**基于深度神经网络的时间序列预测**

摘要

时间序列是日常生活中常见的一种数据，多元时间序列预测在金融，交通和电力等多种领域中都有相当广泛的应用。

传统的学习方法在时间序列预测任务上存在着难以处理长周期和短周期混合序列的问题，而深度学习模型在此类时间序列中有较好表现，但现有模型存在着需要人为提供周期的先验知识和对于时间序列中存在的动态周期不够敏感这两点局限性。

为了突破这种局限，我们提出了一种具备动态跳跃连接机制的深度强化学习模型LST-Dynamic，该模型针对具备混合周期的时间序列设计了提取短期周期的循环层和提取长期动态周期的动态循环跳跃层，通过强化学习的手段，让模型能够自主捕捉多元时间序列中的动态周期关系，同时引入了Attention机制和AR组件，改善了模型在周期性不强的多元时间序列上的表现。为了证明模型结构的有效性，实验选取了五种多元时间序列数据集，结果表明LST-Dynamic的表现相较于LST-Skip和LST-Attn模型在绝大多数场合都有所提升。

关键词：多元时间序列预测，深度学习，强化学习

**MODELING LONG- AND SHORT-TERM TEMPORAL PATTERN WITH DYNAMIC SKIP CONNECTIONS**

**ABSTRACT**

Time series is a type of comman data in our daily life. Multivariate time series forecasting is widely used in many field such as finance, transportation and electricity. The traditional learning method has the difficult in time series forecasting tasks, which is processing with time series with long and short term. However, deep learning is proved very effective in this kind of time series forecasting, but the current mode is limited to 1) manually provide periodically Prior Knowledge 2)not sensitive enough toward dynamic-period in time series. To break this upper limit, we’ve raised a type of model called LST-Dynamic, which are able to dynamic skip connections. Aiming at time series with dynamic-period, this model is capable of recurrent layer for capturing short term dependencies, also dynamic skip recurrent layer for captureing long-term dynamic period. Through the method of reinforcement learing, the model is able to capture the dynamic period in multivariate time series. With the help of attention mechanism and AR module, the models’ performance on multivariate time series with nonperiod patterns is improved. To prove the effectivity of the module model, this research chose 5 types of multivariate time series dataset. According to the result, compared to LST-Skip and LST-Attn, LST-Dynamic has shown significant improvement under most circumstances.

**Key words:**Time Series Forecasting, Deep Learning, Reinforcement Learning

**目录**

[1 绪论 1](#_Toc39584393)

[1.1 研究背景 1](#_Toc39584394)

[1.2 研究现状 1](#_Toc39584395)

[1.3 研究内容与目的 2](#_Toc39584396)

[2 相关工作 3](#_Toc39584397)

[2.1 传统时间序列预测 3](#_Toc39584398)

[2.1.1 自回归模型 3](#_Toc39584399)

[2.1.2 机器学习模型 3](#_Toc39584400)

[2.2 深度学习时间序列预测 4](#_Toc39584401)

[2.2.1 循环神经网络 4](#_Toc39584402)

[2.2.2 注意力机制 6](#_Toc39584403)

[2.2.3 深度强化学习 6](#_Toc39584404)

[2.2.4 混合模型 7](#_Toc39584405)

[2.3 本章小结 7](#_Toc39584406)

[3 模型 9](#_Toc39584407)

[3.1 问题形式化 9](#_Toc39584408)

[3.2 框架 9](#_Toc39584409)

[3.2.1 卷积组件 11](#_Toc39584410)

[3.2.2 循环组件 11](#_Toc39584411)

[3.2.3 时间注意力层 12](#_Toc39584412)

[3.2.4 动态连接组件 13](#_Toc39584413)

[3.2.5 自回归组件 16](#_Toc39584414)

[3.3 目标函数 17](#_Toc39584415)

[3.3.1 回归任务 17](#_Toc39584416)

[3.3.2 策略梯度 18](#_Toc39584417)

[3.4 优化策略 18](#_Toc39584418)

[3.5 本章小结 19](#_Toc39584419)

[4 实验 21](#_Toc39584420)

[4.1 实验设置 21](#_Toc39584421)

[4.1.1 数据集 21](#_Toc39584422)

[4.1.2 对照组 24](#_Toc39584423)

[4.1.3 评估方法 24](#_Toc39584424)

[4.1.4 超参规定 24](#_Toc39584425)

[4.2 实验流程 25](#_Toc39584426)

[4.2.1 实现环境 25](#_Toc39584427)

[4.2.2 数据预处理 25](#_Toc39584428)

[4.2.3 主要流程 27](#_Toc39584429)

[4.3 实验结果 28](#_Toc39584430)

[4.3.1 模型性能 28](#_Toc39584431)

[4.3.2 模型结构有效性 31](#_Toc39584432)

[4.3.3 超参影响 32](#_Toc39584433)

[4.4 本章小结 32](#_Toc39584434)

[5 总结与未来展望 34](#_Toc39584435)

[5.1 总结 34](#_Toc39584436)

[5.2 未来展望 34](#_Toc39584437)

[5.2.1 奖励函数 34](#_Toc39584438)

[5.2.2 普适性 34](#_Toc39584439)

[5.2.3 特殊点处理 34](#_Toc39584440)

[参考文献 36](#_Toc39584441)

[致谢 40](#_Toc39584442)

# 绪论

## 研究背景

多元时间序列是一种在多种领域中普遍存在的数据类型，它反映了一段时间内相互影响的分量的变化规律和发展趋势，例如太阳能电站的发电量，某个地区所有住户的耗电量和交通堵塞的情况等等。我们希望通过对历史时间序列进行分析来预测未来时刻的状态，以此指导生产生活或预警潜在的危险事件。目前，时间序列预测已被应用到电力、交通、经济、医学、互联网等诸多领域，具有广阔前景和巨大的现实意义[1]。

## 研究现状

时间序列数据与其他类型的数据的最大区别在于当前时刻数据值和过去时刻数据值存在着联系[2]，该特点表明时间序列数据隐含有周期性规律，而多元时间序列预测面临的一个重要挑战就是如何捕捉动态的周期关系。具体来说，现实世界中的时间序列的周期模式通常是长周期与短周期的结合体，例如高速公路每小时的占有率往往有两种周期模式：每日和每周，前者描绘早晚高峰，后者反映了工作日和周末的区别。将多种周期性因素同时纳入考量的模型往往能得到更准确的预测结果。但是，传统方法对此类混合周期因素不做特殊处理，而是独立地进行建模。

近年来，随着深度学习研究的发展，有人提出使用深度神经网络解决此类问题，如LST-Net[3]，并取得了一定的进展。但现存方法依旧存在一个问题，即需要用户显式的提供周期的长度，因此模型没法很好的适应动态的周期关系，缺乏灵活性。

解决现存方法中的缺点是本篇文章的主要目的，为了克服固定周期所造成的限制，我们将强化学习和深度学习结合，提出了动态跳跃连接组件，并和循环神经网络与自回归组件组合形成了一个新的模型，即LST-Dynamic。捕捉时间序列周期的难点在于，通常没有标签指示哪些时刻的数据间存在周期依赖关系，手动输入的周期值与真实数据存在一定的误差，且人为指定的周期是一个定值，对周期的波动性不敏感，会忽略到动态的周期关系。为了解决这个问题，我们采用深度强化学习和循环神经网络相结合的方法，在循环神经网络的两个隐藏层单元中设置智能体，智能体让每个隐藏层单元与过去时刻最合适的隐藏层单元建立连接。因此，当时间序列的周期关系产生波动时，本模型能捕捉到此类变化并取得更好的结果。

## 研究内容与目的

本文的主要贡献有如下几点：

（1）提出基于长短周期动态获取的时间序列预测模型LST-Dynamic。采用强化学习方法，从所观察的时间序列中计算超参数。减少用户的干预。

（2）在LST-Dynamic中集成Attention机制，提出LST-D&A模型，既能获取动态周期，还能用于解决LSTM/RNN模型输入序列较长的时候很难获得最终合理的向量表示问题。

（3）在多个公开数据集中的实验表明，LST-Dynamic和LST-D&A均能获得更好的性能。

# 相关工作

## 传统时间序列预测

### 自回归模型

早期的时间序列预测模型主要是传统线性模型，经典的模型包括移动平均模型(Moving Average, MA)、自回归模型(Auto Regressive, AR)，自回归移动平均模型（Auto Regressive Moving Average, ARMA）[4]等。这三种模型因其统计特性和著名的BOX-Jenkins方法[2]，常常被整合到其他时间序列模型中。但ARIMA模型因其在长周期关系的建模中计算成本较高，因此很少用于高维多元时间序列预测。与此相反，AR模型拓展到多变量环境的变体模型：向量自回归（VAR）模型因其简单而易于计算，成为在多元时间序列中使用最广泛的模型之一[5]。假设xt为t时刻的时间序列值，AR模型仅利用过去p时刻的时间序列数据来回归预测xt，其公式可以表示如下：

其中wk表示权重，bk表示偏差。

虽然自回归模型简单易用，但此类模型存在有在时间窗口上线性增长，在变量数量上平方增长的问题，在处理较长或具有长周期特征的时间序列时，形成的大型模型容易过拟合。另一方面，线性模型假设时间序列过去值与未来值之间存在的是线性函数关系，而实际生活中的时间序列并不总是线性关系序列。这两项缺点极大的限制了自回归模型在现代大规模时间序列预测任务上的应用，因此探索其他方法进行时间序列预测，尤其是能适用于非线性关系的时间序列预测模型逐渐成为研究的重点[6]。

### 机器学习模型

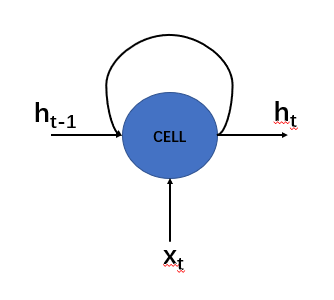
时间序列预测本质上与机器学习中的标准回归问题类似，各类经典的模型如支持向量机（Support Vector Mechine,SVM）[7]，矩阵分解 (Matrix Factorization,MF)和高斯过程（GP）在时间序列预测中也得到了广泛的运用。支持向量机基于统计学习，其VC维理论和建立在几何基础上的特点使其在小样本高维数据上具有优势。SVM早期被直接用于预测股票价格指数，贝叶斯证据框架和混合专家系统结构的引入使SVM更适用于时间序列预测。矩阵分解的目的是找到两个小规模矩阵，使小规模矩阵的乘积近似拟合原始矩阵，高维时间序列可以表示为矩阵形式。Zhang等在原本矩阵分解的基础上提出了使用稀疏正则矩阵分解[8]，进一步利用了时空信息，减少了模型在定位上的误差。实验证明，矩阵分解模型在缺失数据的时间序列预测中能取得较好的结果。机器学习模型在处理序列问题上有相当成熟的解决方案，因此使用这些线性方法进行多元时间序列预测效率很高。但线性机器学习模型有和自回归模型同样的缺点，即难以捕捉多元时间信号复杂的非线性关系，在同等效率下表现较差。

高斯过程是一种非参数方法，用于对连续分布的函数进行建模。这与由参数化的功能类别（例如VAR和SVR）定义的模型形成对比。高斯过程可用于多元时间序列预测任务，采用核函数大大增强了高斯过程在非线性问题上的处理能力，并且可用作贝叶斯推断中函数空间的先验[9]。例如，Roberts提出了一种完全使用贝叶斯方法的GP先验，用于非线性状态空间模型，它能够捕获复杂的动态现象。虽然高斯过程性能十分强大，但高斯过程的强大功能伴随着高计算复杂性。高斯过程中需要进行内核矩阵的矩阵求逆操作，用于多元时间序列预测的高斯过程的直接实现在观察数量上具有立方复杂性。

## 深度学习时间序列预测

### 循环神经网络

深度学习在语音识别、图像识别、自然语言处理等多个领域中已有广泛的应用，为各个领域内的大量问题提出了一种更有效的解决方案。深度学习的典型代表是神经网络，神经网络能极大程度地拟合非线性函数，但常规神经网络不附带输入结点的时序关系，输出的各结点之间互不连接，没有考虑到时间序列先后的依赖关系[10]。为解决此问题，人们提出了循环神经网络（Recurrent Neural Network, RNN），该结构同时考虑过去时刻信息和当前时刻信息，前一时刻的输出作为后一时刻的部分输入，使网络具有记忆力，能保存过去的信息，因此在具有时间顺序的数据上取得了较好结果。RNN的网络结构如下所示：



**图 2‑1 循环神经网络结构**

循环神经网络的梯度更新策略和传统神经网络一样都为反向传播算法，所有循环神经网络单元共享权值，因此减少了参数量，但因为梯度更新时会受到前面节点的影响，循环神经网络在处理长时序的任务时出现了梯度消失或梯度爆炸的问题，适用的场景受到了限制。针对这一问题，Hochreiter等人基于RNN结构提出了长短时记忆网络(Long Short Term Memery, LSTM)[11]，LSTM采用门控机制来控制信息的存留，且改变了处理过去时刻信息和当前时刻信息的方法，因此LSTM在长距离时间序列上不易出现梯度消失或梯度爆炸的问题。J.Chung等人在LSTM的基础上提出了门控循环单元(Gated Recurrent Unit, GRU)结构[12]，减少了LSTM中的门控数量，仅设置更新门和重置门，能以更少的参数、更快的收敛速度达成与LSTM相差无几的效果。虽然LSTM在处理长距离时间序列上相较于RNN已有较大的提升，但其在捕捉远距离信息的能力仍不满足实际的需求。这一问题在序列任务中引申出了两种解决途径，一是采用跳跃连接机制强化LSTM的远距离信息[13]，二是使用注意力机制，每个结点间的距离恒定，即在后文中介绍的transformer模型。在自然语言处理领域中， LSTM和GRU呈现出能够有效捕捉输入序列的中期和短期依赖的特性，因此在机器翻译，语音识别等任务中取得了重大突破。RNN与其变体也常被用于多时间尺度的时间序列任务中[14-17]。

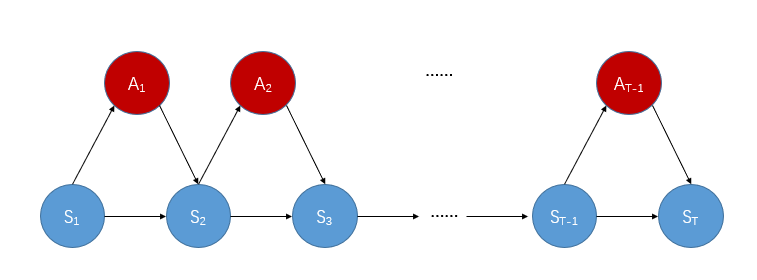
除RNN外，神经网络的主要模型还有卷积神经网络(Convolutional Neural Networks)。CNN被广泛应用于计算机视觉领域中，CNN 主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于 CNN 的特征检测层通过训练数据进行学习，所以在使用 CNN 时，避免了显示的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。因此，CNN中的部分操作常常作为特征处理的手段用于多个领域中[18,19]。例如，通过因果卷积，膨胀卷积和残差连接的组合，形成的时间卷积网络[20]在时间序列预测的某些场景下取得了比RNN更优秀的结果。

### 注意力机制

Google提出了著名的基于自注意力(self Attention)的Transformer模型[21]，在序列预测任务中取得了巨大的成功。Transformer模型被广泛应用于翻译、语音、音乐和图像生成等领域。然而，Transformer模型需要计算缩放关注度，在超长序列中的计算量急剧增加，并且自注意力的空间复杂度会随着时间结点数量的增长而成倍增长。这导致在细粒度且具有长期周期依赖的时间序列预测领域中，直接使用Transformer模型往往得不到理想的效果。但Attnetion机制在时间序列预测领域仍有出色的表现[22-24]，Huang s等人提出的DSANet (Dual Self-Attention Net)[25]和Shih S Y提出的TPANet[26]正是基于自注意力机制的深度学习模型。TPANet结合了CNN和自注意力机制，DSANet基于LSTNet结构的基础上，将两个循环层替换为两个自注意力层，分别提取短周期和长周期模式。这些模型在多元时间序列预测中也取得了较好的表现，但依旧存在transformer模型固有的问题。

### 深度强化学习

强化学习（Reinforcement Learining,RL），又称增强学习，指一类从环境的交互中不断学习以完成特定的目标的方法，一般来说，强化学习的目标被设定为获取最大的奖励。强化学习的过程可以被描述为一个智能体探索环境得到当前状态St后，根据当前状态St做出相应的动作At，由状态St进入到状态St+1并得到做出动作后的奖励Rt+1[27]。强化学习的理论基于马尔可夫性质，智能体和环境交互的过程可以看作是一个马尔可夫决策过程，马尔可夫性质（Markov Property） 是指一个随机过程在给定现在状态及所有过去状态情况下，其未来状态的条件概率分布仅依赖于当前状态。强化学习中的马尔可夫决策过程可以用下图表示：



**图 2‑2 马尔可夫决策过程**

强化学习在时间序列预测中也有应用[28]，但更多的是作为辅助求解中间过程中的决策问题，例如本文中使用强化学习决定循环神经网络中的跳跃间隔[29]。

### 混合模型

基于深度学习的模型在序列预测任务上往往能得到较为优秀的表现，具体到时间序列预测领域中，早期的工作直接使用深度学习模型进行预测，如使用带有受限波尔茨曼机的深度置信网和集成去噪编码器。但通常我们更倾向于结合时间序列数据的特点构建混合模型，例如ARIMA模型和多层感知机混合，RNN和线性模型混合，如R2N2（Residual RNN）[30]利用RNN预测线性模型的残差。随着深度网络的发展，CNN和RNN的混合模型也被应用于时间序列预测中，LST-Net使用CNN提取短期依赖关系，利用两个RNN层获取长期依赖关系，增加自回归单元解决神经网络对数据尺度变化不敏感的问题，LST-Net提出了一种新的循环结构，即循环跳跃组件，循环跳跃组件中的每个隐藏层单元与一个周期前的隐含层单元连接，以此捕获超长期依赖关系。然而，LST-Net需要用户提供一个固定周期值来针对性的获取周期关系，这导致模型对动态周期关系的捕获能力不强，灵活性较差。

## 本章小结

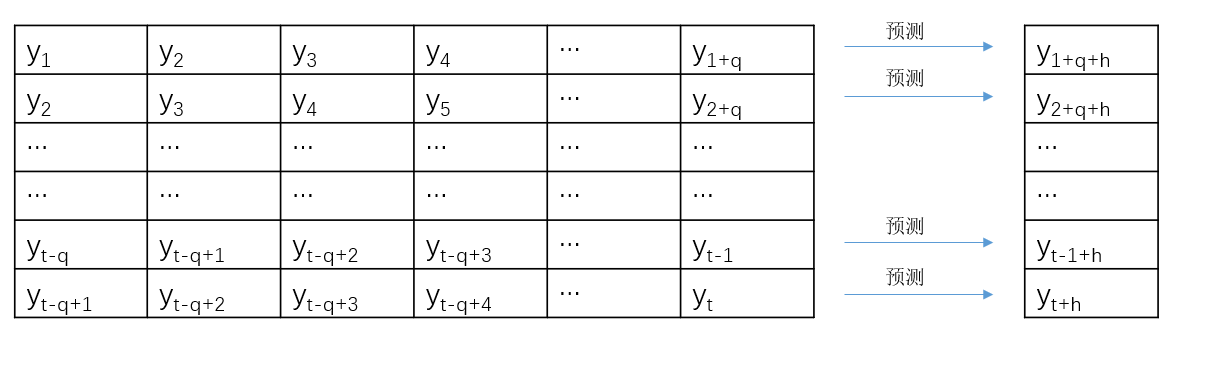
时间序列预测这项任务已进行了不少研究工作，传统方法如AR模型的优势在于简单易于计算，但往往局限于在单一的时间维度上的预测，而真实世界的时间序列往往是周期性和趋势性的混合体，因此传统方法难以拟合时间序列的长期趋势和周期。循环神经网络等深度神经网络在处理周期性较强的多元时间序列上更有优势，相较于一般的神经网络适用于更多的场景。基于注意力机制的模型在序列预测中也有广泛应用，循环神经网络和注意力机制的结合让循环神经网络能更为合理的提取依赖关系，完全使用注意力机制的Transformer模型更是在序列预测中取得了非凡的成绩。为了增强神经网络在长期和短期依赖混合的多元时间序列上的性能，在循环神经网络的基础上添加了跳跃连接结构，通过改变循环神经网络的连接跨度使其更适用于周期依赖较强的时间序列，跳跃间隔的不确定性和时间序列周期动态性带来的问题，可以通过强化学习的方法来让模型动态决定跳跃间隔来解决。

# 模型

在这一节中，我们首先定义时间序列预测任务的具体目标，接着讨论LST-Dynamic模型的结构细节，最后我们介绍目标函数和优化策略。

## 问题形式化

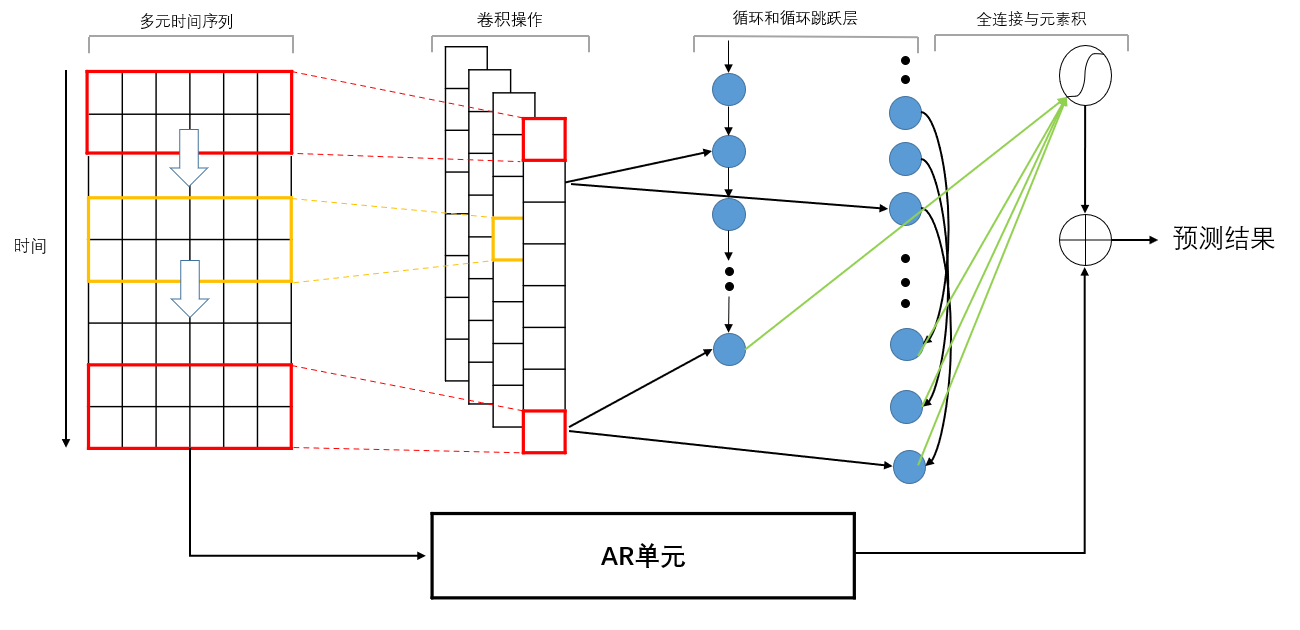
我们假设某时刻可以收集到D个时间序列数据，表示为y= {x1,x2,x3,…,xD}。在实际应用中，我们通常使用滚动预测的方法，将全部的时间序列数据按结点分组表示，称一组为一个窗口，一窗口时间序列数据可表示为Yi = {y1,y2,…,yt }，yt代表本窗口第t个时间结点记录的序列信号，yt∈RD，D代表变量维度。要求每个相邻结点间的时间间隔固定，每个窗口负责预测第t+h时刻的数据yt+h。假设窗口大小为q，共有T个时间结点，全部的时间序列数据可被表示为Y = {YT-q+1,YT-q+2 ，…，YT}，Yi∈Rq×D，表示一个窗口的时间序列数据，i表示窗口中最后一个时间结点在全部数据中的位置。我们假设一个窗口Yi的数据包含有足够的信息来预测之后第h个时刻的数据yi+h，则问题可形式化为已知输入矩阵为Yt = {yt-q+1,yt-q+2,…,yt}∈Rq×D，想要预测目标yt+h∈RD。则已知T时刻的数据，可以预测T到T+h时刻的数据滚动预测的过程可用下图表示：



**图 3‑1 滚动预测流程**

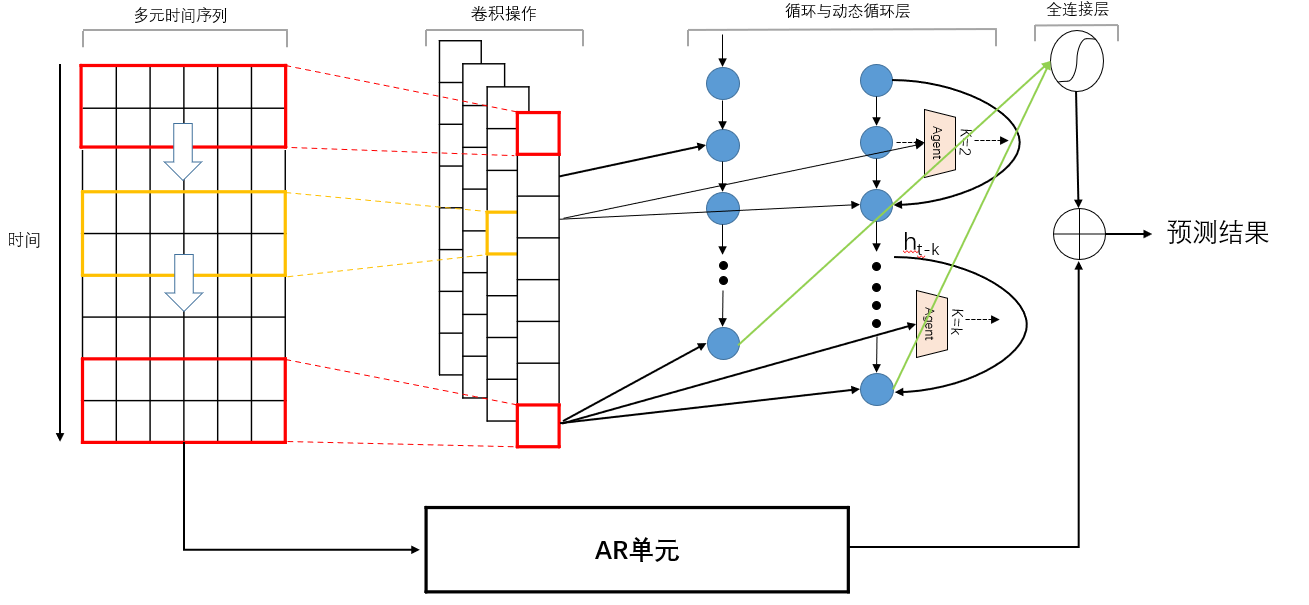
## 框架

LST-Skip的性能在多元时间序列领域中得到了广泛的认同，其基本结构也被应用于多种同任务模型中。本小节我们简单介绍LST-Skip网络的结构，将LST-Skip和LST-Dynamic、LST-D&A的结构进行对比，并详细介绍每个模块。



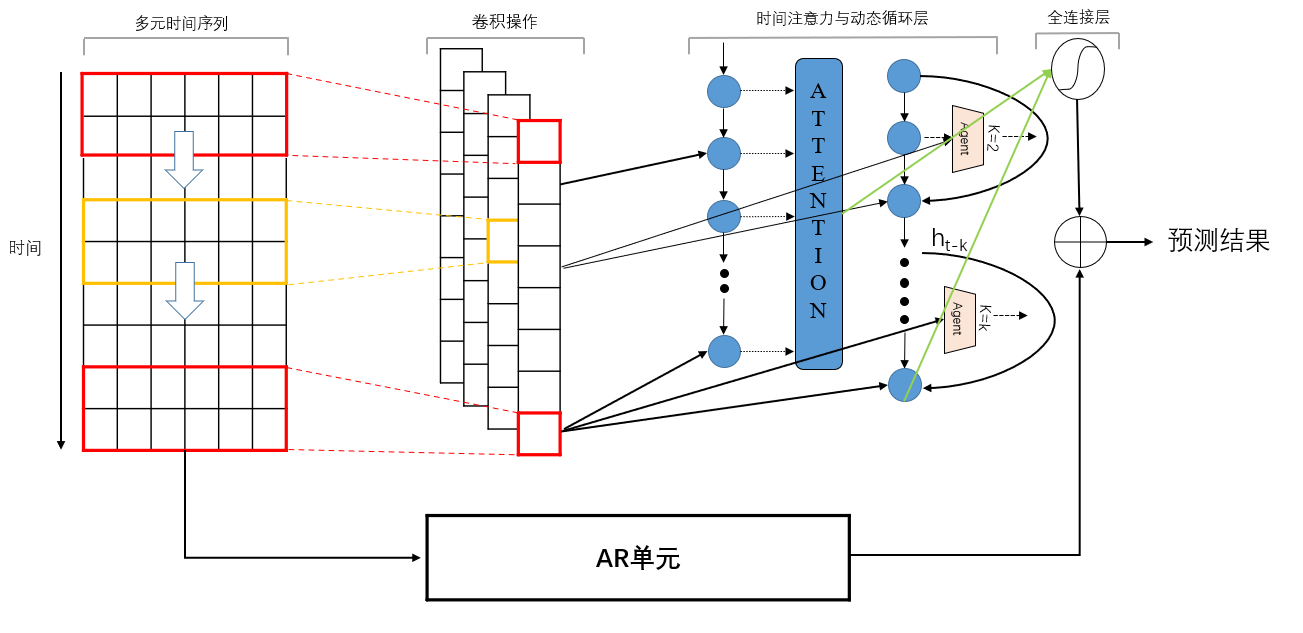
**图 3‑2 LST-Skip结构**

LST-Skip首先在多元时间序列上进行卷积操作，相较于直接使用原始时间序列，使用卷积操作能更有效的提取局部依赖和短周期特征。卷积操作后的输出送入到两个循环层，一个采用标准循环单元，针对短周期特征，另一个采取循环跳跃连接，每固定间隔的循环单元相连，针对长周期特征，最后将两个循环层的输出拼接到一起，全连接得到最终的预测结果。



**图 3‑3 LST-Dynamic 结构**

LST-Skip的循环跳跃层存在着跳跃间隔固定，需要用户提供先验信息，且对周期波动不敏感的问题。为了克服这些问题，我们提出了LST-Dynamic结构，引入基于策略梯度的深度强化学习，让每个智能体自主学习本段时间序列上的周期关系，模型中表现为循环跳跃层每个单元动态选择与之连接的单元，将循环跳跃层更换为动态连接组件。



**图 3‑4 LST-D&A 结构**

我们发现LST-Skip和LST-Dynamic结构在周期性不强的多元时间序列中表现都不佳，而注意力机制能略微提升模型在此类数据集中的表现。在循环层上引入注意力机制，形成时间注意力组件。通过时间注意力组件与动态连接组件的结合，我们得到了LST-D&A模型，与LST-Dynamic模型分别适用于不同种类的数据集。

### 卷积组件

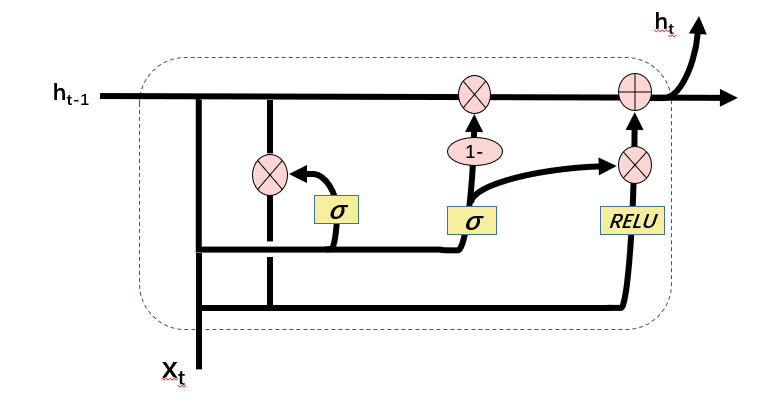
LST-Dynamic的第一层是一个没有池化的卷积网络，其目的是提取时间维度上的短期模式以及变量间的局部依赖关系，卷积层由宽度为w和高度为n的多个卷积核组成，第k个卷积核扫描输入矩阵X生成。

其中\*代表卷积操作，输出ℎ𝑘是一个向量，RELU（x）=max（0，x）。卷积层的输出矩阵大小为𝑑𝑐×T，我们可以选择在输入矩阵X的左边用零填充，使每个向量ℎ𝑘的输出长度和原始序列保持一致。卷积层的输出矩阵大小为𝑑𝑐×T，𝑑𝑐代表过滤器的数目，可以理解为想提取的特征种类。通过卷积操作，我们将原始时间序列的短期特征整合为一个时间点，同时提取到多元时间序列的隐藏特征，使原始多元时间序列的特征能够更好的表达。

### 循环组件

卷积组件的输出直接送入循环组件和Dynamic组件中，循环组件通过隐藏层状态的传递来保留过去时刻的信息。基于时间序列数据的长期性和本模型专注于循环组件采用RNN的变体GRU，GRU能以更少的参数，更快的收敛速度实现与LSTM基本相同的效果，在时间t的隐藏状态被计算为：

⊙代表向量的点积，σ代表sigmoid函数，𝑥𝑡是时刻t时，该层的输入。这一层的输出是每一个时间戳的隐藏状态。然而研究者们习惯于使用tanh作为隐藏更新激活函数，我们的经验发现RELU表现的更为可靠，通过它梯度更容易反向传播。在LST-Dynamic模型中，我们取GRU最后的隐藏层作为输出，中间的隐藏层弃置不用。

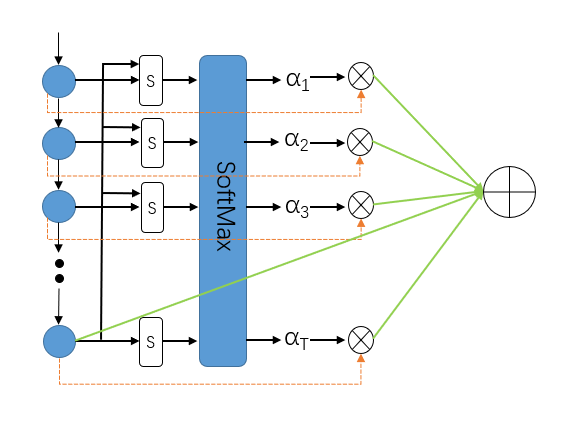


**图 3‑5 标准GRU结构**

### 时间注意力层

为了使循环跳跃层更专注于提取短期的周期关系，我们考虑使用注意力机制学习每个隐藏层时间点所占比重。实验证明，注意力机制能使模型在周期性不明显的时间序列上依旧能取得较好效果。注意力机制的计算分两步：一是对窗口内的隐藏层结点计算注意力分布，二是根据注意力分布来得到带注意力机制的时间向量。常见的注意力打分函数有加性模型和点积模型等，我们选用缩放点积作为注意力打分函数，缩放点积模型能够更好地利用矩阵乘积，且不存在当输入向量维度较高时softmax函数梯度较小的问题。t时刻的注意力分布可以用以下公式计算：

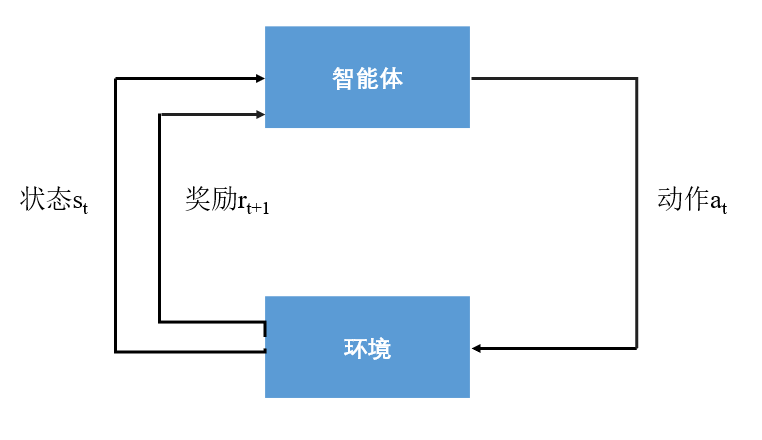
其中Ht = [ht-q,…,ht-1]，由输入窗口中的循环组件隐藏层堆叠而成，D为隐藏层的维度。利用注意力分布得到带权重的上下文向量ct = Ht αt，将ct与ht-1拼接到一起，配合线性映射操作得到带权重的时间向量作为时间注意力层的最终输出，其计算公式如下：



**图 3‑6 时间注意力层结构**

### 动态连接组件

尽管LSTNet缓解了传统时间序列预测模型在处理周期性序列上出现的问题，但使用LSTNet需要一个预定义的超参数p，表示时间序列以p为周期，这不仅需要观察数据以确定合适的值，还难以捕捉动态周期关系，我们希望每个循环单元能够自己从过去的隐藏状态中选择与其有依赖关系的状态，但一个序列的周期特征往往难以标注，并且时间序列的周期往往具有一定的动态性，没有一个标签能够准确地指示这种特征。为了解决这个问题，我们使用强化学习的方法，在探索过程中学习依赖关系，使智能体(Agent)动态决定哪些状态应该相连接。智能体指能够获取环境信息，通过环境信息做出决策的单元。智能体决策的过程如下图所示：



**图 3‑7 智能体决策流程**

强化学习要解决三个问题：环境(Environment)、行动(Action)和奖励函数(Reward)。除此之外，为了使强化学习和循环神经网络相结合，还应该对循环单元进行相应修改。

（1）环境

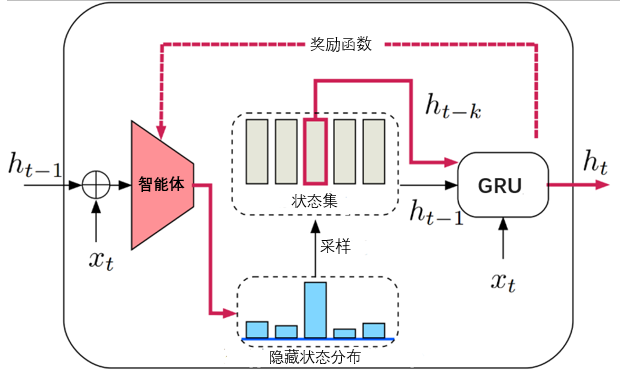
考虑到时间序列当前时刻数据值和过去时刻数据值存在联系的特点，环境的设置应当综合当前时间和过去时间的状态。因此，受到LSTM等结构的启发，当前环境可以被计算如下：

其中，ht-1是上一时刻的隐藏状态，xt是当前时刻的输入，代表拼接操作。当前时刻的时间序列特征和过去时刻的时间序列特征可能并不服从同一分布，因此我们选择拼接操作而非加法操作。

（2）行动

观测到环境st后，智能体需要决定哪个过去状态对接下来的循环单元影响最大，我们记录最近K个观测到的状态，Sk =[st-K, st-K+1,. . . , st-1] （K需提前设定）。智能体根据多项分布，在Sk中采样出最佳状态。

其中，MLP是一个多层感知机，将st转化为一个K维向量，softmax函数将该K维向量转化为概率分布P。代表采样操作，k=i时，[k=i]为1，否则为0。St-k被选中进行接下来循环单元的计算。智能体选择的过程如下图所示：



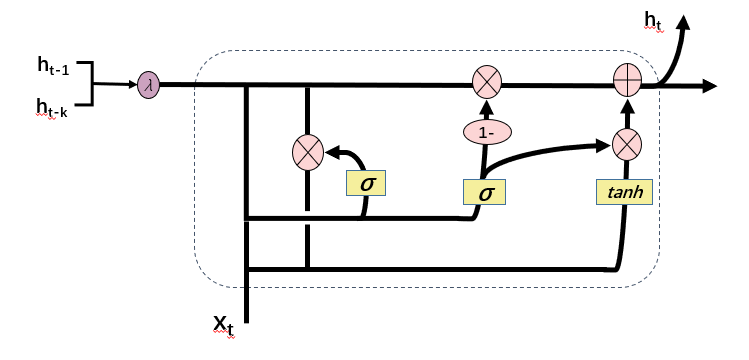
**图 3‑8智能体在t时刻决策流程**

（3）奖励函数

奖励函数是跳跃连接的指示器，一个合适的奖励函数能引导智能体选择一系列最优的跳跃行为从而得到更好的预测结果。我们使用点积比来衡量预测值和真实值之间的相似程度（值越大越相似），这样的奖励函数能鼓励智能体选择最合适的跳跃连接从而提升预测结果和真实值的相似程度。

经过智能体动态选择出ht-k后，我们使用标准的GRU单元来对选中的状态进行处理，在实际的预测中，我们发现结合ht-k和ht-1两个隐藏层的状态往往能取得更好的效果，因此，在模型中，我们使用超参数λ来连接这两个状态。本GRU单元t时刻的隐藏层状态可由以下公式计算：

其中，k∈{1,2,……,K}，*λ*为预设的超参，用来决定过去状态和当前状态所占比例，新的GRU单元可以用下图表示：



**图 3‑9 动态连接结构的GRU单元**

我们使用全连接来组合Dynamic组件和RNN组件和输出。设RNN最终隐藏层的输出为，Dynamic-RNN层最终隐藏层的输出为，最终输出可以被计算如下：

其中是神经网络部分的最终预测结果，即下一节中描述的关注周期依赖的非线性部分。

### 自回归组件

由于卷积和循环成分的非线性特性，神经网络模型的一个主要缺点是输出尺度对输入尺度不敏感。并且在实际的数据集中，输入序列的尺度的变化往往不具备周期性，这大大降低了神经网络模型的预测精度。受到时间序列分解观念的启发，我们将最终模型分解为两部分以改善该缺陷，最终预测结果由一个主要关注局部尺度变化问题的线性部分和一个关注周期依赖的非线性部分组成。

在LST-Dynamic架构中，我们采用经典的自回归（AR）模型作为线性组件。表示AR组件的预测结果为∈，AR模型的系数表示为war∈ℝqar，bar∈ℝ，其中qar输入矩阵上输入窗口的大小。注意到，在我们的模型中，所有的维度共享相同的线性参数集。AR模型被公式化表示如下：（与向量自回归的区别是，向量自回归不共享相同的线性参数集）



然后将神经网络部分的输出和AR部分的输出相结合，得到LST-Dynamic的最终预测结果：

其中表示模型在时间戳t时的最终预测结果（在t时刻进行预测得到t+h时刻的结果）



## 目标函数

训练的目标是得到智能体的最优参数θa和标准GRU等其他结构的最优参数θg。故LST-Dyanmic最终的目标函数应该由两部分组成：一部分相等于回归任务，记为J1(θg)，另一部分针对智能体，记为J2(θa)。

### 回归任务

可以直接用传统时间序列预测的方法来解决θg 的优化。例如，平方误差是许多预测任务的默认损失函数，相应的优化目标被公式化为：

θg表示我们模型的参数集， 𝛺Train 是用于训练的时间戳集合， ‖⋅‖𝐹是弗罗本尼斯范数，h是预测范围。传统的使用平方损失函数的线性回归模型称为线性岭，它等价于使用岭正则化的向量自回归模型。实验表明，在某些数据集中，线性支持向量回归（Linear SVR）优于线性岭模型[31]。线性支持向量回归和线性岭的唯一区别是目标函数。线性SVR的目标函数是：

其中，C和𝜖是超参数，针对不同的数据集选择合适目标函数。例如，我们假设𝜖=01，上述目标函数则被简化为绝对损失函数(L1-loss)，这种损失函数会使模型在真实的时间序列数据集中更具鲁棒性，我们将这部分的目标函数记为J1(θg)：

### 策略梯度

因为强化学习的目标是使跳跃策略分布下的奖励函数与熵正则项最大化，我们将这部分的目标函数记为J2(θg)，可以由下式计算得出：

H(π(a1:T)) 是熵项，能使智能体探索更多不同的地方，能够证明，在对语言序列的建模中，使用带熵项的强化学习模型比基于注意力机制连接的模型更好。

最终的损失函数被定义如下：

其中N为一个batch的时间结点数目。

## 优化策略

本模型的优化策略和传统时间序列预测模型相同。当输入窗口大小为q时，假定t时刻的输入为Xt = {yt-q+1,…yt}，则问题可以被转化为回归任务：输入矩阵Xt预测yt+h。由于强化学习中采样操作的不可微分性，我们通过Williams提出的策略梯度方法[32,33]对参数进行优化，从而可以进行梯度的反向传播，该公式如下所示：

通过这样的处理，我们最终可以使用小批量梯度下降法（Mini-Batch Gradient Descent, MBGD）或Adam来对模型进行优化。随机梯度下降法与批量梯度下降法的不同之处在于，批量梯度下降法考虑每次迭代的所有样本，一次迭代针对所有样本在目标函数上的偏差进行梯度更新。而随机梯度每次迭代针对一个样本的偏差对参数进行更新，这样能使训练速度更快，但容易陷入局部最小值，即鞍点。而小批量梯度下降结合了两种方法的特点，每次迭代选取少量样本进行梯度更新，从而能在更快的完成训练目标的同时更准确的找到下降的方向。

Adam算法由Diederik等人提出[34]，Adam算法结合了AdaGrad和RMSProp特点，不仅充分利用了梯度的一阶矩估计和二阶矩估计来自适应调整参数的学习率，还提升了算法在非稳态和在线问题上的性能。总体来说，Adam算法的优势在于保有计算高效性的同时减少了所需内存，适合解决非稳态目标、高噪声和梯度稀疏等问题。

## 本章小结

在本章中我们定义了时间序列预测的任务，将问题形式化。在模型的对比中描述了LST-Dynamic模型提出的意义和解决的问题，最后详细的描述了模型的结构与优化策略。

时间序列预测任务可以被定义为在一个时间窗口内的回归任务，通过在时间序列上滑动窗口可以得到未来一段时间内的时间序列数据。针对长期与短期周期关系混合的时间序列，我们提出了一种以神经网络为基础，结合强化学习和注意力机制的时间序列预测模型。模型首先通过卷积神经网络抽取时间序列的特征，之后针对短周期依赖和长周期依赖，分别设置了两个循环结构，一个基于循环门控单元和注意力机制，专注于捕捉短周期依赖，另一个基于循环门控单元的跳跃连接变体，结合强化学习，使模型能够学习到合适的跳跃间隔，即发现时间序列中潜藏的周期关系，减少了用户的参与同时能捕捉到周期的动态变化。通过传统时间序列预测方法中的自回归方法提升模型对数据尺度变化的敏感性。

因为采用了强化学习的方法，模型的目标函数可以被分为回归任务和策略梯度两部分，根据数据集的不同，回归任务采用绝对误差或平方误差作为目标函数。策略梯度的损失函数全部时刻采取动作的概率的负对数和此时可得到的奖励之积的总和，采用REINFORCE方法使策略梯度可以使用标准的反向传播算法更新，最终损失函数为两部分之差，使用L2正则化来抑制过拟合，采用随机梯度下降法来进行优化。

# 实验

本节中，我们将会在真实世界的数据集上评估我们提出的模型并检验LST-Dynamic各部分的有效性，我们以回答下列问题为目标进行实验。

* RQ1 LST-Dynamic模型为什么能取得优秀的成绩？
* RQ2带有Attention机制的LST-Dynamic结构中的不同组件会对模型造成怎样的影响？
* RQ3 LST-Dynamic的训练效率和可拓展性如何？
* RQ4超参会对LST-Dynamic模型的性能造成怎样的影响？

## 实验设置

### 数据集

为了有效评估我们提出的模型是否广泛适用于各种场景，分析各组件在其中起到的作用，我们使用了传统多元时间序列数据集和最新的新冠病毒数据集（COVID-19）作为评估用的数据集，所有数据集均可在网上获得。

(1)太阳能：阿拉巴马洲2006年光伏发电站收集到的太阳能数据，每个记录间隔为1天。共52560条记录，每条记录137个特征。

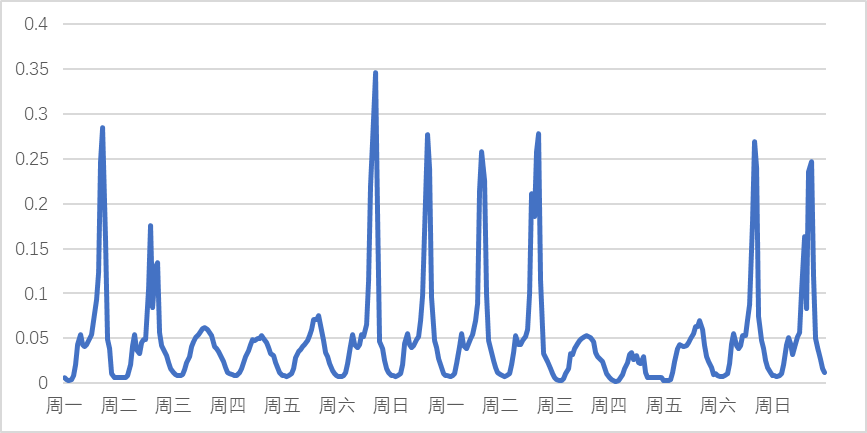
(2)交通：由加州运输部门提供，2015-2016两年间旧金山湾区传感器提供的道路占有率（0到1之间），每个记录间隔为1小时。共17544条记录，每条记录862个特征。

(3)电力：某地321户居民两年间的耗电量，以千瓦时（kwh）为单位，每个记录间隔为1小时。共26304条记录，每条记录321个特征。

(4)汇率：八个国家（澳大利亚、英国、加拿大、中国、日本、新西兰、新加坡、瑞士）1990到2016年间的汇率，每个记录间隔为1天。共7588条记录，每条记录8个特征。

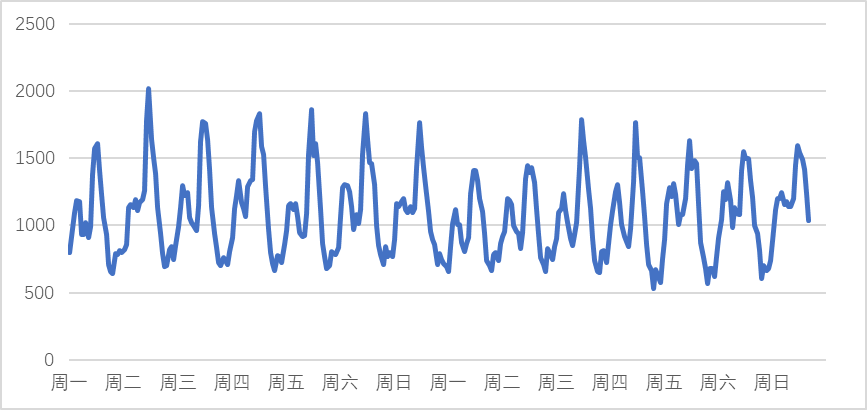
(5) COVID-19：全球265个地区自2020年1月20日到2020年4月17日之间新冠病毒确诊人数，每个记录间隔为1天。共87条记录，每条记录265个特征。

接下来我们将数据可视化，观察原始数据的周期特征，我们可以以此判断不同的结构是否有效，在怎样的场合下发挥了作用。接下来我们将结合图表对不同数据集进行具体分析，以此作为基准判断模型是否在特定的数据集上发挥我们想要的效果。



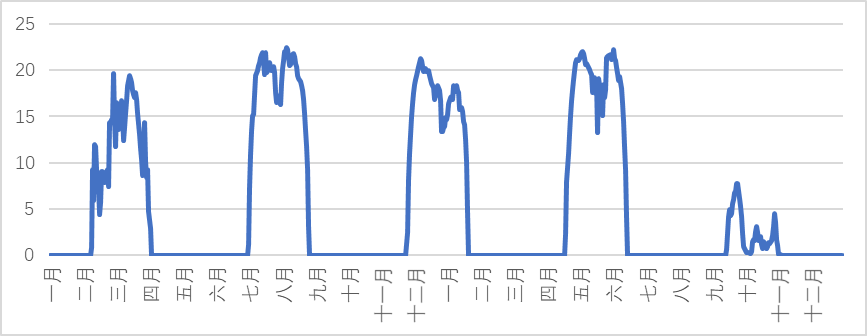
**图 4‑1 某传感器测得两周内道路占有率数据**

可以看到，交通数据集中的数据具有明显的两种周期模型，以一周为长周期，以一天为短周期，属于典型的长短周期混合的多元时间序列。



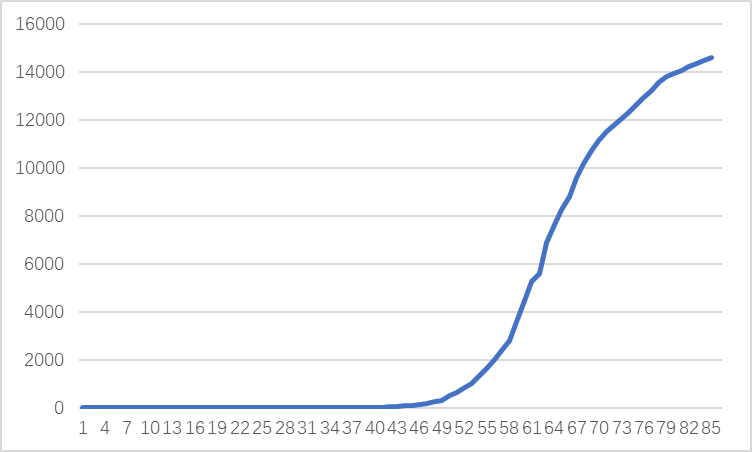
**图 4‑2某户两周耗电量数据**

电力数据集呈现出较明显的短周期模式，以一天为短周期，其长周期模式不明显。



**图 4‑3 太阳能两年数据**

太阳能数据集呈现出较明显的长周期模式，通常以五个月为长周期，其短周期模式动态性较强，不存在定式。



**图 4‑4 某地COVID-19确诊人数**

COVID-19数据集因为数据量不足和数据本身的局限，并没有呈现出较明显的周期特征。

通过对数据集的观察，我们可以发现在交通和电力两个数据集中有明显的长周期和短周期结合的特征，具体来说，两个数据集中都存在以周为单位和以天为单位的周期，这和真实世界中的耗电量和交通流量完全相符。而汇率数据集中难以观察到较长的周期特征，仅存在少量的短期局部特征。COVID-19数据集则难以观察到周期性特征。

### 对照组

我们采取以下方法和我们的模型进行对比

(1) LST-Skip：使用skip-RNN层的LSTNet模型

(2) LST-Attn：使用注意力机制代替skip-RNN层的LSTNet模型

(3) LST-Dynamic：我们提出的使用Dynamic-RNN层的模型

(4) LST-D&A：我们提出的同时使用Dynamic-RNN层和Attention机制的模型

### 评估方法

我们采用以下两种常用的指标作为评估模型的方法：

* 相对平方根误差(Root Relative Squared Error,RSE)
* 经验相关系数(Empircial Correlation Coefficient,CORR)

RSE是均方根误差的另一种版本，能够在不同规模的数据集上更为直观的评估模型，CORR表示两个序列的相关性，可以以此衡量两个序列间的周期性关系。RSE指标越低越好，CORR指标越高越好。

### 超参规定

我们使用L2正则化，dropout和提前停止(early stopping)等方法防止模型过拟合。L2正则化能有效减少模型权重的大小，从而避免某些特征产生过大的影响，dropout通过随机抑制中间层结点的表达和输出，从而达成一种隐式的特征自动选择的效果，提前停止法能较为直接的避免过拟合，但需要使用者参与，并存在一定的随机性。

窗口大小从{12,24,42,96,168}中选取，循环组件和动态循环组件的隐藏层维度从{50,100,200}中选取，卷积组件的隐藏层维度从{20,50,100}中选取。K默认为24，需要根据数据集大小做相应的改变。λ从{0.3,0.5,0.7}中挑选，在通常选用0.7时能得到最佳结果。我们在除输入与输出外的每一层后面进行dropout操作，比率设置为0.1或0.2。采用Adam算法加速模型的参数优化。

在汇率数据集中，所有模型的CNN和RNN隐藏层维度设为50，在COVID-19数据集中，LST-Skip和LST-Attn的CNN和RNN隐藏层维度设为50，LST-Dynamic和LST-D&A的CNN与RNN隐藏层维度设为100。

预测范围(horizon)从{3,6,12,24}中选取，范围越大，预测的难度越高。COIVD-19数据集因数据量不足的问题，预测范围只从{3,6,12}中选取进行测试。

通过改变奖励函数的设置，我们还可以改变Dynamic结构中循环单元的隐藏层维度，通常我们将最终的奖励函数设置为第K个时间点后奖励函数累计之和，这种情况下Dynamic隐藏层的输出维度只能和时间序列数据原始维度相同，我们可以将奖励函数设置为模型最终输出与真实值的对比，不考虑中间量，则可以通过调整Dynamic隐藏层维度来更好的获取时间序列的特征，但这样会带来一定的调参工作量，且通过实验对比，发现这两种奖励函数的设置区别并不太大，因此我们的实验中一般采用前一种奖励函数设置方法。

## 实验流程

### 实现环境

我们采用google提供的云上服务Colaboratory作为模型训练的平台，平台提供一个免费的Jupyter环境，提供最高32G内存和无限制CPU和受限的16G显存Tesla P100-PCIE免费使用。具体的环境细节如下：

显卡：Tesla P100-PCIE

语言：python 2.7

框架：pytorch

### 数据预处理

首先，我们对原始数据格式进行一定的规定，原始数据的每行代表一个时间结点观察到的所有特征，数据数值之间用逗号隔开，原始数据一般以txt格式存储，这样便于数据的读取和处理。我们此处以实验中使用的汇率数据集为例，汇率数据集每行有八个特征，代表了某一天各国的具体汇率，总共7588行，即总共记录了7588天的数据，数据集的格式和意义如下表所示：

**表 4‑1 汇率数据集的原始数据格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 澳大利亚 | 英国 | 加拿大 | …… | 新加坡 |
| T1 | 0.785500 | 1.611000 | 0.861698 | …… | 0.525486 |
| T2 | 0.781800 | 1.610000 | 0.861104 | …… | 0.523972 |
| …… | …… | …… | …… | …… | …… |
| T7588 | 0.720825 | 1.233905 | 0.744131 | …… | 0.690942 |

我们采用了SGD和Adam等梯度下降算法，并且需要对数据进行归一化，因此在实际训练前，我们要对数据进行一定的预处理。我们按照4.1.4中的规则设置好超参数，在数据进入模型实际开始训练前，应先进行分批和归一化操作，所有数据归一化到（0,1）区间中。随后，我们将所有数据集按照时间顺序分为训练集（60%），验证集（20%），测试集（20%）。在每种集合中，我们按照窗口大小，进一步将数据分为多个窗口，每个窗口的数据为实际的输入矩阵，设窗口最后时刻为T，T+h时刻的数据则为预测的目标。下面我们通过实际训练的结果对归一化的必要性进行分析。

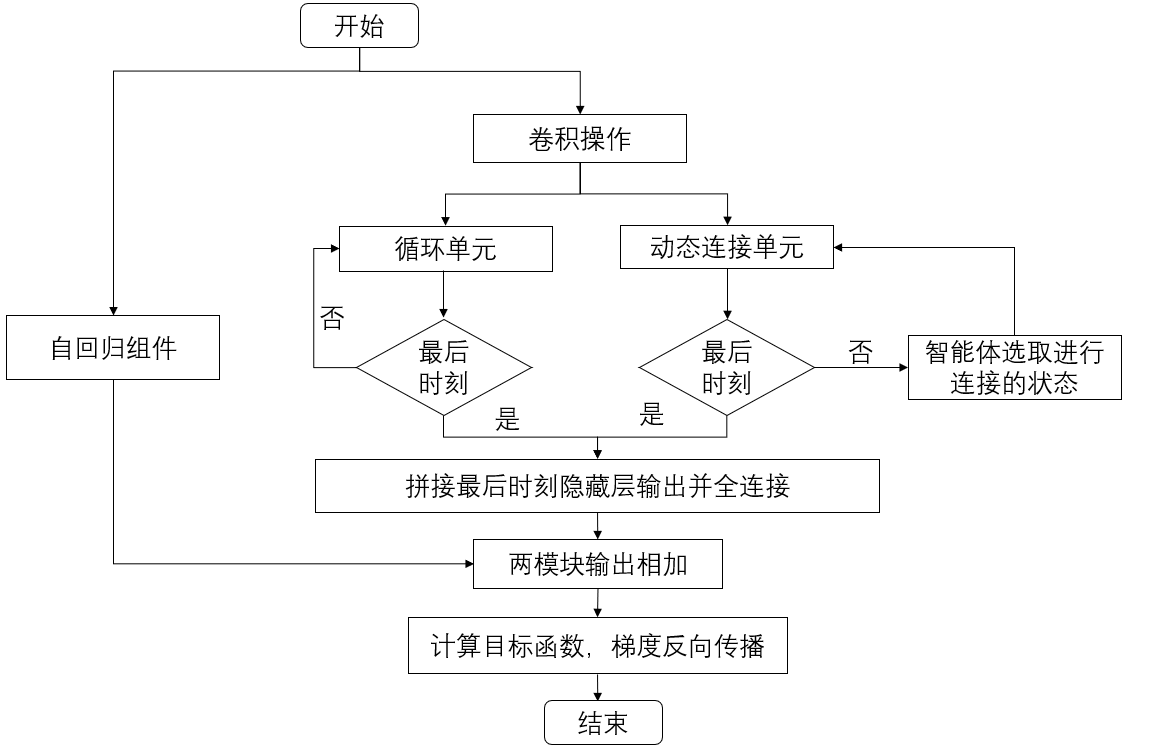
**表 4‑2 归一化操作对相关系数的影响**

**表 4‑3归一化操作对均方差的影响**

在不进行归一化处理时，模型根本无法在电力数据集上无法收敛，在太阳能数据集上性能明显变差。结合前一部分中数据的大小可知，归一化操作在特征值之间差异过大时对模型有较大的影响。

### 主要流程

数据预处理后即可正式进行训练，一个batch的数据在模型中训练的主要流程如下图所示：



**图 4‑5 模型训练主要流程**

神经网络除了针对目标函数添加正则项之外，还可以使用在训练过程中的小技巧来解决过拟合问题。例如dropout机制[35]通过随机设置隐藏层神经元输出为零来抑制某些特征的表达，从而在某种程度上实现特征选择。在本实验中，我们在除了自回归组件和输出层外的每一层后使用dropout，并且使用提前停止（Early Stop）来抑制过拟合。

## 实验结果

### 模型性能

下面我们将通过在不同大小的预测范围内进行实验得到的模型评价指标来对不同模型的性能进行分析，加粗项代表所有模型中的最优结果，LST-Skip和LST-Attn的数据均来自原论文 [3]。

**表 4‑4 模型在太阳能与交通流量数据集上的性能**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | | | 太阳能 | | | | 交通流量 | | | |
|  | | | 预测范围 | | | | 预测范围 | | | |
| 模型 | 指标 | | 3 | 6 | 12 | 24 | 3 | 6 | 12 | 24 |
| LST-Skip | | RSE | 0.1843 | 0.2559 | 0.3254 | 0.4643 | 0.4777 | 0.4893 | 0.4950 | 0.4973 |
| CORR | **0.9843** | 0.9690 | 0.9467 | 0.8870 | 0.8721 | 0.8690 | 0.8614 | 0.8588 |
| LST-Attn | | RSE | **0.1816** | 0.2538 | 0.3466 | **0.4403** | 0.4897 | 0.4973 | 0.5173 | 0.5300 |
| CORR | 0.9848 | 0.9696 | 0.9397 | 0.8995 | 0.8704 | 0.8669 | 0.8540 | 0.8429 |
| LST-Dynamic | | RSE | 0.1949 | 0.2472 | 0.3345 | 0.4841 | 0.4782 | 0.4915 | 0.4980 | 0.5109 |
| CORR | 0.9830 | 0.9706 | 0.9443 | 0.8735 | 0.8752 | 0.8676 | 0.8655 | 0.8559 |
| LST-D&A | | RSE | 0.1885 | **0.2414** | **0.3166** | 0.4467 | **0.4728** | **0.4843** | **0.4923** | 0.4992 |
| CORR | 0.9833 | **0.9713** | **0.9502** | **0.9012** | **0.8777** | **0.8712** | **0.8688** | **0.8671** |

**表 4‑5 模型在电力与汇率数据集上的性能**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | | 电力 | | | | 汇率 | | | |
|  | | 预测范围 | | | | 预测范围 | | | |
| 模型 | 指标 | 3 | 6 | 12 | 24 | 3 | 6 | 12 | 24 |
| LST-Skip | RSE | 0.0864 | 0.0931 | 0.1007 | 0.1007 | 0.0226 | 0.0280 | **0.0356** | 0.0449 |
| CORR | 0.9283 | 0.9135 | 0.9077 | **0.9119** | 0.9735 | 0.9658 | 0.9511 | 0.9354 |
| LST-Attn | RSE | 0.0864 | **0.0931** | **0.1007** | 0.1007 | 0.0226 | 0.0280 | 0.0356 | **0.0449** |
| CORR | 0.9283 | 0.9135 | 0.9077 | 0.9119 | 0.9735 | 0.9658 | 0.9511 | 0.9354 |
| LST-Dynamic | RSE | 0.0858 | 0.0951 | 0.1017 | 0.1021 | 0.0239 | 0.0297 | 0.0409 | 0.0573 |
| CORR | **0.9303** | **0.9221** | **0.9176** | 0.9036 | 0.9730 | 0.9654 | 0.9524 | 0.9353 |
| LST-D&A | RSE | **0.0844** | 0.0962 | 0.0989 | **0.0996** | **0.0218** | **0.0275** | 0.0362 | 0.0540 |
| CORR | 0.9288 | 0.9192 | 0.9152 | 0.9106 | **0.9751** | **0.9672** | **0.9532** | **0.9391** |

**表 4‑6 模型在COVID-19数据集上的性能**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COIVD-19 | | 模型 | | | |
| 窗口大小-预测范围 | 指标 | LST-Skip | LST-Attn | LST-Dynamic | LST-D&A |
| 42-3 | RES | 0.9942 | **0.9844** | 1.025 | 1.0194 |
| CORR | 0.8222 | 0.8256 | 0.8717 | **0.8866** |
| 42-6 | RES | 1.0059 | **0.9839** | 1.0201 | 1.0166 |
| CORR | 0.7921 | 0.8261 | 0.8297 | **0.8731** |
| 24-12 | RES | **0.9840** | 1.0232 | 1.0129 | 1.0005 |
| CORR | 0.7791 | 0.8441 | **0.8459** | 0.8268 |

LST-Dynamic模型相较于原论文中的两个模型，在交通数据集上CORR和RSE两个指标表现都有提升，在电力数据集的CORR指标上有提升，在太阳能数据集中有部分提升，汇率数据集上基本没有提升。

采用动态连接机制和时间注意力机制，即LST-D&A模型，由图表可直观观察到，在除电力数据集和COVID-19数据集外的所有数据集上大多取得最佳，虽然电力数据集上未取得最佳，但表现也普遍比LST-SKIP模型好，仅比LST-Dynamic模型差。

可以看到，在16组实验所得到的32项指标中，LST-Dynamic模型有3项指标最优，LST-D&A模型有20项指标最优，其余模型有9项指标最优，由此可见LST-D&A模型在多数数据集上有较好效果，深度强化学习和注意力机制的引入能提升模型在具有混合周期的多元时间序列中的性能。

在COVID-19数据集上，LST-Skip和LST-Dynamic模型表现都不佳，传统模型在此数据集上表现反而较好，这和我们的假设相符合，本模型在不具备明显周期性的时间序列上和传统方法相比并不具备优势。

接下来我们通过预测值和真实值的数据曲线对比，进一步分析模型的优缺点

**图 4‑6 LST-D&A 交通数据集预测范围为6时预测与真实值**

交通数据集具有较明显的混合周期特征，我们可以看到，模型能够较好的捕捉到这些周期关系，预测值的整体趋势与真实值相差无几。存在的问题主要集中在数据尺度变化上，即模型对于数据尺度变化依旧不够敏感，体现在预测值很难准确预测到真实数据的极值。

**图 4‑7 LST-D&A 太阳能数据集预测范围为6时预测值与真实值**

太阳能数据集作为长周期特征的代表，我们可以看到预测值和真实值曲线基本相符，因为数据尺度的变化不算太大，难以捕捉极值的问题在本数据集上体现的并不明显。

**图 18 LST-D&A 电力数据集预测范围为6时预测值与真实值**

电力数据集是短周期特征的代表，该时间序列的另一个特征是数值的变化频率较高，因此模型对数据尺度变化不敏感的问题体现的较为明显。尽管如此，我们依旧能发现模型对数据的周期能做到较为准确的捕捉，预测值和真实值的变化趋势基本相同。

**图 19 LST-D&A 汇率数据集预测范围为6时预测值与真实值**

汇率数据集周期性不太明显，本模型并非为此类数据集设计，预测值和真实值之间存在一定的延后，但整体趋势仍能捕捉到。我们引入的注意力机制增强了模型的普适性，使其在此类周期性不明显的数据集上的表现略有增强。

通过上述数据集预测值与真实值的对比，我们可以得出结论，LST-Dynamic能够有效提取到时间序列的混合周期关系，并且在提取长周期关系这一点上表现突出，动态连接结构的引入使模型更善于处理多元时间序列的周期波动，但尽管引入了注意力机制和自回归模块，模型在处理短周期关系和数据尺度变化较大的时间序列时的提升并不显著，仍有可改善的空间。

### 模型结构有效性

从LST-Skip和LST-Dynamic在交通和电力数据集上的对比我们可以发现，引入强化学习后的模型能更为精准的获取时间序列的周期关系，这一点从相关系数与根方差的提升中都能体现。从LST-Dynamic和LST-D&A模型的对比中，我们可以发现引入注意力机制能提升模型捕捉时间序列中短周期依赖的能力，这一点体现在引入注意力机制后，模型在汇率数据集上的表现普遍提升。以上两项对比可以证明，强化学习引入确实能有效提升模型对长期短期关系混合的多元时间序列上的表现。

在实验中我们曾考虑直接引入著名的Transformer结构来替代循环结构提取短期依赖关系，但在实际训练的过程中，Transformer结构往往需要更多的GPU资源和更长的训练时间才可能达到和本模型类似的效果，而层数较少的Transformer结构和本模型相比表现较差，因此放弃了Transformer结构。

### 超参影响

调整循环结构隐藏层维度造成的影响较大，其余参数影响较小，以汇率数据集为例，在horizon=3时，隐藏层维度为100，CORR仅为0.93，隐藏层维度都调整到50后，CORR提升到0.97，能明显看出隐藏层维度的设置受时间序列的维度影响较大，汇率数据集特征维度仅为8，因此模型的维度应相应的调小。

在引入动态连接机制后，我们减少了需要调整的超参，通过将循环跳跃层替换成动态连接组件，强化学习的智能体自行选择对预测结果有利的周期值。同时因为奖励函数的设置，动态连接组件的隐藏层输出应与原始时间序列相同。LST-Dynamic模型只需要给出周期的最大值，这两点优化减少了用户的干预，同时使模型更擅长提取动态周期关系，更具有灵活性。

## 本章小结

本章中我们对准备的数据集进行了介绍，并从每个数据集中任选了某一特征观察了时间序列的周期特征。我们使用了多个不同类型的数据集对提出的模型进行测试，这些数据集中有些具备长短周期混合的特征，有些则难以观察到周期关系，通过不同种类的数据集我们可以验证模型是否按照计划工作，是否在具备混合周期的数据集上表现出优秀的性能。通过消蚀实验，将模型的各个模块拆分提取后再进行训练，验证模型各个模块发挥的作用。

实验表明，LST-Dynamic模型更善于提取时间序列的动态混合周期关系，通过强化学习的方法的确能够捕捉到时间序列中的周期依赖。在引入注意力机制后，模型在仅表现出短周期性的时间序列上的表现有所提升，证明注意力机制有辅助模型提取短期依赖的作用。但在不具备明显周期关系的数据集上，模型表现的不太稳定，这一方面是因为数据量不够，强化学习未能探索到更优的策略，另一方面是模型并非为此类时间序列数据设计，不具备周期性的时间序列数据更适合使用传统方法解决问题。

# 总结与未来展望

## 总结

文中提出了用于多元时间序列预测任务的新型深度神经网络LST-Dynamic和LST-D&A，此模型结合了深度学习和强化学习的优点，通过智能体和环境之间的交互，让模型能够自主挖掘出时间序列中的周期性关系，减少了用户干预的同时，使模型更善于捕捉多元时间序列中的混合周期模式。我们采用了多种类型数据集对模型进行测试，并通过与其他同类模型的对比，证明了本模型确实能在具备混合周期的多元时间序列中取得更好性能。

## 未来展望

在未来的研究中，有以下几种方向可以作为拓展工作。

### 奖励函数

强化学习的奖励函数设置直接影响到智能体决策的正确与否。本模型强化学习部分的奖励函数设置较为简单，只考虑了最后输出，忽略了中间过程，导致奖励函数延迟较长，提取周期性关系的能力还有提升的空间。另一方面，评估时间序列预测结果的方法也不尽相同，是否存在一个更适合时间序列数据或更短延迟的奖励函数还有待研究。

### 普适性

本模型针对具有周期关系的时间序列数据，因此周期关系不强的时间序列数据上表现一般，但通过引入了注意力机制让模型在周期关系不明显的数据集上性能有所提升，因此模型的普适性可能还存在一定的改善空间，如何将模型推广到更多的应用场景是一个有趣的问题。

### 特殊点处理

本模型神经网络模块和自回归模块使用的是简单的加法集成，设置自回归模块的目的是解决神经网络部分对输出尺度不敏感的问题。通过预测值和真实值的对比图可以看出，自回归模块在某些输出数据尺度变化较大的特殊点上效果不佳。对于解决这一问题，我们有两个猜想：一是两个模块的集成方法并不能完全发挥自回归组件的性能，可以考虑更好的集成方式[36]。二是多增加一个处理特殊时期数据的模块，类似Facebook提出的prophet模型[37]，列出特殊时间区间，对特殊点做单独的处理。

# 参考文献

[1] 原继东, 王志海. 时间序列的表示与分类算法综述[J]. 计算机科学, 2015, 42(03):1-7.

[2] 杨海民, 潘志松, 白玮. 时间序列预测方法综述[J]. 计算机科学, 2019, 46(01):21-28.

[3] LAI G, CHANG W-C, YANG Y, et al. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks[C]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 2018:95–104.

[4] ZHANG G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50(50):159-175.

[5] QIU H, XU S, HAN F, et al. Robust Estimation of Transition Matrices in High Dimensional Heavy-tailed Vector Autoregressive Processes[C]. international conference on machine learning, 2015:1843-1851.

[6] KILIAN L. NEW INTRODUCTION TO MULTIPLE TIME SERIES ANALYSIS , by Helmut Lütkepohl, Springer, 2005[J]. Econometric Theory, 2006, 22(05):961-967.

[7] 张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, (01):36-46.

[8] ZHANG Y, ROUGHAN M, WILLINGER W, et al. Spatio-temporal compressive sensing and internet traffic matrices[C]. acm special interest group on data communication, 2009:267-278.

[9] RASMUSSEN C E. Gaussian processes in machine learning[J]. Lecture Notes in Computer Science, 2003:63-71.

[10] 沈旭东. 基于深度学习的时间序列算法综述[J]. 信息技术与信息化, 2019, (01):71-76.

[11] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[12] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. arXiv: Computation and Language, 2014.

[13] CAMPOS V, JOU B, GIROINIETO X, et al. Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks[J]. arXiv: Artificial Intelligence, 2017.

[14] 李洁, 林永峰. 基于多时间尺度RNN的时序数据预测[J]. 计算机应用与软件, 2018, 35(07):33-37+62.

[15] 张旭东, 杜家浩, 黄宇方, et al. 基于多尺度层级LSTM网络的时间序列预测分析[J]. 计算机科学, 2019, 46(S2):52-57.

[16] 张顺然 谢 周 吴. 基于时空关注度LSTM的行为识别[J]. 计算机学报:1-16.

[17] YU R, LI Y, SHAHABI C, et al. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting[M]. 2017.

[18] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(06):1229-1251.

[19] YANG J, NGUYEN M N H, SAN P P, et al. Deep convolutional neural networks on multichannel time series for human activity recognition[C]. international conference on artificial intelligence, 2015:3995-4001.

[20] BAI S, KOLTER J Z, KOLTUN V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling[J]. arXiv: Learning, 2018.

[21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[C]. neural information processing systems, 2017:5998-6008.

[22] QIN Y, SONG D, CHENG H, et al. A dual-stage attention-based recurrent neural network for time series prediction[C]. international joint conference on artificial intelligence, 2017:2627-2633.

[23] 李梅, 宁德军, 郭佳程. 基于注意力机制的CNN-LSTM模型及其应用[J]. 计算机工程与应用, 2019, 55(13):20-27.

[24] LI S, JIN X, XUAN Y, et al. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting[C]. neural information processing systems, 2019:5243-5253.

[25] HUANG S, WANG D, WU X, et al. DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting[C]. conference on information and knowledge management, 2019:2129-2132.

[26] SHIH S-Y, SUN F-K, LEE H-Y. Temporal pattern attention for multivariate time series forecasting[J]. Machine Learning, 2019, 108(8):1421-1441.

[27] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报, 2019, 42(06):1406-1438.

[28] 孙若莹, 范厚明, 赵刚. 基于强化学习的非线性时间序列智能预测模型[J]. 大连海事大学学报, 2017, 43(04):97-103.

[29] GUI T, ZHANG Q, ZHAO L, et al. Long Short-Term Memory with Dynamic Skip Connections[C]. national conference on artificial intelligence, 2019:6481-6488.

[30] KANI J N, ELSHEIKH A H. DR-RNN: A deep residual recurrent neural network for model reduction[J]. arXiv: Computational Engineering, Finance, and Science, 2017.

[31] VAPNIK V, GOLOWICH S E, SMOLA A J. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing[C]. neural information processing systems, 1996:281-287.

[32] WILLIAMS R J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning[J]. Machine Learning, 1992, 8(3):229-256.

[33] NACHUM O, NOROUZI M, XU K, et al. Bridging the Gap Between Value and Policy Based Reinforcement Learning[C]. neural information processing systems, 2017:2775-2785.

[34] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[J]. arXiv: Learning, 2014.

[35] GAL Y, GHAHRAMANI Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks[J]. arXiv: Machine Learning, 2015.

[36] 何正义, 曾宪华, 曲省卫, et al. 基于集成深度学习的时间序列预测模型[J]. 山东大学学报(工学版), 2016, 46(06):40-47.

[37] TAYLOR S J, LETHAM B. Forecasting at Scale[J]. The American Statistician, 2018, 72(1):37-45.

# 致谢

键盘敲击到致谢两字落下的时候，才有一种一切结束的实感，这篇毕业论文将会是我留在东华的最后记忆之一。从选题到完成论文的写作，期间有学习基础知识的艰难，有各项报告前的紧张，也有等待实验结果时的期待与欣喜，我在此过程中认识到了自己的不足，也获得了鼓励与前进的动力，更重要的是初步体验了科研工作的魅力。在毕业设计的过程中，还经历了考研和疫情，时间也从从秋日到了立夏，六个月的独特时光赋予了这篇论文特殊的意义，也让它成为了一种我独有的人生财富。

在本文的撰写过程中，我最需要感谢的是燕彩蓉老师。她作为我的指导老师，在我整个毕业设计过程中提供了莫大的帮助与鼓励。老师不仅为我们指出了研究的方向，鼓励我们看论文多思考，还不厌其烦的每周召开小组报告会倾听我们的每周进度。从定题到实验，从实验到论文写作，老师面面俱到地为我们提供一次次指导，指出我们的缺陷与不足，肯定我们的进步与成果。在实验结果不够理想的时候提出了几种模型可能存在的问题以及问题的突破方向，在我进展不顺时为我恰到好处的鼓励。在这个过程中我对自己研究的课题逐渐有了更深入的了解，对科研工作产生了一定的兴趣，也懂得了什么是严谨而认真的科研态度。没有老师的一路支持和帮助，这次的毕业设计就不会顺利的完成，我想在此感谢老师的指导和付出。

此外，还要感谢一直帮助我的陈一洲学长，对于我的疑问，学长总能热情而耐心的给出解答，分享他的经验和看法，从而拓宽了我的视野。除此之外，学长还帮助我修改论文的结构与格式，为我的论文写作打下了良好的基础。

同时，我还要感谢大学四年间遇见的每一位老师。没有你们专业而严谨的课程教学与硬核却有趣的课后作业，我也许不会那么坚决的选择计算机这条道路。这些课程经历是完成此次毕业设计必不可少的基础，也是开启我研究生涯或踏上社会的必由之路。

最后，我当然要感谢我的家人与朋友。没有家人的陪伴与开导，在疫情之中我也无法安心进行研究与实验，没有朋友间的互助与探讨，学习的压力和盲目的探索会让论文的写作更加艰难。正如没有全国的支援，湖北难以走出困境，没有家人与朋友相伴，

在今后的工作与学习中，我将汲取毕业设计中教训与领域，带着在东华四年里收获的知识和经验不断前行。