

# 径向基概率神经网络结构的遗传优化\*

赵温波<sup>1,2</sup>, 黄德双<sup>2</sup>, 郭 璘<sup>2</sup>

(1. 中国科学技术大学自动化系, 合肥 230031; 2. 中科院合肥智能机械研究所, 合肥 230031)

**摘要:**运用遗传算法(GA)来优化设计径向基概率神经网络(RBPNN)结构, 优选了隐中心矢量和优化求取对应的核函数控制参数. 提出的染色体编码方式, 充分体现了所选隐中心矢量在模式样本空间中的数量及位置分布, 同时还包含了相适应的核函数控制参数信息. 新构造的适应度函数不仅有效地控制了网络输出的误差精度, 而且还能够使得RBPNN结构优化趋于最简. 将IRIS分类问题用于检验该算法的有效性并与ROLSA和MKM进行了比较研究, 结果表明, GA的优化效率最高, 而且GA优化后的RBPNN在推广能力方面也没有明显下降.

**关键词:**遗传算法; 径向基概率神经网络; 隐中心矢量; 结构优化

中图分类号: TP39

文献标识码: A

## 0 引言

径向基概率神经网络(RBPNN)<sup>[1]</sup>是在径向基函数神经网络(RBFNN)<sup>[2,3]</sup>和概率神经网络(PNN)<sup>[4]</sup>的基础之上发展起来的. 该网络的主要优点是: 在训练时间和测试速度方面, 比RBFNN要快得多, 同时它也克服了PNN无法避免的模式交错影响. 从总体性能上来说, RBPNN很好地实现了RBFNN和PNN之间的折衷, 在某些方面甚至超过前两种网络. 如图1所示, RBPNN的结构包括四层: 一个输入层、两个隐层及一个输出层. 其中, 第一隐层是非线性处理层, 它实现输入的非线性变换, 或输入样本的非线性划分; 第二隐层是对第一隐层的输出有选择性地求和与聚类. 这两层完成了该网络的主要功能. 假定RBPNN有 $L$ 个输入节点, 即输入样本 $x$ 为 $L$ 维向量, 第二隐层和输出层有 $M$ 个节点, 即输出有 $M$ 个模式, 共有 $N$ 个训练样本, 则RBPNN的输出可以描述为

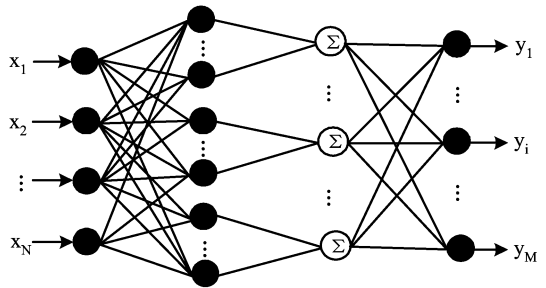


图1 径向基概率神经网络结构

Fig.1 The structure of the radial basis probabilistic neural works

\* 收稿日期: 2002-12-13

基金项目: 国家自然科学基金资助项目(60173050)

作者简介: 赵温波, 男, 1972年生, 博士生. E-mail: wbzhao@iim.ac.cn

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1M} \\ y_{21} & y_{22} & \cdots & y_{2M} \\ \vdots & \vdots & & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NM} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1M} \\ w_{21} & w_{22} & \cdots & w_{2M} \\ \vdots & \vdots & & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MM} \end{bmatrix}, \quad (1)$$

其中

$$h_{ij}(x) = \sum_{j=1}^{n_i} \phi_{ij}(\|x - c_{ij}\|_2, \sigma). \quad (2)$$

将方程(1)简写为

$$Y = HW, \quad (3)$$

其中:  $y_{ij}$ 、 $h_{ij}$  分别为网络的输出层和第二隐层第  $i$  个单元对应于第  $j$  个输入样本时的输出,  $w_{ij}$  是第二隐层第  $i$  个单元和输出层第  $j$  个单元的连接权重;  $n_i$  为网络第一隐层第  $i$  个类别的隐中心矢量数;  $c_{ij}$  是第一隐层第  $i$  个类别的第  $j$  个隐中心矢量;  $\|\cdot\|$  为 2-范数,  $\sigma$  为核函数的控制参数。(3)式为该网络映射的向量-矩阵形式.  $\phi(\cdot)$  为第一隐层的非线性映射函数(核函数), 通常使用高斯核函数, 即

$$\phi_{ij}(\|x - c_{ij}\|_2, \sigma) = \exp(-\|x - c_{ij}\|_2^2 / \sigma^2). \quad (4)$$

与 RBFNN 相类似, RBPNN 隐中心矢量和核函数控制参数是主要的网络结构参数, 如何选取这些结构参数是 RBPNN 性能发挥的关键. 如果把所有训练样本都作为隐中心矢量, 尽管可能使网络收敛到满意的精度, 但这样会使得网络结构过于庞大, 造成训练和测试时间过长. 核函数控制参数同样也是 RBPNN 的一个重要参数, 其作用机理与人类视神经系统极为相似, 宽度选择过大, 则只能观察到目标的“概貌”, 很难分辨其细微特征, 反之, 得到的只是目标的局部信息, 而无法了解目标的整体情况. 因此, 如何找到合适的控制参数, 使得 RBPNN 在不损失网络性能的条件下, 得到最简的网络规模是本文研究的重点.

遗传算法(GA)是一种模拟自然选择和遗传机制, 在计算机上实现的概率性全局寻优搜索算法. GA 的寻优机理是这样的, 首先在优化目标的高维可行解空间中随机产生多个起始点, 利用 GA 适应度函数提供的搜索方向从初始可行解同时并行搜索, 同其它的传统优化方法相比, 搜索区域宽而且效率较高. GA 无需计算梯度等启发信息, 因而对求解的目标函数也没有严格的限制, 不要求目标函数具有连续、可微等数学条件. 因此 GA 已经成为一种具有全局性、并行性、快速性、鲁棒性的成熟优化算法, 并在工程技术中得已广泛应用.

对于 RBPNN 来说, 由于必要的隐中心矢量和核函数的控制参数是由多种因素所决定的, 有时很难用精确的数学模型来求解, 即使能把模型解析式表达出来, 也很难满足可微、连续等严格的数学条件, 从而不能采用传统的最优化方法来解决. 而遗传算法在这方面却没有严格的限制, 它只要求待求解问题具有可比性就可以了. 因此, 本文讨论使用 GA 来实现 RBPNN 的结构优化.

## 1 遗传算法的构造

设计遗传算法时, 一般来说有以下几个步骤要执行, 即确定编码方式、构造适应度函数、选择(繁殖)策略的确定、设计遗传算子、选取算法控制参数等. 下面我们来详细讨论针对

## RBPNN 结构优化时 GA 的算法设计.

## 1.1 染色体编码

我们已经讨论过设计 RBPNN 网络的关键是如何选择第一隐层所必需的隐中心矢量及其相应的核函数控制参数. 隐中心矢量的选择不仅与初始数目有关, 还与其空间分布有关. 一般来说, 在不损失网络性能, 特别是网络收敛精度的条件下, 选择的隐中心矢量在数目上要尽可能少. 这是因为较少的隐中心矢量将大大简化网络的训练和测试过程. 核函数控制参数取值与隐中心矢量的数目及分布息息相关, 特定数目及空间分布的隐中心矢量, 需要特定的核函数控制参数才能使网络收敛到满意的精度, 同时, 特定取值的控制参数反过来需要不同的隐中心矢量的数目及空间分布. 有鉴于此, 我们在确定染色体编码方式时, 应该同时考虑隐中心矢量及核函数控制参数, 以期在满足网络精度且得到隐中心矢量数目最少这一优化目标时, 找出最佳的隐中心矢量分布及相应的控制参数取值.

基于上述考虑, 本文提出一种如图 2 所示的新编码方式. 考虑到隐中心矢量与相应的控制参数在优化时的作用不同, 我们采用二进制与实数混合编码来构造染色体. 图中的每个二进制基因位代表一个初选的隐中心矢量, “1”表示该样本被选择,

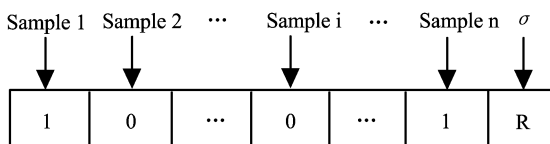


图 2 染色体编码结构

Fig. 2 The coding scheme

而“0”表示不被选择, 染色体的最后一位是实数基因, 代表相应的核函数控制参数. 从本质上来看, 该染色体的构造包含了两种类型的映射, 其一是染色体中的二进制基因位与初始的隐中心矢量的一一映射; 其二是实现了每个染色体实数位基因所表示的控制参数与该染色体中的二进制基因所表示的隐中心矢量的选择的一一映射. 染色体中二进制基因位中的“1”的个数即表示所选择的隐中心矢量的数目.

## 1.2 适应度函数构造

适应度函数是遗传算法指导搜索的唯一信息, 它的好坏是衡量算法优劣的关键. 如果适应度函数构造得当, 能使遗传算法沿着面向优化参数组合的方向去逼近最优解, 而不会导致算法不收敛, 或陷入局部最优. 从数学上来说, 适应度函数是一种映射, 即样本空间到正实数空间上的一一映射:  $f: S \rightarrow R^+$ .

本文的目标在于, 在给定的网络误差的条件下, 要尽可能少选隐中心矢量以便找出合适的核函数控制参数, 所以, 在适应度函数构造的过程中, 隐中心矢量的数目、给定的误差精度及每个个体的实际误差都必须同时考虑. 根据适应度函数最大化原则, 在满足给定的误差时, 个体的适应度值应该随着所选择的隐中心矢量数目的增加而减少. 当某一个体对应的网络实际误差超过给定的误差标准时, 说明该个体的隐中心矢量的选取不合适或者相应的核函数控制参数的取值不当, 这时该个体的适应度值应该置 0, 不必再去探讨隐中心矢量分布及相应的控制参数. 本文的适应度函数定义如下:

$$f(e, X_i) = \frac{C(s+1)}{N_c 2^s}, \quad (5)$$

$$s = \text{sign}(\varepsilon - e) \quad N_c = \sum_{k=1}^{l-1} X_i(k), \quad (6)$$

其中:  $e$  为个体对应的实际输出误差,  $X_i$  表示当前个体,  $C$  为常数,  $s$  为判别当前个体是否满足给定误差的开关量,  $N_c$  表示当前染色体中隐中心矢量数目,  $\varepsilon$  是给定的误差标准,  $X_i(k)$  表示个体  $X_i$  中第  $k$  个基因值,  $l$  代表染色体长度.

在上述适应度函数的构造中, 个体的实际输出误差  $e$  是未知参数, 下面来分析这一参数的求解过程. 首先, 讨论 RBPNN 模型的代价函数. 对于大多数外监督学习神经网络来说, 通常使用的优化准则是均方误差准则. RBPNN 模型所使用的代价函数可以定义为

$$J(W) = \frac{1}{2} \|Y_d - HW\|_F^2, \quad (7)$$

其中,  $Y_d$  指外监督信号(期望信号)矩阵,  $\|\cdot\|_F$  是 Frobenius 范数(即 F-范数). 对于(7)中的  $H$  进行矩阵的正交分解适当变换后得

$$H = Q[R, \dots, 0]^T, \quad (8)$$

$$Q^T Y_d = [\hat{Y} \quad \bar{Y}]^T, \quad (9)$$

其中,  $Q$  为正交矩阵,  $R$  为上三角方阵.  $\hat{Y}$ 、 $\bar{Y}$  分别为正交化后的期望输出矩阵与残差矩阵.

将(8)和(9)式代入(7)式, 基于矩阵 F-范数的正交不变性, (7)式可以变成如下形式:

$$\begin{aligned} J(W) &= \frac{1}{2} \left\| \left\{ Q \begin{bmatrix} \hat{Y} \\ \bar{Y} \end{bmatrix} - Q \begin{bmatrix} R \\ 0 \end{bmatrix} W \right\} \right\|_F^2 = \frac{1}{2} \left\| \begin{bmatrix} \hat{Y} - RW \\ \bar{Y} \end{bmatrix} \right\|_F^2 \\ &= \frac{1}{2} \|\hat{Y} - RW\|_F^2 + \frac{1}{2} \|\bar{Y}\|_F^2, \end{aligned} \quad (10)$$

其中  $\|\bar{Y}\|_F^2$  是  $J(W)$  的残差, 也就是网络的实际误差, 即

$$e = \|\bar{Y}\|_F^2. \quad (11)$$

当然, 还可以通过求取网络连接权重的途径来计算网络的实际误差, 具体的计算过程如下:

以权重矩阵  $W$  为变量, 当(10)式取最小值时, 必有  $\frac{1}{2} \|\hat{Y} - RW\|_F^2 = 0$ , 即

$$\hat{Y} = RW, \quad (12)$$

则权重矩阵为

$$W = R^{-1}\hat{Y}. \quad (13)$$

有了权重矩阵  $W$ , 实际误差为

$$e = \|Y_d - HW\|_F^2, \quad (14)$$

实际上, (11)式与(14)式在数值上是完全相等的.

### 1.3 选择策略(繁殖策略)

选择(繁殖)策略直接影响遗传算法的性能. 选择策略与种群中的选择压力相关, 选择压力大, 将导致种群中最优个体被大量复制, 算法会很快收敛, 但有时也会陷入局部最优, 使算法丧失全局搜索能力. 反之, 较小的选择压力增加了群体的多样性, 从概率意义上, 增加了算法收敛到全局最优解的能力, 但算法的收敛速度一般比较慢. 通常的选择策略有基于适应度比例选择、基于排名选择及基于局部竞争机制选择等<sup>[5]</sup>. 本文采用基于适应度比例选择及最佳个体保持(elitist strategy)的选择(繁殖)策略. 假定个体  $X_i$  的适应度为  $f_i$ , 则其被选择的概率  $P_{si}$  为

$$P_{si} = f_i / \sum f_j \quad (15)$$

其中  $\sum f_j$  为群体中的个体适应度的总和。

我们用赌轮选择方式<sup>[5]</sup>来实现按照概率  $P_{s_i}$  选择个体  $X_i$ 。具体的步骤如下:先计算当前种群每个个体的选择概率  $P_{s_1}, P_{s_2}, \dots, P_{s_N}$ , 然后产生一个  $[0, 1]$  内的随机数  $r$ , 若  $P_{s_1} + P_{s_2} + \dots + P_{s_{i-1}} < r \leq P_{s_1} + P_{s_2} + \dots + P_{s_i}$ , 则个体  $X_i$  被选择到配对库; 否则, 该个体就不被选择。

最佳个体保持繁殖策略的基本思想为: 选择当前种群中的最优个体  $X_i = \arg \max_j \{f(X_j)\}$ , 使其不参与交叉和变异, 直接成为下一代种群的成员。可以证明<sup>[6]</sup>, 采用最佳个体保持繁殖策略的种群进化序列  $X_1, X_2, \dots, X_{G_{\max}}$  ( $X_i$  为第  $i$  代种群, 即  $X_i = \{X_{i1}, X_{i2}, \dots, X_{iS_g}\}$ ,  $S_g$  为种群规模,  $G_{\max}$  为最大进化代数), 必将以概率 1 收敛于最优种群集  $M$  的子集  $M_0^*$ , 即  $\lim_{n \rightarrow \infty} P\{X_n \in M_0^* / X_0\} = 1$  ( $X_0$  为初始种群)。

#### 1.4 遗传算子的设计

遗传算子主要包括交叉和变异两个算子。交叉算子是遗传算法的重要组成部分, 是模仿生物进化中遗传基因的重组原理, 通过交换、重组两个母体中的部分染色体结构, 从而生成新个体的操作。交叉算子使得遗传算法对最优解的搜索能力得以大幅度提高。一般来说, 交叉算子的实现分两步进行: 首先以交叉概率随机地从母体配对库中取出两个配对染色体, 然后以某种交叉方法对这两个配对染色体进行交叉。常见的交叉方式有单点(随机)交叉、两点(随机)交叉、多点(随机)交叉及均匀交叉等。文中采用多点(随机)交叉方式。由于本文染色体采用混合编码, 而且只有一个实数编码基因, 故将该实数基因位作为一个固定的交叉点, 其它几个交叉点由随机方式产生, 这样, 实数基因的固定交叉与二进制基因的随机交叉将共同实现求解 RBPNN 的最佳网络结构的多点随机交叉操作。

变异算子通过对所有染色体中随机地确定基因位以变异概率改变该基因值, 目的是增强遗传算法的局部搜索能力, 同时增加了群体的多样性。较早的遗传算法偏重于把变异算子作为辅助手段, 近年来的研究表明, 变异算子对遗传算法的贡献同交叉算子一样具有举足轻重的地位。本文针对求解最佳 RBPNN 网络结构这一问题的特殊性, 提升了变异算子的作用, 主要是通过增大变异概率的取值来实现。

本文采用的是实数与二进制混合编码, 由于这两种编码方式在变异操作上有很大的差异, 所以我们加以区别对待, 对于二进制编码, 以变异概率对每个基因位进行逻辑取反操作; 而对于实数编码, 采用高斯变异方式。具体的变异方法如下:

$$\begin{cases} B' = \text{not}(B) & \text{二进制基因,} \\ \sigma' = \sigma + a \sqrt{f_i} \mu(0, 1) & \text{实数基因.} \end{cases} \quad (16)$$

$B, B'$  分别为染色体中二值基因位变异前后的取值,  $\text{not}(\cdot)$  为逻辑取反运算,  $\sigma, \sigma'$  分别为染色体中实数基因位变异前后的取值,  $a$  为  $[-1, 1]$  间的实数,  $\mu(0, 1)$  标准正态分布的随机数发生器,  $f_i$  为个体适应度。对于二值基因, 变异概率在  $[0.001, 0.010]$  之间取值。对于实数基因, 变异概率在  $[0.08, 0.10]$  内取值, 提高了实数基因位的变异概率, 这样做的目的是为了扩展对于核函数控制参数的寻优能力。

#### 1.5 选取控制参数

控制参数主要包括群体规模、算法执行的最大迭代数、交叉概率以及变异概率等。

群体规模直接影响遗传算法的性能, 规模太小, 则会使种群缺乏多样性, 易使算法陷入

局部最小,规模太大时,对于 RBPNN 来说,在初始隐中心矢量数目比较多的情况下,大大增加了算法的计算复杂度.一般来说,种群规模在 50 ~ 200 之间时,能够很好地实现种群多样性与算法复杂度之间的折衷.

由于遗传算法不包含目标函数的梯度等启发信息,当然也就无法确定个体在整个解空间的位置分布,更没有办法使用传统方法来判定算法的收敛与否,所以一般使用最大迭代代数来终止算法,也就是初始种群的最大进化代数.

交叉概率及变异概率是统计意义上个体参与交叉与变异操作的度量标准,在参数取值上,也能充分反映交叉算子与变异算子对遗传算法的贡献.交叉及变异概率可以是固定取值,也可以自适应取值.本文使用固定的概率.

为了下文说明的方便,本文分别使用  $S_g$ 、 $G_{\max}$ 、 $P_c$  及  $P_m$  代表群体规模、最大进化代数、交叉概率、变异概率等控制参数.

## 2 模拟结果与讨论

为了验证本文所提算法的有效性,我们使用 RBPNN 来分类 IRIS 问题,并使用 GA 来优化网络结构.有关控制参数为:( I )种群规模为  $S_g = 64$ ; ( II )交叉概率为  $P_c = 0.8$ ; ( III )二值基因变异概率为  $P_m^b = 0.005$ ,实数基因变异概率为  $P_m^r = 0.09$ ; ( IV )算法执行的最大代数为  $G_{\max} = 1\ 000$  代;由于本文的目标在于,在给定的收敛误差标准下得到最少隐中心矢量数目及其对应的最优核函数控制参数.对于分类问题来说,给定的收敛误差标准还可以理解为只要没有或很少错误划分训练样本,就应该认为所选的网络结构满足要求.因此,( 14 )式可以改写为

$$e = \|Y_d - \text{round}(HW)\|_F^2. \quad (17)$$

$\text{round}(\cdot)$  为四舍五入算子,可以理解为,对于给定的输入训练样本  $x_i$  (属于第  $k$  类),期望信号为向量  $[0, \dots, \underset{k}{1}, \dots, 0]^T$ ,则在网络的实际输出中,只有当输出向量  $[o_1, \dots, o_k, \dots, o_m]^T$  满足第  $k$  个分量  $o_k \geq 0.5$  且  $o_i < 0.5, i = 1, 2, \dots, k-1, k+1, \dots, N$  时,才能认为已经对输入样本  $x_i$  进行了正确分类.当期望信号  $Y_d$  为二值信号时,  $e$  的可能值为:  $0, 1, 2, 3, \dots, N$ . 这就是本文算法对于网络输出正确分类的判别准则.本实验的收敛误差标准为 0,即,对于所有的训练样本,网络无一例错误分类.网络测试时的判别准则与训练时相同.

IRIS 是一个经典的模式识别问题<sup>[7]</sup>,它有 3 种花型,即 *setosa*、*versicolor* 和 *virginica*,每个花型按花瓣长、宽,及花萼长、宽采样,形成了维数为 4 的样本.假设三类花型共生成 150 个样本.我们将 150 个样本分为两部分,即训练样本集与测试样本集,每个样本集包括 75 个样本.用训练样本集训练网络并优选网络的隐中心矢量和相应的核函数控制参数,然后,用测试样本集来检测优化后的网络性能.

结果显示,75 个初始隐中心矢量被压缩到 8 个,相应的核函数控制参数值为 0.344 8,隐中心矢量数目随着进化代数的变化见图 3.可以看出,所选的隐中心矢量数目(8)仅占总训练样本(75)的 11%,因此该网络结构已被大大简化.相应的核函数控制参数的进化情况如图 4,可以看出,在算法迭代过程中,所选取的隐中心矢量所对应的核函数控制参数的取值并不唯一,而是一个取值范围,并且不同数目的隐中心矢量对应的核函数控制参数的取值

也不同. 另外, 被选中的隐中心矢量的数目越少, 相应的核函数控制参数的取值范围也越小. 当隐中心矢量数目收敛到 8 时, 相应的控制参数则在  $[0.30, 0.35]$  范围内取值. 控制参数取值范围规律为网络的设计和训练提供了更多的灵活性.

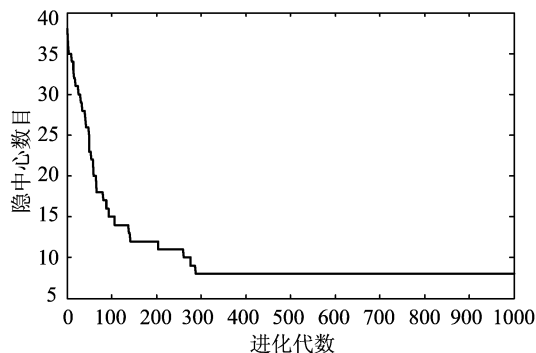


图3 隐中心矢量数目的进化曲线

Fig. 3 The evolving curve of the number of hidden center vectors

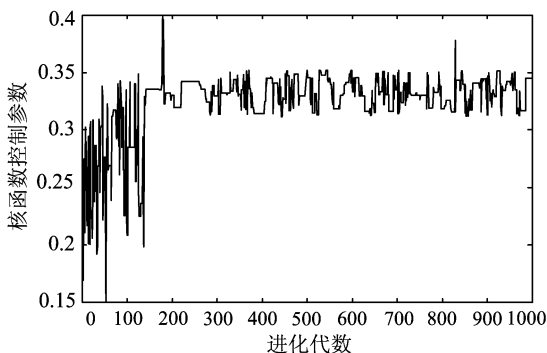


图4 核函数控制参数的进化曲线

Fig. 4 The evolving curve of the controlling parameter of kernel function

为了对比 GA 的优化性能, 分别使用 ROLSA<sup>[8]</sup> 和改进的 K 均值聚类方法 (Modified K-Means Algorithm, MKM)<sup>[9]</sup> 来优化 RBPNN 的隐中心矢量. 由于 ROLSA 与 MKM 只能优化隐中心矢量, 而不具有对核函数控制参数的优化求取能力, 因此, 这两种方法的核函数控制参数的取值采用文献[10]中的启发性表达式, 即,  $\sigma = d_{\max} / \sqrt{N} d_{\max}$  为训练样本的最大欧拉距离). 需要指出的是, MKM 主要用于已知类别数的聚类分析方法, 在用于优化 RBPNN 的隐中心矢量时, 为了尽可能地缩减所选的隐中心矢量数目, 对于每个聚类, 只选择聚类中心(最大最小中心<sup>[9]</sup>)为隐中心矢量. 此外, 我们并不关心样本集的最大聚类数,

表1 GA, ROLSA 及 MKM 对于求解 IRIS 分类问题时的 RBPNN 的结构优化性能对比

Tab. 1 Comparison of the structure optimization performance for the RBPNNs solving IRIS classification problem by GA, ROLSA and MKM

| 算法    | 隐中心矢量数 | 结构压缩率 | 控制参数   |
|-------|--------|-------|--------|
| GA    | 8      | 11%   | 0.3448 |
| ROLSA | 12     | 16%   | 0.5998 |
| MKM   | 26     | 35%   | 0.5998 |

而是随着聚类数的增加, 即隐中心矢量数目增加的情况下, 如果所构造的网络能够正确分类所有训练样本, 那么就停止聚类. 由三种方法的优化结果表 1 可以看出, GA 优化的隐中心矢量对比 ROLSA 与 MKM, 在数目上最少, 即 GA 对 RBPNN 结构压缩性能最好, 而且 GA 方法还能够同时提供与所选择的隐中心矢量相适应的核函数控制参数, 这是 ROLSA 及 MKM 算法所不具备的.

为了进一步验证 GA 优化后的 RBPNN 的性能, 研究 GA 所选的隐中心矢量在数目最少的情况下, 网络的推广能力是否下降. 我们用带有不同噪声电平的测试样本集来测试三种方法构造的网络. 75 个原始测试样本混合不同方差的高斯白噪声(零均值), 组成 20 组不同噪声电平的测试样本集. 用这些带噪声的测试样本来测试这三个网络及未进行结构优化(所有 75 个训练样本都作为隐中心矢量)的 RBPNN, 测试的结果详见表 2. 从得到的结果可以看出, 尽管 GA 对 RBPNN 的结构压缩率最高, 但在推广能力方面, GA 与 ROLSA、MKM 及未

优化的网络没有明显的区别,这说明使用 GA 优化 RBPNN 的隐中心矢量及核函数控制参数是成功的,并没有在减少隐中心矢量数目的同时以损失网络的推广能力为代价。

表 2 GA,ROLSA,MKM 优化的 RBPNN 和未优化的 RBPNN 的推广能力比较

Tab.2 Comparison of the generalization ability of the RBPNNs by GA, ROLSA, MKM and non-optimized RBPNN

| 算法   | 噪声方差 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |
|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
|      | 0    | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |    |
| GA   | 正确   | 72  | 68  | 66  | 61  | 65  | 58  | 53  | 57  | 58  | 56  | 54  | 44  | 56  | 51  | 58  | 54  | 51  | 53  | 47  | 43 |
|      | 误识   | 3   | 7   | 9   | 14  | 10  | 17  | 22  | 18  | 17  | 18  | 21  | 26  | 18  | 18  | 13  | 19  | 19  | 20  | 16  | 19 |
|      | 拒识   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 5   | 1   | 6   | 4   | 2   | 5   | 2   | 12  | 13 |
| ROLS | 正确   | 68  | 66  | 67  | 62  | 63  | 59  | 54  | 63  | 60  | 61  | 55  | 49  | 57  | 51  | 51  | 48  | 54  | 56  | 49  | 36 |
|      | 误识   | 7   | 9   | 8   | 13  | 12  | 16  | 21  | 12  | 15  | 14  | 20  | 21  | 17  | 21  | 20  | 25  | 17  | 17  | 17  | 26 |
|      | 拒识   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 5   | 1   | 3   | 4   | 2   | 4   | 2   | 9   | 13 |
| MKM  | 正确   | 69  | 65  | 67  | 61  | 64  | 59  | 55  | 61  | 61  | 58  | 55  | 52  | 61  | 56  | 59  | 49  | 51  | 53  | 45  | 37 |
|      | 误识   | 6   | 10  | 8   | 14  | 11  | 16  | 20  | 14  | 14  | 17  | 20  | 20  | 13  | 16  | 14  | 24  | 20  | 21  | 21  | 24 |
|      | 拒识   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 1   | 3   | 2   | 2   | 4   | 1   | 9   | 14 |
| FULL | 正确   | 72  | 68  | 66  | 62  | 65  | 64  | 57  | 64  | 60  | 60  | 57  | 50  | 59  | 56  | 58  | 52  | 56  | 52  | 53  | 39 |
|      | 误识   | 3   | 7   | 9   | 13  | 10  | 11  | 18  | 11  | 15  | 15  | 18  | 23  | 15  | 17  | 15  | 23  | 17  | 22  | 18  | 26 |
|      | 拒识   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 1   | 2   | 2   | 0   | 2   | 1   | 4   | 10 |

注:FULL 代表未优化的 RBPNN.

### 3 结论

本文讨论使用遗传算法来优化径向基概率神经网络(RBPNN)的结构,即优化隐中心矢量数目及空间的分布以及求取最优的核函数控制参数.文中所给方法的优势在于能够充分利用遗传算法的全局搜索能力,使得在网络分类精度不损失的情况下,得到尽可能简化的 RBPNN 结构.本文新颖之处在于提出了一种新的染色体编码方式和适应度函数构建方法,来优化 RBPNN 结构.所提出的染色体编码不仅考虑了初始隐中心的数目,还兼顾了隐中心矢量在样本空间的分布信息,同时新构造的适应度函数还将网络精度控制充分考虑在内.最后,IRIS 分类问题用于检验该算法的有效性,结果表明,75 个训练样本中只选取了 8 个典型矢量,从而大大简化了 RBPNN 的模型结构.此外,GA 还被用来与 ROLSA 和 MKM 算法进行了比较研究,结果显示该 GA 方法对 RBPNN 的结构优化效果最好,而且 GA 优化后的 RBPNN 在推广能力方面并没有明显下降.

### 参 考 文 献

- [1] Huang D S. Radial basis probabilistic neural networks: Model and application [J]. International Journal of Pattern Recognition and Artificial Intelligence. 1999,13(7):1 083-1 101.
- [2] Moody J E, Darken C J. Fast learning in networks of locally tuned processing units [J]. Neural Computation, 1989, 1:281-289.
- [3] 缪报通,陈发来.径向基函数网络在散乱数



- 据插值中的应用 [ J ]. 中国科学技术大学学报, 2001, 31( 2 ): 135-142.
- [ 4 ] Specht D F. Probabilistic neural networks [ J ]. Neural Networks. 1993, 3: 109-118.
- [ 5 ] 陈国良, 王煦法, 庄镇泉, 等. 遗传算法及其应用 [ M ]. 北京: 人民邮电出版社, 1996.
- [ 6 ] 张文修, 梁怡. 遗传算法的数学基础 [ M ]. 西安: 西安交通大学出版社, 2000.
- [ 7 ] Fisher R A. The use of multiple measurements in taxonomic problems [ J ]. Annals of Eugenics, 1936, ( 7 ): 179-188.
- [ 8 ] Zhao W B, Huang D S. Application of recursive orthogonal least squares algorithm to the structure optimization of radial basis probabilistic neural networks [ A ]. The 6th Int. Conf on Signal Processing ( ICSP02 ) Proceeding [ C ]. Beijing, 2002. 1 211-1 214.
- [ 9 ] Wilpon J G, Rabiner L R. A modified K-Means clustering algorithm for use in isolated work recognition [ J ]. IEEE Trans. Acoustics, Speech, and Signal Processing, 1985, 33( 3 ): 587-594.
- [ 10 ] Haykin S. Adaptive Filter Theory [ M ]. Upper Saddle River: Prentice-Hall, 1996.

## Genetic Optimization of the Structure of Radial Basis Probabilistic Neural Networks

ZHAO Wen-Bo<sup>1,2</sup>, HUANG De-Shuang<sup>2</sup>, GUO Lin<sup>2</sup>

( 1. Department of Automation, University of Science and Technology of China )

( 2. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, Anhui, China )

**Abstract:** The genetic algorithm is used to optimize the structure of the radial basis probabilistic neural networks ( RBPNN ), including selecting the hidden center vectors of the first hidden layer and determining the corresponding controlling parameters of the kernel function of RBPNN. The proposed genetic encoding method completely embodies the number and position distribution for the selected hidden center vectors in the pattern sample space, and involves information of the corresponding controlling parameters of the kernel function. The novel constructed fitting function can not only efficiently control the error accuracy of the RBPNN output, but also make the optimized RBPNN approach the most parsimonious structure. For a validity test, the proposed GA is used to solve IRIS classification problem and compared with ROLSA and MKM. The experimental results show that, the optimized performance of the proposed GA is the best among the three methods, with no obvious drop observed in the generalization performance of the optimized RBPNN by GA.

**Key words:** genetic algorithms; radial basis probabilistic neural networks; hidden center vectors; structure optimization