多层前向网络的交叉覆盖设计算法*

张 铃[1,3] 张 钹[2,3] 殷海风[1]

[1](安徽大学人工智能研究所 合肥 230039)

[2](清华大学计算机科学与技术系 北京 100084)

[3](清华大学智能技术与系统国家重点实验室 北京 100084)

E-mail: zling@mars.ahu.edu.cn

摘要 文章根据多层前向权、阈值神经网络的设计原则,将设计过程分成两步进行:先用尽可能少的领域将样本中的各类分隔开来,然后再用作者提出的交叉覆盖算法进行网络设计.文章给出两个很有代表性的模拟例子,一个是平面上两根螺线的分离问题,另一个是"无穷样本学习"的例子.模拟结果证明了此方法的有效性.

关键词 多层神经网络,设计原则,交叉覆盖算法.

中图法分类号 TP18

在文献[1]中,我们利用 M-P 神经元模型的几何意义,给出一种设计神经网络(作为分类器)的原则方法,称为 FP 覆盖算法.本文将按照文献[1]中给出的构造 FP 覆盖算法的原则,深入地进行讨论,给出一种常用的覆盖算法,暂且称之为"交叉覆盖算法".本算法在一定意义上解决了多年来一直未解决的(作为分类器)多层前向网络的设计问题.最后给出两个模拟的例子,并指出本算法可以用来解决"无穷样本的学习问题"(而无穷样本学习问题用其他学习算法是无法解决的),这充分说明本算法的有效性.为方便读者阅读,下面简单介绍文献[1]中所用到的一些内容(详细情况见文献[1]).

M-P 神经元模型的几何意义简介.

若我们限定输入向量的长度相等,即输入向量是限定在 $n+1$ 维空间的某个球面 S^n 上(其中心在原点,半径为 R),那么这时($W*x-\theta$)>0(其中 W 是权向量,θ 是阈值),就表示球面上落在由超平面 P(其方程为:($W*x-\theta$)=0)所分割的正半空间的部分,这个部分恰好是球面上的某个"球形领域".若取 W 与 x 等长,则这个"球形领域"的中心恰好是 W,其半径为 $r(\theta)$($r(\theta)=R(\cos^{-1}(\theta/R^2))$),如图 1 所示.

若我们令 $\sigma'(x)=\begin{cases}1, & \text{当 } x>0\\ 0, & \text{其他}\end{cases}$,且取神经元的激励函数为 $\sigma'(W*x-\theta)$,则一个神经元的激励函数正好是它所代表的球面上"球形领域"的特征函数,这样,我们就将神经元与球面上的球形领域对应起来.利用神经元的这种几何意义,我们就能非常直观地进行神经网络的各种研究.

由上面给出的神经元的几何意义得知,构造一个网络,使对给定的样本集能进行符合要求的分类,等价于求出一组领域,对给定样本集 K 中的点,能按分类的要求用领域覆盖将它们分隔开来.这样,我们就将神经网络的最优设计问题转化成某种求最优覆盖的问题.

当给定的输入向量的长度不相等时,可用下面给出的方法,将它变换成长度相等的情况.

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,作变换

---

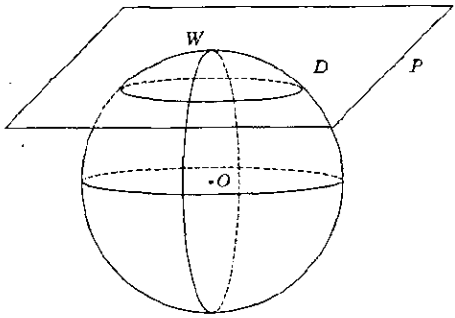
* 本文研究得到国家自然科学基金和国家 863 高科技项目基金资助.作者张铃,1937 年生,教授,主要研究领域为人工智能理论,人工神经网络理论.张钹,1935 年生,教授,博士生导师,中国科学院院士,主要研究领域为人工智能理论及应用,计算机应用技术.殷海风,1970 年生,硕士,主要研究领域为人工神经网络理论及应用.

本文通讯联系人:张铃,合肥 230039,安徽大学人工智能研究所

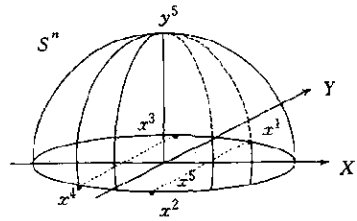
本文 1998-01-16 收到原稿,1998-08-24 收到修改稿

$$T:D\to S^n, x\in D, T(x)=\left(x,\sqrt{(d^2-|x|^2)}\right),$$

其中 $d\geqslant\max\{|x||x\in D\}$.



(a) 超平面 P 与超球面相交，形成"球形领域"的示意图



(b) 从 $D\to S^n$ 变换的示意图

图1

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. 如上面所述，这时每一个神经元 (W,θ) 就是在超球面 S^n 上，以 W 为中心，以 $r(\theta)$ 为半径的一个"球形领域"的特征函数(其中 $r(\theta)=r(\cos^{-1}(\theta/R^2))$).

1 多层前向神经网络的交叉覆盖设计算法

1.1 问题的提出

设给定一输入集 $K=\{x^1,x^2,\ldots,x^p\}$ (K 是 n 维欧氏空间的点集)，设 K 分为 s 个子集 $K^1=\{x^1,x^2,\ldots,x^{m(1)}\},\ldots,K^s=\{x^{m(s-1)+1},x^{m(s-1)+2},\ldots,x^p\}$. 现求作一个三层网络 N，满足：通过这个网络后，属于 K^i 的点的输出均为"y^i"，其中 $y^i=(0,\ldots,1,0,\ldots,0)$(即其第 i 个分量为 1，其余分量为 0 的向量). $i=1,2,\ldots,s$.

下面，为讨论方便，令 $s=2$，且令 $y^1=1, y^2=-1$.

1.2 交叉覆盖算法

由文献[1]知，用三层神经网络构造分类器，等价于求出一组领域，这组领域能将不同类的点分隔开来. 下面，我们给出一个称为"交叉覆盖法"的算法. 其主要思路是：先求一个领域 C^1，它只覆盖 K^1 中的点，而不覆盖 K^2 中的点，然后将被 C^1 覆盖的点删去. 对余下的点求另一领域 C^2，它只覆盖 K^2 的点，而不覆盖 K^1 的点，然后将被 C^2 覆盖的点删去，...，如此交叉进行覆盖，直到 K^1(或 K^2)的点全部被删除为止.

在上面求 C^i 时，当然希望它覆盖的点越多越好，因为这样就能用较少的领域完成覆盖的任务，也就是，说所得到的网络的元件个数就越少. 下面给出一个求 C^i 的方法(此法未必能求到最优的领域 C^i，但一般可得到较优的领域). 其求法的要点是：用"求重心+求领域"和"平移+求领域"交互进行，求得较好的领域 C^i.

算法 1. 求交叉覆盖的步骤

首先将 K^1,K^2 的点映射到球面 S^n 上(S^n 是 $n-1$ 维空间中，中心在原点，半径 $=R$ 的 n 维球面，取其半径 $R>\max|x^i|$)仍记为 K^1,K^2.

第 1 步：作一覆盖 $C(i)$(开始时 $i=1$)，它只覆盖住 K^1 的点，被 $C(i)$覆盖的 K^1 中的子集为 $K^{1'}$，

令 $K^2\leftarrow K^1/K^{1'}, K^1\leftarrow K^2$，若 K^1 或 K^2 为空集，停止. 否则，$i=i+1$，返回第 1 步.

由算法 1 求到一个覆盖集合，记为 $C=\{C_1,C_2,\ldots,C_p\}$.

网络设计：

第 1 元件层，取 p 个元件 $A^1,A^2,\ldots,A^p$，其中 A^i 为对应于 C^i 的神经元，$i=1,2,\ldots,p$. 其功能函数暂设为特征函数.

第 2 层取一个 p 输入的神经元 B，设其权阈值为 (u,a)，u_i 和 a 可由下面的方法求得(不妨设最后 K^1 为空

集，而 K^2 不为空集，以及 K^1 对应的输出为 1，K^2 对应的输出为 -1）：

$$\begin{aligned} & -a<0 \quad (\text{剩下的 } K^2 \text{ 对应的方程}), \\ u_p \quad & -a>0 \quad (\text{第 } p \text{ 个覆盖中各点满足的方程}), \\ u_{p-1}+u_p \quad & -a<0 \\ \text{或 } u_{p-1} \quad \ldots \quad & -a<0 \quad (\text{第 } p-1 \text{ 个覆盖中的各点满足的方程}), \\ u_1+b_2u_2+b_3u_3+\ldots+b_pu_p \quad & -a>0 \quad (\text{第 1 个覆盖中的各点满足的方程}), \end{aligned}$$

其中 b_i 取 1 或 0，由各点的具体情况而定. 不管 b_i 取何值，上式一定有解. 可取 $a=1, u_p=2, u_{p-1}=-2, u_{p-2}=4$，一般 $u_{p-2i}=2-\left(\sum_{t=1}^{i}u_p-(2t-1)\right)$，$u_{p-(2i+1)}=-\sum_{t=0}^{i}-u_{p-2t}$.

这样的网络就构成分类器，将 K 分为 K^1 和 K^2（即对应于 $K^1(K^2)$ 中的点，其输出 $=1(-1)$）.

注：按上述方法求到的输出层神经元的权系数，将随样本个数的增加，呈指数增加. 为避免这个问题，我们可以增加一个隐层，则可使输出层的权系数至多只按样本数呈线性增加.

算法 2. 求覆盖 $C(i)$ 的步骤

第 1 步：若 K^1 或 K^2 有一个是空集，则停止；否则（不妨设 $K^1\neq\varnothing$）任取 a_i（开始时，$j=1, i=1$）属于 K^1.

第 2 步：求以 a^i 为中心的领域 $C(a^i)$. 令 $C(a^i)\cap K^1=D_i, i=1,2,\ldots, D_0=\varnothing$，

[$C(a^i)$ 对应的权和阈值 $(W^i=(w_j^i), \theta=(\theta_i))$，可按下面的公式求得：

令

$$d^1(i)=\max_{X\in K^1}\{\langle a^i,x\rangle\},$$

$$d^2(i)=\min_{x\in K^1}\{\langle a^i,x\rangle \mid \langle a^i,x\rangle>d^1(i)\},$$

$$d(i)=\frac{d^2(i)+d^1(i)}{2},$$

$$\theta_i=d(i),$$

$$W=(a^i), \theta=(\theta_i)].$$

第 3 步（求重心）：若 D_{i-1} 是 D_i 的真子集，则求 D_i 重心 a^-，令 $a^{i+1}\leftarrow a^-$，$i\leftarrow i+1$，返回第 2 步. 否则，进入第 4 步.

第 4 步（平移）：求 a^i 的平移点 a'，令 $a^{i+1}=a'$（a' 的求法见算法 3），并求对应的领域 $C(a^{i+1})$（求领域 C 时，需将 a^{i-1} 投射到球面上），得 D_{i+1}.

若 D_i 是 D_{i+1} 的真子集，则求 D_{i+1} 重心 a^-，令 $a^{i+1}\leftarrow a^-$，$i\leftarrow i+1$，返回第 2 步.

否则，令 $C_j=C(a^i)$，这样我们就求到一个覆盖 C_j.

然后，将被 C_j 覆盖的点删去，即令 $K^{1j}=C_j\cap K^1, K^2\leftarrow K^1/K^{1j}, K^1\leftarrow K^2, j\leftarrow j+1$，回到算法 1 的第 1 步，求另一个覆盖. 最后得到一组领域 $\{C_1, C_2, \ldots, C_p\}$.

算法 3. 求平移算法（求 a 的平移点 a'）

设 $a\in K^1, B=\min_{x\in K^1}\{x \mid d(a,x)\}$，其中 $d(a,x)$ 表示 a 与 x 的距离.

第 1 步：若 $|B|=k>n$，则取 $a'=a$. 若成功，则停止.

否则，求 a 到 $P(B)$（其中 $P(B)$ 是由 B 构成的线性流型）的垂足 b，令 $P(k)=P(B)$，再对每个 $x\in K^2/P(k)$，求 $d(x)$：

$$d(x)=\frac{\langle a,c-x\rangle}{\langle b,c-x\rangle},$$

其中 c 是 $P(k)$ 中的任一点.

若存在 x：$\langle a,c-x\rangle=0$，则令 $x=c_{k+1}$，取 $a'=a$，令 $P(k+1)=P(k)\cup\{c_{k+1}\}$. 进入第 2 步.

否则，令

$$d=d(x^*)=\min_{x\in K^1}\{d(x)\}, \tag{1}$$

令 $$a'=\frac{R(a-db)}{|a-db|},$$

其中 R 是球面 S^n 的半径. 即将 $(a-db)$ 的向量投影到 S^n 球面上. 再取 $c_{k+1}=x^*$.

第 2 步:令 $P(k-1)=P(k)\cup\{c_{k+1}\}$(这里,我们为简单起见,用 $P(k)\cup\{c_{k+1}\}$ 表示由 $P(k)$ 和 c_{k+1} 构成的线性流型).

若 $k-1>n$,则 a' 为所求. 若成功,则停止.

否则,求 a' 到 $P(k+1)$(开始时,$k=|B|$)的投影 b_{k+1},令 $b=b_{k+1}$,$a=a'$,$k\leftarrow k+1$,返回第 1 步.

求投影算法.

设 $P(k)=\{c_1,c_2,\ldots,c_k\}$($P(k)$是算法 3 中所定义的 $P(k)$),a 不属于由 $P(k)$构成的流型(仍记为 $P(k)$),求 a 到 $P(k)$的投影.

我们可用先求归一化的正交基,然后再用求投影的方法来求投影.

求 $P(k)$的正交归一化基的方法为

$$P(k)=\{c_1,c_2,\ldots,c_k\},d_i=c_i-c_1,\ i=2,\ldots,k,$$

令 $$e_2=\frac{d_2}{|d_2|},e_3=\frac{[d_3-(d_3,e_2)e_2]}{|[d_3-(d_3,e_2)e_2]|},\ldots,e_k=\Big[d_k-\sum_i^{k-1}(d_k,e_i)e_i\Big]\Big/\Big|\Big[d_k-\sum_i^{k-1}(d_k,e_i)e_i\Big]\Big|,$$

于是,求 a 到 $P(k)$的投影为

$$a'=b+\sum_i^k(a-b,e_i)e_i.$$

这样,当 $P(k-1)$的正交基已求得,再求 $P(k)$的正交归一化基时,只要再求 e_k,连同 $e_1,\ldots,e_{k-1}$,就构成 $P(k)$的正交归一化的基.

下面我们要证明的是,按算法 3,将 a 平移后得到 a' 点,若以 a' 为中心,作只覆盖 K^1 点的领域,则此领域的半径比以 a 为中心,只覆盖 K^1 点的领域的半径大. 这样就证明了,通过平移,有可能求到覆盖更多 K^1 点的领域.

命题 1. 设 a,$P(k)$是算法 3 中对应的符号,则有 a 到 $P(k)$中各点的距离相等.

证明:当 $k\leqslant|B|$时,由 B 的定义知,命题结论成立,现用归纳法加以证明.

若存在 x:$(a,c-x)=0$,由 $(a,c)=(a,x)$,直接得 $d(a,c)=d(a,x)$;若求到 x^*,取 $c_{k+1}=x^*$,由 $d(x^*)=\frac{(a,c-x^*)}{(b,c-x^*)}$,得 $(a-d(x^*)b,c)=(a-d(x^*)b,x^*)$,故有 $d(a',c)>=d(a',x^*)=d(a',c_{k+1})$. □

命题 2. 设 a' 是 a 按算法 3 求到的平移点,则 $C(a)\subset C(a')$,且 $C(a')$只覆盖 K^1 的点.

证明:作 $C(a')(C(a))$是以 $a'(a)$为中心,以 $a'(a)$到 c(c 是 $P(k)$中的任一点)的距离为半径的领域,于是 $C(a')(C(a))$的方程为 $(a',x-c)>0((a,x-c)>0)$. 下面证明,若 x 满足 $(a,x-c)>0$,则 x 一定也满足 $(a',x-c)>0$.

由于当 x 落在 $C(a)$内时,有 $d(x)=\frac{(a,c-x)}{(b,c-x)}<0$,于是 $(a',x-c)=(a-d(x^*)b,x-c)=(a,c-x)-d(x^*)(b,c-x)$,由 $(a,c-x)>0$ 及 $d(x)<0$ 得,$(b,c-x)<0$,故有 $(a',c-x)>0$,即 x 也属于 $C(a')$,得 $C(a)\subset C(a')$.

另外,由 $d(x^*)$的定义易得,$C(a')(C(a))$只覆盖 K^1 的点. □

由命题 1、2 知,按算法 3 求平移点所得到的对应的领域,覆盖 K^1 点的范围将会越来越大,这正是我们所要达到的目的.

注:我们这里给出的只是求覆盖领域的方法之一. 此方法未必能保证每次求到的领域 $C(a)$覆盖 K^1 的点达到最多,故算法还有改进的余地.

2 模拟结果

本节给出两个很有代表性的例子,由这些例子就可看出本文所给出的方法的潜力.

例 1:平面上双螺旋线的识别问题(如图 2 所示).

设 K^1 是在极坐标系中曲线 $r=\theta$ 上的子集,K^2 是在极坐标系中曲线 $r=-\theta$ 上的子集,$\pi/2\leqslant\theta\leqslant6\pi$.

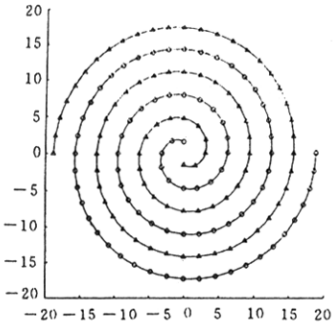
$K^1(K^2)$上各有 77 点,其 θ 值为 $\theta=i\pi/2$,$i=1,\ldots,12$,以及 $\pi/2$ 到 π,取二等分点,π 到 $3\pi/2$ 取二等分点,一

般 $i\pi/2$ 到 $(i+1)\pi/2$ 区间取 $i+1$ 等分点.

求一个三层前向神经网络，将 K^1，K^2 分开. 这个例子是神经网络学习中有名的难学习的问题之一.

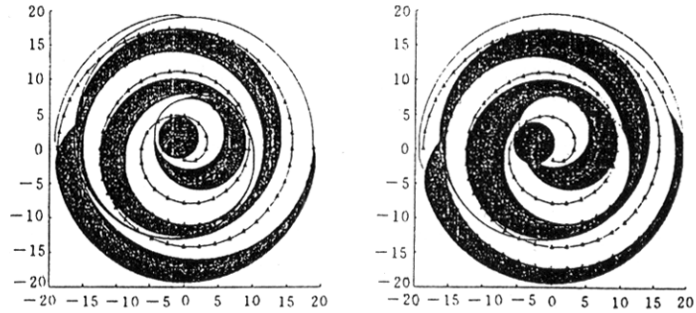
在文献[2]中，Raum 等人用 BP 算法求解双螺旋线问题，结果失败了. 在文献[3]中，Chen 等人提出“生成-收缩”法，来求解双螺旋线学习问题，经过 3 000 次的迭代，得到一个解，此解识别的正确率只有 89.6%. 在文献[4]中，Fahlman 等人用“级联式”网络结构才勉强解决了双螺旋线学习问题，所得的网络要用几十个神经元. 可见此问题的难度. 下面我们用交叉覆盖算法求解此问题，得到了非常令人满意的结果.

解：用交叉覆盖算法求解此问题，得网络第 1 层共有 10 个神经元(如图 3 所示，其中椭圆是各领域在二维平面上的投影). 第 2 层仅一个神经元，整个网络共用 11 个神经元. 网络对样本的分类正确率为 100%.



曲线为双螺旋线，
○及△表示两类训练样本点

图 2　平面双螺旋分类问题



(a) 训练数据为双螺旋上 156 个样本点　　(b) 训练数据为 20 000 个数据点

图中黑色表示网络所划分的第 1 类样本的覆盖区域，
白色表示第 2 类样本的覆盖区域，○及△表示两类训练样本点

图 3　采用逐次覆盖法对平面双螺旋分类问题的求解结果

然后我们在曲线 $r=\theta(r=-\theta)$ 上随机各取 1 万个点作为测试样本，输入所得到的网络进行识别，正确率达到 98.245%. 另外，我们又随机地在 K^1，K^2 上各取 1 万个点作为样本点，然后进行学习，仍得到 10 个神经元(其在二维上的投影如图 3(b)所示)，这时不但其分类的正确率为 100%，而且我们在两类中各随机取 1 万个点进行识别，其识别正确率也达到 99.995%.

另外，我们还进行了平面三螺旋线和三维空间三螺旋线的学习问题，都得到了非常令人满意的结果，见表 1.

表 1　用交叉覆盖算法学习的情况表

问题	训练样本(样本数)	学习时间	覆盖数	测试样本(样本数)	识别率(%)
平面双螺旋线	1　(156)	<1ms	10	(20 000)	98.245
	2(20 000)	5.16s	10	(20 000)	99.995
平面三螺旋线	1　(234)	0.06s	24	(30 000)	91.11
	2(30 000)	14.61s	26	(30 000)	99.067
空间三螺旋线	1　(183)	0.07s	28	(30 000)	96.293
	2(30 000)	19.34s	37	(30 000)	99.653

例 2：求三维空间上两根螺旋线的识别问题.

设在柱坐标系中，$K^1=\{(\theta,r,z)\mid z=\theta,r=1,0\leqslant\theta\leqslant4\pi\}$，$K^2=\{(\theta,r,z)\mid z=\theta+\pi,r=1,0\leqslant\theta\leqslant4\pi\}$，求一个三层前向神经网络将 K^1，K^2 分开.

注：例 2 是一个无穷样本的识别问题，用其他方法(如 BP 算法)是无法解决的. 用本文提供的方法，可以比较圆满地解决这个学习问题.

解：用交叉覆盖设计法(要求覆盖住 K^1，K^2 曲线上的所有点)求得的网络，第 1 层 10 个元件，第 2 层 1 个元件.

我们随机地在曲线 K^1 和 K^2 上共取 30 000 个点，当作测试样本输入所得的网络进行识别，结果全部识别正确.

上面的两个例子可以充分说明本文所给的设计方法的有效性. 此两例若用其他算法(如各种改进过的 BP 算法)进行设计将是相当困难的.

3 结束语

本文利用 M-P 神经元模型的几何意义得出一个领域覆盖的设计算法. 这个算法在一定意义上考虑到网络的结构优化问题(指网络的规模最小)，并且方法切实可行，从而解决了多年来一直未能很好地得到解决的多层前向网络的设计问题. 本文给出的几个模拟例子也充分说明了本方法的潜力.

另外，我们这里给出的覆盖设计算法只是领域覆盖设计算法中的一个特例，任何一个求最小(次小)覆盖算法与本文提出的设计原则相结合，都可得出一个新的设计算法. 这就给研究网络设计问题开辟了一个新途径.

致谢 清华大学的李凌同学为本文的表 1 提供了模拟例子，在此表示感谢.

参考文献

1 张铃，张钹. M-P 神经元模型的几何意义及其应用. 软件学报，1998，9(5)：334～338
(Zhang Ling, Zhang Bo. A geometrical representation of M-P neural model and its applications. Journal of Software, 1998, 9(5): 334～338)

2 Baum E B, Lang K J. Constracting hidden units using examples and queries. In: Lippman R P *et al* eds. Neural Information Processing. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1991. 904～910

3 Chen Q C *et al*. Generating-shrinking algorithm for learning arbitrary classification. Neural Networks, 1994, 5(7): 1477～1489

4 Fahlman S E, Lebiere C. The cascade-correlation learning archtecture. In: Tourdtzhy D S ed. Advances in Neural Information-processing System. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1990. 524～532

An Alternative Covering Design Algorithm of Multi-layer Neural Networks

ZHANG Ling[1,3] ZHANG Bo[2,3] YIN Hai-feng[1]

[1](*Institute of Artificial Intelligence Anhui University Hefei* 230039)
[2](*Department of Computer Science and Technology Tsinghua University Beijing* 100084)
[3](*Laboratory of Intelligent Technology and Systems Tsinghua University Beijing* 100084)

Abstract In this paper, the authors present a new principle for designing a sort of multi-layer weight-sum- and -threshold neural networks. The design process consists of two steps. First, each class of the given training samples is covered by using neighbor coverings as less as possible. Then a neural network is specifically designed by an approach called alternative covering algorithm. The simulation results of two representative hard classification problems are shown. One is so called "two spirals separation" problem, the other is "the learning problem of infinitetraining samples". These simulation results are given to illustrate the effectiveness of the new method.

Key words Multi-layer neural network, principle for design, alternative covering algorithm.