Vol. 40 No. 12 Dec. 2017

# 多层自适应模块化神经网络结构设计

张昭昭"乔俊飞》余文"

1)(辽宁工程技术大学电子与信息工程学院 辽宁 葫芦岛 125105) 2)(北京工业大学信息学部 北京 100124) 3)(墨西哥国立理工高级研究中心自动控制中心 墨西哥城 墨西哥 07360)

摘 要 针对单一全互连前馈神经网络难以应对复杂问题及模块化神经网络应用时其结构难以确定的问题,该文基于脑式信息处理是采用无监督学习-半监督学习-监督学习的学习机制以及大脑是由多个功能模块组成,每个功能模块中又包含多个子模块,大脑对信息的学习是有目的的选择不同功能模块中多个子模块协同学习的事实,提出一种多层自适应模块化神经网络结构设计方法.其实质是首先对所有的训练数据采用概率密度峰值快速聚类算法确定训练数据的聚类中心,以此确定模块化神经网络中功能模块的个数,其次采用条件模糊聚类实现对每个功能模块中子模块的划分并确定每个子模块的训练样本集;对功能模块中的每一个子模块采用训练误差峰值构造RBF网络的增长算法,该算法能根据分配来的训练样本自适应构建子模块结构;在子模块集成方面,采用基于距离测度的子模块集成方法,该方法能从不同的功能模块中选择不同的子模块对训练样本协同处理.该文提出的模块化神经网络结构设计方法只需要2个人工参数且学习速度提高了近10倍,在一定程度上实现了神经网络的黑箱效应.最后,文中基于人工数据集的复杂函数拟合问题、双螺旋分类问题以及真实数据集的回归问题进行了实验,并与当前国际流行的网络结构进行了对比,结果显示文中提出的模块化神经网络网络结构解决了全互连RBF网络难以应对的复杂问题,而且学习精度高,学习速度快,最终网络的泛化性能强.

关键词 模块化神经网络;自适应;径向基函数;脑式信息处理;协同学习中图法分类号 TP18 **DOI**号 10.11897/SP.J.1016,2017.02827

# Structure Design of Hierarchical Adaptive Modular Neural Network

ZHANG Zhao-Zhao<sup>1)</sup> QIAO Jun-Fei<sup>3)</sup> YU Wen<sup>3)</sup>

1) (Institute of Electronic and Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, 2) (Faculty of Information Technology, Beijing University of Technology, Beijing 100124)
3) (Department of de Control Automatic, CINVESTAV-IPN, Mexico City D. F. 07360, Mexico)

Abstract Modular neural network is an effective method to solve the complex problems that the monolithic fully coupled feedforward neural networks are difficult to learn. The most difficult problem of the modular neural network design method now facing is how to determine the number of function modules and the structure of the sub-modules in each function module under the condition of lack of the learned objects back ground knowledge. In this paper, we presents a hierarchical adaptive modular neural network structure design method based on the facts that the brain-like information process uses the mechanism of unsupervised learning, semi-supervised learning and supervised learning, and the brain networks demonstrate the property of hierarchical modularity, within each module there will be a set of function modules, and within each function there will be a set of sub-modules, and the brain-like information learning processing is purposeful to select several sub-modules from different function module to collaboratively learning. In essence, firstly, an

收稿日期:2016-06-28;在线出版日期:2017-05-02. 本课题得到国家自然科学基金(61440059,61533003)、国家杰出青年科学基金(61225016)、辽宁省自然科学基金(201602363)、国家留学基金委资助. 张昭昭,男,1973年生,博士,副教授,主要研究方向为智能信息处理、神经网络结构优化设计. E-mail: zzzhao123@126. com. 乔俊飞,男,1968年生,博士,教授,博士生导师,主要研究领域为计算智能、智能特征建模和智能优化控制. 余 文,男,1965年生,博士,教授,博士生导师,主要研究领域为神经网络、大数据、智能优化控制.

unsupervised clustering algorithm named fast finding of density peaks cluster algorithm is adopted to identify the cluster centers based on all training samples, then the number of the function module and the training set of each function module can be determined. Secondly, a semi-supervised clustering algorithm named conditional fuzzy clustering algorithm is used to divide the training set of each function module into several groups to determine the number of sub-modules in each function module. For each sub-module, an incremental design of RBF network algorithm based on training error peak is applied to construct the structure of sub-module, this algorithm can adaptively build the structure of the sub-modules based on the training samples that allocated to the sub-modules. In sub-modules integration, a sub-module integrative approach based on relative distance measure is applied which can select different sub-modules from different function modules to collaboratively learning the training samples. The modular neural network structure design method we presented in this paper solves the problem of how to determine the number of sub-modules and the structure of the sub-modules on the situation of lack of the back ground knowledge of the learning objects, moreover, the presented modular neural network design method requires only 2 artificial parameters ( $H_i$  the number of sub-modules in each function module, Kthe membership degree of training samples to function module training sets) compared with other modular neural network design methods, and the learning speed increased nearly by 10 times compared with monolithic fully coupled RBF neural network, therefore, the proposed modular neural design method really achieves the black box effect on neural network application. At the end of this paper, we evaluate the proposed hierarchical modular neural network on several benchmark problems, include complex function fitting problem and double helix classification problem based on artificial data set, and complex data regression problems based on real data sets. The extensive results show that the modular neural network architecture proposed in this paper can not only solve the complex problems that the fully coupled RBF difficult to deal with, but also has high learning precision, fast learning speed and high generalization ability.

**Keywords** modular neural network; adaptive; RBF neural network; brain-like information process; collaborative learning

# 1 引 言

全互连前馈神经网络的 RBF 网络以其结构简单、非线性拟合能力强大、收敛速度快、不易陷于局部极小点、鲁棒性强等优点,已经广泛应用于信号处理、模式分类、工业控制和非线性系统建模领域[1-4]. RBF 神经网络应用的前提是网络的参数学习和结构设计算法<sup>[5]</sup>. 目前为止,虽然国内外学者提出了许多 RBF 网络结构设计算法和网络学习算法<sup>[6]</sup>,然而,作为全互连前馈神经网络的 RBF 网络,依然很难摆脱不能有效应对高纬输入数据问题和大数据量复杂问题的窘境,即当 RBF 网络面对上述问题时不是学习效果不理想就是学习时间过长<sup>[7]</sup>.

为克服全互连前馈神经网络的局限,许多学者根据人脑"功能分区"的事实,把模块化的概念引入

到神经网络结构设计中来[8]. 典型的模块化神经网 络(Modular Neural Network, MNN)有两种学习方 式,一种是采用"分而治之"的策略,将一个复杂问题 分解成多个相对简单的子问题,每个子模块都独立 处理分配给自己的子问题,最后通过对各个子模块 学习结果的集成而实现对整个复杂问题的求解[9]. 这种学习方式的优点是能够学习复杂问题,缺点是 对于分类边界数据的学习效果不理想;另一种模块 化神经网络的学习方式不对任务进行分解,而是用 多个不同结构的网络学习同一任务,最后将多个网 络的学习结果进行集成. 这种学习方式的优点是能 够提高网络的学习精度和泛化性能,缺点是不能有 效学习复杂问题. 相对于单一全互连神经网络而言, 模块化神经网络有很多优点:模型的复杂性低、鲁棒 性和容错能力强、可扩展性强、计算效率高、学习能 力和泛化能力强等[10].正是由于模块化神经网络有

如此多的优点,因此很快在各个领域得到了广泛的 应用. Lu 和 Ito[11]提出了高斯零交叉最小最大模块 化神经网络,该网络把训练样本空间分解为多个子 模块,每个子模块处理的是一个仅包含两类训练样 本的二分类问题. 然而,该模块化神经网络会导致网 络中产生大量的冗余模块,其学习的空间复杂度和时 间复杂度是训练样本的平方[12],这必然会浪费大量 的存储空间和学习时间. Molina-Vilaplana 等人[13] 提出一种辨识机器人手抓物体形状的模块化神经网 络,Gradojevic 等人[14]将模块化神经网络成功应用 于期权定价问题. 然而上述模块化神经网络的结构设 计需要极强的专业领域知识. Tseng 和 Almogahed[15] 采用遗传算法对模块化神经网络的子模块进行集 成,该模块化神经网络中子模块的划分是对训练样 本空间的随机分类,然而,对于许多实际问题,尤其 是分类问题,采用对训练样本随机划分的方法确定 模块化神经网络中子模块的数量并不一定能有效提 高模块化神经网络的学习性能[16]. 薄迎春等人[17]根 据视觉系统工作原理提出了一种多模块协同参与的 神经网络,该网络对于复杂问题具有极强的学习能 力. 然而,该网络子模块的确定采用 K-means 聚类 算法,该聚类算法聚类结果对初始聚类中心的选择 非常敏感,而且聚类的个数需事先人为确定[18].最 为重要的是,上述所有模块化神经网络中的子模块 的结构在实际应用时必须事先人为确定,因此这些 模块化神经网络在实际应用时不得不耗费大量的时 间反复实验以确定合适的子模块结构. 综上,模块化 神经网络结构设计存在以下几个问题:一是采用"分 而治之"策略学习的模块化神经网络对分类边界信 息学习效果不佳;二是在缺少学习对象专业领域知 识的前提下,很难确定整个网络中的子模块数;三是 子模块的网络结构也需要人为确定. 近年来,神经网 络结构设计逐渐向脑科学靠拢[19], David 和 Clauset 等人[20-21]指出人的大脑是由空间分布的多个不同功 能模块组成,每个功能模块中又包含多个子模块,大 脑对信息的处理是有目的地选择不同功能模块中的 多个子模块协同完成.

针对上述模块化神经网络结构设计中存在的问题,本文提出一种多层自适应模块化神经网络结构(Hierarchical Adaptive Modular Neural Network, HAMNN),与传统模块化神经网络不同,HAMNN中包含多个功能模块,每个功能模块中又包含若干个子模块,每个子模块的网络结构由本身的学习任务自适应确定.需要指出的是,HAMNN具有两种

学习策略,对于分类属性明确的训练样本将会在某个功能模块中选择一个子模块进行学习,而对处于分类边界的训练样本,则会由多个不同功能模块中的多个子模块协同完成学习任务.最后,本文基于人工数据集和真实数据集的回归问题和分类问题验证了 HAMNN 的性能.

# 2 HAMNN 结构

本文所提 HAMNN 结构如图 1 所示.

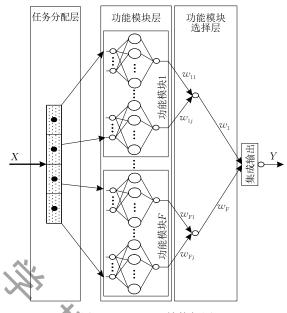


图 1 HAMNN 结构框图

如图 1 所示, HAMNN 结构和学习方式明显区别于传统的模块化神经网络结构, 其对信息的处理采取的是多级分层处理的方式. HAMNN 由多个功能模块组成,每个功能模块中又有多个子模块. HAMNN 的学习方式是:首先通过任务分配层对不同的训练样本选择 HAMNN 中不同功能模块中的一个子模块进行学习, 然后对不同功能模块中子模块的输出进行优化选择和集成而得到最终的学习结果. 需要指出的是, HAMNN 中功能模块数和子模块的结构由训练样本本身自适应确定, 不需要人为参与; 其学习方式也不同于传统模块化神经网络, 在HAMNN 的学习过程中, 就一个具体的训练样本而言, 尤其是处于分类边界的数据, 会由多个不同功能模块中的子模块协同完成学习任务.

在 HAMNN 中,每个具体的子模块为结构相对简单的 RBF 网络,其结构根据任务分配层分配来的训练样本自适应构建,其任务是学习任务层分配来的训练样本.不失一般性,设 HAMNN 中第 F 个

功能模块中第  $H_i$ 个子模块的 RBF 网络结构为 I-M-1(I 个输入节点,M 个隐节点,1 个输出节点),分配给子模块  $H_i$ 的第 k 个 I 维样本为  $x_k = (x_{k,1},x_{k,2},\cdots,x_{k,I})$ ,则第  $H_i$ 个子模块的输出为

$$O_{H_i,k} = \sum_{h=1}^{M} w_h \phi_h(\mathbf{x}_k) + w_0$$
 (1)

$$\phi_h(\mathbf{x}_k) = \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{c}_h\|^2}{\sigma_h}\right)$$
 (2)

其中, $w_h$ 为第h 个隐节点与输出节点的连接权值, $w_0$ 为输出节点的偏置连接权值, $c_h$  和  $\sigma_h$  分别表示第h 个隐节点的核函数中心和宽度, $O_{H_i,k}$  为第 $H_i$  个子模块对训练样本  $x_k$  的输出, $\phi_h(x_k)$  为  $H_i$  个子模块中第h 个隐节点对训练样本  $x_k$  的输出.

HAMNN 结构设计过程包括: 功能模块的划分、子模块的划分、学习过程中针对不同的训练样本如何选择不同功能模块中的子模块学习、子模块如何根据分配来的训练样本自适应构建自身的网络结构和子模块学习结果的集成.

# 3 HAMNN 结构设计

#### 3.1 功能模块的划分

在缺乏学习对象背景知识的情况下,聚类算法常被用来确定模块化神经网络中的子模块的数量<sup>[22]</sup>.文献[18]提出了一种基于数据点概率密度峰值的快速聚类算法,该算法的核心思想在于对聚类中心的刻画,认为聚类中心应该同时具备两个条件:一是本身的局部密度必须大,即它被局部密度均不超过它的邻居所包围;二是与其他局部密度更大的数据点之间的距离相对更远.该算法用两个参数  $\rho_i$ 和 $\delta_i$ 描述聚类中心,其中  $\rho_i$ 表示数据点 i 的局部概率密度, $\delta_i$ 表示数据点 i 与其他具有更高局部密度数据点之间的最小距离.

根据以上描述,设神经网络的训练样本集为  $S = \{(\mathbf{x}_k, y_k) | \mathbf{x}_k \in R^I, y_k \in R^1, k = 1, \dots, N\}$ ,对于 S 中的任意一个训练样本  $\mathbf{x}_i, \rho_i$ 和  $\delta_i$ 的定义为

$$\rho_{i} = \sum_{i \neq i}^{N} \exp \left(-\left(\frac{d_{ij}}{d_{c}}\right)^{2}\right)$$
 (3)

$$\delta_i = \min_{j:\rho_i > \rho_i} d_{ij} \tag{4}$$

其中, $d_{ij}$  = dist( $\mathbf{x}_i$ , $\mathbf{x}_j$ )表示样本  $\mathbf{x}_i$ 和  $\mathbf{x}_j$ 之间的欧式 距离, $d_c$ >0 表示截断距离.

本文采用文献[18]所提算法思想来确定训练样本的聚类中心,从而实现根据训练样本空间分布划分HAMNN中功能模块的作用,具体算法描述如算法1.

## 算法 1. 辨识训练样本的聚类中心.

1. 初始化及预处理

给定训练样本集  $S = \{(\mathbf{x}_k, y_k), \mathbf{x}_k \in R^I, y_k \in R^1, k = 1, \dots, N\}$ ;

计算距离  $d_{ii}$ ,并令  $d_{ii} = d_{ii}$ ,i < j,i, $j = 1, \dots, N$ ;

将  $d_{ij}$ (共 M = N(N-1)/2 个)进行升序排列,得到序列  $d_1 \le d_2 \le \cdots \le d_M$ ,令  $d_c = d_{f(M)}$ ,其中 f(Mt)表示对 Mt 进行四舍五人后得到的整数;

- 2. 根据式(3)计算  $\rho_i$  ( $i=1,\dots,N$ ),并对  $\rho_i$  按照降序排列;
- 3. 根据式(4)计算  $\delta_i$  ( $i=1,\cdots,N$ ),并对  $\delta_i$  按照降序排列;
  - 4. 令  $\gamma_i = \rho_i \delta_i$ ,并生成决策图以确定聚类中心.

设上述聚类算法辨识出的训练样本集的聚类中心为 $\{c_1,c_2,\cdots,c_F\}$ ,共F个聚类中心.基于上述聚类算法辨识出的聚类中心,每个聚类中心针对所有训练样本依据式(5)建立模糊集,则共有F个模糊集:

$$f_{ik} = \exp\left(\frac{-\|\mathbf{x}_k - \mathbf{c}_i\|^2}{0.02}\right) \tag{5}$$

其中, $f_{ik}$ 表示训练样本 $x_k$ 隶属于第i个模糊集的模糊隶属度.显然,在所有的训练样本中, $x_k$ 距离 $c_i$ 越近,则 $x_k$ 隶属于 $c_i$ 的隶属度就越大.

由式(5)知,由于每个聚类中心都是针对所有的训练样本建立其模糊集,所以在 HAMNN 中就可根据训练样本的空间分布,划分出 F 个功能模块,且每个功能模块都有与其对应的训练样本集.如果设定一个模糊隶属度阈值,即该功能模块中的子模块只学习隶属度大于该阈值的训练样本,则每个功能模块中就有数量不等的训练样本,且每个训练样本也有可能隶属于多个不同的功能模块,尤其是处于分类边界的训练样本.

#### 3.2 功能模块中子模块的划分

功能模块的划分初步实现了对训练样本的分割,但对于训练样本数量很多的复杂问题来说,分配给每个功能模块的训练样本绝对数量有可能还是很大.为进一步提高 HAMNN 的学习性能,对基于式(5)建立的 F 个功能模块所对应的 F 个训练样本模糊集进一步进行分割.

对 F 个模糊集中的每一个模糊集,将该模糊集中的训练样本  $x_k$  按照式(6)和式(7)进行条件模糊聚类[23]:

$$u_{ijk} = \frac{f_{ik}}{\sum_{m=1}^{H_i} \left( \frac{\|\mathbf{x}_k - \mathbf{v}_{ij}\|}{\|\mathbf{x}_k - \mathbf{v}_{im}\|} \right)^2}$$
(6)

$$\mathbf{v}_{ij} = \frac{\sum_{k=1}^{N} (u_{ijk})^2 \mathbf{x}_k}{\sum_{k=1}^{N} (u_{ijk})^2}$$
(7)

其中, $v_{ij}$ 表示第  $i(i=1,\cdots,F)$ 个模糊集对应的第  $j(j=1,\cdots,H_i)$ 个训练样本模糊集的聚类中心, $H_i$ 表示第 i 个模糊集所对应的训练样本条件模糊聚类 个数.  $\mathbf{H} = [u_{ijk}] \in R^{F \times H_i \times N}$  为第 i 个模糊集对应的 训练样本条件模糊聚类的隶属度矩阵,其中 uiik 表 示训练样本 $x_k$ 隶属于第i个功能模块中第j个子模 块的隶属度,满足条件如下:

$$\sum_{i=1}^{H_i} u_{ijk} = f_{ik} , \ u_{ijk} \in [0,1]$$
 (8)

采用上述对每个功能模块中训练样本的划分方 法,就可将 F 个功能模块中的训练样本集划分为  $H_{\tau}$ 个样本子集,满足条件如下:

$$H_T = \sum_{i=1}^F H_i \tag{9}$$

对训练样本的两次划分方法如算法 2 描述.

算法 2. 功能模块和子模块训练样本的划分.

- 1. 给定训练样本集  $S = \{(x_k, y_k), x_k \in R^1, y_k \in R^1, k = \}$  $1, \dots, N$ } 及算法 1 辨识出的训练样本集聚类中心  $\{c_1, c_2, \dots, c_n\}$  $c_F$  },设每个功能模块训练样本集条件模糊聚类的个数为  $H_i$ ,  $i=1,\dots,F$ , 即每个功能模块中有  $H_i$ 个子模块;
  - 2. 根据式(5)计算  $f_{ik}$ ,  $k=1,\dots,N$ ;
  - 3. 根据式(6)计算  $u_{ijk}$ ,  $j=1,\dots,H_i$ ;
- 根据式(7)修正第 i 个功能模块中第 j 个子模块研 练样本聚类中心;
- 5. 计算新聚类中心与旧聚类中心的距离,若满足条件 则条件模糊聚类结束,否则转步骤 3.

经过上述算法对训练样本集的两次划分,就可 以在 HAMNN 中划分出 F 个功能模块(用  $FM_i$  表 示第i 个功能模块, $i=1,\dots,F$ ),每个功能模块中又 划分出数量不等的  $H_i$ 个子模块(用  $SM_{ii}$  表示第 i 个 功能模块中的第 i 个子模块,  $i=1,\dots,H_i$ ), 且每个 子模块都有其对应的训练样本集.

#### 3.3 功能模块中子模块的选择

依照上述对神经网络训练样本集的划分方法, 每个训练样本与功能模块中每个子模块之间都存在 一定的隶属关系. 如果训练样本  $x_k$ 与第 i 个功能模 块中的第j个子模块 $SM_{ij}$ 距离比较近,则 $x_k$ 隶属于  $SM_{ij}$ 的可能性就比较大. 我们采用相对距离测度的 方法来度量  $x_k$ 隶属于  $SM_{ij}$  的程度.

构建性能指标函数如下;

$$J_{i} = \sum_{j}^{H_{i}} w_{ij} d_{ij}; \sum_{j=1}^{H_{i}} w_{ij} = 1, w_{ij} \in [0,1] \quad (10)$$

$$d_{ij} = \frac{\|\mathbf{x}_{k} - \mathbf{v}_{ij}\|}{da_{ij}}, da_{ij} = \frac{1}{N_{ij}} \sum_{m=1}^{N_{ij}} \|\mathbf{x}_{m} - \mathbf{v}_{ij}\| \quad (11)$$

其中, $da_{ij}$ 为子模块 $SM_{ij}$ 中所有训练样本与该子模

块聚类中心间的平均距离, $d_{ii}$ 为训练样本 $x_k$ 与子模 块 $SM_{ij}$ 的相对距离测度 $,N_{ij}$ 为子模块 $SM_{ij}$ 训练样 本总数, $w_{ij}$ 为训练样本 $x_k$ 隶属于子模块 $SM_{ij}$ 的隶 属度.

依据式(10)和式(11),则每个功能模块中的每 个子模块对训练样本  $x_k$ 的隶属度  $w_{ij}$ 可通过下式计 算得到:

$$w_{ij} = \begin{cases} 1, & d_{ij} = 0\\ \frac{\left(\frac{1}{d_{ij}}\right)}{\sum_{j=1}^{H_i} \left(\frac{1}{d_{ij}}\right)}, & d_{ij} \neq 0 \end{cases}$$
 (12)

由式(12)知,相对距离测度 dij 越大,子模块  $SM_{ij}$ 的  $w_{ij}$ 越小,即训练样本  $x_k$ 隶属于子模块  $SM_{ij}$ 的隶属度就越小;反之则训练样本  $x_{i}$ 隶属于子模块  $SM_{ii}$ 的隶属度就越大. 采用最大隶属度法,就可以 在每个功能模块中选择出一个最适合学习该训练样 本的子模块,此时每个功能模块的输出实际上就是 该功能模块中被选中的子模块的输出.

由功能模块的划分过程可知, $x_k$ 有可能距离某 个或某几个功能模块的聚类中心很远,采用上述方 法仍然需要在该功能模块中选择出一个子模块来学 习该训练样本. 为提高 HAMNN 的学习速度和精 度,需要对所有选中的子模块再进行一次选择.

对于第一次选中的子模块构建性能指标函数:

$$J = \sum_{i} w_{i} d_{i}, \sum_{i=1}^{r} w_{i} = 1, w_{i} \in [0,1]$$
 (13)  
$$d_{i} = \frac{\|\mathbf{x}_{k} - \mathbf{c}_{i}\|}{da_{i}}, da_{i} = \frac{1}{N_{i}} \sum_{j=1}^{N_{i}} \|\mathbf{x}_{k} - \mathbf{c}_{i}\|$$
 (14)

$$d_i = \frac{\|\mathbf{x}_k - \mathbf{c}_i\|}{da_i}, da_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \|\mathbf{x}_k - \mathbf{c}_i\|$$
(14)

其中 $,d_i$ 为训练样本 $x_k$ 对功能模块 $FM_i$ 的相对距离 测度 $,da_i$ 为隶属于功能模块 $FM_i$ 的所有训练样本与 该功能模块聚类中心间的平均距离, $N_i$ 为功能模块  $FM_i$ 中训练样本的总数, $c_i$ 表示功能模块 $FM_i$ 所对 应的聚类中心,

依据式(13)和式(14),则功能模块中被选中可 以学习训练样本  $x_k$ 的子模块的集成权值  $w_i$ 就可通 过下式计算得到:

$$w_{i} = \begin{cases} 1, & d_{i} = 0\\ \frac{\left(\frac{1}{d_{i}}\right)}{\sum_{i=1}^{F} \left(\frac{1}{d_{i}}\right)}, & d_{i} \neq 0 \end{cases}$$
 (15)

由式(15)知,训练样本 $x_k$ 距离某个功能模块  $FM_i$ 的聚类中心 $c_i$ 越远,则该功能模块中被选中的

子模块的集成权值  $w_i$ 就越小,即表明该功能模块中 的子模块不适合对训练样本  $x_{\iota}$  进行学习,反之则适 合. 为此,我们可以设置一阈值 K,当  $w_i > K$  时,每 个功能模块中被选中的子模块才参与对训练样本  $x_i$ 的学习.

每个功能模块中子模块的选择算法见算法 3.

## 功能模块中子模块的选择.

- 1. 依据算法 2 划分的每个子模块的训练样本集及聚类 中心,根据式(10)、式(11)和式(12)计算每个功能模块中不 同子模块对训练样本  $x_{i}$  进行学习的隶属度值  $w_{ii}$ ;
- 2. 采用最大隶属度法从每个功能模块中选择一个最 适合学习训练样本 $x_k$ 的子模块;
- 3. 根据式(13)、式(14)和式(15)计算被选中的子模块 学习训练样本  $x_k$ 的集成权值  $w_i$ ;
- 4. 对选择的子模块进行筛选,设置阈值 K,选择  $w_i >$ K 的子模块学习训练样本 $x_k$ .

采用上述功能模块中子模块的选择算法,设置 合适的阈值 K,则对于给定的训练样本  $x_k$ ,该算法 就会根据 $x_k$ 在训练样本空间中的分布而从不同的 功能模块中选择数量不等的子模块协同学习...

## 3.4 功能模块中子模块结构的自适应设计

HAMNN 中不同功能模块中的子模块为 RBF 网 络,根据训练样本设计 RBF 网络结构一直是 RBF 网络应用的难题. 文献[24]对 Levenberg-Marquardt 算法进行了改进,在此基础上,文献[6]提出一种根 据训练误差峰值构建 RBF 网络结构的 ErrCor 算 法,该算法能够根据训练样本本身设计出结构最为 紧凑的 RBF 网络.

Levenberg-Marquardt 算法参数更新规则为

$$\Delta_{k+1} = \Delta_k - (Q_k + \mu_k I)^{-1} g_k$$
 (16)  
其中, $\Delta$  表示 RBF 网络中的可调参数(RBF 隐节点

核函数的中心 c、宽度  $\sigma$  及 RBF 网络隐节点与输出 节点的连接权值 w); Q 为 Quasi-Heissian 矩阵; i 为 单位矩阵; μ为组合系数; g 为梯度向量.

通过对 Subquasi-Hessian 矩阵求和就可计算 出 Quasi-Hessian 矩阵 Q:

$$Q = \sum_{p=1}^{P} q_{p}; \ q_{p} = \boldsymbol{j}_{p}^{T} \boldsymbol{j}_{p}$$
 (17)

对子梯度向量  $\eta$ 。求和则可计算出梯度向量 g:

$$\mathbf{g} = \sum_{p=1}^{P} \mathbf{\eta}_{p}; \; \mathbf{\eta}_{p} = \mathbf{j}_{p}^{\mathrm{T}} e_{p}$$
 (18)

设 RBF 网络训练误差 e,为

$$e_p = y_p - o_p \tag{19}$$

其中,y,为 RBF 网络期望输出,o,为 RBF 网络实际 输出.

Jacobian 矩阵行元素  $j_{\mathfrak{o}}$ 可根据下式计算:

$$\boldsymbol{j}_{p,n} = \frac{\partial e_p}{\partial \boldsymbol{\Lambda}_n} \tag{20}$$

依据式(19)和式(20),则 Jacobian 矩阵中行元 素值为

$$\mathbf{j}_{p} = \left[ \frac{\partial e_{p}}{\partial w_{0}}, \frac{\partial e_{p}}{\partial w_{1}} \cdots \frac{\partial e_{p}}{\partial w_{h}} \cdots \frac{\partial e_{p}}{\partial w_{H}}, \frac{\partial e_{p}}{\partial c_{1,1}} \cdots \frac{\partial e_{p}}{\partial c_{1,i}} \cdots \frac{\partial e_{p}}{\partial c_{1,i}} \cdots \frac{\partial e_{p}}{\partial c_{h,1}} \right]$$
(21)

依据式(1),式(2)和式(19),设参与训练的p个训练样本为 X<sub>0</sub>,根据微分链式法则,则式(21) Jacobian 矩阵中行元素的值为

$$\frac{\partial e_p}{\partial w_h} = -\varphi_h(\mathbf{X}_p), \quad \frac{\partial e_p}{\partial w_0} = -1$$
 (22)

$$\frac{\partial e_p}{\partial c_{h,i}} = -\frac{2w_h \varphi_h(\boldsymbol{X}_p)(x_{p,i} - c_{h,i})}{\sigma_h} \tag{23}$$

$$\frac{\partial e_{p}}{\partial c_{h,i}} = -\frac{2w_{h}\varphi_{h}(\boldsymbol{X}_{p})(x_{p,i} - c_{h,i})}{\sigma_{h}} \qquad (23)$$

$$\frac{\partial e_{p}}{\partial \sigma_{h}} = -\frac{w_{h}\varphi_{h}(\boldsymbol{X}_{p})\|\boldsymbol{X}_{p} - \boldsymbol{c}_{h}\|^{2}}{\sigma_{h}^{2}} \qquad (24)$$

对参与训练的所有训练样本,依据式(22)~ 式(24),可计算出 Jacobian 矩阵  $j_a$ 的所有行元素, 进而依据式(17)和式(18)分别求得 Quasi-Heissian 矩阵 Q 和梯度向量 g,最后根据式(16)对 RBF 网络 中三个可调参数进行调整.

RBF 网络结构设计的思想是在每次迭代计算 后,根据式(19)计算出迭代运算后的最大误差,并在 误差曲面的最大误差所对应的训练样本处增加一个 隐节点. 该算法详细请参考文献[6].

依照上述学习算法和 RBF 网络结构增长算法, 在 HAMNN 中,每个功能模块中的子模块都能根 据任务分配层分配来的训练样本自适应构建自身的 网络结构,不需要人工参与,

## 3.5 子模块学习结果的集成

设训练样本为 $x_k$ ,每个功能模块中被选中学习 该训练样本的子模块的集成权值为  $\mathbf{w} = (w_1, \cdots, w_m)$  $(w_F)$ , 若  $(w_i)$  小于设定的阈值  $(K, \mathbb{N})$  号  $(w_i)$  = 0, 并对  $(w_i)$ 进行归一化处理,则 HAMNN 的集成输出为

$$Y = \sum_{i=1}^{C} w_i O_i \tag{26}$$

其中,C表示经过二次选择,最终参与样本 $x_k$ 学习 的子模块的个数;O<sub>i</sub>表示被选中的第i个子模块的 输出,如式(1)所示.

# 性能评估

为了验证 HAMNN 的性能, 选取典型回归问

题、双螺旋分类问题和基于实际数据集的回归问题进行实验.为便于性能比较,选择 *RMSE*(Root Mean Square Error)为性能比较参数:

$$RMSE = \sqrt{\frac{1}{P} \sum_{p=1}^{P} e_p^2}$$
 (26)

其中,p为样本个数, $e_p$ 为学习误差或测试误差.

实验中的 ErrCor 算法和 HAMNN 算法的代码均由我们自己实现,基于真实数据集的仿真实验是每种算法各独立运行 30 次并将最佳平均结果进行的汇总.实验的环境是: Window 10 64-bit, Intel Core i7-2600 CPU@3.4-GHz, 8-GB RAM. 仿真软件为 Matlab 2012.

### 4.1 人工数据回归问题

本实验的目的在于检验 HAMNN 功能模块划分的准确性及对复杂非线性函数的建模性能. 选取典型非线性函数如下:

$$z(x,y) = \frac{1}{30} \exp(-1 - 6x - 9x^2) - \frac{1}{30} \exp(-3x^2 - 9x^2) + \frac{1}{30} \exp(-9x^2 - 9y^2) + \frac{1}{30}$$

$$(0.3-1.8x+2.7x^2)\exp(-1-6y-9x^2-9y^2)$$
 (27)

与其它实验方法不同,该实验的训练样本不是在 $x,y \in (-1,1)$ 内随机产生均匀分布的训练样本,而是以高斯分布在其定义域内分别产生 5 个不同区域的训练样本((x,y),z(x,y)) 共 2000 个. 仿真时设置 K=0.1,每个功能模块中子模块的个数均为 3.

图 2 所示为本实验训练样本分布图;图 3 所示 为任务分配层的聚类算法在训练样本空间内根据其 空间分布所确定的聚类中心. 从图 3 可以看出,有 5个训练样本的局部概率密度与最小距离乘积之值 最为突出,表明该聚类算法确实能够根据训练样本 的空间分布准确识别聚类中心;图 4 所示为 1000 个 测试样本的测试效果,该测试曲面非常平滑,验证了 本文所提出的子模块选择和子模块集成算法的有效 性,最终测试 RMSE 为 0.0052. 为了验证本文所提 出的 HAMNN 同传统 MNN 的区别,采用同样的训 练样本集和测试样本集,我们构造了一个传统 MNN, 其方法是采用 K-means 聚类算法对训练样本进行 "硬分类",分类类别数同 HAMNN 子模块数量相 等,共15个分类.对每个分类采用 ErrCor 算法分别 构建相应的子模块. 测试时, 距离测试样本最近的聚 类中心所在的子模块的输出为传统模块化神经网络 对该测试样本的输出,最终测试 RMSE 为 0.0097. 图 5 所示为传统 MNN 的测试输出曲面,从该图可 以看出,传统 MNN 学习时很难避免分类边界出现 "突变"现象,这也进一步表明,本文所提 HAMNN 对分类边界数据采取模糊处理、协同学习的策略确实能够提高对分类边界数据的处理能力.针对同样的问题,文献[6]采用有 20 个隐节点的全互连 RBF 网络,最终测试样本的测试 RMSE 为 0.0141,因此本文所提 HAMNN 不仅具有很强的拟合能力,而且最终网络具有更强的泛化性能.

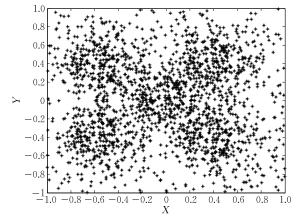


图 2 训练样本分布图

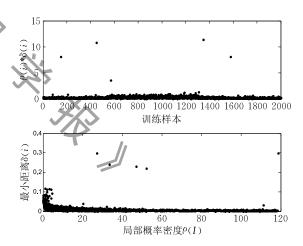


图 3 HAMNN 任务分配层确定的聚类中心

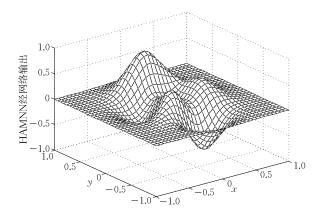


图 4 HAMNN 测试样本输出曲面图

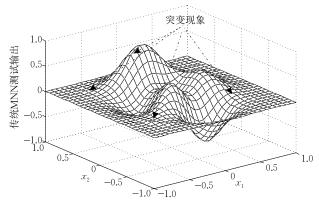


图 5 传统模块化神经网络测试样本输出曲面图

表1给出了本次实验 HAMNN 的最终结构以及每个子模块的训练样本数,从表1可看出,每个功能模块中的子模块只学习分配到该功能模块的部分训练样本,且每个子模块都能根据所分配来的训

练样本自适应构建自身的网络结构.表2所示为HAMNN同其他典型神经网络结构设计算法GAP<sup>[25]</sup>、MRAN<sup>[26]</sup>、RAN<sup>[27]</sup>、SVR<sup>[28]</sup>、I-ELM<sup>[29]</sup>、CI-ELM<sup>[30]</sup>、EI-ELM<sup>[31]</sup>、ErrCor<sup>[6]</sup>针对同一回归问题的性能比较.从该表可以看出,尤其同文献[6]所提ErrCor算法相比较,虽然测试时间比ErrCor长,但无论是网络的训练时间、学习精度,还是最终网络的泛化性能,HAMNN均优于ErrCor算法.究其原因,是HAMNN对训练样本进行了两次划分,使得每个训练样本空间聚集区域都有对应的不同功能模块中的多个子模块协同处理,因此每个子模块处理的训练样本相对较少,进而使得HAMNN学习和网络结构构建的时间就短很多,这不仅能够提高HAMNN的学习速度,也有助于提高HAMNN的学习速度,也有助于提高HAMNN的学习速度,也有助于提高HAMNN的学习速度和泛化性能.

表 1 HAMNN 结构及训练样本分布

	子模块1 隐节点数	子模块 2 隐节点数	子模块 3 隐节点数	子模块 1 训练样本数	子模块 2 训练样本数	子模块 3 训练样本数
功能模块1	7	11	7	324	408	289
功能模块 2	6	11	13	278	316	415
功能模块3	8	6	7	333	339	246
功能模块 4	7	9	13	266	311	369
功能模块 5	6	12		302	373	304

表 2 HAMNN 同其他典型神经网络学习算法性能比较

算法	训练时间/s	训练 RMSE	测试时间/μs	测试 RMSE
GGAP	24.808	0.0265	54.16	0.0265
MRAN	78.572	0.0458	52.5	0.0490
RAN	45.514	0.0671	112.2	0.0686
SVR	0.2552	0.0346	2496.0	0.0361
I-ELM	0.5509	0.0831	239.0	0.0361
CI-ELM	0.5597	0.1356	239.0	0.1378
EI-ELM	5.3991	0.0728	239.0	0.0755
ErrCor	48.530	0.0141	23.90	0.0141
HAMNN	28.216	0.0122	31.12	0.0052

#### 4.2 双螺旋分类问题

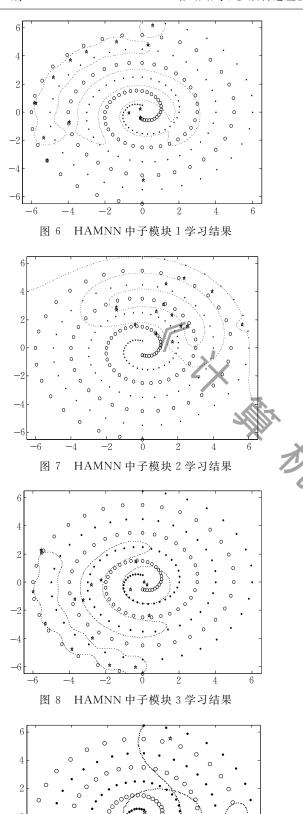
双螺旋分类问题由两条相互缠绕的螺旋线构成,每条螺旋线对应一个分类.由于两条螺旋线相互缠绕,即分类之间重叠,因此该问题是最困难的模式分类问题之一,也是评估神经网络结构或学习算法性能的基准问题,训练样本产生方式如下:

$$\begin{cases} \theta = i \times \pi/16 \\ r = 6.5 \times (104 - i)/104 \\ x = r \times \sin(\theta) \\ y = r \times \cos(\theta) \end{cases}$$
 (28)

实验时,产生 388 个训练样本点分属于两个点集,测试样本对为 17 161 个. 本文所提 HAMNN 任务分配层将训练样本集划分为 3 个功能模块,仿真时

设置 K=0.1,每个功能模块中子模块个数均为 2.

图 6~图 11 所示为本次实验中 HAMNN 中 6 个 子模块针对各自分配的训练样本的学习情况(图中 ○和●分别表示双螺旋分类中的两种不同类别,☆ 表示子模块学习结束后 RBF 隐层神经元的中心,虚 线包围区域表示子模块对这部分训练样本的学习效 果). 从这6个图中可以看出几个事实,一是每个子 模块都能够准确地实现对所分配来的训练样本的分 类识别:二是本文所提训练样本分割方法能够根据 训练样本的空间分布将训练样本分配到不同的功能 模块和子模块. 仔细观察 6 个子模块的训练样本分 布,每个子模块学习的训练样本都没有相互缠绕现 象,即每个子模块的学习任务相对比较简单,这表明 本文所采用的对训练样本的划分方法确实能够准确 地将复杂的任务分解为多个简单的子任务;三是每 个子模块都能够根据分配来的训练样本自适应构建 自身的网络结构,不需要人为确定子模块的网络结 构. 从这 6 个子模块的学习结果图还可以看出,不同 子模块的训练样本之间有部分重叠的现象,这是由 于本文所提训练样本分割方法采用模糊决策的方 法,该方法能够使得训练样本分类边界的数据能 够由不同功能模块中的多个子模块协同处理,这不



HAMNN 中子模块 4 学习结果

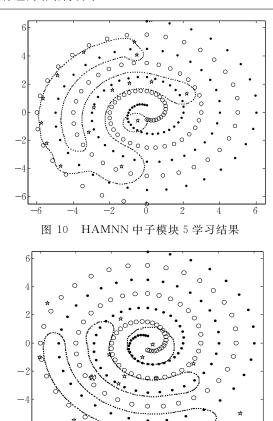
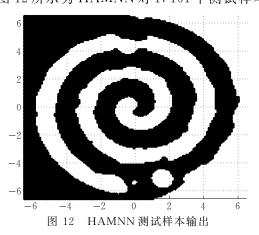


图 11 HAMNN 中子模块 6 学习结果

仅有助于提高网络对分类边界数据的学习精度,也有助于提高网络的泛化性能. 学习结束后,6个子模块 RBF 网络隐节点数分别为 14、14、17、12、16、13. 而针对同样的问题,文献[6] RBF 网络中隐节点数为 20,文献[32] RBF 网络中隐节点数为 74,文献[33] RBF 网络隐节点数为 70,文献[34] RBF 网络隐节点数为 64. 由前述知,由于 HAMNN 对大多数类别属性明确的训练样本只选择一个功能模块中的子模块进行学习,只有处于分类边界的训练样本才会从选择多个不同功能模块中的子模块协同学习,因此,本文所提 HAMNN 网络具有更强的学习能力.图 12 所示为 HAMNN 对 17 161 个测试样本的



测试效果图, HAMNN 对这些测试样本的分类正确率达到 99.27%, 表明本文所提方法设计的 HAMNN 网络具有很强的泛化性能.

#### 4.3 实际数据回归问题

为检验 HAMNN 对实际数据回归问题的处理能力,我们在 UCI 实际数据集上选择训练样本数量大、输入维数高的 7 种数据集,并在 7 种真实数据集上将 HAMNN 与典型 RBF 网络算法 RAN、MRAN和 ErrCor 算法进行了比较,实际数据集详细情况见表 3. 在本实验中,所有数据集的输入都归一化到[-1,1]区间,所有数据集的输出都归一化到[0,1]区间.

几种算法的训练/测试 RMSE 见表 4. 从表 4 可以看出, HAMNN 在所有数据集上的测试 RMSE 都比其它几种算法小,表明 HAMNN 具有更好的泛化性能.表 5 汇总了几种算法的时间消耗, 从表 5 可以

看出,HAMNN 算法的训练速度明显快于 ErrCor和其他算法,考虑到 HAMNN 是模块化设计,每个模块其实都可以独立训练,因此,虽然 HAMNN 子模块结构设计算法也采用的是 ErrCor算法,但针对同样的问题,HAMNN 至少比 ErrCor算法快近 10倍.从表 4 还可以看出,HAMNN 算法的测试时间稍长于 ErrCor算法,这是由于在测试样本测试时,HAMNN 算法需要采用模糊策略从不同的功能模块中选择子模块.

表 3 实际数据回归数据集信息

* * * * * * * * * * * * * * * * * * * *			
数据集	训练样本	测试样本	输入维数
Abalone	2000	2177	8
Delta ailerons	3000	4129	5
Delta elevators	4000	5517	6
Computer activity	4000	4192	8
Census	10000	12784	8
Bank domains	4500	3692	8
California housing	8000	1246	8

表 4 几种算法的 RMSE 比较

数据集	RAN 的 RMSE		MRAN 的 RMSE		ErrCor(RBF)的 RMSE		HAMNN 的 RMSE	
	训练样本	测试样本	训练样本	测试样本	训练样本	测试样本	训练样本	测试样本
Abalone	0.0931	0.0978	0. 0836	0.0837	0.0758	0.0765	0.0492	0.0501
Delta ailerons	0.0514	0.0552	0.0461	0.0495	0.0418	0.0431	0.0272	0.0280
Delta elevators	0.0678	0.0733	0.0617	0.0630	0.0551	0.0573	0.0359	0.0375
Computer activity	0.0610	0.0649	0.0552	0.0571	0.0493	0.0507	0.0306	0.0328
Census	0.0851	0.0905	0.0830	0.0857	0.0692	0.0707	0.0444	0.0461
Bank domains	0.0545	0.0579	0.0531	0.0623	0.0443	0.0452	0.0289	0.0294
California housing	0.1154	0.1434	0.1126	0.1407	0.0938	0.1012	0.0600	0.0655

#### 表 5 几种算法的时间比较

数据集	RAN 时间/s		MRAN 时间/s		ErrCor(RBF)时间/s		HAMNN 时间/s	
	训练样本	测试样本	训练样本	测试样本	训练样本	测试样本	训练样本	测试样本
Abalone	105. 17	4. 13 e-4	255.80	1.05 e-4	4. 808	3. 59 e-6	2.070	4.88 e-6
Delta ailerons	114.12	4.50 e-4	278.57	1. 14 e-4	5. 219	3.91 e-6	3.760	5.92e-6
Delta elevators	131.46	5.41 e-4	319.75	1.37 e-4	5.997	4.70 e-6	3.902	5.81 e-6
Computer activity	120.94	4.96 e-4	284.17	1.26 e-4	5.519	4.31 e-6	3. 153	5.65 e-6
Census	241.89	1.02 e-3	568.34	2.58 e-4	11.06	8.82 e-6	7.480	7. 11 e-5
Bank domains	147.55	5.62 e-3	358.12	1.43 e-3	6.750	4.87 e-5	4.330	5. 17 e-5
California housing	212.44	8.97 e-3	516.72	2.03 e-3	9.710	6.92e-5	7.422	9.19e-5

# 5 结束语

针对全互连 RBF 网络难以应对大数据量的复杂问题,以及模块化神经网络应用时结构难以确定的问题,本文从脑式信息处理的角度出发,提出一种多层自适应模块化神经网络结构,该网络首先采用根据训练样本空间分布的概率密度峰值快速聚类算法确定 HAMNN 中的功能模块数,在此基础上,用条件模糊聚类将分配到各功能模块中的训练样本进一步分割以确定每个功能模块中的子模块数;对于

每一个子模块,采用根据训练误差峰值构造 RBF 网络的增长算法,该算法能够根据分配来的训练样本自适应构造自身的网络结构,不需要人工参与;在HAMNN的学习过程中,对于每一个训练样本,通过基于距离测度的子模块集成方法,选择不同功能模块中的子模块协同学习,以提高 HAMNN的学习速度和泛化性能.通过人工数据集和实际数据集的回归问题和分类问题的仿真实验可以看出,本文所提数据分割方法能够准确有效地将神经网络的训练样本根据训练样本的空间分布进行分割;HAMNN的学习精度以及最终网络的泛化性能均

优于其他经典的学习模型,尤其是其学习速度比ErrCor算法提高近10倍;相比于其他模块化神经网络结构设计方法,HAMNN网络结构要求人为设置的参数仅有2个,在一定程度上实现了神经网络的黑箱效应,这无疑为HAMNN网络的实际应用提供了有效的途径.

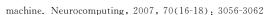
## 参考文献

- [1] Xu Lei, Qian Fang, Li Ya-Ping, et al. Resource allocation based on quantum particle swarm optimization and RBF neural network for overlay cognitive. Neurocomputing, 2016, 173(3): 1250-1256
- [2] Xiong Tao, Bao Yu-Kun, Hu Zhong-Yi, Chiong Raymond. Forecasting interval time series using fully complex-valued RBF neural network with DPSO and PSO algorithms. Information Sciences, 2015, 305(1): 77-92
- [3] Liu Quan, Xiao Fei, Fu Qi-Ming, et al. Collaborative Q-V value function approximation model based on adaptive normalized radial basis function network. Chinese Journal of Computers, 2015, 38(7): 1386-1396(in Chinese) (刘全,肖飞,付启明等. 基于自适应归—化RBF 网络的 Q-V 值函数协同逼近模型. 计算机学报,2015,38(7): 1386-1396)
- [4] Jiao Li-Cheng, Yang Shu-Yun, Liu Fang, et al. Seventy years beyond neural networks: Retrospect and prospect. Chinese Journal of Computers, 2016, 39(8): 1697-1716(in Chinese)
  (焦李成,杨淑媛,刘芳等.神经网络七十年:回顾与展望. 计算机学报, 2016, 39(8): 1697-1716)
- [5] Cruz D P F, Miaia R D, da Silva L A, de Castro L N. BeeRBF: A bee-inspired data clustering approach to design RBF neural network classifiers. Neurocomputing, 2016, 172(8): 427-437
- [6] Yu Hao, Reiner P D, Xie Tian-Tian, et al. An incremental design of radial basis function networks. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(10): 1793-1803
- [7] Zhang Zhao-Zhao, Qiao Jun-Fei. Analysis and Design of Modular Neural Network. Liaoning: Liaoning Science and Technology Publishing House, 2014(in Chinese)
  (张昭昭,乔俊飞. 模块化神经网络结构分析与设计. 辽宁: 辽宁科学技术出版社, 2014)
- [8] Jacobs R. A, Michael I, Jordan A. Modular connectionist architecture for learning piecewise control strategies// Proceedings of the American Control Conference. Boston, USA, 1991; 343-224
- [9] Qiao Jun-Fei, Zhang Zhao-Zhao, Bo Ying-Chun. An online self-adaptive modular neural network for time-varying systems. Neurocomputing, 2014, 125(1): 7-16

- [10] Hapel B, Murre J. Design and evolution of modular neural network architectures. Neural Networks, 1994, 7(6): 9985-1004
- [11] Lu B L, Ito M. Task decomposition and module combination based on class relations: A modular neural network for pattern classification. IEEE Transactions on Neural Networks, 1999, 10(5): 1244-1256
- [12] Zhao Zhong-Qiu, Gao Jun, Herve Glotin, Wu Xin-Dong. A matrix modular neural network based on task decomposition with subspace division by adaptive affinity propagation clustering. Applied Mathematical Modelling, 2010, 34(12): 3884-3895
- [13] Molina-Vilaplana J, Feliu-Batlle J, Lopez-Coronado J. A modular neural network architecture for step-wise learning of grasping tasks. Neural Networks, 2007, 20: 631-645
- [14] Gradojevic N, Gencay R, Kukolj D. Option pricing with modular neural networks. IEEE Transactions on Neural Networks, 2009, 20(4): 626-637
- [15] Tseng H C, Almogahed B. Modular neural networks with applications to pattern profiling problems. Neurocomputing, 2009, 72: 2093-2100
- [16] González B, Valdez F, Melin P, Prado-Arechiga G. Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition. Expert Systems with Applications, 2015, 42: 5839-5847
- [17] Bo Ying-Chun, Qiao Jun-Fei, Yang Gang. A multi-modules cooperative neural networks. CAAI Transaction on Intelligent Systems, 2011, 3: 225-230(in Chinese)
  (薄迎春, 乔俊飞, 杨刚. 一种多模块协同参与的神经网络. 智能系统学报, 2011, 3: 225-230)
- [18] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344: 1492-1496
- [19] Yu Q. Tang H, Tan K C, Yu H. A brain-inspired spiking neural network model with temporal encoding and learning. Neurocomputing, 2014, 138(22): 3-13
- [20] Meunier D, Lambiotte R, Bullmore E T. Modular and hierarchically modular organization of brain networks. Frontiers in Neuroscience, 2010, 4: 1-11
- [21] Clauset A, Moore C, Newman M E J. Hierarchical structure and the prediction of missing links in networks. Nature, 2008, 453; 98-101
- [22] Park H S, Pedrycz W. Grannular neural netowrks and their development through context-based clustering and adjustable dimensionality of receptive fields. IEEE Transactions on Neural Networks, 2009, 20(10): 1604-1616
- [23] Pedrycz W. Conditional fuzzy flustering in the design of radial basis function neural networks. IEEE Transactions on Neural Networks, 1998, 9(4); 601-612
- [24] Wilamowski B M, Yu H. Improved computation for levenberg-marquardt training. IEEE Transactions on Neural Networks, 2010, 21(6): 930-937
- [25] Huang G B, Saratchandran P, Sundararajan N. An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks. IEEE Transactions on Neural Networks, 2005, 16(1): 57-67

计

- [26] Babu G S, Suresh S. Sequential projection-based metacognitive learning in a radial basis function network for classification problems. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(2): 194-206
- [27] Plant J. A resource-allocating network for function interpolation. Neural Computation, 1991, 3(2): 213-225
- [28] Qiao Jun-Fei, Han Hong-Gui. A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modeling. International Journal of Neural Systems, 2010, 20(1): 63-74
- [29] Reniner P, Wilamowski B M. Efficient incremental construction of RBF networks using quasi-gradient method. Neurocomputing, 2015, 150(Part B); 349-356
- [30] Huang G-B, Chen L. Convex incremental extreme learning



- [31] Huang G-B, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing, 2008, 71(16-18): 3460-3468
- [32] Chaiyaratana N, Zalzala A M S. Evolving hybrid RBF-MLP networks using combined genetic/unsupervised/supervised learning//Proceedings of the UKACC. Swansea, Britain, 1998; 330-335
- [33] Neruda R, Kudova P. Learning methods for radial basis function networks. Future Generation Computer Systems, 2005, 21(7): 1131-1142
- [34] Kaminski W, Strumillo P. Kernel orthonormalization in radial basis function neural networks. IEEE Transaction on Neural Networks, 1997, 8(5): 1177-1183



QIAO Jun-Fei, born in 1968, Ph. D., professor, Ph. D. supervisor. His research interests include computational intelligence, smart feature modeling, intelligent optimization control.

YU Wen, born in 1965, Ph.D., professor, Ph.D. supervisor. His research interests include neural network, big data, intelligent optimization control.

#### Background

Modular neural network is an effective method to solve the complex problems that fully coupled neural networks, such as BP and RBF, are difficult to learn. The most serious problem of the MNN design methods now facing is that it is difficult to determine the number of sub-modules and the structure of the sub-modules under the condition of lack of the learned objects background knowledge. In recent years, the MNN architecture design methods gradually move to brain science. Based on the fact of brain information process, the authors present an adaptive hierarchical modular neural network design method. The proposed HAMNN design method solves the problem of how to determine the number of sub-modules and the structure of the sub-modules, moreover, the proposed method requires few artificial parameters com-

pared with other MNN architecture design methods, therefore, the proposed method really achieves the black box effect on neural network in some degree. Compared with some classic fully coupled RBF network and other learning machines, HAMNN improves the learning accuracy and the generalization performance.

This research is sponsored by the National Natural Science Foundation of China under Grant Nos. 61440059, 61533002, the National Science Fund for Distinguished Young Scholars No. 61225016, the Natural Science Foundation of Liaoning Province No. 201602363, and the China Scholarship Council. The main purpose of this research is to provide a predictive model to the complex industrial systems.