

DOI: 10.11992/tis.201702014

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20170702.0439.024.html>

基于快速密度聚类的 RBF 神经网络设计

蒙西^{1,2}, 乔俊飞^{1,2}, 李文静^{1,2}

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

摘 要: 针对径向基函数 (radial basis function, RBF) 神经网络隐含层结构难以确定的问题, 提出一种基于快速密度聚类的网络结构设计算法。该算法将快速密度聚类算法良好的聚类特性用于 RBF 神经网络结构设计中, 通过寻找密度最大的点并将其作为隐含层神经元, 进而确定隐含层神经元个数和初始参数; 同时, 引入高斯函数的特性, 保证了每个隐含层神经元的活性; 最后, 用一种改进的二阶算法对神经网络进行训练, 提高了神经网络的收敛速度和泛化能力。利用典型非线性函数逼近和非线性动态系统辨识实验进行仿真验证, 结果表明, 基于快速密度聚类设计的 RBF 神经网络具有紧凑的网络结构、快速的学习能力和良好的泛化能力。

关键词: RBF 神经网络; 快速密度聚类; 结构设计; 神经元活性; 二阶算法; 泛化能力; 函数逼近; 系统辨识

中图分类号: TP273 **文献标志码:** A **文章编号:** 1673-4785(2018)03-0331-08

中文引用格式: 蒙西, 乔俊飞, 李文静. 基于快速密度聚类的 RBF 神经网络设计[J]. 智能系统学报, 2018, 13(3): 331-338.

英文引用格式: MENG Xi, QIAO Junfei, LI Wenjing. Construction of RBF neural networks via fast density clustering[J]. CAAI transactions on intelligent systems, 2018, 13(3): 331-338.

Construction of RBF neural networks via fast density clustering

MENG Xi^{1,2}, QIAO Junfei^{1,2}, LI Wenjing^{1,2}

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

Abstract: To design a hidden layer structure in radial-basis-function (RBF) neural networks, a novel algorithm based on fast density clustering is proposed. The algorithm searches for the point with the highest density and then uses it as the neuron of the hidden layer, thereby ascertaining the number of neurons in the hidden layer and the initial parameters. Moreover, the activity of each hidden neuron is ensured by introducing the Gaussian function. An improved second-order algorithm is used to train the designed network, increasing the training speed and improving the generalization performance. In addition, two benchmark simulations—the typical nonlinear function approximation and the nonlinear dynamic system identification experiment—are used to test the effectiveness of the proposed RBF neural network. The results suggest that the proposed RBF neural network based on fast density clustering offers improved generalization performance, has a compact structure, and requires shorter training time.

Keywords: RBF neural networks; fast density clustering; structure design; neuron activity; second-order training; generalization performance; function approximation; system identification

径向基函数 (radial basis function, RBF) 神经网络是一种模拟生物神经元局部响应特性的前馈神经网络

网络, 因其结构简单且具有良好的非线性映射能力而被广泛用于多个领域^[1-6]。RBF 神经网络构建的核心问题在于结构设计^[7]。早期隐含层结构的确定多采用经验试凑法, 但此类方法很难在保证神经网络精度的前提下得到一个紧凑的网络结构。因此, 许多学者针对 RBF 神经网络的结构设计问题展开

收稿日期: 2017-02-24. 网络出版日期: 2017-07-02.

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面上项目 (4182007); 北京工业大学日新人才项目 (2017-RX-(1)-04).

通信作者: 乔俊飞. E-mail: junfeiq@bjut.edu.cn.

了大量研究。

聚类算法常被用来确定 RBF 神经网络的结构, 如 K-Means、Fuzzy C-Means (FCM)^[8-10]等, 该类算法能根据各聚类中心之间的距离确定隐含层神经元的径向作用范围, 但却无法确定隐含层神经元的个数, 会影响神经网络的泛化能力。因此, 根据研究对象自适应设计 RBF 神经网络中隐含层神经元的个数, 提高网络性能已成为当前 RBF 神经网络研究的一个热点。Platt^[11]首次提出了一种结构自适应调整的 RBF 神经网络——资源分配网络 (resource allocation network, RAN), 该网络能根据待处理任务动态增加 RBF 神经元, 但容易导致网络结构过大的问题。在 RAN 的基础上, Yingwei 等构建了最小资源神经网络 (minimal resource allocation network, MRAN)^[12]。在自适应增加隐含层神经元的同时, MRAN 能通过删减策略去除冗余神经元, 但该网络收敛速度较慢。Huang 等^[13]提出了一种增长修剪型 RBF 神经网络 (growing and pruning RBF, GAP-RBF), 该网络基于神经元的显著性构建隐含层结构。文献[14]中构建了一种自组织 RBF 神经网络 (self-organizing RBF, SORBF), SORBF 网络基于 RBF 神经元径向作用范围对隐含层结构进行自适应增删。GAP-RBF 网络和 SORBF 网络中涉及的算法参数较多, 对算法最优参数的寻求会影响网络的性能。Wilamowski 等^[15]基于误差补偿算法来构建 RBF 神经网络 (error-correction RBF, ErrCor-RBF), 实验表明该方法能获得较精简的神经网络结构, 但该网络仍然需要通过大量的迭代寻求最优结构。

鉴于以上存在的问题, 结合 RBF 神经元激活函数本身的特性, 本文对一种快速密度聚类算法进行了相应的改进, 然后将其用于 RBF 神经网络的结构设计中。同时, 针对传统梯度下降、算法收敛较慢且易陷入局部极小的问题, 结合结构设计中确定的初始参数, 选用改进的二阶梯度算法来训练 RBF 神经网络, 提高了网络的收敛速度和泛化能力。最后, 通过两个基准仿真实验验证, 提出的基于快速密度聚类的 RBF 神经网络 (fast density clustering RBF, FDC-RBF), 能够以紧凑的网络结构和较快的收敛速度获取较好的非线性映射能力。

1 RBF 神经网络

RBF 神经网络是一种典型的前馈型神经网络, 其网络结构如图 1 所示 (L 个输入神经元, J 个隐含层神经元, M 个输出层神经元)。隐含层激活函数为径向基函数, 常选用标准的高斯函数, 即

$$\theta_j(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{\sigma_j^2}} \quad (1)$$

其中: $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_L]^T$ 为网络输入向量, \mathbf{c}_j 为第 j 个神经元的中心向量, σ_j 为第 j 个神经元的径向作用范围。输出层第 m 个神经元的输出 y_m 为

$$y_m = \sum_{j=1}^J w_{jm} \theta_j \quad (2)$$

式中: w_{jm} 为第 j 个隐含层神经元到第 m 个输出神经元的连接权值。

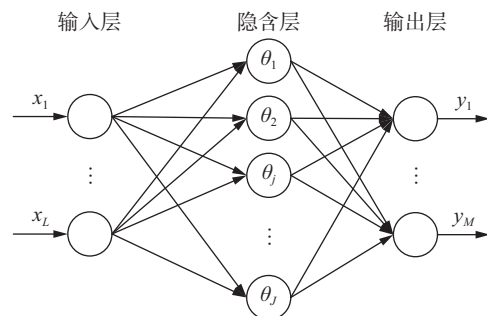


图 1 RBF 神经网络结构图

Fig. 1 Structure of the RBF neural network

2 基于快速密度聚类的 RBF 神经网络设计

RBF 神经网络的设计包括结构构建和参数训练两部分。文中采用改进的快速密度聚类算法确定网络初始结构和初始参数, 在此基础上, 用一种改进的二阶梯度算法对网络进行训练。

2.1 快速密度聚类算法

聚类分析是基于相似度将样本划分成若干类别, 目前大多数聚类算法普遍存在两点不足: 需要提前确定聚类的类别数; 需要通过大量迭代来寻求最优聚类结果。

针对以上问题, Alex 等^[16]提出了一种快速密度聚类算法, 该算法无需提前确定聚类类别数, 通过寻找局部密度峰值作为聚类中心从而实现对数据样本的快速聚类。

快速密度聚类算法的核心思想在于聚类中心被其他密度值较小的点紧紧包围, 且远离其他密度值较大的点。在聚类中心的寻找过程中, 对于任一数据点 i , 需计算两个值: 每个点的局部密度值 ρ_i 和该点到其他密度值较大点的最小距离 δ_i 。数据点 i 的局部密度值计算公式为

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (3)$$

式中: 当 $x < 0$ 时, $\chi(x) = 1$; 当 $x \geq 0$ 时, $\chi(x) = 0$; d_c 为需要提前设定的截断距离。

数据点 i 到其他密度值较大点的最小距离 δ_i 计算公式为

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}) \quad (4)$$

选出局部密度值较大和到其他点距离较小的点作为聚类中心,其他非中心数据样本依次分配到距离其最近且密度值较大的聚类中心所在类,由此完成整个聚类过程。

可见,该算法的聚类过程是一步完成的无需通过多次迭代来寻求最优结果。但其仍然存在两点不足:聚类前需要获取整个数据样本,因而不利于实现在线聚类;聚类的效果受到截断距离 d_c 的影响。

针对以上问题,结合高斯函数的特性,对该算法进行了一定的改进,并将其运用于RBF神经网络的结构设计中。

2.2 基于快速密度聚类算法的RBF网络结构设计

类似于快速密度聚类算法,本文通过寻找局部密度值较大的点作为隐含层神经元中心,进而确定RBF神经网络的结构。针对快速密度聚类算法需要预先知道所有的数据样本且聚类效果受到截断距离影响的问题,本文做出相应的改进。

1) 由于RBF神经元的激活函数为高斯函数,引入隐含层神经元活性评价指标如下:

$$AC_{ij} = e^{-\|x_i - c_j\|^2 / \sigma_j^2} \geq V \quad (5)$$

式中: AC_{ij} 为第 j 个隐含层神经元被第 i 个样本激活后的活性, AC 值越大,神经元的活性越强; V 为神经元活性阈值,以保证隐含层神经元的活性足够大。输入向量、中心向量、径向作用范围需要满足以下关系:

$$\frac{\|x_i - c_j\|}{\sigma_j} \leq \sqrt{\ln V} \quad (6)$$

即输入向量与隐含层神经元的中心向量间的距离需要满足以下关系:

$$\|x_i - c_j\| \leq \sigma_j \sqrt{\ln V} \quad (7)$$

式中神经元活性阈值 V 根据实验进行取值。因此截断距离与神经元的径向作用范围以及神经元的活性相关。

2) 为了实现在线聚类,本文在确定神经网络结构时,训练样本依次进入神经网络对结构进行调整:增加一个隐含层神经元或对已有的隐含层神经元进行调整。

文中隐含层结构设计的核心思想在于:判断当前样本在激活其最近隐含层神经元时是否能保证该神经元具有足够的活性。如果能保证其活性则能归入当前隐含层神经元所在类,反之则不能;其次,通

过密度比较,将密度较大的点作为隐含层神经元。

因此文中基于快速密度聚类的RBF神经网络结构设计可以分为两种情况:神经元增长机制;神经元调节机制。

设神经网络的训练样本是由 P 个输入输出对 (x, y_d) 组成,其中, x 为 L 维输入向量, y_d 为相对应的 M 维期望输出。初始时刻,网络隐含层的神经元个数为 0。

① 神经元增长机制

把第一个数据样本作为第一个隐含层神经元中心,同时设定相应的径向作用范围和输出权值。

$$c_1 = x_1 \quad (8)$$

$$\sigma_1 = 1 \quad (9)$$

$$w_1 = y_{d1} \quad (10)$$

在 k 时刻,假设已经存在 j 个隐含层神经元,当第 k 个数据样本进入网络时,找到距离当前样本最近的隐含层神经元 k_{\min} :

$$k_{\min} = \arg \min_{h \in [1, j]} \{\text{dist}(x_k, c_h)\} \quad (11)$$

$$D = \text{dist}(x_k, c_{k_{\min}}) \quad (12)$$

将此距离与该隐含层神经元的径向作用范围进行比较,如果 $D > \sigma_{k_{\min}} \sqrt{\ln V}$,则认为当前样本不能保证该神经元的活性,将第 k 个样本作为新增神经元的中心,同时设定其径向作用范围和输出权值:

$$c_{j+1} = x_k \quad (13)$$

$$\sigma_{j+1} = 1 \quad (14)$$

$$w_{j+1} = y_{dk} \quad (15)$$

② 神经元调整机制

在 k 时刻,若 $D \leq \sigma_{k_{\min}} \sqrt{\ln V}$,则认为当前网络能够对新样本进行学习,比较当前样本与该隐含层神经元的局部密度值,选出密度值较大的点作为新的隐含层神经元,数据点 i 的局部密度计算公式为

$$P_i = \sum_{c_i \neq x_j} \exp(-\|c_i - x_j\|^2 / d_i^2) \quad (16)$$

式中: x_j 是 c_i 作用范围内所包括的样本点; d_i 是该数据点的局部作用范围。

从式 (16) 可以看出,若一个数据点的局部密度值越大,代表该点附近聚集的样本点越多;同理,隐含层神经元密度越大,则代表该神经元激活的样本数越多。

将当前输入样本点 k 的局部密度值与隐含层神经元 k_{\min} 的局部密度值进行比较:

若 $P_k > P_{k_{\min}}$,则当前输入样本替换已有的隐含层神经元,成为新的隐含层神经元,初始参数设置为

$$c_{k_{\min}} = x_k \quad (17)$$

$$w_{k_{\min}} = \frac{n_{k_{\min}} w_{k_{\min}} + y_{dk}}{n_{k_{\min}} + 1} \quad (18)$$

$$\sigma_{k_{\min}} = \max(\text{dist}(c_{k_{\min}}, x_{k_{\min}})) \quad (19)$$

式中: $\mathbf{x}_{k_{\min}}$ 表示第 k_{\min} 个隐含层神经元激活的所有样本; $n_{k_{\min}}$ 表示激活的样本数量。

反之, 若 $P_j \leq P_{k_{\min}}$, 则已有的隐含层神经元不变, 只需调整该神经元的径向作用范围以及到输出层的连接权值, 见式 (18) ~ (19)。

由此, 可以得到 FDC-RBF 神经网络结构设计算法如下:

- 1) 初始时刻, 隐含层神经元个数为 0。
- 2) 当第 1 个样本进入网络后, 将其作为第 1 个隐含层神经元, 并按照式 (8) ~ (10) 对其中心、径向作用范围和连接权值进行设置。
- 3) 当第 k 个样本进入网络后, 计算其与当前所有隐含层神经元的距离, 找出与第 k 个样本距离最近的隐含层神经元 k_{\min} 。
- 4) 判断输入样本是否能保证第 k_{\min} 个神经元的活性, 若不能保证则为网络新增加一个隐含层神经元, 并按照式 (13) ~ (15) 赋予初始参数, 然后转向 6); 否则, 执行下一步。
- 5) 若能保证神经元活性, 则比较当前样本与最近隐含层神经元的局部密度值, 选出密度值较大的点作为新的隐含层神经元, 按照式 (17) ~ (19) 对中心、径向作用范围和权值进行更新, 转向 6)。
- 6) 若所有的样本比较完毕, 则神经网络结构确定; 否则, $k=k+1$, 转向 3)。

该算法将快速密度聚类的思想用于 RBF 网络结构设计中, 并结合高斯函数特性进行相应改进, 使网络具有紧凑的结构。同时, 结构构建过程中设定的较优的初始参数又能提高网络的收敛速度。

2.3 神经网络学习算法

确定网络结构后, 需要对网络参数进行调整。本文用一种改进的二阶算法对 RBF 网络进行训练, 提高了网络的收敛速度和泛化能力。传统的 LM 算法更新规则如式 (20) 所示^[17]:

$$\mathbf{A}_{k+1} = \mathbf{A}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{e}_k \quad (20)$$

式中: \mathbf{A} 指所有需要调整的参数 (中心向量、径向作用范围、连接权值); \mathbf{J} 为雅克比矩阵; \mathbf{e} 为误差向量; \mathbf{I} 是单位矩阵; μ 为学习率参数。

误差向量计算公式为

$$\mathbf{e} = [e_{11} \ e_{12} \dots e_{1P} \ e_{21} \dots e_{PM}]^T \quad (21)$$

$$e_{pm} = y_{dpm} - y_{pm} \quad (22)$$

式中: P 是样本数量, M 是输出向量维数, y_{dpm} 和 y_{pm} 分别是第 p 个样本进入时第 m 个输出神经元对应的期望输出和实际输出。

雅克比矩阵计算公式为

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_{11}}{\partial \mathbf{A}_1} & \frac{\partial e_{11}}{\partial \mathbf{A}_2} & \dots & \frac{\partial e_{11}}{\partial \mathbf{A}_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{1M}}{\partial \mathbf{A}_1} & \frac{\partial e_{1M}}{\partial \mathbf{A}_2} & \dots & \frac{\partial e_{1M}}{\partial \mathbf{A}_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{P1}}{\partial \mathbf{A}_1} & \frac{\partial e_{P1}}{\partial \mathbf{A}_2} & \dots & \frac{\partial e_{P1}}{\partial \mathbf{A}_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{PM}}{\partial \mathbf{A}_1} & \frac{\partial e_{PM}}{\partial \mathbf{A}_2} & \dots & \frac{\partial e_{PM}}{\partial \mathbf{A}_N} \end{bmatrix} \quad (23)$$

式中 N 是算法中所有参数的个数。

由式 (23) 可以看出, 在 LM 算法的执行过程中, 雅克比矩阵的计算与训练样本数量、参数个数以及输出向量的维数都有关。当样本数量过多时, 则会影响算法的收敛速度。

针对以上问题, 本文用一种改进的二阶算法对 RBF 网络进行训练, 提高了神经网络的收敛速度。改进二阶算法的更新规则为^[18]

$$\mathbf{A}_{k+1} = \mathbf{A}_k - (\mathbf{Q}_k + \mu_k \mathbf{I})^{-1} \mathbf{g}_k \quad (24)$$

式中: \mathbf{A} 仍指所有需要调整的参数 (中心向量, 径向作用范围, 连接权值); \mathbf{Q} 为类海森矩阵; \mathbf{g} 为梯度向量; μ 为学习率参数。

为了克服传统 LM 算法中存在的不足, 减小存储空间, 提高收敛速度, 将类海森矩阵的计算转化为 $P \times M$ 个子矩阵的和, 如式 (25)、(26) 所示:

$$\mathbf{Q} = \sum_{p=1}^P \sum_{m=1}^M \mathbf{q}_{pm} \quad (25)$$

$$\mathbf{q}_{pm} = \mathbf{j}_{pm}^T \mathbf{j}_{pm} \quad (26)$$

同样, 将梯度向量的计算也转化为 $P \times M$ 个子向量的和:

$$\mathbf{g} = \sum_{p=1}^P \sum_{m=1}^M \boldsymbol{\eta}_{pm} \quad (27)$$

$$\boldsymbol{\eta}_{pm} = \mathbf{j}_{pm}^T \mathbf{e}_{pm} \quad (28)$$

这样, 对类海森矩阵和梯度向量的计算就转化为对雅克比分量的计算:

$$\mathbf{j}_{pm} = \begin{bmatrix} \frac{\partial e_{pm}}{\partial w_{jm}} & \dots & \frac{\partial e_{pm}}{\partial \sigma_j} & \dots & \frac{\partial e_{pm}}{\partial c_{jm}} & \dots \end{bmatrix} \quad (29)$$

联立式 (1)、(2)、(24), 对雅克比分量中每个参数的偏微分计算如下:

$$\frac{\partial e_{pm}}{\partial w_{jm}} = -\frac{\partial y_{dpm}}{\partial w_{jm}} = -\theta_j(\mathbf{x}_p) \quad (30)$$

$$\frac{\partial e_{pm}}{\partial \sigma_j} = -\frac{\partial y_{dpm}}{\partial \sigma_j} = -\frac{\partial y_{dpm}}{\partial \theta_j(\mathbf{x}_p)} \frac{\partial \theta_j(\mathbf{x}_p)}{\partial \sigma_j} = \frac{2w_{jm}\theta_j(\mathbf{x}_p)\|\mathbf{x}_p - \mathbf{c}_j\|^2}{\sigma_j^3} \quad (31)$$

$$\frac{\partial e_{pm}}{\partial c_{jm}} = -\frac{\partial y_{dpm}}{\partial c_{jm}} = -\frac{\partial y_{dpm}}{\partial \theta_j(\mathbf{x}_p)} \frac{\partial \theta_j(\mathbf{x}_p)}{\partial c_{jm}} = \frac{2w_{jm}\theta_j(\mathbf{x}_p)\|\mathbf{x}_p - \mathbf{c}_j\|}{\sigma_j^2} \quad (32)$$

训练过程中,学习率参数 μ 随着误差的变化自适应调整。如果后一时刻的误差小于前一时刻的误差,则 μ 值减小,并保留当前训练后的各参数值;反之,则 μ 值增大,各参数值恢复到调整前的值。

3 仿真实验

文中选用非线性函数逼近和非线性动态系统辨识两个基准实验对FDC-RBF网络进行仿真验证,并与其他算法进行了对比。

3.1 非线性函数 $\sin E$ 逼近

选取的典型非线性函数 $\sin E$ 为

$$y = 0.8 \exp(-0.2x) \sin(10x) \quad (33)$$

式中 $0 \leq x \leq 2$ 。非线性函数 $\sin E$ 经常被用来检验RBF神经网络的性能。

随机选取200个样本对神经网络进行训练,200个样本进行测试,训练的期望均方误差(mean square error, MSE)设为0.01,神经元活性阈值 V 设为0.55。用改进的二阶算法训练后的神经网络测试效果如图2所示,隐含层神经元数的变化如图3所示,二阶算法学习性能曲线如图4所示。此外,为了验证不同的学习算法对神经网络性能的影响,在RBF网络初始结构确定后,又用梯度下降法对该网络进行了训练,学习性能曲线如图5所示。

由图2可以看出,训练后的FDC-RBF网络能够较好地拟合 $\sin E$ 曲线。从图4和图5的对比可以看出,梯度下降算法需要在第45步左右才能完全收敛,而改进的二阶算法在第13步就已完全收敛。由此可以看出,当用改进的二阶算法训练该网络时,收敛速度大大提高了。

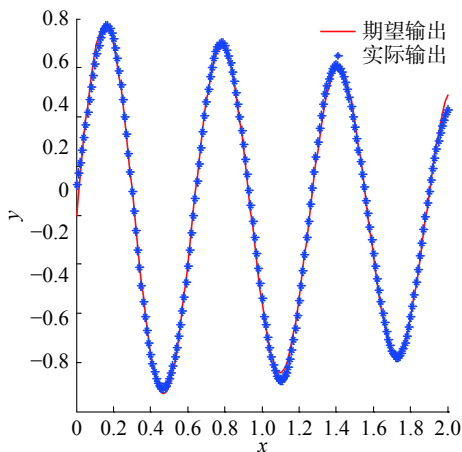


图2 函数逼近效果图

Fig. 2 Results of the function approximation problem

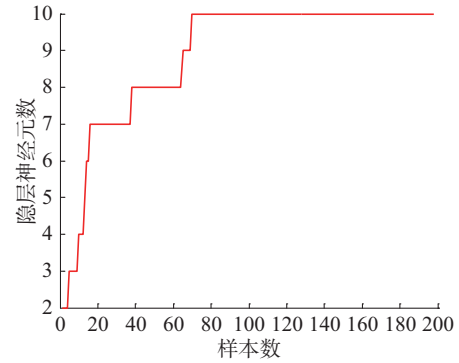


图3 隐含层神经元变化曲线

Fig. 3 Structure construction process of the FDC-RBF network

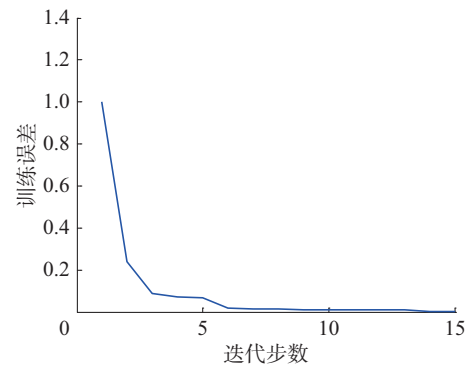


图4 二阶算法学习性能曲线

Fig. 4 Performance curve of the second-order learning algorithm

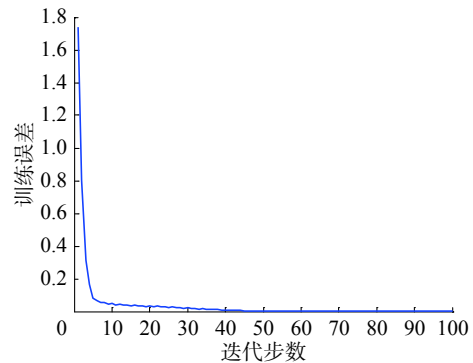


图5 梯度下降学习性能曲线

Fig. 5 Performance curve of the gradient descent algorithm

同时,为了显示FDC-RBF网络的优良性,本文将其与DFNN^[19]、GAP-RBF^[13]、SORBF^[14]、ART-RBF^[20]进行了对比,如表1所示。结果显示,与DFNN、GAP-RBF、SORBF相比,基于本文提出的算法设计的网络隐含层神经元个数要少于以上网络,结构要更为精简;同时,与其他4种网络相比,FDC-RBF网络收敛速度更快,训练时间更短;此外,从测试误差的对比可以看出,FDC-RBF网络的测试误差也要小于其他算法。因此可以得出,FDC-RBF神经网络在逼近该非线性函数时,网络结构较其他算法更为精简,收敛速度更快,泛化能力更好。

表1 函数逼近实验结果对比

Table 1 Performance comparison for the function approximation problem

网络类型	训练时间/s	测试 MSE	隐含层神经元数/个
DFNN	62.32	0.086 1	25
GAP-RBF	26.86	0.041 5	19
SORBF	22.67	0.024 18	12
ART-RBF	13.86	0.013 7	10
FDC-RBF	6.52	0.010 9	10

3.2 非线性系统动态辨识

非线性动态系统辨识常被用来验证所设计的神经网络的性能,文中的非线性系统由式(34)表示:

$$y(t+1) = \frac{y(t)y(t-1)(y(t)+2.5)}{1+y^2(t)+y^2(t-1)} + u(t) \quad (34)$$

式中: $u(t) = \sin(2\pi t/25)$, $t \in [1, 400]$, $y(0) = 0$, $y(1) = 0$ 。选取 100 个点作为训练样本, 其中 $t \in [1, 100]$, 另外选取 100 个点作为测试样本, 其中 $t \in [301, 400]$, 训练的期望均方误差 MSE 设为 0.01, 神经元活性阈值 V 设为 0.33。FDC-RBF 网络的辨识结果如图 6 所示, 其中用改进的二阶算法训练的神经网络学习性能曲线如图 7 所示, 隐含层神经元个数的变化如图 8 所示。

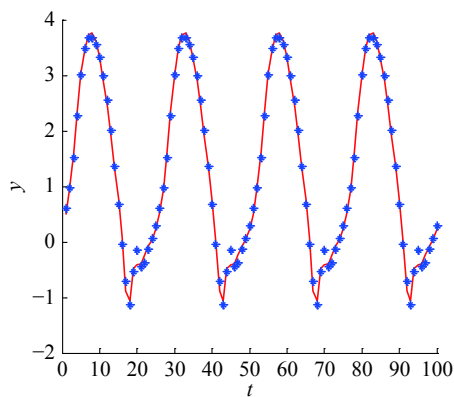


图6 系统辨识结果

Fig. 6 Results of the system identification problem

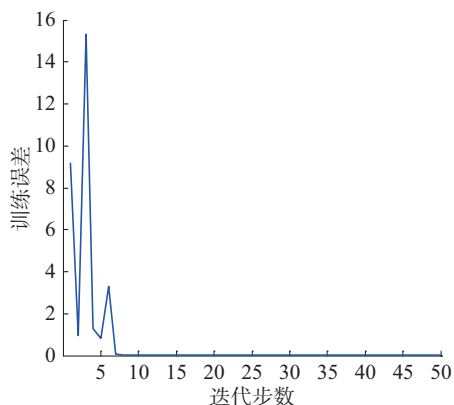


图7 二阶算法学习性能曲线

Fig. 7 Performance curve of the second-order learning algorithm

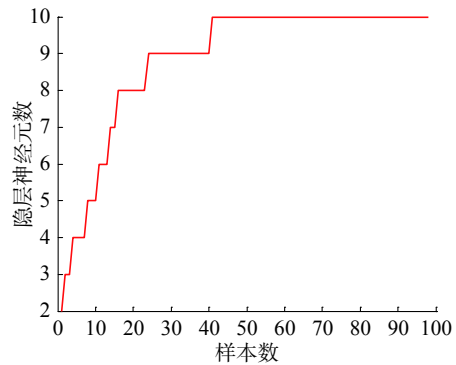


图8 隐含层神经元变化曲线

Fig. 8 Structure construction process of the FDC-RBF network

由图 6 可以看出, FDC-RBF 网络能够较好地辨识该非线性系统。同时, 从图 7 的学习性能曲线变化可以看到, 训练过程中, 当迭代到第 8 步后网络完全收敛, 由此可以看出, 所设计网络的收敛速度非常快。

此外, 本文还将 FDC-RBF 网络的非线性系统辨识结果与其他算法进行了对比, 结果如表 2 所示。

表2 系统辨识实验结果对比

Table 2 Performance comparison for the dynamic system identification problem

网络类型	训练时间/s	测试 MSE	隐含层神经元个数
GAP-RBF	49.38	0.021 2	12
DFNN	153.24	0.026 4	15
SORBF	31.26	0.011 2	7
ART-RBF	22.17	0.010 0	7
FDC-RBF	15.26	0.009 6	10

从对比结果可以看出, 与其他网络相比, FDC-RBF 网络的训练时间要优于其他网络。同时, 即使在期望的训练误差设为 0.01 的情况下, 测试误差仍然能达到 0.009 6, 由此可以看到, 该网络的辨识效果要好于其他网络。虽然隐含层神经元个数要多于 SORBF 和 ART-RBF, 但网络结构却仍然要比 GAP-RBF 与 DFNN 精简。由此可以得出, 在用 FDC-RBF 神经网络进行非线性动态系统辨识时, 能够在保证结构精简的前提下用较快的时间达到最好的泛化能力。

4 结束语

针对 RBF 网络的结构设计问题, 本文将快速密度聚类算法进行了相应改进, 在保证神经元活性的前提下通过寻找局部密度值较大的点来确定神经网络结构; 同时, 基于结构设计所确定的初始参数, 用

改进的二阶梯度算法对所设计的神经网络进行训练;最后,通过非线性函数逼近和非线性系统辨识两个基准实验进行仿真验证得到以下结论:

1) 文中提出的RBF网络结构设计算法不依赖于全部的训练样本,能够根据实际任务自适应调整网络结构;

2) 通过保证神经元活性和寻求密度最大的点作为隐含层神经元,最终获得的RBF神经网络结构精简且泛化能力好;

3) 结合结构设计中所确定的初始参数,用改进的二阶梯度算法训练神经网络,大大提高了收敛速度,缩短了训练时间。

尽管FDC-RBF网络在非线形函数逼近和非线形系统动态辨识中取得了较好的效果,但其解决实际问题的能力还有待进一步验证。

参考文献:

- [1] CHEN Sheng, WOLFGANG A, HARRIS C J, et al. Symmetric RBF classifier for nonlinear detection in multiple-antenna-aided systems[J]. IEEE transactions on neural networks, 2008, 19(5): 737–745.
- [2] 乔俊飞, 韩红桂. RBF神经网络的结构动态优化设计[J]. 自动化学报, 2010, 36(6): 865–872.
QIAO Junfei, HAN Honggui. Optimal structure design for RBFNN structure[J]. Acta automatica sinica, 2010, 36(6): 865–872.
- [3] BARANDIARAN X, MORENO A. On the nature of neural information: a critique of the received view 50 years later[J]. Neurocomputing, 2008, 71(4/5/6): 681–692.
- [4] 侯远杭, 黄胜, 梁霄. PSO训练的弹性RBFNN在船型优化中的应用研究[J]. 哈尔滨工程大学学报, 2017, 38(2): 175–180.
HOU Yuanhang, HUANG Sheng, LIANG Xiao. Ship hull optimization based on PSO training FRBF neural network [J]. Journal of Harbin engineering university, 2017, 38(2): 175–180.
- [5] 蒙西, 乔俊飞, 韩红桂. 基于ART的RBF网络结构设计[J]. 控制与决策, 2014, 29(10): 1876–1880.
MENG Xi, QIAO Junfei, HAN Honggui. RBF neural network based on ART neural network[J]. Control and decision, 2014, 29(10): 1876–1880.
- [6] CECATI C, KOLBUSZ J, RÓŻYCKI P, et al. A novel RBF training algorithm for short-term electric load forecasting and comparative studies[J]. IEEE transactions on industrial electronics, 2015, 62(10): 6519–6529.
- [7] 乔俊飞, 韩红桂. 神经网络结构动态优化设计的分析与展望[J]. 控制理论与应用, 2010, 27(3): 350–357.
- [8] TAGLIAFERRI R, STAIANO A, SCALA D. A supervised fuzzy clustering for radial basis function neural networks training[C]//Proceedings of the 9th IFSA World Congress and 20th NAFIPS International Conference. Vancouver, BC, Canada, 2001: 1804–1809.
- [9] RUBIO J J, PACHECO J. An stable online clustering fuzzy neural network for nonlinear system identification[J]. Neural computing and applications, 2009, 18(6): 633–641.
- [10] WANG Di, ZENG Xiaojun, KEANE J A. A clustering algorithm for radial basis function neural network initialization[J]. Neurocomputing, 2012, 77(1): 144–155.
- [11] PLATT J. A Resource-allocating network for function interpolation[J]. Neural computation, 1991, 3(2): 213–225.
- [12] LU Yingwei, SUNDARARAJAN N, SARATCHANDRAN P. A sequential learning scheme for function approximation using minimal radial basis function neural networks[J]. Neural computation, 1997, 9(2): 461–478.
- [13] HUANG Guangbin, SARATCHANDRAN P, SUNDARARAJAN N. An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks[J]. IEEE transactions on systems, man, and cybernetics, part B (Cybernetics), 2004, 34(6): 2284–2292.
- [14] HAN Honggui, CHEN Qili, QIAO Junfei. Research on an online self-organizing radial basis function neural network[J]. Neural computing and applications, 2010, 19(5): 667–676.
- [15] YU Hao, REINER P D, XIE Tiantian, et al. An incremental design of radial basis function networks[J]. IEEE transactions on neural networks and learning systems, 2014, 25(10): 1793–1803.
- [16] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492–1496.
- [17] HAGAN M T, MENHAJ M B. Training feedforward networks with the Marquardt algorithm[J]. IEEE transactions on neural networks, 1994, 5(6): 989–993.
- [18] WILAMOWSKI B M, YU Hao. Improved computation for Levenberg–Marquardt training[J]. IEEE transactions on neural networks, 2010, 21(6): 930–937.
- [19] WU Shiqian, ER M J, GAO Yang. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks[J]. IEEE transactions on fuzzy

systems, 2002, 9(4): 578–594.

- [20] MENG Xi, QIAO Junfei, HAN Honggui. An ART-like algorithm for constructing RBF neural networks[C]//Proceedings of 2015 International Joint Conference on Neural Networks. Killarney, Ireland, 2015.

作者简介:



蒙西, 女, 1988 年生, 博士研究生, 主要研究方向为人工神经网络、类脑智能模型以及智能信息处理。获得授权国家发明专利 1 项。发表学术论文 5 篇, 被 SCI 收录 2 篇, EI 收录 3 篇。



乔俊飞, 男, 1968 年生, 教授, 博士生导师, 教育部长江学者特聘教授, 国家杰出青年基金获得者, 主要研究方向为计算智能、智能特征建模、自组织控制和智能优化。在 Automatica、IEEE Trans. 刊物、自动化学报等权威期刊上发表学术论文百余篇。



李文静, 女, 1985 年生, 副教授, 博士, 主要研究方向为神经计算、人工神经网络、模式识别。申请美国发明专利 1 项。发表学术论文 10 余篇, 被 SCI 收录 8 篇。

2018 年第二届图像及信号处理国际会议 (ICGSP 2018) The 2nd International Conference on Graphics and Signal Processing (ICGSP 2018)

Oct. 6-8, 2018, Sydney, Australia

The 2nd International Conference on Graphics and Signal Processing (ICGSP 2018) aims to be a prime international forum for both researchers and industry practitioners to exchange the latest fundamental advances in the state of the art and practice of graphics and signal processing, identify emerging research topics, and define the future of Signal Processing. ICGSP offers a three-day technical program including keynote speeches, hot-topic oral&poster sessions, tutorials.

Papers that present original work, validated by experimentation, simulation, or analysis, are solicited. Practical experiences and experimental efforts from both industry and academia, duly documenting the lessons learned from test-beds, field-trials, or real deployments, are also welcome.

Accepted conference papers (full paper only) after authors' attendance will be published in the conference proceedings, which will be online and indexed.

Important Date

- Abstract Submission Deadline: May 10th, 2018
- Full manuscript due: May 30th, 2018
- Acceptance notification: June 20th, 2018
- Camera Ready: July 30, 2018

Website: <http://icgsp.org/>