软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(9):4072−4092 [doi: 10.13328/j.cnki.jos.007257] [CSTR: 32375.14.jos.007257]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

增量构造式随机循环神经网络*

李文艺[1,2,3], 代 伟[1,2], 南 静[1,2], 刘从虎[3]

[1](中国矿业大学 人工智能研究院, 江苏 徐州 221116)
[2](中国矿业大学 信息与控制工程学院, 江苏 徐州 221116)
[3](宿州学院 机械与电子工程学院, 安徽 宿州 234000)
通信作者: 代伟, Email: weidai@cumt.edu.cn

摘 要: 针对循环神经网络 (recurrent neural network, RNN) 的结构不易确定、参数学习过程复杂等问题, 提出一种增量构造式随机循环神经网络 (incremental-construction for random RNN, IRRNN), 实现了 RNN 结构的增量构造与参数的随机学习. 首先建立隐含节点增量构造的约束机制, 同时利用候选节点池策略实现隐含节点的优选, 避免了网络随机构造的盲目性; 进一步, 从模型参数的局部优化与全局优化两个角度考虑, 提出模型参数的两种增量随机 (incremental random, IR) 学习方法, 即 IR-1 与 IR-2, 并证明了其万能逼近特性; 同时通过研究 IRRNN 的动态特性, 分析了 IRRNN 的泛化性能. 通过实验验证了 IRRNN 在动态特性、紧凑性和精度等多个方面具有良好特性.

关键词: 增量构造; 随机学习; 随机权神经网络; 循环神经网络; 稳定性
中图法分类号: TP18

中文引用格式: 李文艺, 代伟, 南静, 刘从虎. 增量构造式随机循环神经网络. 软件学报, 2025, 36(9): 4072–4092. http://www.jos.org.cn/1000-9825/7257.htm
英文引用格式: Li WY, Dai W, Nan J, Liu CH. Incremental-construction for Random Recurrent Neural Network. Ruan Jian Xue Bao/Journal of Software, 2025, 36(9): 4072–4092 (in Chinese). http://www.jos.org.cn/1000-9825/7257.htm

Incremental-construction for Random Recurrent Neural Network

LI Wen-Yi[1,2,3], DAI Wei[1,2], NAN Jing[1,2], LIU Cong-Hu[3]

[1](Artificial Intelligence Research Institute, China University of Mining and Technology, Xuzhou 221116, China)
[2](School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China)
[3](School of Mechanical and Electronic Engineering, Suzhou University, Suzhou 234000, China)

Abstract: Due to the difficulty in determining the structure and training the parameters of recurrent neural network (RNN), an incremental-construction for random RNN (IRRNN) is proposed to realize the incremental construction of RNN structures and the random learning of network parameters. The IRRNN establishes an incremental constraint mechanism for hidden nodes and uses the candidate node pool strategy to realize the optimal selection of hidden nodes, avoiding the blindness of random construction of the network. Two incremental random learning methods, termed IR-1 and IR-2, are designed for local and global optimization of model parameters. Additionally, their universal approximation property is proved. Meanwhile, the dynamic property of the IRRNN model is studied to analyze its generalization performance. Experiments validated that the IRRNN exhibits favorable dynamic properties, compactness, and accuracy.

Key words: incremental construction; random learning; random weight neural network; recurrent neural network (RNN); stability

循环神经网络 (recurrent neural network, RNN) 具有短期记忆能力, 能够实现上下文瞬时依赖关系的学习, 因此 RNN 成为深度学习领域重要的模型之一[1,2], 目前 RNN 模型在诸多领域有成功应用[3−5]. 由于 RNN 的隐含节

* 基金项目: 国家自然科学基金 (61973306, 62373361); 江苏省优秀青年基金 (BK20200086); 安徽省教育厅重点项目 (2022AH051361, 2023AH052233)
收稿时间: 2023-10-06; 修改时间: 2024-01-11, 2024-05-30; 采用时间: 2024-07-25; jos 在线出版时间: 2024-12-25
CNKI 网络首发时间: 2024-12-26

点之间存在耦合, 隐含层状态在时间上存在前后依赖, 因此与前馈神经网络相比, RNN 的参数学习更加复杂、耗时. Horn 等人[6]指出在 RNN 的误差平面上存在更多的伪极小值, 从而使 RNN 的参数学习更容易陷入局部最优. 鉴于 RNN 对时序数据处理能力强、应用范围广, 进一步研究 RNN 模型的构造方法及其学习方法有重要理论与实践意义.

RNN 的典型学习方法有沿时间反向传播 (back propagation through time, BPTT) 算法[7], 实时循环学习 (real-time recurrent learning, RTRL) 算法[8], 卡尔曼滤波 (Kalman filter, KF) 方法等[9,10]. BPTT 算法沿时间反向传递每一时刻的误差梯度, 然后用误差梯度更新网络参数. RTRL 算法首先利用前向传播计算梯度, 再利用梯度信息更新网络参数. 由于 BPTT 算法利用了所有时刻的梯度信息, 因此占用存储空间大, 且算法存在长程依赖问题. RTRL 算法仅保留当前的梯度信息, 空间复杂度小, 但是参数更新频繁, 计算量偏大. KF 方法是把 RNN 视为一个动态系统输入与内部状态的函数, 其模型参数作为系统状态, 从而采用状态估计方法来配置 RNN 参数.

当前 RNN 在网络拓扑结构有诸多改进[11]. 文献 [12] 提出了一种对角循环神经网络 (diagonal RNN, DRNN), 其仅保留了 RNN 中每个隐含节点到自身的反馈连接权重, 删除了隐含节点之间的相互连接权重, 通过结构简化, 减少了模型参数, 实现了网络的轻量化. 为捕获更长间隔的信息依赖关系, 长短期记忆 (long short-term memory, LSTM) 网络与门控循环单元 (gated recurrent unit, GRU) 网络[13,14]及其改进结构, 例如时空 LSTM (spatio-temporal LSTM, ST-LSTM)[15]、双向 LSTM (bi-directional LSTM, Bi-LSTM)[16]、变门 GRU (variant GRU)[17]等被相继提出. LSTM 是将内部存储状态与门控机制嵌入到 RNN 中, 从而可控制内部信息的存贮与传递过程. GRU 是 LSTM 的简化版本, 其仅考虑引入门控机制实现信息传递的有效调控. ST-LSTM 可实现内部信息沿空间与时间两个维度传递, 从而能够学习空间和时间两个维度上的依赖关系, 在人体姿态识别中取得良好效果[15]. Bi-LSTM 是在 LSTM 中设置了两个独立的隐含层, 实现了隐含层信息的双向传递, 使得网络既能利用历史信息也能利用未来信息[16]. 变门 GRU 通过改变 GRU 的门控信号生成机制, 使得门控信号的产生更加简单, 同时保持了 GRU 的性能[17]. 然而, 上述模型在具体应用中通常要使用试凑法来确定网络结构, 用迭代方法训练网络参数, 因此网络学习过程复杂、速度较慢.

将随机学习用于神经网络训练是一种提升其学习速度的有效方法[18,19], 并产生了一系列轻量化的机器学习模型, 如随机向量函数链接网络 (random vector functional-link network, RVFLN)[20], 极限学习机[21]等. 上述方法首先随机生成部分模型参数, 其余模型参数采用非迭代优化算法来确定, 其学习速度快. 神经网络的增量式随机学习实现了网络结构与参数的同步配置[22−26], 实现了模型结构根据学习任务的自主构建, 其基本思想是逐步向网络中添加随机隐含节点, 直至网络达到某些设定值时停止学习. 该方法既能实现网络结构的增量构造又能实现网络参数的随机学习[22−26]. 其中, 随机配置网络 (stochastic configuration network, SCN) 是一种最新的增量式随机学习[23−26], 其学习过程是: 逐步向网络中添加随机生成的、且满足一定约束条件的隐含节点, 并实时优化新增模型参数, 直到网络残差达到期望要求时学习停止. 由于 SCN 引入了随机参数约束机制, 有效保证了网络的万能逼近能力. 目前针对不同的应用环境, SCN 衍生出了块增量、并行增量等版本[24−26].

目前, RNN 的随机学习模型主要是回声状态网络 (echo state network, ESN)[27], 其采用随机方法生成输入权重与内部反馈权重, 再采用线性回归方法计算输出权重, 避免了耗时的误差梯度反传计算, 实现了 RNN 参数的快速学习[28]. 然而, 在实际应用中 ESN 结构的构建问题并没有很好的解决. 针对如何确定 ESN 结构, 当前方法包括剪枝方法[29−32]、构造方法[33]、构造-剪枝方法[34]. 文献 [29] 采用粒子群优化方法对 ESN 的输出权重优化, 删除贡献度较小的输出权重. 文献 [30] 把 ESN 隐层输出视为高维特征, 把输出权重的优化问题转化为特征选择问题, 进而实现输出权重的剪枝. 文献 [31] 利用 L1 正则化方法, 实现了 ESN 输出权重的剪枝. 文献 [32] 利用灵敏度分析方法判断子储备池模块的贡献度, 并根据网络规模适应度确定子储备池模块的个数, 删除灵敏度低的子模块. 文献 [33] 提出了生长型回声状态网络 (growing ESN, GESN), 其基本思想是: 每次向 ESN 中添加一个子储备池, 直到 ESN 满足设定值时停止学习. 文献 [34] 提出了一种自组织的 ESN (pseudo-inverse decomposition-based self-organizing modular ESN, PDSM-ESN), 其首先采用生长方法添加子储备池, 然后剪枝掉对泛化性能不利的子储备池, 从而实现隐含节点的自组织. 上述方法在一定程度上解决了 ESN 的结构学习, 但所得到的 ESN 内部权重连接

具有随机性, 模型结构不够紧凑.

本文以 DRNN 的拓扑结构为基础[12], 借鉴了神经网络增量式随机学习思想[23], 提出一种增量构造式随机循环神经网络 (incremental-construction for random recurrent neural network, IRRNN), 其实现了 RNN 结构的增量构造与参数的随机学习.

本文的主要贡献如下: (1) 在 IRRNN 的构造过程建立了随机学习的约束机制, 确保了学习过程的收敛性与 IRRNN 的稳定性, 并采用候选节点池策略对隐含节点进行优选, 使随机学习得到的隐含节点更有效率, 避免产生低质隐含节点. (2) 从局部优化与全局优化两个角度考虑, 提出了 IRRNN 的两种增量式随机 (incremental random, IR) 学习算法: 即算法 IR-1 与算法 IR-2. (3) 分析了 IRRNN 的逼近能力与动态特性, 证明了 IRRNN 存在唯一稳定平衡点的条件, 避免了初始状态对泛化性能的影响. (4) 通过多个动态系统仿真实验与工业软测量实验对 IRRNN 进行了验证, 实验结果表明 IRRNN 具有紧凑结构和良好精度.

1 相关理论基础

1.1 对角循环神经网络

DRNN 是一种单隐含层 RNN[12], 在 DRNN 中仅保留了 RNN 的隐含节点到自身的反馈连接权重, 删除了隐含节点之间的互连权重. 因此, DRNN 简化了 RNN 的结构, 减少了其参数量, 使网络更加轻量化. 虽然 DRNN 的结构简单, 但是在应用中其仍具有良好效果[35,36]. DRNN 可表示为:

$$\begin{cases} S(t) = f(Wx(t) + W^{\text{in}}S(t-1) + B) \\ y_{\text{DRNN}}(t) = VS(t) \end{cases} \tag{1}$$

其中, f 表示隐含节点的激活函数, t 表示时间 ($t = 1,2,\ldots,N$). $x(t) \in \mathbb{R}^p$, $y_{\text{DRNN}}(t) \in \mathbb{R}^q$ 分别表示 t 时刻的输入与输出. $S(t) = [s_1(t), s_2(t), \ldots, s_n(t)]^{\text{T}} \in \mathbb{R}^n$, $S(t-1) = [s_1(t-1), s_2(t-1), \ldots, s_n(t-1)]^{\text{T}} \in \mathbb{R}^n$ 分别表示隐含层在 t 时刻与 $t-1$ 时刻的输出状态向量, 其中 $s_i(t), s_i(t-1) \in \mathbb{R}\,(i = 1,2,\ldots,n)$ 分别表示第 i 个隐含节点在 t 时刻与 $t-1$ 时刻的输出状态. $W = [w_1, w_2, \ldots, w_n]^{\text{T}} \in \mathbb{R}^{n\times p}$、$W^{\text{in}} = \text{diag}(w_1^{\text{in}}, w_2^{\text{in}}, \ldots, w_n^{\text{in}}) \in \mathbb{R}^{n\times n}$、$V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{q\times n}$、$B = [b_1, b_2, \ldots, b_n]^{\text{T}} \in \mathbb{R}^n$ 分别表示输入层权重矩阵、隐含层反馈权重矩阵、输出层权重矩阵、隐含层偏置量. 公式 (1) 中的输出可表示为 $y_{\text{DRNN}}(t) = \sum_{i=1}^{n} v_i s_i(t)$.

1.2 前馈神经网络增量式随机学习

RVFLN 是 Pao 等人[20]提出的一种典型的前馈神经网络随机学习模型, 其由输入层、随机隐含层 (隐含层节点称为增强节点) 和输出层组成, 且在输入与输出之间存在直链. RVFLN 的训练过程包括: 在固定参数范围内随机生成增强节点的输入权重和偏置量, 并在此后的训练中保持不变, 通过 Moore-Penrose 伪逆求解输出权重矩阵 (包括直链权重). 由于上述学习过程未涉及迭代优化, 因此 RVFLN 的学习速度快. 传统 RVFLN 中增强节点的输入权重和偏置量完全随机, 导致部分随机节点质量较低, 且模型结构需要通过实验确定. 为解决 RVFLN 模型结构构建问题, 文献 [37] 对 RVFLN 扩展和改进, 提出了增量式的随机向量函数链接网络 (incremental RVFLN, IRVFLN), 其在 RVFLN 的基础上添加了动态增量学习机制, 采用构造方法使得网络能够逐步学习到合适的结构. 但由于 IRVFLN 中的隐含节点是在一个固定的区间内完全随机产生的, 当区间设置不恰当时网络性能难以提升, 且参数无约束地随机生成, 难以避免产生低质、冗余节点.

针对 IRVFLN 中随机学习中存在的问题, Wang 等人[23]在 2017 年提出了具有约束条件的增量式随机学习模型, 即 SCN. SCN 的基本思想是: 在一定约束条件下随机生成一些节点, 从中选择出使模型残差下降最大的节点添加至网络中, 再利用解析方法配置输出权重. SCN 的学习过程可简洁地表示为: $y_L(x) = y_{L-1}(x) + v_L f_L(x)$, 其中, y_{L-1} 为增量前网络 (含有 $L-1$ 个隐含节点) 输出, y_L 为增量后网络 (含有 L 个隐含节点), f_L 为第 L 个新增随机隐含节点输出, v_L 为新增节点 f_L 的输出权重, x 为输入量. 由于 SCN 对随机参数施加一定的约束, 使得 SCN 的隐含节点更加高效, 且随机范围能够随训练过程自主调整, 从而确保了 SCN 有万能逼近特性.

2 增量构造式随机循环神经网络

2.1 IRRNN 拓扑结构

IRRNN 的拓扑结构如图 1 所示, 其中隐含层节点仅含有自身反馈连接 (本文用 tanh 作为隐含节点的激活函数). 增量后的 IRRNN 输出是增量前网络输出与新增节点输出状态的线性组合, 即 $y_L(t)=y_{L-1}(t)+v_L s_L(t)$, 其中, y_L 与 y_{L-1} 分别表示含有 L ($L\geqslant 2$) 个隐含节点与含有 $L-1$ 个隐含节点的 IRRNN, s_L 表示新增隐含节点 f_L 的输出状态, v_L 表示 f_L 的输出权重, t 表示时刻.

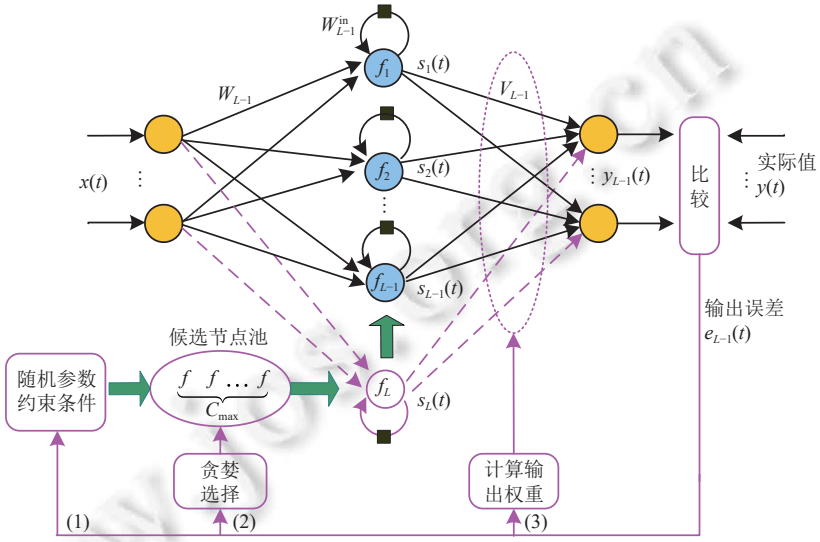


图 1 IRRNN 拓扑结构

图 1 中 $x(t)\in\mathbb{R}^p$ 和 $y_{L-1}(t)\in\mathbb{R}^q$ ($L\geqslant 2$) 分别表示网络 y_{L-1} 在 t 时刻的输入与输出. $W_{L-1}=[w_1,w_2,\dots,w_{L-1}]^{\mathrm{T}}\in\mathbb{R}^{(L-1)\times p}$、$V_{L-1}=[v_1,v_2,\dots,v_{L-1}]\in\mathbb{R}^{q\times(L-1)}$、$W^{in}_{L-1}=\mathrm{diag}(w^{in}_1,w^{in}_2,\dots,w^{in}_{L-1})\in\mathbb{R}^{(L-1)\times(L-1)}$、$B_{L-1}=[b_1,b_2,\dots,b_{L-1}]^{\mathrm{T}}\in\mathbb{R}^{L-1}$ 分别表示网络 y_{L-1} 的输入权重矩阵、输出权重矩阵、隐含层反馈权重矩阵、隐含层偏置量, 其中, w_i^{T} 是 W_{L-1} 的第 i 行, v_i 是 V_{L-1} 的第 i 列, w^{in}_i 是 W^{in}_{L-1} 的对角线上第 i 个元素, b_i 是 B_{L-1} 的第 i 个元素 ($i=1,2,\dots,L-1$). y_{L-1} 的隐含层在 t 时刻的输出状态向量为:

$$S_{L-1}(t)=[s_1(t),s_2(t),\dots,s_{L-1}(t)]^{\mathrm{T}}\in\mathbb{R}^{L-1} \tag{2}$$

其中, 状态 $s_i(t)$ 表示隐含节点 f_i 在 t 时刻的输出, 即 $s_i(t)=f_i(t)(i=1,2,\dots,L-1)$, y_{L-1} 在 t 时刻的输出为:

$$y_{L-1}(t)=V_{L-1}S_{L-1}(t)=\sum_{i=1}^{L-1}v_i s_i(t) \tag{3}$$

$X=[x(1),x(2),\dots,x(N)]$ 表示所有输入量, X 输入后, 隐含节点 f_i 的所有输出状态为:

$$s_i(X)^{\mathrm{T}}=[s_i(1),s_i(2),\dots,s_i(N)]^{\mathrm{T}}\in\mathbb{R}^N \tag{4}$$

X 输入后, 隐含层在所有时刻的输出状态记为隐含层输出矩阵 $S_{L-1}(X)$, 如公式 (5) 所示:

$$S_{L-1}(X)=[s_1(X)^{\mathrm{T}},\ s_2(X)^{\mathrm{T}},\dots,\ s_{L-1}(X)^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{(L-1)\times N} \tag{5}$$

y_{L-1} 所有时刻的输出为:

$$y_{L-1}(X)=[y_{L-1}(1),y_{L-1}(2),\dots,y_{L-1}(N)]=V_{L-1}S_{L-1}(X)=\sum_{i=1}^{L-1}v_i s_i(X) \tag{6}$$

其中, $y_{L-1}(t)$ 表示 y_{L-1} 在 t 时刻的输出. y_{L-1} 的输出误差为:

$$e_{L-1}=[(e^1_{L-1})^{\mathrm{T}},(e^2_{L-1})^{\mathrm{T}},\dots,(e^q_{L-1})^{\mathrm{T}}]^{\mathrm{T}}=Y-y_{L-1}(X)\in\mathbb{R}^{q\times N} \tag{7}$$

其中, $Y=[y(1),y(2),\dots,y(N)]$ 表示期望输出, $(e^j_{L-1})^{\mathrm{T}}\in\mathbb{R}^N(j=1,2,\dots,q)$ 表示第 j 维输出误差.

2.2 IRRNN 的学习方法

假设动态系统 $g:\mathbb{R}^p\to\mathbb{R}^q$ 在 t (t=1,2,…,N) 时刻的输入、输出分别为 $x(t)\in\mathbb{R}^p$, $y(t)\in\mathbb{R}^q$. 矩阵 $X=[x(1),x(2),\dots,x(N)]\in\mathbb{R}^{p\times N}$ 表示输入样本集, 矩阵 $Y=[y(1),y(2),\dots,y(N)]\in\mathbb{R}^{q\times N}$ 表示输出样本集. 建立一个 p 维输入 q 维输出的 IRRNN 逼近动态系统 g 的学习过程可概述为如下: 首先, 构造含有单个隐含节点的 IRRNN 作为初始网络. 然后, 每当需要给 IRRNN 添加新的隐含节点时, 先根据随机参数约束条件生成一组候选节点形成候选节点池, 再利用贪婪策略从候选节点池中选择一个最优节点添加到已有 IRRNN 中, 最后计算增量后 IRRNN 的输出权重, 直至 IRRNN 的输出误差满足设定值或者隐含节点数达到最大值时停止学习. 整个学习过程主要包括隐含节点生成, 隐含节点优选和输出权重计算 3 部分, 下面进行详细阐述.

(1) 隐含节点生成

随机生成的节点 f_L(即 f_L 的权重 w_L、w_L^{in}、b_L 是随机生成的, 其中 $L\geqslant 2$) 应满足以下不等式约束.

$$\frac{<(e_{L-1}^j)^{\mathrm{T}},s_L(X)^{\mathrm{T}}>^2}{\|(e_{L-1}^j)^{\mathrm{T}}\|^2\|s_L(X)^{\mathrm{T}}\|^2}\geqslant r_L\ (j=1,2,\dots,q) \tag{8}$$

其中, $s_L(X)^{\mathrm{T}}\in\mathbb{R}^N$ 表示节点 f_L 的所有输出状态, $(e_{L-1}^j)^{\mathrm{T}}\in\mathbb{R}^N$ 如公式 (7) 中所示, r_L 为非负数, 且 $\prod_{L=1}^{\infty}(1-r_L)=0$.

当 $L=1$ 时, 令公式 (7) 中 $y_0(X)=0$, 初始误差 $e_0=Y-0=Y$, 再按公式 (8) 约束条件生成隐含节点 f_1.

为简化设计, 在实际应用中公式 (8) 中的 r_L 可取为较小的正实数 r. 为确保网络稳定性, 根据第 2.4 节定理 7, f_L 的反馈权重满足 $0<|w_L^{\text{in}}|<1$.

(2) 隐含节点优选

我们希望对新增隐含节点 f_L 进行优选, 使 f_L 更有效率, 避免由于随机的盲目性而产生低效节点. 为叙述方便引入变量 $\xi_L^j(j=1,2,\dots,q)$, 如下所示.

$$\xi_L^j=\frac{<(e_{L-1}^j)^{\mathrm{T}},s_L(X)^{\mathrm{T}}>^2}{\|(e_{L-1}^j)^{\mathrm{T}}\|^2\|s_L(X)^{\mathrm{T}}\|^2}-r_L \tag{9}$$

其中, $s_L(X)$ 表示 f_L 的所有输出状态, 公式 (8) 简记为 $\xi_L^j\geqslant 0$. 对于第 j 维输出误差 e_{L-1}^j 来说, ξ_L^j 越大, 隐含节点 f_L 的效率越高. 令:

$$\xi_L=\sum\nolimits_{j=1}^{q}\xi_L^j \tag{10}$$

对于输出误差 e_{L-1} 来说, ξ_L 越大, 隐含节点 f_L 的效率越高. 基于上述分析, 本文采用候选节点池策略对隐含节点 f_L 进行贪婪优选, 其基本思想是: 首先, 按照公式 (8) 的约束条件随机生成 $C_{\max}$($C_{\max}>1$ 表示候选节点池容量) 个节点形成候选节点池 (记为 Ξ); 然后从 Ξ 中选择使 ξ_L 最大化的节点 f_L 作为最优隐含节点. 其优选过程可用公式 (11) 表示.

$$f_L=\arg\max_{f\in\Xi}\left\{\xi_L=\sum\nolimits_{j=1}^{q}\xi_L^j\,|\,\xi_L^j\geqslant 0\right\} \tag{11}$$

节点 f_L 添加到 y_{L-1} 后, 增量之后的网络 y_L 的隐含层输出矩阵 $S_L(X)$ 如下所示:

$$S_L(X)=\begin{bmatrix}S_{L-1}(X)\\ s_L(X)\end{bmatrix} \tag{12}$$

其中, $s_L(X)$ 表示第 L 个隐含节点 f_L 的所有输出状态组成的向量, 如公式 (4) 所示. $S_{L-1}(X)$ 表示增量之前的网络 y_{L-1} 的隐含层输出矩阵, 如公式 (5) 所示. 其中 $L\geqslant 2$, 当 $L=1$ 时 $S_1(X)=s_1(X)$.

(3) 输出权重计算

当 $L=1$ 时, 最优节点 f_1 为 IRRNN 的第 1 个隐含节点, 其相应参数为 w_1、w_1^{in}、b_1. 此时 IRRNN 仅含有单个隐含节点, 其输入权重矩阵 $W_1=w_1^{\mathrm{T}}$, 反馈权重矩阵 $W_1^{\text{in}}=w_1^{\text{in}}$, 偏置量 $B_1=b_1$.

当 $L>1$ 时, 最优节点 f_L 添加到 y_{L-1} 之后, 增量之后的网络 y_L 的输入权重矩阵 $W_L=[W_{L-1}^{\mathrm{T}},w_L]^{\mathrm{T}}$, 反馈权重矩阵 $W_L^{\text{in}}=\mathrm{diag}(W_{L-1}^{\text{in}},w_L^{\text{in}})$, 偏置量 $B_L=[B_{L-1}^{\mathrm{T}},b_L]^{\mathrm{T}}$, 其中 w_L、w_L^{in}、b_L 是 f_L 的相应参数. 根据输出权重计算方法不同, 本文设计了 IRRNN 的两种增量式随机学习算法 (incremental random, IR), 即算法 IR-1 与算法 IR-2.

● 算法 IR-1: 仅计算新增节点 f_L 的输出权重. f_L 的输出权重 $v_L=[v_L^1,v_L^2,\ldots,v_L^q]^{\mathrm{T}}$, 其中 $v_L^j(j=1,2,\ldots,q)$ 的计算如公式 (13) 所示:

$$v_L^j=\frac{<(e_{L-1}^j)^{\mathrm{T}},s_L(X)^{\mathrm{T}}>}{\|s_L(X)^{\mathrm{T}}\|^2} \tag{13}$$

增量后网络 y_L 在 t ($t=1,2,\ldots,N$) 时刻的输出如公式 (14) 所示:

$$y_L(t)=y_{L-1}(t)+v_Ls_L(t) \tag{14}$$

其中, $s_L(t)$ 为新增隐含节点 f_L 的输出状态. y_L 的输出权重 $V_L=[V_{L-1},v_L]$. X 输入后, y_L 的所有输出如公式 (15) 所示:

$$y_L(X)=[y_L(1),y_L(2),\ldots,y_L(N)]=V_LS_L(X)=y_{L-1}(X)+v_Ls_L(X) \tag{15}$$

若输出误差 $e_L=Y-y_L(X)$ 不满足设定值且隐含节点数未达到最大设定值, 则继续添加新的隐含节点.

● 算法 IR-2: 计算更新 IRRNN 的所有输出权重. 输出权重矩阵满足:

$$V_L^*=[v_1^*,v_2^*,\ldots,v_L^*]=\arg\min_{V_L}\|Y-V_LS_L(X)\| \tag{16}$$

由最小二乘法可得:

$$V_L^*=[v_1^*,v_2^*,\ldots,v_L^*]=YS_L(X)^+ \tag{17}$$

其中, $S_L(X)^+$ 为 $S_L(X)$ 的 Moore-Penrose 逆矩阵. 添加 f_L 之后, 增量后的网络记为 y_L^*, y_L^* 所有输出为:

$$y_L^*(X)=[y_L^*(1),y_L^*(2),\ldots,y_L^*(N)]=V_L^*S_L(X)=y_{L-1}^*(X)+v_L^*s_L(X) \tag{18}$$

其中, v_L^* 表示新增隐含节点 f_L 的输出权重. y_L^* 的输出误差 $e_L^*=[(e_L^{1*})^{\mathrm{T}},(e_L^{2*})^{\mathrm{T}},\ldots,(e_L^{q*})^{\mathrm{T}}]^{\mathrm{T}}=Y-y_L^*(X)$. 在 t 时刻 y_L^* 的输出为:

$$y_L^*(t)=y_{L-1}^*(t)+v_L^*s_L(t) \tag{19}$$

其中, $y_L^*(t)$, $y_{L-1}^*(t)$ 分别表示增量前网络 y_L^* 与增量后网络 y_{L-1}^* 在 t 时刻的输出, 若输出误差 e_L^* 不满足设定值且隐含节点数未达到最大设定值, 则继续添加新的隐含节点.

上述两种算法分别从局部优化与全局优化的角度考虑对网络输出权重进行优化, 两种算法的差异如下: 当有新增节点时, 算法 IR-1 仅计算新增节点 f_L 的输出权重, 其余节点的权重保持不变, 这体现了局部优化的思想. 当有新增节点时, 算法 IR-2 重新增量后网络的所有输出权重, 这体现了全局优化的思想.

2.3 IRRNN 的逼近性能

由第 2.2 节可知算法 IR-1 与算法 IR-2 是 IRRNN 输出权重的两种计算方法, 本节定理 1 证明算法 IR-1 具有万能逼近能力. 借助定理 1 的结论, 定理 2 证明了算法 IR-2 也具有万能逼近能力. 文中 $\|\cdot\|$ 表示向量的 2 范数, 或者矩阵的 F 范数.

定理 1. $X=[x(1),x(2),\ldots,x(N)]\in\mathbb{R}^{p\times N}$, $Y=[y(1),y(2),\ldots,y(N)]\in\mathbb{R}^{q\times N}$ 分别为动态系统的输入与输出; y_{L-1} 为含有 $L-1$ 个隐含节点的 IRRNN, 其输入为 X, 输出为 $y_{L-1}(X)\in\mathbb{R}^{q\times N}$, 输出误差如公式 (7) 所示; 给定非负数列 $\{r_L\}$, 满足 $\prod_{L=1}^{\infty}(1-r_L)=0$; 若第 L 个隐含节点 f_L 的所有输出状态 $s_L(X)$ 满足公式 (8), f_L 的输出权重为 $v_L=[v_L^1,v_L^2,\ldots,v_L^q]^{\mathrm{T}}$, 其中 $v_L^j(j=1,2,\ldots,q)$ 如公式 (13) 所示. 把 f_L 添加到 y_{L-1} 后, 新增之后的 IRRNN 为 y_L, 其输出 $y_L(X)$ 如公式 (15) 所示, 则 $\lim_{L\to\infty}\|Y-y_L(X)\|=0$.

证明: 分别从 e_{L-1}, e_L 中取 e_{L-1}^j, $e_L^j(j=1,2,\ldots,q)$. 新增节点 f_L 使公式 (8) 成立, f_L 的输出权重按公式 (13) 计算, 得:

$$\begin{aligned}\frac{\|(e_L^j)^{\mathrm{T}}\|^2}{\|(e_{L-1}^j)^{\mathrm{T}}\|^2}&=\frac{((e_{L-1}^j)^{\mathrm{T}}-v_L^js_L(X)^{\mathrm{T}})^{\mathrm{T}}((e_{L-1}^j)^{\mathrm{T}}-v_L^js_L(X)^{\mathrm{T}})}{\|e_{L-1}^j\|^2}\\&=1-\frac{<(e_{L-1}^j)^{\mathrm{T}},s_L(X)^{\mathrm{T}}>^2}{\|(e_{L-1}^j)^{\mathrm{T}}\|^2\|s_L(X)^{\mathrm{T}}\|^2}\leqslant 1-r_L<1\end{aligned} \tag{20}$$

随着 L 的增大, $\|(e_L^j)^{\mathrm{T}}\|$ 单调递减. 令 e_0^j 为 Y 的第 j 行, 若 $\|(e_0^j)^{\mathrm{T}}\|\neq 0$, 则有:

$$\frac{\|(e_1^j)^{\mathrm{T}}\|^2}{\|(e_0^j)^{\mathrm{T}}\|^2}\times\frac{\|(e_2^j)^{\mathrm{T}}\|^2}{\|(e_1^j)^{\mathrm{T}}\|^2}\times\ldots\times\frac{\|(e_L^j)^{\mathrm{T}}\|^2}{\|(e_{L-1}^j)^{\mathrm{T}}\|^2}=\frac{\|(e_L^j)^{\mathrm{T}}\|^2}{\|(e_0^j)^{\mathrm{T}}\|^2}\leqslant\prod_{i=1}^{L}(1-r_i)\tag{21}$$

由 $\prod_{L=1}^{\infty}(1-r_L)=0$, 因此:

$$\lim_{L\to\infty}\frac{\|(e_L^j)^{\mathrm{T}}\|^2}{\|(e_0^j)^{\mathrm{T}}\|^2}\leqslant\lim_{L\to\infty}\prod_{i=1}^{L}(1-r_i)=0\tag{22}$$

得 $\lim_{L\to\infty}\|e_L^j\|^2=0$. 由于 $\|e_L\|^2=\sum_{i=1}^{q}\|e_{L,i}\|^2$, 所以有:

$$\lim_{L\to\infty}\|Y-y_L(X)\|^2=\lim_{L\to\infty}\|e_L^{\mathrm{T}}\|^2=\lim_{L\to\infty}\left(\sum_{j=1}^{q}\|e_{L,j}^{\mathrm{T}}\|^2\right)=0\tag{23}$$

由公式 (23) 可知定理 1 成立. 证毕.

定理 2. $X=[x(1),x(2),\ldots,x(N)]\in\mathbb{R}^{p\times N}$, $Y=[y(1),y(2),\ldots,y(N)]\in\mathbb{R}^{q\times N}$ 分别为动态系统的输入与输出; y_{L-1} 为含有 $L-1$ 个隐含节点的 IRRNN, 其输入为 X, 输出为 $y_{L-1}(X)\in\mathbb{R}^{q\times N}$, 输出误差如公式 (7) 所示; 隐含层输出矩阵 $S_{L-1}(X)$ 如公式 (5) 所示; 给定非负数列 $\{r_L\}$, 满足 $\prod_{L=1}^{\infty}(1-r_L)=0$; 若第 L 个隐含节点 f_L 的输出状态 $s_L(X)$ 满足公式 (8), 把 f_L 添加到 y_{L-1} 后, 隐含层输出矩阵 $S_L(X)$ 如公式 (12) 所示, 新增之后的网络为 y_L^*, 其输出如公式 (18) 所示. 其中输出权重 V_L^* 如公式 (16) 所示, 则 $\lim_{L\to\infty}\|Y-y_L^*(X)\|=0$.

证明: 由公式 (16) 可知 $V_L^*=YS_L(X)^+$ 是 $Y=V_LS_L(X)$ 的极小范数最小二乘解, 由极小范数最小二乘解的性质可知 V_L^* 使得误差范数 $\|e_L^*\|=\|Y-V_L^*S_L(X)\|$ 最小, 即:

$$\|e_L^*\|=\|Y-V_L^*S_L(X)\|\leqslant\|Y-V_LS_L(X)\|=\|e_L\|\tag{24}$$

其中, V_L 中元素按照定理 1 计算. 由定理 1 可知 $\lim_{L\to\infty}\|e_L\|=0$, 所以 $\lim_{L\to\infty}\|e_L^*\|=0$. 证毕.

2.4 IRRNN 的动态特性

IRRNN 是一个动态系统, 因此必须了解其动态特性才能更好地应用该模型. 由公式 (3) 可知 IRRNN 的输出是隐含节点状态的线性叠加, 因此研究单个隐含节点的动态特性对于研究 IRRNN 的动态特性有重要意义. 本节首先给出单个隐含节点动态特性的相关结论, 然后给出 IRRNN 稳定性条件.

IRRNN 中第 i 个隐含节点在 t 时刻的输出状态 $s_i(t)=f_i(w_ix(t)+w_i^{\mathrm{in}}s_i(t-1)+b_i)$, 其中 $x(t)$ 为输入, $s_i(t-1)$ 为前一时刻的输出状态, w_i 为输入权重, w_i^{in} 为反馈权重, b_i 为偏置量. 当外部输入为 0 时, 第 i 个隐含节点的状态 $s_i(t)=f_i(w_i^{\mathrm{in}}s_i(t-1)+b_i)$, 此时激活函数 f_i 的输入、输出关系如图 2 中曲线 l_2 所示, 其中横轴表示输入 $s_i(t-1)$, 纵轴表示输出 $s_i(t)$. 当 f_i 处于平衡点时, 输入与输出相等, 即 $s_i(t)=s_i(t-1)$, 因此平衡点是直线 $l_1: s_i(t)=s_i(t-1)$ 与曲线 l_2 的交点. 例如 w_i^{in}=1.7, $b_i=-0.1$ 时, l_2 与 l_1 的 3 个交点是 $A(-0.934,-0.934)$, $B(0.144,0.144)$, $C(0.887,0.887)$. l_2 与横轴的交点是 $M=0.059$, 如图 2 所示. 当 w_i^{in} 与 b_i 变化时 M 的位置改变, 同时 l_2 与 l_1 的交点发生变化 (即平衡点的数量与位置改变). 为表达简洁, 以下定理中及仿真实验中用 f 表示隐含节点 f_i, 用 w, w^{in}, b 表示其相应权重.

定理 3. 每个隐含节点必定有平衡点, 平衡点个数可能是 1 个、2 个或者 3 个.

定理 4. 如果 $f(f(s))=s$ 的解不是隐含节点平衡点, 则必定是隐含节点的震荡点.

定理 5. $f'(s_e)$ 是激活函数 f 在平衡点 s_e 处的导数, 当 $|f'(s_e)|<1$ 时, s_e 是稳定平衡点; 当 $|f'(s_e)|>1$ 时, s_e 是不稳定平衡点; 当 $|f'(s_e)|=1$ 时, 不能用 $|f'(s_e)|$ 判定平衡点 s_e 的稳定性.

定理 6. 若 $s_{e1}<s_{e2}<s_{e3}$ 是隐含节点的 3 个平衡点, 则 s_{e2} 是不稳定平衡点, s_{e1} 和 s_{e3} 是两个稳定平衡点, s_{e1} 和 s_{e3} 的稳定域分别为 $(-1,s_{e2})$, $(s_{e2},1)$.

定理 7. 当 IRRNN 中任意隐含节点的反馈权重 $0<|w^{\mathrm{in}}|\leqslant 1$ 时, IRRNN 有唯一稳定平衡点, 且 IRRNN 的稳态输出由输入唯一确定与隐含节点的初始状态无关.

定理 3–定理 7 的证明过程见本文附录.

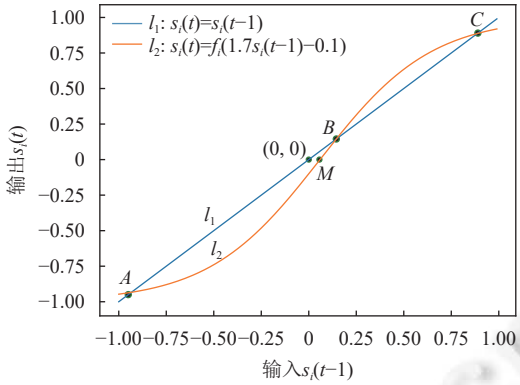


图 2 隐含节点的平衡点

定理 3 证明了 IRRNN 平衡点的存在性. 定理 4 给出了其稳定平衡点的解析计算方法. 定理 5 可以判断该平衡点是否稳定. 定理 6 给出了稳定平衡点的稳定域的确定方法. 定理 7 表明如果每个新增随机节点有唯一稳定平衡点, 则 IRRNN 有全局唯一稳定平衡点, 此时随时间的推移, 初始状态对输出的影响可以忽略. 定理 7 进一步表明当 IRRNN 有唯一稳定平衡点时, 相同的输入会产生相同的输出, 避免初始状态对网络稳态输出的影响, 进而避免初始状态对泛化性能的影响. 同时定理 7 为算法 IR-1 与算法 IR-2 提供了直接的理论支撑, 根据定理 7, 在算法 IR-1 与算法 IR-2 中应设定随机权重 $0<|w^{\text{in}}|\leqslant 1$.

2.5 算法实现

根据第 2.2 节中新增节点学习过程表述, 算法 IR-1 与算法 IR-2 的伪代码如下所示. 算法 IR-1 伪代码如算法 1.

算法 1. 算法 IR-1.

训练数据: $X\in\mathbb{R}^{p\times N}$, $Y\in\mathbb{R}^{q\times N}$.

初始化: 最大隐含节点数 $L_{\max}$, 候选节点池 Ξ 容量 $C_{\max}$, 输出容许误差 ε, 实数 r, $L=1$, $e_0=Z$. 输入权重范围 $[-\lambda_W,\lambda_W]^p$、反馈权重范围 $[-\lambda_{W^{\text{in}}},\lambda_{W^{\text{in}}}]$、偏置量范围 $[-\lambda_b,\lambda_b]$. Ω, $\Xi:=[]$, 输入, 反馈, 输出与偏置量参数 W, W^{in}, V, $B:=[]$.

1. **WHILE** $L<L_{\max}$ and $\|e_0\|>\varepsilon$ Do

//第 1 阶段: 隐含节点生成 (步骤 2–11)

2. **FOR** $k=1:C_{\max}$ //生成容量为 $C_{\max}$ 的候选节点池
3. 随机生成 $w_L\in[-\lambda_W,\lambda_W]^p, b_L\in[-\lambda_b,\lambda_b]$, $w_L^{\text{in}}\in[-\lambda_{W^{\text{in}}},\lambda_{W^{\text{in}}}]$,
4. 根据公式 (9) 计算 ξ_L^j $(j=1,2,\ldots,q)$
5. **IF** $\min\{\xi_L^1,\xi_L^2,\ldots,\xi_L^q\}>0$
6. 根据公式 (10) 计算 ξ_L, 保存 ξ_L 到 Ω 中.
7. 保存相应 w_L,b_L,w_L^{in} 到 Ξ 中, 转步骤 2.
8. **ELSE**
9. 转步骤 3.
10. **END IF**
11. **END FOR** //结束第 2 步循环

//第 2 阶段: 隐含节点优选 (步骤 12–15)

12. **IF** Ξ 非空
13. 取 Ω 中最大 ξ_L 所对应 Ξ 中的 w_L,b_L,w_L^{in} 分别保存到 W,B,W^{in}.
14. 若 $L=1$, 则 $S_1(X)=f_1(X)$, 否则按公式 (12) 构造 $S_L(X)$.
15. **END IF**

//第 3 阶段: 计算输出参数 (步骤 16, 17)

16. 按公式 (13) 计算并构造 v_L, 保存 v_L 到 V.

17. 计算 $\|e_L\| = \|Y - VS_L(X)\|$, 更新 $\|e_0\| := \|e_L\|, L := L + 1$.

18. **END WHILE**

19. **RETURN** V, W, W^{in}, B

算法 IR-2 伪代码见算法 2.

算法 2. 算法 IR-2.

训练数据: 同算法 IR-1.

初始化: 输出权重 $V^* := []$, 其余参数同算法 1.

1. **WHILE** $L < L_{\max}$ and $\|e_0\| > \varepsilon$ Do

//第 1、2 阶段: 与 IR-1 第 1、2 阶段相同.

//第 3 阶段: 确定输出参数 (步骤 2, 3)

2. 根据公式 (16) 计算 $V^* = YS_L(X)^+$,

3. 计算 $\|e_L^*\| = \|Y - V^*S_L(X)\|$, 更新 $\|e_0\| := \|e_L^*\|$, $L := L + 1$

4. **END WHILE**

5. **RETURN** V^*, W, W^{in}, B

第 L 个隐含节点 f_L 添加到 IRRNN 后, 利用算法 IR-1 计算输出权重的时间复杂度为 O(NL); 利用算法 IR-2 计算输出权重的时间复杂度为 O(NL^2)(采用 QR 分解计算逆矩阵). 上述分析表明, 对于单独一次计算, 算法 IR-2 的时间复杂度较大, 但是后继实验表明利用算法 IR-2 训练 IRRNN 时需要较少的隐含层节点即可实现较好的学习效果, 因此实际应用中算法 IR-2 耗时未必偏大.

在设定参数时满足: $\lambda_W > 0$, $0 < \lambda_{W^{\text{in}}} \leqslant 1$, $\lambda_b > 0$. 算法 IR-1 伪代码中"取 Ω 中最大 ξ_L 所对应 Ξ 中的 w_L, b_L, w_L^{in} 分别保存到 W, b, W^{in} 中"即是执行公式 (11) 的优选操作 (从 Ξ 中选择使最优隐含节点 f_L).

3 仿真实验

第 3.1 节验证 IRRNN 中隐含节点的动态特性. 第 3.2 节进行了动态系统逼近实验与时间序列预测实验. 第 3.3 节用 IRRNN 对水泥熟料中游离氧化钙 (f-CaO) 进行软测量, 验证本文模型解决实际问题的能力. 用本文方法与单层 RNN、单层 LSTM、ESN、GESN[33]、PDSM-ESN[34]等方法进行实验比较. 用网络的紧凑性、训练时间、网络精度 (均方误差根和 *RMSE*) 作为评价指标, *RMSE* 的计算如公式 (25) 所示:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\hat{y}(t) - y(t)} \tag{25}$$

其中, $y(t)$ 为真实值, $\hat{y}(t)$ 为网络输出值, N 为样本数量. 实验中给出了多次实验 *RMSE* 的平均值及其标准差. 实验硬件环境: Core(TM) i3-10100 CPU@3.60 GHz. 操作系统 Windows 10, 编程语言为 Python, 编程环境为 Spyder 3.

3.1 IRRNN 的动态特性仿真

首先改变激活函数 $s(t) = f(w^{\text{in}}s(t-1) + b)$ 的参数 w^{in} 与 b, 验证其动态特性与权重之间的关系. 当 $w^{\text{in}} = 0.7$, b 分别取{−0.8, −0.1, 0.1, 0.8}时, 隐含节点的唯一平衡点分别为{−0.89, −0.30, 0.30, 0.89}. 隐含节点初始状态分别为{−0.99, −0.594, −0.198, 0.198, 0.594, 0.99}时, 其零输入响应如图 3 中曲线所示. 由图 3 可知, 当 b 改变时, 平衡点位置随之改变, 但是隐含节点最终收敛到相应的稳定平衡点 (如图 3 中 A, B, C, D 所示).

当 $w^{\text{in}} = 1.7$, $b = -0.3$ 时, 隐含节点 3 个平衡点分别是{−0.959, 0.498, 0.753}. 其中, 0.753 和−0.959 是稳定平衡点, 0.498 是不稳定平衡点. 从 [−1, 1] 随机取 50 个值做为初始状态, 其输出响应如图 4 所示. 由图 4 可知: 初始状态

大于 0.498 时, 输出状态收敛到 0.753, 如图 4 中 A 所示. 初始状态小于 0.498 时, 输出状态收敛到−0.959; 如图 4 中 B 所示; 0.498 是不稳定平衡点, 如图 4 中 C 所示. 当 $w^{\text{in}} = -1.1$, $b = -0.1$ 时, 隐含节点有平衡点 0.0476 和一对震荡点{−0.5377, 0.4554}. $|f'(-0.0476)| = 1.098 > 1$, 因此平衡点 0.0476 不稳定. 当初始状态为−0.2 和 0.9 时, 隐含节点的最终输出在−0.5377 与 0.4554 两点之间产生震荡, 如图 5(a), (b) 所示. 当 $w^{\text{in}} = -1.1$, $b = -1$ 时, −0.4586 是唯一稳定平衡点. 当隐含节点的初始状态为 – 0.2 和 0.9 时, 其状态响应曲线分别如图 6(a), (b) 所示, 此时隐含节点的输出状态震荡收敛. 同理可验证当 $w^{\text{in}} > 1$ 时, 隐含节点也可能存在唯一稳定平衡点 (例如 $w^{\text{in}} = 1.5$, $b = -0.8$ 时, 唯一稳定平衡点为−0.9788). 实验表明当 $w^{\text{in}} < -1$ 时, 隐含节点能够存在唯一稳定平衡点. 上述实验表明 $|w^{\text{in}}| < 1$ 不是隐含节点有唯一稳定平衡点的必要条件, 但是设置为 $|w^{\text{in}}| < 1$ 便于 IRRNN 参数的随机学习.

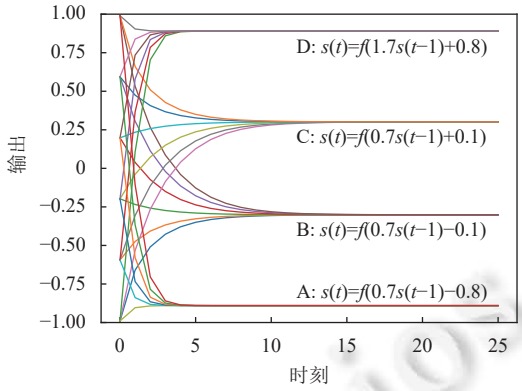


图 3 反馈权重 $w^{\text{in}} = 0.7$, 改变 b

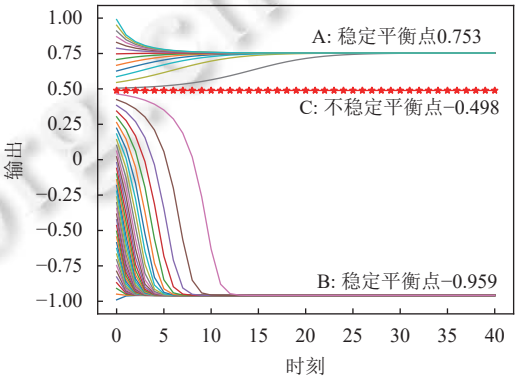
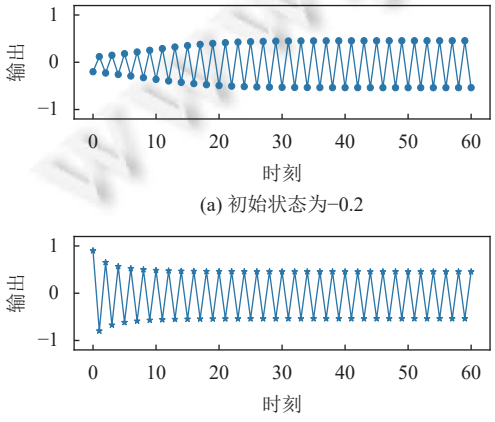


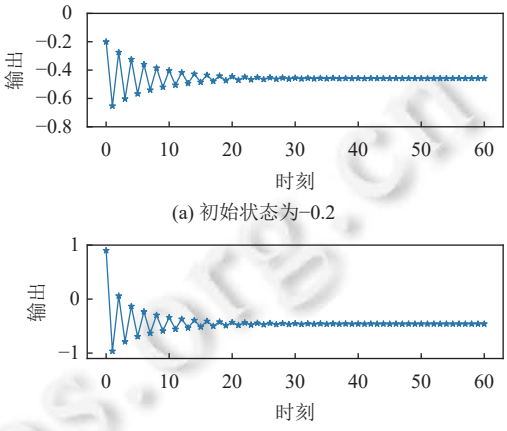
图 4 隐含节点 $s(t) = f(1.7s(t-1) - 0.3)$



(a) 初始状态为−0.2

(b) 初始状态为0.9

图 5 隐含节点 $s(t) = f(-1.1s(t-1) - 0.1)$(震荡)



(a) 初始状态为−0.2

(b) 初始状态为0.9

图 6 隐含节点 $s(t) = f(-1.1s(t-1) - 1)$(震荡收敛)

以下实验中隐含节点的初始状态分别取{−0.95, −0.57, −0.19, 0.19, 0.57, 0.95}. 当 $w = 0.8$, $w^{\text{in}} = 0.7$, $b = -0.1$ 时, 隐含节点有唯一稳定平衡点. 此时, 若输入 $x(t) = 0.3\sin(t) - 0.1 (t \geqslant 0)$, 隐含节点的输出状态响应如图 7 所示. 由图 7 可知: 随着时间推移, 隐含节点的稳态输出相同. 上述实验表明当隐含节点有唯一稳定平衡点时, 其稳态输出由输入唯一确定与初始状态无关. 当 $w = 0.8, w^{\text{in}} = 1.7, b = -0.1$ 时, 隐含节点有 3 个平衡点{−0.934, 0.144, 0.887}. 此时, 若输入 $x(t) = 0.3\sin(t) - 0.1 (t > 0)$, 隐含节点的输出状态变化如图 8 所示. 由图 8 可知, 当初始状态分别为 0.19, 0.57, 0.95 时, 隐含节点有相同的稳态输出 (如图 8 中 A 所示); 当初始状态分别为−0.95, −0.57, −0.19 时, 隐含节点有相同的稳态输出 (如图 8 中 B 所示).

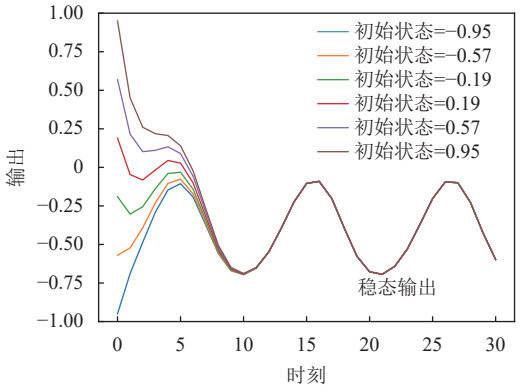


图 7 隐含节点 $s(t)=f(0.8x(t)+0.7s(t-1)-0.1)$ 输入 $x(t)=0.3\sin(t)-0.1$

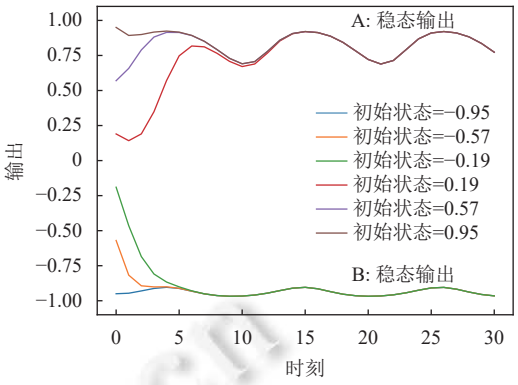


图 8 隐含节点 $s(t)=f(0.8x(t)+1.7s(t-1)-0.1)$ 输入 $x(t)=0.3\sin(t)-0.1$

保持参数固定, 输入分别为 $x(t)=\cos(t)-0.5$ 和 $x(t)=\cos(t)+0.4(t>0)$ 时, 隐含节点输出状态变化如图 9 和图 10 所示. 图 9 与图 10 表明, 若隐含节点有多个稳定平衡点, 输出有可能仅受输入信号影响而与初始状态无关.

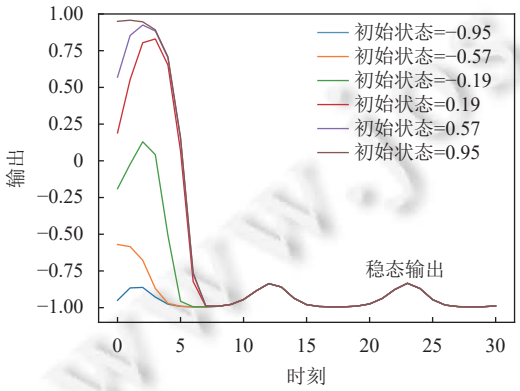


图 9 隐含节点 $s(t)=f(0.8x(t)+1.7s(t-1)-0.1)$ 输入 $x(t)=\cos(t)-0.5$

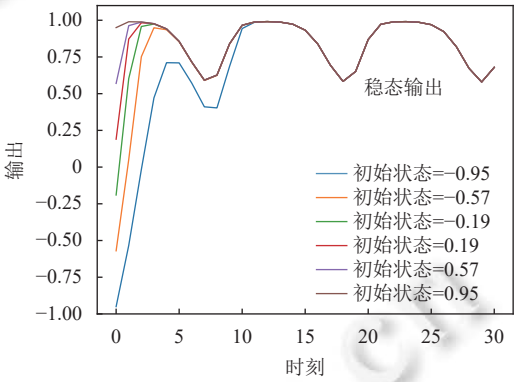


图 10 隐含节点 $s(t)=f(0.8x(t)+1.7s(t-1)-0.1)$ 输入 $x(t)=\cos(t)+0.4$

上述实验表明: 当隐含节点有多个稳定平衡点时, 其稳态输出受初始状态的影响, 当初始状态不同时, 即使相同的输入也可能产生不同的稳态输出 (如图 8 所示). 实验结果表明, 在应用中应确保 IRRNN 的每个隐含节点有唯一稳定平衡点, 避免初始状态对稳态输出的影响, 进而避免初始状态对泛化性能的影响.

3.2 IRRNN 逼近性能仿真

我们采用 4 个算例对本文提出 IRRNN 模型及其算法 IR-1、算法 IR-2 进行验证, 其中算例 1 与算例 2 是非线性系统辨识实验, 算例 3 是 Mackey Glass 系统的时间序列预测, 算例 4 是 Henon map 系统时间序列预测.

算例 1: 非线性系统辨识实验. 系统如公式 (26) 所示[38,39]:

$$y(t+1)=\frac{y(t)y(t-1)y(t-2)(y(t-2)-1)x(t-1)+x(t)}{1+y(t-1)^2+y(t-2)^2} \tag{26}$$

其中, $x(t)$, $y(t)$ 分别表示系统在 t ($t\geqslant 0$) 时刻的输入与输出. 与文献 [38] 类似, 本文的训练数据集分两部分, 前半部分为 [−1, 1] 上均匀分布随机数, 后半部分为正弦信号, 具体数据如下所示.

$$x_{\text{train}}(t)=\begin{cases} rand(-1,1), & 0\leqslant t<450 \\ 1.05\sin(t/45), & 450\leqslant t<900 \end{cases} \tag{27}$$

在模型测试阶段选用与文献 [38] 相同的数据, 如下所示:

$$x_{\text{test}}(t)=\begin{cases}\sin\left(\dfrac{\pi t}{25}\right), & t\leqslant 250\\ 1.0, & 250<t<500\\ -1.0, & 500<t<750\\ 0.3\sin\left(\dfrac{\pi t}{25}\right)+0.1\sin\left(\dfrac{\pi t}{32}\right)+0.6\sin\left(\dfrac{\pi t}{10}\right), & 750<t<1000\end{cases} \tag{28}$$

算法 IR-1 与 IR-2 的参数为: $L_{\max}=100$, $C_{\max}=10$, $\varepsilon=0.033$, $r=10^{-9}$, $\lambda_W=1$, $\lambda_{W^{\text{in}}}=0.5$, $\lambda_b=1$. 算法 IR-1 与 IR-2 的实验结果分别如图 11 与图 12 所示. 由图 11(a) 与图 12(a) 可见算法 IR-2 使 IRRNN 收敛更快, 最终隐含节点更少. 图 11 与图 12 中 (b), (c) 分别是 IRRNN 在测试集上的输出与误差, 由此可知算法 IR-2 的误差更小.

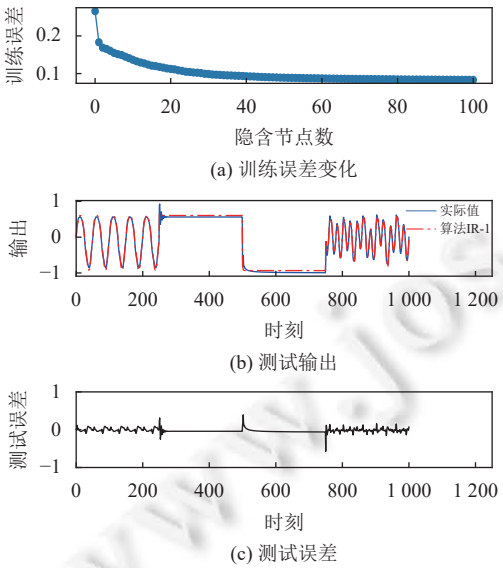


图 11 算例 1 的实验结果 (算法 IR-1)

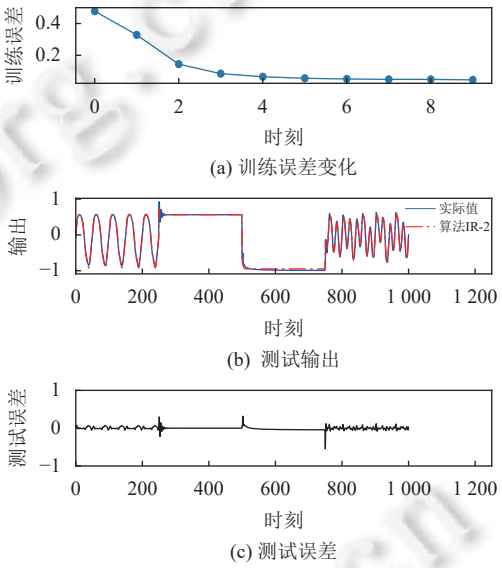


图 12 算例 1 的实验结果 (算法 IR-2)

采用 RNN, LSTM, GESN, PDSM-ESN 等模型进行实验对比, 各模型参数设定如下所述: RNN 与 LSTM 的学习速率为 0.001, epoch=500, 隐含节点数为 20, 使用 Adam 优化算法. ESN 的输入权重取值范围为 [−2, 2], 储备池大小为 50, 谱半径为 0.5, 储备池稀疏程度为 0.1. GESN 与 PDSM-ESN 的输入权重取值范围为 [−1, 1], 子储备池大小为 5, 子储备池最大数为 10 个, 子储备池谱半径为 0.5, 终止误差为 0.033. 采用上方法进行 50 次独立实验, 实验结果如表 1 所示. 由表 1 可知算法 IR-2 得到的参数量最少, 训练误差的平均值最小, 表现出良好的泛化性能. 由于 ESN 未进行网络结构的学习, 其一次性计算即可完成输出权重的学习, 因此其训练时间最短. IR-2 的训练时间小于 RNN 与 LSTM. 在该实验中算法 IR-2 得到的模型紧凑, 有较好的精度.

表 1 算例 1 的实验结果

方法	隐含节点/参数量	训练时间 (s)	训练误差 (mean, std)	测试误差 (mean, std)
RNN	20/481	19.9	0.076, 0.006	0.140, 0.012
LSTM	20/1 861	1.17	0.087, 0.009	0.124, 0.016
ESN	50/350	**0.016**	0.110, 0.014	0.137, 0.015
GESN	50/350	0.083	0.107, 0.013	0.159, 0.017
PDSM-ESN	50/350	0.205	0.085, 0.007	0.094, 0.008
算法IR-1	100/400	14.97	0.045, 0.005	0.048, 0.005
算法IR-2	**11/44**	0.86	**0.032**, **0.004**	**0.046**, **0.005**

算例 2: 非线性系统辨识. 系统如公式 (29) 所示[38,39].

$$y(t+1)=0.72y(t)+0.025y(t-1)x(t-1)+0.01x(t-2)^2+0.2x(t-3) \tag{29}$$

其中, $x(t)$, $y(t)$ 分别表示系统在 t ($t\geqslant 0$) 时刻的输入与输出. 算例 2 的训练数据、测试数据、各网络模型参数设定与算例 1 相同. 算法 IR-1 与 IR-2 的实验结果如图 13 与图 14 所示. 由图 13(a) 与图 14(a) 可知算法 IR-2 使 IRRNN 收敛更快, 隐含节点更少. 图 13 与图 14 中 (b)、(c) 分别是算法 IR-1 与 IR-2 在测试集上的输出与误差, 由图可知算法 IR-2 所得到的模型的误差更小, 隐含节点更少.

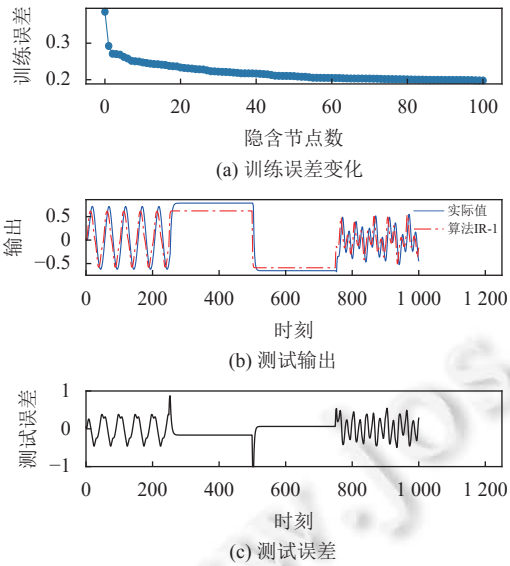


图 13 算例 2 的实验结果 (算法 IR-1)

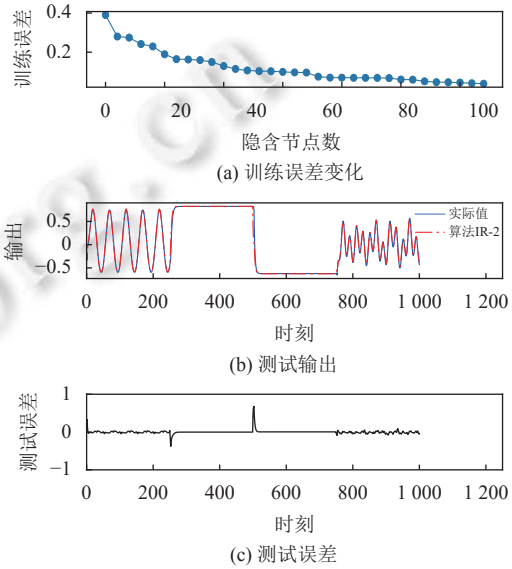


图 14 算例 2 的实验结果 (算法 IR-2)

针对算例 2 进行 50 次独立实验, 实验结果如表 2 所示. 由表 2 可知, 算法 IR-2 所需的参数量最少, 虽然 RNN 的训练误差要小于算法 IR-2, 但是算法 IR-2 的测试误差更小. 实验结果表明针对该实验算法 IR-2 有更好的泛化性能. 由表 2 可知算法 IR-2 训练时间大于 ESN、GESN、PDSM-ESN 等方法, 但是其训练时间小于 RNN. 由表 2 可知算法 IR-2 学习所得模型参数量最少, 模型结构更紧凑, 在测试集上表现出良好的泛化性能.

表 2 算例 2 的实验结果

方法	隐含节点/参数量	训练时间 (s)	训练误差 (mean, std)	测试误差 (mean, std)
RNN	20/481	19.25	**0.028**, **0.003**	0.054, 0.006
LSTM	20/1 861	1.81	0.242, 0.008	0.307, 0.031
ESN	50/350	**0.016**	0.116, 0.015	0.168, 0.017
GESN	50/350	0.088	0.086, 0.009	0.097, 0.010
PDSM-ESN	50/350	0.206	0.077, 0.005	0.086, 0.006
算法IR-1	100/400	14.78	0.181, 0.012	0.210, 0.019
算法IR-2	**32**/**128**	1.76	0.038, 0.003	**0.049**, **0.004**

算例 3: Mackey Glass 系统时间序列预测[27,34]. 该系统如公式 (30) 所示.

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=\frac{ax(t-\tau)}{1+x^n(t-\tau)}+bx(t) \tag{30}$$

Mackey Glass 系统是一个典型的混沌系统, 常用来验证模型的学习能力. 当 $\tau>16.8$ 时, Mackey Glass 系统有一个吸引子, 本文中取 $\tau=17$, 其余参数取典型值: $n=10$, $a=0.2$, $b=-0.1$, 初始值 $x(0)=0.12$. 利用系统生成 2000 个

数据, 前 1 000 个数据为训练集, 后 1 000 个为测试集. 模型训练时输入为 $[x(t), x(t+4), x(t+8), x(t+12), x(t+16)]^{\mathrm{T}}$, 输出为 $y(t)=x(t+22)$. 算法 IR-1 与 IR-2 的参数如下: $L_{\max}=200$, $C_{\max}=3$, $\varepsilon=2\mathrm{E}-4$, $r=10^{-9}$, $\lambda_W=1$, $\lambda_{W^{\mathrm{in}}}=1$, $\lambda_b=0.5$. 算法 IR-1 与 IR-2 的实验结果如图 15 与图 16 所示. 由图 15(a) 与图 16(a) 可知算法 IR-2 使 IRRNN 收敛更快, 隐含节点更少. 图 15 与图 16 中 (b)、(c) 分别是算法 IR-1 与 IR-2 在测试集上的部分输出及其误差, 由此可知算法 IR-2 的误差更小.

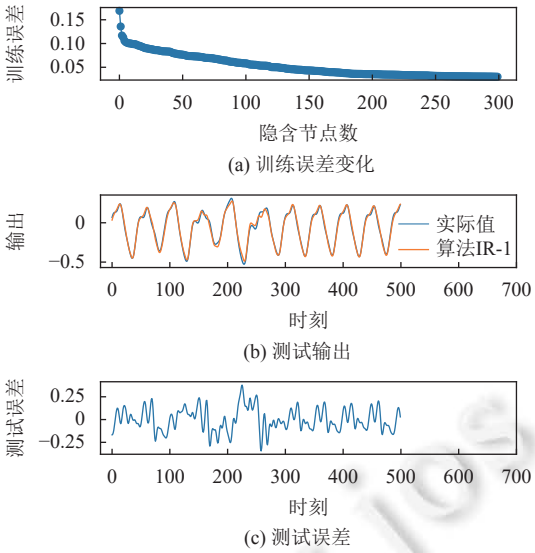


图 15 算例 3 的实验结果 (算法 IR-1)

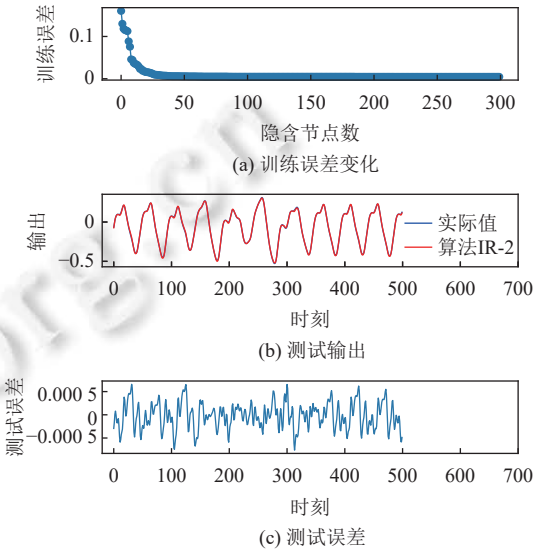


图 16 算例 3 的实验结果 (算法 IR-2)

分别采用 RNN, LSTM, ESN, GESN, PDSM-ESN 等方法进行 50 次独立实验, 实验结果如表 3 所示. 其中 RNN 与 LSTM 的学习速率为 0.001, epoch=500, 隐含节点为 150, 用 Adam 优化算法. ESN 输入权重取值范围为 [−1, 1], 储备池大小为 250, 谱半径为 0.5, 储备池稀疏程度为 0.1. GESN 与 PDSM-ESN 的输入权重取值范围为 [−1, 1], 子储备池大小为 5, 子储备池最大数为 50, 终止误差为 0.000 2. GESN 的谱半径为 0.5, PDSM-ESN 子储备池的谱半径小于 1. 由表 3 可知, 针对该实验 LSTM 有最好的预测效果, 但是其参数量要远多于算法 IR-2, 且 IR-2 与 LSTM 测试集上的误差相差不大. 结果表明 IR-2 得到的模型结构紧凑, 有较好的精度.

表 3 算例 3 的实验结果

方法	节点数/参数个数	训练时间 (s)	训练误差 (mean, std)	测试误差 (mean, std)
RNN	150/23 701	25.12	3.6E–4, 4.7E–5	6.6E–4, 8.2E–5
LSTM	150/94 351	4.42	**3.2E–4**, **9.7E–6**	**4.2E–4**, **7.6E–5**
ESN	250/7 750	**0.198**	4.5E–4, 5.1E–5	5.6E–4, 6.2E–5
GESN	250/2 750	1.92	4.3E–4, 3.9E–5	4.8E–4, 5.5E–5
PDSM-ESN	250/2 750	3.11	3.8E–4, 3.2E–5	4.4E–4, 5.2E–6
算法IR-1	300/2 400	10.25	2.1E–2, 3.2E–3	3.6E–2, 4.9E–5
算法IR-2	**200/1 600**	11.44	3.6E–4, 6.5E–5	4.6E–4, 7.8E–5

算例 4: Henon map 系统时间序列预测. Henon map 系统如公式 (31) 所示.

$$\begin{cases} x(t+1)=y(t)-ax(t)^2+1 \\ y(t+1)=bx(t) \end{cases} \tag{31}$$

Henon map 系统是一个具有混沌行为的离散动态系统, 该系统把点 $[x(t), y(t)]^{\mathrm{T}}$ 映射为点 $[x(t+1), y(t+1)]^{\mathrm{T}}$, 当

参数 $a=1.4, b=0.3$ 时系统有唯一吸引子. 取初始值 $x(0)=0, y(0)=0$ 由该系统生成 1 200 组数据. 采用前 1 000 组数据作为训练集, 后 200 组数据作为测试集. 实验中采用 $[x(t), y(t)]$ 预测 $x(t+1)$. 算法 IR-1 与 IR-2 的参数设定与算例 3 相同, 算法 IR-1 与算法 IR-2 的实验结果如图 17 与图 18 所示. 由图 17(a) 与图 18(a) 可知算法 IR-2 使 IRRNN 收敛更快, 隐含节点更少. 图 17 与图 18 中 (b)、(c) 分别是算法 IR-1 与 IR-2 在测试集上的输出与误差, 由图可知算法 IR-2 得到的模型误差更小.

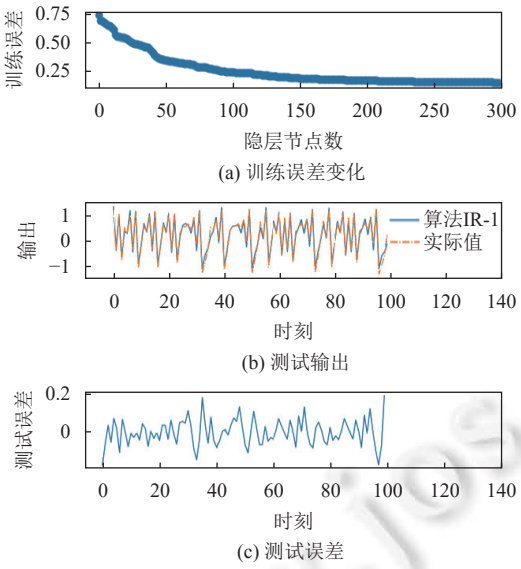


图 17 算例 4 的实验结果 (算法 IR-1)

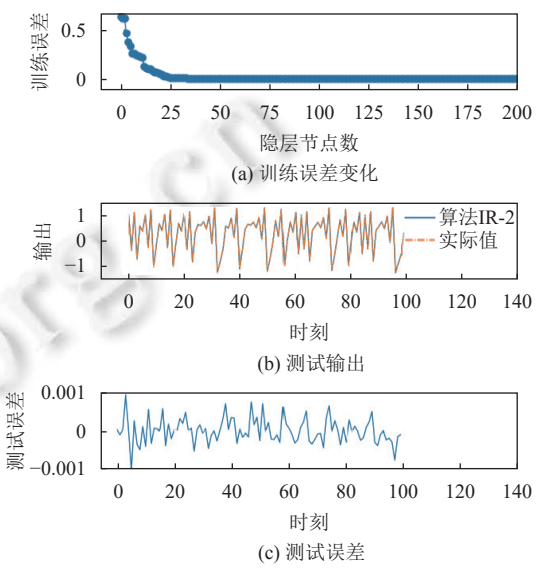


图 18 算例 4 的实验结果 (算法 IR-2)

分别用 LSTM、RNN、ESN、GESN、PDSM-ESN 等方法进行 50 次独立实验, 各网络参数设定与算例 3 相同, 实验结果如表 4 所示. 由表 4 可知针对 Henon map 的预测实验中, PDSM-ESN 具有最好的学习效果, 而算法 IR-2 的学习效果与之相近, 但是 IR-2 的参数量更少.

表 4 算例 4 的实验结果

方法	节点数/参数个数	训练时间 (s)	训练误差 (mean, std)	测试误差 (mean, std)
RNN	200/40 801	26.12	1.1E–2, 1.8E–4	1.4E–2, 2.8E–4
LSTM	200/162 601	5.89	1.7E–2, 2.2E–4	2.5E–2, 2.2E–4
ESN	250/7 000	**0.18**	4.5E–4, 6.0E–5	5.6E–4, 9.8E–5
GESN	250/2 000	1.85	4.4E–4, 9.7E–6	5.5E–4, 6.7E–5
PDSM-ESN	250/2 000	3.09	**2.0E–4**, **7.5E–6**	**2.3E–4**, **2.3E–5**
算法IR-1	300/1 500	10.05	2.2E–2, 3.6E–4	2.4E–2, 5.6E–4
算法IR-2	**200/1 000**	10.02	2.5E–4, 6.7E–5	3.7E–4, 9.3E–5

3.3 水泥熟料中 f-CaO 软测量

水泥熟料中游离氧化钙 (f-CaO) 含量对水泥质量有重要影响, f-CaO 含量过高会使水泥安定性变差, 含量过低会使水泥强度不够、生产过程能耗大. 实践表明水泥熟料中 f-CaO 含量在 0.5%–1.5% 之间较为合理. 在实际生产中 f-CaO 的含量不能直接测量, 需要采用离线化验方法得到, 化验时间滞后大, 这给系统的反馈控制与优化造成不便, 因此实现 f-CaO 实时在线软测量对于提高水泥质量有重要意义.

水泥熟料中 f-CaO 的含量主要受水泥生料煅烧时间与煅烧温度等因素的影响. 水泥回转窑的主窑转速 $x_1(t)$、主电机电流 $x_2(t)$、窑头罩温度 $x_3(t)$、窑尾烟室温度 $x_4(t)$、二次风温度 $x_5(t)$ 等物理量能够反映煅烧温度和煅烧时间, 取上述变量作为辅助变量. 水泥回转窑内部发生复杂的化学物理变化是一个动态过程, 因此 f-CaO 的含量 $y(t)$

表示为一个动态系统的形式 $y(t)=g(y(t-1),x(t))$, 其中 $x(t)=[x_1(t),x_2(t),x_3(t),x_4(t),x_5(t)]^{\mathrm{T}}$. 考虑到实际工程条件, 每隔 30 min 采集水泥熟料进行化验, 再经过线性插值得到间隔为 10 min 的 f-CaO 含量的数据, 同时按照 10 min 间隔采集对应的辅助变量. 经数据采集和插值处理后得到 f-CaO 含量正常、偏大、偏小的数据分别是 1 000 组、200 组、200 组. 从正常、偏低、偏大的数据中按照时间先后顺序分别取相邻的 100 组、35 组、40 组数据作为测试集, 其余的数据作为训练集.

采用归一化之后的数据对单层 LSTM, 单层 RNN, ESN, GESN 与本文方法进行训练与测试. LSTM 与 RNN 的隐含节点个数都设定为 100, epoch=300, 采用 Adam 算法优化参数, 学习速率都为 0.000 5. ESN, GESN 与 PDSM-ESN 的参数设置与算例 3 相同. 在算法 IR-1 与 IR-2 中 $L_{\max}=200$, $C_{\max}=10$, $\varepsilon=0.015$, $r=1\times10^{-8}$, $\lambda_W=1.5$, $\lambda_{W^{\mathrm{in}}}=0.8$, $\lambda_b=0.2$. 算法 IR-1 与 IR-2 的实验结果如图 19 和图 20 所示. 由图 19(a) 与图 20(a) 可知, 随隐含节点增多 IRRNN 的输出误差逐步减少. 图 19(b) 与图 20(b) 是测试结果, 由图可知 IR-2 算法得到的 IRRNN 更紧凑、精度更高. 采用上述参数进行 50 次实验, 实验结果如表 5 所示. 由表 5 可知, 虽然算法 IR-2 的训练时间不是最短的, 但是本文所提出的算法 IR-2 在训练集与测试集上的效果要优于其他方法, 同时算法 IR-2 得到的网络参数量更少. 实验结果表明算法 IR-2 能够较好地实现水泥熟料中 f-CaO 的软测量.

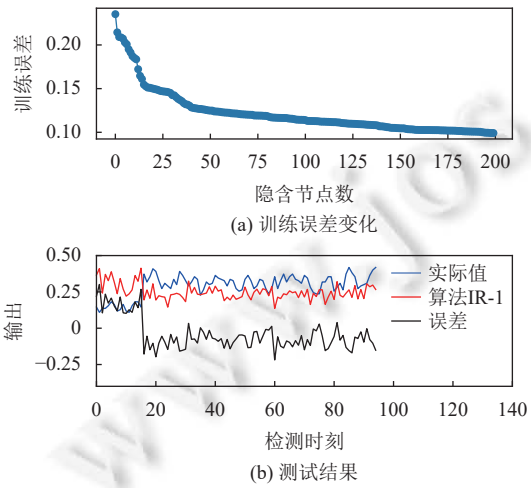


图 19 算法 IR-1 结果

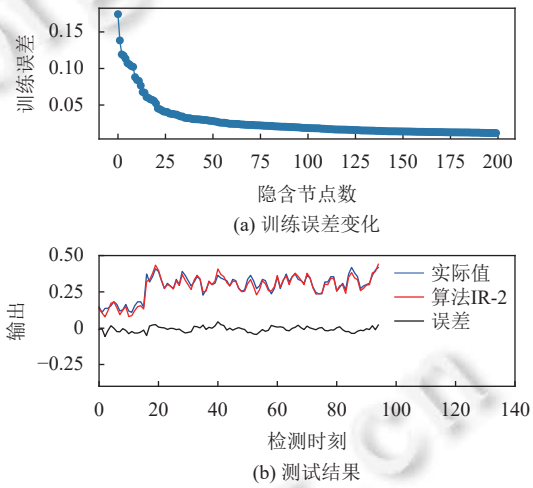


图 20 算法 IR-2 结果

表 5 不同方法软测量结果

测量方法	节点数/参数个数	训练时间 (s)	训练误差 (mean, std)	测试误差 (mean, std)
LSTM	100/42 901	3.25	0.037 2, 0.004 2	0.081 5, 0.006 7
RNN	100/10 801	27.15	0.029 9, 0.003 1	0.035 2, 0.003 6
ESN	200/5 200	**0.167**	0.018 0, 0.001 5	0.025 6, 0.004 5
GESN	200/1 700	0.353	0.010 2, 0.001 2	0.026 1, 0.004 4
PDSM-ESN	200/2 200	3.389	0.009 8, 0.001 2	0.015 4, 0.004 7
算法IR-1	**200/1 600**	18.461	0.038 2, 0.001 3	0.040 9, 0.003 7
算法IR-2	**200/1 600**	16.41	**0.009 2**, **0.000 6**	**0.012 5**, **0.002 4**

3.4 讨 论

IRRNN 隐含节点的动态特性较为复杂, 其隐含节点可能存在多个平衡点, 也可能存在震荡. 第 3.1 节仿真结果表明 IRRNN 稳定平衡点的个数对网络的性能有重要影响. 当 IRRNN 有唯一稳定平衡点时, 隐含节点稳态输出由输入决定而与初始状态无关. 当隐含节点有多个平衡点时, 其稳态输出可能受初始状态与输入的共同影响. 在随机学习过程中设置 $0<|w^{\mathrm{in}}|\leqslant 1$ 确保 IRRNN 有全局唯一平衡点, 避免局部平衡点对网络的泛化性能的影响.

第 3.2 节与第 3.3 节对算法 IR-1 与 IR-2 进行了验证. 算法 IR-1 采用了局部优化的思想, 当新增一个隐含节点时仅需计算新增节点的输出权重, 而其余节点的权重保持不变, 因此对于单个新增隐含节点, IR-1 的计算量小, 但是算法 IR-1 使 IRRNN 收敛慢, 最终 IRRNN 的误差较大. 算法 IR-2 采用了全局优化的思想, 当新增一个隐含节点时需要更新所有隐含节点的输出权重, 因此当新增一个节点时, 算法 IR-2 的计算量偏大, 但是 IR-2 使 IRRNN 收敛快, 最终网络的误差小. 在实际应用中, 当需要快速建模但是对模型的精度要求不高时, 可以选用算法 IR-1; 当需要建立精确模型时, 可采用算法 IR-2. 从本文的实验结果来看, 与其他方法相比算法 IR-1 得到的结果没有优势, 但是从理论上分析算法 IR-1 是收敛的, 并且算法 IR-1 是算法 IR-2 收敛性分析的基础, 因此在本文中对算法 IR-1 进行了实验对比.

算例 1 与算例 2 的仿真结果与工业应用实验结果显示算法 IR-2 有良好的泛化性能. GESN, PDSM-ESN 及本文所提出的 IRRNN 都实现了循环神经网络参数与结构的学习, 但上述网络存在明显区别. 首先, IRRNN 与 GESN 都采用了增量构造方法, 但是 GESN 采用完全随机的参数, 未对随机参数优选, 而 IRRNN 采用了约束机制与候选节点池策略对随机参数进行优选, 使得 IRRNN 参数更少, 结构更紧凑. PDSM-ESN 与 IRRNN 都进行了网络的结构优化, 但是 PDSM-ESN 内部结构不清晰, 各个隐含节点的耦合关系存在随机性, 而 IRRNN 的数量更少, 内部结构更清晰.

理论分析与仿真实验表明 IRRNN 具有以下特点: (1) IRRNN 未采用误差反传方法进行学习, 网络的学习速度较快, 有较好的精度; (2) 在 IRRNN 的增量学习过程中能够确保网络有全局唯一稳定平衡点, 避免初始状态对稳态输出的影响, 使其具有较好的泛化能力; (3) IRRNN 把一个复杂系统表示为多个简单非线性系统的叠加, 网络参数更少, 结构更加紧凑, 内部结构清晰.

4 结 论

本文提出的 IRRNN 实现了循环神经网络结构的增量构造与参数的随机学习. 在 IRRNN 的学习过程中建立了随机学习的约束机制, 实现了对随机节点的初步筛选, 再用候选节点池策略实现对隐含节点的优选. 从局部优化与全局优化角度考虑, 分别给出了 IRRNN 的两种学习算法, 即算法 IR-1 与算法 IR-2. 在理论方面, 证明了 IRRNN 对动态系统有万能逼近能力, 分析了 IRRNN 的稳定性、平衡点数量对输出的影响, 给出了其稳定性判别方法. 系统逼近仿真实验与工业软测量实验表明 IRRNN 的结构紧凑、精度较高. 本文所提出的 IRRNN 还存在以下待解决的问题. (1) 算法 IR-2 中权重矩阵的计算利用了广义逆矩阵, 当训练样本数据量较大、隐含节点较多时计算广义逆矩阵时间开销大, 网络的学习速度变慢, 因此如何应对大规模样本需进一步深入研究. (2) IRRNN 是一种短期记忆网络, 如何设计具有长时间记忆的 IRRNN 也需要进一步研究. (3) 实验表明本文所提出的方法与回声状态类的网络在不同的对象上表现出不同的效果, 我们将继续研究在数据受干扰、数据不完整等情况下的建模方法.

References:

[1] Zhang ZJ, Ren XH, Xie JL, Luo YM. A novel swarm exploring varying parameter recurrent neural network for solving non-convex nonlinear programming. IEEE Trans. on Neural Networks and Learning Systems, 2023, 35(9): 12642–12652. [doi: 10.1109/TNNLS.2023.3263975]

[2] Liu Y, Yang PF, Zhang LJ, Wu ZL, Feng Y. Survey on robustness verification of feed forward neural networks and recurrent neural networks. Ruan Jian Xue Bao/Journal of Software, 2023, 34(7): 3134–3166 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6863.htm [doi: 10.13328/j.cnki.jos.006863]

[3] Hou HS, Luo C, Zhang H, Wu GC. Frequency domain approach to the critical step size of discrete-time recurrent neural networks. Nonlinear Dynamics, 2023, 111(9): 8467–8476. [doi: 10.1007/s11071-023-08278-0]

[4] Chen JX, Hao W, Zhang PW, Min CD, Li YC. Emotion classification of spatiotemporal EEG features using hybrid neural networks. Ruan Jian Xue Bao/Journal of Software, 2021, 32(12): 3869–3883 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6123.htm [doi: 10.13328/j.cnki.jos.006123]

[5] Wang YB, Wu HX, Zhang JJ, Gao ZF, Wang JM, Yu PS, Long MS. PredRNN: A recurrent neural network for spatiotemporal predictive learning. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2023, 45(2): 2208–2225. [doi: 10.1109/TPAMI.2022.3165153]

[6] Horn J, De Jesus O, Hagan MT. Spurious valleys in the error surface of recurrent networks—Analysis and avoidance. IEEE Trans. on Neural Networks, 2009, 20(4): 686–700. [doi: 10.1109/TNN.2008.2012257]

[7] Werbos PJ. Backpropagation through time: What it does and how to do it. Proc. of the IEEE, 1990, 78(10): 1550–1560. [doi: 10.1109/5.58337]

[8] Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1989, 1(2): 270–280. [doi: 10.1162/neco.1989.1.2.270]

[9] Puskorius GV, Feldkamp LA. Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. IEEE Trans. on Neural Networks, 1994, 5(2): 279–297. [doi: 10.1109/72.279191]

[10] Jaeger H. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. Technical Report, 159, German National Research Center for Information Technology, 2002. https://www.ai.rug.nl/minds/uploads/ESNTutorialRev.pdf

[11] Liu JW, Song ZY. Overview of recurrent neural networks. Control and Decision, 2022, 37(11): 2753–2768 (in Chinese with English abstract). [doi: 10.13195/j.kzyjc.2021.1241]

[12] Ku CC, Lee KY. Diagonal recurrent neural networks for dynamic systems control. IEEE Trans. on Neural Networks, 1995, 6(1): 144–156. [doi: 10.1109/72.363441]

[13] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

[14] Cho K, van Merriënboer B, Gulcehre G, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Doha: Association for Computational Linguistics, 2014. 1724–1734. [doi: 10.3115/v1/d14-1179]

[15] Liu J, Shahroudy A, Xu D, Wang G. Spatio-temporal LSTM with trust gates for 3D human action recognition. In: Proc. of the 14th European Conf. on Computer Vision (ECCV). Amsterdam: Springer, 2016. 816–833. [doi: 10.1007/978-3-319-46487-9_50]

[16] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 2005, 18(5–6): 602–610. [doi: 10.1016/j.neunet.2005.06.042]

[17] Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks. In: Proc. of the 60th IEEE Int'l Midwest Symp. on Circuits and Systems (MWSCAS). Boston: IEEE, 2017. 1597–1600. [doi: 10.1109/MWSCAS.2017.8053243]

[18] Scardapane S, Wang DH. Randomness in neural networks: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2017, 7(2): e1200. [doi: 10.1002/widm.1200]

[19] Cao WP, Wang XZ, Ming Z, Gao JZ. A review on neural networks with random weights. Neurocomputing, 2018, 275: 278–287. [doi: 10.1016/j.neucom.2017.08.040]

[20] Pao YH, Takefuji Y. Functional-link net computing: Theory, system architecture, and functionalities. Computer, 1992, 25(5): 76–79. [doi: 10.1109/2.144401]

[21] Zhang QS, Tsang ECC, He Q, Guo YT. Ensemble of kernel extreme learning machine based elimination optimization for multi-label classification. Knowledge-based Systems, 2023, 278: 110817. [doi: 10.1016/j.knosys.2023.110817]

[22] Gong XR, Zhang T, Chen CLP, Liu ZL. Research review for broad learning system: Algorithms, theory, and applications. IEEE Trans. on Cybernetics, 2022, 52(9): 8922–8950. [doi: 10.1109/TCYB.2021.3061094]

[23] Wang DH, Li M. Stochastic configuration networks: Fundamentals and algorithms. IEEE Trans. on Cybernetics, 2017, 47(10): 3466–3479. [doi: 10.1109/TCYB.2017.2734043]

[24] Dai W, Li DP, Zhou P, Chai TY. Stochastic configuration networks with block increments for data modeling in process industries. Information Sciences, 2019, 484: 367–386. [doi: 10.1016/j.ins.2019.01.062]

[25] Dai W, Li DP, Yang CY, Ma XP. A model and data hybrid parallel learning method for stochastic configuration networks. Acta Automatica Sinica, 2021, 47(10): 2427–2437 (in Chinese with English abstract). [doi: 10.16383/j.aas.c190411]

[26] Zhang CL, Ding SF, Guo LL, Zhang J. Research progress on stochastic configuration network. Ruan Jian Xue Bao/Journal of Software, 2024, 35(5): 2379–2399 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6804.htm [doi: 10.13328/j.cnki.jos.006804]

[27] Jaeger H, Lukoševičius M, Popovici D, Siewert U. Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks, 2007, 20(3): 335–352. [doi: 10.1016/j.neunet.2007.04.016]

[28] Yildiz IB, Jaeger H, Kiebel SJ. Re-visiting the echo state property. Neural Networks, 2012, 35: 1–9. [doi: 10.1016/j.neunet.2012.07.005]

[29] Wang HS, Yan XF. Optimizing the echo state network with a binary particle swarm optimization algorithm. Knowledge-based Systems, 2015, 86: 182–193. [doi: 10.1016/j.knosys.2015.06.003]

[30] Dutoit X, Schrauwen B, van Campenhout J, Stroobandt D, van Brussel H, Nuttin M. Pruning and regularization in reservoir computing. Neurocomputing, 2009, 72(7–9): 1534–1546. [doi: 10.1016/j.neucom.2008.12.020]

[31] Han M, Ren WJ, Xu ML. An improved echo state network via L_1-norm regularization. Acta Automatica Sinica, 2014, 40(11): 2428–2435 (in Chinese with English abstract). [doi: 10.3724/SP.J.1004.2014.02428]

[32] Wang L, Qiao JF, Yang CL, Zhu XX. Pruning algorithm for modular echo state network based on sensitivity analysis. Acta Automatica Sinica, 2019, 45(6): 1136–1145 (in Chinese with English abstract). [doi: 10.16383/j.aas.c180288]

[33] Qiao JF, Li FJ, Han HG, Li WJ. Growing echo-state network with multiple subreservoirs. IEEE Trans. on Neural Networks and Learning Systems, 2017, 28(2): 391–404. [doi: 10.1109/TNNLS.2016.2514275]

[34] Wang L, Su Z, Qiao JF, Deng F. A pseudo-inverse decomposition-based self-organizing modular echo state network for time series prediction. Applied Soft Computing, 2022, 116: 108317. [doi: 10.1016/j.asoc.2021.108317]

[35] Kumar R, Srivastava S, Gupta JRP. Diagonal recurrent neural network based adaptive control of nonlinear dynamical systems using Lyapunov stability criterion. ISA Trans., 2017, 67: 407–427. [doi: 10.1016/j.isatra.2017.01.022]

[36] Peng XW, Gao HL. Improved diagonal recursion neural network and PI control of permanent magnet synchronous motor. Electric Machines and Control, 2019, 23(4): 126–132 (in Chinese with English abstract). [doi: 10.15938/j.emc.2019.04.016]

[37] Li M, Wang DH. Insights into randomized algorithms for neural networks: Practical issues and common pitfalls. Information Sciences, 2017, 382–383: 170–178. [doi: 10.1016/j.ins.2016.12.007]

[38] Zhao HQ, Zhang JS. Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network. Neurocomputing, 2009, 72(13–15): 3046–3054. [doi: 10.1016/j.neucom.2009.04.001]

[39] Kumar R, Srivastava S. A novel dynamic recurrent functional link neural network-based identification of nonlinear systems using Lyapunov stability analysis. Neural Computing and Applications, 2021, 33(13): 7875–7892. [doi: 10.1007/s00521-020-05526-x]

附中文参考文献:

[2] 刘颖, 杨鹏飞, 张立军, 吴志林, 冯元. 前馈神经网络和循环神经网络的鲁棒性验证综述. 软件学报, 2023, 34(7): 3134–3166. http://www.jos.org.cn/1000-9825/6863.htm [doi: 10.13328/j.cnki.jos.006863]

[4] 陈景霞, 郝为, 张鹏伟, 闵重丹, 李玥辰. 基于混合神经网络的脑电时空特征情感分类. 软件学报, 2021, 32(12): 3869–3883. http://www.jos.org.cn/1000-9825/6123.htm [doi: 10.13328/j.cnki.jos.006123]

[11] 刘建伟, 宋志妍. 循环神经网络研究综述. 控制与决策, 2022, 37(11): 2753–2768. [doi: 10.13195/j.kzyjc.2021.1241]

[25] 代伟, 李德鹏, 杨春雨, 马小平. 一种随机配置网络的模型与数据混合并行学习方法. 自动化学报, 2021, 47(10): 2427–2437. [doi: 10.16383/j.aas.c190411]

[26] 张成龙, 丁世飞, 郭丽丽, 张健. 随机配置网络研究进展. 软件学报, 2024, 35(5): 2379–2399. http://www.jos.org.cn/1000-9825/6804.htm [doi: 10.13328/j.cnki.jos.006804]

[31] 韩敏, 任伟杰, 许美玲. 一种基于 L_1 范数正则化的回声状态网络. 自动化学报, 2014, 40(11): 2428–2435. [doi: 10.3724/SP.J.1004.2014.02428]

[32] 王磊, 乔俊飞, 杨翠丽, 朱心新. 基于灵敏度分析的模块化回声状态网络修剪算法. 自动化学报, 2019, 45(6): 1136–1145. [doi: 10.16383/j.aas.c180288]

[36] 彭熙伟, 高瀚林. 永磁同步电机的改进对角递归神经网络 PI 控制策略. 电机与控制学报, 2019, 23(4): 126–132. [doi: 10.15938/j.emc.2019.04.016]

附录 A

定理 3. 每个隐含节点必定有平衡点, 平衡点个数可能是 1 个、2 个或者 3 个.

证明: 激活函数 $s(t)=f(w^{\text{in}}s(t-1)+b)$. $w^{\text{in}}=0$ 时是静态系统, 因此仅考虑 $w^{\text{in}}\neq 0$. 令 l_1 表示函数 $s(t)=s(t-1)$ 的几何图形. 当 $w^{\text{in}}>0$ 与 $w^{\text{in}}<0$ 时, 分别用 l_2 与 l_3 表示激活函数 $s(t)=f(w^{\text{in}}s(t-1)+b)$ 对应的曲线 $(s(t-1)\in(-1,1))$, 如图 A1 所示. 当隐含节点处于平衡状态点时, 其输入输出相等, 由图 A1 可知平衡点为 l_1 与 l_2 两条线的(或 l_1 与 l_3) 交点. $w^{\text{in}}<0$ 时根据函数单调性易知 l_1 与 l_3 仅有一个交点. $w^{\text{in}}>0$ 时易知 l_1 与 l_2 可能有 1 个、2 个或者 3 个交点. 证毕.

定理 4. 如果 $f(f(s))=s$ 的解不是隐含节点平衡点, 则必定是隐含节点的震荡点.

证明: 假设 a 是方程 $f(f(s))=s$ 的解, 即 $f(f(a))=a$. 若 $f(a)=a$, 则:

$$f(f(a))=f(a)=a \tag{A1}$$

a 是激活函数 f 的平衡点. 若 $f(a)=b\neq a$, 则:

$$f(f(a))=f(b)=a \tag{A2}$$

又有 $f(a)=b$, 因此 a, b 构成一组震荡点. 综上所述 $f(f(s))=s$ 的解是隐含节点的平衡点或震荡点. 证毕.

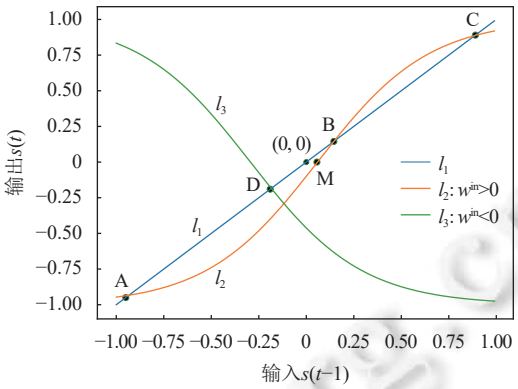


图 A1 隐含节点的几何表示

定理 5. $f'(s_e)$ 是激活函数 f 在平衡点 s_e 处的导数, 当 $|f'(s_e)|<1$ 时, s_e 是稳定平衡点; 当 $|f'(s_e)|>1$ 时, s_e 是不稳定平衡点; 当 $|f'(s_e)|=1$ 时, 不能用 $|f'(s_e)|$ 判定平衡点 s_e 的稳定性.

证明: 若 s_e 是平衡点, 则 $f(s_e)=s_e$, 在 s_e 处线性化得:

$$s(t)\approx f(s_e)+f'(s_e)(s(t-1)-s_e)=s_e+f'(s_e)(s(t-1)-s_e) \tag{A3}$$

令 $z(t)=s(t)-s_e, z(t-1)=s(t-1)-s_e$, 得:

$$z(t)=f'(s_e)z(t-1) \tag{A4}$$

由离散系统稳定性判别方法可知, 当 $|f'(s_e)|<1$ 时, s_e 是稳定平衡点, 当 $|f'(s_e)|>1$ 时, s_e 是不稳定平衡点; 当 $|f'(s_e)|=1$ 时, 不能用 $|f'(s_e)|$ 判定平衡点 s_e 的稳定性. 证毕.

定理 6. 若 $s_{e1}<s_{e2}<s_{e3}$ 是隐含节点的 3 个平衡点, 则 s_{e2} 是不稳定平衡点, s_{e1} 和 s_{e3} 是两个稳定平衡点, s_{e1} 和 s_{e1} 的稳定域分别为 $(-1,s_{e2})$, $(s_{e2},1)$.

证明: 由定理 3 可知 $w^{\text{in}}>1$ 时隐含节点可能有 3 个平衡点, 如图 A1 中 l_1 与 l_2 有 3 个交点 A,B,C. 令 $s_{e1}=A$, $s_{e2}=B$, $s_{e3}=C$, 由图 A1 易知:

$$0<f'(s_{e1})<1,\ \ 1<f'(s_{e2}),\ \ 0<f'(s_{e3})<1 \tag{A5}$$

根据定理 5 可知: s_{e1} 和 s_{e3} 是稳定的平衡点 s_{e2} 是不稳定的平衡点.

设在平衡点 s_{e2} 处受到较小的干扰 δ $(\delta>0)$, 则:

$$f(s_{e2}+\delta)\approx f(s_{e2})+f'(s_{e2})\delta>s_{e2}+\delta \tag{A6}$$

由 f 是单调递增函数, 可知:

$$f(f(s_{e2}+\delta))>f(s_{e2}+\delta) \tag{A7}$$

即随着时间推移激活函数 f 的输出逐步增大, 状态远离平衡点 s_{e2}. 由于 $s_{e2}<s<1$ 时 s_{e3} 是唯一平衡点, 所以 s_{e3} 的稳定域为 $(s_{e2},1)$. 由于

$$f(s_{e2}-\delta)\approx f(s_{e2})-f'(s_{e2})\delta<s_{e2}-\delta \tag{A8}$$

由 f 是单调递增函数, 可知:

$$f(f(s_{e2}-\delta))<f(s_{e2}-\delta) \tag{A9}$$

即随着时间推移激活函数 f 的输出逐步远离平衡点 s_{e2}. 由于 $-1<s<s_{e2}$ 时 x_{e1} 是唯一稳定平衡点, 因此 x_{e1} 的稳定域为 $(-1,s_{e2})$. 证毕.

定理 7. 当 IRRNN 中任意隐含节点的反馈权重 $0<|w^{\text{in}}|\leqslant 1$ 时, IRRNN 有唯一稳定平衡点, 且 IRRNN 的稳态

输出由输入唯一确定与隐含节点的初始状态无关.

证明: 由于 IRRNN 的输出是隐含状态的线性叠加, 因此若能证明任意隐含节点的反馈权重 $0<|w^{\text{in}}|\leqslant 1$ 时, 该隐含节点有唯一稳定平衡点, 且稳态输出与初始状态无关, 则可证明该定理.

w, w^{in}, b 表示任意隐含节点的输入权重, 反馈权重与偏置量, 隐含节点的状态记为 $s(t)\in\mathbb{R}(t\geqslant 0)$, 输入为 $x(t)$. 设该隐含节点的任意两个初始状态 $s^{\text{I}}(0)$, $s^{\text{II}}(0)$ 所对应的输出状态分别为 $s^{\text{I}}(t)$, $s^{\text{II}}(t)(t\geqslant 1)$, 由拉格朗日中值定理可知:

$$\begin{aligned}s^{\text{II}}(t)-s^{\text{I}}(t)&=f(wx(t)+w^{\text{in}}s^{\text{II}}(t-1)+b)-f(wx(t)+w^{\text{in}}s^{\text{I}}(t-1)+b)\\&=f'(\eta_t)(s^{\text{II}}(t-1)-s^{\text{I}}(t-1))\end{aligned}\tag{A10}$$

可得:

$$\begin{cases}s^{\text{II}}(1)-s^{\text{I}}(1) &=f'(\eta_1)(s^{\text{II}}(0)-s^{\text{I}}(0))\\ s^{\text{II}}(2)-s^{\text{I}}(2) &=f'(\eta_2)(s^{\text{II}}(1)-s^{\text{I}}(1))=f'(\eta_2)f'(\eta_1)(s^{\text{II}}(0)-s^{\text{I}}(0))\\ &\cdots\\ s^{\text{II}}(t)-s^{\text{I}}(t) &=\left(\sum_{i=1}^{t}f'(\eta_i)\right)(s^{\text{II}}(0)-s^{\text{I}}(0))\end{cases}\tag{A11}$$

若 $|w^{\text{in}}|<1$, 则对于任意 η_t, 有:

$$|f'(\eta_t)|=|(1-f(\eta_t)^2)w^{\text{in}}|<1\tag{A12}$$

因此,

$$\lim_{t\to\infty}\left(\sum_{i=1}^{t}f'(\eta_i)\right)=0\tag{A13}$$

即 $t\to\infty$ 时 $s^{\text{II}}(t)-s^{\text{I}}(t)=0$.

当 $|w^{\text{in}}|=1$ 时, 激活函数为 $s(t)=f(s(t-1)+b)$. 当 $s(t-1)=-b$ 时, $|f'(-b)|=|(1-f(-b)^2)|=1$ 取最大值. 此时曲线上任意两个点 $(s^{\text{II}}(t-1),s^{\text{II}}(t))$ 与 $(s^{\text{I}}(t-1),s^{\text{I}}(t))$ 之间的斜率的绝对值小于. 仅当 $s^{\text{II}}(t-1)\to -b$, $s^{\text{I}}(t)=-b$(或 $s^{\text{II}}(t-1)=-b$, $s^{\text{I}}(t)\to -b$) 时, 有:

$$|f'(-b)|=\left|\lim_{\substack{s^{\text{II}}(t-1)\to -b\\ s^{\text{I}}(t-1)=-b}}\frac{f(s^{\text{I}}(t-1))-f(s^{\text{II}}(t-1))}{s^{\text{I}}(t-1)-s^{\text{II}}(t-1)}\right|=1\tag{A14}$$

因此, 若 $s^{\text{II}}(t-1)\neq s^{\text{I}}(t-1)$, 必定有 $f'(\eta_t)<1$. 可得:

$$\lim_{t\to\infty}\left(\sum_{i=1}^{t}f'(\eta_i)\right)=0\tag{A15}$$

即 $t\to\infty$ 时 $s^{\text{II}}(t)-s^{\text{I}}(t)=0$.

上述过程表明 $0<|w^{\text{in}}|\leqslant 1$ 时, 任意隐含节点有唯一稳定平衡点, 且稳态输出与初始状态无关. 由于 IRRNN 的输出是隐含状态的线性叠加, 因此可得 IRRNN 有唯一稳定平衡点, 且稳态输出与状态的初始值无关. 证毕.

李文艺(1980—), 男, 博士生, 主要研究领域为神经网络, 复杂系统建模.

南静(1992—), 男, 博士生, 主要研究领域为神经网络, 增量式学习.

代伟(1984—), 男, 博士, 教授, 博士生导师, 主要研究领域为复杂系统建模, 机器学习.

刘从虎(1981—), 男, 博士, 教授, 主要研究领域为再制造技术, 智能制造.