

# 控制与决策

Control and Decision

## 一种改进的在线自适应模块化神经网络

郭鑫, 李文静, 乔俊飞

引用本文:

郭鑫, 李文静, 乔俊飞. 一种改进的在线自适应模块化神经网络[J]. 控制与决策, 2020, 35(7): 1597–1605.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1463>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于AQPSO的RBF神经网络自组织学习

AQPSO-based self-organization learning of RBF neural network

控制与决策. 2018, 33(9): 1631–1636 <https://doi.org/10.13195/j.kzyjc.2017.0595>

### 基于PSO算法的动态模块化神经网络结构设计

Dynamic modular neural network structure design based on PSO algorithm

控制与决策. 2018, 33(6): 1055–1061 <https://doi.org/10.13195/j.kzyjc.2017.0060>

### 一种基于混合梯度下降算法的模糊神经网络设计及应用

Design and application of hybrid gradient descent-based fuzzy neural network

控制与决策. 2017, 32(9): 1635–1641 <https://doi.org/10.13195/j.kzyjc.2016.1032>

### 基于LM算法的在线自适应RBF网结构优化算法

Online self-adaptive optimal algorithm for RBF network based on Levenberg–Marquardt algorithm

控制与决策. 2017, 32(7): 1247–1252 <https://doi.org/10.13195/j.kzyjc.2016.0887>

### 基于Hammerstein模型的感应电机变频器调速系统神经网络控制

Speed-regulating system for induction motor and inverter based on Hammerstein model and neural network control

控制与决策. 2015(6): 1148–1152 <https://doi.org/10.13195/j.kzyjc.2014.0572>

# 一种改进的在线自适应模块化神经网络

郭 鑫<sup>†</sup>, 李文静, 乔俊飞

(1. 北京工业大学 信息学部, 北京 100124; 2. 计算智能与智能系统北京市重点实验室, 北京 100124)

**摘 要:** 针对在线模块化神经网络难以实时有效划分样本空间, 提出一种改进的在线自适应模块化神经网络. 首先, 基于距离与密度实时更新样本局部密度及 RBF 神经元中心, 实现样本空间在线划分; 然后, 将子网络模块数根据划分样本空间的个数进行自适应增减, 每个子网络模块对属于对应样本空间的样本进行学习; 最后, 集成模块对子网络模块的输出结果进行集成, 输出最终结果. 针对在线梯度下降法要求样本有足够的随机性问题, 提出一种具有固定记忆的在线梯度下降法对网络进行在线学习. 通过对典型非线性时变系统及污水处理过程中出水氨氮浓度进行预测, 验证了该模块化神经网络能够实时有效地更新 RBF 神经元中心, 并减少学习过程中子网络模块不必要的增减, 且网络结构更加简洁, 能够准确预测不同的时变系统.

**关键词:** 自适应; 模块化神经网络; 局部密度; 固定记忆; 在线梯度下降; 时变系统

中图分类号: TP183

文献标志码: A

## An improved online adaptive modular neural network

GUO Xin<sup>†</sup>, LI Wen-jing, QIAO Jun-fei

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; 2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China)

**Abstract:** An improved online adaptive modular neural network is proposed for the online modular neural network which is difficult to effectively divide sample space in real time. Firstly, the proposed network performs on-line task decomposition by real-time updating local sample density and RBF neuron center based on distance and density. Then the number of sub-network modules is adaptively increased or decreased according to the number of divided sample spaces, and each sub-network module learns the corresponding sample. Finally, integration module integrates the output of sub-network modules and outputs the final result. For the problem of the online gradient descent method requiring the sample with enough randomness, an online gradient descent method with fixed memory is proposed. Based on the prediction of the typical nonlinear time varying systems and the effluent ammonia nitrogen values from waste water treatment processes, the experimental results show that, the proposed network can effectively update the RBF neuron center in real time, which reduces unnecessary increase or decrease of sub-network modules during the learning process, has more concise structure and can accurately predict the time-varying systems.

**Keywords:** self-adaption; modular neural network; local density; fixed memory; online gradient descent; time-varying

## 0 引 言

前馈神经网络(feedforward neural network, FNN)因具有良好的非线性逼近能力、简单的结构, 已广泛应用于控制、系统建模等领域<sup>[1-2]</sup>, 其结构与学习算法被广泛研究<sup>[3-4]</sup>, 但全互连前馈神经网络存在灾难性遗忘问题, 且随着数据的增大, 网络结构的复杂性也变大<sup>[5]</sup>.

模块化神经网络(modular neural network, MNN)是受神经生物学启发, 模仿脊椎动物大脑的组织和

功能所设计的, 采用“分而治之”的策略, 将一个复杂的任务分解为多个简单独立的子任务, 并根据不同分配策略, 将每个子任务分配给相应子网络模块进行学习, 最后对各子网络模块学习的结果集成, 从而实现神经网络对复杂任务的学习<sup>[6]</sup>, 提高神经网络泛化能力, 降低任务的复杂性. 在学习过程中, MNN 增加新子网络模块(专家网络)对新知识进行学习, 而不改变神经网络原来的结构, 所以不会忘记旧知识, 有效地解决了 FNN 的灾难性遗忘问题<sup>[7]</sup>. 近年

收稿日期: 2018-10-26; 修回日期: 2019-03-04.

基金项目: 国家自然科学基金项目(61533002, 61603009); 北京市自然科学基金项目(4182007); 北京工业大学日新人才计划项目(2017-RX(1)-04).

责任编辑: 关治洪.

<sup>†</sup>通讯作者. E-mail: guo\_0xin@163.com.

来, MNN 引起了学者的广泛重视, 并应用在各个领域. 如 Sánchez 等<sup>[8]</sup> 针对人体识别, 提出了一种基于粒计算与萤火虫算法的模块化神经网络优化设计方法, 优化模块化神经网络参数为: 模块的数量、训练阶段的数据百分比、学习算法、目标误差、隐含层的数量和神经元数量. Melin 等<sup>[9]</sup> 提出一种基于多目标层次遗传算法的模块化神经网络优化方法, 该方法对 MNN 的模块个数、训练阶段数据集、目标误差、学习算法、隐含层数及该层神经元数等不同参数进行优化, 这种 MNN 可应用于不同的应用领域, 如人体识别、分类问题、时间序列等. Hu<sup>[10]</sup> 提出一种简化复杂分类任务的矩阵模块化神经网络分类器 (MMNN), 主要由神经网络矩阵和集成机制两部分组成, 不同样本进入神经网络矩阵中的每个模块单元, 每个单元产生一个输出结果, 所有结果由一个输出矩阵组成, 最后集成机制根据该输出矩阵进行集成分类决策. Sánchez 等<sup>[11]</sup> 提出一种采用 1 型模糊逻辑集成的模块化神经网络进行预测时间序列. 该方法将数据集划分为多个子数据集, 每个模块学习一个特定时间段的时间序列, MNN 的模块数是固定的; 对于不同的数据, MNN 具有不同的模块数, 最后采用 1 型模糊逻辑对每个模块结果进行集成. Qiao 等<sup>[12]</sup> 设计了一种基于自适应粒子群优化算法的动态模块化神经网络 (APSO-DMNN), 可以动态地优化 MNN 的权重, 对每个模块输出最优化集成, 最后得到动态调整的 MNN. 但是, 对于应用于各领域的 MNN, 大多处理离线问题或针对具体实时问题设计不同的自适应 MNN 结构, 该类 MNN 不具有普遍适用性. 另外, 进化算法近年来常被用于优化 MNN 结构<sup>[13-14]</sup>, 但工业自动化已成为现代工业的发展趋势, 其核心应该是对“变化”的响应能力, 大多数工业系统都是时变系统, 这类 MNN 需要很大的计算空间, 不适合处理实时问题.

针对时变系统, 在线任务分解是实现 MNN 在线学习的关键, 所以许多聚类算法结合具有“局部映射”特性的 RBF 神经网络用于任务分解, 提高了 MNN 的性能. 文献[15]提出了一种局部门控网络的混合专家模型对任务在线分解的 SAMNN(structure adaptive modular neural network), 初始网络结构为一个专家网络, 根据网络局部误差(修剪法)逐个增加(删减)专家网络, 对时变系统 SAMNN 能有效地自适应调整网络结构, 但无法及时准确反映时变系统. 文献[16]提出了 SGMN(self-regulation growing multi-experts network), 根据高斯基函数特性, 采用一种完

全自组织的自适应共振理论聚类, 删减或增加局部线性专家模型, 并采用在线梯度下降法修正网络参数, 但在线梯度下降法及局部线性专家模型都有固有的局限性, 不能很好地预测时变系统. 在此基础上, 文献[17]提出了 OSAMNN(an online self-adaptive modular neural network for time-varying systems), 利用 RBF 神经网络特性及“与门”, 结合在线减法聚类算法实现对在线任务分解, 该网络可根据在线工况数据自动调整网络模块个数, 能很好地实时反映时变系统. 但在线减法聚类算法更新聚类中心根据的是新样本与当前中心的密度, 没有考虑新样本对其他附近样本的影响, 不能及时反映样本间的相关性及聚类中心 (RBF 神经元), 而且网络学习算法采用在线 BP 算法, 存在其收敛性受到样本顺序影响等问题.

针对以上问题, 本文提出一种基于在线聚类算法的在线任务分解, 设计一种改进的自适应模块化神经网络 (online adaptive modular neural network, OAMNN), 其子网络模块数可根据基于距离和密度的在线聚类算法自动调整, 能实时更新样本局部密度, 增加样本间的相关性, 及时准确地更新聚类中心 (RBF 神经元), 减少网络在学习过程中子网络模块的不必要增减, 且网络结构更加简洁. 同时, 针对在线梯度下降法的局限性, 提出一种具有固定记忆的在线梯度下降法. 最后通过实验验证了在样本连续顺序或随机顺序情况下, 该网络均能稳定收敛.

## 1 OAMNN 结构

根据模块化神经网络的一般结构<sup>[17]</sup>, OAMNN 结构可分为输入层、任务分解层、子网络层、集成输出层, 如图 1 所示.

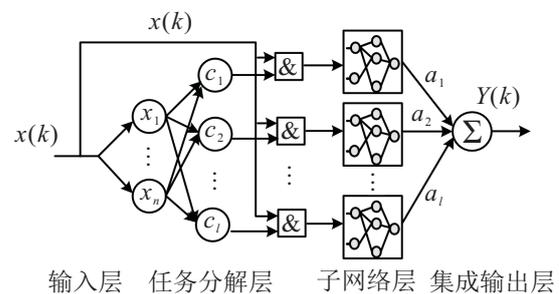


图 1 在线模块化神经网络结构

OAMNN 结构各层描述如下:

输入层: 该层输入节点数  $n$  由  $k$  时刻样本维数 ( $x(k)$  维数) 决定,  $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$ .

任务分解层: 该层由  $l$  个 RBF 神经元及对应的“与门”组成, 其中第  $i$  个 RBF 神经元激活函数如下:

$$\varphi_i(x(k), c_i) = e^{-\frac{\|x(k) - c_i\|^2}{\delta_i^2}} \quad (1)$$

其中:  $c_i$  为RBF神经元中心,表示第  $i$  个子任务(聚类)中心;  $\delta_i^2$  为核函数的扩展常数;  $\varphi_i(x(k), c_i)$  为硬限幅函数,有

$$\varphi_i(x(k), c_i) = \begin{cases} 1, & \|x(k) - c_i\| \leq \delta_i^2; \\ 0, & \|x(k) - c_i\| > \delta_i^2. \end{cases} \quad (2)$$

当此刻输入样本  $x(k)$  在RBF神经元  $i$  的超球面内时,激活该神经元,输出为1,相应的子网络对输入样本进行学习;否则不激活该神经元,输出为0. 该层RBF神经元及“与门”实现了输入样本空间的分解,并分配给相应的子网络模块.

子网络层: 该层的子网络模块与RBF神经元相对应,每个子网络模块为3层前馈神经网络,对上层分解的子任务进行训练和学习. 当  $k$  时刻输入样本  $x(k)$  时,子网络层中第  $h$  个子网络模块输出为

$$y_h(x(k)) = w_h f(v_h x(k) + b_{1h}) + b_{2h}. \quad (3)$$

其中:  $v_h$  为子网络模块输入层与隐含层之间的权值,  $w_h$  为子网络模块隐含层与输出层之间的权值,  $b_{1h}$ 、 $b_{2h}$  分别为隐含层、输出层的偏置. 隐含节点的激活函数为sigmoid函数.

集成输出层: 该层主要对子网络层中参与输入样本  $x(k)$  学习的子网络模块输出进行集成. 本文采用输入样本  $x(k)$  只激活一个子网络模块,激活的子网络模块输出是  $k$  时刻模块化神经网的输出.  $k$  时刻,模块化神经网络输出为

$$Y(k) = a(k)y(k). \quad (4)$$

其中:  $y(k) = [y_1(k), y_2(k), \dots, y_l(k)]^T$  为子网络模块的输出,  $a(k) = [a_1, a_2, \dots, a_l]$  为集成矩阵,  $a(k)$  矩阵中只有一个元素(激活子网络模块)为1,其他为0.

## 2 OAMNN结构设计

### 2.1 在线任务分解算法

OAMNN 主要通过在线分解任务实时调整任务分解层中RBF神经元数及RBF神经元中心  $c_i$  位置(聚类中心),本文提出的在线任务分解是基于距离与密度的聚类算法,根据新样本局部密度实时更新全局样本密度及聚类中心,使网络更真实地适应时变环境. 在线任务分解算法更新RBF神经元数、 $c_i$  中心及样本局部密度过程如下:

#### 1) 创建第1个RBF神经元.

在线任务分解开始时,RBF神经元数为0,随着样本不断输入,根据下式计算每个样本的局部密度  $\rho$ :

$$\rho_i = \sum_j \chi(d_{ij} - d_c). \quad (5)$$

其中:  $d_{ij}$  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

$$\chi = \begin{cases} 1, & d_{ij} \leq d_c; \\ 0, & d_{ij} > d_c. \end{cases} \quad (6)$$

当  $k$  时刻样本  $i$  的局部密度  $\rho_i(k)$  大于密度阈值  $\rho_o$  时,创建第1个RBF神经元,以样本  $i$  为神经元中心  $c_1$ ,减少了神经中心位置的更新频率,更真实地反映数据样本中心. 当样本  $j$  到神经元中心  $c_1$  距离  $d_{1j}$  小于类距离阈值  $d_l$  时,激活该神经元,否则存储到未分类Uc列表中.

任务分解过程中,聚类中心  $c$  局部领域内的样本数约为总样本数的1%~2%<sup>[18]</sup>,所以本文密度阈值  $\rho_o$  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%,聚类过程对类距离阈值  $d_l$  不敏感. 且本文MNN的任务分解可以增加子任务和融合子任务,密度阈值  $\rho_o$  和类距离阈值  $d_l$  对任务分解影响不是很大,实验中选择  $d_l \leq 2d_c$ .

#### 2) 更新RBF神经元中心.

新样本输入时,计算得到各RBF神经元中心  $c_i$  的距离  $d_i$ ,当距离  $d_i$  小于类距离阈值  $d_l$  时,表明新样本可激活该神经元,并将神经元中心  $c_i$  存储到  $S$  列表中. 本文采用输入样本  $x(k)$  只激活一个神经元的规则,当新样本可激活多个神经元时,分别计算新样本激活  $S$  列表中神经元时的局部密度,最后只激活新样本激活神经元时具有最大局部密度的神经元.

图2为新样本  $x(k)$  可激活两个RBF神经元  $a$ 、 $b$  的情况,可见新样本激活神经元  $a$  的局部密度明显大于  $b$  的局部密度,则新样本属于  $a$  类,激活神经元  $a$  输出为1,神经元  $b$  输出为0. 通过对该样本分类规则描述可知,样本分类对类距离阈值  $d_l$  是不敏感的,取决于局部密度. 确定新样本属于  $a$  类后,计算新样本局部密度及更新新样本  $d_c$  局部区域内样本的局部密度,并判断更新样本及新样本的局部密度是否存在大于RBF神经元局部密度的样本,如果存在则更新RBF神经元中心  $c_a$ .

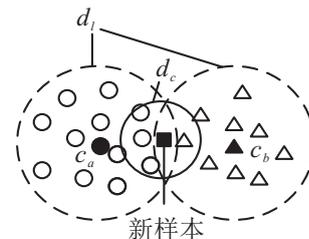


图2 样本分类规则示意图

当新样本未可激活RBF神经元时,新样本存储到未分类Uc列表中,计算Uc列表中样本的局部密度. 判断是否存在样本局部密度大于密度阈值  $\rho_o$ ,如

果存在则创建一个新的RBF神经元,该样本为神经元中心.

### 3) RBF神经元融合.

当 $k$ 时刻模块化神经网络具有两个或两个以上RBF神经元时,计算各神经元中心之间的距离,如果两神经元之间距离小于距离阈值 $d_c$ ,则融合两神经元.

## 2.2 具有固定记忆的在线梯度下降法

OAMNN子网络模块采用一种改进的具有固定记忆的在线梯度下降法进行学习.在线梯度下降法(随机梯度下降法)已有较多研究<sup>[19-20]</sup>,但现有的在线梯度下降法对样本输入顺序有一定要求,且稳定性较差.本文根据在线顺序极端学习机(OS-ELM)<sup>[21]</sup>学习算法,采用逐个或逐块地学习具有固定大小的变化数据机制,改进在线梯度下降法,使子网络稳定收敛,削弱样本顺序的影响.

图3为具有固定记忆的逐块学习示意图.在梯度下降学习过程中,按数据块进行迭代学习,每块数据的大小固定为 $N$ ,并以大小为 $L$ 的数据进行遗忘.随着新样本不断输入,当新样本积累大小为 $L$ 的数据块时,固定大小为 $N$ 的遗忘块前端大小为 $L$ 的数据块,后端加入新样本数据块,并进行1个迭代.图3中 $t$ 时刻,网络的一次迭代学习为1个数据块, $t+1$ 时刻网络迭代学习下一个固定大小的数据块,本文实验对样本逐个学习,即 $L=1$ .

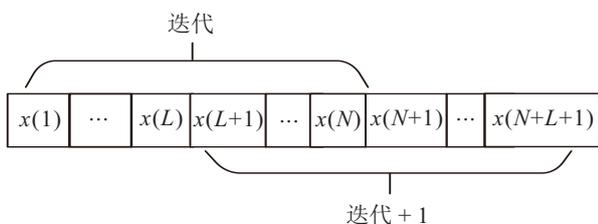


图3 具有固定记忆学习方案

在学习算法中,固定数据块 $N$ 的大小对其性能影响不是很大,一般选取为总样本的2%~5%.样本数较大时, $N$ 的值不是很大;样本数较少时,可根据样本的复杂性选取 $N$ .

OAMNN子网络为3层前馈神经网络,其改进的学习算法权重更新如下:

$$E(t) = \frac{1}{2} \sum_{i=t_0}^t (d(i) - y(i))^2; \quad (7)$$

$$E^p(t) = \frac{1}{2} \sum_{i=t_0}^t e(i)^2 + \frac{1}{2} \lambda \left( \|w(t)\|^2 + \sum_{j=1}^p \|v_j(t)\|^2 \right); \quad (8)$$

$$E_{v_j}^p(t) = \sum_{i=t_0}^t \frac{\partial E^p(i)}{\partial v(i)} =$$

$$\sum_{i=t_0}^t (-e(i)w_j(i)y^{(h)}(i)x(i) + \lambda v_{j,i}(i)); \quad (9)$$

$$E_w^p(t) = \sum_{i=t_0}^t \frac{\partial E^p(i)}{\partial w(i)} = \sum_{i=t_0}^t (-e(i)y^{(h)}(i) + \lambda w(i)); \quad (10)$$

$$v_{:j}(t+1) = v_{:j}(t) - \eta_t E_{v_{:j}}^p(t), \quad j = 1, 2, \dots, p; \quad (11)$$

$$w(t+1) = w(t) - \eta_t E_w^p(t); \quad (12)$$

$$t_0 = \begin{cases} 1, & t \leq N; \\ t - N + 1, & t > N. \end{cases} \quad (13)$$

其中: $e(i) = d(i) - y(i)$ ;  $x(i)$ 为 $i$ 时刻网络的输入; $d(i)$ 为网络期望输出; $y(i)$ 为网络实际输出; $y^{(h)}(i)$ 为隐含层节点的输出; $\lambda$ 为惩罚系数; $N$ 为当前学习的数据块大小; $v_{:j} = [v_{1j}, v_{2j}, \dots, v_{pj}]^T$ 为输入矢量到第 $j$ 个隐含层的权值; $w = [w_1, w_2, \dots, w_p]^T$ 为隐含层到输出层的权值; $p$ 为隐含层节点数; $\eta_t \propto t^{-k}$  ( $0 < k \leq 1$ )为学习率,随时间变化<sup>[22]</sup>.

## 2.3 OAMNN学习过程

OAMNN通过在线任务分解划分输入样本空间,相应地增加或删除子网络模块,从而实现模块化神经网络结构根据实时情况在线调整,具体步骤如下.

step 1: 初始化局部距离阈值 $d_c$ 、类距离阈值 $d_l$ 及密度阈值 $\rho_o$ 参数.

step 2: 随着新样本输入,由式(5)和(6)计算样本的局部密度,当 $k$ 时刻存在样本 $x(i)$ 的局部密度 $\rho_i$ 大于密度阈值 $\rho_o$ 时,创建第1个聚类(RBF神经元),并增加1个相应的子网络模块.本文子网络结构是固定的,如结构为 $n-m-1$ ,表示输入层有 $n$ 个节点,隐含层有 $m$ 个节点,输出层有1个节点.采用所提出的在线梯度下降法对子网络进行训练,聚类中心设为

$$c_1 = x(i). \quad (14)$$

根据类距离阈值 $d_l$ 对样本分类,未分类的数据存储到Uc列表中.

step 3: 第1个聚类确定后,计算得到各聚类中心(RBF神经元中心)的距离 $d$ .

当 $d_i < d_l$ 时,新样本可激活神经元 $i$ ,并将神经元存储到S列表中.

step 4: 根据新样本是否可激活神经元可分为两种情况:

1) 当 $S \neq \emptyset$ 时,新样本可激活一个或多个神经元,分别计算新样本只激活S列表中神经元时的局部密度,选择新样本局部密度最大时激活的神经元为新

样本最终的激活神经元,即新样本属于该神经元的聚类,相应的子网络模块对新样本学习.更新新样本局部区域 $d_c$ 内的样本局部密度(样本局部密度加1),判断更新样本及新样本的局部密度是否存在大于RBF神经元局部密度的样本,如果存在则更新RBF神经元中心.

2) 当 $S = \emptyset$ 时,新样本没有可激活神经元,存储到未分类Uc列表中.

step 5: 计算Uc列表中各样本的局部密度 $\rho$ ,若列表中样本 $j$ 局部密度满足

$$\rho_j > \rho_o, \quad (15)$$

则创建一个新聚类(RBF神经元),相应地创建子网络模块(同step 2),聚类中心为

$$c_{\text{new}} = x(j). \quad (16)$$

对未分类Uc列表中的样本进行分类,未分类的样本存储到未分类Uc列表中.

step 6: 当模块化神经网络具有两个或两个以上RBF神经元时,计算各神经元中心的距离.如果存在 $i, j$ 两神经元中心距离小于距离阈值 $d_c$ ,即

$$\|c_i - c_j\| < d_c, \quad (17)$$

则两神经元进行融合,其新神经元中心 $c_{\text{new}}$ 局部密度为

$$\rho_{\text{new}} = \max(\rho_i, \rho_j). \quad (18)$$

新神经元中心 $c_{\text{new}}$ 为局部密度最大的神经元中心.

神经元合并机制:新神经元中心,局部密度及相应子网络模块的当前训练数据块 $N$ 均与局部密度最大的神经元一致,删除局部密度小的神经元及相应的子网络模块,删除的子网络训练的当前数据块从新输入到新子网络模块进行训练,这样新子网络模块既能保持原来局部密度大的子网络模块学习到的知识,又能学习当前局部密度小的子网络模块知识,从而消除两神经元合并时引起的网络振荡.

step 7: 对 $k$ 时刻激活的子网络进行训练.

step 8: 下一时刻,新样本输入,返回step 3.

## 2.4 OAMNN时间复杂度分析

由第2.3节学习过程步骤,OAMNN的时间复杂度为 $O(N(n_{uc} + l + l^2 + an_i + mN_{io}))$ .其中: $N$ 为OAMNN训练样本, $l$ 为模块化数, $a$ 为 $k$ 时刻被 $x(k)$ 激活的子网络个数, $n_{uc}$ 为未分类的样本数, $n_i$ 为 $i$ 类聚类的样本数, $m$ 为子网络隐含层节点数, $N_{io}$ 为OAMNN输入节点数.

因为step 1的时间复杂度为 $O(0)$ ,step 2的时间

复杂度为 $O(n_{uc})$ ,step 3的时间复杂度为 $O(l)$ ,step 4样本 $x(k)$ 激活 $a$ 个子网络时,时间复杂度为 $O(an_i)$ ,step 5的时间复杂度为 $O(n_{uc})$ ,step 6计算各神经元之间的距离时间复杂度为 $O(l^2)$ ,step 7在 $k$ 时刻对激活的子网络学习的复杂度为 $O(mN_{io})$ .所以整个OAMNN学习过程的时间复杂度可计算为 $O(N(n_{uc} + l + l^2 + an_i + mN_{io}))$ .

OSAMNN的时间复杂度已被证明<sup>[17]</sup>,其值为 $O(N(2l + l^2 + 8a + 2p^2mN_{io}))$ ,其中 $p$ 为以模糊策略选择参与集成的子网络个数.在OAMNN学习过程中,因为分类样本数 $n_{uc}$ 略大于密度阈值 $\rho_o$ ,是一个较小的值, $n_i$ 为激活的聚类的样本数.在任务分解时,OAMNN的时间复杂度比OSAMNN的时间复杂度大 $O(Nan_i)$ ,但OSAMNN采用激活多个子网络策略,所以时间复杂度比OAMNN时间复杂度大 $O(N(8a + (2p^2 - 1)mN_{io}))$ .综合上述分析,对于不同时变系统,OAMNN的时间复杂度约等于或略大于OSAMNN的时间复杂度.

## 3 实验结果及分析

本文选取Mackey-Glass时间序列预测、Lorenz时间序列预测及实际应用中污水处理过程的出水氨氮(NH<sub>3</sub>-N)预测实验对OAMNN性能进行测试,并与SGMN<sup>[16]</sup>、OSAMNN<sup>[17]</sup>及其他在线网络模型进行比较.

选取均方根误差(root mean square error, RMSE)和子网络模块个数作为模块化神经网络性能的评价标准, $t$ 时刻,网络RMSE计算公式为

$$\text{RMSE} = \sqrt{\frac{1}{t - t_0 + 1} \sum_{k=t_0}^t (d(k) - y(k))^2}. \quad (19)$$

其中: $t_0$ 为 $t$ 时刻的数据块大小,同式(13); $d(k)$ 为期望输出; $y(k)$ 为网络输出. RMSE值越小,表示网络预测精度越高.

为了更好地反映网络性能,采用nondimensional error index (NEDI)误差指标评估网络预测的好坏,有

$$\text{NEDI} = \frac{1}{\sigma} \sqrt{\frac{\sum_{k=0}^{t-1} (d(k) - y(k))^2}{t}}, \quad (20)$$

其中 $\sigma$ 为网络输出标准差. RMSE值越小,表示网络预测精度越高, NEDI指标越小,表示网络预测越稳定、越精确.

### 3.1 Mackey-Glass时间序列预测

Mackey-Glass时间序列是一个广泛应用的在线模型基准,被许多学者用来验证预测算法.其时间序

列由一个延迟 Mackey-Glass 微分方程产生,如下所示:

$$\dot{y}(t) = \frac{0.2(\tau - t)}{1 + y^{10}(\tau - t)} - 0.1y(t). \quad (21)$$

实验时  $\tau = 17$ ,采用四阶-龙库塔方法产生 6000 个实验数据,其中  $201 \leq t \leq 3200$  的数据用来训练神经网络,  $5001 \leq t \leq 5500$  的数据用来测试神经网络.  $[y(t); y(t-6); y(t-12); y(t-18)]$  作为输入向量,预测 85 步  $y(t+85)$  的值.

初始化 OAMNN 参数:局部距离阈值  $d_c = 0.4$ ,类距离阈值  $d_l = 0.8$ ,密度阈值  $\rho_o = 4$ ,子网络层中子网络为固定结构 3-5-1(输入层有 3 个节点,隐含层有 5 个节点,输出层有 1 个节点),采用自适应学习率,初始网络权值随机设置,在线梯度下降学习算法采用逐个学习方式,固定数据块  $N = 150$ .

图 4、图 5 分别为 OAMNN 在学习过程中子网络训练 RMSE 变化曲线及网络的预测结果. 图 4 反映了子网络能在线稳定收敛,图 5 可得出该模块化神经网络能准确地预测 Mackey-Glass 时间序列 85 步的值,具有较好的泛化能力.

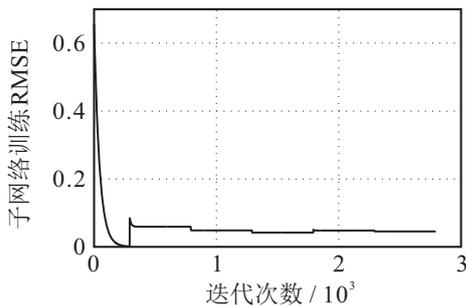


图 4 Mackey-Glass 时间序列预测子网络训练 RMSE

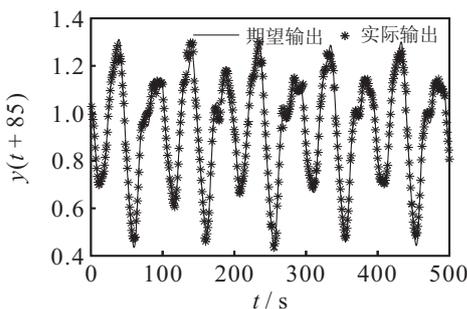
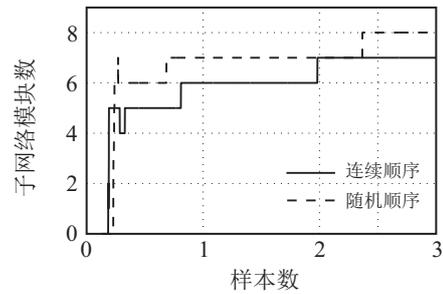


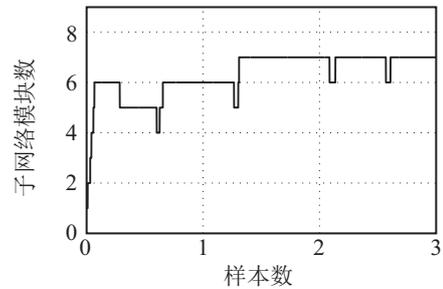
图 5 Mackey-Glass 时间序列预测结果

图 6(a) 为 OAMNN 对连续顺序、随机顺序 Mackey-Glass 时间序列学习过程中子网络模块个数动态变化,图 6(b) 为 OSAMNN 学习过程中子网络模块个数动态变化. 由图 6 可见, OAMNN 能够根据动态数据增加或删减子网络模块(RBF 神经元),实时有效地反映真实数据信息,减少了学习过程中子网络模块不必要的增加和删减.

表 1 为 OAMNN 对 Mackey-Glass 时间序列训练



(a) OAMNN 学习过程



(b) OSAMNN 学习过程

图 6 Mackey-Glass 时间序列子网络模块数动态变化

及测试的性能,“—”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, OAMNN 分别对连续顺序样本、随机顺序样本进行训练、学习,验证了该网络对不同顺序样本都具有很好的性能,并与在线模型 OSAMNN<sup>[17]</sup>、SGMN<sup>[16]</sup>、DPFNN<sup>[23]</sup> 和 ORBF 进行了性能对比. 由实验结果可以得出, OAMNN 与在线模块化神经网络及基于具有移动滑窗的最小二乘法(TLLS)模糊神经网络相比,具有较高的预测精度,且 NDEI 最小.

表 1 Mackey-Glass 时间序列预测实验结果

网络类型	模块数	隐含层节点数	训练 RMSE	测试 RMSE	NDEI
OAMNN 连续顺序	7	35	0.059 2	0.028 1	0.051
OAMNN 随机顺序	8	40	0.061 9	0.023 2	0.042
OSAMNN	7	35	0.032 4	0.031 9	0.263
SGMN	—	18	—	0.044	—
DPFNN	1	11	—	—	0.096
ORBF	1	15	0.046 3	0.095 7	0.312

### 3.2 非线性动态系统辨识

非线性动态系统辨识常被用于验证神经网络性能,有

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t). \quad (22)$$

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

本文通过式(22)产生 1000 个实验数据以验证在线模块化神经网络的性能,其中  $1 \leq t \leq 800$  的数据用来训练神经网络,  $800 < t \leq 1000$  的数据用来测试神

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.

初始化OAMNN参数: 局部距离阈值  $d_c = 0.4$ , 类距离阈值  $d_l = 0.8$ , 密度阈值  $\rho_o = 8$ , 子网络层中子网络为固定结构3-5-1, 采用自适应学习率, 初始网络权重随机设置, 在线梯度下降学习算法采用逐个学习方式, 固定数据块  $N = 20$ .

图7、图8分别为OAMNN在学习过程中子网络训练RMSE变化曲线及网络的预测结果. 图7反映了子网络能在线稳定收敛; 由图8可知该模块化神经网络能准确地预测非线性动态系统辨识1步的值, 具有较好的泛化能力.

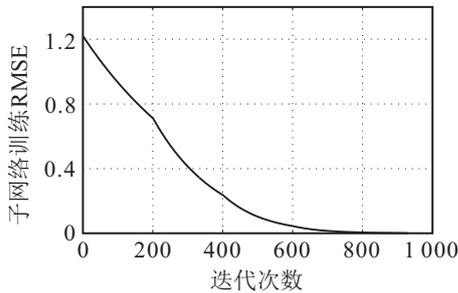


图7 非线性动态系统辨识子网络训练RMSE

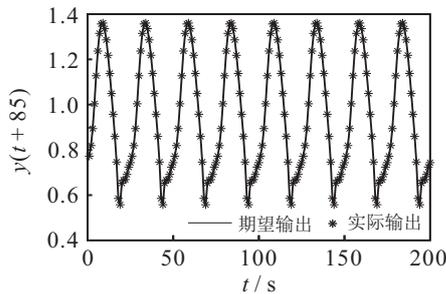
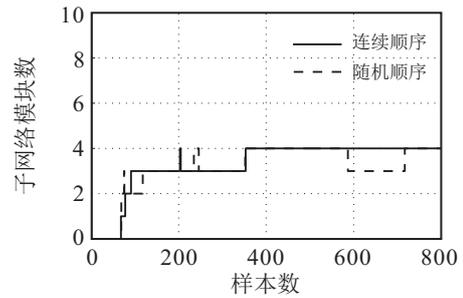


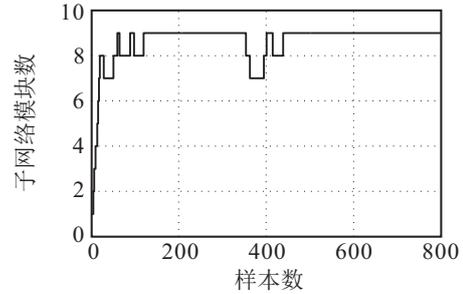
图8 非线性动态系统辨识预测结果

图9(a)为OAMNN对连续顺序、随机顺序非线性动态系统辨识学习过程中子网络模块个数动态变化, 图9(b)为OSAMNN学习过程中子网络模块个数动态变化. 由图9可见, OAMNN能够根据动态数据增加或删除子网络模块, 实时有效地反映真实数据信息, 减少了学习过程中子网络模块不必要的增加或删除, 且最终子网络模块个数(4个)较少, OAMNN具有更为简洁的结构.

表2为OAMNN对非线性动态系统辨识训练及测试的性能, “-”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, OAMNN分别对连续顺序样本、随机顺序样本进行训练、学习, 验证了该网络对不同顺序样本都具有很好的性能, 并与在线模型OSAMNN<sup>[17]</sup>、OLS<sup>[24]</sup>、FAOS-PFNN<sup>[25]</sup>和ORBF进行了性能对比. 由实验结果可以看出, OAMNN与在线模块化神经网络、基于正交最小二乘法的径向基函



(a) OAMNN学习过程



(b) OSAMNN学习过程

图9 非线性动态系统辨识子网络模块数动态变化

表2 非线性动态系统辨识实验结果

网络类型	模块数	隐含层节点数	训练RMSE	测试RMSE
OAMNN 连续顺序	4	20	0.004 1	0.008 4
OAMNN 随机顺序	4	20	0.003 4	0.008 0
OSAMNN	7	35	0.003 4	0.009 6
OLS	1	65	-	0.028 8
FAOS-PFNN	1	5	-	0.025 2
ORBF	1	15	0.066 6	0.069 5

数网络和基于扩展卡尔曼滤波算法的模糊神经网络相比, 具有较高的预测精度, 且结构更加简洁(FAOS-PFNN在线模型结构最简洁).

### 3.3 污水处理过程出水氨氮预测

污水处理是一个具有时变、不稳定、非线性、大滞后等特点的过程, 其中出水氨氮( $\text{NH}_3\text{-N}$ )是重要指标. 测量 $\text{NH}_3\text{-N}$ 浓度有多种方法, 如仪器分析法、电化学分析法等, 但无法满足在污水处理过程中实时检测 $\text{NH}_3\text{-N}$ 浓度的要求, 因此污水处理过程中在线检测出水 $\text{NH}_3\text{-N}$ 浓度仍是一个难题.

实验数据为北京市某污水处理厂2014年采集数据, 除去缺失或异常数据, 共有198组数据. 实验选取150组数据训练模块化神经网络, 48组数据用于测试网络. 分别选取: 温度 $T$ 、入水TP、厌氧末端ORP、好氧前端DO、好氧末端SS、出水pH共6个变量作为输入, 预测出水 $\text{NH}_3\text{-N}$ 浓度.

初始化OAMNN参数: 局部距离阈值  $d_c = 1.0$ , 类距离阈值  $d_l = 1.1$ , 密度阈值  $\rho_o = 3$ , 子网络层中子网络为固定结构6-5-1, 采用自适应学习率, 初始网络权

值随机设置,在线梯度下降学习算法采用逐个学习方式,固定数据块  $N = 30$  (样本复杂性较大).

图10、图11分别为OAMNN在学习过程中子网络训练RMSE变化曲线及网络的预测结果. 图10反映了在训练过程中,子网络模块能在线稳定收敛,由图11可知该模块化神经网络能精确地预测出水氨氮浓度.

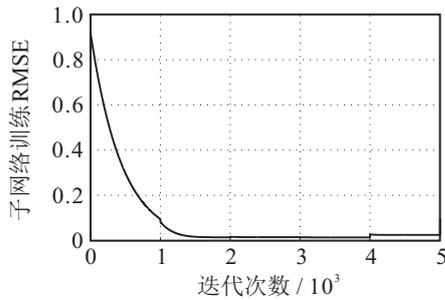


图10 出水氨氮浓度预测子网络训练RMSE

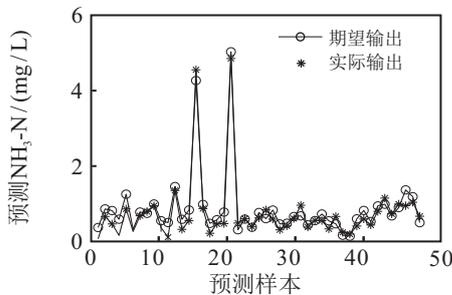
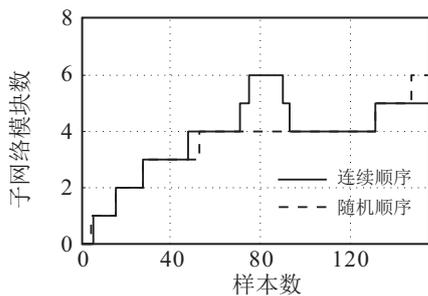
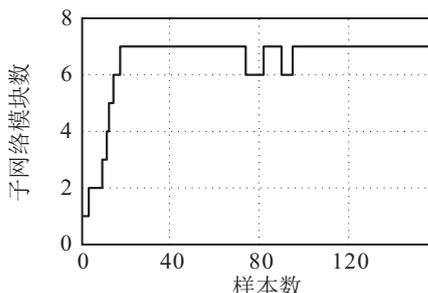


图11 出水氨氮预测结果

图12(a)为OAMNN对连续顺序、随机顺序样本学习过程中子网络模块个数动态变化情况,图12(b)为OSAMNN学习过程中子网络模块个数动态变化情况. 由图12可见,OAMNN能够根据在线出水氨氮



(a) OAMNN学习过程



(b) OSAMNN学习过程

图12 出水氨氮预测子网络模块数动态变化

浓度增加或删减子网络模块,实时有效地反映实际情况中局部数据信息,减少了OAMNN在学习过程中子网络模块不必要的增加或删减,且最终子网络模块个数(5或6)相对较少,OAMNN具有更为简洁的结构.

表3为OAMNN预测出水氨氮浓度的性能,实验结果为独立10次实验的平均值,OAMNN分别对连续顺序样本、随机顺序样本进行训练、学习,验证了该网络对不同顺序样本都具有很好的性能,并与在线模型OSAMNN<sup>[17]</sup>、ASOA-FNN<sup>[26]</sup>和ORBF进行了性能对比. 由实验结果可以看出,OAMNN与在线模块化神经网络、基于自适应二阶算法(ASOA)的模糊神经网络相比,具有较高的预测精度,且结构更加简洁(ASOA-FNN在线模型结构最简洁).

表3 出水氨氮预测实验结果

网络类型	模块数	隐含层节点数	训练RMSE	测试RMSE
OAMNN 连续顺序	5	25	0.254 1	0.209 0
OAMNN 随机顺序	6	30	0.176 1	0.249 3
OSAMNN	7	35	0.102 8	0.264 9
ASOA-FNN	1	15	0.090 8	0.310 1
ORBF	1	20	0.101 7	0.389 5

## 4 结论

本文针对在模块化神经网络对在线任务分解时难以有效反映子任务实时情况及样本间的信息,提出了一种改进的自适应模块化神经网络. 该网络设计了基于距离与密度的在线聚类对任务在线分解,能实时更新样本的局部密度及聚类中心(RBF神经元中心),实现OAMNN根据动态数据自组织网络结构,且有效地减少了子网络模块不必要的增加或删减,网络具有更加紧凑的结构. 另外,针对样本顺序对在线梯度下降法的限制,提出一种固定记忆的在线梯度下降法,实现了OAMNN对不同顺序样本都能稳定收敛,与其他算法相比,具有较高的预测精度.

## 参考文献(References)

- [1] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. *Neural Networks*,1989, 2(5): 359-366.
- [2] Chen W C, Tseng L Y, Wu C S. A unified evolutionary training scheme for single and ensemble of feedforward neural network[J]. *Neurocomputing*, 2014, 143: 347-361.
- [3] Qiao J F, Li S Y, Han H G, et al. An improved algorithm for building self-organizing feedforward neural networks[J]. *Neurocomputing*, 2017, 262(1): 28-40.
- [4] Zhang H, Wu W, Liu F, et al. Boundedness and

- convergence of online gradient method with penalty for feedforward neural networks[J]. *IEEE Transactions on Neural Networks*, 2009, 20(6): 1050-1054.
- [5] 张昭昭. 动态自适应模块化神经网络结构设计[J]. *控制与决策*, 2014, 29(1): 64-70.  
(Zhang Z Z. Dynamic adaptive modular neural network architecture design[J]. *Control and Decision*, 2014, 29(1): 64-70.)
- [6] Hapel B L M, Murre J M J. Design and evolution of modular neural network architectures[J]. *Neural Networks*, 1994, 7(6/7): 985-1004.
- [7] Kai O E, Mouret J B, Clune J. Neural modularity helps organisms evolve to learn new skills without forgetting old skills [J]. *Plos Computational Biology*, 2015, 11(4): 1-24.
- [8] Sánchez D, Melin P, Castillo O. Optimization of modular granular neural networks using a firefly algorithm for human recognition[J]. *Engineering Applications of Artificial Intelligence*, 2017, 64(1): 172-186.
- [9] Melin P, Sánchez D. Multi-objective optimization for modular granular neural networks applied to pattern recognition[J]. *Information Sciences*, 2018, 460/461(1): 594-610.
- [10] Hu P. A classifier of matrix modular neural network to simplify complex classification tasks[J]. *Neural Computer & Application*, 2018, DOI: 10.1007/s00521-018-3631-x.
- [11] Sánchez D, Melin P. Design of intelligent systems based on fuzzy logic, neural networks and nature-inspired optimization [M]. Cham: Springer, 2018: 141-145.
- [12] Qiao J F, Lu C, Li W J. Design of dynamic modular neural network based on adaptive particle swarm optimization algorithm[J]. *IEEE Access*, 2018, 6(1): 10850-10857.
- [13] Valdez F, Castillo O, Melin P. Ant colony optimization for the design of modular neural networks in pattern recognition[C]. 2016 International Joint Conference on Neural Networks. Canada: IEEE, 2016: 24-29.
- [14] Uriarte A, Melin P, Valdez F. Optimization of modular neural network architectures with an improved particle swarm optimization algorithm[M]. New York: Recent Developments and the New Direction in Soft-Computing Foundations and Applications, 2018: 165-174.
- [15] Ramamurti V, Ghosh J. Structurally adaptive modular networks for nonstationary environments[J]. *IEEE Transactions on Neural Networks*, 1999, 10(1): 152-160.
- [16] Loo C K, Rajeswari M, Rao M V C. Novel direct and self-regulating approaches to determine optimum growing multi-experts network structure[J]. *IEEE Transactions on Neural Networks*, 2004, 15(6): 1378-1395.
- [17] Qiao J F, Zhang Z Z, Bo Y C. An online self-adaptive modular neural network for time-varying systems[J]. *Neurocomputing*, 2014, 125(3): 7-16.
- [18] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. *Science*, 2014, 344(6191): 1492-1496.
- [19] Cao X R. A basic formula for online policy gradient algorithms[J]. *IEEE Transactions on Automatic Control*, 2005, 50(5): 696-699.
- [20] Zhang H, Tang Y. Online gradient method with smoothing  $l_0$  regularization for feedforward neural networks[J]. *Neurocomputing*, 2017, 224(1): 1-8.
- [21] Wang X Y, Han M. Online sequential extreme learning machine with kernels for nonstationary time series prediction[J]. *Neurocomputing*, 2014, 145: 90-97.
- [22] Zhang H S, Wu W, Liu F, et al. Boundedness and convergence of online gradient method with penalty for feedforward neural networks[J]. *IEEE Transactions on Neural Networks*, 2009, 20(6): 1050-1054.
- [23] Pratama M, Er M J, Li X, et al. Data driven modeling based on dynamic parsimonious fuzzy neural network[J]. *Neurocomputing*, 2013, 110(1): 18-28.
- [24] Chen S, Cowan C F N, Grant P M. Orthogonal least squares learning algorithm for radial basis function networks[J]. *IEEE Transactions on Neural Networks*, 1991, 2(2): 302-309.
- [25] Wang N, Er M J, Meng X Y. A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks[J]. *Neurocomputing*, 2009, 72(1): 3818-3829.
- [26] Han H G, Ge L M, Qiao J F. An adaptive second order fuzzy neural network for nonlinear system modeling[J]. *Neurocomputing*, 2016, 214(1): 837-847.

## 作者简介

郭鑫(1990—), 男, 博士生, 从事模块化神经网络的研究, E-mail: guo\_0xin@163.com;

李文静(1985—), 女, 副教授, 博士, 从事神经计算、神经网络建模等研究, E-mail: wenjing.li@bjut.edu.cn;

乔俊飞(1968—), 男, 教授, 博士生导师, 从事智能控制与智能信息处理、复杂过程建模与优化控制等研究, E-mail: junfeiq@bjut.edu.cn.

(责任编辑: 郑晓蕾)