



稀疏连接的异步池计算网络

薄迎春*, 张欣, 刘宝, 王平

中国石油大学(华东)控制科学与工程学院, 青岛 266580

* 通信作者. E-mail: boyingchun@sina.com.cn

收稿日期: 2019-08-26; 修回日期: 2019-11-07; 接受日期: 2020-01-09; 网络出版日期: 2021-04-12

国家自然科学基金(批准号: 21606256)资助项目

摘要 针对池计算网络的构建问题, 提出了一种稀疏连接的异步神经元池构造方法, 该方法将多个子神经元池顺序连接, 并在子神经元池之间设置滞后环节, 以实现各子神经元池对输入信息的异步处理, 进而构成串行的记忆. 为实现信息高效传输, 子神经元池之间采用稀疏的连接方式. 实验表明, 所提方法能够有效地提高神经元池的记忆容量, 易于解决长时依赖问题. 此外, 该结构能够使神经元池产生丰富的动力学行为, 对初始参数也有较好的鲁棒性.

关键词 人工神经网络, 池计算, 记忆, 鲁棒性, 动力学

1 引言

池计算 (reservoir computing, RC) 网络是一种以神经元池为核心的递归神经网络, 其典型代表是回声状态网络 (echo state network, ESN) 及流型状态机 (liquid state machine, LSM) [1,2]. 常规 RC 网络由输入、隐层、输出 3 部分组成, 其中隐层是一个由大量随机连接的神经元组成的神经元池 [3]. 理论研究表明, RC 网络能够以任意精度逼近任意动力学系统 [4]. 与其他递归神经网络相比, RC 网络只需学习隐层到输出之间的连接权值, 这使 RC 网络训练十分容易, 一般的线性求解方法即可得到最优的权值 [5~7]. 目前, RC 网络已经在时间序列预测 [1,8~10]、生物特征识别 [11]、控制 [12,13] 等领域得到了广泛的应用. 随着 RC 网络研究的不断深入, 常规 RC 的缺点也逐渐体现. 首先, 常规 RC 中存在大量随机给定并且始终保持不变的权值, 使得 RC 网络构建过程具有较大的随机性. 为获得高性能的网络, 需要进行大量测试 [5,14]. 其次, 常规 RC 网络的短时记忆 (short term memory, STM) 容量受池内神经元数量限制 [9], 对于长时依赖问题, 必须构建规模庞大的神经元池 [8,9], 这无疑会增加网络构建难度. 最后, 常规 RC 网络的池内神经元彼此耦合, 难以产生丰富的动力学行为 [8,14~16].

为解决上述问题, 一些改进的 RC 网络被陆续提出. 如解耦的 RC 网络将一个大的神经元池分解为多个子神经元池, 通过为子神经元池设置不同的参数以使其产生多样化的动力学行为 [14]. 文献 [15]

引用格式: 薄迎春, 张欣, 刘宝, 等. 稀疏连接的异步池计算网络. 中国科学: 信息科学, 2021, 51: 764-778, doi: 10.1360/SSI-2019-0183

Bo Y C, Zhang X, Liu B, et al. Sparsely connected asynchronous reservoir computing network (in Chinese). Sci Sin Inform, 2021, 51: 764-778, doi: 10.1360/SSI-2019-0183

进一步提出了子神经元池自适应增长算法, 以实现子神经元池数量及规模的优化. 平衡的 RC 网络通过设置反馈连接并调节反馈参数使输入信息与反馈信息达到平衡, 二者共同驱动神经元池, 以使神经元池产生更丰富的状态^[16]. D&S RC 在神经元池内采用带通滤波神经元, 通过设置不同的滤波参数使神经元池产生多种频率的信号^[8]. 随着深度学习研究不断深入, 深度 RC (deep RC, DRC) 也被提出^[16~20]. DRC 的神经元池由多个子神经元池顺序连接而成, 前一个子神经元池的状态作为下一个子神经元池的输入, DRC 通过多级映射使神经元池产生复杂的动力学行为. 上述改进方法主要目的是使神经元池产生多样化的动力学特征, 从而解决复杂的动力学建模问题. 为解决长时依赖问题, 记忆容量 (memory capacity, MC) 也成为 RC 网络研究的一个重要课题^[8,9,17,21,22]. 长时依赖问题的输出与过去较长时间的输入相关, 这需要 RC 网络具有较大的记忆容量. D&S RC 网络在池内每个神经元后面都附加了一个滞后环节^[8], 通过调节滞后时间可使每个神经元输出不同时刻的信息, 大幅增加了神经元池的记忆容量. 时间分割的池计算网络 (temporally segregated reservoir computing, TSRC) 以 DRC 为基础^[22], 在每两个相邻的子神经元池之间加入了一个滞后环节, 这种结构可以实现输入信息的分段记忆, 使一个小规模的神经元池也可以具有较大的记忆容量.

为进一步提高 RC 网络的动力学映射能力及解决长时依赖问题的能力, 提出了一种子神经元池稀疏连接的异步深度池计算网络 (sparsely connected asynchronous reservoir computing network, SARC). SARC 延续了 TSRC 中分段记忆的思想, 同时借鉴了 DRC 网络的深度结构. SARC 中各子神经元池能够异步处理输入信息及实现输入特征的分时记忆, 从而使神经元池具有较大的记忆容量. 此外, 为降低子神经元池输出信息的冗余, 在子神经元池之间采取了稀疏的连接方式^[23,24]. 实验表明, SARC 具有较大的记忆容量, 易于解决长时依赖问题. 同时, SARC 的状态具有较低的相关性, 这有助于提高神经元池的动力学映射能力及降低网络对随机初始参数的敏感性.

本文的主要工作如下: (1) 提出了 SARC 网络结构并深入分析了 SARC 的记忆模式及记忆容量; (2) 研究了 SARC 网络解决长时依赖问题及复杂动力学问题的能力; (3) 探讨了稀疏连接对 SARC 性能的影响.

2 SARC 网络结构

SARC 网络由多个子神经元池顺序连接而成 (图 1), 第 l (l 为子神经元池编号) 与第 $l+1$ 个子神经元池之间存在一个滞后环节 z^{-D^l} , D^l 为相应的滞后时间. SARC 的动力学模型可表示为

$$\mathbf{s}^l(k) = f(\mathbf{W}_I^l \mathbf{u}^l(k) + \mathbf{W}_R^l \mathbf{s}^l(k-1)), \quad (1)$$

$$\mathbf{y}(k) = \mathbf{W}_O^T \left[(\mathbf{s}^1(k))^T, \dots, (\mathbf{s}^L(k))^T \right]^T, \quad (2)$$

其中 $\mathbf{u}^l(k)$ 为第 l 个子神经元池输入; L 为子神经元池个数; $\mathbf{s}^l(k)$, \mathbf{W}_I^l , \mathbf{W}_R^l 分别为第 l 个子神经元池的内部状态、输入连接权值矩阵以及内部连接权值矩阵. \mathbf{W}_O 为整个神经元池 (包含所有子神经元池) 与输出之间的连接权值矩阵; $\mathbf{y}(k)$ 为网络输出; k 为时间步.

在 SARC 中, 第 1 个子神经元池的输入为网络外部输入 $\mathbf{u}(k)$, 即 $\mathbf{u}^1(k) = \mathbf{u}(k)$. 第 l ($l = 2, \dots, L$) 个子神经元池的输入为 $\mathbf{u}^l(k) = \mathbf{s}^l(k - \tau^l)$, 其中

$$\tau^l = \sum_{j=1}^{l-1} D^j. \quad (3)$$

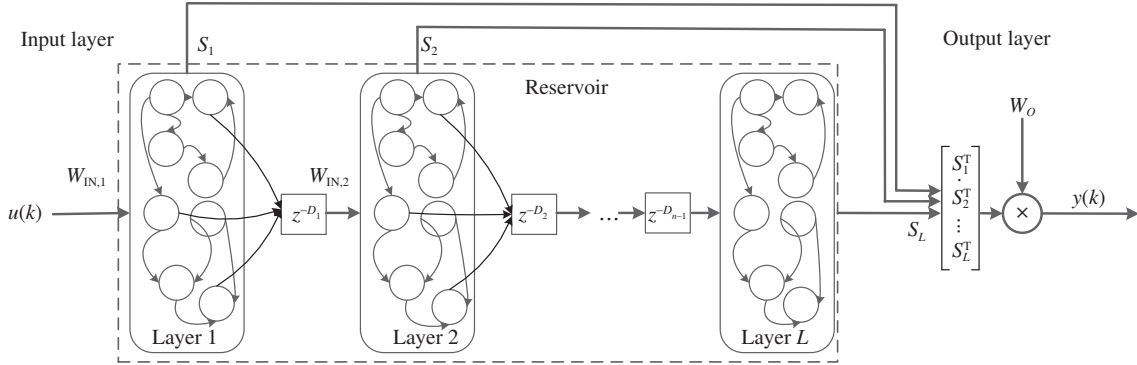


图 1 SARC 结构图

Figure 1 Scheme diagram of SARC

在当前时刻 k , SARC 中不同子神经元池接收到的信息是不同时刻的, 即每个子神经元池具有不同的记忆时段, 通过调整滞后时间就可以实现对特定时段信息的记忆. 这使得 SARC 的记忆非常灵活. SARC 中子神经元池之间采用稀疏连接的方式, 这可以降低信息传输的冗余, 提高信息处理的效率. SARC 的输出层负责对各子神经元池的状态进行整合, 以产生期望的输出. 由于 SARC 中神经元池存储了多个时段的信息, 这可以使输出端观察到更完整的输入特征, 有利于精确实现输入到输出的动力学映射. 此外, 由于记忆的信息特征不同, 各子神经元池的动力学行为自然存在差异. 因此神经元整体能够呈现出更丰富的动力学特征. 与常规 RC 网络相同, 在 SARC 中, 只有 \mathbf{W}_O 需要训练^[25, 26]. 为提高训练的鲁棒性, 这里采用岭回归最小二乘方法计算 \mathbf{W}_O 的值^[27]:

$$\mathbf{W}_O = \left(\mathbf{S}\mathbf{S}^T + \alpha\mathbf{I} \right)^{-1} \mathbf{S}\hat{\mathbf{Y}}^T, \quad (4)$$

其中 $\hat{\mathbf{Y}}$ 为训练样本期望输出矩阵, \mathbf{S} 为所有子神经元池的内部状态构成的矩阵, α 为岭回归系数.

3 SARC 网络短时记忆容量分析

3.1 SARC 网络短时记忆容量定义

如图 2(a) 所示, 常规 RC 只有一个神经元池, 只能记忆最近一段时间的输入信息^[9], 记忆区间记为 $[k - MC, k)$, MC 为神经元池的记忆容量. SARC 中每个子神经元池具有不同的记忆时段, 如图 2(b) 所示. 在 k 时刻, 第 1 到第 L 个子神经元池的记忆区间分别为 $[k - MC^1, k), \dots, [k - \tau^L - MC^L, k - \tau^L)$, MC^l 为第 l 个子神经元池的记忆容量. 根据式 (3), 调节 D^l 即可实现记忆区间的调整.

定义1 设 SARC 网络共有 L 个子神经元池, 各子神经元池的记忆区间分别为 $[k - MC^1, k), \dots, [k - \tau^L - MC^L, k - \tau^L)$, 则 SARC 网络的总记忆区间 U 为各子神经元池记忆区间的并集, 即

$$U = [k - \tau^L - MC^L, k - \tau^L) \cup [k - \tau^2 - MC^2, k - \tau^2) \cup \dots \cup [k - MC^1, k). \quad (5)$$

定义2 SARC 网络的记忆容量 MC 为区间 U 的时间宽度.

命题1 如果 $\forall l (l = 1, \dots, L - 1)$, 都有 $D^l \geq MC^l$, 则 SARC 网络的记忆容量达到最大值, 并且最大的记忆容量为 $MC_{\max} = \sum_{l=1}^L MC^l$.

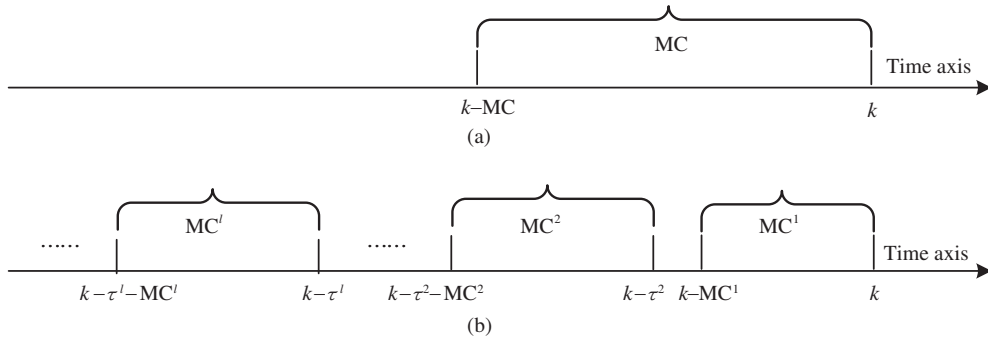


图 2 (a) Traditional RC 和 (b) SARC 记忆模式
Figure 2 Memory modes of (a) traditional RC and (b) SARC

证明 根据式 (3), 当 $D^l \geq MC^l$ 时, $\tau^{l+1} \geq \tau^l + MC^l$, 所以, $k - \tau^{l+1} \leq k - \tau^l - MC^l$. 即当 $D^l \geq MC^l$ 时, 第 $l+1$ 及第 l 个子神经元池的记忆区间完全不重叠, 两个子神经元池的记忆区间并集宽度为 $MC^{l+1} + MC^l$. 当 $D^l \geq MC^l$ 对所有 $l (l = 1, \dots, L-1)$ 都成立时, U 的宽度为 $\sum_{l=1}^L MC^l$, 这也是 U 的理论最大值.

命题2 如果 $\exists l (l = 1, \dots, L-1)$, 满足 $D^l < MC^l$, 则 SARC 网络的记忆容量 $MC < \sum_{l=1}^L MC^l$.

证明 假设当 $l = i$ 时, $D^i < MC^i$, 根据式 (3), $\tau^{i+1} < \tau^i + MC^i$. 即第 $i+1$ 及第 i 个子神经元池的记忆区间存在重叠, 此时两个子神经元池的记忆区间并集宽度小于 $MC^{i+1} + MC^i$. 所以, 当 $D^i < MC^i$ 时, $MC < \sum_{l=1}^L MC^l$.

命题 1 和 2 表明, 在 SARC 中, 可通过调节滞后时间实现神经元池中的柔性记忆. 当需要长时连续记忆时, 可以设置 $D^l < MC^l$; 当只需要记忆一些离散的信息片段时, 可设置 $D^l \geq MC^l$.

3.2 SARC 网络短时记忆容量测试

文献 [9] 提出的常规 RC 网络记忆容量测试方法如下:

$$MC = \sum_{d=0}^{\infty} MC_d, \tag{6}$$

其中,

$$MC_d = \frac{\rho^2(u(k-d), y_d(k))}{\sigma^2(u(k))\sigma^2(y_d(k))}, \tag{7}$$

$y_d(k)$ 为网络的第 d 个输出, $u(k)$ 为网络输入, ρ^2 及 σ^2 分别表示协方差及方差. MC_d 是 $y_d(k)$ 及 $u(k-d)$ 的相关系数. MC 反映了 RC 网络输出对过去 d 个时刻输入信号的复现能力. 虽然 SARC 结构与常规 RC 不同, 但如果将所有子神经元池看作是一个整体, 仍可以用文献 [9] 的方法测试 SARC 网络的记忆容量.

(1) 子神经元池数量与 MC 关系测试. 根据定义 1 和 2, 在 SARC 中, 每增加一个子神经元池, 相当于增加了一个记忆区间. 所以, 在这个实验中, 从只有 1 个子神经元池的常规 RC 开始, 逐渐增加子神经元池数量到 $L = 10$. 每增加 1 个子神经元池, 都按照式 (6) 和 (7) 计算 SARC 网络的记忆容量. 为减少随机因素的影响, 增加子神经元池及计算记忆容量的实验独立重复 10 次. 增加子神经元池具有相同的宏观参数: 神经元池规模 (N^l) 为 40, 谱半径 (ρ^l) 为 0.95, 稀疏度 (SD^l) 为 0.1, 子神经元池之间采用全连接方式. $D^l (l = 1, \dots, L-1)$ 均设置为相同的值. 实验中网络输入 $u(k)$ 为 $[-0.5, 0.5]$

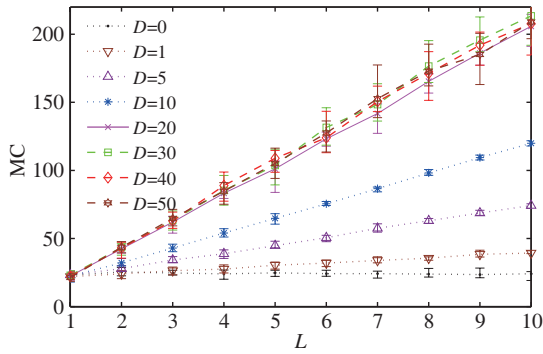


图 3 (网络版彩图) MC 随子神经元池数量的变化
Figure 3 (Color online) Change of MC vs. L

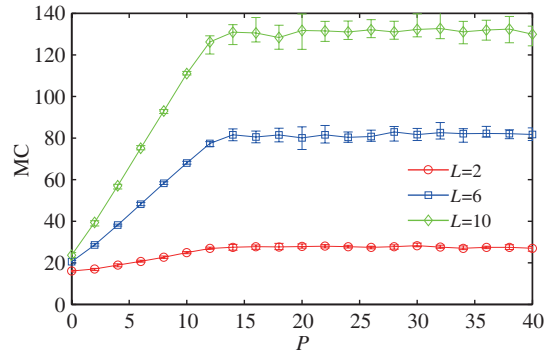


图 4 (网络版彩图) MC 随滞后时间的变化
Figure 4 Change of MC vs. D

表 1 MC 随子神经元池数量的变化

Table 1 Change of MC vs. number of sub-reservoirs

	$L = 2$	$L = 4$	$L = 6$	$L = 8$	$L = 10$	ΔMC
MC ($D = 0$)	15.31 ± 0.55	16.53 ± 0.38	17.22 ± 0.50	17.65 ± 0.46	17.63 ± 0.56	0.46 ± 0.55
MC ($D = 1$)	16.33 ± 0.29	22.09 ± 0.36	27.54 ± 0.29	32.88 ± 0.24	38.57 ± 0.58	2.84 ± 0.35
MC ($D = 5$)	20.06 ± 0.17	33.62 ± 0.50	46.70 ± 0.20	59.96 ± 0.60	73.15 ± 0.50	6.66 ± 0.21
MC ($D = 10$)	24.21 ± 0.31	46.05 ± 0.54	68.06 ± 0.71	90.15 ± 0.50	111.28 ± 0.60	10.77 ± 0.25
MC ($D = 20$)	27.33 ± 0.43	54.09 ± 1.51	81.25 ± 0.82	105.78 ± 4.19	128.86 ± 2.32	12.72 ± 1.39
MC ($D = 30$)	26.86 ± 0.53	53.26 ± 1.23	80.57 ± 2.59	109.54 ± 2.51	129.02 ± 6.02	12.79 ± 2.44
MC ($D = 40$)	26.73 ± 1.28	55.68 ± 1.94	82.36 ± 1.88	106.28 ± 2.82	128.61 ± 4.13	12.76 ± 2.48
MC ($D = 50$)	27.22 ± 0.76	54.56 ± 1.47	80.03 ± 1.55	105.73 ± 3.27	130.90 ± 3.55	13.00 ± 0.58

均匀分布的随机信号, 网络期望输出为 $u(k-d)$ ($d = 1, \dots, 500$). 实验样本总数为 3500, 前 2500 个用于训练, 后 1000 个用于测试, 训练集中初始的 500 个样本用来冲刷初始状态^[1]. 第 1 个子神经元池输入权值从 $[-0.1, 0.1]$ 中随机选取, 后续各子神经元池输入权值从 $[-1, 1]$ 中随机选取. 实验结果如图 3 及表 1 所示. 当 $D > 0$ 时, MC 随 L 的增加而增加. 并且每增加一个子神经元池, 网络增加的记忆容量大体相当 (见表 1, 存在差异主要是网络生成过程中随机因素的影响). 图 3 也显示当 $D > 0$ 时, MC 与 L 呈现出良好的线性关系. 当 $D = 0$ 时, MC 随 L 增加十分缓慢, 并且随着 L 的增加 MC 增幅逐渐减小, 这说明 $D = 0$ 时, 增加 L 不会出现记忆线性叠加的现象. 按照定义 1 和 2, 当 $D = 0$ 时, $MC = \max\{MC^l, l = 1, \dots, L\}$. 所以 MC 不会随 L 的增加而显著增加. $D = 0$ 时 MC 的缓慢增加主要是由于动力学映射能力增强导致的. 当 $D = 20 \sim 50$ 时, MC 随 L 变化曲线基本重合, 如图 3 所示, 命题 1 表明, 当 $D \geq \max\{MC^l, l = 1, \dots, L\}$ 时, 记忆容量达到最大值, 此时继续增大滞后时间不会再增加 MC. 通过这个实验可以看出, 定义 2 中的记忆容量与采用文献 [9] 方法测试得到的记忆容量具有较好的一致性.

(2) 滞后时间与 MC 关系测试. 接下来固定 L , 将滞后时间 D 从 0 逐步增加到 40 (步长为 2), 其他参数及实验设置与 (1) 相同. 每增加一次滞后时间, 计算相应的 MC (重复 10 次). 从图 4 可见, 在初始阶段, MC 随滞后时间线性增加, 这是因为当 D 较小时, 子神经元池的记忆区间存在重叠, 随着 D 的增加, 记忆重叠逐渐减小, 当 D 增大到一定值时, 记忆重叠消失, 此时记忆容量达到最大值.

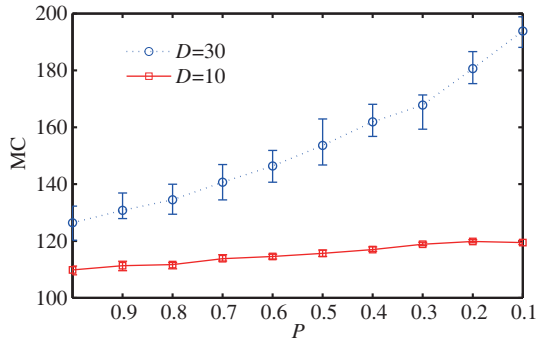


图 5 (网络版彩图) MC 随 P 的变化
Figure 5 (Color online) Change of MC vs. P

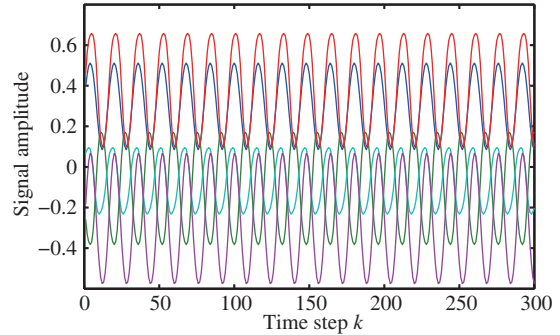


图 6 (网络版彩图) 正弦信号驱动下神经元池的状态变化
Figure 6 (Color online) State change of neuron reservoir under external sine signal driver

Figure 6 (Color online) State change of neuron reservoir under external sine signal driver

(3) 子神经元池之间连接度对 MC 影响. 接下来保持 (2) 中 $L = 10$ 的网络参数不变, 将各子神经元池的连接度 P 从 1 减小到 0.1 (步长为 -0.1), 采用随机连接方式, 即随机选择一定比例 (P) 的神经元状态作为下一个子神经元池的输入. 从图 5 可见, 随着 P 的降低, MC 呈上升趋势 (当 P 从 1 减小到 0.1 时, MC 增幅分别为 8.7% ($D = 10$) 和 72.4% ($D = 30$)). 所以, 子神经元池之间的稀疏连接对提高 SARC 的记忆容量是有利的. 当子神经元池采用全连接的模式时, 外部输入作用较强, 容易导致内部神经元产生与外部输入同步的动作, 如图 6 所示. 当连接度降低时, 外部输入信号减弱, 神经元池的动力学丰富程度增加, 有助于提高子神经元池的记忆能力. 从另一个角度, 神经元池中包含大量随机连接的神经元, 信息不可避免存在冗余, 此时采用全连接是不必要的. 当然, 并不是所有问题都需要神经元池具有大的 MC. 所以, 连接度的设置不能只以 MC 为指标, 还应考虑传输到下一个子神经元池的信息是否具有代表性.

4 滞后时间及连接度优化方法

4.1 滞后时间优化

滞后时间优化只需用到训练集样本, 首先将训练样本分为训练集和校验集两部分, 然后从只有一个神经元池的 RC 网络开始, 逐步增加子神经元池数量, 每增加一个子神经元池, 就对相应的滞后时间进行优化. 优化采用遍历搜索的方法, 即 D^l 从 0 增加到最大允许的滞后时间 D_{\max} . D^l 每增加一步, 都对网络进行训练和校验, 并记录所得的校验误差 E_v . 最优的 D^l (记为 \hat{D}^l) 是 \hat{E}_v 对应的 D^l . 当增加 L 不能使 \hat{E}_v 减小时, 优化过程结束 (见算法 1). \hat{E}_v 定义如下:

$$\hat{E}_v = \min_{0 \leq D^{l-1} \leq D_{\max}} \{E_v\}. \quad (8)$$

4.2 子神经元池之间连接度优化

神经元所携带的信息量可以用香农 (Shannon) 熵衡量. 第 l 个子神经元池的第 i 个神经元的熵可

算法 1 滞后时间优化算法

```

1: 初始化  $L_{\max}, D_{\max}, E_b, N^l, \rho^l, SD^l, P = 1, l = 1$ ;
2: 构建参数为  $N^1, \rho^1, SD^1$  的常规 RC 网络, 用训练集训练 SARC, 用校验集测试 SARC 性能并得到校验误差  $E_v$ ;
3:  $E_b = E_v$ ;
4: FLAG = 0;
5: while  $l < L_{\max}$  && FLAG == 0 do
6:    $l = l + 1$ ;
7:    $D^{l-1} = 0$ ;
8:   while  $D^{l-1} \leq D_{\max}$  do
9:      $D^{l-1} = D^{l-1} + 1$ ;
10:    用训练集训练 SARC; 用校验集测试 SARC 性能并得到校验误差  $E_v$ ;
11:   end while
12:   if  $\hat{E}_v \leq E_b$  then
13:      $E_b = E_v$ ;
14:      $D^{l-1} = \hat{D}^{l-1}$ ;
15:   else
16:      $L = l - 1$ ;
17:     FLAG = 1;
18:   end if
19: end while

```

用下式计算:

$$H(s_i^l) = \sum_{j=1}^q (-p(s_{ij}^l) \log_2 p(s_{ij}^l)), \quad (9)$$

其中 $H(s_i^l)$ 为信息熵, q 为样本数, s_{ij}^l 为第 i 个神经元对第 j 个样本的输出, $p(s_{ij}^l)$ 为 s_{ij}^l 出现的概率.

为便于优化实施, 这里设所有子神经元池连接度 P 相同. 第 l 个子神经元池中信息熵最大的 $P \times N^l$ (N^l 为第 l 个子神经元池规模) 个神经元与下一个神经元池连接. 优化过程从 $P = 1$ 开始, 按一定步长减小 P 值, 同时计算相应的校验误差, 校验误差不能降低时, 优化过程结束, 如算法 2 所示.

算法 2 连接度优化算法

```

1:  $P = 1, P_{\text{step}} = -0.05, P_{\text{min}} = 0.05, \text{FLAG} = 0$ ;
2: 训练并校验 SARC, 得到校验误差  $E_v, E_b = E_v, \text{FLAG} = 0$ ;
3: while  $P \geq P_{\text{min}}$  && FLAG == 0 do
4:    $P = P - P_{\text{step}}$ ;
5:   输入训练集信息, 并获得各子神经元池状态;
6:   利用式 (9) 估计各子神经元池所有神经元的熵并按从大到小排序;
7:   令第  $l$  个子神经元池中熵最大的  $P \times N^l$  个神经元状态输出到第  $l + 1$  个子神经元池 ( $l = 1, \dots, L$ );
8:   训练及校验连接度改变后的 SARC 并得到校验误差  $E_v$ ;
9:   if  $E_v \leq E_b$  then
10:      $E_b = E_v$ ;
11:   else
12:      $P = P + P_{\text{step}}$ ;
13:      $\hat{P} = P$ ;
14:     FLAG = 1;
15:   end if
16: end while

```

表 2 稀疏 NMA 参数设置

Table 2 Parameter settings for sparse NMA task

	Traditional RC	RC _{delay}	D&S RC	SARC
Input	$u(k)$	$[u(\tau), u(k - \tau), u(k - 2\tau)]$	$u(k)$	$u(k)$
L	1	1	1	3
N^l	100	100	100	33
D^l	—	—	—	$[\tau - 1, \tau]$
SD ^l	0.1	0.1	0.1	0.1
ρ^l	0.99	0.99	0.8	0.8

5 实验研究

(1) 稀疏的 NMA (nonlinear moving average) 建模^[8]. 稀疏的 NMA 模型如下:

$$y(k) = \left(\sum_{i=0}^3 u(k-i) \right) \left(\sum_{i=0}^3 u(k-\tau-i) \right) \left(\sum_{i=0}^3 u(k-2\tau-i) \right), \quad (10)$$

其中 τ 为滞后时间, $u(k)$ 为在 $[0.5, 1]$ 均匀分布的白噪声信号.

该问题的输出 $y(k)$ 只与过去 3 个时段的输入信息相关, 3 个时段分别是 k 至 $k-3$; $k-\tau$ 至 $k-\tau-3$; $k-2\tau$ 至 $k-2\tau-3$. 由于常规 RC 只能记忆一个时段的信息, 所以, 要采用常规 RC 解决此问题, 就必须构造一个规模庞大的神经元池, 以使神经元池能够记忆区间 $[k-2\tau-3, k)$ 的输入特征, τ 增加, 神经元池的规模也要随之增加. SARC 具有分段记忆特点, 可以构建 3 个神经元池 (滞后时间为 τ). 第 1 个神经元池输入为 $u(k)$, 相应的记忆区间为 $[k-MC^1, k)$; 第 2 个子神经元池输入时刻为 $k-\tau$, 相应的记忆区间为 $[k-\tau-MC^2, k-\tau)$; 第 3 个子神经元池输入时刻为 $k-2\tau$, 相应的记忆区间为 $[k-2\tau-MC^3, k-2\tau)$. 本问题只需每个子神经元池记住 3 个时间步的信息, 即 $MC^l \geq 3$ ($l=1, 2, 3$) 即可. 所以理论上 SARC 网络只需要 3 个子神经元池及少量的神经元即可解决上述稀疏的 NMA 问题.

实验中参数设置如表 2 所示, 为便于比较, 将所有神经元池总规模均设置为 100 (SARC 包含 3 个子神经元池, 每个子神经元池规模为 33). 常规 RC, RC_{delay} 以及 D&S RC 均为单神经元池网络. RC_{delay} 有 3 个外部输入: $u(k)$, $u(k-\tau)$ 及 $u(k-2\tau)$, 这使 RC_{delay} 能够记忆问题所需 3 个时段的信息特征. D&S RC 在池内每个神经元后附加一个滞后环节, 这相当于在一个神经元池内采用不同神经元完成不同时刻的记忆. 实验中共生成 $8000 + 4\tau$ 个样本, 前 2500 个样本用于冲刷初始状态, 接下来的 $5000 + 2\tau$ 个样本用于训练, 剩余的 $500 + 2\tau$ 个样本用于测试. 采用测试集中最后 500 个样本计算网络性能, 性能指标采用规范化的均方根误差 (normalized root mean square error, NRMSE)^[8].

图 7(a) 为 4 种模型的测试结果. 当 τ 较小时, 一个规模较小的神经元池即可满足记忆要求, 所以, 4 种模型都获得了较高的精度, 随着 τ 逐渐增加, 常规 RC 由于不能满足记忆容量需求, 逐渐失去解决问题能力. 其他 3 种模型在 τ 较大时也能较好地解决问题. 这说明后 3 种模型均可实现对所需信息的记忆. 相比之下, D&S RC 解决问题能力弱于 RC_{delay} 及 SARC, 这是由于 D&S RC 需要设置的滞后参数较多, 如何合理设置这些滞后参数是一个难题. 而 RC_{delay} 及 SARC 的滞后参数较少, 设置相对简单. 从性能上看, RC_{delay} 与 SARC 相当. 尽管如此, RC_{delay} 与 SARC 的记忆原理存在很大差别. RC_{delay} 将过去的信息和当前的信息并行输入到神经元池, 从而使神经元池能够对多个时段的信息形成记忆. 所以, RC_{delay} 采用的是并行的记忆方式, 即一个神经元池内同时存在不同时段输入特

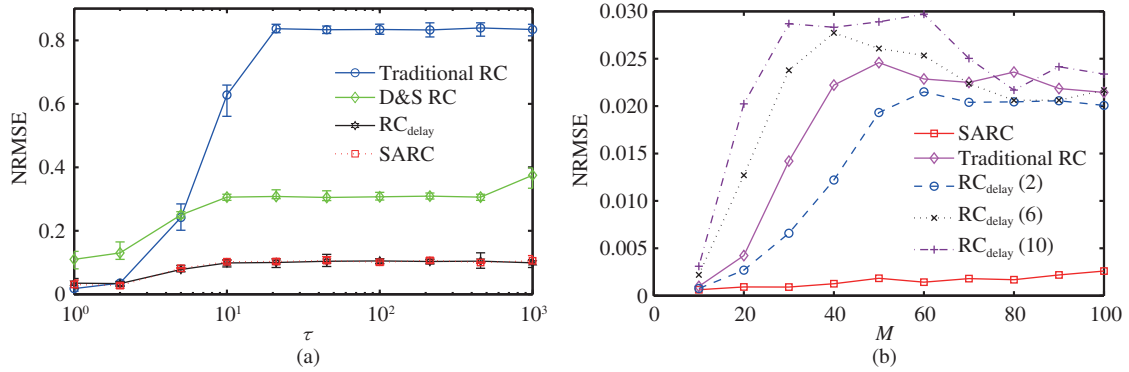


图 7 (网络版彩图) (a) 稀疏 NMA 问题和 (b) 常规 NMA 问题的测试结果

Figure 7 (Color online) Testing results of (a) sparse NMA task and (b) traditional NMA task

征. SARC 则是将不同时段的信息独立记忆于不同的子神经元池, 越久远的信息存储于越深层次的子神经元池, 这是一种串行的记忆方式. 从生物学的角度^[28], 过去的信息应该是以某种形式存储于记忆中, 外部输入会唤醒某些时段的记忆. 大脑内部实现当前信息和过去记忆的整合, 进而产生新的输出. 所以从宏观的角度, SARC 的信息处理模式比 RC_{delay} 更接近生物学的本质.

(2) 常规 NMA 建模. 考虑如下模型:

$$y(k) = \frac{1}{M} \sum_{i=1}^M u(k-i)u(k-i-1), \quad (11)$$

其中 $u(k)$ 为在 $[0.5, 1]$ 均匀分布的白噪声信号.

本问题中模型输出 $y(k)$ 与过去 M 个时刻的输入有关, 这要求神经元池能够记忆区间 $[k-M-1, k)$ 的输入特征. 与问题 (1) 不同, 这个问题需要神经元池能够记忆一段连续时间步的信息. 实验中设置神经元池总规模为 400. SARC 设置 10 个子神经元池, 每个子神经元池规模为 40, 子神经元池之间的滞后时间设置为 $\lfloor M/9 \rfloor$ (非优化值), 连接度设置为 0.40. 网络其他参数与表 2 相同. 实验中共生成 5500 个样本, 前 3000 个用于训练, 接下来的 2500 个用于测试. 训练集的前 500 个样本用于冲刷初始状态. NRMSE 采用测试集中最后的 300 个样本计算. 图 7(b) 显示了当 M 从 10 增加到 100 时常规 RC, RC_{delay} 以及 SARC 的测试结果. 其中, RC_{delay}(I) (I 为正整数) 为有 I 个输入的 RC_{delay} 网络, 输入为 $[u(k), \dots, u(k - \lfloor M/I \rfloor)]$. 经测试, 常规 RC 的记忆容量约为 38.96 ± 0.73 . 随着 M 的增加, 常规 RC 的性能越来越差, 当 $M > 40$ 时, 常规 RC 失去解决问题能力. RC_{delay}(2) 将过去的输入 $u(k - \lfloor M/2 \rfloor)$ 引入神经元池, 从而实现过去和当前信息的并行记忆. RC_{delay}(2) 能够记忆两个时段的信息, 理论上每个时段的记忆长度不会超过常规 RC 的记忆容量. 实际上, $M > 50$ 时, RC_{delay}(2) 就失去了解决问题的能力 (图 7(b)), 并且在 $M \leq 50$ 时, RC_{delay}(2) 的性能也随着 M 的增加而下降. 这主要由于 RC_{delay}(2) 的记忆方式会使不同时段的信息发生耦合, 对于复杂的问题容易造成记忆失真. 从图 7(b) 可以看出, 随着 I 的增加, RC_{delay}(I) 的性能越来越差, 这说明输入越多, 记忆耦合引起的记忆失真越严重. 当 $I \geq 6$ 时, RC_{delay} 的性能甚至不如常规 RC. M 从 10 增加到 100 时, SARC 一直保持了较好的解决问题能力, 这说明 SARC 有能力实现所需的连续记忆 (经测试, SARC 记忆容量为 115.43 ± 0.52). 根据命题 1, 只要满足 $\lfloor M/9 \rfloor \leq MC^l$ ($l = 1, \dots, 10$), 各子神经元池的记忆区间是连续的. 在此问题中, SARC 的每个子神经元池只需完成所需记忆的 $1/10$ 即可, 这对每个子神经元池来说都是容易实现的. 同时 SARC 的这种记忆方式能够大幅减弱记忆耦合引起的记忆失真. 所以, SARC

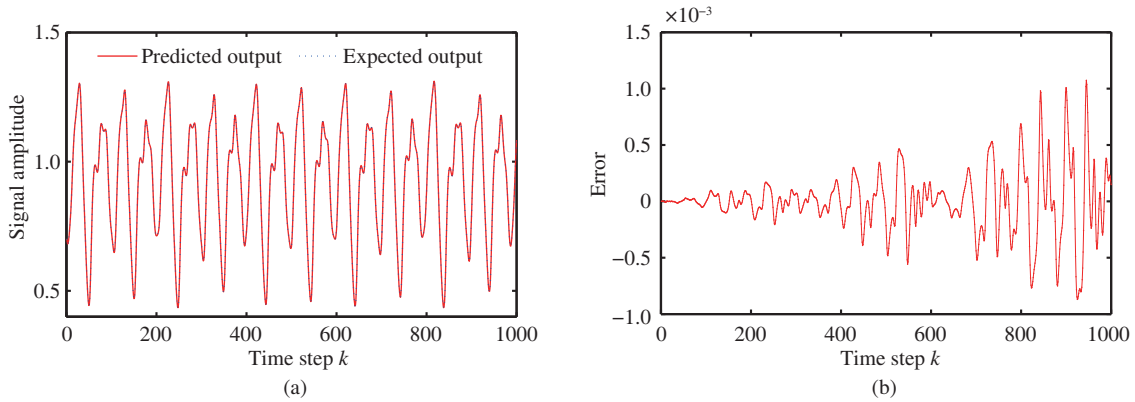


图 8 (网络版彩图) Mackey-Glass (a) 测试结果 (SARC) 和 (b) 测试误差 (SARC)
 Figure 8 (Color online) (a) Testing result and (b) testing error of SARC for Mackey-Glass

表 3 Mackey-Glass 参数设置

Table 3 Parameter settings of RC models for Mackey-Glass task

	Traditional RC	RC _{delay}	SARC
Input	$u(k)$	$[u(k), u(k - 6), u(k - 12), u(k - 18)]$	$u(k)$
L	1	1	4
N^l	1000	1000	250
D^l	–	–	[6, 6, 6]
SD ^l	0.1	0.1	0.2
ρ^l	0.99	0.99	0.99

在解决具有较大记忆容量需求的长时依赖问题方面具有显著优势。

(3) Mackey-Glass 时间序列预测. Mackey-Glass 时间序列预测是人工神经网络领域的一个典型问题, 其模型定义如下^[1]:

$$\dot{y}(t) = \frac{0.2y(t - \tau)}{1 + y(t - \tau)^{10}} - 0.1y(t). \quad (12)$$

当 $\tau > 16.8$ 时, 系统具有混沌状态, 本实验设置 $\tau = 17$.

实验共生成 4000 个样本, 前 100 个用来冲刷初始状态, 接下来的 1900 个用于训练, 最后的 2000 个用于测试. 在测试阶段, 网络输入信号不再从样本中获取, 而是将上一时刻的输出反馈到输入端作为输入信号. 也就是说, 在测试阶段, 网络处于一个自激运行状态. 由于预测输出存在误差, 这个误差反馈到输入端, 带有误差的输入会在输出端产生更大的误差, 如此往复, 网络的输出误差会越来越大, 最终偏离期望的 Mackey-Glass 混沌状态, 如图 8(b) 所示. 该实验主要测试模型对混沌时间序列的长时预测能力. 实验中采用测试样本的前 500 步来计算 NRMSE. 网络参数设置如表 3 所示, 为便于比较, 各网络的神经元池规模均设为 1000, 优化后的 SARC 网络的神经元池由 4 个子神经元池组成 (每个子神经元池有 250 个神经元), 子神经元池之间的滞后时间均为 6. 为减弱初始随机参数的影响, 对每种网络进行 100 次独立测试, 取 NRMSE 最小的 50 次数据进行性能分析.

从表 4 可见, 常规 RC, RC_{delay} 以及 SARC 均具有解决 Mackey-Glass 时间序列问题的能力. 相比之下, SARC 的 50 次测试 NRMSE 均值比常规 RC 及 RC_{delay} 低了一个数量级, 同时方差更小, 这说明 SARC 网络性能更加稳定, 对初始随机参数有更好的鲁棒性. Mackey-Glass 不属于长时依赖问题,

表 4 各类型网络性能比较 —— Mackey-Glass
Table 4 Performance comparison of various RC networks—Mackey-Glass

P	Maximum value	Minimum value	Mean value	Variance
RC	6.25×10^{-4}	3.87×10^{-5}	3.43×10^{-4}	1.68×10^{-4}
RC _{delay}	4.21×10^{-4}	1.73×10^{-5}	2.56×10^{-4}	1.02×10^{-4}
SARC	1.07×10^{-4}	1.71×10^{-5}	5.40×10^{-5}	2.55×10^{-5}

表 5 MSO 任务参数优化结果
Table 5 Optimization results of delay time for MSO tasks

Parameter	MSO ₂	MSO ₅	MSO ₈	MSO ₁₂	MSO ₁₆
L	5	5	5	5	4
D	6	10	13	16	16
P	0.45	0.40	0.50	0.45	0.45

所以并不需要神经元池具有大的记忆容量. 但 Mackey-Glass 时间序列的混沌特点需要神经元池能够产生足够丰富的动力学行为. 常规 RC 采用大规模的神经元池产生多样化的信号. SARC 在神经元池规模不变的情况下, 进一步采用异步的处理方式, 将不同时间的信号特征存储于不同的神经元池, 使池内信号的耦合程度降低. 此外, 不同时段的信息特征存在差异, 所以, SARC 神经元池的动力学行为比常规 RC 网络更为丰富, 从而使 SARC 网络能够取得更好的性能.

(4) MSO (multiple superimposed oscillator) 建模. MSO 模型如下:

$$y(k) = \sum_{i=1}^Q \sin(\alpha_i k), \quad (13)$$

其中 Q 为正弦波数量. $\alpha_i = 0.2 + 0.11(i - 1)$.

MSO 的输出由多个不同频率的正弦信号叠加而成. Q 越大, MSO 模型越复杂. MSO 模型是测试神经元池能否产生丰富动力学行为的一个经典问题^[8, 14, 16, 22, 29]. 文献^[29]认为, 常规 RC 的神经元池很难产生多种频率的信号, 所以常规 RC 难以解决复杂的 MSO 问题. 随着研究的深入, 陆续提出了多种可以解决复杂 MSO 问题的改进型 RC 网络^[8, 14, 16, 22]. 实验中, 共生成 1500 个样本, 前 1000 个为训练样本, 后 500 个为测试样本, 最开始的 100 个训练样本用来冲刷初始状态, 采用测试集前 300 个样本计算 NRMSE. 神经元池规模均设置为 400, SARC 中每个子神经元池规模为 $\lfloor 400/L \rfloor$ (表 5 为优化后参数); 谱半径均设置为 0.99, 池内神经元稀疏度为 0.1. 为降低初始参数的影响, 对每个 MSO 问题都进行 100 次独立测试, 用 NRMSE 最小的 50 个结果来分析综合性能.

各网络的测试性能如表 6 所示. 常规 RC 只能解决 2 阶 MSO 问题, 并且性能较差, 这验证了常规 RC 网络的神经元池很难产生复杂的动力学行为. 当 $Q = 2$ 时, SARC 的性能比 D&S RC 和 Ballanced RC 差, 这主要是由于当 $Q = 2$ 时, MSO 模型比较简单, 由于 SARC 结构的复杂性, 使其在解决简单的动力学问题时优势不明显. 当 $Q \geq 5$ 时, SARC 的性能显著超过 D&S RC 和 Ballanced RC. SARC 的性能也明显优于 RC_{delay}, 这是由于 RC_{delay} 只有一个神经元池, 虽然引入过去的输入能够提高神经元池信息的多样性, 但池内神经元的耦合同时也会抑制神经元池产生丰富动力学的能力. 与 RC_{delay} 相比, SARC 的多神经元池结构减弱了信息耦合, 同时采用异步的信息处理方式使不同的子神经元池产生不同的动力学行为, 增加了神经元池整体的动力学多样性. 图 9 显示了 100 次实验的误差分布, 从误差分布结果来看, SARC 的性能比 RC_{delay} 更稳定, 即 SARC 对初始的随机参数有更好的鲁棒性.

表 6 MSO 任务性能比较

Table 6 Performance comparison of various RC networks for MSO tasks

NRMSE	Traditional RC	D&S RC	Balanced RC	RC _{delay}	SARC
MSO ₂	1.48×10^{-4}	3.02×10^{-9}	2.51×10^{-12}	4.08×10^{-7}	2.49×10^{-8}
MSO ₅	—	8.21×10^{-5}	1.06×10^{-6}	1.38×10^{-6}	6.16×10^{-8}
MSO ₈	—	—	2.73×10^{-4}	1.01×10^{-5}	1.27×10^{-7}
MSO ₁₂	—	—	—	9.39×10^{-5}	2.10×10^{-6}
MSO ₁₆	—	—	—	1.20×10^{-3}	1.35×10^{-5}

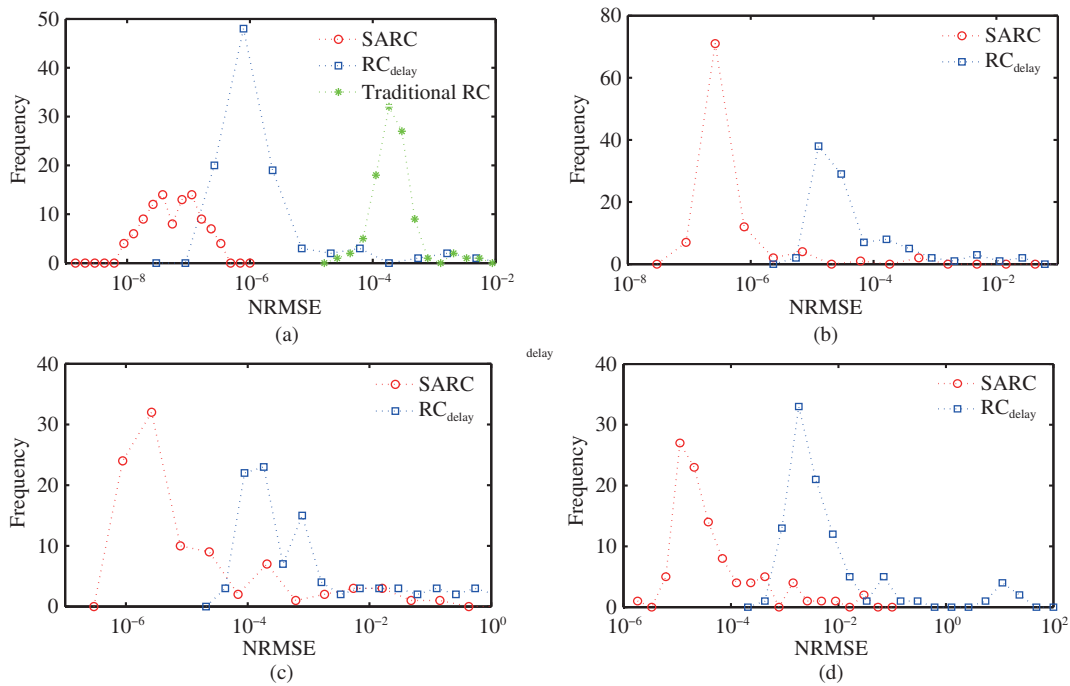


图 9 (网络版彩图) MSO 问题的误差分布

Figure 9 (Color online) Error distribution of MSO tasks. (a) MSO₂; (b) MSO₈; (c) MSO₁₂; (d) MSO₁₆

为衡量神经元池输出信号的总体差异, 定义平均相关度 \bar{R} 指标如下:

$$\bar{R} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N |R(\mathbf{S}_i, \mathbf{S}_j)|, \quad (14)$$

其中 $\mathbf{S}_i, \mathbf{S}_j$ 分别为第 i 及第 j 个池内神经元状态形成的序列, $R(\mathbf{S}_i, \mathbf{S}_j)$ 为 \mathbf{S}_i 及 \mathbf{S}_j 的 Pearson 相关系数.

从表 7 可见, 对于所有 MSO 问题, SARC 的 \bar{R} 要低于 RC_{delay} 及常规 RC, 这说明 SARC 的神经元池内信号差异性较大, 动力学特性也更为丰富.

(5) 稀疏度对系统性能的影响. 图 10 显示了稀疏度对 SARC 性能的影响. 对于稀疏的 NMA 问题、常规 NMA 问题、Mackey-Glass 时间序列预测问题以及 12 阶 MSO 问题, 进行连接度优化后, 平均 NRMSE 比全连接分别提高了 6.58%, 2.69%, 80.05% 以及 88.87%. 所以, SARC 中神经元池稀疏的连接有助于提升网络性能. 对于 NMA 问题, 输入为随机信号, 随机分布本身为最大熵分布, 信息冗

表 7 神经元池信号相关度比较

Table 7 Correlation degree comparison of various RC networks

\bar{R}	Traditional RC	RC _{delay}	SARC
MSO ₂	0.7243 ± 0.0292	0.4538 ± 0.0091	0.4473 ± 0.0074
MSO ₈	0.5793 ± 0.0234	0.3117 ± 0.0082	0.2089 ± 0.0106
MSO ₁₂	0.5158 ± 0.0181	0.3095 ± 0.0105	0.1870 ± 0.0089
MSO ₁₆	0.4817 ± 0.0269	0.2880 ± 0.0107	0.1619 ± 0.0072

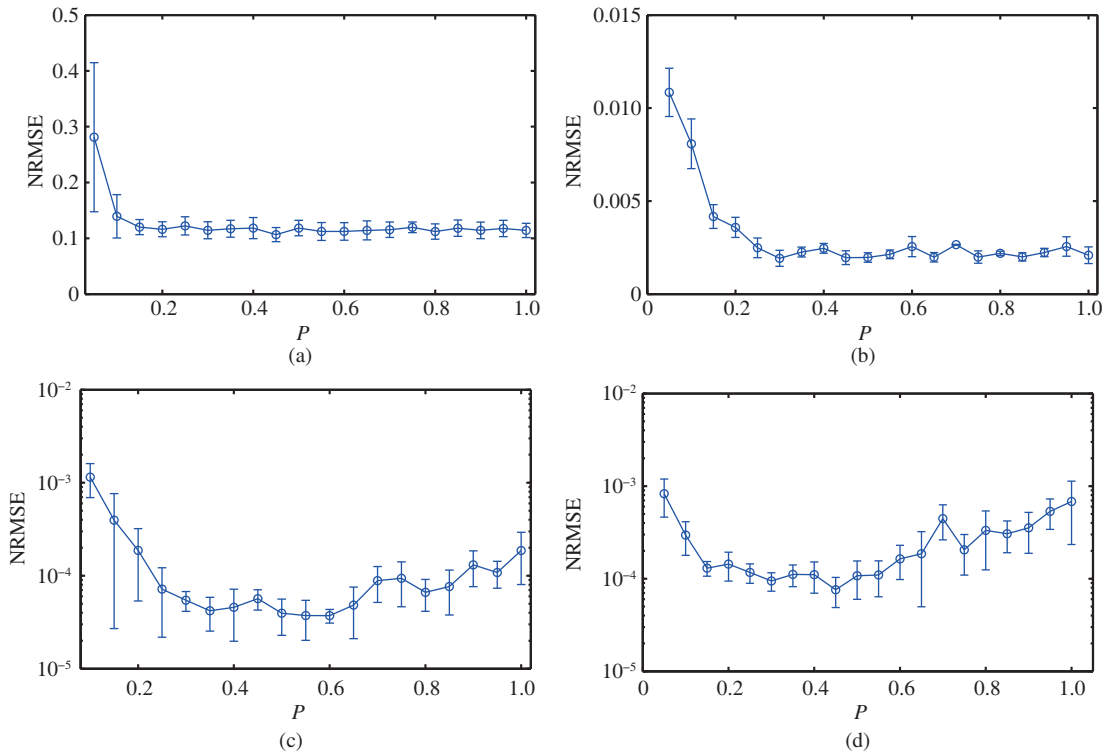


图 10 (网络版彩图) 连接度对网络性能的影响

Figure 10 (Color online) Effects of connection degree to network performance. (a) Sparse NMA; (b) traditional NMA; (c) Mackey-Glass; (d) MSO₁₂

余较小, 所以稀疏连接与全连接性能差别不大, 但采用稀疏连接能够使子神经元池之间的连接矩阵包含大量零元素, 可降低计算负担. 对于 Mackey-Glass 及 MSO 问题, 输入为有规律的信号, 池内信息也会随外部输入规律的变化而变化, 池内神经元状态会部分趋同, 信息冗余较大, 这时采用稀疏连接模式能够降低信息的冗余, 使网络性能提高.

6 结论

记忆与智能密切相关, SARC 是从记忆的角度提出的一种新的神经元池构建方法, 该方法采用异步信息处理方式实现了信息的串行记忆, 这种记忆具有深度的结构, 越久远的信息存储于越深层次的神经元池. SARC 的这种记忆模式可以实现长时记忆任务的分解, 降低完成长时记忆任务的难度. 此

外, SARC 的记忆更加柔性, 通过调整滞后时间, 可以记忆需要的信息, 同时丢弃不需记忆的信息, 这使得 SARC 能够用非常简单的结构解决长时依赖问题. SARC 中子神经元池采用了稀疏的连接方式, 这可以降低信息冗余, 使信息得到更有效的传输, 进一步提高网络的性能. 滞后时间的选择对于 SARC 的性能有重要影响, 其与子神经元池的宏观参数 (规模, 谱半径等) 相关, 在不同的子神经元池参数下, 优化得到的滞后时间也可能不同. 从文中的实验可见, 多数情况下, 各子神经元池采用相同的宏观参数, 相同的滞后时间和相同的连接度即可满足大多数问题要求, 这在很大程度上简化了参数的优化过程.

参考文献

- 1 Jaeger H, Haas H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, 2004, 304: 78–80
- 2 Maass W, Markram H. On the computational power of circuits of spiking neurons. *J Comput Syst Sci*, 2004, 69: 593–616
- 3 Lyburn T, Khor A, Stemler T, et al. Consistency in echo-state networks. *Chaos: an interdisciplinary. J Nonlin Sci*, 2019, 29: 023118
- 4 Grigoryeva L, Ortega J P. Echo state networks are universal. *Neural Netw*, 2018, 108: 495–508
- 5 Ozturk M C, Xu D, Príncipe J C. Analysis and design of echo state networks. *Neural Computat*, 2007, 19: 111–138
- 6 Qiao J, Wang L, Yang C, et al. Adaptive levenberg-marquardt algorithm based echo state network for chaotic time series prediction. *IEEE Access*, 2018, 6: 10720–10732
- 7 Yang C, Qiao J, Ahmad Z, et al. Online sequential echo state network with sparse RLS algorithm for time series prediction. *Neural Netw*, 2019, 118: 32–42
- 8 Holzmam G, Hauser H. Echo state networks with filter neurons and a delay&sum readout. *Neural Netw*, 2010, 23: 244–256
- 9 Jaeger H. Short-Term Memory in Echo State Networks. Technical Report no. 152. German National Research Center for Information Technology Bremen, 2002
- 10 Xu M, Yang Y, Han M, et al. Spatio-temporal interpolated echo state network for meteorological series prediction. *IEEE Trans Neural Netw Learn Syst*, 2019, 30: 1621–1634
- 11 Koprinkova-Hristova P, Stefanova M, Genova B, et al. Features extraction from human eye movements via echo state network. *Neural Comput Applic*, 2019, 32: 4213–4226
- 12 Bo Y C, Zhang X. Online adaptive dynamic programming based on echo state networks for dissolved oxygen control. *Appl Soft Comput*, 2018, 62: 830–839
- 13 Jordanou J P, Antonelo E A, Camponogara E. Online learning control with echo state networks of an oil production platform. *Eng Appl Artif Intell*, 2019, 85: 214–228
- 14 Xue Y, Yang L, Haykin S. Decoupled echo state networks with lateral inhibition. *Neural Netw*, 2007, 20: 365–376
- 15 Qiao J, Li F, Han H, et al. Growing echo-state network with multiple subreservoirs. *IEEE Trans Neural Netw Learn Syst*, 2017, 28: 391–404
- 16 Koryakin D, Lohmann J, Butz M V. Balanced echo state networks. *Neural Netw*, 2012, 36: 35–45
- 17 Gallicchio C, Micheli A, Pedrelli L. Design of deep echo state networks. *Neural Netw*, 2018, 108: 33–47
- 18 Gallicchio C, Micheli A. Richness of deep echo state network dynamics. In: *Proceedings of International Work-Conference on Artificial Neural Networks*, 2019. 480–491
- 19 McDermott P L, Wikle C K. Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*, 2019, 30: e2553
- 20 Gallicchio C, Micheli A, Silvestri L. Local Lyapunov exponents of deep echo state networks. *Neurocomputing*, 2018, 298: 34–45
- 21 Ma Q, Zhuang W, Shen L, et al. Time series classification with echo memory networks. *Neural Netw*, 2019, 117: 225–239
- 22 Bo Y C, Zhang X, Liu B. Temporally segregated reservoir computing and its dynamics. *Control Theory Appl*, 2019, 36: 1316–1321 [薄迎春, 张欣, 刘宝. 时间分割的池计算网络及其动力学. *控制理论与应用*, 2019, 36: 1316–1321]

- 23 Erhan G, Fraenz C, Caroline S, et al. Diffusion markers of dendritic density and arborization in gray matter predict differences in intelligence. *Nat Commun*, 2018, 9: 1905
- 24 Rodriguez N, Izquierdo E, Ahn Y Y. Optimal modularity and memory capacity of neural reservoirs. *Netw Neurosci*, 2019, 3: 551–566
- 25 Yang C, Qiao J, Han H, et al. Design of polynomial echo state networks for time series prediction. *Neurocomputing*, 2018, 290: 148–160
- 26 Yang C, Zhu X, Ahmad Z, et al. Design of incremental echo state network using leave-one-out cross-validation. *IEEE Access*, 2018, 6: 74874–74884
- 27 Shi Z W, Han M. Ridge regression learning in ESN for chaotic time series prediction. *Control Decision*, 2007, 22: 258–267 [史志伟, 韩敏. ESN 岭回归学习算法及混沌时间序列预测. *控制与决策*, 2007, 22: 258–267]
- 28 Moscovitch M, Cabeza R, Winocur G, et al. Episodic memory and beyond: the hippocampus and neocortex in transformation. *Annu Rev Psychol*, 2016, 67: 105–134
- 29 Jaeger H. The Echo State Approach to Analysing and Training Recurrent Neural Networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001

Sparsely connected asynchronous reservoir computing network

Yingchun BO*, Xin ZHANG, Bao LIU & Ping WANG

College of Control Science and Engineering, China University of Petroleum (Huadong), Qingdao 266580, China

* Corresponding author. E-mail: boyingchun@sina.com.cn

Abstract In order to solve the reservoir computing network construction problem, a sparsely connected asynchronous neuron reservoir construction method is proposed. The method connects several sub-reservoirs sequentially and sets lag links among sub-reservoirs in order to handle input signals asynchronously in sub-reservoirs, and further constitutes serial memory. In order to achieve efficient information transmission, sparse connections are used among sub-reservoirs. Experimental results show that the proposed method can effectively improve the memory capacity of reservoir and it is easy to deal with long-term dependence problems. In addition, the proposed structure makes the reservoir produce more abundant dynamic behavior and has better robustness to the initial parameters.

Keywords artificial neural network, reservoir computing, memory, robustness, dynamics



Yingchun BO was born in 1977. He received his Ph.D. degree in Beijing University of technology, Beijing, in 2013. His research interests include artificial neural networks and optimal control, and he is a lecturer of China University of Petroleum.



Xin ZHANG was born in 1982. She received her Ph.D. degree in Northeastern University, Shenyang in 2013. Her interests include intelligent optimal control and machine learning, and she is a lecturer of China University of Petroleum.



Bao LIU was born in 1968. He received his Ph.D. degree in Donghua University, Shanghai, in 2006. His research interests include artificial neural networks and optimal control, and he is a professor of China University of Petroleum.



Ping WANG was born in 1982. He received his Ph.D. degree in China University of Petroleum, Qingdao, in 2012. His research interests include modeling of Complex Systems and optimal control, and he is a lecturer of China University of Petroleum.