

## 注意力改进的动态自组织模块化神经网络结构设计及应用

张昭昭, 潘浩然, 朱应钦

### 引用本文

张昭昭, 潘浩然, 朱应钦. [注意力改进的动态自组织模块化神经网络结构设计及应用](#)[J]. 计算机科学, 2024, 51(11A): 231000069-9.

ZHANG Zhaozhao, PAN Haoran, ZHU Yingqin. [Design and Application of Attention-enhanced Dynamic Self-organizing Modular Neural Network](#) [J]. Computer Science, 2024, 51(11A): 231000069-9.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

#### Similar articles recommended (Please use Firefox or IE to view the article)

##### [MB-ATMK:融合属性权重和时序元知识的多行为序列推荐模型](#)

MB-ATMK:Multi-behavior Sequential Recommendation Integrating Attribute Weights and Temporal Meta-knowledge

计算机科学, 2024, 51(11A): 231100047-9. <https://doi.org/10.11896/jsjcx.231100047>

##### [FCTNet:基于双域深度学习的公交车到站时间预测方法](#)

FCTNet:Bus Arrival Time Prediction Method Based on Dual Domain Deep Learning

计算机科学, 2024, 51(11A): 231000180-7. <https://doi.org/10.11896/jsjcx.231000180>

##### [基于季节分解的混合神经网络的时间序列预测](#)

Time Series Prediction of Hybrid Neural Networks Based on Seasonal Decomposition

计算机科学, 2024, 51(11A): 231200008-7. <https://doi.org/10.11896/jsjcx.231200008>

##### [基于相对位置编码转换器模块的深度步态识别网络](#)

Deep Gait Recognition Network Based on Relative Position Encoding Transformer

计算机科学, 2024, 51(11A): 240400064-6. <https://doi.org/10.11896/jsjcx.240400064>

##### [基于改进超像素采样的立体匹配网络](#)

Stereo Matching Network Based on Enhanced Superpixel Sampling

计算机科学, 2024, 51(11A): 231100005-7. <https://doi.org/10.11896/jsjcx.231100005>

# 注意力改进的动态自组织模块化神经网络结构设计及应用

张昭昭<sup>1</sup> 潘浩然<sup>1</sup> 朱应钦<sup>2</sup>

<sup>1</sup> 西安科技大学计算机科学与技术学院 西安 710054

<sup>2</sup> 墨西哥国立理工高级研究中心自动控制中心 墨西哥城 07360

(zzzhao123@126.com)

**摘要** 针对混沌时间序列的复杂性和非线性特点,提出了一种专注于此类挑战的新型神经网络模型,即注意力改进的动态自组织模块化神经网络模型(ADAMNN)。该模型基于分而治之的思想,通过注意力机制计算不同子网络与输入数据的相似度,并利用层次聚类自适应地划分子网络。随后,采用基于层次聚类的动态生长机制,对子网络簇进行增减,最后通过激活的子网络簇对输入样本进行在线学习;同时,结合传统的集成输出方法,提出了一种基于注意力机制的子网络加权集成输出方法。最终分别在 Mackey-Glass 时间序列、M-G 快时变时间序列、非线性系统辨识、煤矿开采过程中在瓦斯浓度数据集上进行了实验,ADAMNN 展现出了实时更新子网络中心、动态构建子网络簇的能力,而且与基于欧几里得空间的动态自适应模块化神经网络相比,预测准确度提高了约 40%。

**关键词:** 模块化神经网络;自组织神经网络;混沌时间序列;注意力机制;层次聚类

**中图分类号** TP183

## Design and Application of Attention-enhanced Dynamic Self-organizing Modular Neural Network

ZHANG Zhaozhao<sup>1</sup>, PAN Haoran<sup>1</sup> and ZHU Yingqin<sup>2</sup>

<sup>1</sup> Institute of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

<sup>2</sup> Departamento de Control Automático CINVESTAV-IPN(National Polytechnic Institute), Mexico City 07360, Mexico

**Abstract** In response to the complexity and non-linear characteristics of chaotic time series, this paper proposes a novel neural network model specifically designed to address these challenges: the attention-enhanced dynamic self-organizing modular neural network(ADAMNN). Grounded in the divide-and-conquer philosophy, this model employs an attention mechanism to compute the similarity between different sub-networks and input data, facilitating an adaptive partitioning of sub-networks through hierarchical clustering. Subsequently, a dynamic growth mechanism, based on hierarchical clustering, adjusts the size of sub-network clusters. Ultimately, activated sub-network clusters are employed for online learning of input samples. Simultaneously, we introduce a novel attention-based sub-network weighted ensemble output method, integrating traditional ensemble output approaches. Ultimately, experiments were conducted on the Mackey-Glass time series, the rapidly varying MG time series, in the realm of nonlinear system identification, and using gas concentration datasets from coal mining operations. The ADAMNN model exhibited its proficiency in real-time updates of sub-network centers and the dynamic formation of sub-network clusters. Moreover, compared to dynamic self-organizing modular neural networks based on Euclidean space, ADAMNN exhibits an approximately 40% improvement in prediction accuracy.

**Keywords** Modular neural networks, Self-organization neural networks, Chaotic time series, Attention mechanism, Hierarchical clustering

混沌时间序列是一类具有高度复杂性和不确定性的时间序列数据<sup>[1-2]</sup>。不同于传统的周期性或规律性时间序列,混沌时间序列的演化过程具有非线性、非周期性和高度敏感性的特点<sup>[3]</sup>。这种类型的时间序列常常存在着难以捉摸的内在规律和深层结构,给预测任务带来了巨大的挑战。

为了实现相比传统神经网络更加准确的混沌时间序列预测,研究者提出了自组织神经网络<sup>[4]</sup>。自组织神经网络以其自适应性和全局优化能力,能够捕捉混沌时间序列中的非线性、非周期性和高度敏感性特征,发现混沌时间序列中的潜在

模式和依赖关系,从而提供更准确的预测结果。Qiao 等<sup>[5]</sup>提出了一种基于相对贡献指标的自组织 RBF(Radial Basis Function,径向基函数)神经网络设计方法,通过结构调整和参数更新实现快速收敛和高预测精度,在非线性函数仿真和污水处理建模中展现出优越性能。El\_sousy 等<sup>[6]</sup>提出了一种用于双轴运动控制系统的识别和控制的自适应非线性扰动观测器(ANDO),结合多种控制器结构,实现了扰动抑制和参数不确定性的改善。Li 等<sup>[7]</sup>通过遗传算法优化海底沉积物纹理特征,降低了特征冗余,将降维后的特征项应用于自组织映

基金项目:中华人民共和国教育部国家留学基金委(CSC)项目(202310120001)

This work was supported by the China Scholarship Council(CSC)(202310120001).

通信作者:潘浩然(21208088019@stu.xust.edu.cn)

神经网络模型,显著提高了水下机器人对海底沉积物的自主分类感知能力,为海底沉积物分类识别提供了更优异的效果。

然而,混沌时间序列的复杂性和不确定性使得建立准确的神经网络模型变得困难,需要更复杂的模型结构和更多的训练数据。神经网络模型在处理长期依赖、捕捉混沌特性和避免过拟合等方面仍然存在限制。因此,有研究者提出了模块化神经网络<sup>[8]</sup>,以克服传统神经网络在混沌时间序列预测中的局限性。模块化神经网络通过将复杂任务分解为子任务,并使用不同的神经网络模块处理每个子任务,从而提供了一种有效的解决方案。研究者也提出了多种神经网络构建的改进方法,Zhang等<sup>[9]</sup>提出了一种多重自适应模块化神经网络,通过概率密度峰值快速聚类确定模块数量、条件模糊聚类划分子模块、训练误差峰值构造子模块结构,并基于距离测度进行子模块集成,实现了自适应网络构建。该方法仅需两个人工参数,提升了学习速度,有效应对了复杂问题,并在多项实验中展现了优于传统方法的性能。Terekhov等<sup>[10]</sup>提出了一种可以重复使用深度神经网络的架构,用于减少训练的权重数量。Watanabe等<sup>[11]</sup>提出了从分层神经网络中提取和简化结构的方法,从而分割问题并缩短计算时间。Phan等<sup>[12]</sup>提出了一种基于垂直分割的子网络分解方法,显著改善了神经网络的泛化能力。Ferreira等<sup>[13]</sup>提出了基于动力学FNN的方法估计CNN的有效架构,且与通过经验和计算密集型策略定义的复杂架构有相同的有效性。

针对混沌时间序列预测,模块化神经网络学习的重点之一是任务分解——任务分配,研究者通常使用结合“局部映射”特性的RBF神经网络进行任务分解,并在RBF神经网络后加入“与门”实现将分解后的任务分配给相应的子网络模块,以提高模块化神经网络的性能。Zhang等<sup>[14]</sup>提出了动态自适应模块化神经网络(Dynamic Adaptive Modular Neural Network, DAMNN),能够基于在线减法聚类辨识数据的空间分布,并结合模糊策略将不同子样本的空间数据动态分配给不同的子网络,虽然空间特征可以捕捉数据本身的特性,但难以捕获子网络之间的潜在关联和依赖关系,这可能导致模型在处理数据时无法充分利用子网络之间的相关信息。

为实现动态任务分配,Qiao等<sup>[15]</sup>提出了OSAMNN(Online Self-adaptive Modular Neural Network),利用RBF神经网络特性及“与门”,结合在线减法聚类实现在线任务分解,该网络可以根据在线工况数据自动调整网络模块数,但没有考虑样本之间的相关性和新样本对其他样本的影响。在此基础上,Guo等<sup>[16]</sup>提出了一种基于在线聚类算法的在线任务分解的改进自适应模块化神经网络(Online Adaptive Modular Neural Network, OAMNN),能够基于距离与密度对任务进行在线分解,使得网络具有更紧凑的结构,但基于欧几里得距离减法聚类的任务分配方式只能捕捉任务在欧几里得空间中的距离关系,无法充分利用任务之间的相关信息,从而限制了网络的表达能力。此外,子网络规模较小,难以针对任务的变化对模型进行即时调整,在面对具有大量特征和复杂关系的数据时,可能无法捕捉到数据中的重要模式和细微变化,从而导致模型性能下降。Guo等<sup>[17]</sup>提出了基于经验模式分解和滑动窗口机制的新型自组织模块化神经网络(SWEMD-MNN),通过基于改进经验模式分解(EMD)的任务分解方法,并使用熵和欧氏距离实现任务分配,该网络可以根据输入

数据的时间序列特征有效、动态地分解时间序列,捕捉数据的内在模式和动态特性。然而,该网络只关注了输入数据的时序特征并将子序列动态分配给子网络,在面对复杂问题时,只能增加子网络,缺乏对子网络的动态调整机制,限制了网络在复杂问题上的建模能力和适应性。

同时,上述研究大多着重于任务分解层,却忽略了集成输出层的优化。如何有效地整合不同子网络的输出,充分利用它们直接的差异和互补性仍然面临挑战。模块化神经网络通常使用基于欧几里得距离归一化权重来实现集成输出,但这种方法只考虑了子网络之间的距离差异,未充分考虑子网络之间的关系。此外,不同子网络的输出可能具有不同的置信度和可靠性,但在集成输出时未考虑这些因素。

针对以上问题,本文设计了一种基于注意力改进的动态自适应模块化神经网络(Attention-Enhanced Dynamic Adaptive Modular Neural Network)。该算法利用注意力机制动态分配和激活子网络,并根据注意力权重动态分配资源给最相关的模块。子模块旨在捕获数据中的不同模式和依赖关系,并利用动态注意力机制来衡量每个模块对最终预测的贡献。同时,引入基于注意力的权重计算方法,对数据和子网络之间的相关性进行建模,并将注意力权重分配给不同的子网络,以充分利用子网络之间的差异性和互补性。最后通过多个实验验证了该算法的性能,证明了在不同条件下该网络均能稳定收敛。

## 1 ADAMNN 结构

根据模块化神经网络的通用结构,ADAMNN结构可分为输入层、任务分解层、任务分配层、子网络层和集成输出层。在输入层,模型接收混沌时间序列数据作为输入任务分解层,负责将复杂的预测任务拆解为较小的子任务,使每个子任务专注于捕捉混沌时间序列数据的特定模式和特征。任务分配层通过注意力机制根据子任务的重要性和相关性,灵活地分配资源并将任务分配给不同的模块,从而增强了模型对最具信息量特征的捕捉和利用。子网络层由多个子网络组成,每个子网络专注于处理特定模式。最后,集成输出层,将各个模块的输出进行整合,生成流式数据的最终预测结果。ADAMNN具体的结构如图1所示。

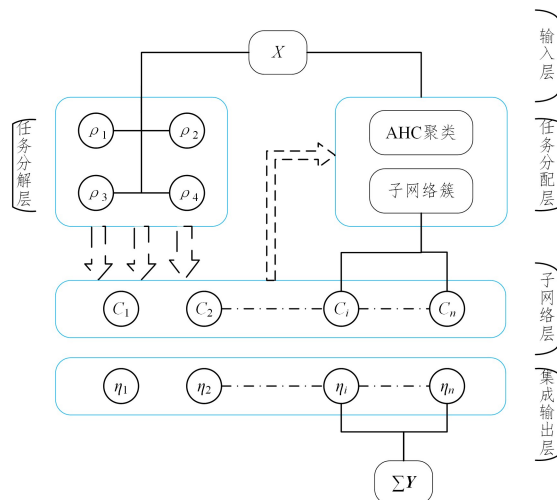


图1 ADAMNN 结构

Fig. 1 ADAMNN architecture

ADAMNN 各层的数学描述和功能如下。

1) 输入层。该层输入节点数  $n$  由  $k$  时刻样本维度  $x(k)$  决定:  $x(k) = [x_1(k) \ x_2(k) \ \cdots \ x_n(k)]^T$ 。

2) 任务分配层。该层中共包括一个注意力层、一个层次聚类层和  $l$  个“与门”。注意力层计算子网络间的自注意力,自注意力机制允许网络根据不同子网络之间的相互作用和重要性分配任务,计算不同子网络之间自注意力分数的方法如下:

$$\alpha_{i,j} = \text{att}(c_i, c_j) = \text{softplus}(c_i^T \cdot c_j) \quad (1)$$

其中,  $c_i$  和  $c_j$  是第  $i$  和  $j$  个子网络的中心,  $x_j$  是  $j$  时刻的输入向量。

通过计算注意力分数后,即可根据注意力分数构建层次聚类簇,并选择与输入数据相似度最高的子网络所在的簇,对于子网络  $i$  和激活的聚类簇  $\omega_j$ ,有:

$$I(i, \omega_j) = \begin{cases} 1, & i \in \omega_j \\ 0, & i \notin \omega_j \end{cases} \quad (2)$$

当子网络  $i$  在被激活的簇  $\omega_j$  中时,激活该子网络,输出为 1,子网络  $i$  准备对输入  $x$  进行学习;反之则不激活子网络  $i$ ,输出为 0。该层通过自注意力机制实现了子网络层的构建和任务分配。

3) 子网络层。该层包括  $l$  个结构简单的前馈神经网络(1 个输入层,1 个隐层,1 个输出层),主要任务是学习任务分配层分配的学习样本。

假设  $k$  时刻 ADAMNN 中第  $h$  个子网络的网络结构为  $(n-m-1)$  ( $n$  个输入节点,  $m$  个隐节点,1 个输出节点),则该子网络的输出为:

$$y_h(k) = \sum_{j=1}^m \omega_j(k) f\left(\sum_{i=1}^n v_{ji}(k) x_i(k)\right) \quad (3)$$

其中,  $\omega_j$  为子网络隐层中第  $j$  个隐节点与输出层节点之间的连接权重;  $v_{ji}$  为子网络输入层中第  $i$  个输入节点与隐层第  $j$  个隐节点之间的连接权重。隐节点激活函数为:

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

4) 集成输出层。该层主要对子网络层参与学习的子网络模块输出进行集成。本文采用了基于注意力的数据融合方法对输出进行集成,集成输出为:

$$Y(k) = \sum_{i=1}^l \alpha_i y_i(k) \quad (5)$$

其中,  $\alpha_i$  是归一化分布后的注意力权重,表示子网络模块  $i$  与输入  $x(k)$  的相关性,  $y_i(k)$  是子网络模块  $i$  在时刻  $k$  的输出。

## 2 模块化神经网络问题分析

### 2.1 模块化神经网络现存的问题

DAMNN(Dynamic Adaptive Modular Neural Network) 是一种动态自适应模块化神经网络<sup>[14]</sup>,具有根据输入任务可变的子网络数量和子网络规模。DAMNN 采用了减法聚类算法来在线识别不同工况数据的空间分布,利用径向基函数(RBF)神经元来划分数据样本空间。此外,DAMNN 还引入了模糊策略,使得不同子样本空间的数据可以动态地分配给不同的子网络。最终,通过不同子网络的空间关系,通过欧几里得距离划分权重,将各个子网络的输出集成在一起。其独特之处在于,它能够根据学习时变任务的要求,在线调整子网络的数量和规模。这种自适应性使得 DAMNN 能够更好地应对时变系统的挑战,提升了网络在

预测和跟踪时变系统方面的性能。

尽管 DAMNN 结构具备一定优势,但仍然存在一系列问题:

1) 基于欧几里得空间的在线减法聚类在实时数据流处理中表现出了对数据流中噪声的高度敏感性。

2) 该方法需要预先设定聚类半径,这使其在应对数据分布变化较为剧烈的情况下变得困难。这可能会导致实际应用中的聚类结果不准确甚至不稳定。

3) 基于子网络空间距离的权重划分受到数据范围的限制,这可能导致该方法难以处理具有不同尺度和分布的数据,无法捕捉数据的内在特征和模式。

针对以上问题,本文提出了基于层次聚类和注意力机制的改进方案,通过层次聚类和注意力权重的引入,可以实现更精确的子网络结构调整和特征捕捉,从而增强了网络的预测性能和自适应能力。具体改进内容如下:

1) 使用基于自注意力的层次聚类替换基于欧几里得距离的在线减法聚类,结合基于注意力的任务分配层,实现更为准确的子网络聚类和任务分配,增强模型的鲁棒性。

2) 引入注意力机制,以更灵活地分配权重,解决基于子网络空间距离的权重划分受限的问题,使得网络更好地适应不同尺度和分布的数据,捕捉更丰富的特征和模式。

### 2.2 实验验证:快时变系统案例

为了验证 ADAMNN 在解决 DAMNN 存在的问题方面的有效性,证明基于注意力的层次聚类生长机制和子网络权重融合的方法相比传统方法的优越性,针对 Mackey-Glass 时间序列混合得到的快时变系统进行了实验。

快时变系统模型通过混合 3 个不同的静态 Mackey-Glass 时间序列获得,即由  $\tau = 17 \rightarrow \tau = 30, \tau = 30 \rightarrow \tau = 50, \tau = 50 \rightarrow \tau = 100$  变化的数据各 700 个。通过对比变化阶段不同模型的响应速度和收敛速度,验证层次聚类的有效性,图 2 给出了不同模型学习阶段的子网络数量变化,以观察在快时变系统的变化阶段,层次聚类能够以较快的速度响应并改变网络结构,生长机制能够在网络结构趋于稳定后修剪掉多余的子网络,以控制模型的规模。相反,在线减法聚类无法控制子网络总规模,且子网络利用率较低,导致模型存在更多的冗余。

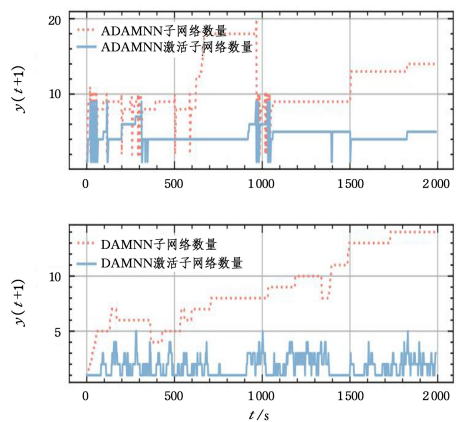


图 2 学习阶段的子网络变化图

Fig. 2 Subnetwork evolution during the learning phase

表 1 列出了 DAMNN、结合层次聚类的 DAMNN(AHC-DAMNN)、ADAMNN 对快时变系统的预测结果,由结果可知,将在线减法聚类替换为层次聚类后,预测精度仍然保持稳定,但不同的聚类方式会导致簇的规模不同,进而导致模型对

子网络的利用率不同,在加入注意力权重集成后,预测精度出现显著提升,证明相比距离权重,注意力权重能够更好地捕获模型的特征,并为子网络赋予更合理的权重。

表1 M-G 快时变系统预测实验结果

神经网络类型	模块数	学习 RMSE	运行时间/s
DAMNN	13	0.0384	26.8
AHC-DAMNN	14	0.0396	29.3
ADAMNN	14	0.0163	31.1

### 3 原理和方法

#### 3.1 自注意力动态任务分解算法

动态任务分解的目标是在多任务学习中提高模型的效率和性能,同时最大化资源的利用率<sup>[18]</sup>。实现这一目标的方法是将不同的任务动态地分配给不同的子网络。然而,选择合适的子网络进行任务分配是一个具有挑战性的问题,因为不同的子网络具有不同的特征和性能,并且任务之间的相似度可能会随时间变化。

为了解决上述问题,本文提出了一种基于自注意力机制的动态任务分配方法。该方法可以为所有子网络构建一个注意力矩阵,以表示它们之间的相似度。相似度的计算采用自注意力机制,即每个子网络计算自身的注意力分数,并将它们组合起来衡量子网络之间的相似度。

自注意力是一种基于注意力机制的模型<sup>[19]</sup>,它允许模型在输入序列中动态地关注不同位置的信息。在自注意力机制中,每个位置的表示都可以同时考虑到其他所有位置的表示,因此可以更好地捕捉序列中不同位置之间的依赖关系。

自注意力机制的计算包括3个步骤:首先计算一个查询向量 $Q$ 、一个键向量 $K$ 和一个值向量 $V$ ;其次通过单维元素计算得到注意力权重矩阵;最后将注意力权重矩阵与输入向量进行逐元素乘法得到最终的表示。具体过程如下:

对于任意输入值 $X=[x_1, x_2, x_3, \dots, x_N] \in \mathbb{R}^{d \times N}$ 和子网络组 $C=c_1, c_2, c_3, \dots, c_M \subseteq \mathbb{R}^d$ ,给定输入值 $x_k$ ,选择点积(Dot Product)<sup>[20]</sup>作为得分函数,即可通过式(6)得到 $x_k$ 与子网络组 $C$ 内任意子网络中心 $c_i \in C$ 的注意力分数。

$$\alpha_{ki} = \text{att}(x_k, c_i) = \text{softplus}(x_k^T \cdot c_i) \quad (6)$$

式(1)使用了 softplus 函数,而非传统的 softmax 函数。

softplus 函数的定义为:

$$\text{softplus}(x) = \log(1 + e^x)$$

由于指数的作用,softmax 函数会导致结果过度集中在单个元素上,这种被称为单峰性(unimodality)的属性阻碍了自注意力对输入信息整体状态的捕捉。softplus 是一个单调递增函数,它可以将注意力分数 $s(c_i, c_j) \in \mathbb{R}$ 映射到 $\alpha_{i,j} \in \mathbb{R}$ 。与 softmax 不同,当 $s(c_i, c_j) \in \mathbb{R}$ 为负数时,softplus 函数会将其映射为接近0的正数,而当 $s(c_i, c_j) \in \mathbb{R}$ 为正数时,softplus 则接近恒等映射。因此, $s(c_i, c_j) \in \mathbb{R}$ 的分布不再是单峰,得到的权重也不再过度集中在单个元素上,而是考虑全局的输入状态。

基于自注意力机制的动态任务分解方法为多任务学习中的子网络选择提供了一种新的解决方案,它不仅可以提高模型的效率和性能,还可以更好地利用资源。

为了进一步优化模型的性能,本文采用层次聚类对所有

子网络进行聚类,将相似的子网络分组在一起,以便于任务的动态分配。

层次聚类算法是一种通过计算相似度来不断合并或拆分聚类簇的算法。本文设定相似度为注意力权重 $\alpha_{i,j}, \forall i \in [0, n]$ 。

在使用式(1)计算完所有子网络之间的相似度后,可以构建注意力矩阵 $A$ ,用于表示各个子网络之间的注意力分布,对于 $n$ 个子网络,有:

$$A = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \vdots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \vdots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & \vdots & \alpha_{n,n} \end{pmatrix} \quad (7)$$

通过计算注意力矩阵 $A$ ,可以得到各个子网络之间的注意力分布。随后使用凝聚型层次聚类算法对子网络进行划分,本算法采用基于“自下而上”的思想的 Agglomerative Hierarchical Clustering(AHC)算法<sup>[21]</sup>,AHC 算法通过逐步合并最相似的样本或簇来构建聚类树(聚类的层次结构),直到最终形成一个包含所有样本的大簇,且不需要提前设定初始聚类簇中心和聚类数量即可将子网络组划分为不同的子网络簇。

具体流程如下:

Step1 初始化层次聚类簇集合 $C=\{C_1, C_2, \dots, C_n\}$ ,其中 $C_i$ 表示第 $i$ 个子网络,为一个单独的簇。

Step2 在注意力矩阵 $A$ 中找到相似度最高的两个簇 $C_i$ 和 $C_j$ ( $C_i, C_j = \text{argmax}(A[i, j]), i \neq j$ ),并将其合并为一个簇。

Step3 簇 $C_i$ 和 $C_j$ 合并为一个新的簇 $C_k$ ,更新注意力矩阵 $A$ ,以反映 $C_k$ 与其他簇之间的相似度 $A_{k,n} = \max\{A_{i,n}, A_{j,n}\}, \forall C_n \notin C_k$ ,其中, $A_{k,n}$ 表示簇 $C_k$ 与簇 $C_n$ 之间的相似度, $A_{i,n}$ 表示簇 $C_i$ 与簇 $C_n$ 之间的相似度, $A_{j,n}$ 表示簇 $C_j$ 与簇 $C_n$ 之间的相似度。

Step4 重复 Step2 和 Step3,直到所有子网络都被合并为一个大簇,大簇中子簇的数量由初始化的层次聚类层数决定。

接下来,根据输入任务 $x_k, n$ 个子网络集合 $c_i = \{c_1, c_2, \dots, c_n\}$ ,通过式(8)计算 $x_k$ 与子网络集合 $c_n$ 之间的相似度 $\alpha$ :

$$\alpha_i = \text{softmax}\left(\sum_{i=1}^n x_k c_i\right) \quad (8)$$

随后选择相似度的子网络 $c_{max}$ 所在的簇,将属于同一簇的子网络激活,从而实现任务的动态分配。这种方法可以根据输入任务的特征选择最适合处理输入变量的子网络,提高模型的效率和性能。

#### 3.2 注意力改进的子网络输出集成

选定合适的子网络后,需要解决的问题是如何确定每个子网络的权值,以正确整合它们的输出。这是由于不同的子网络可能对输入的不同方面具有不同的敏感性。因此,正确整合子网络的输出对于获取最佳性能至关重要<sup>[22]</sup>。

解决方法是使用注意力机制计算每个子网络在输入任务中的相对重要性。通过计算每个子网络与输入任务之间的相似度,可以得到一个注意力分布,该分布可用来指示每个子网络对输入的相对重要性。然后,根据注意力分布对不同子网络的输出进行加权求和,得到最终的整合结果。注意力加权融合可以最大化模型性能,提高其鲁棒性,合理地整合不同子

网络的输出,从而消除由单个子网络引起的潜在偏差和误差。

假设存在  $n$  个已选中并激活的子网络,每个子网络的输出为  $y_i \in \mathbb{R}^m$ ,其中  $m$  表示输出的维度。本文旨在通过将这  $n$  个子网络的输出集成为一个输出向量  $\hat{\mathbf{y}} \in \mathbb{R}^m$ ,并通过权重向量  $\boldsymbol{\alpha} \in \mathbb{R}^n$  来表示每个子网络的重要性,以实现更好的综合性能。若要确定权重向量  $\boldsymbol{\alpha}$ ,则需要计算每个子网络的重要性。具体而言,可以通过点积计算注意力分数来计算输入  $x_i$  和子网络  $j$  的输出向量  $y_j$  之间的权重。子网络  $j$  的注意力权重向量  $\boldsymbol{\omega}_{i,j}$  可表示为:

$$\boldsymbol{\omega}_{i,j} = (x_i)^T (y_j) \quad (9)$$

然后,可以将这些注意力权重向量归一化为注意力分布  $\mathbf{A}$ ,以表示每个子网络的重要性,具体地:

$$\alpha_i = \frac{\text{softmax}(\boldsymbol{\omega}_i)}{\sum_{j=1}^n \text{softmax}(\boldsymbol{\omega}_j)} \quad (10)$$

其中,  $\boldsymbol{\omega}_i = [\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,n}] \in \mathbb{R}^{nm}$  为所有子网络的注意力权重向量  $\boldsymbol{\omega}_{i,j}$  拼接而成的向量,  $\boldsymbol{\omega} \in \mathbb{R}^{n \times nm}$  为一个权重矩阵。

最后,可以将注意力分布作为权重向量  $\boldsymbol{\alpha}$ ,以计算集成后的输出向量  $\hat{\mathbf{y}}$ :

$$\hat{\mathbf{y}} = \sum_{i=1}^n \alpha_i y_i \quad (11)$$

其中,  $\alpha_i$  为注意力分布的第  $i$  个元素,即第  $i$  个子网络的注意力权重。这种方法可以动态地为不同的子网络分配不同的权重,从而使模型在不同任务和不同场景下取得更好的性能。

### 3.3 ADAMNN 学习过程

ADAMNN 通过动态任务分配调整子网络模块的结构和参数,实现对不同任务的灵活学习。ADAMNN 学习过程的步骤如下。

Step1 初始化密度阈值  $\rho_0$ , 设为小于总样本数的 2%, 局部距离阈值  $d_c$  设为输入矢量距离范围的 10%~20%, 注意力分数阈值  $att_{\min}$  输入注意力距离范围的 70%~85%。

Step2 数据输入层输入第 1 个数据,以第 1 个数据  $x(1)$  为聚类中心,并初始化聚类密度  $\rho_1 = 1$ ,初始化对应的子网络。

Step3 随着新样本的输入,由式(12)计算样本的局部数据密度  $\rho_i$ 。

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (12)$$

其中,  $d_{i,j}$  为样本  $x(i)$  到样本  $x(j)$  的欧氏距离;  $d_c$  为局部距离阈值;  $\chi$  为:

$$\chi = \begin{cases} 1, & d_{ij} \leq d_c \\ 0, & d_{ij} > d_c \end{cases} \quad (13)$$

当  $k$  时刻样本  $x(i)$  的局部密度  $\rho_i$  大于密度阈值  $\rho_0$  时,以样本  $x(i)$  为中心创建一个 RBF 子网络。

Step4 当子网络状态发生变化时,由式(6)计算并构建子网络簇  $c_i = \{c_1, c_2, \dots, c_n\}$  的相似度矩阵  $\mathbf{A}$ 。并基于相似度矩阵,设定目标聚类簇(手动选取,通常可设定为 4 和 5),构建层次聚类的过程可以将子网络组分成不同的聚类簇,每个簇中都包含相似的子网络。

Step5 对于  $k$  时刻输入的数据  $x(i)$ ,通过式(1)计算输入数据  $x(i)$  与所有子网络中心  $c_i$  的相似度,选取相似度最高的子网络,并激活该子网络所在的簇用于后续预测。

Step6 根据比较的结果,考虑以下几种情况对聚类中心  $c_i$  进行动态调整。

Step7 当  $\chi = 1$  时,说明当前数据比周围其他数据的描述能力更强,目前现有的子网络规模难以满足模型的预测需求,此时需要增加一个新的子网络,并令输入  $x(k)$  成为新增子网络的中心:  $c_{t+1} = x(k)$ 。

Step8 当两个不同子网络  $i, j$  的注意力分数  $\alpha_i$  和  $\alpha_j$  相差低于注意力阈值  $att_{\min}$  时,说明两个子网络对输入数据的捕获能力相似,此时可以合并两个子网络,并令注意力分数较高的子网络中心作为新的子网络中心,以降低模型规模。

Step9 当  $\chi = 0$  时,说明当前数据的描述能力较低,需要删除局部密度最小的子网络。

Step10 对  $k$  时刻激活的子网络进行训练。

Step11 下一时刻,新样本输入,返回 Step3。

## 4 仿真实验

本文选取 Mackey-Glass 时间序列预测、太阳黑子时间序列预测、非线性系统辨识及实际应用中瓦斯开采过程中的瓦斯浓度预测实验对 ADAMNN 性能进行测试,并与 DAMNN, SPRWNN, SWEMD-MNN 及其他在线神经网络模型进行比较。

选取均方根误差(Root Mean Square Error, RMSE)和子网络模块数作为模块化神经网络性能的评价标准,  $t_0$  时刻网络 RMSE 的计算式为:

$$RMSE_{t_0} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - Y_{Pred_i})^2} \quad (14)$$

其中,  $N$  为  $t_0$  时刻的样本数,  $Y_i$  是真实目标值,  $Y_{Pred_i}$  是模型在  $t_0$  时刻的预测输出值。RMSE 值越小,表示网络预测精度越高。

### 4.1 Mackey-Glass(M-G) 时间序列预测

ADAMNN 的性能通过 Mackey-Glass(M-G)混沌时间序列[23]进行测试。该系统的表达式如下:

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x^n(t-\tau)} + bx(t) \quad (15)$$

可以看出, M-G 系统是一个单变量时间序列预测问题。实验采用四阶-龙库塔方法产生 6 000 个数据,其中  $201 \leq t \leq 3000$  的数据用于训练,  $5001 \leq t \leq 5800$  的数据用于测试。  $[y(t); y(t-6); y(t-12); y(t-18)]$  作为输入向量,预测 85 步  $y(t+85)$  的值。

针对静态 M-G 时间序列预测的结果如图 3 所示。可以观察到, ADAMNN 能够很好地跟踪 M-G 时间序列的期望输出值。此外, ADAMNN 学习过程中的误差分布如图 4 所示。从误差结果可以看出,在模型的训练阶段,预测误差较大,在  $x=495$  可达到  $-0.025$ , 当  $x=1190$  时,误差可达到  $-0.04$ 。当网络结构稳定后, ADAMNN 的误差范围为  $[-0.01, 0.01]$ , 这表明 ADAMNN 具有较高的预测精度和稳定性。

关于模块数量,图 5 给出了 ADAMNN 在学习过程中激活子网络数量的变化情况。从图中可以观察到,在 ADAMNN 学习的初期阶段,子网络的数量迅速增加,构建出了完整的网络结构。一旦网络结构建立完成,子网络的数量就会逐渐趋于稳定,在整个学习过程中只会出现轻微的变化。同时,激活的模块数量被有效地控制在一个合理的范围内,证明了 ADAMNN 动态子网络构建的有效性。通过动态调整模块数量, ADAMNN 能够灵活地适应不同问题的复杂度和

要求,提高预测性能和适应性。

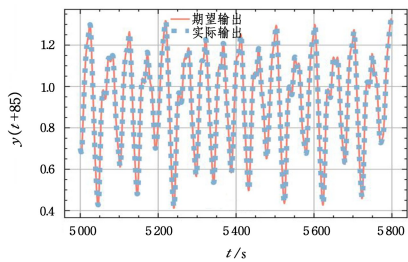


图3 Mackey-Glass 时间序列预测结果

Fig. 3 Predicted results of Mackey-Glass time series data

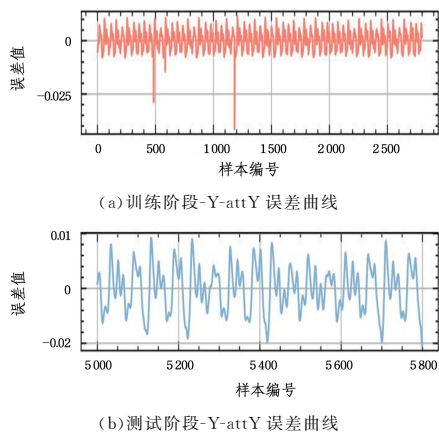


图4 训练与测试阶段的预测误差

Fig. 4 Prediction errors during training and testing phases

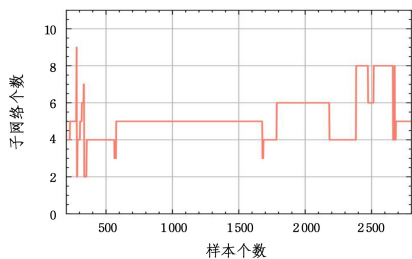


图5 ADAMNN 学习阶段激活子网络数量

Fig. 5 Number of activated subnetworks during adamnn learning phase

表2列出了不同算法对Mackey-Glass时间序列的预测结果,“—”表示在原文献中未给出。实验结果为独立10次实验的平均值,ADAMNN与在线模型DAMNN<sup>[14]</sup>,OAMNN<sup>[16]</sup>,EMD-MNN<sup>[17]</sup>和SPRWNN<sup>[24]</sup>进行了性能对比。由实验结果可以得出,ADAMNN与在线模块化神经网络相比,具有较高的预测精度,尽管模型整体规模较大,但激活的子网络却能保持在与其它模型相似的规模,使得模型预测速度较快。

表2 Mackey-Glass 时间序列预测实验结果

Table 2 Mackey-Glass time series prediction experiment results

神经网络类型	模块数	测试 RMSE	运行时间/s
DAMNN	7	0.0135	11.2
OAMNN	7	0.0281	—
SPRWNN	4	0.0123	1976.233
EMD-MNN	5	0.0051	27.7
DDMNN	4	0.0190	66.8
SWEMD-MNN	5	0.0070	25.6
ADAMNN	8	0.0042	13.6

综上所述,ADAMNN在Mackey-Glass时间序列预测中展现出了良好的性能。其学习过程中的子网络数量变化、输出与期望输出的对比以及误差分布等结果表明了其准确性和稳定性。此外,动态任务分配机制使得ADAMNN能够适应不同问题的特性和变化,进一步提升了其预测能力和适应性。

#### 4.2 太阳黑子时间序列预测

为了进一步评估ADAMNN对混沌时间序列预测的效果,针对太阳黑子时间序列进行了仿真实验。

太阳活动的变化对太阳周期的长短、地球气候的稳定性以及卫星与空间任务的执行产生了重要影响。然而这一过程的复杂性使得太阳周期的预测异常困难。太阳黑子时间序列是太阳活动周期性变化的良好指标。本实验从SIDC-太阳影响数据分析中心获取了每月的平滑数据,并选取了1834年11月—2001年6月共计2000个数据点,数据被标准化到 $[-1,1]$ 范围。前1000个样本用于训练,剩余的样本用于测试。

ADAMNN与其他模型(EMD-MNN,DDMNN,SWEMD-MNN)的性能比较总结如表3所列。

表3 太阳黑子时间序列预测实验结果

Table 3 Sunspot time series prediction experiment results

神经网络类型	模块数	测试 RMSE	运行时间/s
DAMNN	6	0.0527	15.3
EMD-MNN	7	0.0220	30.1
DDMNN	5	0.0460	124.8
SWEMD-MNN	7	0.0180	35.3
ADAMNN	10	0.0304	17.8

ADAMNN针对太阳黑子预测训练的误差如图6所示,通过整合多网络的输出,测试阶段预测误差范围为 $[-0.5, 0.3]$ ,与DDMNN和EMD-MNN相比,基于注意力的MNN模型能够更有效地整合子网络输出,提供更好的预测结果。同时对比基于EMD的MNN,ADAMNN只需计算在子网络发生变化时重新计算子网络的注意力矩阵,复杂度相对较低,运行时间更短,且能保持较优的预测性能。

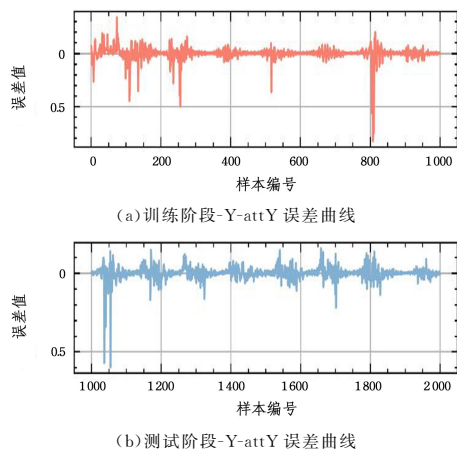


图6 太阳黑子时间序列预测误差

Fig. 6 Prediction errors of sunspot time series

学习阶段的激活子网络模块数如图7所示,与实验1的结果类似,在太阳黑子预测中,子网络数量最终保持在4~8之间。进一步证明了ADAMNN在面对复杂的非线性时序序列能够动态调整子网络的数量,以更好地适应系统的变化。

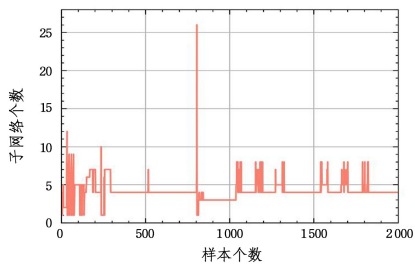


图7 太阳黑子预测学习阶段激活子网络数量

Fig. 7 Activation of subnetworks during the learning phase of sunspot prediction

### 4.3 非线性系统辨识

非线性系统辨识常被用于验证神经网络性能。

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t) \quad (16)$$

其中,非线性动态系统初始值为: $u(t) = \sin(2t/25)$ ,  $y(0) = 0$ ,  $y(1) = 0$ 。

本文通过式(15)生成了1800个实验数据,用于验证在线模块化神经网络的性能,其中 $1 \leq t \leq 1000$ 的数据用来训练神经网络,  $1001 \leq t \leq 1800$ 的数据用来测试神经网络。 $[y(t); y(t-1); u(t)]$ 作为输入向量,预测1步 $y(t+1)$ 的值。

图8和图9分别给出了网络的预测结果和学习过程中子网络训练RMSE的变化曲线。图8表明,模型能够较为准确地预测非线性动态系统的1步预测值,这表明模型具有较好的泛化能力。图9给出了子网络在测试阶段能够将RMSE控制在较小的值,这表明模型能够在线稳定收敛。

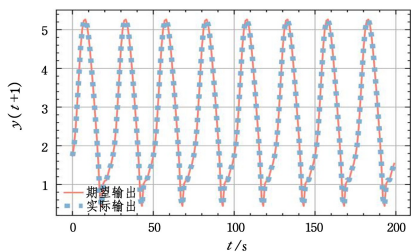


图8 非线性系统辨识结果

Fig. 8 Nonlinear system identification results

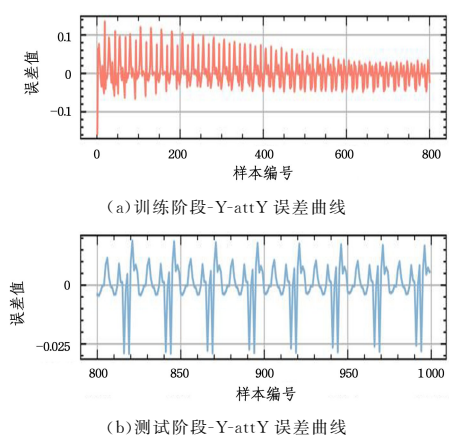


图9 训练与测试阶段的预测误差

Fig. 9 Prediction errors during training and testing phases

表4列出了不同算法在非线性系统辨识中的实验结果。实验结果为独立10次实验的平均值,由实验结果可以得出,ADAMNN与在线模型DAMNN<sup>[14]</sup>, OAMNN<sup>[16]</sup>和OS-

AMNN<sup>[15]</sup>进行了性能对比。由实验结果可以得出,在复杂的非线性系统拟合实验中,ADAMNN仍具有较高的预测精度,且能够保持模型的稳定性和性能。

表4 非线性系统辨识实验结果

Table 4 Nonlinear system identification prediction experiment

results			
神经网络类型	模块数	测试 RMSE	运行时间/s
OAMNN	4	0.0084	—
OSAMNN	7	0.0096	—
ADAMNN	15	0.0097	11.2

### 4.4 煤矿瓦斯浓度预测

近年来,煤矿普遍面临瓦斯排放管理和控制的挑战,瓦斯突出事故往往伴随着大量的瓦斯气体释放,对地下开采作业构成了巨大威胁,准确预测瓦斯浓度对于确保煤矿地下开采作业的安全至关重要<sup>[25]</sup>。然而,由于煤矿中瓦斯气体动态的复杂性和不确定性,准确测量瓦斯浓度仍然是一项具有挑战性的任务<sup>[26]</sup>。因此,本文提出了一种基于ADAMNN的时间序列预测方法,仅使用煤矿内瓦斯浓度的历史数据预测未来时刻的瓦斯浓度。

实验选取2021年10月30日—2021年11月18日从中国河北的煤矿收集得到的一段时间内不同区域的1300组瓦斯浓度数据进行在线学习,其中 $1 \leq t \leq 1000$ 的数据用于训练,  $1001 \leq t \leq 1300$ 的数据用于测试。

图10给出了ADAMNN对瓦斯浓度的预测结果,显示模型在实际问题中表现出较高的预测精度和准确性。表5列出了DAMNN和ADAMNN在测试阶段的性能对比。结果显示,ADAMNN在面对实际问题时能够保持较高的预测精度和准确性,表现出了良好的拟合效果。图11给出了训练阶段和测试阶段的预测误差分布,表明ADAMNN的预测结果与实际测量值之间的偏差较小。表5中的测试RMSE和运行时间数据显示,相比DAMNN,ADAMNN在预测性能和计算效率上均取得显著的改进。

ADAMNN对不同实验的预测误差箱线图如图12所示。可以看出在每个实验上ADAMNN都能取得较好的预测性能,同时,4个实验的预测误差标准差均较小,表明ADAMNN具有很强的鲁棒性。这种鲁棒性使得ADAMNN能够成为一种适用于实际煤矿场景的有效工具。

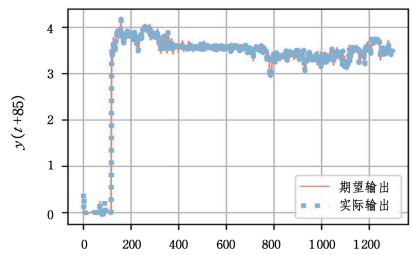


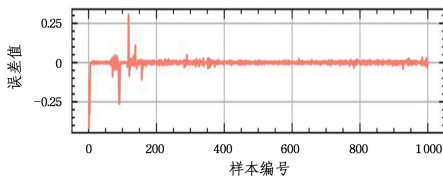
图10 瓦斯浓度预测实验结果

Fig. 10 Experimental results of gas concentration prediction

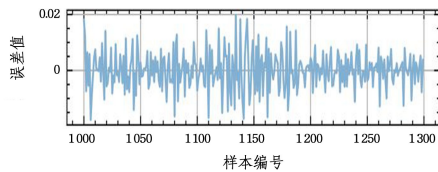
表5 瓦斯浓度预测实验结果

Table 5 Gas concentration prediction experiment results

神经网络类型	模块数	测试 RMSE	运行时间/s
DAMNN	8	0.1750	17.8
ADAMNN	22	0.0204	21.1



(a) 训练阶段-Y-attY 误差曲线



(b) 测试阶段-Y-attY 误差曲线

图 11 训练与测试阶段的预测误差

Fig. 11 Prediction errors during training and testing phases

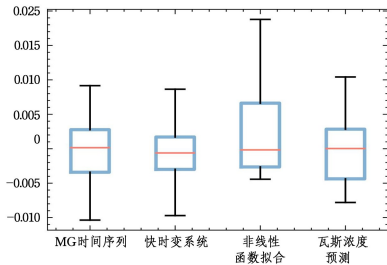


图 12 预测误差箱线图

Fig. 12 Box Plot of prediction errors

**结束语** 本文提出的 ADAMNN(注意力模块化神经网络)在多种应用中在混沌时间序列预测和煤矿瓦斯浓度预测方面展现了良好的性能。对于 ADAMNN 的进一步研究和应用,可以从以下几个方面展开。首先,可以通过进化算法优化网络结构,提高其适应性和灵活性,以适应不同问题领域并提升泛化能力。其次,可以将 ADAMNN 扩展到处理多变量时间序列,通过特征选择、降维和引入上下文信息等技术,来更好地捕捉和利用多个变量之间的复杂依赖关系。此外,还可以探索新的算法和优化技术,如改进的模糊聚类算法、模糊推理机制和自适应学习率策略,以提高训练速度和预测准确性。最后,可以进一步探索 ADAMNN 在其他领域的应用潜力,如股票市场预测、天气预测和交通流量预测,以提供准确的预测和决策支持。这些研究方向将有助于进一步改进和推动 ADAMNN 在实际应用中的效果和应用范围。

## 参 考 文 献

- [1] BALKISSOON S, FOX N, LUPO A, et al. Determining chaotic characteristics and forecasting tall tower wind speeds in Missouri using empirical dynamical modeling(EDM)[J]. *Renewable Energy*, 2021, 170:1292-1307.
- [2] CHENG W, WANG Y, PENG Z, et al. High-efficiency chaotic time series prediction based on time convolution neural network [J]. *Chaos, Solitons & Fractals*, 2021, 152:111304.
- [3] HAN M, ZHANG S, XU M, et al. Multivariate chaotic time series online prediction based on improved kernel recursive least squares algorithm[J]. *IEEE Transactions on Cybernetics*, 2018, 49(4):1160-1172.
- [4] BECKER S, HINTON G E. Self-organizing neural network that

- discovers surfaces in random-dot stereograms[J]. *Nature*, 1992, 355(6356):161-163.
- [5] QIAO J F, AN R, HAN H G. Design of self-organizing RBF neural network based on relative contribution index[J]. *CAAI Transactions on Intelligent Systems*, 2018, 13(2):159-167.
- [6] EL-SOUSY F F M, ABUHASEL K A. Adaptive nonlinear disturbance observer using a double-loop self-organizing recurrent wavelet neural network for a two-axis motion control system [J]. *IEEE Transactions on Industry Applications*, 2017, 54(1):764-786.
- [7] LI W L, GAO H W, JI D X, et al. Optimization Method of Seabed Sediment Texture Feature Based on Genetic Algorithm[J]. *Computer Science*, 2016, 43(Z6):130-133.
- [8] JACOBS R A, JORDAN M I, NOWLAN S J, et al. Adaptive Mixtures of Local Experts[J]. *Neural Computation*, 1991, 3(1):79-87.
- [9] ZHANG Z Z, QIAO J F, YU W. Structure Design of Hierarchical Adaptive Modular Neural Network[J]. *Chinese Journal of Computers*, 2017, 40(12):2827-2838.
- [10] TEREKHOV A V, MONTONE G, O'REGAN J K. Knowledge transfer in deep block-modular neural networks[C]// *Biomimetic and Biohybrid Systems; 4th International Conference, Living Machines 2015*, Barcelona, Spain. Springer International Publishing, 2015:268-279.
- [11] WATANABE C, HIRAMATSU K, KASHINO K. Modular representation of layered neural networks[J]. *Neural Networks*, 2018, 97:62-73.
- [12] PHAN K T, MAUL T H, VU T T. A parallel circuit approach for improving the speed and generalization properties of neural networks[C]// *2015 11th International Conference on Natural Computation(ICNC)*. IEEE, 2015:1-7.
- [13] ERREIRA M D F, CORRÊA D C, NONATO L G, et al. Designing architectures of convolutional neural networks to solve practical problems[J]. *Expert Systems with Applications*, 2018, 94:205-217.
- [14] ZHANG Z Z. Design of dynamic adaptive modular neural network structure. *Control and Decision*, 2014, 29(1), 64-70.
- [15] QIAO J, ZHANG Z, BO Y. An online self-adaptive modular neural network for time-varying systems [J]. *Neurocomputing*, 2014, 125:7-16.
- [16] GUO X, LI W, QIAO J. An improved online adaptive modular neural network. *Control and Decision*, 2020, 35(7), 1597-1605.
- [17] GUO X, LI W, QIAO J. A self-organizing modular neural network based on empirical mode decomposition with sliding window for time series prediction [J]. *Applied Soft Computing*, 2023:110559.
- [18] ZHANG Y, YANG Q. A survey on multi-task learning [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 34(12):5586-5609.
- [19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// *Advances in Neural Information Processing Systems*, 2017, 30:6000-6010.
- [20] PHAM H, LUONG M T, MANNING C D. 2015. Effective Approaches to Attention-based Neural Machine Translation[C]// *Proceedings of the 2015 Conference on Empirical Methods in*

Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015;1412-1421.

- [21] MURTAGH F, LEGENDRE P. Ward's hierarchical agglomerative clustering method; which algorithms implement Ward's criterion? [J]. Journal of classification, 2014, 31: 274-295.
- [22] BALTRUŠAITIS T, AHUJA C, MORENCY L P. Multimodal machine learning: A survey and taxonomy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(2): 423-443.
- [23] MACKEY M C, GLASS L. Oscillation, and chaos in physiological control systems[J]. Science, 1977, 197(4300): 287-289.
- [24] SU Y, YANG C, QIAO J. Self-organizing pipelined recurrent wavelet neural network for time series prediction[J]. Expert Systems with Applications, 2023, 214: 119215.
- [25] LAI X W, ZHENG W B, WU Y Q, et al. Task Collaborative Process Network Model and Time Analysis of Mine Accident Emergency Rescue Digital Plan[J]. Computer Science, 2021, 48(S1): 596-602.

- [26] WANG E Y, ZHANG G R, ZHANG C L, et al. Research progress and prospect on theory and technology for coal and gas outburst control and protection in China[J]. Journal of China Coal Society, 2022, 47(1): 297-322.



**ZHANG Zhaozhao**, born in 1973, associate professor, master's supervisor. His main research interests include intelligent information processing and optimization design of neural network structures.



**PAN Haoran**, born in 1998, master's research student. His main research interests include optimization design of neural network structures.