文章编号：1000-8152(2004)03-0655-04

采样算子调整的径向基网络增量映射学习算法

游培寒，毕笃彦，王振家
（西安空军工程大学 工程学院，陕西 西安 710038）

摘要：为了提高增量映射学习(IPL)算法的效率，调整了径向基神经网络基函数的中心及方差，以达到调整采样算子的目的，同时，通过神经元函数相关性的计算，确定添加新神经元时，相关函数的阈值，为系统结构调整提供相应依据.新方法步骤相对简单，所以算法速度较快；仿真结果表明，由于系统参数得到调整，对于同一问题，改进IPL算法得到的径向基神经网络结构较一般算法得到的网络结构简单，输出结果也较为精确.

关键词：增量映射学习(IPL)算法；径向基(RBF)神经网络；三相训练法
中图分类号：TP183　　**文献标识码**：A

Incremental projection learning algorithm for constructing radial basis function network based on the adjustment of the sample operator

YOU Pei-han, BI Du-yan, WANG Zhen-jia
(Institute of Engineering, Air Force Engineering University, Shaanxi Xi'an 710038, China)

Abstract: Firstly, we adjusted the parameters of the radial basis function neural network, including the center and the covariance of the base function, and readoped the sampling operator of the algorithm; Then we raised a threshold to decide whether the radial basis function neural network needs a new neuron to reduce the output error of the system or not, and we could get it through calculating the correlation of the system's hidden neurons' base functions and the correlation between the neuron's base functions and stock functions. Through iteration of training, adjustment and selection, this method could adjust the system structure logically. Because of the simplicity of the new method, the improved incremental projection learning (IPL) algorithm's calculation speed was faster than before. The simulation results showed that the new algorithm can induce a simpler network structure than the former algorithm, and the output of the new IPL inducing network is more accurate than before.

Key words: incremental projection learning (IPL) algorithm; radial basis function (RBF) network; three-phase-training algorithm

1 前言(Introduction)

上世纪90年代末，日本学者M. Sugiyama, H. Ogawa提出了神经网络IPL算法，该算法根据网络的数学模型提出网络规模扩大因子 a，为网络在不同应用环境下的规模确定提供了依据. M. Sugiyama, H. Ogawa近期发表的论文就IPL算法的一些性质进行了讨论[1,2]，并提出了一些相关的递推算法；但是这些算法只适用于采样算子 A 固定的IPL网络. 而事实上，采样算子 A 的适当调整有利于网络逼近能力的大幅度提高.

本文首先介绍IPL算法的基本原理；再介绍IPL算法结合三相训练法构造径向基网络的方法；最后通过仿真，证明新方法的有效性.

2 IPL算法的基本原理(Basic theory of IPL algorithm)

1) 神经网络学习训练过程的数学模型. RBF网络的学习训练过程通常可以通过下式表示：

$$f_m = X_m A_m f + X_m n^{(m)}. \tag{1}$$

其中 $A_m = \sum_{i=1}^{m} e_i^{(m)} \otimes \bar{\phi}_i$ 是由 m 个隐层神经元的网络对于希尔伯特空间中的任意函数 f 的采样算子；"$\otimes$"代表Neumann-Schatten算子，如果 ϕ_i 离散，C^m 空间(单式空间)的标准基 $e_i^{(m)} = [0_1, 0_2, \cdots, 1_i, \cdots, 0_m]$，$\phi_i$ 被称之为采样基函数，也就是RBF网络的基函数，$\bar{\phi}_i$ 是其共轭. $n^{(m)} = (n_1, n_2, \cdots, n_m)$ 是采样误差向量，X_m 是映射算子. 该公式可以这样理解：神经网络通过采样算子 A_m 将样本向量映射到希尔伯特空间的基上，再通过 X_m 算子将映射结果投射到

收稿日期：2002-05-16；收修改稿日期：2003-06-23.

样本子空间.设网络的隐层神经元数为 $m-1$,忽略采样误差的影响,则采样过程可表示为

$$A_m f = \begin{cases} A_{m-1}, & \text{如果第 } m \text{ 个激励函数冗余}, \\ A_m, & \text{如果第 } m \text{ 个激励函数不冗余}. \end{cases}$$

当然,网络训练过程中采样误差是不可避免的,训练网络的目的就是使函数 $J[X_m] = E_m \| X_m n^{(m)} \|$ 达到最小,也就是网络的输出与 f 误差达到最小. $E_m(s)$ 代表求 m 维向量 s 的均值;X_m 算子也被称为学习算子[1,2].就样本空间离散且样本数量有限的网络而言,

$$\phi_i = [\Psi_{i1}, \Psi_{i2}, \cdots, \Psi_{is}],$$

$$A_m = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \cdots & \Psi_{1s} \\ \Psi_{21} & \Psi_{22} & \cdots & \Psi_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ \Psi_{m1} & \Psi_{m2} & \cdots & \Psi_{ms} \end{bmatrix}.$$

s 是样本数,m 是网络隐层神经元的个数.

2) X_m 及权值向量的求法. X_m 是通过逆运算得到的,设采样误差的协方差矩阵是 $Q_m = E_m(n^{(m)} \otimes \bar{n}^{(m)})$. 就样本空间离散且样本数量有限的网络设

$$N^{(m)} = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1s} \\ n_{21} & n_{22} & \cdots & n_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ n_{m1} & n_{m2} & \cdots & n_{ms} \end{bmatrix},$$

则有

$$Q_m = \frac{1}{s} N^{(m)} N^{(m)*}.$$

设 C^m 空间内的映射算子 $U_m = A_m A_m^* + Q_m$,设希尔伯特空间内的映射算子 $V_m = A_m^* \mathrm{pinv}(U) A_n$, $\mathrm{pinv}(s)$ 代表 s 的 Moore-Penrose 伪逆算子, A_m^* 是 A_m 算子的伴随算子;这样可以得到算子 X_m 的表达式

$$X_m = \mathrm{pinv}(V_m) A_m^* \mathrm{pinv}(U_m) + Y_m(I_m - U_m \mathrm{pinv}(U_m)). \tag{2}$$

其中: I_m 是 C^m 空间的单位向量, Y_m 是从 C^m 空间到希尔伯特空间的任意映射算子.对于径向基神经网络来说,如果采样误差足够小,式(2)中第二项的影响可以忽略.式(2)证明详见文献[1].本文通过对式(1)结果的推导得到一般径向基神经网络的输出权值向量的表达式

$$\vec{\omega} = f_m \mathrm{pinv}(A_m) = X_m A_m f \mathrm{pinv}(A_m),$$

即

$$\vec{\omega} = [f_m(x_1)\ f_m(x_2)\ \cdots\ f_m(x_s)] \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1m} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{s1} & \varphi_{s2} & \cdots & \varphi_{sm} \end{bmatrix}, \tag{3}$$

$$\vec{\omega} = (\omega_1, \omega, \cdots, \omega_m),$$

$$\mathrm{pinv}(A_m) = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1m} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{s1} & \varphi_{s2} & \cdots & \varphi_{sm} \end{bmatrix}.$$

3) a_{m+1} 因子的用法.当网络输入增加训练样本时,IPL 算法先判断样本与新的采样函数的相关性: $\phi_{m+1}(x_{m+1}) = \langle \phi_{m+1}, f(x^{m+1}) \rangle$. $\langle \cdot, \cdot \rangle$ 代表内积操作,它可以被认为是第 $m+1$ 个采样函数对第 $m+1$ 批样本的采样结果,这里 $x_{\lambda+1} \supset x_m$,也就是说,训练样本的数量是递增的.同样可以得到原有采样结果对于新的采样函数的相关因子: $\langle t_{m+1}, s_{m+1} \rangle$,其中 s_{m+1} 是原有的采样基对于新的采样函数的采样结果:

$$s_{m+1} = A_m \phi_{m+1} + q_{m+1},\ q_{m+1} = E_m(n^{(m)} \bar{n}_{m+1}).$$

$\bar{n}_{m+1}$ 是第 $m+1$ 个神经元的采样误差的转置, t_{m+1} 可以看成是 $m+1$ 个神经元的输出权值向量, $t_{m+1} = \mathrm{pinv}(U_m) s_{m+1}$, 则

$$a_{m+1} = \phi_{m+1}(x_{m+1}) - \langle t_{m+1}), s_{m+1} \rangle. \tag{4}$$

这里引用文献[1]中的定理:

当 $a_{m+1} = \Psi_{m+1}(x_{m+1}) - \langle t_{m+1}, s_{m+1} \rangle = 0$时,说明 $f_{m+1} = f_m$.

具体的证明请见文献[1],该定理可以简单的解释为:当新的采样函数与样本函数的相关性和其与原有采样结果的相关性相同时,新的采样函数就不必存在.以上定理为隐层神经元的构造提供了依据.

4) IPL 算法的缺点. IPL 算法提高网络效率,只能不断扩大 A_m 算子的规模.通过文献[3]可知,当网络神经元数量超过一定限度,网络的逼近能力非但不继续增加,反而会减小.

3 一种新的调节参数的 IPL 算法(New parameter adjustable IPL algorthm)

结合径向基网络三相算法[4],提出了一种采样算子调整的 IPL 算法,新算法具体如下:

1) 建立参数统一的采样函数库,对网络输入第一批样本 $\bar{x}$,通过计算 $a_1^i = \phi_i(\bar{x}_1)$ 找出最大 a_1^i 所对应的第 i 个采样函数,作为网络第一个神经元的基函数, $m-1$.

2) $m = m+1$,利用式(4)计算 a_m^i,找出最大 a_m^i 所对应的神经元,计算 θ_a 作为阈值当 $\max(a_m^i) > \theta_a$ 时,添加该 a_m^i 所对应的采样函数为新神经元的基函数.否则,撤消输入,重新抽取随机样本组 $\bar{x}_m$, 返回2).

3) 按如下公式训练网络参数:

$$\Delta c_{ji} = \eta \sum_p (f - O_{m'}) \omega_j h_j \frac{(x_{m'p} - c_{ji})}{\sigma_{ji}^2},$$
$$\Delta \sigma_{ji} = \eta \sum_p (f - O_{m'}) \omega_j h_j \frac{(x_{m'p} - c_{ji})}{\sigma_{ji}^3}. \tag{5}$$

其中: η 代表的是步长, x_{mp} 代表 m 批训练样本中的第 p 个样本; $H_{m'} = [h_1, h_2, \cdots, h_{m'}]$ 代表网络神经元的输出向量, m' 是网络的隐层神经元数, 一般情况下 $m' \neq m$; $O_{m'}$ 是网络的输出, c_{ji} 代表第 j 个神经元的第 i 维均值, σ_{ji} 代表第 j 个神经元的方差矩阵对角线上第 i 个元素.

4) 计算网络的输出均方总误差 err, 当 $err < Err$ 时或当 $m > M$ 时, 训练结束, Err 是要求的误差上限, M 是网络的最大迭代次数; 否则, 返回 2).

在仿真中, 为了减少网络神经元之间的冗余性, 一般取

$$\theta_a = \begin{cases} \phi_{m+1}(x_{m+1}) - \sum_{i=1}^{m+1} t_i D(\phi_{m+1}), \\ \qquad \text{if } \phi_{m+1}(x_{m+1}) - \sum_{i=1}^{m+1} t_i D(\phi_{m+1}) > 0, \\ 0, \qquad \text{if } \phi_{m+1}(x_{m+1}) - \sum_{i=1}^{m+1} t_i D(\phi_{m+1}) \leqslant 0. \end{cases} \tag{6}$$

$D(\phi_{m+1})$ 是 ϕ_{m+1} 的自相关函数. 仿真中, M(网络训练的最大迭代次数) 一般取 $M = \xi\gamma$, γ 是网络所要使用的最大的神经元数量, 也就是采样函数库中函数的数量. ξ 是比例系数, 一般 $1 \leqslant \xi \leqslant 2$.

4 仿真结果(Simulation results)

1) 本文用新算法与 IPL 算法同时逼近一个非线性函数

$$f = 2x - 14e^{-3(x-2.5)^2} - 5e^{-6(x-0.5)^2} + 3e^{-3x^2} + 12e^{-(x+2.5)^2}.$$

由于样本以离散形式给出, 式(4)中 t_{m+1}, s_{m+1} 均以离散形式得到, 也就是分别将样本输入新基函数及原有基函数以得到离散形式的 t_{m+1}, s_{m+1}, 再通过向量内积得到 a_m^i 因子. 新算法与 IPL 算法都迭代 30 次, 每一批训练数据增加一个随机选取的样本, 每增加一个样本网络通过梯度下降法训练 25 次. 通过式(6)计算 $\theta_a = 4.33$, 采样函数的初始标准差为: $\sigma = 0.5$, 初始均值为: $c_j = -\pi + 0.3j, j = 1, 2, \cdots, 25$. 三段法的训练步长为 $\eta = 0.01$. 利用 IPL 算法得到均方误差为: $err_{\text{IPL}} = 0.1607$; 利用本文提出的新算法得到的网络输出的均方误差为: $err_{\text{NEW}} = 0.0711$. 显然, 新算法在简化网络结构的基础上提高了网络的逼近精度. 图 1 是利用 IPL 算法得到的网络训练结果, 图 2 是利用新算法得到的网络训练结果.

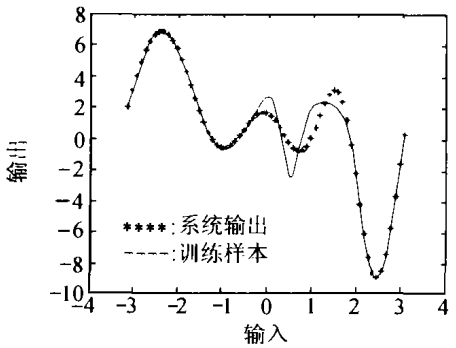

图 1(a)　一般 IPL 系统输入输出关系
Fig. 1(a)　Input-output relation of IPL

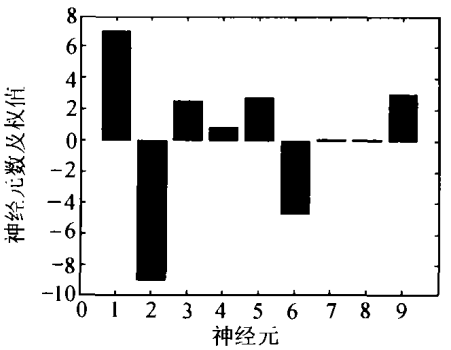

图 1(b)　一般 IPL 系统权值
Fig. 1(b)　IPL system's weights

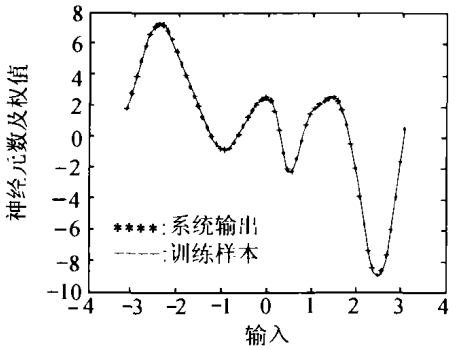

图 2(a)　新算法得到的网络输出
Fig. 2(a)　Output of the new algorithm

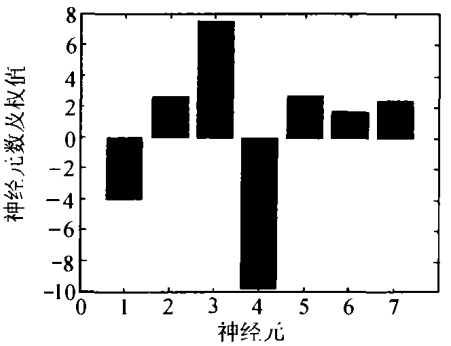

图 2(b)　新算法得到的网络神经元数及输出
Fig. 2(b)　New model's network weights

2) 利用新算法滤除系统噪声.随意选择一个系统函数如图3(d),加入的白噪声,得到网络的训练样本.代表原函数的幅值.

首先建立采样函数库.函数库的初始均方矩阵按如下方法取值: $\sigma_{j1} = \sigma_{j2} = 0.45$. 文中取50个高斯型的采样函数,其均值向量均匀分布于输入空间.

$$\phi_j(\vec{x}) = \exp((\vec{x} - \vec{c}_j)R_j^{-1}(\vec{x} - \vec{c}_j)),$$

$$R_j = \begin{bmatrix} \sigma_{j1} & 0 \\ 0 & \sigma_{j2} \end{bmatrix}, \ \vec{c}_j = [c_{j1}, c_{j2}].$$

R_j 是第 j 个神经元激励函数的协方差矩阵,$\vec{c}_j$ 是均值向量.

选择训练参数进行学习训练:迭代次数选取50次,每迭代一次增加5个随机选取的满足高斯概率分布的随机样本.每增加一个神经元,网络按梯度下降法迭代训练网络20次.

得到如下结果:新算法使用了21个神经元,得到的网络输出相对于Peak函数的均方误差为: $err_{NEW} = 0.073$. 本文还利用原IPL算法进行了同样的运算,样本选取方法相同;经过50次迭代计算,得到的径向基网络使用23个隐层神经元.网络输出相对于原Peak函数的均方误差为 $err_{IPL} = 0.242$. 具体训练结果见图3.

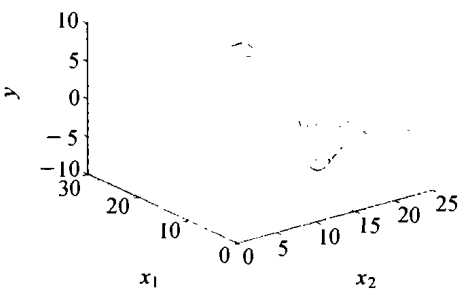

图3(a) 新算法的输出结果
Fig. 3(a) Trianing result of the new algorithm

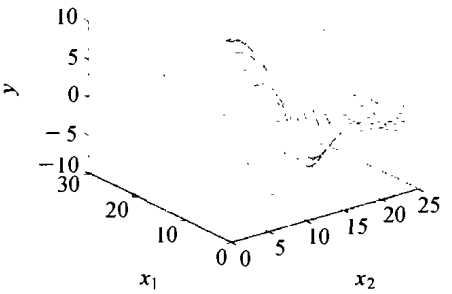

图3(b) Matlab的原函数
Fig. 3(b) Teacher value of Matlab

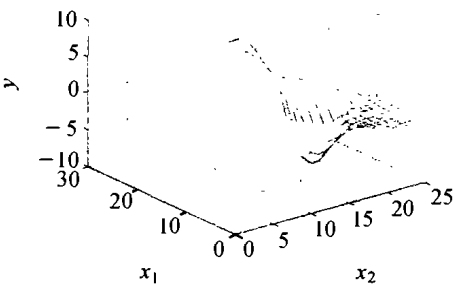

图3(c) 一般IPL算法得到的系统输出
Fig. 3(c) Training result of former IPL algorithm

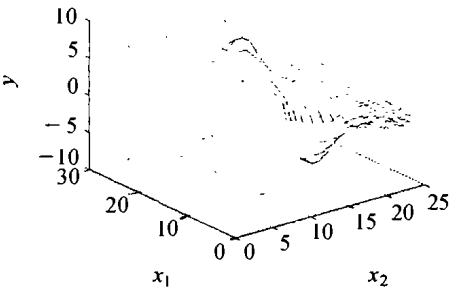

图3(d) 加入噪声的训练样本
Fig. 3(d) Samples with noise

5 结论(Conclusion)

本文将IPL算法与基于梯度的三段算法相结合提出了新的参数调节的IPL算法;它简化了网络结构,提高了网络的逼近能力;仿真结果证明,新算法在调节网络参数的同时,还控制了网络的规模,是构造RBF神经网络的有效方法.

参考文献(References):

[1] SUGIYAMA M, OGAWA H. Incremental projection learning for optional generalization [J]. *Neural Networks*, 2001, (14): 415 - 422.

[2] SUGIYAMA M, OGAWA H. Properties of incremental projection learning [J]. *Neural Networks*, 2001, (14): 423 - 432.

[3] LEUNG H, LO T, WANG Sichun. Prediction of noisy chaotic time series using an optimal radial basis function neural network [J]. *IEEE Trans on Computers*, 1999, (8): 243 - 256.

[4] MARINARO M, SCARPETTA S. On-line learning in RBF neural networks: a stochastic approach [J]. *Neural Networks*, 2000, (13): 87 - 90.

[5] SCHWENKER F, HANS A, KESTLER Gunther Palm. Three learning phases for radial-basis-function networks [J]. *Neural Networks*, 2001, (14): 439 - 458.

作者简介:

游培寒 (1978 —),男,空军工程大学工程学院四系信息与信号系统专业博士研究生,主要研究方向是数字信号处理及人工神经网络,E-mail: davidyou025@163.com;

毕笃彦 (1965 —),男,空军工程大学工程学院四系教授,博士生导师,研究方向:图像处理及信号处理;

王振家 (1945 —),男,空军工程大学工程学院基础部电工教研室副教授、硕士生导师,研究方向:电路理论与非线性系统.