines [C/OL]//LEE D D, SUGIYAMA M, VON LUXBURG U, et al. Advances in Neural

Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. 2016: 3351-3359. https://proceedings.neurips.cc/paper/2016/hash/158fc2ddd52ec2cf54d3c161f2dd6517-Abstract.html.

[59] FRÍAS-PAREDES L, MALLOR F, LEÓN T, et al. Introducing the temporal distortion index to perform a bidimensional analysis of renewable energy forecast [J]. Energy, 2016, 94:180-194.

[60] GUEN V L, THOME N. Shape and time distortion loss for training deep time series forecasting models[C/OL]//WALLACH H M, LAROCHELLE H, BEYGELZIMER A, et al. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. 2019: 4191-4203. https://proceedings.neurips.cc/paper/2019/hash/466accbac9a66b805ba50e42ad715740-Abstract.html.

[61] BLONDEL M, MENSCH A, VERT J. Differentiable divergences between time series[C/OL]//BANERJEE A, FUKUMIZU K. Proceedings of Machine Learning Research: volume 130 The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event. PMLR, 2021: 3853-3861. http://proceedings.mlr.press/v130/blondel21a.html.

[62] CHEN C, LEE W, KUO H, et al. The study of a forecasting sales model for fresh food[J/OL]. Expert Syst. Appl., 2010, 37(12):7696-7702. https://doi.org/10.1016/j.eswa.2010.04.072.

[63] YAN Y, HAO H, XU B, et al. Image clustering via deep embedded dimensionality reduction and probability-based triplet loss[J/OL]. IEEE Trans. Image Process., 2020, 29:5652-5661. https://doi.org/10.1109/TIP.2020.2984360.

[64] WAIBEL A H, HANAZAWA T, HINTON G E, et al. Phoneme recognition using time-delay neural networks[J/OL]. IEEE Trans. Acoustics, Speech, and Signal Processing, 1989, 37(3):328-339. DOI: 10.1109/29.21701.

[65] SHISKIN J. The x-11 variant of the census method ii seasonal adjustment program: number 15[M]. [S.l.]: US Department of Commerce, Bureau of the Census, 1967.

[66] ROBERT C, WILLIAM C, IRMA T. Stl: A seasonal-trend decomposition procedure based on loess[J]. Journal of official statistics, 1990, 6(1):3-73.

[67] HYNDMAN R J, ATHANASOPOULOS G. Forecasting: principles and practice [M]. [S.l.]: OTexts, 2018.

[68] DAGUM E B, BIANCONCINI S. Seasonal adjustment methods and real time trend-cycle estimation[M]. [S.l.]: Springer, 2016.

[69] LUNDBERG S M, LEE S. A unified approach to interpreting model predictions [C/OL]//GUYON I, VON LUXBURG U, BENGIO S, et al. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. 2017: 4765-4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.

[70] CAPISTRÁN C, CONSTANDSE C, RAMOS-FRANCIA M. Multi-horizon inflation forecasts using disaggregated data[J]. Economic Modelling, 2010, 27(3): 666-677.

[71] ZHANG J, NAWATA K. Multi-step prediction for influenza outbreak by an adjusted long short-term memory[J]. Epidemiology & Infection, 2018, 146(7): 809-816.

[72] LIM B. Forecasting treatment responses over time using recurrent marginal structural networks[C/OL]//BENGIO S, WALLACH H M, LAROCHELLE H, et al. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. 2018: 7494-7504. http://papers.nips.cc/paper/7977-forecasting-treatment-responses-over-time-using-recurrent-marginal-structural-networks.

[73] COURTY P, LI H. Timing of seasonal sales[J]. The Journal of Business, 1999, 72(4):545-572.

[74] BOESE J, FLUNKERT V, GASTHAUS J, et al. Probabilistic demand forecasting at scale[J/OL]. Proc. VLDB Endow., 2017, 10(12):1694-1705. http://www.vldb.org/pvldb/vol10/p1694-schelter.pdf. DOI: 10.14778/3137765.3137775.

简历与科研成果

基本信息

郝鸿延，男，汉族，1994 年 04 月出生，吉林省通化市辉南县人。

教育背景

2018 年 9 月 — 2021 年 7 月 南京大学计算机科学与技术系 硕士

2013 年 9 月 — 2017 年 7 月 南京邮电大学管理学院 本科

攻读硕士学位期间完成的学术成果

1. **Hongyan Hao**, Yan Wang, Jian Zhao, Furao Shen, "Temporal Convolutional Attention-based Network For Sequence Modeling" in *arXiv preprint arXiv*:2002.12530, 2020.
2. Yuanjie Yan, **Hongyan Hao**, Baile Xu, Jian Zhao, Furao Shen, "Image clustering via deep embedded dimensionality reduction and probability-based triplet loss" in *IEEE Transactions on Image Processing* 2020, 29: 5652-5661.
3. Siqiao Xue, Xiaoming Shi, **Hongyan Hao**, et al, "A Graph Regularized Point Process Model For Event Propagation Sequence" in 2021 *International Joint Conference on Neural Networks* (*IJCNN*)

攻读硕士学位期间的发明专利

1. 申富饶，**郝鸿延**，张旭. ''一种基于多生物特征的身份验证系统''（201910933448）

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金 ''基于深度感知增量式联想记忆神经网络的信息融合系统研究''（课题年限 2019 年 1 月 — 2022 年 12 月），负责时间序列预测相关问题的研究。

致　　谢

三年的硕士生涯转眼已接近尾声，大家都在感叹时间过得真快，不知不觉竟一起走过了一千多个日夜，很多珍贵的回忆也在这个过程中不经意的产生了。庆幸遇到了平易近人的老师和温暖可爱的同门，和大家的相处过程也让我意识到自己的很多不足，还好小伙伴们都很 nice，让实验室成为我们的快乐星球。硕士三年就像进入社会大门的预科班，我学会了如何自己分析问题，解决问题，如何与别人有效沟通，这些经历也是我宝贵的人生财富，这些积累必然少不了实验室的老师和同门的帮助。

因此，我要感谢我的导师申富饶老师。申老师严谨的学术态度和从问题出法的研究理念深刻影响了我，读研之初我还对计算机领域的研究方向有些茫然，申老师结合自己的经历，耐心指导我，让我寻找自己感兴趣的研究方向，每周与老师的讨论都会收获对研究方向的纠偏或更深入的理解。申老师“从问题出发”的研究理念让我在看待科研中的问题时都能化大为小，逐一攻破。申老师同样重视我们的性格和思想的引导，鼓励我们选择有意义的研究，为社会贡献自己的力量，做一个有用之材。

另外，我还要感谢赵健老师，每次组会赵老师都认真倾听我们的报告，提出很多关键的问题并与大家探讨。赵老师教授我们如何书写论文，还对我的论文做了非常细致的点评修改，让我意识到很多研究和写作上的问题，赵老师精益求精的态度深深感染着我，也是我以后工作学习的榜样。

我还要感谢实验室的同门，大家一起愉快地度过了三年时光，这将是我毕生最珍贵的回忆。大家一起讨论学习，互相指导，在遇到问题的时候积极主动帮助我，大家强大的专业技能总能让我学习到新知识，乐观的生活态度也让我倍感温暖。还要感谢我的室友，融洽的相处环境总能让我感受到了家的温馨。能结交一群如此优秀的朋友也是我这三年最大的收获。最后要感谢我的父母，你们是我最坚实的后盾，很庆幸能拥有通情达理的父母，不论我做出怎样的选择都能理解和支持我，让我不再畏惧困难，勇往直前。

《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：郝鸿延

2021年5月29日

论文题名	基于前馈神经网络的时间序列预测问题研究				
研究生学号	MF1833023	所在院系	计算机科学与技术系	学位年度	2021
论文级别	☐学术学位硕士 ☑专业学位硕士 ☐学术学位博士 ☐专业学位博士 (请在方框内画钩)				
作者 Email	haohy6@163.com				
导师姓名	申富饶，宋方敏				

论文涉密情况：

☑ 不保密

☐ 保密，保密期（_____年_____月_____日 至 _____年_____月_____日）

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。