基于覆盖算法的两层结构分类器设计

万 忠[1],张 铃[1,2],张燕平[1,2],陈 洁[1]

(1.安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039;
2.安徽大学 人工智能研究所,安徽 合肥 230039)

摘 要:分类器设计是模式识别系统中的关键步骤之一。在目前的许多设计方法中,分类器大多采用的是单层结构,即直接将输入模式映射为识别出来的结果,这类结构虽然简单直观,但是往往难于发挥分类器设计算法的最大性能。文中从分类器的结构方面考虑,提出了一种基于覆盖算法的两层结构分类器的设计方法,并且与单层结构分类器做了实验分析对比,得出了在不明显增加构造复杂度的情况下两层结构的设计大大改善了分类器的性能。

关键词:分类器设计;模式识别;两层结构分类器;覆盖算法

中图分类号:TP391.4 **文献标识码**:A **文章编号**:1673-629X(2007)01-0065-04

A Two-Layer Classifier Design Method Based on Covering Algorithm

WAN Zhong[1], ZHANG Ling[1,2], ZHANG Yan-ping[1,2], CHEN Jie[1]

(1. Ministry of Education Key Lab. of Intelligent Computing & Signal Processing at Anhui University, Hefei 230039, China;
2. Institute of Artificial Intelligence, Anhui University, Hefei 230039, China)

Abstract: The classifier design is a key step for pattern recognition systems. Among the many classifier design methods existed, the majority of them adopt single-layer structure, that is to say, directly mapping the input pattern into the comprehensible results. Although the single-layer structure looks simple and intuitionistic, it often becomes an obstacle to thoroughly bring into play the maximized performance of the classifier. This paper takes the structure of classifier itself into consideration, and puts forward a two-layer classifier design method based on the covering algorithm. Then we do an experiment and make analysis compared with the single-layer structure, and in the end we get the conclusion that two-layer structure design observably improve the performance of the classifier with not increasing the constructing complexity of the classifier at the same time.

Key words: classifier design; pattern recognition; two-layer classifier; covering algorithm

0 引 言

分类器设计是模式识别系统中的非常关键的一个步骤[1],它直接影响到系统的识别能力。然而在目前的分类器设计方法中,大多数采用单层的结构,即直接将系统输入的特征模式映射成可人为理解的识别结果,这样整个的映射功能实现就集中于单层之上,这增加了单层结构分类器的功能实现难度,也影响了这类分类器的实际性能。文献[2]中采用了双层结构,它让待识别的手写字符先经过第一层分类器粗分类为字符与数字,然后再经过第二层分类器得出最后的识别结果。然而文献[2]双层结构设计中并没有充分考虑不同类别的学习样本的空间分布特征,由于分类器的学习训练主要依赖于样本,所以采用两层的结构来设计分类器并同时考虑学习样本的空间分布才能既分化了单层结构的功能实现难度又能充分发挥分类器的运行性能。

正是基于上面的基本思路,文中以覆盖算法这一构造性很强的算法为分类器设计算法,提出了一种两层结构的分类器设计方法,并且在每一层的设计过程中充分考虑学习样本本身的空间分布特征。随后所做的字符识别对比实验与分析,说明了与单层结构相比,这种双层结构的设计在没有明显增加分类器设计的计算复杂度的情况下,随着学习样本增加到一定值后,分类器很快获得了更高的识别正确率,同时减少了分类器的学习训练时间。

1 覆盖算法介绍

张铃教授于1997年就给出了M-P神经元模型的几何意义[3],指出用三层神经网络构造分类器,等价于求出

收稿日期:2006-04-07

基金项目:国家973计划资助项目(2004CB318108);国家自然科学基金资助项目(60475017);教育部博士点基金(20040357002)

作者简介:万 忠(1980-),男,安徽合肥人,硕士研究生,研究方向为人工智能、神经网络和智能计算技术以及其在图像处理和智能交通工程的应用;张 铃,教授,博士生导师,从事人工智能理论、机器学习理论和方法、智能计算技术、神经网络技术的研究;张燕平,博士,教授,研究方向为人工神经网络、机器学习、人工智能及在金融工程中的应用。

一组领域,这组领域能将不同类的点分隔开来,并进一步给出覆盖设计算法[4~6]。

定义 1: 覆盖 C 是指 n 维欧氏空间的的一个球形领域(以 a 为中心,以 r 为半径的开超球)。

定义2:给定样本集 S 分为 k 类,表示为集合 $S=\{S^1,S^2,\cdots,S^k\}$,$S^i(i=1,\cdots,k)$ 为属于第 i 类的样本集。如果覆盖集 $C=\{C^1,C^2,\cdots,C^p\}$,$C^j(j=1,\cdots,p)$ 均为覆盖,满足 $C^i\cap C^j$ 为空集($i\neq j$),并且每个 C^j 只和一个 S^i 相交以及 C^j 的并覆盖整个 S,则称 C 为 S 的划分覆盖集。

该算法的主要思路是:先求一个领域 C^1,它只覆盖一类中的点,而不覆盖其它类的点;对余下的点求二类覆盖领域 C^2,它只覆盖二类中的点而不覆盖其它类的点;……如此交叉进行覆盖,直到样本集中的点均被领域覆盖了为止。

覆盖算法的实质就是用求出的覆盖领域作为三层网络的隐含层,输入层为测试集,输出层为测试集的分类结果。构造覆盖算法的三层前向网络 FP 学习算法具体神经网络各层的设计算法[6] 如下:

设给定样本集为 $K=\{x^1,x^2,\cdots,x^k\}$(K 为 n 维欧氏空间的点集)。设 K 分为 s 个子集 $K^1=\{x^1,x^2,\cdots,x^{m(1)}\},\cdots,K^s=\{x^{m(s-1)+1},x^{m(s-2)+2},\cdots,x^k\}$,通过三层网络后,属于 K^i 的点的输出均为"y^i",其中 $y^i=(0,\cdots,1,0,\cdots,0)$(只有第 i 个分量为 1),$i=1,2,\cdots,s$。

第 1 层输入层直接输入测试样本特征向量;

第 2 层隐藏层,设计 p(p 为覆盖个数)个神经元 $A^1,\cdots,A^p$ 分别对应于覆盖 C^i,$i=1,2,\cdots,p$;

第 3 层输出层,取一个输入 x^j,其与 A^i(设以 a_i 为中心,r_i 为半径)神经元的权和阈值($W^i=(\omega^i)$,$\theta=(\theta_i)$),则 $\theta_i=r_i$,$W=(a^i)$,$\theta=(\theta_i)$,$r(x)=<a^i,x>$。如 $r(x)\geqslant r_i$,则对应的 y^j 的第 i 个分量为 1,否则为 0。这样的三层前向神经网络就构成了分类器,功能是将 K 输入样本自动分为 s 个类别。

2 两层结构分类器设计

2.1 分类器的结构与功能

这里以一个 0-9 字符识别的实例来描述这种分类器的设计方法。分类器采用两层结构的分类器:第一层分类器是粗分类器,将待识别字符分为几个大类,大类的划分原则是考虑两个字符间的特征向量分布相似性;第二层分类器是细分类器,给每个大类别分别设计一个细分类器,划分的标准就是将大类中划分为字符的 0-9 这 10 个类别。该算法处理过程示意图如图 1 所示。

2.2 两层分类器的详细设计

文中以处理的 0-9 字符作为样本,共获得归一化的样本 1326 个,其中 0-9 的字符个数分别为 200,200,200,155,89,42,200,56,131 和 53,各字符占字符总数的比例依次为 15.1%,15.1%,15.1%,11.7%,6.7%,3.2%,15.0%,4.1%,10.0% 和 4.0%。

下面对粗分类器和细分类器分别进行介绍:

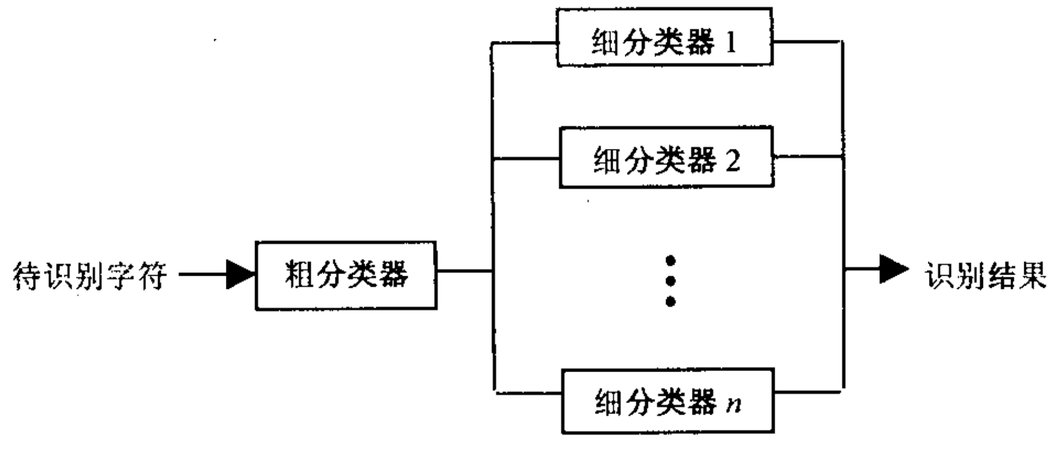

图 1 两层结构的分类器结构

1) 粗分类器:

设需要学习的标准样本为 $K=\{(X^1,y^1),(X^2,y^2),\cdots,(X^t,y^t)\}$,其中 $\boldsymbol{X}^i$ 表示第 i 个字符样本的特征向量,y^i 表示第 i 个字符样本的类别标记,文中的系统主要识别 0-9 这 10 个类别,$y^i\in\{0,1,\cdots,9\}$,$i=1,2,\cdots,t$。这里 $\boldsymbol{X}^i$ 是单个字符经过归一化的 15 * 20 模板按照行顺序所排列的 300 维像素点特征向量,即 $\boldsymbol{X}^i=\{x_1^i,x_2^i,\cdots,x_{300}^i\}$,其中 $x_j^i\in\{0,1\}$,$i=1,2,\cdots,t$,$j=1,2,\cdots,300$。

粗分类器的划分按照下面来进行,将特征向量比较接近的都划分为一个大类,这里并不是利用聚类算法来对所有学习样本进行划分,因为文中处理的字符主要是 10 个数字字符,类别数不多,所以这里采用一种间接的方法。首先同一个类别的样本一定属于是特征向量比较接近的,另外一类不同类别之间存在有特征向量比较接近的。这里用下列的定量分析来寻找不同类别间的比较接近的特征向量。取 0-9 这 10 个学习的样本字符各 25 个,然后对每个字符类别,求得 25 个字符样本的重心作为该类别的标准样本,然后求出各类别的标准样本间的距离,这里求出了内积距离。以 R_i 表示类别 i 的标准样本,$i=0,1,\cdots,9$。

因为文中的分类器采用的是覆盖算法,覆盖算法中在使用将样本点投影在一个半超球面上(如图 2 所示),设输入的定义域为 n 维空间中的有界集合 D,令 S^n 是 $n+1$ 维空间中的 n 维的超球面:

$T:D\rightarrow S^n$,$T(x)=[x,\sqrt{(d^2-|x|^2)}]$,其中 $d\geqslant\max\{|x|\,|\,x\in D\}$

这个变换可从几何上直观地理解为:将 D 看成是位于 $n+1$ 维空间中过原点的一个 n 维超平面[5] 上,而且 D 位于 S^n 的内部,则变换 T 就是将 D 上的点垂直投射到 S^n 的上半球面上。这种变换显然是一一对应的。然后以两样本的内积距离作为相似性度量方法,内积值越大,说明这两个样本越是相似。这里将样本点投影到半超球面的半径值 R 为 12。

对照表 1,可做下面的归纳:

与其他类别相比,和 R_1 明显最为接近的是类 R_7;

a. 对于 R_2 和 R_4,这两个类别与其他类别的标准模式相差较多,没有明显模式接近的;

b. 类 R_0 与 R_8、R_6 模式接近,R_3 与 R_8、R_9 模式接近,R_5 与类 R_8、R_6 模式接近,类 R_6 与类 R_8、R_5 模式接近,类 R_8 与类 R_6、R_3 模式接近,类 R_9 与类 R_3、R_8 模式接近。

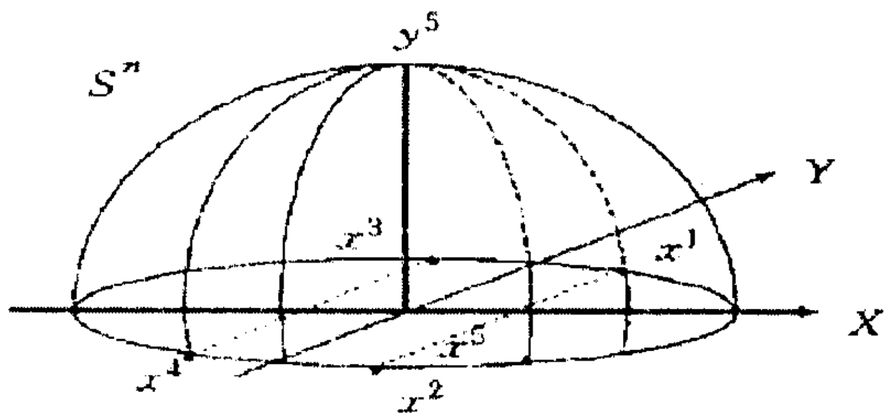

图 2　从 $D \to S^n$ 变换的示意图

表 1　0 − 9 字符标准样本内积距离

	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9
R_0	144	94.24	115.84	125.82	112.16	126.49	131.43	101.89	132.14	128.94
R_1	94.24	144	102.33	114.48	117.14	106.15	92.579	128.99	103.35	113.01
R_2	115.84	102.33	144	125.53	106.82	116.23	114.92	115.84	122.96	122.11
R_3	125.82	114.48	125.53	144	122.63	132.17	125.52	122.54	134.76	132.98
R_4	112.16	117.14	106.82	122.63	144	116.81	110.41	115.37	120.62	119.68
R_5	126.49	106.15	116.23	132.17	116.81	144	133.46	111.02	134.12	130.32
R_6	131.43	92.579	114.92	125.52	110.41	133.46	144	98.487	136.12	123.86
R_7	101.89	128.99	115.84	122.54	115.37	111.02	98.487	144	110.53	118.34
R_8	132.14	103.35	122.96	134.76	120.62	134.12	136.12	110.53	144	134.05
R_9	128.94	113.01	122.11	132.98	119.68	130.32	123.86	118.34	134.05	144

依照上面的归纳,可以按照下列方式进行划分:令 $R(i,j)$ 表示 R_i, R_j 的内积距离,$R(i)$ 表示 $R(i,j)$ 对所有 $j \neq i$ 求平均值,如果 $R(i,j) > aR(i)$,则将 i,j 分为一个大类;如果 i,j 为同一大类,j,k 为同一大类,则 i,j,k 为同一大类,这里 a 取 0.88。这样粗分类器的大类别划分具体如下:分类 4 个大类,分别为 $\{R_2\}$,$\{R_4\}$,$\{R_1, R_7\}$,$\{R_0, R_3, R_5, R_6, R_8, R_9\}$,大类别标记为 a 类,b 类,c 类和 d 类。$K = \{(X^1, y^1), (X^2, y^2), \cdots, (X^t, y^t)\}$ 中的 $y^i (i = 1,2,\cdots,t)$ 按照大类划分更新为 a、b、c 或者 d。接着依照上面的介绍,利用覆盖算法构造 3 层神经网络作为图 1 中的粗分类器。

2) 细分类器:

对照图 1,还要构造两个细分类器分类处理 c 类和 d 类这两个大类,即 $\{R_1, R_7\}$ 和 $\{R_0, R_3, R_5, R_6, R_8, R_9\}$。而对于 a 类和 b 类因为已经识别出了最后的字符类别,可以在识别过程中直接将 a 类转换为“2”,b 类转换为“4”输出结果。

对于处理 c 类的细分类器,其学习的样本集合为 $\{(\boldsymbol{X}^i, y^i) \mid (\boldsymbol{X}^i, y^i) \in K, y^i \in \{1,7\}\}$,其中 $K = \{(X^1, y^1), (X^2, y^2), \cdots, (X^t, y^t)\}$;对于处理 d 类的细分类器,其学习的样本集合为 $\{(\boldsymbol{X}^i, y^i) \mid (\boldsymbol{X}^i, y^i) \in K, y^i \in \{0,3,5,6,8,9\}\}$。然后按照上面介绍的覆盖方法分别构造出这两个细分类器。这两个分类器以粗分类器的识别类别为 c 类或者 d 类的输入特征向量为其输入,识别结果为 c 类包含的 R_1, R_7 以及 d 类包含的 $R_0, R_3, R_5, R_6, R_8, R_9$。

3　实验与分析

实验在 P42.9GHz、512M 内存以及 Win2003 平台下 Matlab7.0 运行。

实验按照学习样本与测试样本的比例分别是 1:2、1:1、3:1、5:1 和 7:1 这 5 种情况,每一种比例下分别给出覆盖算法构造的单层分类器与两层结构分类器的识别正确率、训练时间(学习样本所花的时间)、领域覆盖数(主要是分类器构造的复杂程度描述)。其中两层结构分类器的总评估参数按照下面的方式计算:训练时间与领域覆盖数可以将粗分类器和两个细分类器相应相加即可;识别正确率按照以下方式计算,设 CR_1, CR_{2c}, CR_{2d} 分别表示第一层粗分类器、第二层 c 类细分类器、第二层 d 类细分类器的识别正确率(CR 即 Correct Ratio),Pr_c, Pr_d 分别为待识别字符需要通过 c 类和 d 类细分类器处理的概率,这个利用样本分布得到,$Pr_c = 15.1\% + 4.1\% = 19.2\%$,$Pr_d = 15.1\% + 11.7\% + 3.2\% + 15.0\% + 10\% + 4.0\% = 59.0\%$,则总的识别正确率 $CR = CR_1 * (0.192 * Pr_c + 0.59 * Pr_d + (1 - 0.192 - 0.59))$。

将实验结果做成图,用来说明当学习样本比例增加时,这两种分类器构造方法之间的关于识别正确率、训练时间以及领域覆盖数的变化图分别如图 3、图 4 和图 5 所示。图 3 至图 5 中,虚线带“*”的部分描述是单层分类器的实验测试结果,实线带“o”的描述的是两层结构分类器的实验测试结果。

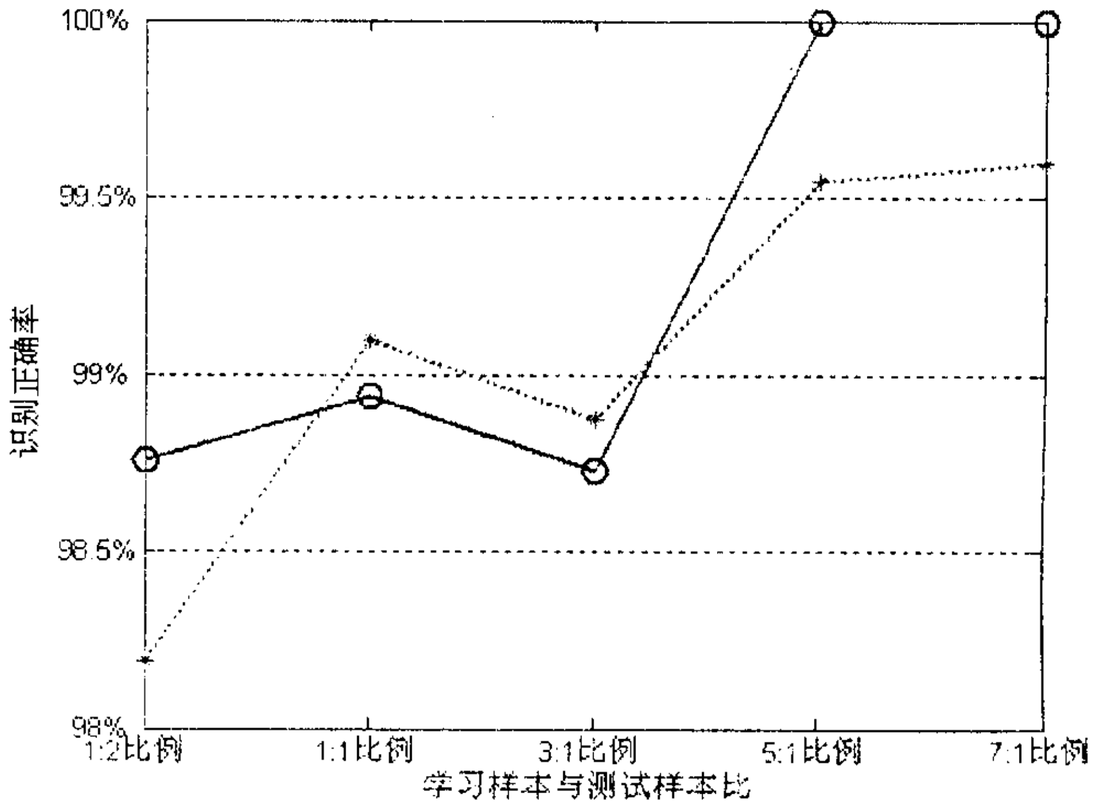

图 3　学习样本增加时的识别正确率变化图

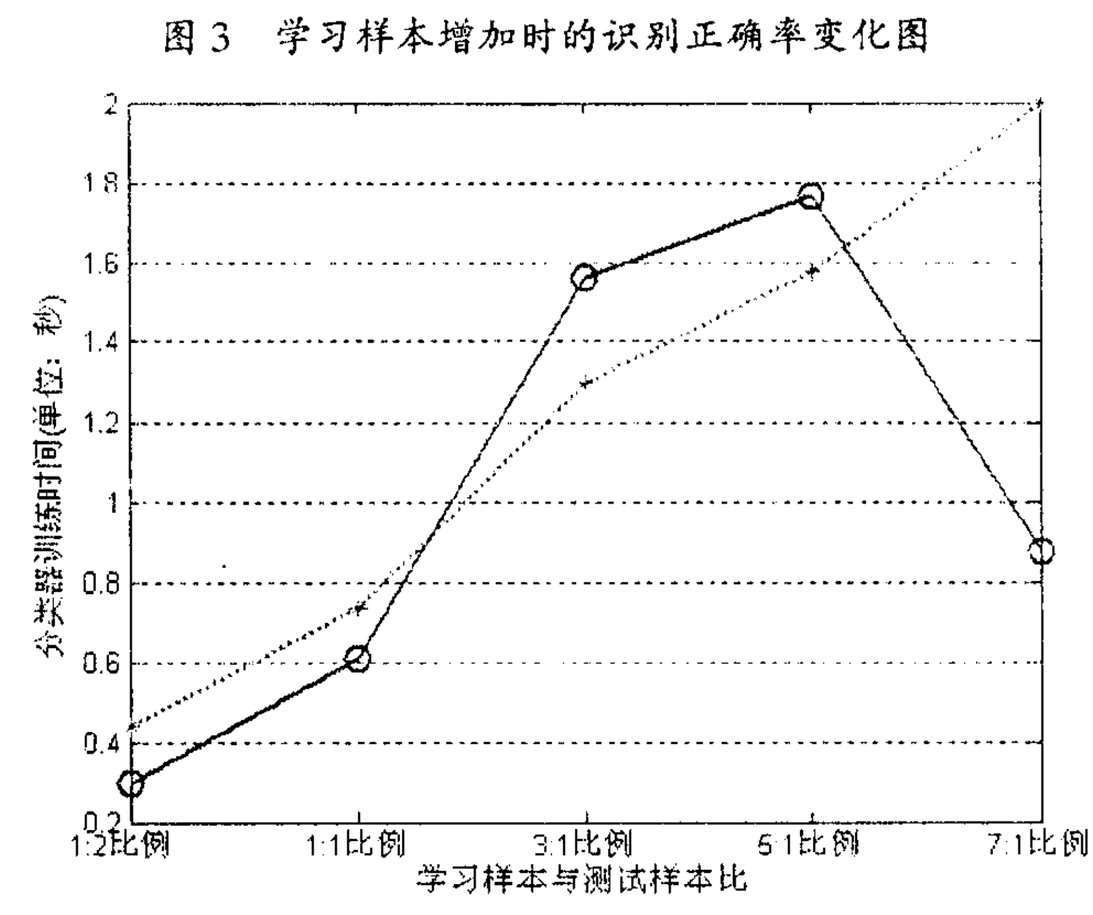

图 4　学习样本增加时的分类器训练时间变化图

由图 3 可知,当学习的样本数不断加大时,传统单层

分类器的识别很难大幅度提高,而文中提出的两层结构分类器设计可以很快达到满意的程度。

由图 4 可知,随着学习了更多的样本,单层分类器的领域训练时间一直是上升的趋势,而且上升都保持较高幅度;而两层结构分类器设计,当学习样本增长到一定程度后,反而出现了下降的趋势,在"5:1 比例"处,训练时间出现了下降,这是因为在两层结构分类器设计中,将和单层分类器一样的学习样本按照分层次的结构进行学习的结果。

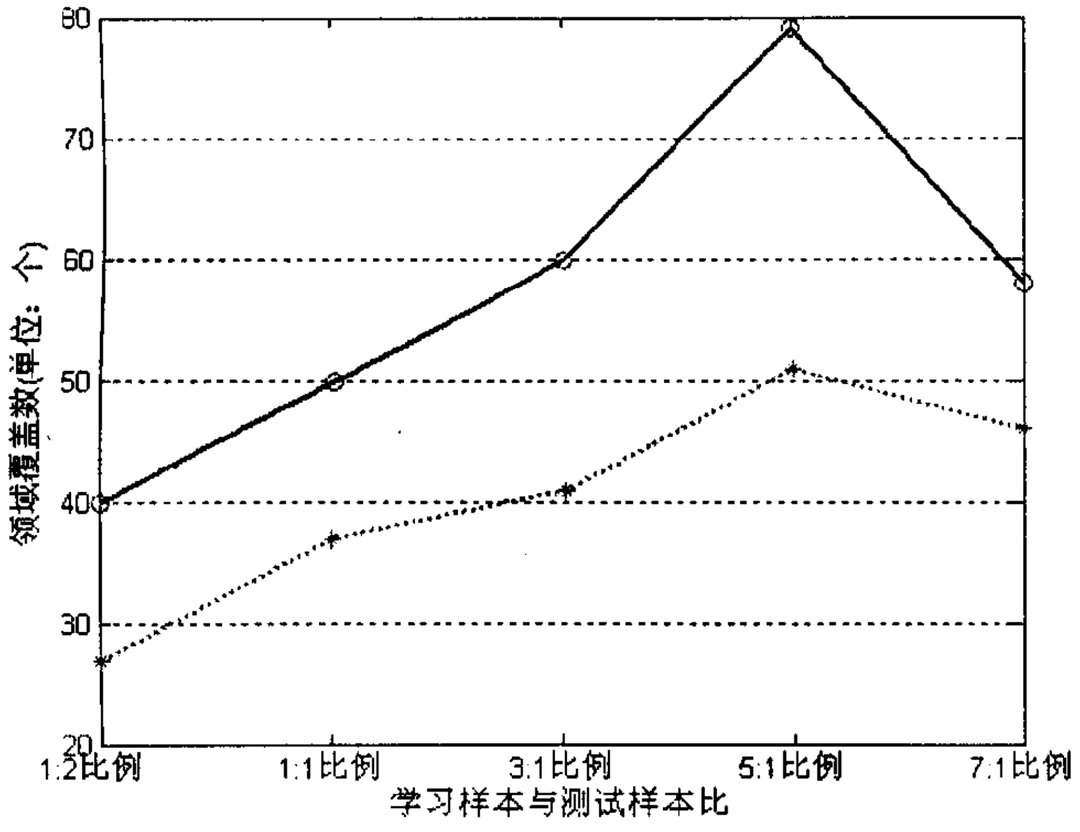

图 5 学习样本增加时的领域覆盖数变化图

由图 5 可知,当学习样本相对于测试样本比例不断增大时,两层结构分类器设计比单层分类器的领域覆盖数一直要多,即分类器设计的复杂度要高一些。但是动态增长的幅度基本上两种方法趋于一致。

4 结 论

综合以上分析可知,当学习样本与测试样本比例大到一定数值时,与单层的覆盖算法构造的分类器相比,两层结构分类器在不明显增加分类器设计复杂度的情况下,使分类器的性能主要是识别正确率和分类器的训练学习时间分别得到了较大的优化,特别是最重要的识别正确率大幅度地提高到了令人满意的程度。

参考文献:

[1] 边肇祺,张学工.模式识别[M].第 2 版.北京:清华大学出版社,1999:176-212.

[2] 沈淑娟,姜建国,曹建春.手写体字符识别的多特征多分类器设计[J].计算机工程与应用,2004(16):116-118.

[3] Zhang Ling, Zhang Bo. A Geometrical Representation of McCulloch-Pitts Neural Model and Its Applications[J]. IEEE Trans. on Neural Networks, 1999, 10(4):925-929.

[4] 张 铃,张 钹.多层反馈神经网络的 FP 学习和综合算法[J].软件学报,1994,8(4):252-258.

[5] 张 铃,张 钹.多层前向网络的交叉覆盖设计算法[J].软件学报,1999,10(7):737-742.

[6] 张 铃,张 钹.神经网络的规划学习算法[J].计算机学报,1994,17(9):669-675.

(上接第 64 页)

$f(e) \geqslant 0.90$。表 1 为各结点间线路规划代价。经过计算,可得到最小代价为 588,网络拓扑结构如图 2 所示。

表 1 线路规划代价

	1	2	3	4	5	6	7	8
1	0	39	57	123	54	81	69	87
2		0	21	96	81	117	93	78
3			0	102	102	138	102	90
4				0	126	177	192	54
5					0	51	108	75
6						0	99	126
7							0	162
8								0

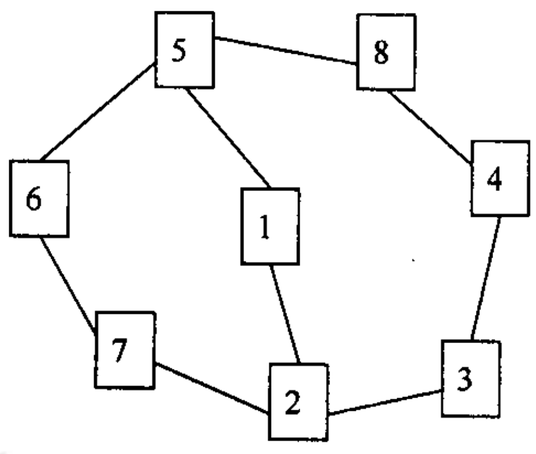

图 2 网络优化拓扑结构图

4 结 论

改进了的交叉算子和变异算子的自适应遗传算法能够根据适应度的大小自适应地选取交叉算子和变异算子,从而可以避免问题解的"早熟"或收敛速度过慢现象出现。通过一个简单网络优化问题的仿真,验证了这种策略具有较好的效果。

参考文献:

[1] Jan Ronghong, Huang Fungien, Cheng Shengtzong. Topological optimization of a communication network subject to a reliability constraint[J]. IEEE Trains Relia1, 1993, 42:63-70.

[2] Zhao Lianchang, Shao Fangming. Optimization of connecting two communication network subject to a reliability constraint [J]. Microelectron Reliab, 1997, 37(4):629-633.

[3] Kumar A, Pathak R A, Gupta Y P. Genetic-algorithm-based reliability optimization for computer network expansion [J]. IEEE Trans Relial, 1995, 44:63-72.

[4] 阎平凡,张长水.人工神经网络与模拟进化计算[M].北京:清华大学出版社,2000.

[5] Jan Ronghong. Design of reliable networks[J]. Computers Ops, 1993, 20(1):25-34.

[6] Srinivas M, Patnailk L M. Adaptive probabilities of crossover and mutations in GAs[J]. IEEE Trans on SMC, 1994, 24(4):656-666.