

本科生毕业论文(设计)



大规模时空数据辨识与 预测理论及其应用

学生姓名 刘潇奕

学 号 201813140011

学 院 数学与统计学院

专 业 信息与计算科学

指导教师 李顺杰

二〇二二年五月十日

声 明

本人郑重声明：

- 1、 以“求实、创新”的科学精神从事科学研究工作。
- 2、 本论文中除引文外，所有测试、数据和相关材料均为真实有效的。
- 3、 本论文是我个人在指导教师的指导下进行的研究工作和取得的研究成果，请勿用于非法用途。
- 4、 本论文中除引文和致谢的内容外，并未抄袭其他人或其他机构已经发表或撰写过的研究成果。
- 5、 关于其他同志对本研究所做的贡献均已在论文中作了声明并表示了谢意。

作者签名：刘 涛 英

日期：2022年5月10日

目 录

1. 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状与技术瓶颈	2
1.2.1 基于辨识的预测方法	2
1.2.2 基于复杂网络的预测方法	3
1.2.3 基于混合模型的预测方法	4
1.3 章节安排	4
2. 时空数据辨识与预测方法的预备知识	5
2.1 时空数据的稀疏辨识预测方法	5
2.1.1 稀疏辨识的几种描述	5
2.1.2 稀疏辨识的几种常用求解方法与优劣分析	5
2.2 时空数据的复杂网络预测方法	7
2.2.1 卷积神经网络的基础知识介绍	7
2.2.2 常用时空预测 CNN 框架的搭建与优化方式	8
2.2.3 复杂网络预测的局限性	9
2.3 本章小结	9
3. 时空数据辨识与预测一体化框架的构建	10
3.1 周期轴	10
3.1.1 周期发现与周期基函数的选取	10
3.1.2 周期辨识	11
3.2 趋势轴	12
3.2.1 周期均值趋势辨识	12
3.2.2 周期分段趋势辨识	12
3.3 突变轴	14
3.3.1 突变数据	14
3.3.2 网络搭建	14
3.4 耦合关系与算法流程	16
3.4.1 时空数据的长期辨识耦合关系	16
3.4.2 时空数据的短期预测耦合关系	17
3.4.3 时空数据辨识预测一体化算法流程可视化	18
3.5 本章小结	18
4. 基于时空辨识预测一体化模型的全球温度预测结果分析	19

4.1 数据及其预处理.....	19
4.1.1 数据简介.....	19
4.1.2 数据预处理.....	19
4.1.3 数据集的划分.....	20
4.2 预测结果分析.....	20
4.2.1 长期温度趋势辨识结果分析.....	21
4.2.2 短期温度精确预测分析.....	22
4.3 本章小结	23
5. 展望与总结	24
5.1 算法的优缺点总结.....	24
5.1.1 长期趋势辨识的优劣势分析.....	24
5.1.2 短期精确预测的优劣势分析.....	24
5.2 展望与未来	25
参考文献:	26
致谢	28
附录	I
A. 一些算法的流程图.....	I
B. 部分结果的可视化.....	IV
C. 核心代码.....	VI
(1) 数据预处理和评估代码.....	VI
(2) 四季切换辨识代码.....	VII
(3) abrupt-CNN 网络模型代码.....	IX
(4) 文中部分图像生成代码以及出处	X

基于可解释的复杂网络模型对大规模时空数据的预测

刘潇奕

南京信息工程大学 数学与统计学院, 江苏 南京 210044

摘要: 随着大数据时代的到来, 大规模时空数据的辨识与预测越来越成为各国学者研究的重点。本文提出了一个大规模时空数据辨识与预测的一体化通用框架, 该框架将大规模时空数据分为周期轴、趋势轴和突变轴分别进行辨识与预测。周期轴和趋势轴用于辨识长期趋势, 而突变轴将与辨识数据叠加用于短期精确预测。该框架将辨识方法与复杂网络完美结合, 克服了传统辨识精度低和复杂网络可解释性差的问题。本文将该框架应用于全球气温数据, 对全球各地的气温分别进行了 20 年的时空数据辨识和 1 年的时空数据预测。辨识和预测得到的 MAE 分别可以达到 1.74°C 与 1.58°C 。该结果好于当前大部分预测模型的结果。

关键词: 大规模时空预测; 稀疏辨识; 复杂网络; 全球气温

Prediction of large-scale spatiotemporal data based on interpretable complex network model

Xiaoyi Liu

School of Mathematics & Statistics, NUIST, Nanjing 210044, China

Abstract: With the advent of the era of big data, the identification and prediction of large-scale spatiotemporal data has attracted much attention by scholars from all over the world. This paper proposes an integrated general framework for large-scale spatiotemporal data identification and prediction, which divides large-scale spatiotemporal data into periodic axis, trend axis and abrupt axis for identification and prediction respectively. The periodic and trend axes are used to identify long-term trends, while the abrupt axis will be superimposed with the identification data for accurate short-term forecasts. The framework perfectly combines the identification method with the complex network, overcoming the problems of low accuracy of traditional identification and poor interpretability of complex networks. We applied this framework to global temperature data, and carried out 20-year data identification and 1-year data prediction for temperatures around the world. The MAE obtained by identification and prediction can reach 1.74°C and 1.58°C , respectively. This result is better than that of most current prediction models.

Key words: Large scale spatial-temporal prediction; Sparse identification; Complex network; Global temperature

符号说明

符号	说明与解释
$\ \cdot\ _0$	向量的零范数，即 $\sum x_i^0$ （补充定义 $0^0 = 0$ ）
$\ \cdot\ _p$	向量的 l_p 范数，即 $(\sum x_i ^p)^{1/p}$
$\ \cdot\ $	在不给定下标时，本文默认为向量的 l_2 范数，即 $\sqrt{\sum x_i^2}$
$\boldsymbol{W}[k]$	\boldsymbol{W} 是一个向量，该符号表示其第 k 个分量
$\mathcal{N}(\mu, \sigma^2)$	均值为 μ 方差为 σ^2 的正态分布
$\{w_k\}_{k=1}^P$	集合中 w_1, w_2, \dots, w_P 这些元素
$\text{diag}(\gamma_1^2, \gamma_2^2, \dots, \gamma_P^2)$	以 $\gamma_1^2, \gamma_2^2, \dots, \gamma_P^2$ 为对角线元素的对角矩阵
$f * g$	f 与 g 卷积， f, g 为函数为连续卷积， f, g 为矩阵为离散卷积
∇_α	∇ 为梯度算子，该符号表示函数在 α 方向的梯度
$\boldsymbol{A} \otimes \boldsymbol{B}$	\boldsymbol{A} 和 \boldsymbol{B} 是张量(tensor)，且维数相同，则 \otimes 表示 Hadamard 积
$\text{sum}(\boldsymbol{A})$	\boldsymbol{A} 是张量(tensor)，表示将张量中所有元素相加
$\text{pool}(\boldsymbol{A})$	\boldsymbol{A} 是张量(tensor)，表示对张量进行池化操作，池化定义见 2.2.1
$(\delta)_p$	$p \in (0,1)$ ，表示一个时间序列的 p 分位数
$\lceil \cdot \rceil, \lfloor \cdot \rfloor$	分别表示向上取整，向下取整
$\big\ _{\text{dim}=i}^k$	张量合并符号，表示按照字母 k ，在第 i 维合并张量
i.i.d	independent identically distributed 的缩写，表示独立同分布
s.t.	subject to 的缩写，表示受控于
$\arg \min(\max)$	arg 是 argument 的缩写，表示使得约束成立的自变量取值

1. 绪论

1.1 研究背景与意义

随着大数据时代的到来，大规模时空数据的预测备受各国学者的关注^[19,23,25]。大规模时空数据具有时间序列长、空间站点多、时空关系复杂等特点。因此，相比于传统的简单时序预测，大规模时空数据的预测难度大大增加。但是，大规模时空数据的预测是当今不可避免的一个问题，例如，本文选取的全球气温数据，如果能通过全球各地站点长期记录的数据，较为准确的预测温度的长期趋势和温度的短期剧变可以有效的避免灾难的发生。

温度的长期预测可以洞悉温度变化趋势，通过温度变化趋势可以制定相关政策防微杜渐。例如，Root 表示^[9]，近 100 年来，全球气温已经上升了 0.6°C ，而在局部地区（例如极地），这种现象更加严重。**图 1** 显示了全球一百多年来温度距平变化情况，可以发现全球温度距平正在波动上升，这体现了全球变暖在加剧。如果能够较为精确的预测每个地区温度变化趋势，制定合适的碳排放策略将会大大减缓全球变暖的趋势。

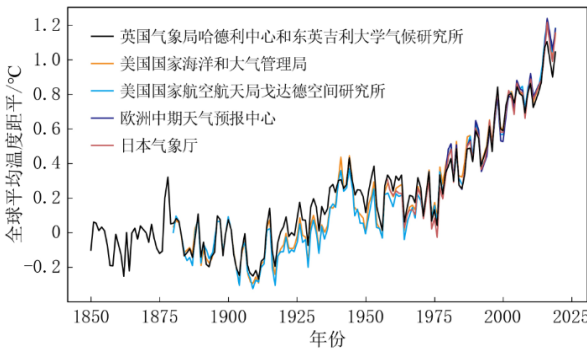


图 1 全球一百多年来气温距平变化情况

而在温度短期精确预测在很多极端天气预测中也都有举足轻重的地位。例如，我们熟知的 ENSO 是地球上最具影响力的海洋——大气气候现象之一^[10]。ENSO 会影响海洋生态系统，商业渔业，农业和公共安全（**图 2 左**），故而短时间精确预测 ENSO 的发生是十分重要的；又如，在去年 7·20 河南暴雨中，郑州的降水量超过了 500mm（**图 2 右**），正是因为有了提前的红色降雨精确预警，才避免了更大的损失^[11]。



图 2 自然灾害影响下的生活

（左：ENSO 影响下秘鲁渔场的鱼大量死亡，右：7·20 河南暴雨中郑州的一幕）

因此，在科技发达的今天，我们应当充分利用全球统计到的气温数据，利用大规模时空数据的预测方法，对长期温度趋势和短期温度变化进行预测。这对于预防自然灾害、气温突变有着重要的现实意义。

1.2 国内外研究现状与技术瓶颈

大规模时空数据预测一直是国内外学者研究的重点内容。一般来说，传统的时空预测有两种方法，一种是基于物理模型（model-based）的预测方法，另一种是基于纯数据（data-driven）的预测方法。这两种方法在国内外研究中均有长足的发展，但均有技术瓶颈有待突破。近年来，也有些学者尝试将两种传统的预测方法结合，形成一种复杂的时空数据预测方式，这种方式兼具两个传统预测方式的优点，更具实用性。本节将选取 model-based 中辨识方法、data-driven 中复杂网络方法以及他们的结合的一些方法详细阐述在大规模数据的时空预测研究现状与瓶颈。

1.2.1 基于辨识的预测方法

辨识方法是给定一些基函数，通过数据辨识给基函数赋予一定的权值，并用于预测的方法。国内外辨识方法的发展主要分为三个阶段。本节将分阶段讲述辨识的发展与各个阶段的技术瓶颈。

(1) 平凡辨识阶段（19 世纪末~20 世纪末）最早的辨识方法是在 19 世纪末高尔顿^[12]提出的。他发现父母身高可以预测孩子的身高，于是将孩子的身高与父母的身高这一现象拟合成线性关系，形成一元线性回归，成功预测了孩子的身高。这个方法也开启了通过辨识方法预测数据的先河。在 20 世纪初，对一元回归进行改进，进一步地提出多元线性回归，它描述了因变量与多个自变量的关系，可以更加精确的预测随时间发展的数据。该方法即为平凡的辨识方法，也称为最小二乘法，在当时医疗、生物、社会数据分析^[13]中均有较好的效果。但可能会造成共线性、过拟合等问题，需要在大规模变量的辨识中对变量进行取舍。

(2) 稀疏辨识阶段（1991 年~2001 年）在 20 世纪末，随着大数据时代到来，需要辨识的变量逐渐增多，传统的辨识方法弊端凸显较为严重。Robert 提出稀疏辨识问题，旨在给定误差范数小于 ε 的情况下，最小化系数的 l_0 范数，这是一个 NP 问题，人们只能寻找其近似解。Mallt 提出了基于贪心思想的正交匹配追踪(OMP)算法^[14]，该方法将一个 NP 问题转化为一个多项式时间内可以求解的问题，并给出了该问题的近似解。Robert 尝试对原问题条件修改，将原问题中最小化系数的 l_0 范数分别改为 l_1 范数与 l_2 范数，分别提出 LASSO 和岭回归方法给出了稀疏解^[15]。Tipping 等人提出相关向量机（RVM）^[16]，假设系数与误差均满足正态分布先验，从贝叶斯学习的角度进行回归，在其假设条件下又快又准的解决了原问题。这些求解方法很好的解决了大规模数据的辨识。但是，该方法在当时又两个技术瓶颈，一方面，由于这些方法或是基于贪心思想或是更改了原问题，得到的结果并非全局最优；另一方面，仅仅在代数领域的辨识应用范围不广，难以用于现代物理、生物领域。

(3) 应用辨识阶段(21 世纪初至今)在稀疏辨识之后,不同领域的学者开始根据自己所在的领域改进辨识方法,应用于不同的时空数据预测。2012 年, Liu 等人^[17]为总结常用的稀疏贝叶斯方法,包括了非正态分布先验的假设。2016 年, Brunton 等人^[18]提出阈值最小二乘法用于解决动力学辨识问题,成功通过稀疏辨识方法拟合了复杂的常微分方程组。2018 年, Dong 在北京智源大会的“人工智能的数理基础专题论坛”上提出 PDE-Net 可用于通过少量数据迅速求解偏微分方程,开创了人工智能求解偏微分方程的先河。2019 年, Yuan 等人^[19]提出了 IHYDE 模型,完成了分段动力学辨识问题。这些方法突破利用代数方法辨识,从动力学、分段等多角度进行时空数据的辨识,具有一定的进步性。不过这些方法并不具有普适性,仅能在某一种辨识(或某一个特定数据集中)中发挥较好的作用。

针对辨识的发展历程,对于大规模时空数据的辨识方法,本文主要总结其瓶颈在于以下桑方面:(1) 数据规模过大,直接利用稀疏辨识难以抓住问题的“主要矛盾”(即陷入局部最优的结果并不好);(2) 没有直接用于处理大规模时空数据的辨识方法;(3) 由于辨识函数一般取连续函数,难以辨识到数据的突变点。在本文中,针对全球气温数据,改进已有的辨识方法,尽力突破当下大规模时空数据辨识的瓶颈。

1.2.2 基于复杂网络的预测方法

复杂网络的时空数据预测方法与辨识方法有着本质的不同,它并不需要提供任何基函数,而可以通过大量的数据训练网络,并最终自发的学习了到数据的分布情况并进行预测。这是现代兴起的一种端到端的预测模式。本节将简单叙述复杂网络的发展与技术瓶颈。

可用于较为精确预测的多层神经网络可以追溯到 1982 年, Hopfield 提出了连续和离散的 Hopfield 神经网络模型^[21]。他利用该网络给当时的 NP-hard 问题——商旅问题(即 TSP 问题)提出了一个可行解,好于目前的绝大多数可行解,引起了世界范围内的极大关注。利用误差反向传播网络(BP 神经网络)进行大规模数据的预测好于传统方法。不过,由于该方法存在层数少、梯度爆炸、梯度消失、过拟合、需要频繁调参诸多问题,在 20 世纪 90 年代被 Vapnik 发明的支持向量机(SVM)所取代^[22]。复杂网络的再一次兴起是因为在 2012 年 ImageNet 的比赛中, Alex 教授利用其自己改进的卷积神经网络——AlexNet 以极大的优势获得冠军(高于第二名约 10 个百分点)^[1]。他通过卷积的预处理方式使得训练时间降低的同时大大提高了准确率和泛化性。正因为其强大的预测能力,在之后的很多研究聚焦于卷积神经网络(CNN)进行时空数据的挖掘。Jin 等人^[24]设计了地理语义集成神经网络(GSEN)进行城市热点时空预测;而 Elsaadani^[25]等人则通过 CNN 和长短时记忆网络(LSTM)的结合 ConvLSTM 预测了土壤水分的时空变化量。Zhang 等人^[2]利用双流卷积神经网络进行对流风暴临近的预报,避免了不必要的自然灾害损失。近期的研究深刻体现了复杂网络在时空预测中可以发现极端天气、预测异常值的优势。

不过尽管如此,正因为复杂网络的黑箱机制,我们发现该方法进行的预测不可避免的存在

一些难以突破的瓶颈。我们并不能通过该方法得到的预测数据得到启发，数据的发展趋势。同时，通过实验我们发现，该方法只能进行短时的精确预测，对于长期预测精度甚至不比传统模型。为了克服该缺陷，我们在下一节中会考虑将这辨识和复杂网络进行耦合，使得预测模型兼具精度和可解释性。

1.2.3 基于混合模型的预测方法

考虑到传统方法难以突破的瓶颈，近年来，有些学者尝试将两种传统的预测方法结合，形成一种复杂的时空数据预测方式。这种预测方式兼具传统预测方法的优点，既能取得较好的精度，又能兼具一定的可解释性。Kaiser^[26]等人于 2017 年提出将 SINDy 和 MPC（模型预测控制）结合，形成了一种 model-based 与 data-driven 的复合模型，可以用于少量数据情形下的精确预测；Beltagy 等人^[27]提出将随序列长度线性扩展的注意机制结合入 transformer，这种将 data-driven 融入 model-based 方法具有更好的翻译效果；Zhang 等人^[28]利用正交正则化残差 CNN 预测空气污染数据，并在实验证实比单独利用 ResNet 效果要好。这些都体现了混合预测模型的优势。基于此，本文主要针对全球气温数据进行辨识与预测，该数据集是一个大规模的时空数据，本文旨在克服辨识和神经网络各自的缺点，形成一个一体化的预测模型，得到比使用单独模型更好的效果。

1.3 章节安排

本文主要内容包括预测方法的预备知识、时空一体化模型构建、全球气温数据预测结果分析和展望与总结四个部分。本文的以下篇幅将对这四个部分通过四章内容具体展开，具体思路与内容如下：

第 2 章：介绍几种常见的时空数据的预测方法，包括稀疏辨识的预测方法（LASSO, STLS, 稀疏贝叶斯方法）和复杂网络（CNN）的预测方法，我们将详细介绍这些方法在预测中的优劣，这些方法将为后续时空一体化辨识模型的搭建打下基础。

第 3 章：该章是本文的核心章节，在本章将在上一章的基础上提出时空数据辨识与预测一体化预测模型的框架，该框架可以直接用于处理大规模时空数据。我们会详细介绍该框架中进行辨识和预测的三个轴：周期轴、趋势轴和突变轴，并同时介绍这三个轴之间的耦合关系，并推导出本文的两个核心算法：随四季切换的辨识算法和突变 CNN 算法。从该框架中可以充分体现辨识方法与复杂网络方法的耦合关系。

第 4 章：本章将利用提出的模型对全球气温数据进行测试。我们会对预测结果进行可视化，并同时对比常用的数据预测方法，说明该方法在预测大时空规模数据中的优越性。最后，根据温度预测情况，对不同地区提出合理的建议。

第 5 章：总结本文提出的算法的优缺点，展望在该领域的发展前景。对后续该方面的工作提出自己的建议与想法。

2. 时空数据辨识与预测方法的预备知识

为了便于后文时空数据预测一体化框架的搭建，本章首先介绍一些在时空预测中常用的方法和预备知识。其中主要包括 model-based 的预测模型稀疏辨识方法和 data-driven 的预测模型复杂网络方法。

2.1 时空数据的稀疏辨识预测方法

2.1.1 稀疏辨识的几种描述

时空数据的辨识问题可以描述为如下形式：

$$\mathbf{T}_{ij} = \Phi(\mathbf{t})\mathbf{W}_{ij}, \quad (2.1)$$

其中， $\mathbf{T}_{ij} \in \mathbb{R}^M$ ，表示在空间维度 i, j 处的时间序列数据，序列长度 M 即为数据中时间的长度。 $\Phi(\mathbf{t}) = [\varphi_1(\mathbf{t}), \varphi_2(\mathbf{t}), \dots, \varphi_P(\mathbf{t})] \in \mathbb{R}^{M \times P}$ ，表示关于时间序列 $\mathbf{t} \in \mathbb{R}^M$ 的一个字典， P 表示字典中的基函数数量 ($M \gg P$)。 $\mathbf{W}_{ij} \in \mathbb{R}^P$ ，是我们在空间维度 i, j 处寻求的一个稀疏向量。通过该向量可以选取 $\Phi(\mathbf{t})$ 字典中的基函数，以表示不同地区时间数据的变化趋势。

可以看出，式(2.1)大概率是一个矛盾方程组，一般用最小二乘求解 \mathbf{W}_{ij} ，但最小二乘结果往往是不稀疏的，现在我们希望稀疏辨识希望找到“最稀疏”的 \mathbf{W}_{ij} ，即

$$\arg \min \|\mathbf{W}_{ij}\|_0, \quad s.t. \Phi(\mathbf{t})\mathbf{W}_{ij} = \mathbf{T}_{ij}, \quad (2.2)$$

使其与最小二乘解结果相似，同时能抓住问题的“主要矛盾”（例如在该问题中，我们可以从字典中找到时间数据变化的根本原因）。此外，由于(2.2)中并不能保证等式严格成立，我们在式(2.2)表示中允许一定的误差 ε ，因此，稀疏辨识一般不使用(2.2)式，而是用以下两种形式进行表示：

$$\arg \min \|\mathbf{W}_{ij}\|_0, \quad s.t. \|\mathbf{T}_{ij} - \Phi(\mathbf{t})\mathbf{W}_{ij}\|_p^p < \varepsilon, \quad (2.3)$$

或：

$$\arg \min \|\mathbf{W}_{ij}\|_0, \quad s.t. \mathbf{T}_{ij} = \Phi(\mathbf{t})\mathbf{W}_{ij} + \mathbf{\Xi}, \mathbf{\Xi} \sim \mathcal{N}(0, \sigma^2 I). \quad (2.4)$$

稀疏辨识的核心内容就是求解(2.3)-(2.4)中“最稀疏”的 \mathbf{W}_{ij} 。

2.1.2 稀疏辨识的几种常用求解方法与优劣分析

(2.3)-(2.4)中，无论哪种稀疏辨识表示的问题，一般均有无穷多组解。想逐个寻找“最稀疏”的 \mathbf{W}_{ij} 再找出最稀疏的那个是不可能的，这是一个 NP-hard 问题。本节主要介绍三种传统求解该问题的思路：(1) 利用贪心思想直接优化，将问题降为多项式范围可解决的问题。(2) 转化优化范数，将 l_0 范数转化为 l_p 范数并进行求解。(3) 增加先验条件，利用统计方法（主要是 Bayes 方法）进行优化。下面我们将逐个讨论这些思路所对应的算法，以及他们的优缺点。

(1) 贪心求解法

贪心是一种思想,即当一个问题有多步需要求解时,总是选取当前步最好的结果选择作为输出。这样的思想有助于我们快速求解问题,一般都能将一个 NP 问题转化为多项式范围求解。但是这样也会无可避免的陷入局部最优。较为常用的贪心辨识算法为匹配追踪法^[14]和阈值最小二乘法^[18]。匹配追踪方法的初始化向量是零向量,通过不断寻求局部最优,增加将零向量的分量不断更改为非零,从而得到稀疏向量。而阈值最小二乘的初始向量是一个非稀疏向量,通过删去该向量中较小的系数,并不断的重复迭代,得出最终的稀疏向量。这两个方法的思路恰好相反,但是它们都是较为典型的贪心思想,由于这两种方法较为简单,限于篇幅,本文不给出详细过程,仅在附录中给出算法流程(详见附表 1 与附表 2)。但是这种方法不具备泛化性,当更改误差 ε , 阈值 λ 或是将数据做一个微小的变动时,其稀疏结果可能会有较大的改动,这并不适合直接用于时空数据的预测,需要进一步的改进。

(2) l_p 范数求解法

(2.3)-(2.4)难以求解的根本原因在于该问题的优化是非凸优化。Robert 提出,若将原问题中的 l_0 范数更改为 l_1 范数,该问题可以转化为凸优化问题^[15],以便求解。他将式(2.2)的优化条件改成如下优化:

$$\arg \min \|T_{ij} - \Phi(t)W_{ij}\|_2^2, \quad s.t. \|W_{ij}\|_1 < \beta, \quad (2.5)$$

其中 β 是一个常数,用于控制 W_{ij} 的 l_1 范数。类似的,若将(2.5)式 W_{ij} 的 l_1 范数改为 l_2 范数,就是 Robert 提出岭回归方法,限于篇幅不再重复。

求解该问题有两种优化方法,一种是当时 Robert 自己使用的坐标轴下降法,这是一种凸优化方法;另一种是 Bradley 提出的最小角回归方法,它们均可以较好的解决该问题。但是若将该方法直接运用于时空数据的预测会有两个问题:其一是 β 是一个超参数,需要频繁调参才能得到较好的结果;其二是该方法求解的是(2.2)-(2.4)的近似问题,并非原问题,并不能得到理想的稀疏解,因此该方法也需要进一步改进。

(3) 统计学习求解法

由于传统方法难以直接解决该问题,因此人们尝试加入先验信息,从统计学的角度进行求解。较好的稀疏求解方法是利用 Bayes 方法,该方法主要针对式(2.4)进行稀疏辨识。对于稀疏向量 $W_{ij} = [w_1, w_2, \dots, w_p]_{ij}$, 稀疏 Bayes 方法给出先验假设: $\{w_k\}_{k=1}^p$ i.i.d 且分别服从 $\mathcal{N}(0, \gamma_k^2)$, 根据 Bayes 公式,我们得到 W_{ij} 的后验概率:

$$p(W_{ij} | T_{ij}; \gamma) = \frac{p(T_{ij} | W_{ij})p(W_{ij}; \gamma)}{p(T_{ij}; \gamma)}. \quad (2.6)$$

根据先验,我们可以进一步得到:

$$p(T_{ij} | W_{ij}) \sim \mathcal{N}(\Phi(t)W_{ij}, \sigma^2 I), \quad (2.7)$$

$$p(\mathbf{W}_{ij}; \gamma) \sim \mathcal{N}(0, \Gamma), \quad \Gamma = \text{diag}(\gamma_1^2, \gamma_2^2, \dots, \gamma_p^2), \quad (2.8)$$

$$p(\mathbf{T}_{ij}; \gamma) = \int_{\mathbf{W}_{ij}} p(\mathbf{T}_{ij} | \mathbf{W}_{ij}) p(\mathbf{W}_{ij}, \gamma). \quad (2.9)$$

可以发现, (2.7), (2.8)的概率均为正态分布, 而(2.9)的概率是两个正态分布的卷积, 所以 $p(\mathbf{T}_{ij} | \mathbf{W}_{ij}), p(\mathbf{W}_{ij}; \gamma), p(\mathbf{T}_{ij}; \gamma)$ 都是正态分布。而在(2.8)中, 计算后验概率仅涉及到正态分布的乘法, 因此后验概率 $p(\mathbf{W}_{ij} | \mathbf{T}_{ij}; \gamma)$ 也是一个正态分布, 我们仅需要计算该正态分布的均值和方差即可, 这对于不同的 γ_k^2 , 一般会有不同的结果。我们针对得到的后验分布利用极大似然估计, 即可估计出 \mathbf{W}_{ij} 的后验概率。该方法在给出先验的情况下较好的解决了原问题。不过该方法较为依赖先验条件, 在数据不满足先验条件时, 该方法失效。

2.2 时空数据的复杂网络预测方法

时空数据的复杂网络预测方法较多, 这种预测方法一般分为基于卷积神经网络 (CNN) 和循环神经网络 (RNN) 两种复杂网络 (包括 LSTM, GRU 等)。他们都是通过一段已知的时间序列进行后续一段时间的序列的预测, 但 CNN 可以通过卷积的操作将空间数据卷积, 而相比较而言, RNN 仅能处理一维的时间序列问题, CNN 更适合处理时空数据。因此本文主要介绍利用 CNN 进行的时空预测方法。

2.2.1 卷积神经网络的基础知识介绍

在进行 CNN 框架搭建之前, 首先需要介绍一些卷积神经网络的基础知识, 以方便后续对框架的解释与说明。

(1) 连续卷积与离散卷积

连续卷积是信号处理中的一个例子: 给定输入信号 $f(\tau)$ 和系统响应信号 $g(\tau)$, 则系统输出可以表示为:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau. \quad (2.10)$$

这是一个连续卷积的例子, 在信号处理、常 (偏) 微分方程都会用到。如果我们将上式离散化, 用于时空数据的矩阵 (或进行图像卷积), 则可以得到离散卷积关系:

$$(F * G)_{m,n} = \sum_{i=-\lfloor \frac{k}{2} \rfloor}^{\lfloor \frac{k}{2} \rfloor} \sum_{j=-\lfloor \frac{k}{2} \rfloor}^{\lfloor \frac{k}{2} \rfloor} F_{m-i, n-j} G_{i,j}. \quad (2.11)$$

其中 F 为二维图像 (或某一时间切片下的空间矩阵), 而 G 为大小为 k 的卷积核。利用同样的方法也可以得到三维卷积关系式 (即当 F, G 维数为 3 时)。

通过卷积操作, 一方面可以建立空间数据之间的关系。另一方面, 可以不断地对数据进行降维, 以获得更加高级的, 抽象的数据。因此, 卷积操作这在神经网络中意义重大。

(2) 平均池化与最大池化

池化操作是为了进一步减少原数据的维数，从而进行的操作。池化就是使用一个固定大小的划窗（当数据为 3 维时为“划体”）在输入上移动，每次将划窗（体）中的元素作为一个元素输出。当使用平均池化时，我们会输出划窗（体）中所有数据的平均值，而当使用最大池化时，则会输出划窗（体）中的最大值（本文后面将用 `pool()` 来表示池化）。

（3） 填充、步长与通道

在进行卷积核池化操作时，不难发现，这种操作会改变原矩阵的大小。有时我们不希望矩阵大小被改变，有时我们希望矩阵变得更小。这就引出了填充和步长两种参数。填充表示在做卷积或是池化前在原矩阵的边缘进行填充（常用 0 填充），以保证在在卷积、池化操作之后矩阵大小不变；而步长则为每次卷积核、划窗移动的长度，在不设置步长时，默认为 1，当设置步长为 2 或是更多时，则会得到更小的输出矩阵。

通道数指的是在每一次卷积中并不会仅用一个卷积核，当用了两个以上卷积核是即为多通道卷积，多通道卷积可以增加数据的丰富性，提取更多的时空特征。

2.2.2 常用时空预测 CNN 框架的搭建与优化方式

CNN 起源于图像数据的分类，通过对图像的卷积提取特征，配合全连接层输出，可以得出较好的结果。在时空数据预测中，我们可以将时空数据类比于图像，每一个时间切片中的空间格点即为图像的每一个像素，以此进行卷积、预测。下图描述了 AlexNet 的框架。

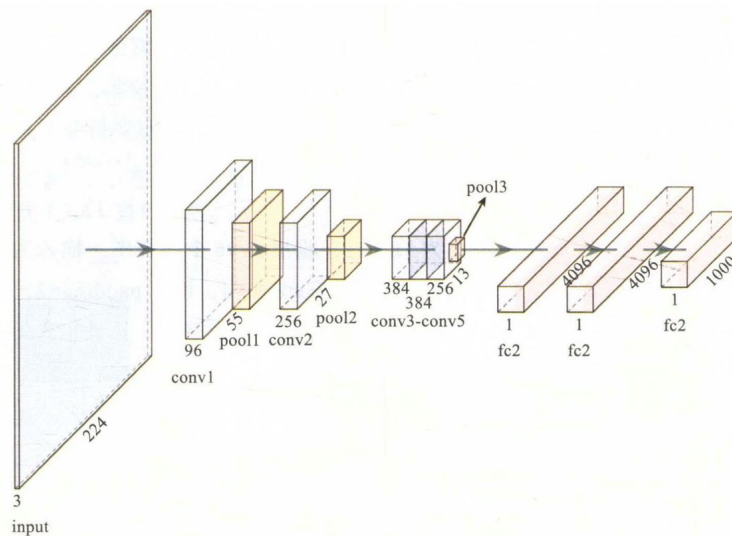


图 3： AlexNet 的基本构架

若将输入的图片换为时空数据，输出序列换为预测序列，利用 CNN 即可实现一般情况下的时空预测。一般常用的时空预测的 CNN 均是采用如上图所示的两个部分，第一部分是卷积部分，通过卷积层、池化层的堆叠降维，选取适合预测的属性；而第二个部分是全连接部分，该部分与 BP 神经网络相同，通过卷积部分已降维的结果最终计算出预测结果。

当前几乎所有的神经网络均使用的是误差反向传播的优化模式，卷积神经网络也不例外。但优化方法有较多种，其中最简单的直接使用梯度下降法，即对于每一个参数 θ_j 以及神经网络 $J(\theta)$ 有：

$$\theta_{j+1} = \theta_j - \alpha \cdot \nabla_{\theta_j} J(\theta), \quad (2.12)$$

其中 α 是学习率，是一个需要自己设置的超参数。该方法沿着梯度进行参数更新，虽然简单明了，但该方法有明显的两个问题：其一是学习率调参难度大。选取不恰当的学习率进行训练网络会导致训练时间过长（学习率小）、无法收敛到局部最优解（学习率大）等问题，其二是该方法是一个线性收敛的算法，在训练速度上有待提升。为了解决这两个问题，近年提出了 Adam 法，共轭梯度法，Adam 可以自适应的完成学习率的寻找并且有很好的鲁棒性。共轭梯度法可以在不求海森逆矩阵的情况下比梯度下降更快的找到最优解（即迭代收敛速度更快）。限于篇幅，详细的算法内容可见附录中附表 3 与附表 4，这些方法的出现也都为深度神经网络的训练打下了基础。

2.2.3 复杂网络预测的局限性

虽然当前复杂网络方法已经非常成熟，已经有专门的框架对其描述（例如 Tensorflow 和 Pytorch），但是利用复杂网络直接进行预测的局限性是多方面的。一方面是复杂网络自身参数的不可解释性，这导致了整个神经网络训练是一个黑箱，我们不能透过它看到时空数据的变化趋势。也正因为如此，该方法仅能进行一小段时长的预测，远低于辨识方法的预测时长。另一方面，调参的难度并没有因为自适应学习率方法的出现而得到简化，因为构造神经网络本身就是一个巨大的参数系统，我们需要不断调整网络结构以得到更好的效果，因此在训练时长较长的深度学习的今天，复杂网络进行预测的方法仍然任重而道远。

2.3 本章小结

本章介绍了两种不同的预测模型：稀疏辨识方法和复杂网络方法。对于稀疏辨识方法，本章介绍了贪心法、 l_p 范数求解法和统计学习求解三种方法，我们发现这些方法虽然一定程度上能解决问题，但均存在一定的局限性，在使用方法时需要具体问题具体分析。而对于复杂网络方法，本章重点介绍了卷积神经网络的架构的前向传播，优化中反向更新的一些知识。我们也可以从中发现虽然复杂网络方法不需要先验条件就可以进行预测，但是因为会出现调参、训练时间长等问题，对于大规模时空数据预测问题挑战性仍然很大。在下一章中，我们将改进一些传统的算法，建立大规模时空辨识与预测一体化的通用框架，以便在现实数据中应用。

3. 时空数据辨识与预测一体化框架的构建

从第二章可以看出,传统的辨识方法、复杂网络预测方法都有一定的局限性,并不能直接用于大规模数据的时空数据的辨识。本章以全球近 70 年的各地气温数据为例,改进传统的辨识方法和复杂网络方法:在辨识方面,本课题受到 Yuan^[19]等人研究的随时间切换动力学辨识的启发,提出随四季切换的辨识方法,该方法可以从周期、四季温度变化趋势两个角度进行辨识,将辨识过程随着四季进行切换与叠加,使得辨识结果更具可解释性;在复杂网络方面,本课题通过计算辨识数据与真实数据的残差寻找突变数据,并构造 abrupt-CNN 来进行温度短期的突变情况的预测。

综合辨识与复杂网络方法,本章进一步提出了时空数据辨识与预测一体化框架。该框架可以从三个维度进行预测,分别是周期轴、趋势轴和突变轴,并最终对这三个维度进行耦合。该框架包含了改进的辨识与复杂网络方法,并将他们有机结合,既可以用于气温的长期辨识,也可以用于短期温度的精确预测,本章将详细阐述该框架的构建过程。

3.1 周期轴

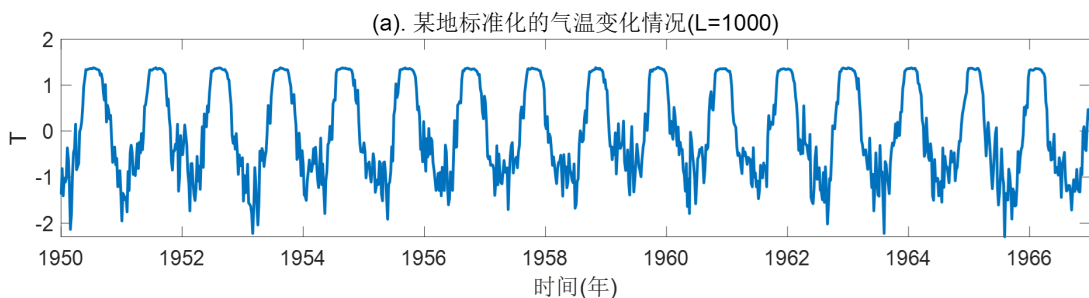
时空数据预测中,周期是一个十分重要的规律,正是稀疏辨识可以正确的辨识周期才使得长时间预测效果好于复杂网络。因此,对于周期轴的构建,首先我们要寻找该函数的周期,其次要选取辨识周期函数,最后对数据进行有效的稀疏辨识。

3.1.1 周期发现与周期基函数的选取

时空数据常常充满噪音,在噪音中直接寻找周期是极为困难的。本文选取了利用傅里叶变换寻找周期的方法,这在信号处理中是一种常用的手段。不妨设数据长度为 L ,本文将原时空数据利用傅里叶变换转化为频域中的数据,根据傅里叶变换的相关知识可知,假设频域中某个高频对应的点对应的横坐标为 x ,则一个周期为 L/x 。本文随机选取了一个地区的气温并对其进行标准化(如图 4.(a)所示)。对于该地区,对其做离散傅里叶变换

$$Y(f)[k] = \sum_{n=0}^{N-1} T[n]e^{-j(2\pi/N)kn} \quad (0 \leq k \leq N-1), \quad (3.1)$$

其中, j 在这里表示虚数单位。同时,画出 $Y(f)-f$ 频谱图像(如图 4.(b)所示)。



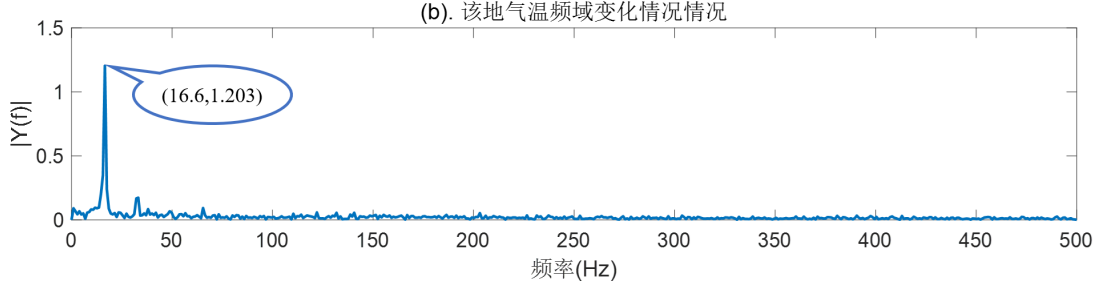


图4 某地标准化的气温与其频谱数值

由上图中高频点 (16.6, 1.203) 可以发现, 该地温度变化周期为 $L/x = 1000/16.6 \approx 60.24$. 与后文提到的一年数据周期 ($T = 61$, 详见 4.1 节) 基本契合。通过类似的方法, 可以找到任意时空数据的周期, 以用于辨识。

在确定数据周期后, 将选取周期函数进行辨识。一般利用傅里叶基进行周期辨识, 即: $\sin(wt), \cos(wt), \dots, \sin(kwt), \cos(kwt), C$ (其中 w 为基函数的频率, $w = 2\pi/T$, C 为常数), 这是因为任意周期函数都可以用正余弦构成的无穷级数所表示, 在确定周期的情况下表示更加简单。(需要注意的是, 如果不确定数据的周期, 仍然可以进行系数辨识。但是由于不知道周期, 往往会用大量的基底进行辨识, 耗时以及精度都会低于已知周期的情况) 在下一节中, 我们将该基底进行时空数据的周期辨识。

3.1.2 周期辨识

由于大规模时空数据不仅仅包含周期, 而且会包含趋势变化以及数据的“突变”(以温度数据为例, 大部分地区在一年内都会经历一些例如寒潮、干旱等自然现象导致温度突变) 如果直接辨识时间序列或是辨识某一个周期内的序列都会造成过拟合, 为了解决这种情况, 本文对于温度数据定义日平均温度, 并依此进行辨识, 即

$$\bar{T}_{i,j}^d = \frac{1}{Y} \sum_{y=1950}^Y T_{i,j}^{y,d}, \quad (3.2)$$

其中, $T_{i,j}^{y,d}$ 表示在经纬度为 i, j 处, 第 y 年, 第 d 天的温度; Y 辨识中的最大年份; $\bar{T}_{i,j}^d$ 则表示在经纬度为 i, j 处, 第 d 天的日平均温度。同时, 定义日平均温度对应矢量 $\bar{\mathbf{T}}_{ij}$, 这表示经纬度为 i, j 处日平均温度序列。之后, 利用 $\bar{\mathbf{T}}_{ij}$ 作为稀疏辨识模型(2.4)的输入, 选取傅里叶基(k 取 6), 并利用上文提到的稀疏 Bayes 方法, 通过(2.6)-(2.9)式计算、输出经纬度为 i, j 处的日平均稀疏变量 $\bar{\mathbf{W}}_{ij}$, 该稀疏向量即为经纬度为 i, j 处的温度变化主要特征。本文以加拿大首都温哥华为例, 其周期辨识的具体过程由图 5 所示。

观察图像图 5, 并思考整个算法流程, 可以发现: 相比于直接使用 \mathbf{T}_{ij} 进行辨识, 利用 $\bar{\mathbf{T}}_{ij}$ 辨识具有较多好处。一方面, $\bar{\mathbf{T}}_{ij}$ 相比于 $\mathbf{T}_{i,j}^y$ 更具代表性。 $\bar{\mathbf{T}}_{ij}$ 是一长段时间温度的平均, 是一年中不同日期温度的综合体现。另一方面, 由于 $\bar{\mathbf{T}}_{ij} - t$ 的图像相比于 $\mathbf{T}_{ij}^y - t$ 的图像更加平滑, 使得稀疏辨识模型(2.4)更易提取主要特征, 结果稀疏且更具解释性。因此本文将使用该方法提取各地区温度特征。

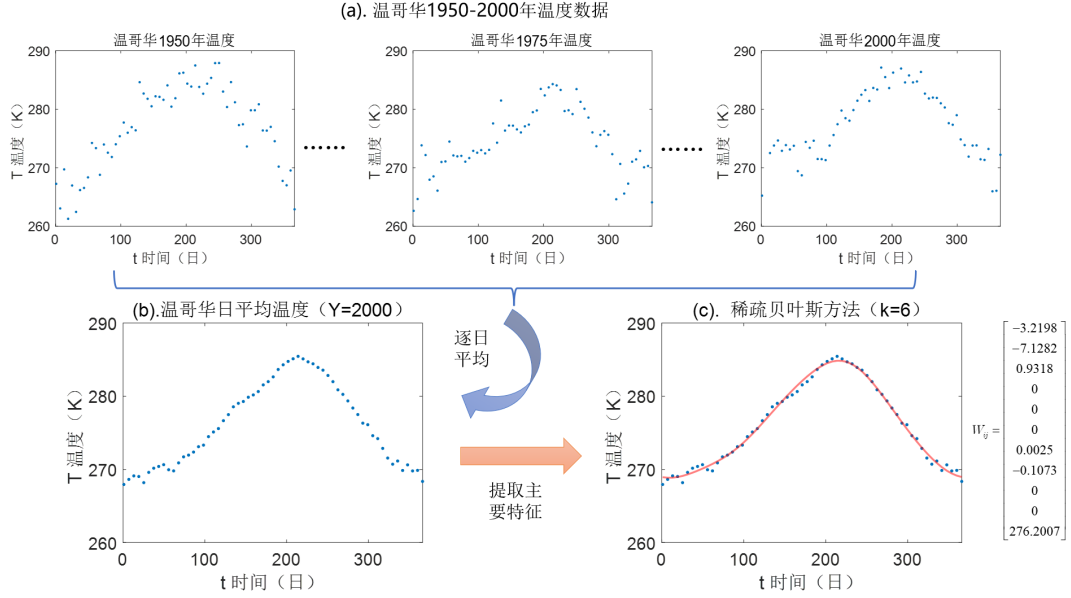


图5 周期辨识具体流程（以温哥华为例，辨识年份取 50 年）

3.2 趋势轴

对于时空数据来说，仅仅辨识出数据的周期是不够的。从上文中，可以看出，我们仅仅选取的是各地的平均温度进行周期性的辨识，并没有考虑数据在每个周期内的变化情况。因此本节将进一步进行时空数据趋势的辨识。

3.2.1 周期均值趋势辨识

由于在数据的趋势辨识中又不能受到周期辨识的影响，因此常用的方法是将每一个地区一个周期的数据的平均值聚合为一个数值，以此来判断数据趋势变化情况。以温度数据为例，定义年均温度 $\tilde{T}_{i,j}^y$ ，模仿(3.2)的定义方式可得：

$$\tilde{T}_{i,j}^y = \frac{1}{365} \sum_{d=1}^{365} T_{i,j}^{y,d}. \quad (3.3)$$

同时，定义年平均温度对应矢量 $\tilde{\mathbf{T}}_{ij}$ ，表示经纬度为 i, j 处年平均温度序列。一般来说，地区温度的年际变化有波动上升、波动下降和稳定波动三种特征，仅需尝试用 $\tilde{\mathbf{T}}_{ij}$ 进行稀疏辨识，辨识温度变化趋势即可。但由于年际温度波动较大，且波动并不符合正态分布，因此本文选取 LASSO 方法进行辨识，选取了表示趋势变化的幂函数基（包括凸函数 t, t^2, t^3, \dots 和凹函数 $t^{1/2}, t^{1/3}, t^{1/4}, \dots$ ）、对数基（ $\log(t)$ ）和常数基进行辨识。该辨识方法较为简单，可以辨识部分地区的年际变化。

3.2.2 周期分段趋势辨识

周期均值趋势辨识是一个很朴素的辨识趋势的方法。但可惜的是，这种方法并非总能准确的抓住数据的趋势的变化。因此，需要首先对周期内数据进行分段，之后再对每一段进行辨识。数据分段有很多方法，包括经验分段、聚合分段、自适应分段等。本文选取的数据是温度数据，

因为该数据集有明显的季节规律，故而采取经验分段，将气温数据分为春、夏、秋、冬四段，再分别进行辨识趋势变化。若不进行数据的分段，以温度数据为例，则有可能出现以下两个问题：

一方面，由于年际温度变化 \tilde{T}_{ij} 和日均温度变化 \bar{T}_{ij} 不同，尽管进行了年温度的平均， $\tilde{T}_{ij} - t$ 的图像仍然不具有平滑性，这加大了辨识难度。在一部分地区的辨识中，温度有上升或下降趋势但是由于被噪声掩盖并不明显，辨识结果仅有常数基，这就失去了趋势辨识的意义。

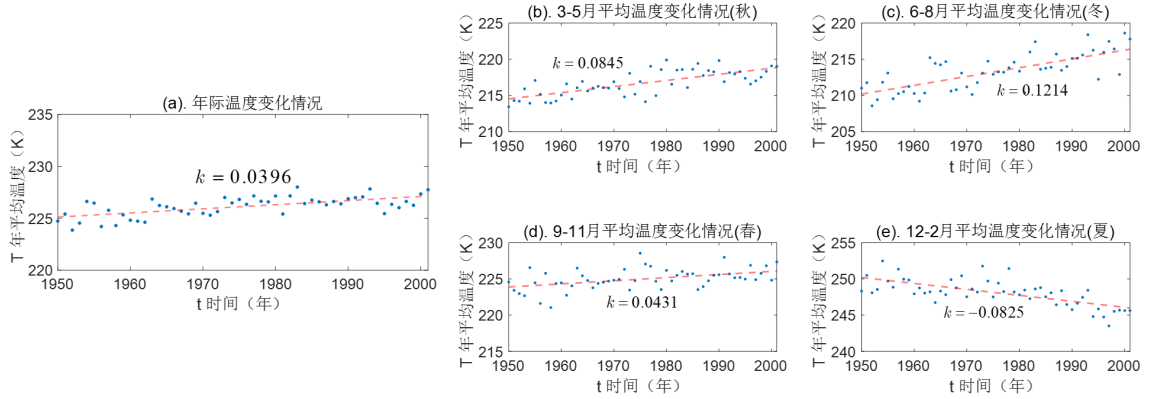


图 6 南极洲（约 79°S，56°E）不同时段温度变化趋势情况

另一方面，即便辨识出了地区温度变化情况，也不能较为准确的描述该地区温度变化趋势。为了更加直观地说明该问题，本文选取南极洲某地区（约 79°S，56°E）为例具体说明。如图 6 所示，除却年平均温度变化 \tilde{T}_{ij} （图 6.(a)），本文还做了该地区四季平均温度（即每年 3~5 月、6~8 月、9~11 月以及 12~2 月的平均温度分别为图 4.(b)-(e)）随时间的变化情况。为了更好的展示温度趋势，本文对变化趋势做了线性回归，斜率 k 即表示温度年际变化的大小。从温度的年际变化曲线中，我们仅可以看出 1950-2000 年的年均温度在缓慢增加。而从四季平均温度变化情况来看，却可以得出该地区春、秋、冬三个季节的增幅各不相同，且均大于年均温度增幅，冬季增幅最大，春季增幅最小，且可以具体量化春、秋、冬三个季节上升比率为 1:2:3。而夏季平均温度却呈现较快的下降趋势。相比于年均温度我们可以得到更多可解释的信息，例如：该地区温度升高主要是因为冬季的温度升高导致的，冬季收到的全球变暖影响较大；该地区夏季温度下降可能是因为“负温室效应”等等，这些通过年际温度变化并不能分析到。此外，在进行温度预测时，由于该地有三个月温度在降低，而辨识结果给出的是上升趋势，可能导致这三个月温度预测结果较差。

基于四季温度变化可解释性更强且预测效果的原因，本文对于温度数据定义四季趋势辨识，通过一个地区四季的温度变化来洞悉该地区的温度趋势变化情形。由于南北半球的春夏秋冬恰好相反，不方便定义，因此本文直接使用月份进行定义，我们令： $\tilde{T}_{ij}^{(1)}, \tilde{T}_{ij}^{(2)}, \tilde{T}_{ij}^{(3)}, \tilde{T}_{ij}^{(4)}$ 分别为该地区每年 3~5 月、6~8 月、9~11 月以及 12~2 月的平均温度。由于满足关系

$$\sum_{k=1}^4 \tilde{T}_{ij}^{(k)} / 4 = \tilde{T}_{ij}. \quad (3.4)$$

因此四季趋势辨识包含了年际温度辨识的内容。此时我们分别对 $\tilde{T}_{ij}^{(1)}, \tilde{T}_{ij}^{(2)}, \tilde{T}_{ij}^{(3)}, \tilde{T}_{ij}^{(4)}$ 使用 LASSO

方法，利用幂函数基、对数基和常数基进行辨识，则可以得到四季变化趋势的稀疏向量 $\tilde{\mathbf{W}}_{ij}^{(1)}, \tilde{\mathbf{W}}_{ij}^{(2)}, \tilde{\mathbf{W}}_{ij}^{(3)}, \tilde{\mathbf{W}}_{ij}^{(4)}$ 这就完成了周期分段趋势辨识。

3.3 突变轴

在成功辨识了数据的周期和趋势后，我们基本可以通过这两者的辨识结果量化出任意地区数据的趋势变化情况了。但是，我们可以明显的看出，本章提到的前两个轴所得到的辨识趋势结果是平滑的（因为我们选取的都是连续函数进行辨识）。但真实数据往往充满了不确定性，以温度数据为例，3~5 月温度并不会一直增长，而可能会出现突然的降温或升温（如图 4.(a)所示），这些数据被称为突变数据，是不能通过前两个轴所辨识出来的。本节旨在找到突变数据，并通过复杂网络的方法预测这些突变数据，以达到短期精确预测的效果。

3.3.1 突变数据

在定义突变数据前，本文首先引入残差数据的概念。前文已经介绍，可以利用周期轴和趋势轴构建一个较好的温度辨识模型（具体耦合过程见 3.4.1 节），获得一个平滑的时空预测结果，记为 $\hat{\mathbf{T}}_{ij}^t$ 。我们定义残差时空数据 δ_{ij}^t （其中上下标表示时间、空间位置）为：

$$\delta_{ij}^t = \hat{\mathbf{T}}_{ij}^t - \mathbf{T}_{ij}^t. \quad (3.5)$$

本文主要将该残差数据 δ 分为两个部分：其中一部分是因白噪声导致的数据误差（即温度的波动），另一部分是因为某些特殊原因，导致数据的突变（即自然灾害）。

在残差数据中，白噪声是不需要预测的，因为这是一种毫无规律的波动。本文旨在从残差数据 δ_{ij}^t 中找到突变数据，我们提出了一种自适应寻找 ζ_{ij}^t 的方法：

$$\zeta_{ij}^t = \begin{cases} \delta_{ij}^t, & |\delta_{ij}^t| > \left(|\delta_{ij}| \right)_p \\ 0, & |\delta_{ij}^t| \leq \left(|\delta_{ij}| \right)_p \end{cases}, \quad (3.6)$$

其中分位数 p 值是一个自适应选取的过程。 p 值会从 1 递减至 0 进行遍历选取，选取的标准是对于该地区在 p 分位数条件下，去掉突变数据的时间序列利用 Ljung-Box Q 检验是否以某置信数值为白噪声（该算法详细过程可见附录中附表 5）。

在确定各个地区的突变数据 ζ_{ij}^t 后，我们将利用各地的突变数据作为复杂网络的输入，并预测将来一段时间内可能出现的特殊突变数据。

3.3.2 网络搭建

本文对于效仿 transformer 中的 encoder-decoder 模块分别构建了编码器和解码器。其中编码器用于编码时空规律，利用多次 3D 卷积将时空数据最终压缩为 4 维：批次、特征、经度和纬度。解码器用于通过前馈神经网络对特征维进行解码，以获得任意长度的数据。

(1) 编码器

编码器旨在收集时空数据的特征。对于欧式空间的数据，常用 CNN 进行收集。对于非欧式空间的数据，常用 GNN 进行收集。由于本文气温数据是经纬度数据，因此应用 CNN 进行收集。但在这里的卷积与图像卷积(2.11)并不相同，因为(2.11)对图像来说是一种二维卷积，在时空数据中，我们首先将突变数据按照周期分为多个时空块 ζ^k ，对于每一个时空块使用 3D 卷积提取特征。这样划分时空块是因为需要保证在每一次 3D 卷积的过程中，周期这个条件作为先验，不被破坏，每个周期内的时间保证独立性。若假设卷积核为 H ，则每一次 3D 卷积与池化运算可以表示为：

$$\left(\zeta_H^{(k)}\right)_{ij}^t = \text{sum}\left(\zeta^{(k)} \otimes H\right), \quad (3.7)$$

$$\zeta_1^{(k)*} = \text{pool}\left(\zeta_H^{(k)}\right), \quad (3.8)$$

其中 $\zeta_H^{(k)}$ 为一次 3D 卷积后的时空特征； $\zeta_1^{(k)*}$ 为一次池化后的时空特征（本文中，池化均选取平均池化）。每做一次卷积和平均池化我们记为一个轮回（用 $\zeta_1^{(k)*}$ 中的角标）。在进行 3D 卷积和池化的过程中，我们会控制填充、步长与通道使得在每一次卷积的每一个通道中，时空维度保持不变，每进行两个轮回，我们会将原数据与卷积数据进行相加形成 ResNet 以防出现梯度消失，即

$$\zeta_i^{(k)*} \leftarrow \begin{cases} \zeta_i^{(k)*} + \zeta_{i-2}^{(k)*}, & i > 2, \\ \zeta_i^{(k)*} + \zeta_H^{(k)}, & i = 2. \end{cases} \quad (3.9)$$

在多次 3D 卷积、池化（即式(3.7)-(3.8)）以及残差相加操作后（即式(3.9)），最后两次 3D 卷积操作我们会压缩时间维度数据（空间维度仍然保持不变），使其转变为任意维度的特征维度数据，该特征数据将作为时空模型的隐变量。然后，再将卷积后得到的时空块在特征维度合并，即：

$$\zeta^* = \left\|_{\text{dim}=3}^k \zeta_i^{(k)*}, \quad (3.10)$$

其中，合并后的结果记为 ζ^* ，而合并后 ζ^* 的维度压缩为 4 维：批次、特征、经度和纬度，该结果就作为编码器的输出。

(2) 解码器

编码器输出的特征 ζ^* 要作为解码器的输入，我们对于每一空间点的数据，将 ζ^* 在空间内的每一个点在解码器中展平得到 ζ_{ij}^* ，并通过全连接神经网络进行解码以获得任意长度的数据 ζ_p ，即：

$$\zeta_p^{ij} = \text{fc}(\text{ReLU}(\text{fc}(\text{ReLU}(\text{fc}(\zeta_{ij}^*))))), \quad (3.11)$$

其中 $\text{fc}()$ 表示线性全连接层， $\text{ReLU}()$ 为非线性变换层。最后再将 ζ_p^{ij} 折叠合并，即：

$$\zeta_p = \left\|_{\text{dim}=1}^i \left\|_{\text{dim}=2}^j \zeta_p^{ij} \right\|. \quad (3.12)$$

得到网络最终的预测数据 ζ_p 。在本数据集中，我们选择输出一年的数据进行预测并输出， ζ_p 的维度恰为一个周期的长度。

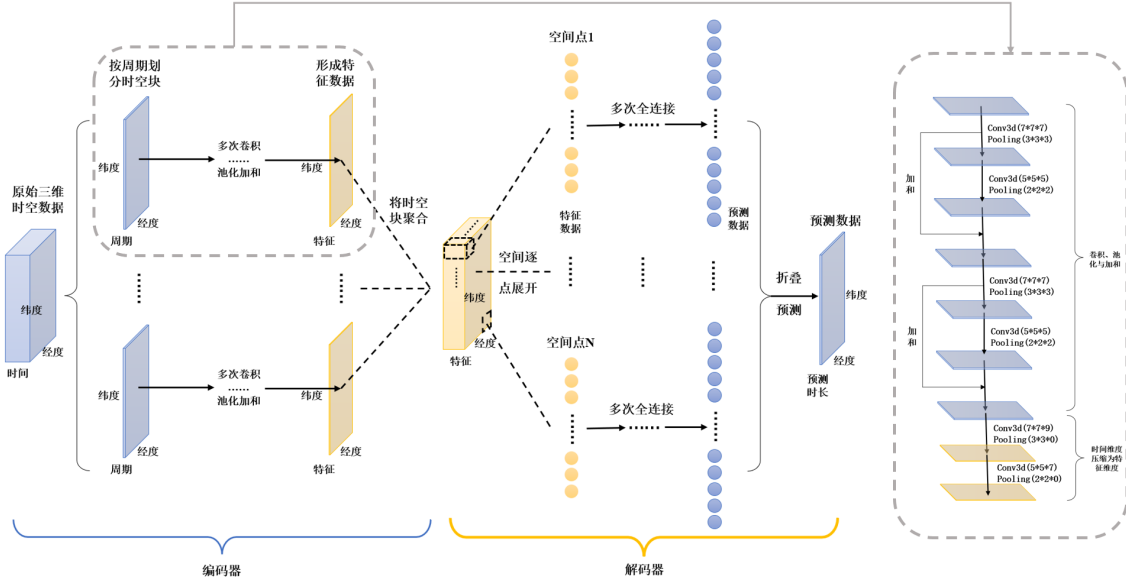


图 7 突变数据预测的 encoder-decoder 网络结构

综合编码器和解码器的关系，我们的总体网络结构搭建如图 7 所示。当我们将气温的突变数据 ζ 放入网络后，可以最终得到预测的突变温度 ζ_p 。该突变温度可用于和周期轴、趋势轴进一步耦合，我们将在下一节详细说明。

3.4 耦合关系与算法流程

在确定了时空辨识模型的三个轴后，我们需要介绍这三个轴之间的耦合关系以用于辨识和预测。在全文的开始我们提到，我们需要时空数据的长期趋势和短期预测。以气温数据为例，本文的辨识与预测主要分为气温的长期趋势辨识和气温的短期精确预测。气温的长期趋势辨识仅需要用到周期轴和趋势轴两个轴，而气温的短期精确预测则需要用到周期轴、趋势轴和突变轴，本节将分别详细讲述辨识与预测的耦合关系，并在本节的最后给出算法流程图。

3.4.1 时空数据的长期辨识耦合关系

时空数据的长期趋势辨识需要用到周期轴和趋势轴，以气温数据为例，气温的长期趋势辨识将气温的周期性变化和四季变化趋势两种辨识叠加，并在每一年随着四季切换，因此我们也称该耦合关系为随四季切换的辨识算法。为了便于叙述，记经纬度为 i, j 处辨识的周期函数为 $f(d), 1 \leq d \leq 365$ ，而辨识的四季变化趋势函数为 $g_k(y), y \geq 1950, k = 1, 2, 3, 4$ ，其中 k 代表季节。

为了使周期性和四季变化趋势得出的模型叠加，我们首先需要对于四季变化趋势函数 $g_k(y)$ 进行平移，使其在训练集中的积分为 0（本文取值为 1950 到 2000 年）。这是因为周期性的温度趋势函数 $f(d)$ 是一个平均温度序列，趋势变化函数与其叠加相当于对应每一年进行的温度根据该年春夏秋冬辨识的结果进行一个微调，以符合该年的温度特征。而由于温度周期性辨识是前 51 年的温度平均，因此，在叠加四季变化趋势时也同样需要前 51 年均值为 0（即 $g_k(y)$ 平移后在 1950 到 2000 年的积分为 0）。故而 $g_k(y)$ 的平移量 a 应该满足：

$$\int_{1950}^{2000} (g_k(y) - a) dy = 0 \Rightarrow a = -\frac{1}{51} \int_{1950}^{2000} g_k(y) dy, \quad (3.13)$$

即可实现辨识的叠加。

同时，为了保证该算法能进行四季温度的切换，我们定义函数日期——季节转换函数

$$\varphi(d) = \begin{cases} 1, d \in (66, 150] \\ 2, d \in (150, 240] \\ 3, d \in (240, 330] \\ 4, d \in (330, 365] \cup [1, 65]. \end{cases} \quad (3.14)$$

该公式使得季节标识 k 与日期 d 相联系。算法即可直接根据日期进行不同季节函数的切换，以实现辨识的四季切换。

综上，我们发现，在辨识的四季切换算法中，叠加与切换两个环节仅需要知道预测的年份与日期，预测温度的公式可以表示为：

$$F(y, d) = f(d) + g_{\varphi(d)}(y) - \frac{1}{51} \int_{1950}^{2000} g_{\varphi(d)}(y) dy, \quad (3.15)$$

其中 $F(y, d)$ 表示在第 y 年的第 d 预测得到的温度值。由于我们的数据均为离散形式，因此，将预测温度离散化为 $\hat{T}_{i,j}^{y,d}$ 后，(2.15)也可以写成如下离散形式：

$$\hat{T}_{i,j}^{y,d} = f(d) + g_{\varphi(d)}(y) - \frac{1}{51} \sum_{y=1950}^{2000} g_{\varphi(d)}(y). \quad (3.16)$$

这就完成了温度辨识的叠加与切换。我们将随四季切换的辨识算法写成流程图的形式放在附录的**附表 6**，可供读者查阅。

3.4.2 时空数据的短期预测耦合关系

在得到时空数据的长期辨识后，我们可以通过辨识的结果，计算时空中存在的突变数据，再通过已知的突变数据利用我们构建的时空神经网络预测将来一段时间的突变数据。我们将预测得到的突变数据与周期轴、趋势轴辨识的数据相结合，以得到更加精确的短期预测结果，鉴于该算法具有对突变数据的预测，我们称该算法为突变 CNN 算法（**abrupt-CNN**）。

以全球气温数据为例，在得到温度的长期趋势辨识后，我们首先需要通过辨识结果的输出 $\hat{T}_{i,j}^{y,d}$ 作为(3.5)-(3.6)的输入，分别计算残差时空数据 δ_{ij}^t 和突变时空数据 ζ_{ij}^t 。其次，我们将突变时空数据 ζ 放入构建好的网络，利用 3D 卷积时空神经网络可以预测出在即将来临的一段时间内的突变气温数据 ζ_p 。然后，我们可以需要叠加突变气温数据 ζ_p 和 3.4.1 节中对应时间的辨识温度数据 \hat{T} 计算出精确预测的温度 \hat{T}_p ，即

$$\hat{T}_p = \hat{T} + \text{filter}(\zeta_p), \quad (3.17)$$

其中 $\text{filter}(\zeta_p)$ 对预测结果进行了一次过滤，将 **abrupt-CNN** 预测得到的结果最大 15% 的数值和最小 15% 的数值滤出作为突变数值，并以之前数据突变的平均值赋值，其余位置均过滤为零。

加入突变数据的 \hat{T}_p 将具有对突变数据的预测能力，这使得预测曲线不再平滑，将在短时间内的预测将会比 \hat{T} 拥有更好的预测效果。

3.4.3 时空数据辨识预测一体化算法流程可视化

在本节的最后，我们给出这三个轴的耦合关系图以及两个算法流程图（见图 8），以便读者更好的理解各个轴之间与两个算法关系。

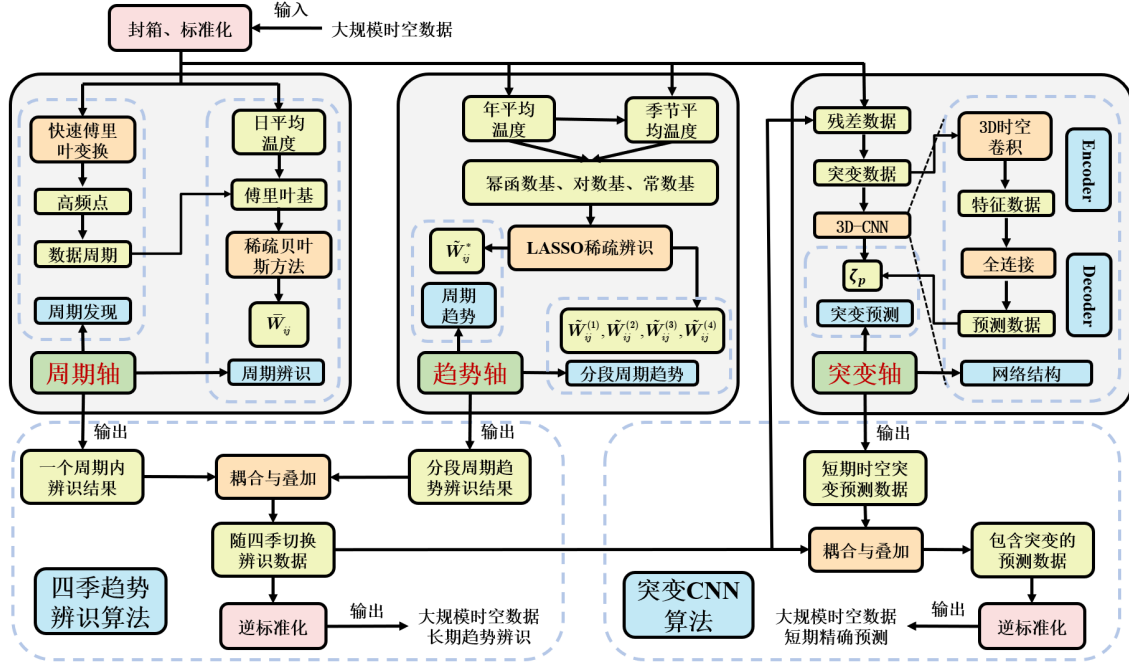


图 8 时空数据辨识预测一体化框架流程示意图

从上图我们可以清楚的看到，时空数据辨识预测一体化算法中的每一个轴都自成体系但又相互关联。该算法深入挖掘大规模时空数据中的周期、趋势与突变情况并将它们有机的结合了起来，并能让同时进行数据的辨识和预测。在下一章，我们将详细叙述该算法在全球气温这个数据集上的辨识、预测表现。

3.5 本章小结

本章是全文的核心章节，在本章，我们提出了一个时空数据辨识预测一体化框架，我们分别介绍了该框架的周期轴、趋势轴、突变轴的计算方法，输入以及输出，同时，详细阐述了它们在大规模时空数据辨识中的优点所在。本章基于这三个轴的输出结果分别提出了四季趋势辨识算法和突变 CNN 算法进行时空数据的长期趋势辨识和短期数据的精确预测，该框架将辨识方法与复杂网络有机结合，具有一定的创新性。在下一章，我们将利用该框架，对全球气温这个现实数据进行辨识与预测。

4. 基于时空辨识预测一体化模型的全球温度预测结果分析

本章将首先引入一个大规模时空数据集——全球气温数据，我们将对该数据集进行基本介绍与预处理，然后将该数据集进行划分以用于模型。然后，我们将处理好的数据作为输入，放入上一章搭建好的时空数据一体化预测辨识框架中，我们将辨识、预测该数据集，并对比其他辨识、预测方法，给出各种方法的精度。同时，我们也会将结果可视化（限于篇幅，部分放于附录）。最后，我们会根据预测结果给出合理的建议。

4.1 数据及其预处理

考虑到大规模时空数据量很大，并不能直接用于模型，我们首先要对该数据集进行解释、预处理和划分，才能放进模型训练。本节将对数据处理做详细的介绍。

4.1.1 数据简介

本文的原始数据格式为.nc 格式，这一般是装载气象数据的格式。这是一个全球气温的三维时空数据，三维分别是时间、经度、纬度，这三维确定的数值为温度，时间、经度、纬度、温度的具体信息如表 1 所示：

表 1 数据具体情况表

特征	时间	经度	纬度	温度
单位	年 (y)	度 (°)	度 (°)	华氏度 (K)
维度	103994	192	96	无
粒度	6h	1.875°	1.875°	无
上限	1950	-180	-90	311.86
下限	2021	180	90	190.48

从上表可知，这是一个大规模的时空数据，总数据量近 20 亿，为 GB 级别数据。若直接操作难度较大。此外，温度数据并非归一化数据，并不能直接适用于神经网络、辨识等模型中，因此我们需要进行一定的数据预处理操作。

4.1.2 数据预处理

数据预处理主要包括对于数据经度、维度和时间三个维度进行封箱操作，以及对于温度进行归一化操作。

封箱操作的目的是缩减数据量，使得神经网络训练参数不至于过多。本文采取平均值封箱操作，这与平均池化的操作基本相同。在时间上，由于该数据集预测时间非常长，不必达到一天四次采样，采取 6 天一次采样即可基本预测变化趋势。在空间上，仅需要对国家地区做到基本的了解，1.875° 的网格精度过于细密，我们选取 7.5° 的网格精度进行预测即可。因此，我们对时空数据中采取 24×4×4（时间×经度×纬度）的封箱操作。

标准化操作是将数据格式统一称为均值为零，方差为 1 的数据，防止在使用神经网络时产生梯度爆炸或是梯度消失。标准化操作为：

$$T_{\text{norm}} = \frac{T - \bar{T}}{\sqrt{\sum (T - \bar{T}) / (N - 1)}}, \quad (4.1)$$

其中， \bar{T} 为 T 的均值， N 为对 T 封箱操作后的数据量。若不加特殊说明，本文应用于模型的数据均为封箱、归一化的 T_{norm} ，而非原始数据 T 。

4.1.3 数据集的划分

(1) 温度的长期趋势辨识

辨识方法可以确定未来很长一段时间趋势，因此我们可以仿照及其学习的方法，按照“73 划分”训练集 \mathcal{P} 和测试集 \mathcal{T} 我们共有 72 年的数据，因此 50.4 年作为测训练集，21.6 年作为测试集。但是由于预测方法涉及周期性，如果割裂了周期关系，在预测中会引起不必要的麻烦，因此本文的取训练集为 51 年，测试集为 21 年，即：

$$\mathcal{P} = \{T_{i,j}^y \mid 1950 \leq y \leq 2000\}, \mathcal{T} = \{T_{i,j}^y \mid 2001 \leq y \leq 2021\}, \quad (4.2)$$

其中 $T_{i,j}^y$ 表示在第 y ，经纬度为 i, j 处的温度。

(2) 温度的短期精确预测

温度的短期精确预测主要是通过神经网络进行突变数据的预测，将突变数据叠加于长期辨识得到的结果。在这种情况下，我们的测试集仅取一年的数据（即 2001 年），而训练集与长期趋势辨识相同，即：

$$\mathcal{P} = \{T_{i,j}^y \mid 1950 \leq y \leq 2000\}, \mathcal{T}^* = \{T_{i,j}^y \mid y=2001\}, \quad (4.3)$$

其中 \mathcal{T}^* 表示短期精确的测试集。

此外，值得注意的是，在进行神经网络训练时，我们需要提前建立好数据的输入输出批次，以便利用误差函数进行误差反向传播。我们的输入一般取 5 年的时空突变数据，而输出取 1 年的时空突变数据进行训练，可以采取以 10 个样本作为 mini-bench 进行训练，这样训练收敛速度会有较大的提升，训练的 Loss 可见附录中附图 4。

4.2 预测结果分析

常用的评判预测结果的方法为均方根误差 (RMSE)、均方误差 (MAE) 和拟合优度 (R^2)，它们在本文中的计算公式分别是：

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum (\hat{T}(\hat{T}_p) - T)^2}, \quad (4.4)$$

$$\text{MAE} = \frac{1}{N} \sum |\hat{T}(\hat{T}_p) - T|, \quad (4.5)$$

$$R^2 = 1 - \sqrt{\sum (\hat{T}(\hat{T}_p) - T)^2} / \sqrt{\sum (T - \bar{T})^2}. \quad (4.6)$$

其中 $\hat{T}(\hat{T}_p)$ 表示温度辨识或预测的数据。这里 RMSE 和 MAE 是描述预测值和真实值接近程度的一个量，一般来说，当其数值越小时，预测结果与真实数据越接近。而 R^2 主要计算的是预测结果与真实数据的相关系数，一般来说，当其数值越大时，预测结果对真实数据的表达能力越好。本节将利用这三个评判标准对温度的辨识与预测进行分析。

4.2.1 长期温度趋势辨识结果分析

下面我们将进行长期温度趋势辨识的结果分析，我们将已处理好的数据放入四季切换辨识模型，计算了三个评判指标，并对比了一些常见的辨识算法，结果如表 2 所示：

表 2 不同辨识算法的效果对比

辨识算法	RMSE	MAE	R^2
最小二乘法	3.3034	2.1060	0.8450
阈值最小二乘方法	3.1080	1.9678	0.8542
LASSO	3.1252	2.0565	0.8534
稀疏贝叶斯方法	3.1079	1.8921	0.8542
四季切换辨识算法	2.8798	1.7408	0.8649

从上述结果中，我们可以发现利用虽然不同的方法进行时空数据的预测结果迥异，但是 R^2 均较大，这说明，辨识模型可以较好的完成模型的辨识。从 RMSE 和 MAE 两个评判指标来看，效果最差的是直接使用最小二乘进行预测，最好的是本文提出分四季切换辨识算法，而直接利用三种不同的稀疏辨识算法效果较为接近。最小二乘辨识精度低是因为这种算法并不能有效的提取特征，虽然在训练集中表现良好，但是到了测试集就因为对测试的过拟合不能较好的完成辨识；而几种稀疏辨识算法虽然在一定程度上可以提取特征，但由于数据规模较大，时间跨度较长，对于趋势特征并不能较为完整的提取。四季切换辨识算法相比于其他稀疏辨识算法，分别深度挖掘了大规模时空数据的周期性和趋势性，具有较好的辨识效果，我们用该方法得到的辨识结果进行输出（由于该辨识图像过长，我们将这张辨识图像放于附录，见附录中附图 1）。

此外，该算法的贡献并不仅仅局限于精度，更多的是因为它有效的提取了各地区的特征，该算法输出的稀疏辨识系数给该地区温度变化提供了较大的可解释性。我们可以从周期轴输出的稀疏特征向量 $\bar{\mathbf{W}}_{ij}$ 洞悉、量化温度的周期变化规律；也可以从趋势轴输出的四季变化的稀疏向量 $\tilde{\mathbf{W}}_{ij}^{(1)}, \tilde{\mathbf{W}}_{ij}^{(2)}, \tilde{\mathbf{W}}_{ij}^{(3)}, \tilde{\mathbf{W}}_{ij}^{(4)}$ 中看出该地区的年际变化情况的缓急。例如，南极洲地区从周期轴提取的稀疏特征向量 $\bar{\mathbf{W}}_{ij}$ 十分密集，且前两项数值较大，这就说明了这两个地区冬夏温度变化很大，且具有复杂的周期规律，此外该地区趋势轴的 $\tilde{\mathbf{W}}_{ij}^{(1)}, \tilde{\mathbf{W}}_{ij}^{(3)}, \tilde{\mathbf{W}}_{ij}^{(4)}$ 系数为正，而 $\tilde{\mathbf{W}}_{ij}^{(2)}$ 系数为负，这说明南极洲在 6~8 月气温有下降的趋势，而其他月份有上升的趋势，进一步，还能看出气温上升速度快去下降速度，进而得出南极洲气温在上升的结论；而赤道地区，相比于南极

洲地区从周期轴提取的稀疏特征向量 $\bar{\mathbf{W}}_{ij}$ 十分稀疏，以吉隆坡周围地区为例，该地区 $\bar{\mathbf{W}}_{ij}$ 仅有两项有数值，这说明该地区年际温度变化较为舒缓，此外该地区趋势轴的 $\tilde{\mathbf{W}}_{ij}^{(1)}, \tilde{\mathbf{W}}_{ij}^{(2)}, \tilde{\mathbf{W}}_{ij}^{(3)}, \tilde{\mathbf{W}}_{ij}^{(4)}$ 系数基本接近于 0，因此我们可以分析得出赤道地区的近 50 年的温度变化也较小。如果我们深入分析每个地区的辨识系数，我们甚至能分析出更多关系，例如通过系数比例关系得到各个基函数的重要性，以此来判断该地区温度变化的主要原因是什么。这些分析，正体现了系统辨识的可解释性，是从复杂网络中学得的参数中不能得到的，也是系统辨识的优势所在。

4.2.2 短期温度精确预测分析

下面我们将进行短期温度趋势精确预测的结果分析，我们将对比突变 CNN 与其他一些预测算法，结果如表 3 所示

表 3 不同辨识算法的效果对比

预测算法	RMSE	MAE	R ²
ARIMA	14.4033	11.5576	-0.1124
BP	4.7664	3.2666	0.7824
3D-CNN	3.3917	2.3740	0.8433
ResNet	3.2453	2.3335	0.8500
LSTM	2.8029	1.8200	0.8712
GRU	2.7972	1.8112	0.8723
Abrupt-CNN	2.6812	1.5823	0.8812

从表中我们可以看出，传统的时间序列预测算法（例如 ARIMA）根本无法进行一年数据的预测，其 RMSE 与 MAE 均非常大，R² 甚至低于了 0，这是因为 ARIMA 的预测结果是单调的（见附录中附图 2）。这也体现了传统方法对于具有噪声的长周期问题难以进行预测。

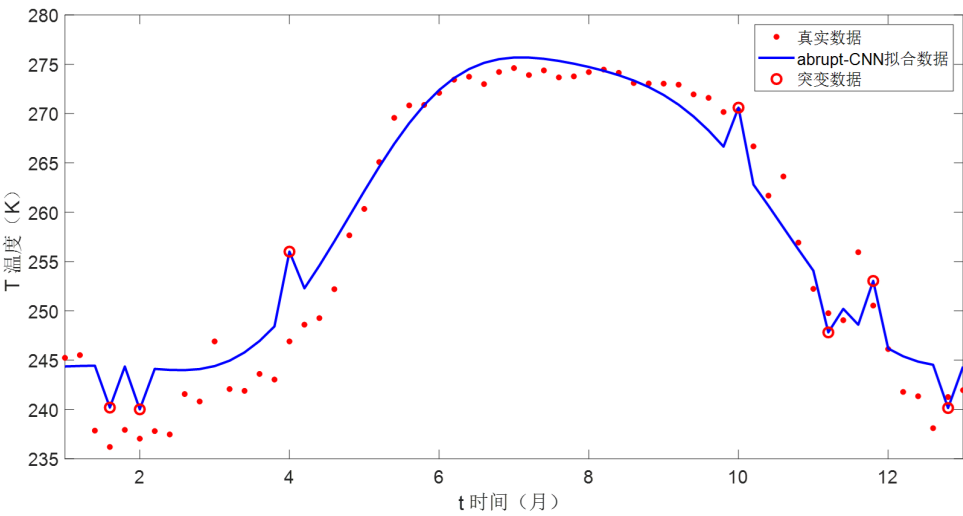


图 9 利用 Abrupt-CNN 对郑州周围 2001 天气预测结果

而对于传统的复杂网络方法（例如 BP, CNN, ResNet），由于这些复杂网络并非针对时序问题，因此预测效果也较为一般，甚至不如辨识结果。而对于循环神经网络 LSTM, GRU 来说，

可以较好的提取各个地区的时序特征，其预测结果已经优于辨识结果，但是由于这两个方法是进行空间内的时序的单点预测，因此难以进行空间关系的耦合与联系（这些网络的预测结果见附录中附图 3 所示）。本文提出的 abrupt-CNN 可以在辨识的基础上，有效预测数据中的突变点，其结果也好于目前大多数复杂网络给出的预测结果。我们将预测结果可视化如图 9 所示。

从图 9 中我们可以更加明确地看出突变 CNN 在辨识的基础上更加明确地发现了突变温度，并进行了精确的预测。本文认为，之所以复杂网络方法在短期精确预测方面达不到更高的精度，一方面，这些预测方法并不能有效抓住时空数据的突变点，导致在突变点处预测与现实数据差异很大。另一方面，这些方法大多不能抓住空间信息，更多的是进行单点预测，而不能进行空间的联系，这导致了预测精度降低。因此 Abrupt-CNN 在大规模数据的短期精确预测中更占优。

4.3 本章小结

本章将全球气温数据放入上一章提出的大规模数据辨识与预测一体化框架中，获得了较好的结果，该框架获得的结果比目前大多数辨识方法、预测网络效果都好。基于此，本文将已知所有数据输入（即将 1950~2021 年全部数据放入模型训练），并预测未来一段时间的温度变化情况，并对全球各地提出合理的意见。

在长期趋势辨识方面，从温度的长期趋势辨识中我们从预测的结果中发现，全球近 70 年来大多数地区都有变暖的趋势，但有些地区尤为严重。在北半球中，北极圈地区、美洲北部和欧洲西部全球变暖较为严重。在四季切换辨识中趋势项的三次函数项系数较大，这说明了这些地方的温度的增速较快，可能持续变暖的倾向较为严重。尤其是欧洲西部的沿海地区荷兰，该地区如果按照目前的趋势持续温度上涨，可能会导致海平面上升，根据目前该地区海平面情况，该地区的部分城市被淹没，需要尽快制定可持续化发展的路径。而在南半球中，南极圈附近的全球变暖现象也尤为严重（尤其是近 20 年来尤为严重）。南极是全球最冷的地方，如果不能保证较低的温度，南极的大多数冰川将会悉数融化，从而导致周围海平面大幅上升，海岛淹没、自然灾害频发将会是今后的常态。因此，我们需要对该地进一步的实施措施保护，防止南极再变暖。

在短期趋势辨识方面，我们预测了全球 2021 全年可能发生突变的天气。我们选择江苏附近地区进行具体分析。在江苏周围的地区，通过 abrupt-CNN 预测到了在春季会有一次较为明显的降温，这和目前已知的天气相符。此外，我们预测了到在江苏的早秋和晚秋将分别有一次较高的升温和降温，或许需要对此进行提前防范。

在下一章，我们将对全文总结，针对本文提出的算法总结优缺点，并对大规模数据预测进行未来发展方向的展望。

5. 展望与总结

5.1 算法的优缺点总结

本文提出了一个对于大规模时空数据预测辨识一体化的框架，其中包括两个算法：大规模时空数据的长期趋势辨识和短期精确预测，我们将这两个算法应用于全球气温数据，得到了较好的辨识与预测结果。在本节我们将分别阐述这两个算法的优缺点。

5.1.1 长期趋势辨识的优劣势分析

(1) 优势

- 该方法相比于传统的稀疏辨识算法解释性更强。由于该辨识算法可以对于周期性和趋势性分别进行辨识（且在辨识的时候是两个完全独立的过程互不干扰），再最终叠加，通过两个辨识过程比传统方法更具可解释性；
- 该方法相比于传统的稀疏辨识算法精度更高。该算法经过了几次辨识的切换和叠加，深层次地剖析了数据规律，因此，相比于直接进行稀疏辨识，精度更高。
- 该算法适用于大规模数据，该算法在进行周期辨识和趋势辨识时，都将大规模数据转化为小规模数据直接进行辨识。该算法即使对于更大规模的数据，也一样适用。

(2) 劣势

- 四季切换辨识算法不能对突变温度进行预测。在附录可视化的结果中，我们可以看出，预测曲线较为平滑，对于极端值难以进行预测。正因为如此，该方法存在误差下限。一般来说，我们测试集的误差不可能比训练集还低，而该算法测试集误差和训练集误差已经十分接近（RMSE 仅差 0.15），这意味着该算法已经基本达到了稀疏辨识预测的极限。
- 运算时间较慢。由于该算法本质上是对于多个传统稀疏辨识算法的叠加与切换，因此，运算时间会慢于直接使用辨识方法。

5.1.2 短期精确预测的优劣势分析

(1) 优势

- 该方法可以有效的预测数据的突变情况，做到系统辨识难以做到的预测，相比于传统方法具有更高的精度。
- 该方法在确保精度的情况下，保留了长期趋势辨识得到的周期和趋势情况，这表明该算法仍然具有一定的可解释性。此外，该算法通过突变轴得到的突变数据，相比于传统方法直接预测时空数据，也具有可解释性。
- 在网络结构的搭建时，也将数据分为时空块分别进行预测，防止了参数数量过多导致训练速度慢的问题，适用于大规模数据的预测。

(2) 劣势

- 该算法中的突变点也可能会预测错误，而突变点的预测错误将会导致更大的误差。
- 网络训练时间较长，且同样避不开对于神经网络的频繁调参问题（例如 3D 卷积核大小，训练批次数量等）。

5.2 展望与未来

大规模时空数据的辨识与预测在近几十年一直是一个热点话题，如何将时间数据和空间数据有机结合并精确预测一直是该问题的瓶颈所在。本文也仅仅是给出了这方面问题的一种求解框架，有待修正和改进。我认为，未来在这方面可以进一步突破的有两点，其一是进一步加强空间数据之间的联系。我们目前大多数算法（无论是辨识还是复杂网络）都仅仅是单点预测，难以做到空间内各点的协同关系。即便是建立了周围数据之间的关系（例如 CNN），学习到的系数不具有可解释性。因此加强空间数据的连接，并赋予可解释性是必要的。其二是进一步加强特征提取，从本文中我们发现，当我们从大规模时空数据中提取日平均温度、年平均温度、突变数据后，数据规模会变小，辨识和预测会变得更加简单，如果我们能更多的提取、整合这样的数据进行预测，相信预测的精度将会进一步提高。

参考文献：

- [1] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J], Communication of the ACM, 2017, 60(6): 84-90.
- [2] Zhang W, Liu H L, Li P F, Han L. A Multi-task Two-stream Spatiotemporal Convolutional Neural Network for Convective Storm Nowcasting[C], 2020 IEEE International Conference on Big Data, 2020: 3953-3960.
- [3] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge[J]. Nature, 2017, 550(19): 354-359.
- [4] Zhang K, Zuo W, Chen Y, et al. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising[J]. IEEE Transactions on Image Processing, 2016, 26(7):3142-3155.
- [5] Hassan A, Mahmood A. Efficient Deep Learning Model for Text Classification Based on Recurrent and Convolutional Layers[C]. 16th IEEE International Conference On Machine Learning And Applications, 2017: 1108-1113.
- [6] Tara N, Brian K, George S, et al. Deep Convolutional Neural Networks for Large-scale Speech Tasks[J]. Neural Networks, 2015, 64(5): 39-48.
- [7] Brunton S L, Proctor J L, Kutz J N. Discovering governing equations from data: Sparse identification of nonlinear dynamical systems[J]. Proceedings of the National Academy of Sciences of the United States of America, 2015, 113(15): 3932-2938.
- [8] Kopsinis Y, Slavakis K, Theodoridis S. Online Sparse System Identification and Signal Reconstruction Using Projections Onto Weighted Balls[J]. IEEE Transactions on Signal Processing, 2011, 59(3):936-952.
- [9] Root T L, Price J T, Hall K R, et al. Fingerprints of global warming on wild animals and plants[J]. Nature, 2003, 421(6918): 57-60.
- [10] Meng J, Fan J F, Ludescher J, et al. Complexity-based approach for El Nino magnitude forecasting before the spring predictability barrier[J]. Proceedings of the National Academy of Science, 2020, 117(1): 177-183.
- [11] Restrepo P J, Eagleson P S. Identification of Independent Rainstorms[J]. Journal of Hydrology, 1982, 55(4): 303-319.
- [12] Galton F. The geometric mean, in vital and social statistics[J]. Proceedings of the Royal Society, 1897, 40(1): 263-289.
- [13] Hocking R R. A Biometrics Invited Paper: The Analysis and Selection of Variables in Linear Regression[J]. Biometrics, 1976, 32(1): 1-49.
- [14] Pati Y C, Rezaifar R, Krishnaprasad P S, et al. Orthogonal Matching Pursuit-Recursive Function Approximation with Applications to wavelet decomposition[C], Conference on Signals, Systems & Computers, 1993: 40-44.
- [15] Tibshirani R. Regression Shrinkage and Selection Via the Lasso[J]. Journal Of The Royal Statistical Society: Series B (Methodological), 1996, 58(1): 267-288.
- [16] Tipping M E. Sparse Bayesian Learning and the Relevance Vector Machine[J]. Journal of Machine Learning Research, 2001, 1(3): 211-244.
- [17] Liu B, Fan H, Lu Z, et al. The Annealing Sparse Bayesian Learning Algorithm[J]. Computer Science, 2012, 1209(1033): 97-106.
- [18] Brunton S, et al. Discovering governing equations from data by sparse identification of nonlinear dynamics. Proceedings of the National Academy of Science, 2016, 113(15):3932-3937.
- [19] Yuan Y, Tang X, Zhou W, et al. Data driven discovery of cyber physical systems[J]. Nature Communications, 2019, 10(1): 2-9.
- [20] Silva B, Champion K, Quade M, et al. PySINDy: A Python package for the Sparse Identification of Nonlinear

- Dynamics from Data[J]. The Journal of Open Source Software, 2020, 5(49): 2104-2105.
- [21] Stewart J. Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements[J]. Journal of Molecular Modeling, 2007, 13(12): 1173-1213.
 - [22] Guyon I, Weston J, Barnhill S, et al. Gene Selection for Cancer Classification using Support Vector Machines[J]. Machine Learning, 2002, 46(3): 389-422.
 - [23] Jing W, Jian T, Xu Z, et al. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach[C] IEEE INFOCOM 2017 - IEEE Conference on Computer Communications. IEEE, 2017: 1-9.
 - [24] Jin G, Sha H, Feng Y, et al. GSEN: An Ensemble Deep Learning Benchmark Model for Urban Hotspots Spatiotemporal Prediction[J]. Neurocomputing, 2021, 455(4): 1-28.
 - [25] Elsaadani M, Habib E, Abdelhameed A M, et al. Assessment of a Spatiotemporal Deep Learning Approach for Soil Moisture Prediction and Filling the Gaps in Between Soil Moisture Observations[J]. Frontiers in Artificial Intelligence, 2021, 4(63): 34-62.
 - [26] Kaiser E, Kutz J N, Brunton S L. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit[J]. Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences, 2017, 474(2219): 1-25.
 - [27] Beltagy L, Matthew E P, and Arman C. Longformer: The long-document transformer[J]. Computation and Language, 2020, 4(5150): 1-17.
 - [28] Zhang L, Li D, Guo Q. Deep Learning From Spatio-Temporal Data Using Orthogonal Regularizaion Residual CNN for Air Prediction[J]. IEEE Access, 2020, 8(99): 66037-66047.

致谢

行文至此，感慨万千，我的本科生活似乎随着我论文最后一个字的落笔而彻底结束了。岁月悠悠，不觉之间，我在南信大已经度过了四个春秋。回首望去，我的本科生活依旧历历在目，或许本科生活是我人生中最最为高光的时刻：在这里，我拿到过国家奖学金，拿到过江苏省三好学生，拿到过全院第一，也拿到过竞赛省一，这是我在步入大学前连想都不敢想的。能取得如此成就，离不开周围人的支持与帮助，谨以此篇对本科阶段帮助过我的人表示感谢。

首先，我想感谢本科阶段帮助过我的老师，你们的辛勤指导与教诲让我如沐春风，如饮甘霖。在这里，尤其想要感谢：指导我完成毕业论文的李顺杰老师、虞文武老师、陈都鑫老师，正是你们的辛勤指导，让我收获良多；我的学术启蒙老师咎立博老师；指导我完成大创，并教我写论文的王廷春老师；指导我学习数模的曹春正老师；常常找我聊天、探讨人生的董宝华老师以及在生活上常常给予我帮助的王文婧辅导员、邵福生辅导员、范萍辅导员。

其次，我想感谢我的父母，是你们默默地在我的背后给了我无限的支持。我不会忘记，在我迷茫的时期，父母陪我畅谈未来，指点人生方向；我不会忘记，在我的保研时期，父母与我并肩作战，共同选择心仪院校；我不会忘记，寒暑假在家时期，父母担心我的营养，变着花样给我做过的饭菜；我更不会忘记，在任何时期，父母都告诉我，家是你永远的港湾……我取得的成就离不开你们的支持与帮助，不管是精神上的还是物质上的。

最后，我还想感谢我身边的朋友，当我遇到困难，深陷低谷之时，我的身边总会出现你们的影子。我想感谢我的数学挚友瞿国亚同学，彻夜畅聊数学题的感受总是让我难以忘怀；我想感谢我的同桌许卓慧同学，在大一时，我们作为晚自习的同桌，你帮助了我很多很多，也给了我无尽的欢乐；我想感谢我的论文合作作者金世龙同学和许巧俏同学，我们合作完成的论文是我人生中发表的第一篇英文论文，这也使我决心走上了学术这条道路；我想感谢王博慧同学，你两年的陪伴占据了我本科时间的半壁江山，风雨同行，一路有你，欢笑与共，心存感激。我还想感谢我的四位舍友：杨杰龙同学、仲信羽同学、李浩然同学和李加因同学，同窗四年的互帮互助让我对 214 这个宿舍留下了最美好的回忆；同时，我也想感谢一下我未来的师兄师姐，是你们在我还没有入校之时就给予了我大量的学术指导，为我未来的路指引了方向。

因风道感谢，情至笔载援，你们于我之相助，我将永远铭记于心。前路漫漫，征程未完，我将带着一颗感激的新继续上路，愿我在写硕/博士毕业论文致谢时，亦，有你有我！

附录

A. 一些算法的流程图

本文涉及到的算法内容较多,为节省篇幅但也让读者明晰算法内核,本文将一部分算法流程放于附录部分,可供参阅。

1. 匹配追踪算法的流程

附表 1 匹配追踪算法流程

输入: $\mathbf{t}, \mathbf{T}_{ij}$ 以及允许误差 ε

1. 给定集合 Ψ , 初始化 $\Psi = \{\varphi_1(\mathbf{t}), \varphi_2(\mathbf{t}), \dots, \varphi_P(\mathbf{t})\}$ 。
 2. 给定残差 $\mathbf{RT}_{ij} \in \mathbb{R}^M$, 并满足 $\|\mathbf{RT}_{ij}\| = \inf$ (例如将 \mathbf{RT}_{ij} 中每一个分量设置为 999999)
 3. 给定初始稀疏向量 $\mathbf{W}_{ij} \in \mathbb{R}^P$, 初始化为 P 维零向量
 4. **while** $\|\mathbf{RT}_{ij}\| < \varepsilon \ \& \ \Psi \neq \emptyset$
 5. 求解 $\arg \max \langle \mathbf{T}_{ij}, \varphi_k(\mathbf{t}) \rangle, \varphi_k(\mathbf{t}) \in \Psi$, 记录 k 值。之后更新残差 \mathbf{RT}_{ij} :
 $\mathbf{RT}_{ij} = \mathbf{T}_{ij} - \langle \mathbf{T}_{ij}, \varphi_k(\mathbf{t}) \rangle \varphi_k(\mathbf{t})$.
 6. 更新集合 Ψ , $\Psi \rightarrow \Psi \setminus \{\varphi_k(\mathbf{t})\}$.
 7. 更新稀疏向量 \mathbf{W}_{ij} , 对其第 k 个分量赋值 $\langle \mathbf{T}_{ij}, \varphi_k(\mathbf{t}) \rangle$.
 8. **end while**
 9. 输出: 稀疏向量 \mathbf{W}_{ij}
-

2. 阈值最小二乘算法的流程

附表 2 阈值最小二乘算法流程

输入: $\mathbf{t}, \mathbf{T}_{ij}$, 允许误差 ε , 阈值 λ

1. 给定一个循环中值标识 flag, 初始化 flag 为 **false**
 2. 给定集合 Ψ , 初始化 $\Psi = \{1, 2, \dots, P\}$.
 3. 对数据进行最小二乘, 得出最小二乘解作为初始稀疏向量 $\mathbf{W}_{ij} \in \mathbb{R}^P$
 4. **for** $m = 1:10$
 5. flag = **false**
 6. **if** $\mathbf{W}_{ij}[k] < \lambda, k \in \Psi$
 $\mathbf{W}_{ij}[k] = 0$, 更新 Ψ 与 flag: $\Psi \rightarrow \Psi \setminus \{k\}$, flag = **true**.
 7. **end if**
 8. **if** flag = **false**, **break**;
 9. 在字典中选择数据集合 Ψ 的列继续做最小二乘, 并将得到的结果用于更新 \mathbf{W}_{ij}
 10. **end for**
 11. 输出: 稀疏向量 \mathbf{W}_{ij}
-

3. Adam 算法

附表 3 自适应学习率 Adam 算法流程

输入： 步长 ε （默认为 0.001），矩阵衰减速率 ρ_1, ρ_2 （默认 0.9, 0.999），终止条件 error.

1. 确定初始参数 θ ，初始化一阶矩，二阶矩变量 $s = 0, r = 0$ ，初始化时间步长 $t = 0$.

2. 选取一个稳定数值格式的小常数 ξ （一般取 $1e-6$ 即可）。

3. **while** MSE > error **do**:

4. 计算梯度: $g = \frac{1}{m} \nabla_{\theta} \sum_i L(f(x^i, \theta) - y^i)$

5. 更新时间: $t = t + 1$

6. 更新有偏一阶估计: $s = \rho_1 s + (1 - \rho_1)g$

7. 更新有偏二阶估计: $r = \rho_2 r + (1 - \rho_2)g$

8. 计算更新: $\Delta\theta = -\varepsilon \cdot \frac{s / (1 - \rho_1)}{\sqrt{r / (1 - \rho_2)} + \xi}$

9. 更新参数: $\theta = \theta + \Delta\theta$

10. **end while**

值得注意的是，该算法中是没有学习率需要输入的，步长和衰减速率都是默认输入，大多数情况一般不需要调整学习率，因此成为自适应学习率的算法，目前在 pytorch 框架下，一般用神经网络的优化方法都是 Adam。

4. 共轭梯度法

附表 4 共轭梯度法

输入： 初始参数 θ ，终止条件 error.

1. 初始化 $\rho_0 = 0, g_0 = 0$

2. 初始化时间迭代参数 $t = 1$

3. **while** MSE > error **do**:

4. 计算梯度: $g = \frac{1}{m} \nabla_{\theta} \sum_i L(f(x^i, \theta) - y^i)$.

5. 利用 Polak-Ribière 方法计算共轭方向: $\beta_t = \frac{\langle g_t - g_{t-1}, g_t \rangle}{\langle g_{t-1}, g_{t-1} \rangle}$.

6. 更新搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$.

7. **if** 代价函数为二次的:

8. 直接给出显示解 ε^* .

9. **else**: 搜索 $\varepsilon^* = \arg \min \frac{1}{m} \sum_i L(f(x^i, \theta) - y^i)$

10. 更新参数 $\theta_{t+1} = \theta_t + \varepsilon^* \rho_t$

11. **end while**

5. 自适应突变时空数据选取算法

附表 5 自适应突变时空数据选取算法

输入: 残差时间序列 δ_{ij}^t , 置信数值 ε

1. 初始化突变矩阵 A , 集合 X, Y
2. **for** i **in** 全球各个地区:
3. **for** $p = 1: -0.01: 0$
4. 取 δ_{ij}^t 绝对值的 p 分位数 $(|\delta_{ij}^t|)_p$.
5. 将 δ_{ij}^t 中绝对值大于 $(|\delta_{ij}^t|)_p$ 的放入集合 X , 否则放入 Y .
6. 对集合 X 中的元素做 Ljung-Box Q 检验, 给出 p 值.
7. **if** $p < \varepsilon$
8. 将集合 X 中的元素放入突变矩阵 A
9. **break.**
10. **end for**
11. **end for**

6. 四季辨识切换算法流程

附表 6 四季辨识切换算法流程

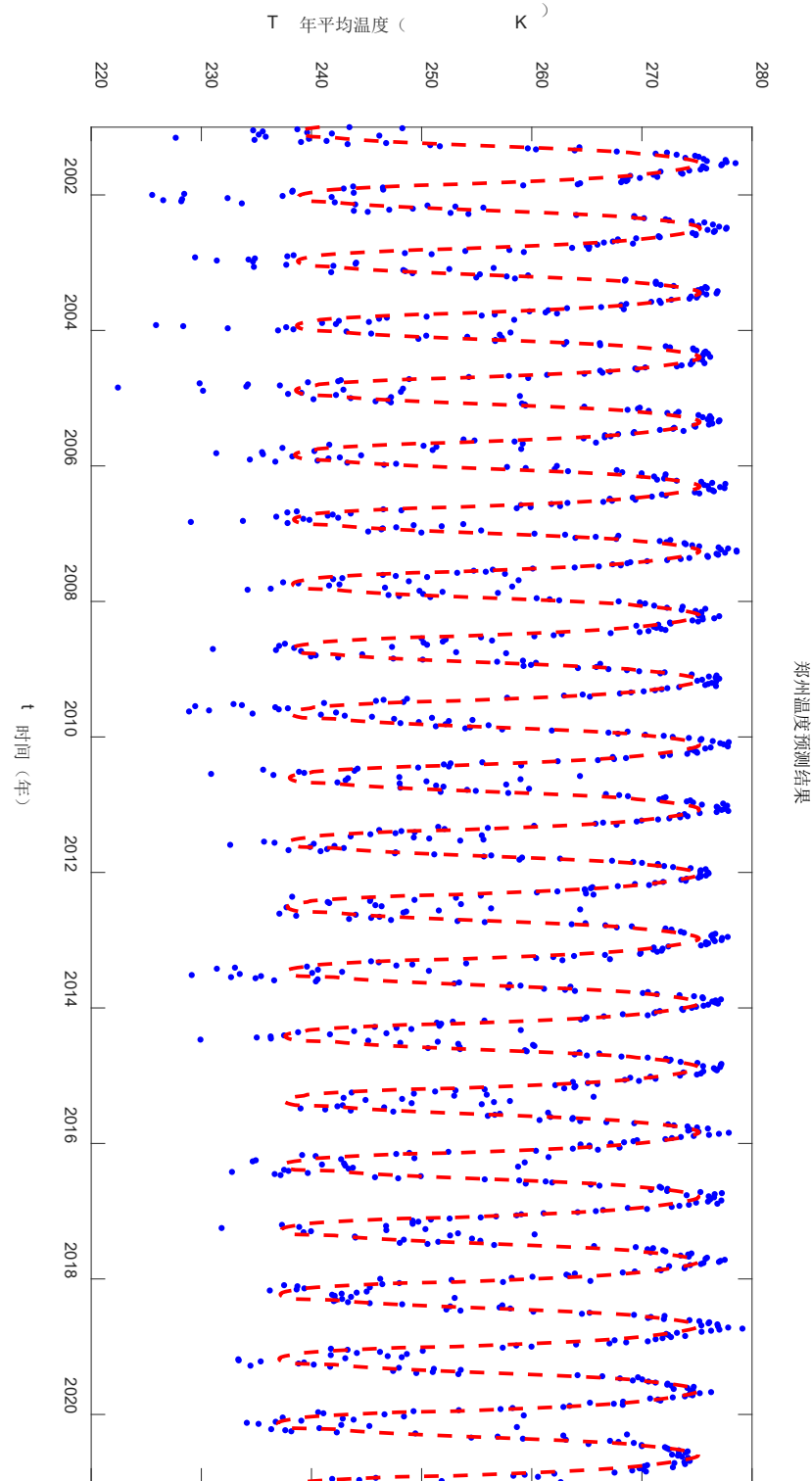
输入: 训练集集 $T_{i,j}^{y,d}$ (本文 $1950 \leq y \leq 2000$), STLS 和 SPB 算法中的允许误差 ε , 阈值 λ .

1. 如(2.15)定义日期——季节转换函数 $\varphi(d)$.
2. **for** $d = 1: 365$
3. $\bar{T}_{i,j}^d = \text{mean}(T_{i,j}^{y,d}, y)$
4. **end for**
5. 利用稀疏贝叶斯方法 (SPB) 对 $\bar{T}_{i,j}^d$ 进行稀疏辨识, 得到辨识函数 $f(d)$.
6. **for** $y = 1950: 2000$
7. $\tilde{T}_{i,j}^{y(k)} = \text{mean}(T_{i,j}^{y,\varphi(d)}, \varphi(d))$ ($k = 1, 2, 3, 4$)
8. **end for**
9. 利用阈值最小二乘 (STLS) 对 $\tilde{T}_{i,j}^{y(k)}$ 进行稀疏辨识, 得到辨识函数 $g_k(y)$
10. **for** $d = 1: 365$
11. **for** $y = 2001: 2020$
12. 由公式 $T_{i,j}^{y,d} = f(d) + g_{\varphi(d)}(y) - \frac{1}{51} \sum_{y=1950}^{2000} g_{\varphi(d)}(y)$ 将周期和四季趋势叠加与切换。
13. **end for**
14. **end for**
15. **输出:** 预测结果 $\hat{T}_{i,j}^{y,d}$ (本文 $2001 \leq y \leq 2021$)

B. 部分结果的可视化

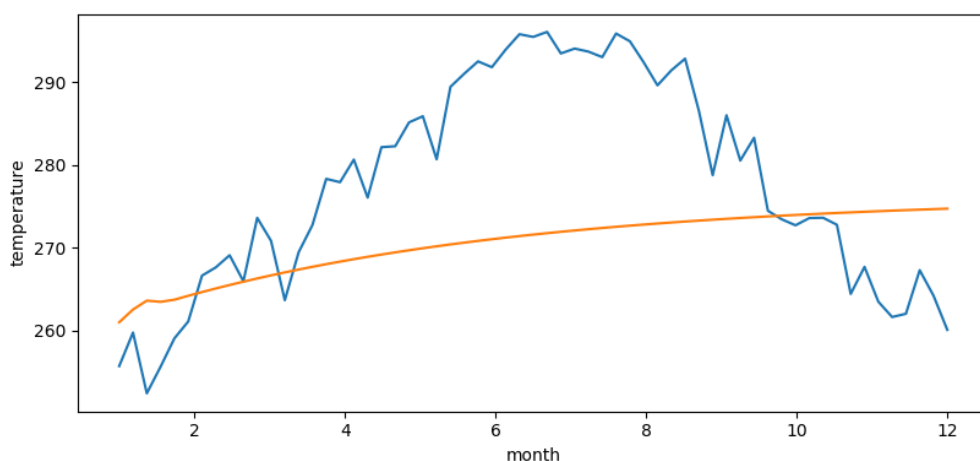
限于正文的篇幅，本文将部分可视化的结果放于附录，以供参阅。

1. 四季切换辨识算法的预测结果。（蓝点为真实数据，红线为辨识结果）



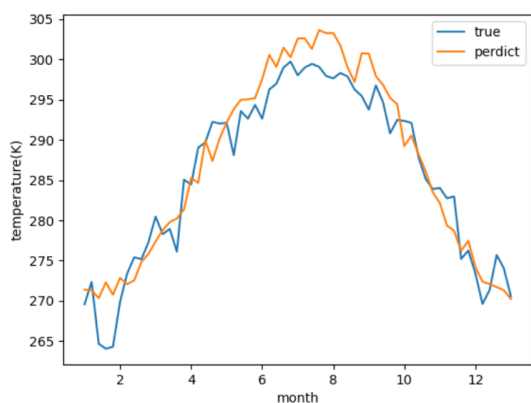
附图 1 郑州市四季切换辨识算法预测结果

2. ARIMA 预测结果图（该结果说明利用传统统计方法 ARIMA 预测失败）

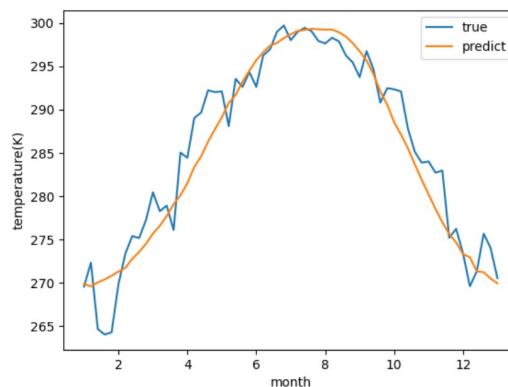


附图 2 ARIMA 预测某地一年气温结果

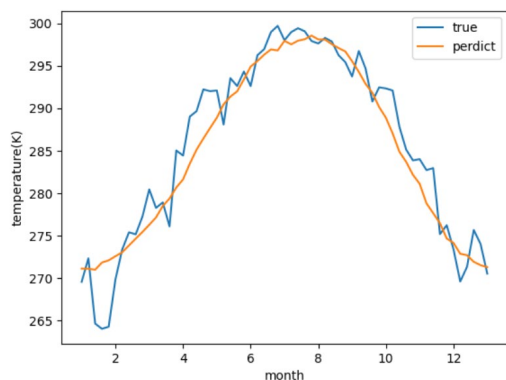
3. 部分网络的预测结果展示



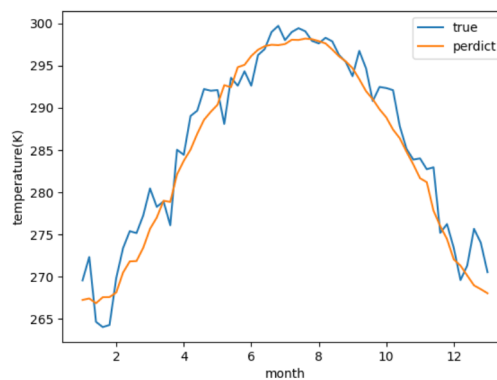
(a) BP 神经网络预测结果



(b) CNN 卷积神经网络预测结果



(c) ResNet 残差网络预测结果

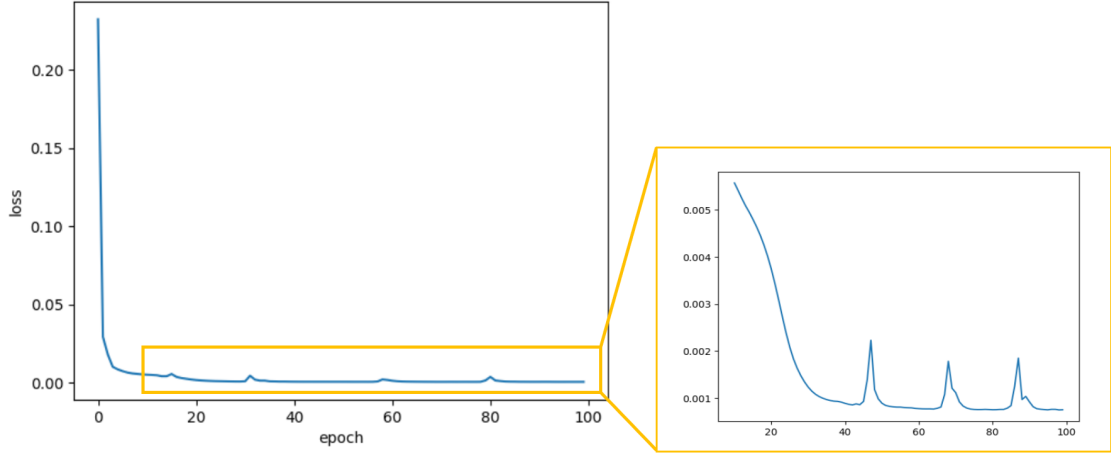


(d) LSTM 循环神经网络预测结果

附图 3 部分网络预测郑州周围地区一年气温结果

4. abrupt-CNN 训练 loss 图

本文训练了 100 个 epoch，其 loss 变化图如附图 4 所示，可以发现在 40epoch 之后基本趋于波动平缓，loss 不再下降，模型趋于稳定。



附图 4 训练网络时 Loss 随 epoch 的变化情况

C. 核心代码

由于本文涉及代码行数过多，难以全部展示，附录中仅展示文章提出的算法中一些核心代码，可供读者作为函数参考，不展示对比方法的代码。

(1) 数据预处理和评估代码

(环境: MATLAB)

```
function air_nor = norm_air(air, air_mean, air_std)
    air_nor = (air - air_mean)/air_std;
end
```

```
function air_inv = norm_inverse(air,air_mean, air_std)
    air_inv = air * air_std + air_mean;
end
```

```
function y = evaluate(T_predict, T_real)
    y(1) = sqrt(mean(mean((T_predict - T_real).^2)));
    y(2) = mean(mean(abs(T_predict - T_real)));
    T_mean = mean(mean(T_real));
    y(3) = 1 - sqrt(sum(sum((T_predict - T_real).^2))./sqrt(sum(sum((T_real - T_mean).^2)));
end
```

```
load air
air_mean_51 = mean(mean(air(:, 1:61*51)));
air_std_51 = std(reshape(air(:, 1:61*51),[],1));
air_pre = air(:, 61*51+1:61*72);
air_norm = norm_air(air, air_mean_51, air_std_51);
evaluate_matrix = zeros(6,3);
```

(2) 四季切换辨识代码

(环境: MATLAB)

```
function T_predict = FSSIA(T_test)
% 这是一个四季切换的辨识算法, 英文为 Four seasons switching identification
% algorithm. 取首字母命名为 FSSIA。

%% 这里是原文的周期辨识
air_avg = zeros(1152,61); % 这里 1152 表示地区, 61 表示 6 天一次记录。
for i = 1:51
    air_avg = air_avg + T_test(:,1+61*(i-1):61+61*(i-1));
end
air_avg = air_avg/51;
day = linspace(1,366,61)';
year = (1:72)';
A = basis(day);
coef = zeros(1152, size(A,2));
for i = 1:1152
    SBM_result = SparseBayes('Gaussian', A, air_avg(i,:));
    coef(i,SBM_result.Relevant) = SBM_result.Value;
end
air_reg = coef * A';

%% 这里是原文的四季辨识
spring = zeros(1152,72);
summer = zeros(1152,72);
autumn = zeros(1152,72);
winter = zeros(1152,72); % 这里 1152 表示地区, 50 代表预测集合有 50 年
for i = 1:72
    spring(:,i) = mean(T_test(:,11+61*(i-1):25+61*(i-1)),2);
    summer(:,i) = mean(T_test(:,26+61*(i-1):40+61*(i-1)),2);
    autumn(:,i) = mean(T_test(:,41+61*(i-1):55+61*(i-1)),2);
    winter(:,i) = mean([T_test(:,56+61*(i-1):61+61*(i-1)),T_test(:,1+61*(i-1):10+61*(i-1))],2);
end
A1 = poly_basis(year);
coef_spr = zeros(1152, size(A1,2)+1);
coef_sum = zeros(1152, size(A1,2)+1);
coef_aut = zeros(1152, size(A1,2)+1);
coef_win = zeros(1152, size(A1,2)+1);
lam = 0.005; % 0.001
```

```

for i = 1:1152
%     SBM_result_spr = SparseBayes('Gaussian', A1(1:50,:), spring(i,1:50));
%     coef_spr(i,SBM_result_spr.Relevant) = SBM_result_spr.Value;
%     SBM_result_sum = SparseBayes('Gaussian', A1(1:50,:), summer(i,1:50));
%     coef_sum(i,SBM_result_sum.Relevant) = SBM_result_sum.Value;
%     SBM_result_aut = SparseBayes('Gaussian', A1(1:50,:), autumn(i,1:50));
%     coef_aut(i,SBM_result_aut.Relevant) = SBM_result_aut.Value;
%     SBM_result_win = SparseBayes('Gaussian', A1(1:50,:), winter(i,1:50));
%     coef_win(i,SBM_result_win.Relevant) = SBM_result_win.Value;

    [temp,temp_info] = lasso(A1(1:51,:), spring(i,1:51)', 'Lambda',lam, 'NumLambda',1);
    coef_spr(i,:) = [temp',temp_info.Intercept];
    [temp,temp_info] = lasso(A1(1:51,:), summer(i,1:51)', 'Lambda',lam, 'NumLambda',1);
    coef_sum(i,:) = [temp',temp_info.Intercept];
    [temp,temp_info] = lasso(A1(1:51,:), autumn(i,1:51)', 'Lambda',lam, 'NumLambda',1);
    coef_aut(i,:) = [temp',temp_info.Intercept];
    [temp,temp_info] = lasso(A1(1:51,:), winter(i,1:51)', 'Lambda',lam, 'NumLambda',1);
    coef_win(i,:) = [temp',temp_info.Intercept];
end
A1 = [A1, ones(length(year),1)];
air_spr = coef_spr * A1';
air_sum = coef_sum * A1';
air_aut = coef_aut * A1';
air_win = coef_win * A1';
air_spr_avg = mean(air_spr(:, 1:51),2);
air_sum_avg = mean(air_sum(:, 1:51),2);
air_aut_avg = mean(air_aut(:, 1:51),2);
air_win_avg = mean(air_win(:, 1:51),2);

%% 这里是原文的辨识的叠加和切换
T_predict = zeros(1152, 61*72);
for i = 1:72
    air_temp = air_reg;
    air_temp(:, 1:10) = air_temp(:, 1:10) + air_win(:, i) - air_win_avg;
    air_temp(:, 11:25) = air_temp(:, 11:25) + air_spr(:, i) - air_spr_avg;
    air_temp(:, 26:40) = air_temp(:, 26:40) + air_sum(:, i) - air_sum_avg;
    air_temp(:, 41:55) = air_temp(:, 41:55) + air_aut(:, i) - air_aut_avg;
    air_temp(:, 56:61) = air_temp(:, 56:61) + air_win(:, i) - air_win_avg;
    T_predict(:, 1+(i-1)*61:i*61) = air_temp;
end

end

```

```

function [T_predict, T_coef] = SPB(T_test)
t = (1:61*72)';
cycle = 61;
A = [sin(2*pi*t/cycle), cos(2*pi*t/cycle), sin(4*pi*t/cycle), cos(4*pi*t/cycle), ...
      sin(6*pi*t/cycle), cos(6*pi*t/cycle), sin(8*pi*t/cycle), cos(8*pi*t/cycle),...
      sin(10*pi*t/cycle), cos(10*pi*t/cycle), sin(12*pi*t/cycle), cos(12*pi*t/cycle),...
      t, t.^2, t.^3, log(t), t.^(1/2), t.^(1/4), ones(length(t),1)];
coef = zeros(1152, size(A,2));
for i = 1:1152
    options = SB2_UserOptions('fixedNoise', true);
    settings = SB2_ParameterSettings('Beta', 30);
    SBM_result = SparseBayes('Gaussian', A(1:61*51, :), T_test(i,1:61*51)', options, settings);
    coef(i,SBM_result.Relevant) = SBM_result.Value;
end
T_predict = coef * A';
T_coef = coef;
end

```

(3) abrupt-CNN 网络模型代码

(环境: PyTorch 框架)

```

class Conv3d_block(nn.Module):
    def __init__(self):
        super(Conv3d_block, self).__init__()
        # batch * input_channels * lat * lon * time(61*year) --> batch * output_channels * lat *
lon * time_lenth
        self.conv3d_block_1 = nn.Sequential(
            nn.Conv3d(1, 1, kernel_size=5, padding=2),
            nn.ReLU()
            # batch * 1 * 24 * 48 * 61 --> batch * 1 * 24 * 48 * 61
        )
        self.conv3d_block_2 = nn.Sequential(
            nn.Conv3d(1, 1, kernel_size=7, padding=3),
            nn.ReLU()
            # batch * 1 * 24 * 48 * 61 --> batch * 1 * 24 * 48 * 61
        )
    def forward(self, x):
        y = torch.Tensor().to(device)
        y = self.conv3d_block_1(x)
        y = self.conv3d_block_2(y)
        y = y + x
        y = self.conv3d_block_1(y)
        y = self.conv3d_block_2(y)
        y = y + x
        # y = self.conv3d_block_2(y)

```

```
return y
```

```
class MyConvNet(nn.Module):
    def __init__(self, year):
        self.year = year
        super(MyConvNet, self).__init__()
        self.conv3_list = nn.ModuleList()
        for i in range(year):
            self.conv3_list.append(Conv3d_block())
        self.fc = nn.Sequential(
            nn.Linear(61*year, 100),
            nn.ReLU(),
            nn.Linear(100, 61),
            nn.ReLU(),
            nn.Linear(61,61*year)
        )
        self.decoder = nn.Sequential(
            nn.Linear(61*year, 100),
            nn.ReLU(),
            nn.Linear(100, 61),
            nn.ReLU(),
            nn.Linear(61,61)
        )
    def forward(self, x):
        y = torch.Tensor().to(device)
        for i in range(self.year):
            temp = self.conv3_list[i](x[:, :, :, 61*i:61*(i+1)])
            y = torch.cat((temp, y), dim=4)
        # y = y.permute(0, 4, 2, 3, 1)
        # y = y.squeeze()

        # y_temp = self.fc(y)
        # y_temp = y + y_temp
        y = self.decoder(y)
        return y
```

(4) 文中部分图像生成代码以及出处

图 1、图 2 均来自网页，以下是网址

图 1: https://m.sohu.com/a/414672415_114731/?pvid=000115_3w_a

图 2（左）: https://m.sohu.com/sa/308561769_479422?strategyid=00014

图 2（右）: https://m.sohu.com/a/480664696_121053574/?pvid=000115_3w_a&strategyid=00014

图 3 来源于数《深入浅出图神经网络》刘忠雨等人著，P48 图 3-7

其余图（包括附录中的附图）均为自己制作，部分制图代码如下。

%% 原文图 5

clc

clear

close all

%% 数据整合

load air

air_avg = zeros(1152,61); % 这里 1152 表示地区，61 表示 6 天一次记录。

for i = 1:50

air_avg = air_avg + air(:,1+61*(i-1):61+61*(i-1));

end

air_avg = air_avg/50;

day = linspace(1,366,61)';

year = (1:50)';

% plot(day, air_avg(1,:), 'r');

A = basis(day);

coef = zeros(1152, 11);

%% 一年温度 SPB 辨识

for i = 1:1152

options = SB2_UserOptions('fixedNoise', true);

settings = SB2_ParameterSettings('Beta', 1);

% SB2_FullStatistics('alpha', 0.05, 'beta', 0.1)

SBM_result = SparseBayes('Gaussian', A, air_avg(i,:), options, settings);

coef(i, SBM_result.Relevant) = SBM_result.Value;

end

air_reg = coef * A';

%% 结果

n = 6;

m = 32;

place = 48*(n-1)+m;

figure(1)

plot(day, air_avg(place,:), 'r', 'markersize', 10);

axis([0,366,260,290])

xlabel('t 时间 (日)', 'FontSize', 18)

ylabel('T 温度 (K)', 'FontSize', 18)

title('(c). 稀疏贝叶斯方法 (k=6)', 'FontSize', 18)

set(gca, 'FontSize', 18)

hold on

p = plot(day, air_reg(place,:), 'r', 'LineWidth', 2);

```

p.Color(4) = 0.5;

figure(2)
plot(day, air(place, 1+61*25:61+61*25),',', 'markersize',10)
axis([0,366,260,290])
xlabel('t 时间 (日)', 'FontSize',18)
ylabel('T 温度 (K)', 'FontSize',18)
title('温哥华 1975 年温度', 'FontSize',18)
set(gca, 'FontSize', 18)

```

%% 原文图 6

```

figure(3)
n = 23; %1:2
m = 8; %24:1,5,7
place2 = 48*(n-1)+m;

coef_spr = zeros(1152,2);
coef_sum = zeros(1152,2);
coef_aut = zeros(1152,2);
coef_win = zeros(1152,2);
coef_year = zeros(1152,2);

year_long = 51;
year = 1950:(1950+year_long);

air_year = zeros(1152,year_long + 1);
for i = 1:year_long + 1
    air_year(:,i) = mean(air(:,1+61*(i-1):61*i),2);
end

spring = zeros(1152,year_long + 1);
summer = zeros(1152,year_long + 1);
autumn = zeros(1152,year_long + 1);
winter = zeros(1152,year_long + 1);

for i = 1:year_long + 1
    spring(:,i) = mean(air(:,11+61*(i-1):25+61*(i-1)),2);
    summer(:,i) = mean(air(:,26+61*(i-1):40+61*(i-1)),2);
    autumn(:,i) = mean(air(:,41+61*(i-1):55+61*(i-1)),2);
    winter(:,i) = mean([air(:,56+61*(i-1):61+61*(i-1)),air(:,1+61*(i-1):10+61*(i-1))],2);
end

```

```

pre_len = 51;
for i = 1:1152
    coef_spr(i,:) = regress(spring(i,1:pre_len)', [year(1:pre_len)', ones(pre_len,1)]);
    coef_sum(i,:) = regress(summer(i,1:pre_len)', [year(1:pre_len)', ones(pre_len,1)]);
    coef_aut(i,:) = regress(autumn(i,1:pre_len)', [year(1:pre_len)', ones(pre_len,1)]);
    coef_win(i,:) = regress(winter(i,1:pre_len)', [year(1:pre_len)', ones(pre_len,1)]);
    coef_year(i,:) = regress(air_year(i,1:pre_len)', [year(1:pre_len)', ones(pre_len,1)]);
end
coef_all = [coef_spr(:,1), coef_sum(:,1), coef_aut(:,1), coef_win(:,1), coef_year(:,1)];
reg_spr = coef_spr * [year', ones(size(year,2),1)]';
reg_sum = coef_sum * [year', ones(size(year,2),1)]';
reg_aut = coef_aut * [year', ones(size(year,2),1)]';
reg_win = coef_win * [year', ones(size(year,2),1)]';
reg_year = coef_year * [year', ones(size(year,2),1)]';

plot(year, air_year(place2, :), '.', 'markersize',20);
hold on
p = plot(year, reg_year(place2,:), 'r--', 'LineWidth', 3);
p.Color(4) = 0.5;
set(gca, 'FontSize', 30)
axis([1950,1950+year_long,220,235])
xlabel('t 时间 (年) ')
ylabel('T 年平均温度 (K) ')
title('(a). 年际温度变化情况')

figure(4)
subplot(2,2,1)
plot(year, spring(place2, :), '.', 'markersize',10);
hold on
p = plot(year, reg_spr(place2,:), 'r--', 'LineWidth', 2);
p.Color(4) = 0.5;
axis([1950,1950+year_long,210,225])
set(gca, 'FontSize', 18)
xlabel('t 时间 (年) ')
ylabel('T 年平均温度 (K) ')
title('(b). 3-5 月平均温度变化情况(秋)')

subplot(2,2,2)
plot(year, summer(place2, :), '.', 'markersize',10);
hold on
p = plot(year, reg_sum(place2,:), 'r--', 'LineWidth', 2);
p.Color(4) = 0.5;
axis([1950,1950+year_long,205,220])
set(gca, 'FontSize', 18)

```



```

xlabel('t 时间（年）')
ylabel('T 年平均温度（K）')
title('(c). 6-8 月平均温度变化情况(冬)')

subplot(2,2,3)
plot(year, autumn(place2, :), '.', 'markersize',10);
hold on
p = plot(year, reg_aut(place2,:), 'r--', 'LineWidth', 2);
p.Color(4) = 0.5;
axis([1950,1950+year_long,215,230])
set(gca, 'FontSize', 18)
xlabel('t 时间（年）')
ylabel('T 年平均温度（K）')
title('(d). 9-11 月平均温度变化情况(春)')

subplot(2,2,4)
plot(year, winter(place2, :), '.', 'markersize',10);
hold on
p = plot(year, reg_win(place2,:), 'r--', 'LineWidth', 2);
p.Color(4) = 0.5;
axis([1950,1950+year_long,240,255])
set(gca, 'FontSize', 18)
xlabel('t 时间（年）')
ylabel('T 年平均温度（K）')
title('(e). 12-2 月平均温度变化情况(夏)')

```